SIEMENS

SINUMERIK

SINUMERIK 840D sl PLC

Function Manual

Valid for

Control system
SINUMERIK 840D sl / 840DE sl

CNC software version 4.92

06/2019
A5E47433877B AB

Preface
Fundamental safety instructions 1
Overview 2
PLC mode selector 3
Reserve resources (timers, counters, FC, FB, DB, I/O) 4
Commissioning hardware configuration of the PLC CPU 5
Starting up the PLC program 6
Coupling of the PLC CPU 7
Interface structure 8
Structure and functions of the basic program 9
SPL for Safety Integrated 10
Assignment overview 11
PLC functions for HMI (DB19) 12
PLC functions for drive components on the integrated PROFIBUS 13
Memory requirements of the basic PLC program 14
NC VAR selector 15
Block descriptions 16
Data lists 17
Interface signals 18
Appendix A

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E47433877B AB
℗ 07/2019 Subject to change

Copyright © Siemens AG 2019.
All rights reserved

Preface

SINUMERIK documentation

The SINUMERIK documentation is organized into the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

Additional information

You can find information on the following topics at the following address (https://support.industry.siemens.com/cs/de/en/view/108464614):

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and search in manuals/information)

If you have any questions regarding the technical documentation (e.g. suggestions, corrections), please send an e-mail to the following address (mailto:docu.motioncontrol@siemens.com).

mySupport/Documentation

At the following address (https://support.industry.siemens.com/My/ww/en/documentation), you can find information on how to create your own individual documentation based on Siemens' content, and adapt it for your own machine documentation.

Training

At the following address (http://www.siemens.com/sitrain), you can find information about SITRAIN (Siemens training on products, systems and solutions for automation and drives).

FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

You can find information about SINUMERIK at the following address (http://www.siemens.com/sinumerik).

Target group

This publication is intended for:

- Project engineers
- Technologists (from machine manufacturers)
- System startup engineers (Systems/Machines)
- Programmers

Benefits

The function manual describes the functions so that the target group knows them and can select them. It provides the target group with the information required to implement the functions.

Standard version

This documentation only describes the functionality of the standard version. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Further, for the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation or maintenance.

Note regarding the General Data Protection Regulation

Siemens observes standard data protection principles, in particular the principle of privacy by design. That means that

this product does not process / store any personal data, only technical functional data (e.g. time stamps). If a user links this data with other data (e.g. a shift schedule) or stores personal data on the same storage medium (e.g. hard drive) and thus establishes a link to a person or persons, then the user is responsible for ensuring compliance with the relevant data protection regulations.

Technical Support

Country-specific telephone numbers for technical support are provided in the Internet at the following address (https://support.industry.siemens.com/sc/ww/en/sc/2090) in the "Contact" area.

Table of contents

Preface **3**

1 Fundamental safety instructions **25**

1.1 General safety instructions 25

1.2 Warranty and liability for application examples 25

1.3 Industrial security 26

2 Overview **29**

2.1 Brief description 29

2.2 Key data of the PLC CPU 31

2.3 PLC operating system version 32

3 PLC mode selector **33**

4 Reserve resources (timers, counters, FC, FB, DB, I/O) **35**

5 Commissioning hardware configuration of the PLC CPU **37**

6 Starting up the PLC program **39**

6.1 Installation of the basic program 39

6.2 Application of the basic program 39

6.3 Version codes 40

6.4 Machine program 41

6.5 Data backup 41

6.6 PLC series startup, PLC archive 42

6.7 Software upgrade 47

6.8 I/O modules (FM, CP modules) 48

6.9 Troubleshooting 49

7 Coupling of the PLC CPU **51**

7.1 General information 51

7.2 Properties of the PLC CPU 51

7.3 Interface with integrated PLC 51

7.4 Diagnostic buffer on PLC 53

8 Interface structure **55**

8.1 Interface 55

8.2 PLC/NCK interface 55

8.3 Interface PLC/HMI 62

8.4 PLC/MCP/HHU interface 68
9 Structure and functions of the basic program 71
9.1 Startup and synchronization of NCK PLC 73
9.2 Cyclic operation (OB1) 73
9.3 Time-interrupt processing (OB35) 75
9.4 Process-interrupt processing (OB 40) 76
9.5 Diagnostic alarm, module failure processing (OB82, OB86) 76
9.6 Response to NCK failure 77
9.7 Functions of the basic program called from the user program 78
9.8 Symbolic programming of user program with interface DB 81
9.9 M decoding acc. to list 83
9.10 PLC machine data 87
9.11 Configuration machine control panel, handheld unit, direct keys 91
9.12 Switchover of machine control panel, handheld unit 100
10 SPL for Safety Integrated 103
11 Assignment overview 105
11.1 Block address range for PLC user program 105
12 PLC functions for HMI (DB19) 109
12.1 Channel selection 109
12.2 Program selection 110
12.3 Activating the key lock 113
12.4 Operating area numbers 113
12.5 Screen numbers 113
12.5.1 Screen numbers: JOG, manual machine 114
12.5.2 Screen numbers: Reference point approach 119
12.5.3 Screen numbers: MDA 119
12.5.4 Screen numbers: AUTOMATIC 120
12.5.5 Screen numbers: Parameters operating area 120
12.5.6 Screen numbers: Program operating area 122
12.5.7 Screen numbers: Program manager operating area 122
12.5.8 Screen numbers: Diagnostics operating area 122
12.6 HMI monitor 123
13 PLC functions for drive components on the integrated PROFIBUS 125
13.1 Overview 125
13.2 Performing a start-up 125
13.3 Example 126

14 Memory requirements of the basic PLC program 129

15 NC VAR selector 133

15.1 Overview 133

15.2 Using NC VAR selector 135
15.2.1 NC VAR selector user interface 135
15.2.2 Description of functions 137
15.2.3 Editing projects 138
15.2.4 Filtering the variable list 139
15.2.5 Specifying the project list 141
15.2.6 Deleting a variable from the project list 142
15.2.7 Alias name 143
15.2.8 Generating code 143
15.2.9 Transfer to STEP 7 145
15.2.10 Using TIA Portal 146
15.2.11 Using SINAMICS parameters 146
15.2.12 Importing variables and parameters 147
15.2.13 Startup, installation 148

16 Block descriptions 149

16.1 FB1: RUN_UP - basic program, start section 149

16.2 FB2: GET - read NC variable 157

16.3 FB3: PUT - write NC variables 165

16.4 FB4: PI_SERV - request PI service 173
16.4.1 List of available PI services 177
16.4.2 PI service: ASUP 178
16.4.3 PI service: CANCEL 179
16.4.4 PI service: CONFIG 180
16.4.5 PI service: DIGION 180
16.4.6 PI service: DIGIOF 180
16.4.7 PI service: FINDBL 181
16.4.8 PI service: LOGIN 181
16.4.9 PI service: LOGOUT 181
16.4.10 PI service: NCRES 182
16.4.11 PI service: SELECT 182
16.4.12 PI service: SETUDT 183
16.4.13 PI service: SETUFR 183
16.4.14 PI service: RETRAC 183
16.4.15 PI service: CRCEDN 184
16.4.16 PI service: CREACE 185
16.4.17 PI service: CREATO 185
16.4.18 PI service: DELECE 186
16.4.19 PI service: DELETO 186
16.4.20 PI service: MMCSEM 186
16.4.21 PI service: TMCRTO 188
16.4.22 PI service: TMFDPL 189
16.4.23 PI service: TMFPBP 190
16.4.24 PI service: TMGETT 191
16.4.25 PI service: TMMVTL 192
16.4.26 PI service: TMPOSM 193

16.4.27 PI service: TMPCIT 194
16.4.28 PI service: TMRASS 195
16.4.29 PI service: TRESMO 195
16.4.30 PI service: TSEARC 196
16.4.31 PI service: TMCRMT 199
16.4.32 PI service: TMDLMT 200
16.4.33 PI service: POSMT 200
16.4.34 PI service: FDPLMT 201
16.5 FB5: GETGUD - read GUD variable 202
16.6 FB7: PI_SERV2 - request PI service 210
16.7 FB9: MtoN - operator panel switchover 211
16.8 FB10: Safety relay (SI relay) 216
16.9 FB11: Brake test 218
16.10 FB29: Signal recorder and data trigger diagnostics 224
16.11 FC2 : GP_HP - basic program, cyclic section 227
16.12 FC3: GP_PRAL - basic program, interruptdriven section 229
16.13 FC5: GP_DIAG - basic program, diagnostic alarm and module failure 231
16.14 FC6: TM_TRANS2 - transfer block for tool management and multitool 233
16.15 FC7: TM_REV - transfer block for tool change with revolver 234
16.16 FC8: TM_TRANS - transfer block for tool management 238
16.17 FC9: ASUP - start of asynchronous subprograms 244
16.18 FC10: AL_MSG - error and operating messages 247
16.19 FC12: AUXFU - call interface for user with auxiliary functions 249
16.20 FC13: BHGDisp - display control for handheld unit 250
16.21 FC17: YDelta - star-delta switchover 254
16.22 FC18: SpinCtrl - spindle control 257
16.23 FC19: MCP_IFM - transfer of MCP signals to interface 267
16.24 FC21: Transfer - data exchange NC/PLC 275
16.24.1 Function 275
16.24.2 Declaration of the function 275
16.24.3 Explanation of formal parameters 276
16.24.4 Function 1, 2: Signals synchronized actions to / from Channel 276
16.24.5 Function 3, 4: Fast data exchange PLC-NC 277
16.24.6 Function 5: Update control signals to channel 281
16.24.7 Function 6: Update control signals to axes 282
16.24.8 Function 7: Update control signals to axes 282
16.25 FC22: TM_DIR - direction selection for tool management 283
16.26 FC24: MCP_IFM2 - transferring MCP signals to the interface 285
16.27 FC25: MCP_IFT - transfer of MCP/OP signals to interface 289
16.28 FC26: HPU_MCP - Transfer of the HT 8/HT 10 signals to the interface 292

16.28.1 Overview of the NC/PLC interface signals of HT 8 297
16.28.2 Overview of the NC/PLC interface signals of HT 8 299
16.29 FC1005: AG_SEND - transfers data to Ethernet CP 299
16.30 FC1006: AG_RECV - receives data from the Ethernet CP 300
17 Data lists 303
17.1 Machine data 303
17.1.1 Display machine data 303
17.1.2 NC-specific machine data 303
17.1.3 Channelspecific machine data 303
18 Interface signals 305
18.1 Interface signals - overview 305
18.1.1 Overview of the PLC blocks 305
18.1.1.1 Organization blocks (OBs) 305
18.1.1.2 Function blocks (FBs) 305
18.1.1.3 Function blocks (FCs) 306
18.1.1.4 Data blocks (DBs) 307
18.1.1.5 Timer block 308
18.1.2 Signals from/to the machine control panel 308
18.1.2.1 M version, signals from the MCP: Input image 308
18.1.2.2 M version, signals to the MCP: Output image 309
18.1.2.3 T version, signals from the MCP: Input image 309
18.1.2.4 T version, signals to the MCP: Output image 310
18.1.2.5 Slimline version, signals from the MCP: Input image 310
18.1.2.6 Slimline version, signals to the MCP: Output image 311
18.1.3 Signals from/to the handheld unit HT 2 312
18.1.3.1 Signals from the handheld unit: Input image 312
18.1.3.2 Signals to the handheld unit: Output image 312
18.1.4 Signals from/to the handheld unit HT 8 314
18.1.4.1 Signals from the handheld unit HT 8: Input image 314
18.1.4.2 Signals to handheld terminal HT 8: Output image 314
18.1.5 Signals from/to the handheld unit HT 10 315
18.1.5.1 Signals from the handheld unit HT 10: Input image 315
18.1.5.2 Signals to handheld terminal HT 10: Output image 316
18.1.6 PLC alarms/messages 317
18.1.6.1 FC 10 alarms in the DB2 (FB1: "ExtendAIMsg" = FALSE) 317
18.1.6.2 FC 10 alarms in the DB2 (FB1: "ExtendAIMsg" = TRUE) 326
18.1.7 Signals from/to the NC, PLC and operating software 339
18.1.7.1 DB10, onboard inputs and outputs of the NC 339
18.1.7.2 DB10, general signals to the NC 340
18.1.7.3 DB10, onboard inputs and outputs from the NC/operating software 341
18.1.7.4 DB10, selection and status signals from the operating software 342
18.1.7.5 DB10, general signals from the NC 344
18.1.7.6 DB10, external digital NC inputs 345
18.1.7.7 DB10, external digital NC outputs 346
18.1.7.8 DB10, external analog NC inputs 347
18.1.7.9 DB10, external analog NC outputs 348
18.1.7.10 DB10, external digital NC inputs and outputs 349
18.1.7.11 DB10, analog NC inputs and outputs 350
18.1.7.12 DB10, collision avoidance: Protection area active 351

18.1.7.13 DB10, collision avoidance: Activate protection area 351
18.1.7.14 DB10, extension, handwheel signals from the NC 352
18.1.7.15 DB10, interface robot status 353
18.1.7.16 DB10, interface robot status 353
18.1.8 Mode group-specific signals 353
18.1.8.1 DB11, mode signals 1 to the NC 353
18.1.8.2 DB11, mode signals 1 from the NC 354
18.1.8.3 DB11, mode signals 2 to the NC 355
18.1.8.4 DB11, mode signals 2 from NC 356
18.1.9 Safety Integrated (SPL) 357
18.1.9.1 DB18, parameterization part 357
18.1.9.2 DB18, data area / errors 357
18.1.9.3 DB18, additional data areas 359
18.1.9.4 DB18, F_SENDDP sender 360
18.1.9.5 DB18, F_SENDDP receiver 360
18.1.9.6 DB18, SPL user data 361
18.1.9.7 DB18, data area / errors: Extended data area 362
18.1.9.8 DB18, additional data areas: Extended data area 364
18.1.9.9 DB18, F_SENDDP sender 365
18.1.9.10 DB18, F_RECDP receiver 367
18.1.10 Control/status signals to/from operator panel (OP) 371
18.1.10.1 DB19, signals to the operator panel (OP) 371
18.1.10.2 DB19, signals from the operator panel (OP) 373
18.1.10.3 DB19, sidescreen - interface for the MCP function 377
18.1.10.4 DB19, sidescreen - interface for the MCP function 377
18.1.11 Defining PLC alarms 378
18.1.11.1 DB20, NC machine data 378
18.1.12 Channel-specific signals 379
18.1.12.1 DB21 - DB30, control signals to the channel (1) 379
18.1.12.2 DB21 - DB30, control signals to the geometry axes 381
18.1.12.3 DB21 - DB30, HMI signals to channel / OEM signals from/to channel 383
18.1.12.4 DB21 - DB30, control signals from the geometry axes 385
18.1.12.5 DB21 - DB30, change signals for auxiliary function transfer from the channel 387
18.1.12.6 DB21 - DB30, transferred M and S functions: 388
18.1.12.7 DB21 - DB30, transferred T/D/DL functions 389
18.1.12.8 DB21 - DB30, transferred H/F functions 390
18.1.12.9 DB21 - DB30, decoded M signals 391
18.1.12.10 DB21 - DB30, active G functions 392
18.1.12.11 DB21 - DB30, protection areas from the channel 393
18.1.12.12 DB21 - DB30, synchronous actions, signals from/to the channel 394
18.1.12.13 DB21 - DB30, control signals from/to the channel 395
18.1.12.14 DB21 - DB30, signals to the orientation axes 396
18.1.12.15 DB21 - DB30, signals from the orientation axes 398
18.1.12.16 DB21 - DB30, tool management functions from the channel 399
18.1.12.17 DB21 - DB30, control signals from/to the channel (2) 400
18.1.13 Axis/spindle signals 403
18.1.13.1 DB31 - DB61, signals to the axis/spindle 403
18.1.13.2 DB31 - DB61, signals from the axis/spindle 408
18.1.14 Safety Integrated 416
18.1.14.1 DB31 - DB61, Safety Control Channel (SCC) 416
18.1.14.2 DB31 - DB61, Safety Info Channel (SIC) 416
18.1.15 Tool management 417

18.1.15.1 DB71, interface for loading/unloading the magazine 417
18.1.15.2 DB72, interface for the spindle as change position 419
18.1.15.3 DB73, interface for the turret 421
18.1.16 Signals from/to the machine control panel and the handheld unit 423
18.1.16.1 DB77, signals from/to the MCP and the HHU 423
18.1.17 Signals for Ctrl-Energy 424
18.1.17.1 DB1000, energy-saving profiles 424
18.1.18 SENTRON PAC 426
18.1.18.1 DB1001, SENTRON PAC 426
18.1.18.2 DB1001, SENTRON PAC, auxiliary devices 428
18.1.19 Spindle temperature sensor 429
18.1.19.1 DB1002: spindle temperature sensors 429
18.1.20 Interface to the tool management, extended area 431
18.1.20.1 DB1071, interface for loading/unloading the magazine Multitool 431
18.1.20.2 DB1072, interface for the spindle: Multitool 432
18.1.20.3 DB1073, interface for the turret: Multitool 433

18.2 Interface signals - detailed description 434
18.2.1 DB10: NC, PLC and HMI 434
18.2.1.1 DB10 DBX0.0 - 7 (digital NC inputs 1 - 8: Lock) 434
18.2.1.2 DB10 DBX1.0 - 7 (digital NC inputs 1 - 8: Set) 435
18.2.1.3 DB10 DBX4.0 - 7 (digital NC outputs 1 - 8: Lock) 435
18.2.1.4 DB10 DBX5.0 - 7 (digital NC outputs 1 - 8: Overwrite) 436
18.2.1.5 DB10 DBX6.0 - 7 (digital NC outputs 1 - 8: Setting value) 436
18.2.1.6 DB10 DBX7.0 - 7 (digital NC outputs 1 - 8: Target) 437
18.2.1.7 DB10 DBX56.1 (Emergency Stop) 438
18.2.1.8 DB10 DBX56.2 (acknowledge Emergency Stop) 438
18.2.1.9 DB10 DBX56.4 - 7 (key-operated switch position 0 - 3) 438
18.2.1.10 DB10 DBX58.0 - 7 (collision avoidance: Deactivate protection area group) 439
18.2.1.11 DB10 DBX60.0 - 7 (digital NC inputs 1 - 8: Actual value) 440
18.2.1.12 DB10 DBX64.0 - 7 (digital NC outputs 1 - 8: Setpoint) 441
18.2.1.13 DB10 DBX93.0 ... 7 (collision avoidance: Deactivate protection area group) 441
18.2.1.14 DB10 DBX97.0 - 3 (channel number geometry axis handwheel 1) 442
18.2.1.15 DB10 DBX98.0 - 3 (channel number geometry axis handwheel 2) 443
18.2.1.16 DB10 DBX99.0 - 3 (channel number geometry axis handwheel 3) 443
18.2.1.17 DB10 DBX100.0 - 4 (axis number handwheel 1) 444
18.2.1.18 DB10 DBX100.5 (define handwheel 1 as contour handwheel) 444
18.2.1.19 DB10 DBX100.6 (handwheel 1 selected) 445
18.2.1.20 DB10 DBX100.7 (machine axis handwheel 1) 446
18.2.1.21 DB10 DBX101.0 - 4 (axis number handwheel 2) 446
18.2.1.22 DB10 DBX101.5 (define handwheel 2 as contour handwheel) 446
18.2.1.23 DB10 DBX101.6 (handwheel 2 selected) 446
18.2.1.24 DB10 DBX101.7 (machine axis handwheel 2) 447
18.2.1.25 DB10 DBX102.0 - 4 (axis number handwheel 3) 447
18.2.1.26 DB10 DBX102.5 (define handwheel 3 as contour handwheel) 447
18.2.1.27 DB10 DBX102.6 (handwheel 3 selected) 447
18.2.1.28 DB10 DBX102.7 (machine axis handwheel 3) 447
18.2.1.29 DB10 DBX103.0 (remote diagnosis active) 447
18.2.1.30 DB10 DBX103.5 (AT box ready) 447
18.2.1.31 DB10 DBX103.6 (HMI temperature limit) 448
18.2.1.32 DB10 DBX103.7 (HMI battery alarm) 448
18.2.1.33 DB10 DBX104.7 (NC-CPU ready) 448
18.2.1.34 DB10 DBX106.1 (Emergency Stop active) 449

18.2.1.35 DB10 DBX107.0 - 1 (probe actuated) 449
18.2.1.36 DB10 DBX107.6 (NCU link active) 449
18.2.1.37 DB10 DBX108.3 (operating software ready) 450
18.2.1.38 DB10 DBX108.5 (drives in cyclic operation) 450
18.2.1.39 DB10 DBX108.6 (drive ready) 450
18.2.1.40 DB10 DBX108.7 (NC ready) 450
18.2.1.41 DB10 DBX109.0 (NC alarm active) 451
18.2.1.42 DB10 DBX109.5 (heat sink temperature alarm, NCU) 451
18.2.1.43 DB10 DBX109.6 (air temperature alarm) 452
18.2.1.44 DB10 DBX109.7 (NC battery alarm) 452
18.2.1.45 DB10 DBX110.0 - 113.7 (software cams: minus cam signal 1 to 32) 452
18.2.1.46 DB10 DBX114.0 - 117.7 (software cams: plus cam signal 1 to 32) 453
18.2.1.47 DB10 DBX122.0 - 7 (digital NC inputs 9 - 16: Lock) 454
18.2.1.48 DB10 DBX123.0 - 7 (digital NC inputs 9 - 16: Set) 454
18.2.1.49 DB10 DBX124.0 - 7 (digital NC inputs 17 - 24: Lock) 454
18.2.1.50 DB10 DBX125.0 - 7 (digital NC inputs 17 - 24: Set) 455
18.2.1.51 DB10 DBX126.0 - 7 (digital NC inputs 25 - 32: Lock) 455
18.2.1.52 DB10 DBX127.0 - 7 (digital NC inputs 25 - 32: Set) 456
18.2.1.53 DB10 DBX128.0 - 7 (digital NC inputs 33 - 40: Lock) 456
18.2.1.54 DB10 DBX129.0 - 7 (digital NC inputs 33 - 40: Set) 456
18.2.1.55 DB10 DBX130.0 - 7 (digital NC outputs 9 - 16: Lock) 457
18.2.1.56 DB10 DBX131.0 - 7 (digital NC outputs 9 - 16: Overwrite) 457
18.2.1.57 DB10 DBX132.0 - 7 (digital NC outputs 9 - 16: Setting value) 458
18.2.1.58 DB10 DBX133.0 - 7 (digital NC outputs 9 - 16: Target) 459
18.2.1.59 DB10 DBX134.0 - 7 (digital NC outputs 17 - 24: Lock) 459
18.2.1.60 DB10 DBX135.0 - 7 (digital NC outputs 17 - 24: Overwrite) 460
18.2.1.61 DB10 DBX136.0 - 7 (digital NC outputs 17 - 24: Setting value) 460
18.2.1.62 DB10 DBX137.0 - 7 (digital NC outputs 17 - 24: Target) 461
18.2.1.63 DB10 DBX138.0 - 7 (digital NC outputs 25 - 32: Lock) 461
18.2.1.64 DB10 DBX139.0 - 7 (digital NC outputs 25 - 32: Overwrite) 462
18.2.1.65 DB10 DBX140.0 - 7 (digital NC outputs 25 - 32: Setting value) 462
18.2.1.66 DB10 DBX141.0 - 7 (digital NC outputs 25 - 32: Target) 463
18.2.1.67 DB10 DBX142.0 - 7 (digital NC outputs 33 - 40: Lock) 464
18.2.1.68 DB10 DBX143.0 - 7 (digital NC outputs 33 - 40: Overwrite) 464
18.2.1.69 DB10 DBX144.0 - 7 (digital NC outputs 33 - 40: Setting value) 465
18.2.1.70 DB10 DBX145.0 - 7 (digital NC outputs 33 - 40: Target) 465
18.2.1.71 DB10 DBX146.0 - 7 (analog NC inputs 1 - 8: Lock) 466
18.2.1.72 DB10 DBX147.0 - 7 (analog NC inputs 1 - 8: Target) 466
18.2.1.73 DB10 DBW148 - 162 (analog NC outputs 1 - 8: Setting value) 467
18.2.1.74 DB10 DBX166.0 - 7 (analog NC outputs 1 - 8: Overwrite) 467
18.2.1.75 DB10 DBX167.0 - 7 (analog NC outputs 1 - 8: Target) 468
18.2.1.76 DB10 DBX168.0 - 7 (analog NC outputs 1 - 8: Lock) 468
18.2.1.77 DB10 DBW170 - 184 (analog NC outputs 1 - 8: Setting value) 469
18.2.1.78 DB10 DBX186.0 - 7 (digital NC inputs 9 - 16: Actual value) 469
18.2.1.79 DB10 DBX187.0 - 7 (digital NC inputs 17 - 24: Actual value) 470
18.2.1.80 DB10 DBX188.0 - 7 (digital NC inputs 25 - 32: Actual value) 470
18.2.1.81 DB10 DBX189.0 - 7 (digital NC inputs 33 - 40: Actual value) 471
18.2.1.82 DB10 DBX190.0 - 7 (digital NC outputs 9 - 16: Setpoint) 471
18.2.1.83 DB10 DBX191.0 - 7 (digital NC outputs 17 - 24: Setpoint) 472
18.2.1.84 DB10 DBX192.0 - 7 (digital NC outputs 25 - 32: Setpoint) 472
18.2.1.85 DB10 DBX193.0 - 7 (digital NC outputs 33 - 40: Setpoint) 473
18.2.1.86 DB10 DBW194 - 208 (analog NC inputs 1 - 8: Actual value) 473

18.2.1.87 DB10 DBW210 - 224 (analog NC outputs 1 - 8: Setpoint) 473
18.2.1.88 DB10 DBX226.0 - 233.7 (collision avoidance: Protection area active) 474
18.2.1.89 DB10 DBX234.0 - 241.7 (collision avoidance: Activate protection area) 475
18.2.1.90 DB10 DBX245.0 - 5 (Ethernet handwheel stationary) 475
18.2.2 DB11: Mode group 476
18.2.2.1 DB11 DBX0.0 (AUTOMATIC mode) 476
18.2.2.2 DB11 DBX0.1 (MDI mode) 476
18.2.2.3 DB11 DBX0.2 (JOG mode) 476
18.2.2.4 DB11 DBX0.4 (mode change inhibit) 477
18.2.2.5 DB11 DBX0.5 (mode group stop) 477
18.2.2.6 DB11 DBX0.6 (mode group stop, axes plus spindles) 478
18.2.2.7 DB11 DBX0.7 (mode group reset) 478
18.2.2.8 DB11 DBX1.0 (TEACH IN machine function) 479
18.2.2.9 DB11 DBX1.1 (REPOS machine function) 479
18.2.2.10 DB11 DBX1.2 (REF machine function) 480
18.2.2.11 DB11 DBX1.6 (single block, type B) 480
18.2.2.12 DB11 DBX1.7 (single block, type A) 480
18.2.2.13 DB11 DBX4.0 (AUTOMATIC mode selected) 481
18.2.2.14 DB11 DBX4.1 (MDI mode selected) 481
18.2.2.15 DB11 DBX4.2 (JOG mode selected) 481
18.2.2.16 DB11 DBX5.0 (TEACH IN machine function selected) 482
18.2.2.17 DB11 DBX5.1 (REPOS machine function selected) 482
18.2.2.18 DB11 DBX5.2 (REF machine function selected) 482
18.2.2.19 DB11 DBX6.0 (AUTOMATIC mode active) 483
18.2.2.20 DB11 DBX6.1 (active MDI mode) 483
18.2.2.21 DB11 DBX6.2 (JOG mode active) 483
18.2.2.22 DB11 DBX6.3 (mode group ready) 483
18.2.2.23 DB11 DBX6.7 (all channels in the "Reset" state) 484
18.2.2.24 DB11 DBX7.0 (TEACH IN machine function active) 484
18.2.2.25 DB11 DBX7.1 (REPOS machine function active) 484
18.2.2.26 DB11 DBX7.2 (REF machine function active) 485
18.2.3 DB18: Safety Integrated, SPL 485
18.2.3.1 DB18 DBB36.0 (SPL_READY) 485
18.2.3.2 DB18 DBB36.1 (STOP_E) 485
18.2.3.3 DB18 DBB38 - 41 (SPL inputs, SPL_DATA.INSEP[1...32]) 486
18.2.3.4 DB18 DBB42.0 - 45.7 (SPL inputs, SPL_DATA.INSEP[33...64]) 486
18.2.3.5 DB18 DBB46.0 - 49.7 (SPL outputs, SPL_DATA.OUTSEP[1...32] 486
18.2.4 DB19: Operator panel 487
18.2.4.1 DB19 DBX0.0 (brighten screen) 487
18.2.4.2 DB19 DBX0.1 (darken screen) 487
18.2.4.3 DB19 DBX0.2 (key lock) 488
18.2.4.4 DB19 DBX0.3 (delete cancel alarms) 488
18.2.4.5 DB19 DBX0.4 (delete recall alarms) 489
18.2.4.6 DB19 DBX0.7 (actual values in the WCS) 489
18.2.4.7 DB19 DBB6 (analog spindle 1, utilization as a percentage) 489
18.2.4.8 DB19 DBB7 (analog spindle 2, utilization as a percentage) 489
18.2.4.9 DB19 DBB8 (channel number) 490
18.2.4.10 DB19 DBB10 (PLC hardkeys) 490
18.2.4.11 DB19 DBX13.5 (NC program: unload) 490
18.2.4.12 DB19DBX13.6 (NC program: load) 490
18.2.4.13 DB19 DBX13.7 (NC program: selection) 490
18.2.4.14 DB19 DBX14.0 - 6 (PLC index) 491

18.2.4.15 DB19 DBX14.7 (selects the file system) 491
18.2.4.16 DB19 DBB15 (PLC line offset) 491
18.2.4.17 DB19 DBB16 (program selection from the PLC: Index of the program list) 491
18.2.4.18 DB19 DBB17 (program selection from the PLC: Program index in the program list) 492
18.2.4.19 DB19 DBX20.1 (screen is dark) 493
18.2.4.20 DB19 DBX20.3 (cancel alarms deleted) 493
18.2.4.21 DB19 DBX20.4 (recall alarms deleted) 493
18.2.4.22 DB19 DBX20.6 (simulation active) 494
18.2.4.23 DB19 DBX20.7 (switch MCS/WCS) 494
18.2.4.24 DB19 DBB22 (actual channel number) 494
18.2.4.25 DB19 DBW24 (current screen number) 494
18.2.4.26 DB19 DBX26.1 (program selection from the PLC: Job completed) 495
18.2.4.27 DB19 DBX26.2 (program selection from the PLC: Error) 495
18.2.4.28 DB19 DBX26.3 (program selection from the PLC: Active) 495
18.2.4.29 DB19 DBX26.5 (program selection from the PLC: Unload) 496
18.2.4.30 DB19 DBX26.6 (program selection from the PLC: Load) 496
18.2.4.31 DB19 DBX26.7 (program selection from the PLC: Selection) 497
18.2.4.32 DB19 DBB27 (program selection from the PLC: Error detection) 497
18.2.4.33 DB19 DBX32.0 - 5 (function number) 498
18.2.4.34 DB19 DBX32.6 (function request) 499
18.2.4.35 DB19 DBX32.7 (status) 499
18.2.4.36 DB19 DBB33 - 35 (parameter 1 - 3) 499
18.2.4.37 DB19 DBB36 (error identification) 500
18.2.5 DB21, ...: Channel 500
18.2.5.1 DB21, ... DBX0.1 (RESU: backward/forward) 500
18.2.5.2 DB21, ... DBX0.2 (RESU: start retrace support) 501
18.2.5.3 DB21, ... DBX0.3 (activate handwheel offset (DRF)) 501
18.2.5.4 DB21, ... DBX0.4 (activate single block) 501
18.2.5.5 DB21, ... DBX0.5 (activate M01) 502
18.2.5.6 DB21, ... DBX0.6 (activate dry run feedrate) 502
18.2.5.7 DB21, ... DBX1.0 (activate referencing) 503
18.2.5.8 DB21, ... DBX2.1 (disable all synchronized actions) 504
18.2.5.9 DB21, ... DBX1.3 (time monitoring active) 504
18.2.5.10 DB21, ... DBX1.4 (clearance control (CLC): stop) 504
18.2.5.11 DB21, ... DBX1.5 (clearance control (CLC): Override) 505
18.2.5.12 DB21, ... DBX1.6 (PLC action completed) 505
18.2.5.13 DB21, ... DBX1.7 (activate program test (PRT)) 506
18.2.5.14 DB21, ... DBX2.0 ... 7 (activate "skip block" (SKP)) 506
18.2.5.15 DB21, ... DBX3.0 (stroke enable) 507
18.2.5.16 DB21, ... DBX3.1 (punch interface 1: Manual stroke initiation) 507
18.2.5.17 DB21, ... DBX3.2 (stroke suppression) 508
18.2.5.18 DB21, ... DBX3.3 (delayed stroke) 508
18.2.5.19 DB21, ... DBX3.4 (stroke inoperative) 508
18.2.5.20 DB21, ... DBX3.5 (punch interface 2: Manual stroke initiation) 509
18.2.5.21 DB21, ... DBB4 (path feedrate override) 509
18.2.5.22 DB21, ... DBB5 (path rapid traverse override) 511
18.2.5.23 DB21, ... DBX6.0 (feedrate disable) 513
18.2.5.24 DB21, ... DBX6.1 (read-in disable) 514
18.2.5.25 DB21, ... DBX6.2 (delete distance-to-go, channel-specific) 514
18.2.5.26 DB21, ... DBX6.4 (program level abort) 515
18.2.5.27 DB21, ... DBX6.6 (path rapid traverse override active) 515
18.2.5.28 DB21, ... DBX6.7 (path feedrate override active) 516

18.2.5.29 DB21, ... DBX7.0 (NC start disable) ... 516
18.2.5.30 DB21, ... DBX7.1 (NC Start) ... 517
18.2.5.31 DB21, ... DBX7.2 (NC Stop at the block limit) ... 517
18.2.5.32 DB21, ... DBX7.3 (NC Stop) ... 518
18.2.5.33 DB21, ... DBX7.4 (NC Stop axes plus spindles) ... 518
18.2.5.34 DB21, ... DBX7.6 (activate configured stop) ... 519
18.2.5.35 DB21, ... DBX7.7 (reset) ... 519
18.2.5.36 DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10) ... 520
18.2.5.37 DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10) ... 520
18.2.5.38 DB21, ... DBX12.0 - 2 (geometry axis 1: activate handwheel) ... 521
18.2.5.39 DB21, ... DBX12.3, 16.3, 20.3 (feedrate stop, geometry axes 1 / 2 / 3) ... 522
18.2.5.40 DB21, ... DBX12.4 (geometry axis 1: traversing key lock) ... 522
18.2.5.41 DB21, ... DBX12.5 (geometry axis 1: rapid traverse override) ... 523
18.2.5.42 DB21, ... DBX12.6 - 7 (geometry axis 1: traversing keys "Plus" / "Minus") ... 523
18.2.5.43 DB21, ... DBX13.0 - 6 (geometry axis 1: machine function request) ... 525
18.2.5.44 DB21, ... DBX15.0 (geometry axis 1: invert handwheel direction of rotation) ... 526
18.2.5.45 DB21, ... DBX16.0 - 2 (geometry axis 2: activate handwheel) ... 526
18.2.5.46 DB21, ... DBX16.4 (geometry axis 2: traversing key lock) ... 527
18.2.5.47 DB21, ... DBX16.5 (geometry axis 2: rapid traverse override) ... 527
18.2.5.48 DB21, ... DBX16.6 - 7 (geometry axis 2: traversing keys "Plus" / "Minus") ... 527
18.2.5.49 DB21, ... DBX17.0 - 6 (geometry axis 2: machine function request) ... 527
18.2.5.50 DB21, ... DBX19.0 (geometry axis 2: invert handwheel direction of rotation) ... 527
18.2.5.51 DB21, ... DBX20.0 - 2 (geometry axis 3: activate handwheel) ... 527
18.2.5.52 DB21, ... DBX20.4 (geometry axis 3: traversing key lock) ... 527
18.2.5.53 DB21, ... DBX20.5 (geometry axis 3: rapid traverse override) ... 528
18.2.5.54 DB21, ... DBX20.6 - 7 (geometry axis 3: traversing keys "Plus" / "Minus") ... 528
18.2.5.55 DB21, ... DBX21.0 - 6 (geometry axis 3: machine function request) ... 528
18.2.5.56 DB21, ... DBX23.0 (geometry axis 3: invert handwheel direction of rotation) ... 528
18.2.5.57 DB21, ... DBX24.1 (activate configured stop (CST) using the operating software) ... 528
18.2.5.58 DB21, ... DBX24.3 (handwheel offset (DRF) requested) ... 529
18.2.5.59 DB21, ... DBX24.4 (select NC-associated M01) ... 529
18.2.5.60 DB21, ... DBX24.5 (M01 requested) ... 530
18.2.5.61 DB21, ... DBX24.6 (dry run feedrate selected) ... 530
18.2.5.62 DB21, ... DBX25.3 (feedrate override selected for rapid traverse) ... 531
18.2.5.63 DB21, ... DBX25.7 (program test (PRT) requested) ... 531
18.2.5.64 DB21, ... DBX26.0 ... 7 ("Skip block" (SKP) selected) ... 532
18.2.5.65 DB21, ... DBX29.0 - 3 (activate fixed feedrate 1 - 4, path/geometry axes) ... 533
18.2.5.66 DB21, ... DBX29.4 (activate PTP travel) ... 534
18.2.5.67 DB21, ... DBX29.5 (tool manager: Deactivate workpiece counter) ... 534
18.2.5.68 DB21, ... DBX29.6 (tool manager: Deactivating wear monitoring) ... 534
18.2.5.69 DB21, ... DBX29.7 (tool manager: Tool lock not active) ... 535
18.2.5.70 DB21 ... DBX30.0 - 2 (activate contour handwheel) ... 535
18.2.5.71 DB21, ... DBX30.3 (simulation contour handwheel: Activate) ... 536
18.2.5.72 DB21, ... DBX30.4 (simulation contour handwheel: Negative direction) ... 537
18.2.5.73 DB21, ... DBX30.5 (activate M0 / M1 associated with NC) ... 537
18.2.5.74 DB21, ... DBX30.6 (JOG circular travel) ... 537
18.2.5.75 DB21, ... DBX31.0 - 2 (REPOS mode) ... 538
18.2.5.76 DB21, ... DBX31.4 (REPOS activation) ... 539
18.2.5.77 DB21, ... DBX31.5 (contour handwheel: invert handwheel direction of rotation) ... 539
18.2.5.78 DB21, ... DBX32.1 (RESU: retrace mode active) ... 539
18.2.5.79 DB21, ... DBX32.2 (retrace support active) ... 540
18.2.5.80 DB21, ... DBX32.3 (action block active) ... 540

18.2.5.81 DB21, ... DBX32.4 (approach block active) 540
18.2.5.82 DB21, ... DBX32.5 (M00/M01 active) 541
18.2.5.83 DB21, ... DBX32.6 (last action block active) 541
18.2.5.84 DB21, ... DBX33.0 (referencing active) 542
18.2.5.85 DB21, ... DBX33.2 (revolutional feedrate active) 542
18.2.5.86 DB21, ... DBX33.3 (handwheel override active) 542
18.2.5.87 DB21, ... DBX33.4 (block search active) 543
18.2.5.88 DB21, ... DBX33.5 (M02 / M30 active) 543
18.2.5.89 DB21, … DBX33.6 (transformation active) 544
18.2.5.90 DB21, ... DBX33.7 (program test active) 544
18.2.5.91 DB21, ... DBX35.0 (program state "running") 545
18.2.5.92 DB21, ... DBX35.1 (program state "wait") 546
18.2.5.93 DB21, ... DBX35.2 (program state "stopped") 547
18.2.5.94 DB21, ... DBX35.3 (program state "interrupted") 547
18.2.5.95 DB21, ... DBX35.4 (program state "aborted") 548
18.2.5.96 DB21, ... DBX35.5 (channel state "active") 549
18.2.5.97 DB21, ... DBX35.6 (channel state "interrupted") 549
18.2.5.98 DB21, ... DBX35.7 (channel state "reset") 550
18.2.5.99 DB21, ... DBX36.2 (all axes that have to be referenced have been referenced) 550
18.2.5.100 DB21, ... DBX36.3 (all axes stationary) 551
18.2.5.101 DB21, ... DBX36.4 (interrupt handling active) 551
18.2.5.102 DB21, ... DBX36.5 (channel ready) 551
18.2.5.103 DB21, ... DBX36.6 (channel-specific NC alarm active) 552
18.2.5.104 DB21, ... DBX36.7 (NC alarm with machining stop active) 552
18.2.5.105 DB21, ... DBX37.0 - 2 (contour handwheel active) 552
18.2.5.106 DB21, ... DBX37.3 (clearance control (CLC): Active) 553
18.2.5.107 DB21, ... DBX37.4 (clearance control (CLC): Stop at the lower motion limit) 554
18.2.5.108 DB21, ... DBX37.5 (clearance control (CLC): Stop at the upper motion limit) 554
18.2.5.109 DB21, ... DBX37.6 (read-in disable is ignored) 555
18.2.5.110 DB21, ... DBX37.7 (stop at block end is ignored during single block (SBL)) 555
18.2.5.111 DB21, ... DBX38.0 (stroke initiation active) 556
18.2.5.112 DB21, ... DBX38.1 (manual stroke initiation: Acknowledgment) 556
18.2.5.113 DB21, ... DBX39.1 (NC alarm with program stop) 557
18.2.5.114 DB21, ... DBX39.4 (stop at block end due to single block) 557
18.2.5.115 DB21, ... DBX39.5 (Contour handwheel: handwheel direction of rotation inversion active) 557
18.2.5.116 DB21, ... DBX39.6 (configured stop is activated) 558
18.2.5.117 DB21, ... DBX39.7 (stop at block end due to configured stop) 558
18.2.5.118 DB21, ... DBX40.0 - 2 (geometry axis 1: handwheel active) 558
18.2.5.119 DB21, ... DBX40.4 - 5 (geometry axis 1: traversing request "Plus" / "Minus") 559
18.2.5.120 DB21, ... DBX40.6 - 7 (geometry axis 1: traversing command "Plus" / "Minus") 560
18.2.5.121 DB21, ... DBX41.0 - 6 (geometry axis 1: active machine function) 561
18.2.5.122 DB21, ... DBX43.0 (geometry axis 1: handwheel direction of rotation inversion active) 561
18.2.5.123 DB21, ... DBX46.0 - 2 (geometry axis 2: handwheel active) 562
18.2.5.124 DB21, ... DBX46.4 - 5 (geometry axis 2: traversing request "Plus" / "Minus") 562
18.2.5.125 DB21, ... DBX46.6 - 7 (geometry axis 2: traversing command "Plus" / "Minus") 562
18.2.5.126 DB21, ... DBX47.0 - 6 (geometry axis 2: active machine function) 562
18.2.5.127 DB21, ... DBX49.0 (geometry axis 2: handwheel direction of rotation inversion active) 562
18.2.5.128 DB21, ... DBX52.0 - 2 (geometry axis 3: handwheel active) 563
18.2.5.129 DB21, ... DBX52.4 - 5 (geometry axis 3: traversing request "Plus" / "Minus") 563
18.2.5.130 DB21, ... DBX52.6 - 7 (geometry axis 3: traversing command "Plus" / "Minus") 563
18.2.5.131 DB21, ... DBX53.0 - 6 (geometry axis 3: active machine function) 563
18.2.5.132 DB21, ... DBX55.0 (geometry axis 3: handwheel direction of rotation inversion active) 563

18.2.5.133 DB21, ... DBB58, DBB60 - 65 (M-, S-, T-, D-, H-, F fct. change) 563
18.2.5.134 DB21, ... DBX59.0 - 4 (M fct. 1-5 not decoded) 564
18.2.5.135 DB21, ... DBB60 - 64, ... DBB66 - 67 (M-, S-, T-, D-, H-, F fct. Additional quick information (quick acknowledgment) 564
18.2.5.136 DB21, ... DBB68 - 97 (M function 1 - 5 and extended address M function 1 - 5) 564
18.2.5.137 DB21, ... DBB98 - 115 (S function 1 - 3 and extended address S function 1 - 3) 565
18.2.5.138 DB21, ... DBB118 (T function 1)....565
18.2.5.139 DB21, ... DBB129 (D function 1) 566
18.2.5.140 DB21, ... DBB140 - 157 (H function 1 - 3 and extended address H function 1 - 3)....566
18.2.5.141 DB21, ... DBB158 - 193 (F function 1 - 6 and extended address F function 1 - 6)....566
18.2.5.142 DB21, ... DBB194 - 206 (dynamic M functions: M0 - M99) 567
18.2.5.143 DB21, ... DBB208 - 271 (active G command of group 1 to 60) 567
18.2.5.144 DB21, ... DBX272.0 - 273.1 (machine-related protection zone 1 - 10 preactivated) 568
18.2.5.145 DB21, ... DBX274.0 – 275.1 (channel-specific protection zone 1 - 10 preactivated) 569
18.2.5.146 DB21, ... DBX276.0 - 277.1 (machine-related protection zone 1 - 10 violated) 569
18.2.5.147 DB21, ... DBX278.0 – 279.1 (channel-specific protection zone 1 - 10 violated)....570
18.2.5.148 DB21, ... DBX280.1 (request disabling of synchronized actions)....570
18.2.5.149 DB21, ... DBX281.1 (acknowledgement of synchronized actions disabled) 570
18.2.5.150 DB21, … DBX300.0 - 307.7 (disable synchronized action ID/IDS)....571
18.2.5.151 DB21, ... DBX308.0 - 315.7 (synchronized actions ID/IDS can be disabled)....571
18.2.5.152 DB21, ... DBB317.1 (workpiece setpoint reached)....571
18.2.5.153 DB21, ... DBX317.6 (PTP traversing active) 572
18.2.5.154 DB21, ... DBX317.7 (TOOLMAN: Tool missing) 572
18.2.5.155 DB21, ... DBX318.0 (ASUB is stopped) 572
18.2.5.156 DB21, ... DBX318.1 (block search via program test is active (SERUPRO)) 573
18.2.5.157 DB21, ... DBX318.2 (online tool length compensation (TOFF) active)....573
18.2.5.158 DB21, ... DBX318.3 (online tool length compensation (TOFF): Compensation motion active)....573
18.2.5.159 DB21, ... DBX318.5 (associated M0 / M1 active)....574
18.2.5.160 DB21, ... DBX319.0 (REPOS mode change acknowledgment)....574
18.2.5.161 DB21, ... DBX319.1 - 3 (active REPOS mode) 575
18.2.5.162 DB21, ... DBX319.5 (REPOS delay) 576
18.2.5.163 DB21, ... DBX320.0 - 2 (orientation axis 1: activate handwheel) 576
18.2.5.164 DB21, ... DBX320.4 (orientation axis 1: traversing key lock) 577
18.2.5.165 DB21, ... DBX320.5 (orientation axis 1: rapid traverse override) 578
18.2.5.166 DB21, ... DBX320.6 - 7 (Orientierungsachse 1: traversing keys "Plus" / "Minus") 578
18.2.5.167 DB21, ... DBX321.0 - 6 (orientation axis 1: machine function request)....580
18.2.5.168 DB21, … DBX323.0 (orientation axis 1: invert handwheel direction of rotation)....581
18.2.5.169 DB21, ... DBX324.0 - 2 (orientation axis 2: activate handwheel) 581
18.2.5.170 DB21, ... DBX324.4 (orientation axis 2: traversing key lock) 581
18.2.5.171 DB21, ... DBX324.5 (orientation axis 2: rapid traverse override) 582
18.2.5.172 DB21, ... DBX324.6 - 7 (orientation axis 2: traversing keys "Plus" / "Minus") 582
18.2.5.173 DB21, ... DBX325.0 - 6 (orientation axis 2: machine function request)....582
18.2.5.174 DB21, … DBX327.0 (orientation axis 2: invert handwheel direction of rotation)....582
18.2.5.175 DB21, ... DBX328.0 - 2 (orientation axis 3: activate handwheel) 582
18.2.5.176 DB21, ... DBX328.4 (orientation axis 3: traversing key lock) 582
18.2.5.177 DB21, ... DBX328.5 (orientation axis 3: rapid traverse override) 582
18.2.5.178 DB21, ... DBX328.6 - 7 (orientation axis 3: traversing keys "Plus" / "Minus") 583
18.2.5.179 DB21, ... DBX329.0 - 6 (orientation axis 3: machine function request)....583
18.2.5.180 DB21, … DBX331.0 (orientation axis 3: invert handwheel direction of rotation)....583
18.2.5.181 DB21, ... DBX332.0 - 2 (orientation axis 1: handwheel active) 583
18.2.5.182 DB21, ... DBX332.4 - 5 (orientation axis 1: traversing request "Plus" / "Minus") 584

18.2.5.183 DB21, ... DBX332.6 - 7 (orientation axis 1: traversing command "Plus" / "Minus") 585
18.2.5.184 DB21, ... DBX333.0 - 6 (orientation axis 1: active machine function) 586
18.2.5.185 DB21, ... DBX336.0 - 2 (orientation axis 2: handwheel active) 586
18.2.5.186 DB21, ... DBX336.4 - 5 (orientation axis 2: traversing request "Plus" / "Minus") 587
18.2.5.187 DB21, ... DBX336.6 - 7 (orientation axis 2: traversing command "Plus" / "Minus") 587
18.2.5.188 DB21, ... DBX337.0 - 6 (orientation axis 2: active machine function) 587
18.2.5.189 DB21, ... DBX340.0 - 2 (orientation axis 3: handwheel active) 587
18.2.5.190 DB21, ... DBX340.4 - 5 (orientation axis 3: traversing request "Plus" / "Minus") 587
18.2.5.191 DB21, ... DBX340.6 - 7 (orientation axis 3: traversing command "Plus" / "Minus") 587
18.2.5.192 DB21, ... DBX341.0 - 6 (orientation axis 3: active machine function) 587
18.2.5.193 DB21, ... DBX344.0 (TM: tool prewarning limit reached) 588
18.2.5.194 DB21, ... DBX344.1 (TM: Tool limit value reached) 588
18.2.5.195 DB21, ... DBX344.2 (TM: Transition to new replacement tool) 588
18.2.5.196 DB21, ... DBX344.3 (TM: Last replacement tool of the tool group) 589
18.2.5.197 DB21, ... DBB376 (PROG_EVENT trigger event) 589
18.2.5.198 DB21, ... DBX377.0 (collision avoidance: stop) 590
18.2.5.199 DB21, ... DBX377.4 (JOG retract active) 590
18.2.5.200 DB21, ... DBX377.5 (JOG retract retraction data available) 590
18.2.5.201 DB21, ... DBX377.6 (JOG circular travel active) 590
18.2.5.202 DB21, … DBX378.0 (ASUB active) 591
18.2.5.203 DB21, … DBX378.1 (stopped ASUB active) 591
18.2.5.204 DB21, ... DBX384.0 (enable GOTOS) 591
18.2.5.205 DB21, ... DBX385.0 - 7 (input signal 1 - 8) 592
18.2.5.206 DB21, ... DBX386.0 - 7 (disabling of input signal 1 - 8) 592
18.2.5.207 DB21, ... DBX387.0 - 7 (status of action 1 - 8) 592
18.2.5.208 DB21, ... DBX390.0 - 7 (enable status of input signal 1 - 8) 593
18.2.5.209 DB21, ... DBX391.0 - 7 (enable request for action 1 - 8) 593
18.2.5.210 DB21, ... DBB392 (selection: Coordinate system for Cartesian manual traversing and handwheel override in automatic mode in the tool direction (DRF)) 593
18.2.6 DB31, ...: Axis/spindle 594
18.2.6.1 DB31, ... DBB0 (feedrate override, axis-specific) 594
18.2.6.2 DB31, ... DBX1.0 (drive test travel enable) 596
18.2.6.3 DB31, ... DBX1.1 (acknowledge fixed stop reached). 597
18.2.6.4 DB31, ... DBX1.2 (fixed stop sensor) 597
18.2.6.5 DB31, ... DBX1.3 (axis/spindle disable) 598
18.2.6.6 DB31, ... DBX1.4 (follow-up mode) 601
18.2.6.7 DB31, ... DBX1.5 - 6 (position measuring system 1 (PMS1) / position measuring system 2 (PMS2)) 602
18.2.6.8 DB31, ... DBX1.7 (override active) 604
18.2.6.9 DB31, ... DBX2.0 (software cams: activation) 605
18.2.6.10 DB31, ... DBX2.1 (controller enable) 605
18.2.6.11 DB31, ... DBX2.2 (spindle reset/delete distance to go) 607
18.2.6.12 DB31, ... DBX2.3 (clamping in progress) 609
18.2.6.13 DB31, ... DBX2.4 - 7 (reference point value 1 - 4) 609
18.2.6.14 DB31, ... DBX3.0 (accept external work offset) 610
18.2.6.15 DB31, ... DBX3.1 (enable travel to fixed stop) 610
18.2.6.16 DB31, ... DBX3.2 - 5 (activate fixed feedrate 1 - 4, machine axes) 611
18.2.6.17 DB31, ... DBX3.6 (velocity/spindle speed limiting) 611
18.2.6.18 DB31, ... DBX4.0 - 2 (activate handwheel) 612
18.2.6.19 DB31, ... DBX4.3 (feedrate/spindle stop, axis-specific) 613
18.2.6.20 DB31, ... DBX4.4 (traversing key disable) 614
18.2.6.21 DB31, ... DBX4.5 (rapid traverse override) 614

18.2.6.22 DB31, ... DBX4.6 - 7 ("Plus" / "Minus" traversing keys) 614
18.2.6.23 DB31, ... DBX5.0 - 6 (machine function request) 615
18.2.6.24 DB31, ... DBX7.0 (invert handwheel direction of rotation) 616
18.2.6.25 DB31, ... DBB8 (request axis/spindle replacement) 617
18.2.6.26 DB31, ... DBX9.0 - (2 select: position controller parameter set) 618
18.2.6.27 DB31, ... DBX9.3 (parameter set definitions disabled by NC) 618
18.2.6.28 DB31, ... DBX10.0 (REPOS delay) 618
18.2.6.29 DB31, ... DBX12.0 (hardware limit switch minus) 619
18.2.6.30 DB31, ... DBX12.1 (hardware limit switch plus) 619
18.2.6.31 DB31,... DBX12.2 (second software limit switch minus) 620
18.2.6.32 DB31, ... DBX12.3 (second software limit switch plus) 620
18.2.6.33 DB31, ... DBX12.4 (modulo rotary axis: activate traversing range limits) 620
18.2.6.34 DB31, ... DBX12.7 (reference point approach deceleration) 620
18.2.6.35 DB31, ... DBX13.0 - 2 (JOG approach fixed point) 621
18.2.6.36 DB31, ... DBX13.3 (JOG travel to position) 621
18.2.6.37 DB31, ... DBX14.0 (suppress program test) 622
18.2.6.38 DB31, ... DBX14.1 (activate program test) 623
18.2.6.39 DB31, ... DBX16.0 - 2 (actual gear stage) 623
18.2.6.40 DB31, ... DBX16.3 (gear stage has been changed) 624
18.2.6.41 DB31, ... DBX16.4 (resynchronize spindle, measuring system 1) 625
18.2.6.42 DB31, ... DBX16.5 (resynchronize spindle, measuring system 2) 625
18.2.6.43 DB31, ... DBX16.7 (delete S value) 626
18.2.6.44 DB31, ... DBX17.4 (resynchronizing the spindle before positioning, measuring system 1) 626
18.2.6.45 DB31, ... DBX17.5 (resynchronizing the spindle before positioning, measuring system 2) 626
18.2.6.46 DB31, ... DBX17.6 (invert M3 / M4) 627
18.2.6.47 DB31, ... DBX18.4 (oscillation controlled by the PLC) 627
18.2.6.48 DB31, ... DBX18.5 (oscillation enable) 628
18.2.6.49 DB31, ... DBX18.6 (oscillation: Clockwise direction of rotation) 628
18.2.6.50 DB31, ... DBX18.7 (oscillation: Counterclockwise direction of rotation) 629
18.2.6.51 DB31, ... DBB19 (speed override, spindle-specific) 629
18.2.6.52 DB31, ... DBX20.1 (ramp-function generator disable) 631
18.2.6.53 DB31, ... DBX21.0 - 4 (motor/drive data set: selection) 632
18.2.6.54 DB31, ... DBX21.5 (motor has been selected) 632
18.2.6.55 DB31, ... DBX21.6 (integrator disable, speed controller) 633
18.2.6.56 DB31, ... DBX21.7 (pulse enable) 633
18.2.6.57 DB31, ... DBB22.0 (SI: SBH/SG deselection) 634
18.2.6.58 DB31, ... DBB22.1 (SI: SBH deselection) 634
18.2.6.59 DB31, ... DBB22.3 - 4 (SI: SG selection) 634
18.2.6.60 DB31, ... DBB23.0 - 2 (gear ratio selection) 635
18.2.6.61 DB31, ... DBB23.4 (SI: SE selection) 635
18.2.6.62 DB31, ... DBB24.1 (control axis) 635
18.2.6.63 DB31, …DBX24.2 (MCS coupling: deactivate or do not permit) 636
18.2.6.64 DB31, ... DBX24.3 (MCS coupling: activate collision protection) 636
18.2.6.65 DB31, ... DBX24.4 (master-slave: activate torque compensatory controller) 637
18.2.6.66 DB31, ... DBX24.5 (setpoint switchover: request drive control) 637
18.2.6.67 DB31, ... DBX24.7 (master-slave: activate coupling) 637
18.2.6.68 DB31, ... DBX26.4 (enable following axis override) 638
18.2.6.69 DB31, ... DBX28.0 (oscillation reversal from external) 638
18.2.6.70 DB31, ... DBX28.1 (PLC-controlled axis: Reset) 639
18.2.6.71 DB31, ... DBX28.2 (PLC-controlled axis: Resume) 639
18.2.6.72 DB31, ... DBX28.3 (set reversal point) 640
18.2.6.73 DB31, ... DBX28.4 (change reversal point) 641

18.2.6.74 DB31, ... DBX28.5 (PLC-controlled axis: Stop at the next reversal point) 641
18.2.6.75 DB31, ... DBX28.6 (PLC-controlled axis: Stop along braking ramp) 641
18.2.6.76 DB31, ... DBX28.7 (request, PLC-controlled axis) 642
18.2.6.77 DB31, ... DBX31.5 (disable synchronization) 643
18.2.6.78 DB31, ... DBX34.0 - 1 (setpoint limitation) 643
18.2.6.79 DB31, ... DBX60.0 (spindle / rotary axis) 644
18.2.6.80 DB31, ... DBX60.1 (NCU link axis active) 644
18.2.6.81 DB31, ... DBX60.2 (encoder limit frequency exceeded 1) 645
18.2.6.82 DB31, ... DBX60.3 (encoder frequency limit exceeded, measuring system 2) 645
18.2.6.83 DB31, ... DBX60.4 (referenced/synchronized 1) 646
18.2.6.84 DB31, ... DBX60.5 (referenced/synchronized 2) 646
18.2.6.85 DB31,... DBX60.6 (position reached with exact stop coarse) 647
18.2.6.86 DB31, ... DBX60.7 (position reached with exact stop fine) 647
18.2.6.87 DB31, ... DBX61.0 (drive test travel request) 648
18.2.6.88 DB31, ... DBX61.1 (axis-specific alarm) 648
18.2.6.89 DB31, ... DBX61.2 (axis ready) 649
18.2.6.90 DB31, ... DBX61.3 (follow-up active) 649
18.2.6.91 DB31, ... DBX61.4 (axis/spindle stationary (n < nmin)) 650
18.2.6.92 DB31, ... DBX61.5 (position controller active) 650
18.2.6.93 DB31, ... DBX61.6 (speed controller active) 651
18.2.6.94 DB31, ... DBX61.7 (current controller active) 651
18.2.6.95 DB31, ... DBX62.0 (software cams active) 652
18.2.6.96 DB31, ... DBX62.1 (handwheel override active) 652
18.2.6.97 DB31, ... DBX62.2 (revolutional feedrate active) 653
18.2.6.98 DB31, ... DBX62.3 (measurement active) 653
18.2.6.99 DB31, ... DBX62.4 (activate travel to fixed stop) 653
18.2.6.100 DB31, ... DBX62.5 (fixed stop reached) 654
18.2.6.101 DB31, ... DBX62.7 (axis container rotation active) 654
18.2.6.102 DB31, ... DBX63.0 (reset executed) 654
18.2.6.103 DB31, ... DBX63.1 (PLC-controlled axis) 655
18.2.6.104 DB31, ... DBX63.2 (axis stop active) 655
18.2.6.105 DB31, ... DBX63.3 (axis/spindle disable active) 655
18.2.6.106 DB31, ... DBX64.0 - 2 (handwheel active) 656
18.2.6.107 DB31, ... DBX64.4 - 5 ("Plus" / "Minus" traversing request) 657
18.2.6.108 DB31, ... DBX64.6 - 7 ("Plus" / "Minus" traversing command) 657
18.2.6.109 DB31, ... DBX65.0 - 6 (active machine function) 658
18.2.6.110 DB31, ... DBX66.0 (MCS coupling: collision protection active) 658
18.2.6.111 DB31, ... DBX67.0 (handwheel direction of rotation inversion active) 659
18.2.6.112 DB31, ... DBB68 (status axis/spindle replacement) 659
18.2.6.113 DB31, ... DBX69.0 - 2 (active position controller parameter set) 659
18.2.6.114 DB31, ... DBX70.0 (REPOS offset) 660
18.2.6.115 DB31, ... DBX70.1 (REPOS offset valid) 660
18.2.6.116 DB31, ... DBX70.2 (REPOS delay acknowledgment) 661
18.2.6.117 DB31, ... DBX71.4 (position restored measuring system 1) 662
18.2.6.118 DB31, ... DBX71.5 (position restored, measuring system 2) 662
18.2.6.119 DB31, ... DBX72.0 (REPOS delay) 662
18.2.6.120 DB31, ... DBX74.4 (modulo rotary axis: traversing range limits active) 663
18.2.6.121 DB31, ... DBX75.0 - 2 (JOG approach fixed point active) 663
18.2.6.122 DB31, ... DBX75.3 - 5 (JOG approach fixed point reached) 664
18.2.6.123 DB31, ... DBX75.6 (JOG travel to position active) 664
18.2.6.124 DB31, ... DBX75.7 (JOG position reached) 664
18.2.6.125 DB31, ... DBX76.0 (lubrication pulse) 665

18.2.6.126 DB31, ... DBX76.4 (path axis) 665
18.2.6.127 DB31, ... DBX76.5 (positioning axis) 666
18.2.6.128 DB31, ... DBX76.6 (indexing axis in position) 666
18.2.6.129 DB31, ... DBX77.0 (collision avoidance: Velocity reduction) 667
18.2.6.130 DB31, ... DBD78 (feedrate, positioning axis) 667
18.2.6.131 DB31, ... DBX82.0 - 2 (set gear stage) 667
18.2.6.132 DB31, ... DBX82.3 (change over gear stage) 668
18.2.6.133 DB31, ... DBX83.0 (speed limit exceeded) 668
18.2.6.134 DB31, ... DBX83.1 (speed setpoint limited) 669
18.2.6.135 DB31, ... DBX83.2 (speed setpoint increased) 670
18.2.6.136 DB31, ... DBX83.3 (geometry monitoring) 671
18.2.6.137 DB31, ... DBX83.5 (spindle in setpoint range) 671
18.2.6.138 DB31, ... DBX83.6 (speed monitoring) 671
18.2.6.139 DB31, ... DBX83.7 (clockwise actual direction of rotation) 672
18.2.6.140 DB31, ... DBX84.1 (grinding wheel peripheral speed active) 672
18.2.6.141 DB31, ... DBX84.3 (tapping without compensating check active) 672
18.2.6.142 DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation) 673
18.2.6.143 DB31, ... DBX84.5 (active spindle mode: positioning mode) 673
18.2.6.144 DB31, ... DBX84.6 (active spindle mode: oscillating mode) 674
18.2.6.145 DB31, ... DBX84.7 (active spindle mode: control mode) 674
18.2.6.146 DB31, ... DBX85.0 (tool with dynamic response limitation) 674
18.2.6.147 DB31, ... DBX85.5 (spindle in position) 675
18.2.6.148 DB31, ... DBW86 (M function for spindle) 675
18.2.6.149 DB31, ... DBD88 (S function for spindle) 676
18.2.6.150 DB31, ... DBX92.1 (ramp-function generator disable active) 676
18.2.6.151 DB31, ... DBX92.4 (drive-autonomous motion active) 677
18.2.6.152 DB31, ... DBX93.0 - 4 (motor/drive data set: display) 677
18.2.6.153 DB31, ... DBX93.5 (drive ready) 678
18.2.6.154 DB31, ... DBX93.6 (integrator disable, speed controller) 678
18.2.6.155 DB31, ... DBX93.7 (pulses enabled) 679
18.2.6.156 DB31, ... DBX94.0 (motor temperature prewarning) 679
18.2.6.157 DB31, ... DBX94.1 (heat sink temperature prewarning) 680
18.2.6.158 DB31, ... DBX94.2 (run-up completed) 680
18.2.6.159 DB31, ... DBX94.3 (|Md| < Mdx) 681
18.2.6.160 DB31, ... DBX94.4 (|nact| < nmin) 682
18.2.6.161 DB31, ... DBX94.5 (|nact| < nx) 682
18.2.6.162 DB31, ... DBX94.6 (nact = nset) 683
18.2.6.163 DB31, ... DBX94.7 (variable signaling function) 683
18.2.6.164 DB31, ... DBX95.1 (ESR: DC link undervoltage) 683
18.2.6.165 DB31, ... DBX95.2 (ESR: Response initiated or generator operation active) 684
18.2.6.166 DB31, … DBX95.3 (generator operation minimum speed fallen below) 684
18.2.6.167 DB31, ... DBX95.7 (alarm of alarm class C is active) 684
18.2.6.168 DB31, ... DBX96.2 (master-slave: fine differential speed) 685
18.2.6.169 DB31, ... DBX96.3 (master-slave: coarse speed difference) 685
18.2.6.170 DB31, ... DBX96.4 (master-slave: compensatory controller active) 685
18.2.6.171 DB31, ... DBX96.5 (setpoint switchover: drive control active) 686
18.2.6.172 DB31, ... DBX96.7 (master-slave: coupling active) 686
18.2.6.173 DB31, ... DBX97.0 (MCS coupling: slave axis) 687
18.2.6.174 DB31, … DBX97.1 (MCS coupling: coupling active) 687
18.2.6.175 DB31, ... DBX97.2 (MCS coupling: mirroring active) 687
18.2.6.176 DB31, ... DBX97.3 (MCS coupling: offset change) 688
18.2.6.177 DB31, ... DBX98.0 (fine synchronous operation) 688

18.2.6.178 DB31, ... DBX98.1 (coarse synchronous operation) 689
18.2.6.179 DB31, ... DBX98.2 (actual value coupling) 689
18.2.6.180 DB31, ... DBX98.4 (overlaid movement) 690
18.2.6.181 DB31, ... DBX98.5 (speed alarm threshold reached) 690
18.2.6.182 DB31, ... DBX98.6 (acceleration alarm threshold reached) 691
18.2.6.183 DB31, ... DBX99.0 (leading spindle active) 691
18.2.6.184 DB31, ... DBX99.1 (following spindle active) 692
18.2.6.185 DB31, ... DBX99.3 (axis accelerates) 692
18.2.6.186 DB31, ... DBX100.2 (oscillation reversal from external active) 692
18.2.6.187 DB31, ... DBX100.3 (oscillation cannot be started) 693
18.2.6.188 DB31, ... DBX100.4 (error during oscillation) 693
18.2.6.189 DB31, …DBX100.5 (sparking-out active) 693
18.2.6.190 DB31, ... DBX100.6 (oscillation active) 693
18.2.6.191 DB31, ... DBX100.7 (oscillation active) 694
18.2.6.192 DB31, ... DBX102.5 (position measuring system 1 activated) 694
18.2.6.193 DB31, ... DBX102.6 (position measuring system 2 activated) 694
18.2.6.194 DB31, ... DBX104.0 - 107.6 (active infeed axes) 695
18.2.6.195 DB31, ... DBX128.0 (suppress program test) 695
18.2.6.196 DB31, ... DBX128.1 (activate program test) 695
18.2.6.197 DB31, ... DBX130.0 - 4 (motor/drive data set: formatting) 696
18.2.6.198 DB31, ... DBX132.0 (sensors available) 696
18.2.6.199 DB31, ... DBX132.1 (sensor S1 available (clamped state)) 697
18.2.6.200 DB31, … DBX132.4 (sensor S4 available (piston end position)) 697
18.2.6.201 DB31, ... DBX132.5 (sensor S5 available (angular position of the motor shaft)) 697
18.2.6.202 DB31, ... DBX133.2 (state value is generated, speed limitation p5043 is active) 698
18.2.6.203 DB31, … DBW134 (state of the clamping system (sensor S1)) 698
18.2.6.204 DB31, ... DBW136 (analog measured value: of the clamping system) 699
18.2.6.205 DB31, … DBX138.4 (sensor S4, piston end position) 699
18.2.6.206 DB31, ... DBX138.5 (sensor S5 angular position of the motor shaft) 699
18.2.7 DB71. Tool management, loading/unloading points 700
18.2.7.1 DB71 DBX0.0 - 1.7 (active status of interfaces 1 - 16) 700
18.2.7.2 DB71 DBX2.0 - 3.7 ("auto" acknowledgment of interfaces 1 - 16) 700
18.2.7.3 DB71 DBX(n+0).0 (command: Loading) 700
18.2.7.4 DB71 DBX(n+0).1 (command: Unload) 701
18.2.7.5 DB71 DBX(n+0).2 (command: Relocate) 701
18.2.7.6 DB71 DBX(n+0).3 (command: Position to load point) 702
18.2.7.7 DB71 DBX(n+0).4 (command: Job/task comes from the NC program) 702
18.2.7.8 DB71 DBX(n+0).5 (command: Position a multitool) 703
18.2.7.9 DB71 DBX(n+1).0 ("auto" acknowledgment negative) 703
18.2.7.10 DB71 DBX(n+1).7 (command: data in the extended range) 704
18.2.7.11 DB71 DBB(n+2) (assigned channel) 704
18.2.7.12 DB71 DBB(n+3) (tool management No.) 704
18.2.7.13 DB71 DBW(n+16) (identifier for loading/unloading point (fixed value 9999)) 705
18.2.7.14 DB71 DBW(n+18) (location number of the loading/unloading point) 705
18.2.7.15 DB71 DBW(n+20) (Magazine No. (Source) for unloading/relocating/positioning) 705
18.2.7.16 DB71 DBW(n+22) (Location No. (Source) for unloading/relocating/positioning) 706
18.2.7.17 DB71 DBW(n+24) (Magazine No. (Target) for unloading/relocating/positioning) 706
18.2.7.18 DB71 DBW(n+26) (location No. (Target) for unloading/relocating/positioning) 706
18.2.7.19 DB71 DBX(n+28).0 (loading/unloading without any magazine movement) 707
18.2.8 DB72: Tool management, change in the spindle 707
18.2.8.1 DB72 DBX0.0-1.7 (active status of interfaces 1-16) 707
18.2.8.2 DB72 DBX2.0-3.7 ("auto" acknowledgment of interfaces 1-16) 708

18.2.8.3 DB72 DBX(n+0).0 (command code: Obligatory change) 708
18.2.8.4 DB72 DBX(n+0).1 (command code: Perform change with M06) 708
18.2.8.5 DB72 DBX(n+0).2 (command code: Prepare change)....709
18.2.8.6 DB72 DBX(n+0).3 (command code: T0) 709
18.2.8.7 DB72 DBX(n+0).4 (command code: Old tool in buffer)....710
18.2.8.8 DB72 DBX(n+0).5 (command code: Load manual tool)....710
18.2.8.9 DB72 DBX(n+0).6 (command code: Unload manual tool) 711
18.2.8.10 DB72 DBX(n+0).7 (command code: Tool remains in spindle) 711
18.2.8.11 DB72 DBX(n+1).0 ("auto" acknowledgment negative)....712
18.2.8.12 DB72 DBX(n+1).7 (command: data in the extended range) 712
18.2.8.13 DB72 DBB(n+2) (assigned channel)....712
18.2.8.14 DB72 DBB(n+3) (tool management No.)....713
18.2.8.15 DB72 DBD(n+4) (user-definable parameter 0 (DInt))....713
18.2.8.16 DB72 DBD(n+8) (user-definable parameter 1 (DInt))....713
18.2.8.17 DB72 DBD(n+12) (user-definable parameter 2 (DInt))....714
18.2.8.18 DB72 DBW(n+16) (buffer magazine number (fixed value 9998)) 714
18.2.8.19 DB72 DBW(n+18) (location in the buffer magazine (spindle)) 714
18.2.8.20 DB72 DBW(n+20) (magazine No. (source) for new tool to be loaded) 715
18.2.8.21 DB72 DBW(n+22) (location No. (source) for new tool) 715
18.2.8.22 DB72 DBW(n+24) (magazine No. (target) for old tool to be removed) 715
18.2.8.23 DB72 DBW(n+26) (location No. (target) for old tool)....716
18.2.8.24 DB72 DBW(n+28) (new tool: Location type) 716
18.2.8.25 DB72 DBW(n+30) (new tool: Size left)....716
18.2.8.26 DB72 DBW(n+32) (new tool: Size right)....717
18.2.8.27 DB72 DBW(n+34) (new tool: Size top)....717
18.2.8.28 DB72 DBW(n+36) (new tool: Size bottom)....717
18.2.8.29 DB72 DBW(n+38) (tool status for new tool)....718
18.2.8.30 DB72 DBW(n+40) (new tool: internal T number of NC) 718
18.2.8.31 DB72 DBW(n+42) (buffer location of the old tool)....719
18.2.8.32 DB72 DBW(n+44) (original magazine of the new tool) 719
18.2.8.33 DB72 DBW(n+46) (original location of the new tool)....719
18.2.9 DB73: Tool management, change in turret 720
18.2.9.1 DB73 DBX0.0 - 1.7 (active status of interfaces 1 - 16) 720
18.2.9.2 DB73 DBX2.0 - 3.7 ("auto" acknowledgment of interfaces 1 - 16)....720
18.2.9.3 DB73 DBX(n+0).0 (command code: Obligatory change) 721
18.2.9.4 DB73 DBX(n+0).1 (command code: Execute change)....721
18.2.9.5 DB73 DBX(n+0).3 (T0))....721
18.2.9.6 DB73 DBX(n+1).0 ("auto" acknowledgment negative)....722
18.2.9.7 DB73 DBX(n+1).7 (command: data in the extended range) 722
18.2.9.8 DB73 DBB(n+2) (assigned channel)....722
18.2.9.9 DB73 DBB(n+3) (tool management No.)....723
18.2.9.10 DB73 DBD(n+4) (user-definable parameter 0 (DInt))....723
18.2.9.11 DB73 DBD(n+8) (user-definable parameter 1 (DInt))....723
18.2.9.12 DB73 DBD(n+12) (user-definable parameter 2 (DInt))....724
18.2.9.13 DB73 DBW(n+20) (magazine number of the new tool)....724
18.2.9.14 DB73 DBW(n+22) (location number of the new tool to be loaded) 724
18.2.9.15 DB73 DBW(n+24) (magazine No. (target) for old tool to be removed) 725
18.2.9.16 DB73 DBW(n+26) (location number of the old tool to be removed)....725
18.2.9.17 DB73 DBW(n+28) (new tool: Location type) 725
18.2.9.18 DB73 DBW(n+30) (new tool: Size left)....726
18.2.9.19 DB73 DBW(n+32) (new tool: Size right)....726
18.2.9.20 DB73 DBW(n+34) (new tool: Size top)....726

18.2.9.21 DB73 DBW(n+36) (new tool: Size bottom) ... 727
18.2.9.22 DB73 DBW(n+38) (tool status for new tool) ... 727
18.2.9.23 DB73 DBW(n+40) (new tool: Internal T no. of the NC) ... 728
18.2.9.24 DB73 DBW(n+42) (original location of the new tool in this turret magazine) ... 728
18.2.10 DB1071: Tool management, load/unload magazine (multitool) ... 728
18.2.10.1 DB1071 DBW(n+0) (distance coding) ... 728
18.2.10.2 DB1071 DBW(n+2) (multitool location number) ... 729
18.2.10.3 DB1071 DBD(n+4) (multitool location distance) ... 729
18.2.10.4 DB1071 DBW(n+8) (multitool number) ... 730
18.2.10.5 DB1071 DBW(n+10) (multitool location number) ... 730
18.2.10.6 DB1071 DBW(n+12) (toolholder) ... 730
18.2.11 DB1072: Tool management, spindle (multitool) ... 731
18.2.11.1 DB1072 DBW(n+0) (distance coding) ... 731
18.2.11.2 DB1072 DBW(n+2) (multitool location number) ... 731
18.2.11.3 DB1072 DBW(n+4) (multitool location distance) ... 731
18.2.11.4 DB1072 DBW(n+8) (multitool number (new tool)) ... 732
18.2.11.5 DB1072 DBW(n+10) (multitool location number (new tool)) ... 732
18.2.11.6 DB1072 DBW(n+12) (multitool number (old tool)) ... 732
18.2.11.7 DB1072 DBW(n+14) (multitool location number (old tool)) ... 733
18.2.11.8 DB1072 DBW(n+16) (new tool: Location type) ... 733
18.2.11.9 DB1072 DBW(n+18) (new tool: Size left) ... 733
18.2.11.10 DB1072 DBW(n+20) (new tool: Size right) ... 734
18.2.11.11 DB1072 DBW(n+22) (new tool: Size top) ... 734
18.2.11.12 DB1072 DBW(n+24) (new tool: Size bottom) ... 734
18.2.11.13 DB1072 DBW(n+26) (new tool status) ... 735
18.2.11.14 DB1072 DBW(n+28) (new tool: internal T number of the NC) ... 735
18.2.11.15 DB1072 DBW(n+30) (tool holder) ... 736
18.2.11.16 DB1072 DBW(n+32) (original magazine of the new tool) ... 736
18.2.11.17 DB1072 DBW(n+34) (original location of the new tool) ... 736
18.2.12 DB1073: Tool management, turret (multitool) ... 737
18.2.12.1 DB1073 DBW(n+0) (distance coding) ... 737
18.2.12.2 DB1073 DBW(n+2) (multitool location number) ... 737
18.2.12.3 DB1073 DBW(n+4) (multitool location distance) ... 738
18.2.12.4 DB1073 DBW(n+8) (multitool number (new tool)) ... 738
18.2.12.5 DB1073 DBW(n+10) (multitool location number (new tool)) ... 738
18.2.12.6 DB1073 DBW(n+12) (multitool number (old tool)) ... 739
18.2.12.7 DB1073 DBW(n+14) (multitool location number (old tool)) ... 739
18.2.12.8 DB1073 DBW(n+16) (new tool: Location type) ... 739
18.2.12.9 DB1073 DBW(n+18) (new tool: Size left) ... 740
18.2.12.10 DB1073 DBW(n+20) (new tool: Size right) ... 740
18.2.12.11 DB1073 DBW(n+22) (new tool: Size top) ... 740
18.2.12.12 DB1073 DBW(n+24) (new tool: Size bottom) ... 741
18.2.12.13 DB1073 DBW(n+26) (new tool status) ... 741
18.2.12.14 DB1073 DBW(n+28) (new tool: internal T number of the NC) ... 742
18.2.12.15 DB1073 DBW(n+30) (tool holder) ... 742
18.2.12.16 DB1073 DBW(n+32) (original magazine of the new tool) ... 742
18.2.12.17 DB1073 DBW(n+34) (original location of the new tool) ... 743

A Appendix ... 745

A.1 Overview ... 745

Index ... 747

Fundamental safety instructions

1

1.1 General safety instructions

⚠ WARNING

Danger to life if the safety instructions and residual risks are not observed

If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Malfunctions of the machine as a result of incorrect or changed parameter settings

As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.

- Protect the parameterization against unauthorized access.
- Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

1.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

1.3 Industrial security

Note

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Products and solutions from Siemens constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. using firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that can be implemented, please visit:

Industrial security (https://www.siemens.com/industrialsecurity)

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they become available, and that only the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at:

Industrial security (https://www.siemens.com/industrialsecurity)

Further information is provided on the Internet:

Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ WARNING

Unsafe operating states resulting from software manipulation

Software manipulations, e.g. viruses, Trojans, or worms, can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- On completion of commissioning, check all security-related settings.
- Protect the drive against unauthorized changes by activating the "Know-how protection" converter function.

Overview 2

2.1 Brief description

General

The PLC basic program organizes the exchange of signals and data between the PLC user program and the NC, HMI and MCP areas. In the case of signals and data, a distinction is made between the following groups:

- Cyclic signal exchange
- Event-driven signal exchange
- Messages

Cyclic signal exchange

The cyclically-exchanged signals consist primarily of bit arrays.

- They contain **commands** transferred from the PLC to the NC (such as start or stop) and **status information** from the NC (such as program running, interrupted, etc.).
- The bit fields are organized into signals for:
  - Mode group
  - Channels
  - Axes/spindles
  - General NC signals

The cyclic exchange of data is performed by the basic program at the start of the PLC cycle (OB1). This ensures, for example, that the signals from the NC remain constant throughout a cycle.

Event-driven signal exchange NC → PLC

PLC functions that have to be executed as a function of the workpiece program are triggered by auxiliary functions in the workpiece program. If a block with auxiliary functions is executed, the type of auxiliary function determines whether the NC has to wait for this function to execute (e.g. tool change) or whether the function will be executed together with the workpiece machining process (e.g. tool loading on milling machines with chain magazine).

Data transfer must be as fast and yet as reliable as possible, in order to minimize the effect on the NC processing. Data transfer is, therefore, interrupt- and acknowledgment-driven. The basic program evaluates the signals and data, acknowledges this to the NC and transfers the data to the application interface at the start of the cycle. If the data does not require user acknowledgment, this does not affect NC processing.

Event-driven signal exchange PLC → NC

An "event driven signal exchange PLC → NC" takes place whenever the PLC transfers a request to the NC (e.g. traversing of an auxiliary axis). In this case, data transfer is also acknowledgment-driven. When performed from the user program, this type of signal exchange is triggered using a function block (FB) or function call (FC).

The associated FBs (Function Blocks) and FCs (Function Calls) are supplied together with the basic program.

Messages

User messages are acquired and conditioned by the basic program. The message signals are transferred to the basic program via a specified bit array. where they are evaluated and, if message events occur, entered in the PLC's interrupt buffer by means of the ALARM S/SQ functions. If an HMI (e.g. SINUMERIK Operate) is being used, the messages are transferred to the HMI and displayed.

PLC/HMI data exchange

In this type of data exchange, the HMI takes the initiative, being referred to as the "client" on the bus system. The HMI polls or writes data. The PLC processes these requests at the cycle control point via the operating system. The PLC basic program is not involved in these exchanges.

Note

The function of the machine is largely determined by the PLC program. Every PLC program in the RAM can be edited with the programming device.

Know-how protection for user blocks

To protect the know-how contained in the the user blocks (OB, FB and FC), they can be encoded with the SBP tool (SIMATIC block protection) contained in SIMATIC STEP 7. These blocks can then no longer be opened, debugged and modified without specifying the password for the encoding.

When encoding, the automation system on whose PLC-CPU the blocks are to be executed, must be specified: SIMATIC and/or SINUMERIK PLC-CPU.

The handling of the blocks, e.g. loading to the CPU, is not affected by the encoding.

Requirement

SIMATIC STEP 7 as of Version 5.5 SP3

2.2 Key data of the PLC CPU

Key data of the PLC CPU

For further information:
The overview of the key data of the PLC CPU integrated in the SINUMERIK NCU can be found in:
NCU 7x0.3 PN Manual, Section "Technical data"

Note

I/O addresses for integrated drives

The I/O addresses above 4096 are reserved for the integrated drives of the NCU and must not assigned otherwise.

Functions of the basic PLC program

Scope		
	Axes/spindles	31
	Channels	10
	Mode groups	10
Functions		
	Status/control signals	+
	M decoders (M00-99)	+
	G group decoders	+
	Aux. function distributors	+
	Aux. function transfer, interrupt-driven	+
	M decoding acc. to list	+
	Move axes/spindles from PLC	+
	ASUP interface	+
	Error/operating messages	+
	Transfer MCP and HHU signals	+
	Display control handheld unit	+
	Read/write NC variables and GUD	+
	PI services	+
	Tool management	+
	Star/delta switchover	+
M to N		+
Safety Integrated		+
Program diagnostics		+

2.3 PLC operating system version

The PLC operating system version is displayed at:

- User interface of SINUMERIK Operate: "Operating area switchover" > "Diagnostics" > "Version" ⇒ version data / system software NCU: Selection "PLC" > "Details" ⇒ version data / system software NCU/PLC: The PLC operating system version is displayed in the first line is at "PLC 3xx…".
  Note
  The displayed version is SINUMERIK-specific. It is not compatible with the basic SIMATIC CPU.
- SIMATIC STEP 7, HW Config: In the properties of the PLC CPU in the SINUMERIK rack: "Properties - CPU 3xx…" > "Order no. / firmware": xxxx / **Vx.y.z**
  Note
  The version of the basic SIMATIC CPU is displayed.

PLC mode selector

3

The PLC mode selector is located on the front of the NCU module. The following PLC operating modes can be set via the PLC mode selector:

S [1)]	Meaning	Remark
0	RUN-P	The PLC program can be changed without activation of the password
1	RUN	Only read access operations are possible using a programming device (PG). It is not possible to make changes to the PLC program until the password has been set.
2	STOP	Processing the PLC program is stopped and all PLC outputs are set to substitute values.
3	MRES	The PLC is switched into the STOP state followed by a PLC general reset (default data).
1) Switch position of the PLC mode selector		

Further information

A detailed description of the position of the PLC mode selector on the front of the NCU module, as well as its use in connection with NC and PLC general reset can be found in:

CNC Commissioning Manual: NC, PLC, Drive:

- Section "Switch-on/power-up" > "Operator control and display elements for power-up"
- Section "Switch-on/power-up" > "NC and PLC general reset"
- Section "General tips" > "Separate NC and PLC general reset"

Reserve resources (timers, counters, FC, FB, DB, I/O) 4

Reserve resources (timers, counters, FC, FB, DB, I/O)

The components below are reserved for the basic program:

- **Timer**
  No reservation
- **Counter**
  No reservation
- **FC, FB, DB**
  FC0 to FC29 and FB0 to FB29 are reserved for the basic program. The number range between 1000 and 1023 is also reserved for FCs and FBs. DB1 to DB61, DB71 to DB80 are reserved for data blocks. The number range 1000 to 1099 is also reserved in addition for DB. The data blocks of channels, axes/spindles and tool management functions that are not activated may be assigned as desired by the user.
- **I/O range**
  The PLC has an I/O address volume of 16384 bytes each for inputs and outputs. The address ranges starting at 4096 are reserved for/occupied by integrated drives. However, diagnostic addresses for modules can be assigned to the highest address range as proposed by STEP 7. The address range between 4080 and 4097 is also assigned for the NC, CP and HMI in rack 0 of the SIMATIC 300 station (for NCU 7x0.3).

Commissioning hardware configuration of the PLC CPU

5

The commissioning of the PLC CPU is described in detail in:

Further information

CNC Commissioning Manual: NC, PLC, Drive:

- Section: "Connect PG/PC to PLC"
- Section: "Commissioning PLC"
- Section: "Basics" > "PLC program"
- Section: "General tips" > "Separate NC and PLC general reset"
- Section: "General tips" > "Integrating PG/PC into the network (NetPro)"

Starting up the PLC program 6

6.1 Installation of the basic program

The installation of the basic program is described in detail in:

Further information

CNC Commissioning Manual: NC, PLC, Drive; Section: "Commissioning PLC" > "Creating a PLC program"

Note

Installation/update

Before installing the toolbox for SINUMERIK 840D sl, SIMATIC STEP 7 must be installed.

It is recommended that the hardware expansions for STEP 7 be installed again from the toolbox after an update of STEP 7.

Contents

The OB source programs, including standard parameterization, interface symbols and data-block templates for the handheld unit and M decoding functions are included in the basic program.

6.2 Application of the basic program

A new CPU program (e.g. "Turnma1") must be set up in a project by means of the STEP 7 software for each installation (machine).

Remark

The catalog structures of a project and the procedure for creating projects and user programs are described in the relevant SIMATIC documentation.

Procedure

The basic program blocks are copied using the SIMATIC Manager and "File" > "Open" > "Library".

The following components must be copied from the library:

- From the block container: FCs, FBs, DBs, OBs, SFC, SFB, UDT
- The source_files (from the source container): GPOB840D
- Possibly MDECLIST, HHU_DB and others
- The symbols table (from the symbols container)

Compatibility with STEP 7

There are no dependencies between the basic program and current STEP 7 versions.

6.3 Version codes

Basic program

The version of the basic program is displayed on the Version screen of the user interface along with the control system type.

The control system type is encoded as follows:

Leftjustified decade of DB17.DBD0 (byte 0)	Control system type
03	SINUMERIK 840D sl (NCU 7x0)

User program version identifications

Users can also display their own PLC version codes on the HMI version screen. For this purpose, a data of type STRING containing a maximum of **54** characters must be defined in any data block. The version, however, is not interpreted, but rather the entered string is accepted. The parameterization on this string is done via a pointer on FB1. For this, the data block must be defined symbolically. See the FB1 block description for more information.

The version identification can be formatted in the string as follows:

- xx.yy
- xx.yy.zz
- ww.xx.yy.zz
- vv.ww.xx.yy.zzz
- x.y
- x.y.z
- w.x.y.z
- v.w.x.y.z

In addition to the version identification, a date can be entered in the string, which given the appropriate formatting, is displayed in the HMI in the version screen. The date, however, is not interpreted, but rather the entered string is accepted. A combination with version identifications is possible. The following formats are currently supported:

- 00/00/0000
- 0000/00/00
- 00/00/00

Examples:

- "Test project version 01.02.03 01/01/2015"
- "1.2 2015/01/01 test project"
- "01/01/15 version 01.02 test project"

6.4 Machine program

The machine manufacturer creates the machine program using the library routines supplied with the basic program. The machine program contains the logic operations and sequences on the machine. The interface signals to the NC are also controlled in this program. More complex communication functions with the NC, e.g. read/write NC data, tool-management acknowledgments, etc., are activated and executed via the FCs and FBs blocks of the basic program.

The machine program can be created in various STEP 7 creation languages, e.g. STL, LAD, FBD, S7-HIGRAPH, S7GRAPH, SCL. The complete machine program must be generated and compiled in the correct sequence.

This means that blocks that are called by other blocks must generally be compiled **before these** blocks.

If blocks that are called by other blocks are subsequently modified in the interface (VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR) as the program is developed, then the call block and all blocks associated with it must be compiled again. This general procedure applies analogously to instance data blocks for FBs. If this sequence of operations is not observed, timestamp conflicts occur when the data is retranslated into STEP 7. As such, the recompilability of the blocks is not ensured and with the function "Status of block" unnecessary conflicts can also appear. It is, moreover, advisable to generate blocks in ASCII-STL by means of the STEP 7 editor when they have been created in Ladder Diagram or in single statements (incremental mode).

6.5 Data backup

The PLC-CPU does not save any symbolic names, but instead only the datatype descriptions of the block parameters VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR and the datatypes of the global data blocks.

Note

No sensible recompilation is possible without the related project for this machine. This especially affects, for instance the function status of the block or the necessary changes done in the PLC-CPU programs later. It is, therefore, necessary to keep a backup copy of the STEP 7 project located in the PLC CPU on the machine. This is a great help for the service case and saves unnecessary consumption of time in restoring the original project.

If the STEP 7 project exists and has been created according to the instructions given above, then symbols can be processed in the PLCCPU on this machine. It may also be advisable to store the machine source programs as ".awl" files in case they are required for any future upgrade.

The source programs of all organization blocks and all instance data blocks should always be available.

6.6 PLC series startup, PLC archive

Once the blocks have been loaded to the PLC CPU, a series archive can be generated via the HMI user interface to back up data on the machine. To ensure data consistency, this backup must be created immediately after block loading when the PLC is in the Stop state. It does not replace the SIMATIC project backup as the series archive saves binary data only. For instance, no symbolic information is present here. In addition, no CPU DBs (SFC 22 DBs) or SDBs generated in the CPU are saved.

Selection of the SINUMERIK archiving program

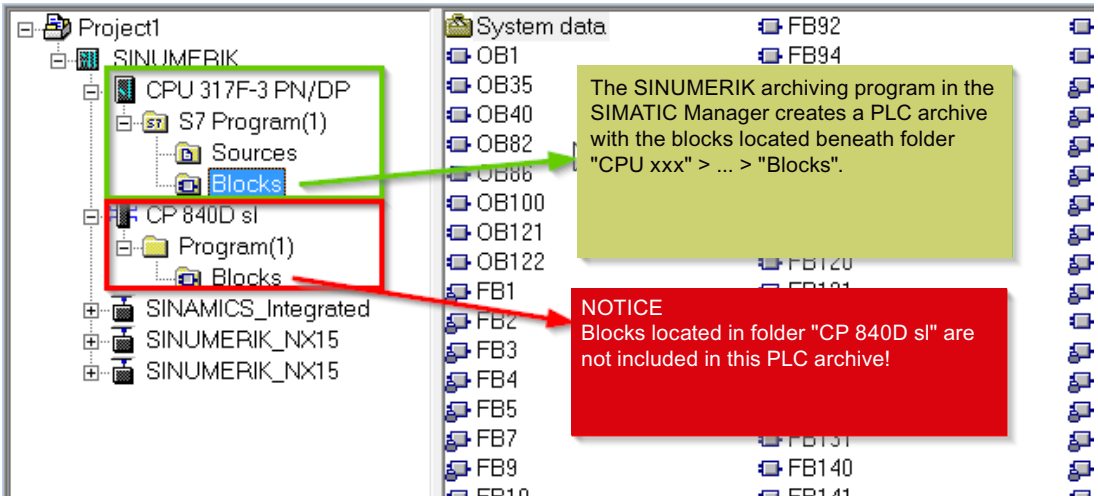
The PLC series archive can be generated directly from the SIMATIC project as an alternative:

1. In the SIMATIC Manager open dialog box "Settings": menu bar "Tools" > "Settings".
2. Open tab "Archiving".
3. In the "Preferred archiving program" drop-down list box, select the SINUMERIK archiving program "SINUMERIK (*.arc)".

Start of the SINUMERIK archiving program

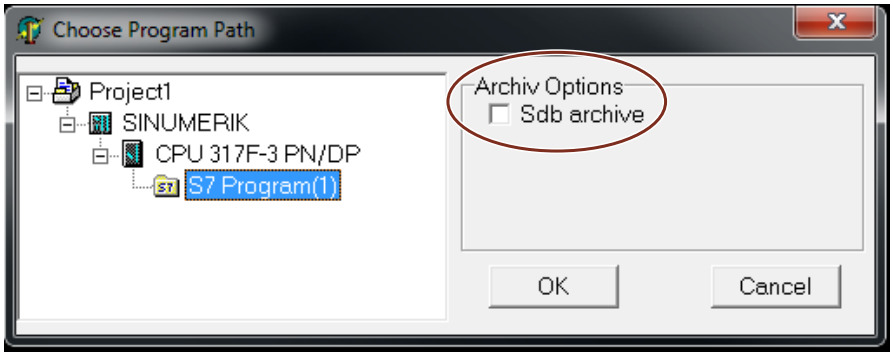
The SINUMERIK archiving program is started in the SIMATIC Manager via the menu command "File" > "Archive".

After you have selected a project and assigned an archive name, the PLC archive is generated. If a project contains several program paths, the S7 program for which the PLC archive will be created can be selected in the dialog box. All blocks contained in folder "CPU ..." > ... > "Blocks" at the selected program path are archived. Blocks in the folder "CP 840D sl" are not included in the PLC archive:



Data blocks that were created with SFC22 (online) in the CPU are also excluded from the archive.

The "Sdb archive" option can be activated or deactivated for the archiving program:



If "Sdb archive" is activated, a PLC archive is created that only contains the system data blocks (SDB) of the selected program path.

Creating application-specific PLC archives

Various archive options are available in Expert mode when creating a PLC archive.

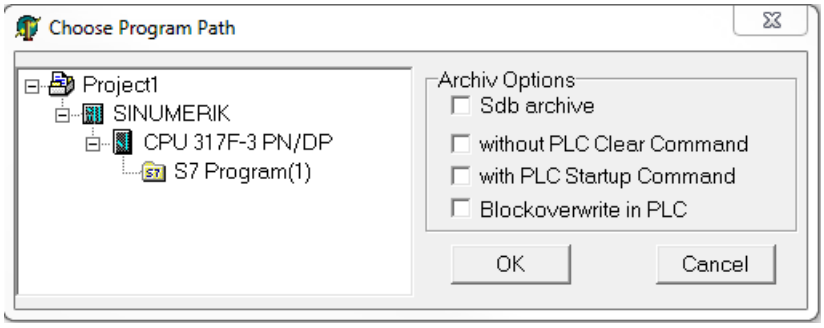
Expert mode is activated with two parameters in the "MMCARC.INI" file:

- Parameter Option = 1
- Parameter Expert = 1

The "MMCARC.INI" file is available as standard via the path "C:\Program Files (x86)\Siemens \Step7\S7bin\". Administrator rights are required in order to change the file.

The behavior and result when importing into the PLC varies depending on the respectively set archive options.

The following figure shows the options for creation of a PLC archive in Expert mode:



PLC reload archive without overwriting

The following table displays the project with the data from which the archive is created and the archive options to be activated:

Data in the PLC project	Archive options
• Program blocks	☑ "Without general PLC reset" ☑ "With PLC start command" (optional)

The program blocks must not exist in the PLC to be loaded.

The project must not include any system data.

Result when importing the archive:

- There is no general PLC reset.
- The program blocks are transferred.
- No blocks are overwritten.
- With the "With PLC start command" option selected, the PLC is started after loading.

PLC reload archive with overwriting

The following table displays the project with the data from which the archive is created and the archive options to be activated:

Data in the PLC project	Archive options
• Program blocks	☑ "Overwrite block in PLC"

The project must not include any system data.

Result when importing the archive:

- There is no general PLC reset.
- The program blocks are transferred and available blocks overwritten in the PLC as necessary.
- Blocks which are only available in the PLC and not available in the archive remain unchanged.
- The PLC is always restarted after loading.

PLC hardware upgrade archive

The following table displays the project with the data from which the archive is created and the archive options to be activated:

Data in the PLC project	Archive options
• System data • Program blocks may be included, however, are not transferred	☑ "Sdb archive"

Result when importing the archive:

- There is no general PLC reset.
- The system data is overwritten in the PLC.
- The PLC is always restarted after loading.

PLC commissioning archive

The following table displays the project with the data from which the archive is created and the archive options to be activated:

Data in the PLC project	Archive options
• System data • Program blocks	☑ "With PLC start command" (optional)

Result when importing the archive:

- There is a general PLC reset.
- System data and program blocks are transferred into the PLC.
- With the "With PLC start command" option selected, the PLC is started after loading.

Automation

The process of generating a series archive can be automated (comparable to the command interface in STEP 7). In generating this series archive, the command interface is expanded.

The following functions are available for this expansion:

The functions (shown here in VB script) are not available until server instantiations and Magic have been called:

```
Const S7BlockContainer = 1138689, S7PlanContainer = 17829889
Const S7SourceContainer = 1122308
set S7 = CreateObject("Simatic.Simatic.1")
rem Instantiate command interface of STEP 7
Set S7Ext = CreateObject("SimaticExt.S7ContainerExt")
Call S7Ext.Magic("")
```

Functions:

- `Function Magic(bstrVal As String) As Long`
- `Function MakeSerienIB (FileName As String, Option As Long, Container As S7Container) As Long`

Description

Function **Magic**(bstrVal As String) As Long

The call provides access to certain functions. The function must be called once after server instantiation. The value of bstrVal can be empty. This initiates a check of the correct STEP 7 version and path name in Autoexec. The functions are enabled with a return parameter of 0.

Return parameter (-1) = incorrect STEP 7 version

Return parameter (-2) = no entry in Autoexec.bat

Function **MakeSerienIB**(FileName As String, Option As Long, Container As S7Container) As Long

Parameter "Option":

0:	Normal series startup file with general reset.
Bit 0 = 1:	Series startup file without general reset. When project contains SDBs, this option is inoperative. A general reset is then always executed.
Bit 1 = 1:	Series startup file with PLC restart

Return value:

0	= OK
-1	= Function unavailable, call Magic function beforehand
-2	= File name cannot be generated
-4	= Container parameter invalid or container block empty
-5	= Internal error (memory request rejected by Windows)
-6	= Internal error (problem in STEP 7 project)
-7	= Write error when generating series startup files (e.g. data storage medium full)

Use in script

Program code

```
If S7Ext.Magic("") < 0 Then
    Wscript.Quit(1)
End If
    Set Proj1 = s7.Projects("new")
    set S7Prog = Nothing
    Set s7prog = Proj1.Programs.Item(1) if there is only one program
For Each cont In s7prog.Next
    If (Cont.ConcreteType = S7BlockContainer) Then
    'Check block container
    Exit For
    End if
    Cont = Nothing
    Next
Error = S7Ext.MakeSerienIB("f:\dh\arc.dir\PLC.arc", 0, Cont)
'Now error analysis
```

The `For Each ... Next` block programmed above can be programmed in the Delphi programming language as follows (The programming in the C, C++ programming languages is similar):

Program code

```
Var
    EnumVar: IEnumVariant;
    rgvar: OleVariant;
    fetched: Cardinal;

//For Each Next
EnumVar := (S7Prog.Next._NewEnum) as IEnumVariant;
While (EnumVar.Next(1,rgvar,fetched) = S_OK) Do Begin
    Cont := IS7Container(IDispatch(rgvar)); // block container,
    Check sources
    If (Cont.ConcreteType = S7BlockContainer) Then Break;
    Cont := NIL;
End;
```

6.7 Software upgrade

A general PLC reset should be performed to achieve a defined initial state before the PLC software is upgraded. In this case, among other things, all user data (program and data blocks) will be deleted. The PLC general reset is described in:

Further information:
Commissioning Manual IBN CNC: NC, PLC, Drive, General Tips,
Section: PLC general reset

Generating a new SIMATIC S7 project

In normal cases, the new PLC basic program is to be linked-in for a new NCU software version. The basic programs blocks must be loaded into the user project for this purpose. If the following program and data blocks are already in the user project, then these should not be transferred with the blocks of the basic PLC program: OB1, OB40, OB82, OB86, OB100, FC12 and DB4. These have been possibly changed by the user, and should not be overwritten. The new basic program must be linked with the user program. The following procedure must be taken into account:

1. Generate the text or source file of all user blocks before copying the basic PLC program.
2. Copy the new basic program blocks into the SIMATIC S7 project (for a description, see Section "Application of the basic program (Page 39)")
3. All user programs "*.awl" must be recompiled in the correct order! (See also "Machine program (Page 41)")
4. This newly compiled SIMATIC S7 project should then be downloaded with STEP 7 into the PLC.

However, it is normally sufficient to recompile the organization blocks (OBs) and the instance data blocks of the S7 project. Prior to upgrading, therefore, only the sources for the organization blocks and the instance data blocks are generated.

NC variables

The latest NC VAR selector can be used for each NC software release (even older versions). The variables can also be selected from the latest list for earlier NC software versions. The data content in DB120 (default DB for variables) does not depend on the software version. This means, variables selected in an older software version need not be reselected when the software is upgraded.

6.8 I/O modules (FM, CP modules)

Additional packages for STEP 7 are generally required for more complex I/O modules (FM, CP modules). Support blocks (FC/FB) are provided in these additional packets. The blocks contain specific functions for operating the relevant module. These functions can be parameterized and called in the user program.

Identical numbers

If handling and basic program blocks have identical numbers, the block numbers of the basic program must remain unchanged. The block numbers of the handling blocks must be renamed to free numbers via STEP 7.

6.9 Troubleshooting

This section describes problems which may occur, their causes and remedies and should be read carefully before hardware is replaced.

Errors, cause/description and remedy			
Serial no. error information	**Errors**	**Cause/description**	**To correct or avoid errors**
1	No connection via MPI to PLC.	The MPI cable is not plugged in or is defective. Possibly, the STEP 7 software is also not correctly configured for the MPI card.	Test: Create a link with the programmer in the STEP 7 editor by means of connection "Direct_PLC". A number of node addresses must be displayed here. If they do not appear, the MPI cable is defective/not plugged in.
2	PLC cannot be accessed in spite of PLC general reset.	A system data block SDB 0 has been loaded with a modified MPI address. This has caused an MPI bus conflict due to dual assignment of addresses.	Disconnect all MPI cables to other components. Create the link "Direct_PLC" with the programmer. Correct the MPI address.
3	All four LEDs on the PLC flash (DI disaster)	A system error has occurred in the PLC. **Measures:** The diagnostic buffer on the PLC must be read to analyze the system error in detail. To access the buffer, the PLC must be stopped (e.g. set "PLC" switch to position 2). A hardware reset must then be performed. The diagnostic buffer can then be read out with STEP 7. Relay the information from the diagnostic buffer to the Hotline / Development Service. A general reset must be carried out if requested after the hardware RESET. The diagnostic buffer can then be read with the PLC in the Stop state.	Once the PLC program has been RESET or reloaded, the system may return to normal operation. Even in this case, the content of the diagnostic buffer should be sent to the Development Office.

Coupling of the PLC CPU 7

7.1 General information

A CPU of the S7-300 automation system is used as PLC for the SINUMERIK 840D sl. The PLC-CPU is integrated into the NCU component as a sub-module. A reference to the performance data of the PLC CPU can be found in Section "Key data of the PLC CPU (Page 31)".

7.2 Properties of the PLC CPU

The PLC integrated in the SINUMERIK 840D sl generally has the same functionality as the corresponding SIMATIC S7-300 PLC.

For differences, see reference in Section "Key data of the PLC CPU (Page 31)".

Owing to differences in their memory system as compared to a SIMATIC S7-300 PLC, certain functions are not available (e.g. save blocks on memory card, save project on memory card).

Note

As with the PLC integrated in SINUMERIK, there is no automatic start of the PLC after power failure and recovery for a SIMATIC S7-300 PLC when a "PLC stop" is triggered by an operator action on the programming device. For safety reasons, the PLC remains in the stop state with an appropriate diagnostic entry. You can start the PLC only by an operator action on the programming device, "Execute a restart", or via the mode selector "Stop" > "Run" (warm restart).

7.3 Interface with integrated PLC

Physical interfaces

With the SINUMERIK 840D sl, the PLC integrated in the NCU offers the option of exchanging signals between the NC and PLC directly via a dual-port RAM.

Data exchange with the operator panel

Data exchange with the operator panel (e.g. TCU/OP) can be performed via Ethernet or PROFIBUS. With a connection via Ethernet, communication takes place via the integrated communication processor (CP 840D sl).

Data exchange with the machine control panel (MCP) and handheld unit (HHU) can be performed via MPI, PROFIBUS or Ethernet.

Programming devices should preferably be connected via Ethernet or via MPI (Multi-Point Interface) directly to the PLC.

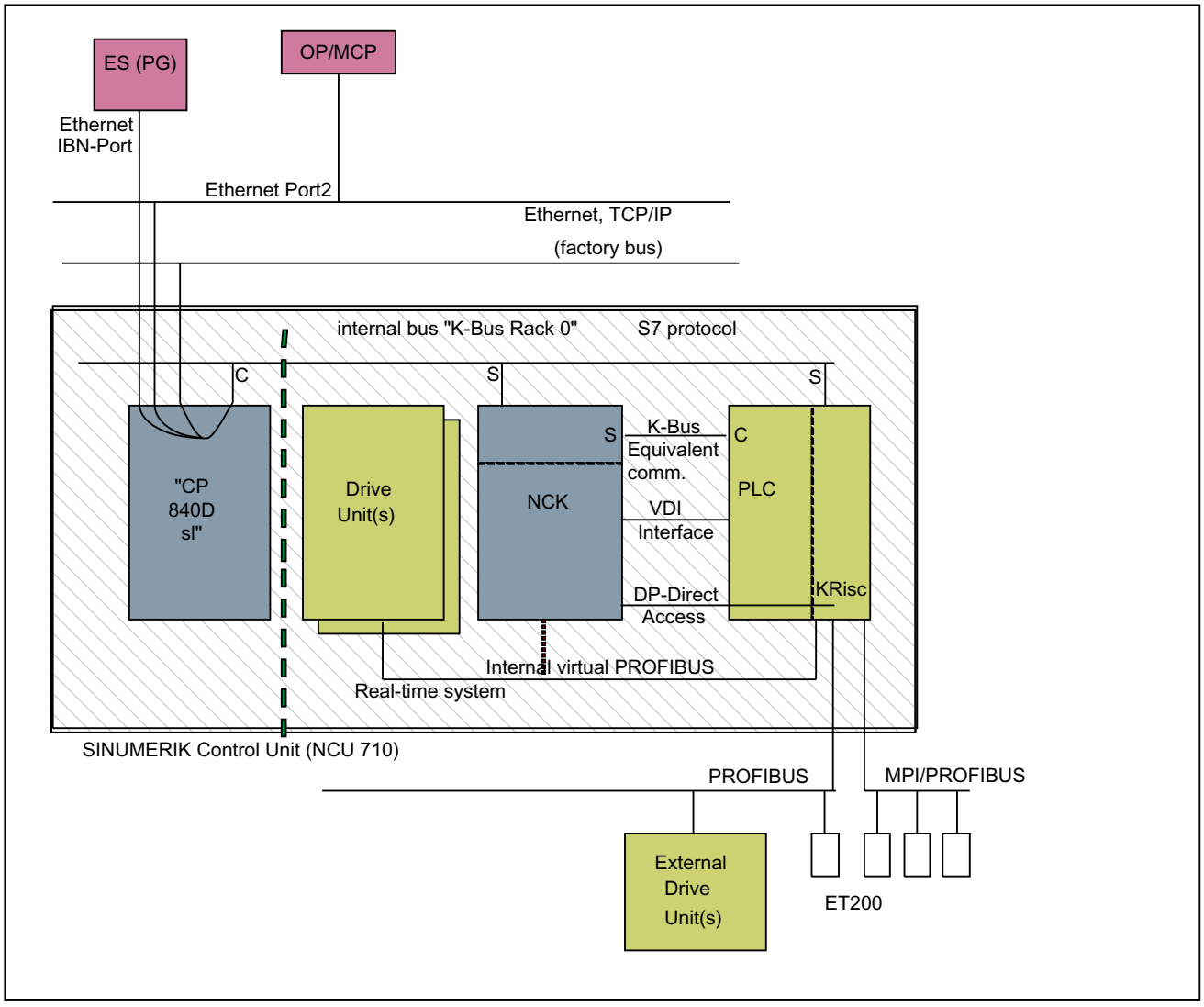


Figure 7-1 NC-PLC coupling on SINUMERIK 840D sl (integrated PLC)

Interface: NC/PLC

The data exchange between NC and PLC is organized by the basic program on the PLC side. The **status information**, such as "Program running", stored by the NC in the NC/PLC interface is copied to data blocks by the basic program at the beginning of the cycle (OB1) and can then be accessed in the user program (user interface). The **control signals** for the NC (e.g. NC start) entered in the interface data block by the user are also written to the internal DPR and transferred to the NC at the start of the cycle.

Workpiece-program-specific **auxiliary functions** transferred to the PLC are first evaluated by the basic program (alarmdriven) and then transferred to the user interface at the start of OB1. If the relevant NC block contains auxiliary functions that require that NC processing is interrupted (e.g. M06 for tool change), the basic program stops the decoding of the NC block initially for one PLC cycle. The user can then use the "read disable" interface signal to halt the block execution until the tool change has been completed. If, on the other hand, the relevant NC block only contains auxiliary functions, which do not require interruption of the decoding (e.g. M08 for

cooling medium on), the transfer of these "fast" auxiliary functions is directly acknowledged in OB40, so that decoding is only insignificantly influenced by the transfer to the PLC.

The evaluation and enabling of the **G commands** transferred from the NC are also alarm-driven and acknowledged, however they are transferred directly to the user interface. If a G command is evaluated at several points in the PLC program, differences in the information of the G command within one PLC cycle may arise.

In the case of **NC actions** triggered and assigned with parameters by the PLC (e.g. traverse concurrent axes), triggering and parameter assignment is performed using FCs and FBs, not interface data blocks. The FCs and FBs belonging to the actions are supplied together with the basic program. The FCs and FBs required must be loaded by the user and called in the PLC program of the machine manufacturer (machine program). For an overview of FC, FB and data blocks, sorted according to basic and extended functions, please refer to Section "Start-up of PLC programs".

Interface: HMI/PLC

HMI/PLC data exchange is performed via the integrated CP, whereby the HMI is always the active partner (client) and the PLC is always the passive partner (server). Data transferred or requested by the HMI is read from and written to the HMI/PLC interface area by the PLC operating system (timing: Cycle control point). From the viewpoint of the PLC application, the data is identical to I/O signals.

Interface: MCP/PLC or HHU/PLC (connection: Ethernet)

MCP/PLC and HHU (HT 2) / PLC data is exchanged via the integrated CP. The CP transfers the MCP/HHU signals to and fetches them from the PLC's internal DPR (Dual-Port RAM). On the PLC side, the basic program handles communication with the user interface. The basic-program parameters (FB1, DB7) define the operand areas (e.g. I/O areas) and the start addresses.

Interface: MCP/PLC (connection: PROFIBUS)

MCP/PLC data exchange takes place via the PLC's PROFIBUS. The MCP's I/O addresses are to be placed in the PLC's process image area and via HW configuration in STEP 7. The MCP*In, MCP*Out pointer variables must be set to the same addresses. The selected DP slave number must be entered in MCP*BusAdr.

Interface: HHU/PLC (connection: MPI)

The HHU/PLC data exchange is performed via the MPI interface on the PLC. The "Communication with global data (GD)" service is used for this purpose (see also STEP 7 User Manual). The PLC operating system handles the transfer of signals from and to the user interface. The STEP 7 "Communication configuration" configuring tool is used to define both GD parameters as well as operand areas (e.g. I/O areas) and their start addresses.

7.4 Diagnostic buffer on PLC

The diagnostic buffer of the PLC (readable using STEP 7) will enter diagnostic information on the PLC operating system.

Interface structure

8

8.1 Interface

Interface DBs

Mapping in interface DBs is necessary due to the large number of signals exchanged between the NC and PLC. These are global DBs from the viewpoint of the PLC program.

8.2 PLC/NCK interface

General

The PLC/NC interface comprises a data interface on one side and a function interface on the other. The data interface contains status and control signals, auxiliary and G commands, while the function interface is used to transfer jobs from the PLC to the NC.

Data interface

The data interface is subdivided into the following groups:

- NC-specific signals
- Mode-group-specific signals
- Channel-specific signals
- Axis/spindle/drive-specific signals

Function interface

The function interface is formed by FBs and FCs. The figure below illustrates the general structure of the interface between the PLC and the NC.

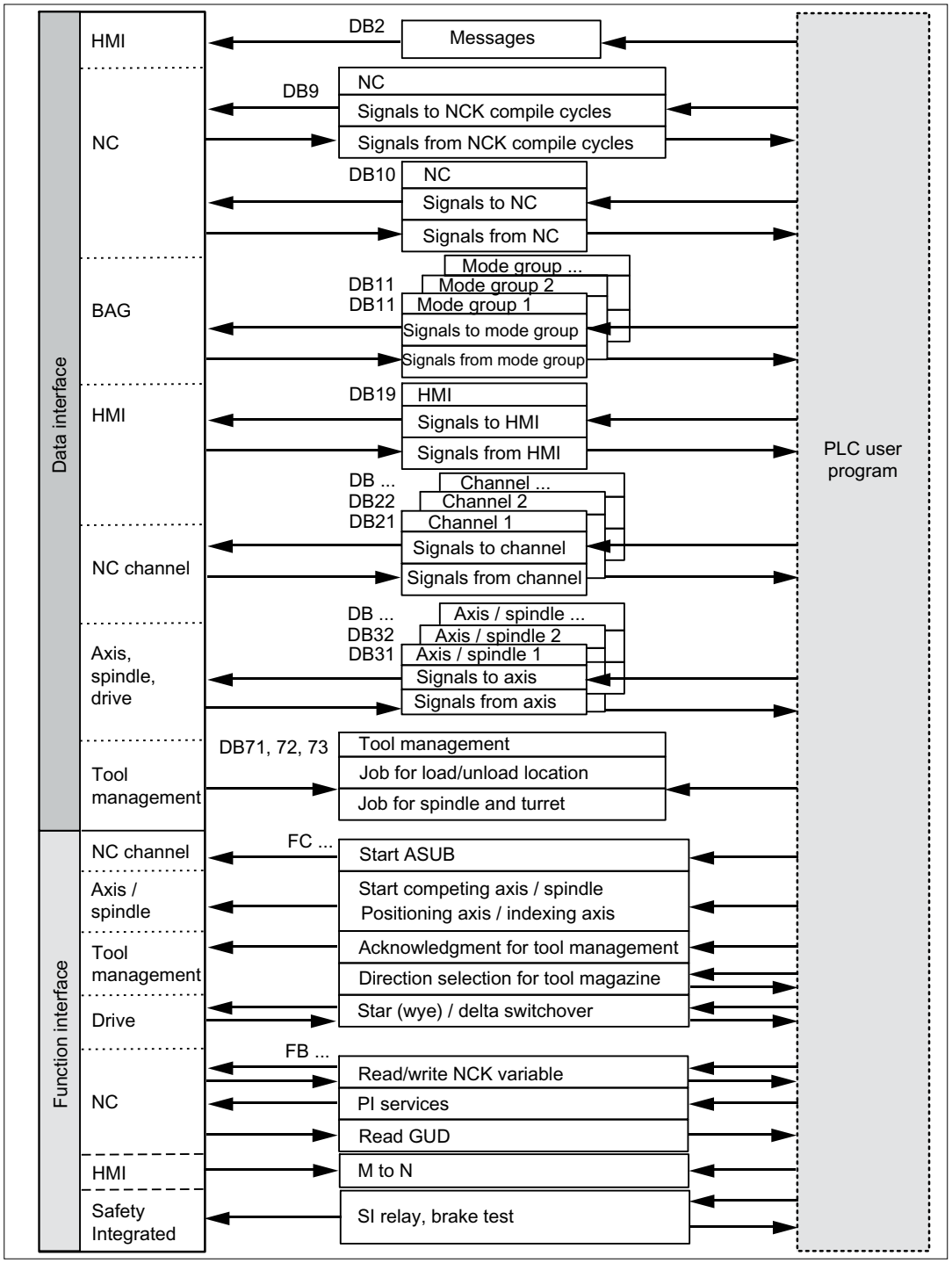


Figure 8-1 PLC-NC user interface

Compile cycle signals

The user has the option of defining compile-cycle data for exchanging with the PLC. A data block (DB9) with the following properties must be created as the PLC user interface for compile cycles:

Data block number:	9
Data block name:	freely selectable
Optimized block access:	no
Minimum size:	Sum of the input bytes and output bytes for compile cycles ($MN_CC_VDI_IN_DATA + $MN_CC_VDI_OUT_DATA)

The associated signals, which are dependent on the compile cycles, are transmitted cyclically at the start of OB1. First, signals are transferred from the PLC to the NC, then from the NC to the PLC. The signals from the PLC to the NC, whose length is specified by machine data $MN_CC_VDI_IN_DATA, are defined at the beginning of DB9. Behind that in DB9, the signals from the NC to the PLC, whose length is specified by machine data $MN_CC_VDI_OUT_DATA, are defined.

To enable data exchange between the NC and PLC, the user may have to synchronize the NC and PLC as necessary, e.g. using the semaphore technique. Signal transmission is asynchronous between NC and PLC. This means, for example, that active NC data transmission can be interrupted by the PLC. This can mean that data is not always consistent.

PLC/NC signals

The group of signals from the PLC to NC includes:

- Signals for modifying the high-speed digital I/O signals of the NC
- Keyswitch and emergency stop signals

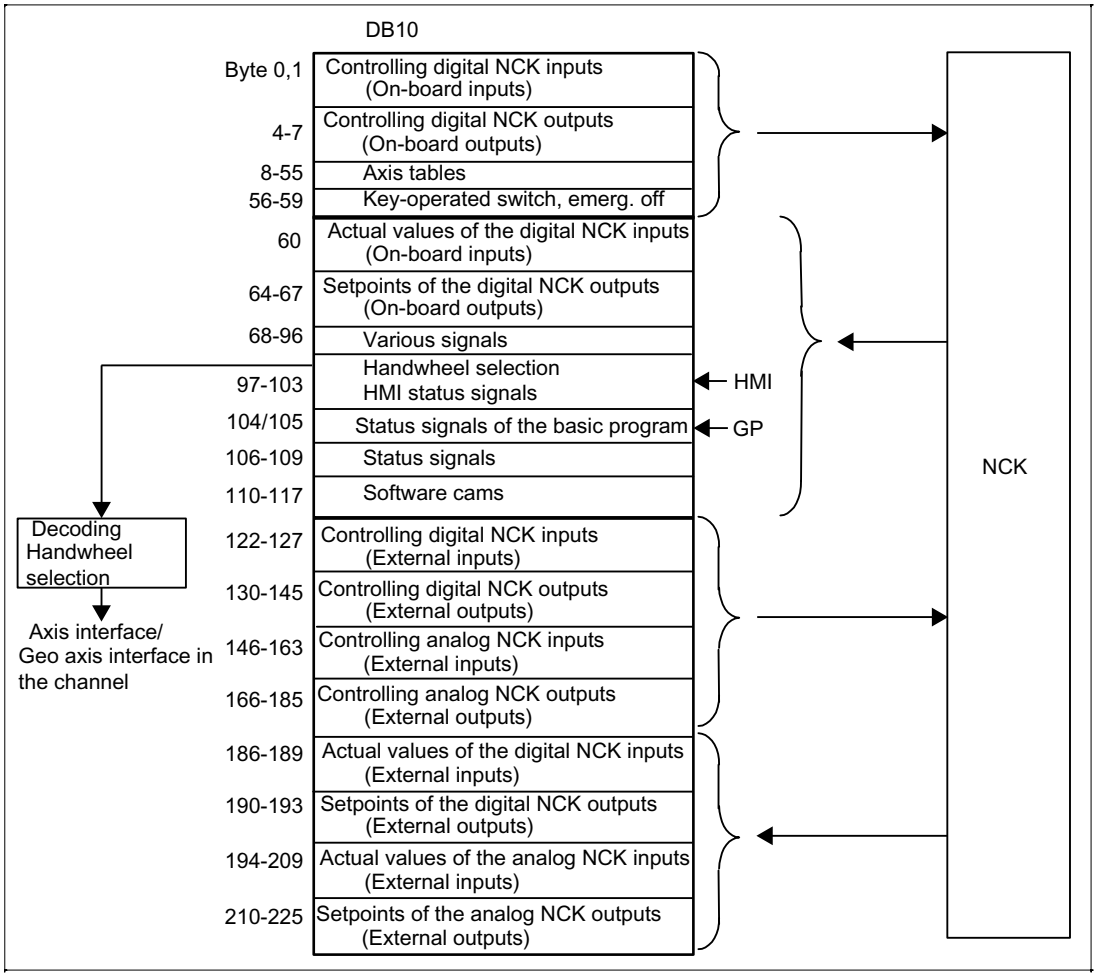


Figure 8-2 PLC/NC interface

NC/PLC signals

The group of signals from the NC to PLC includes:

- Actual values of the digital and analog I/O signals of the NC
- Ready and status signals of the NC

Also stored in this group are the HMI handwheel selection signals and the channel status signals.

The signals for handwheel selection are decoded by the basic program and entered in the machine/axis-specific interface.

Digital/analog inputs/outputs of the NC

The following must be noted with respect to the digital and analog inputs and outputs of the NC:

Inputs:

- All input signals or input values of the NC are also transferred to the PLC.
- The transfer of signals to the NC part program can be suppressed by the PLC. Instead, a signal or value can be specified by the PLC.
- The PLC can also transfer a signal or value to the NC even if there is no hardware for this channel on the NC side.

Outputs:

- All signals or values to be output are also transferred to the PLC.
- The NC can also transfer signals or values to the PLC even if there is no hardware for this channel on the NC side.
- The values transferred by the NC can be overwritten by the PLC.
- Signals and values from the PLC can also be output directly via the NC I/O.

Note

When implementing the digital and analog NC I/O, the information contained in the following documentation must be taken into account:
Further information
Function Manual Basic Functions; Digital and Analog NC I/O

PLC / mode group signals

The mode signals set by the machine control panel or the HMI are transferred to the operating mode group of the NC. These apply to all NC channels. Several mode groups can be optionally defined in the NC.

The mode group reports its current status to the PLC.

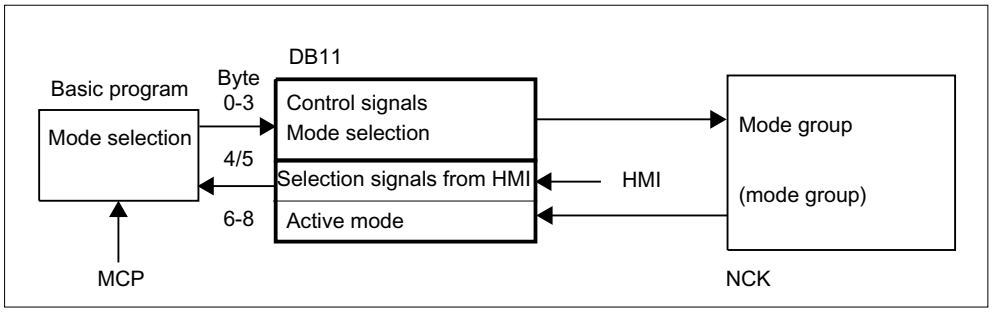


Figure 8-3 PLC / mode group interface

PLC/NC signals

The signal groups below must be considered on the interface:

- Control/status signals
- Auxiliary commands / G commands

- Tool management signals
- NC functions

The **control/status signals** are transferred cyclically at the start of OB1. The signals entered in the channel-specific interface by the HMI (HMI signals are entered by the PLC operating system) are also transferred at this time if they have been defined on the HMI operator panel, not on the MCP.

Auxiliary commands and G commands are entered in the interface data blocks in two ways. First, they are entered with the change signals.

- The **M signals** M00 - M99 (they are transferred from the NC with extended address 0) are also decoded and the associated interface bits set for the duration of one cycle.
- For **G commands**, only the groups selected via machine data are entered in the interface data block.
- The **S values** are also entered together with the related M signals (M03, M04, M05) in the spindle-specific interface. The axis-specific feedrates are also entered in the appropriate axis-specific interface.

When the **tool management (magazine management)** function is activated in the NC, the assignment of spindle or revolver and the loading/unloading locations are entered in separate interface DBs (DB71-73).

The triggering and parameter assignment of **NC functions** is performed by means of PLC function calls.
The following function calls are available:

- Position a linear axis or rotary axis
- Position an indexing axis
- Start a prepared asynchronous subprogram (ASUP)
- Read/write NC variables
- Update magazine and tool motion

Some of the above functions are described in their own function documentation.

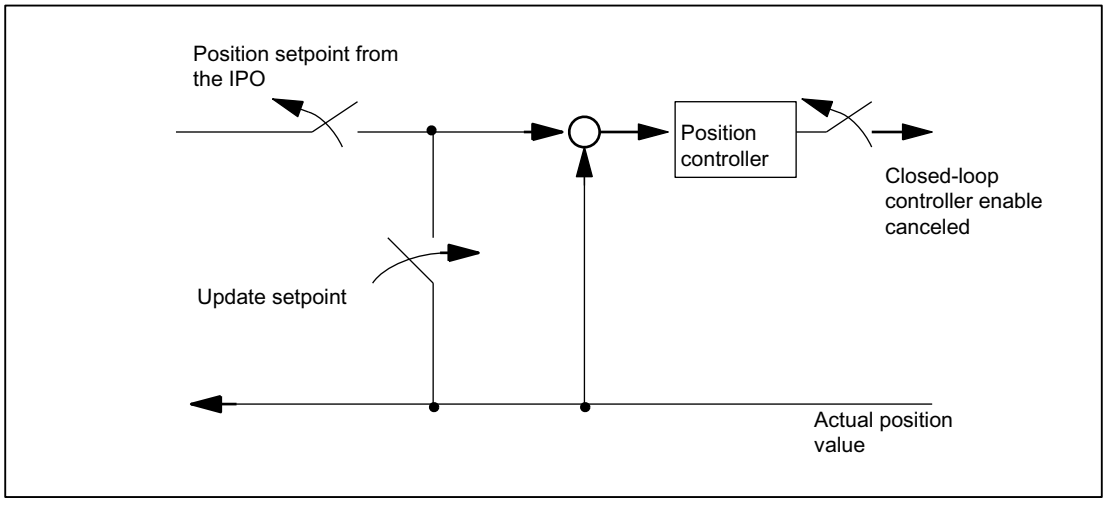


Figure 8-4 PLC/NC channel interface

PLC/axis, spindle, drive signals

The axis-specific and spindle-specific signals are divided into the following groups:

- Shared axis/spindle signals
- Axis signals
- Spindle signals
- Drive signals

The signals are transferred cyclically at the start of OB1 with the following exceptions.

Exceptions include:

- axis-specific F value
- M value
- S value

An **axis-specific F value** is entered via the M, S, F distributor of the basic program if it is transferred to the PLC during the NC program processing.

The **M and S value** are also entered via the M, S, F distributor of the basic program if one or both values require processing.

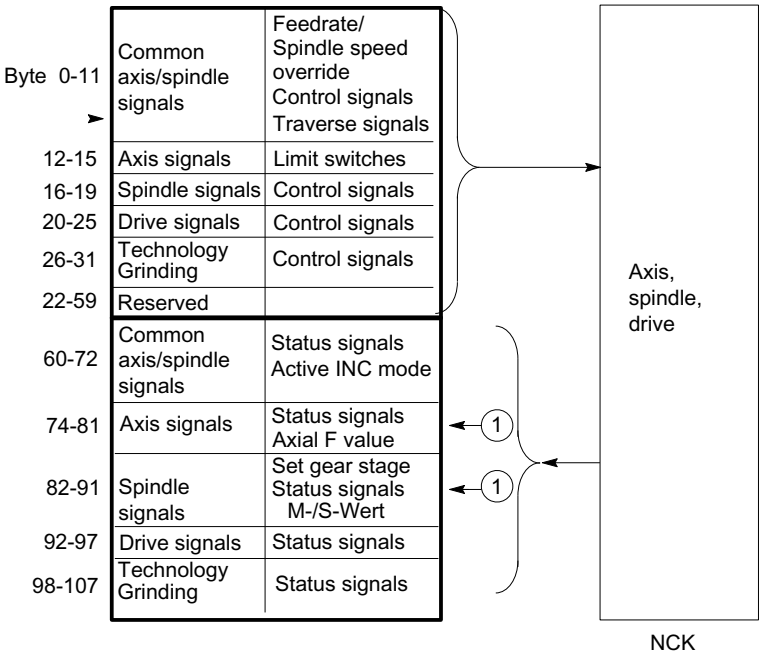


Figure 8-5 Interface PLC / axes/spindles/drives (DB31, …)

8.3 Interface PLC/HMI

General

The following groups of functions are required for the PLC/HMI interface:

- Control signals
- Machine operation
- PLC messages
- PLC status display

Control signals

Some control signals are signal inputs, for example, via the machine control panel, which have to be taken into account by the HMI. This group of signals includes, for example, display actual values in MCS or WCS, key disable, etc. These are exchanged with the HMI via a separate interface data block (DB19).

Machine operation

All operator inputs, which lead to response actions on the machine, are monitored by the PLC. Operator actions are usually performed on the machine control panel (MCP). However, it is also possible to perform some operator actions on the HMI, e.g. mode selection.

The PLC operating system enters the operating signals sent by the HMI directly into the interface data blocks. As standard, the basic program routes these operating signals in such a way that, provided equivalent operator actions are available, these can be performed either on the HMI **or** on the MCP. If required, the user can switch off operation via HMI using an FB1 parameter "MMCToIF".

PLC messages

The signaling functions are based on the system diagnostic functions integrated in the operating system of the AS 300. These have the following characteristics:

- The PLC operating system enters all important system states and state transitions in a **diagnostics status list**. Communication events and I/O module diagnostics data (for modules with diagnostic functions) are also entered.
- Diagnostics events, which lead to a system stop, are also entered with a time stamp in a **diagnostic buffer** (circular buffer) in the chronological order of their occurrence.
- The events entered in the diagnostic buffer are automatically transmitted to human machine interface systems (OP or HMI) via the bus systems once these have issued a ready signal (message service). "Transfer to the node ready" is a function of the PLC operating system. Receipt and interpretation of the messages are executed by the HMI software.
- The PLC user program can also use SFCs (System Function Calls) to enter messages in the diagnostic buffer or ALARM S/ALARM SQ buffer.
- The events are entered in the alarm buffer.
  The associated message texts must be stored on the OP or HMI.

An FC (FC10) for message acquisition is prepared in conjunction with the basic program. This FC records events, subdivides them into signal groups and reports them to the HMI via the alarm buffer.

The message acquisition structure is shown in the figure "Acquisition and signaling of PLC events". The features include:

- Bit fields for events related to the NC/PLC interface are combined in a single data block (DB2) with bit fields for user messages.
- Bit fields are evaluated at several levels by FC10.
  - **Evaluation 1;** acquisition of **group signals**
    A group signal is generated for each group of signals if at least one bit signal is set to "1". This signal is generally linked to the disable signal of the NC/PLC interface (on modules with diagnostic functions). The group signals are acquired completely in cycles.
  - **Evaluation 2;** acquisition of **alarm messages**
    A fixed specification exists to define which signals in a group generate an alarm message when they change from "0" to "1".
  - **Evaluation 3;** acquisition of **operating signals**
    A fixed specification exists to define which signals in a group generate an operational message.
- The scope of the user bit fields (user area) is set by default to 10 areas with 8 bytes each, but the number of areas can also be adjusted to suit the requirements of the machine manufacturer via basic program parameters in FB1.

Acknowledgement concept

The following acknowledgement procedures are implemented for error and operational messages:

Operating messages are intended for the display of normal operating states as information for the user. Acknowledgement signals are, therefore, not required for this type of message. An entry is made in the diagnostic status list for incoming and outgoing messages. The HMI maintains an up-to-date log of existing operating messages using the identifiers "operating message arrived" and "operating message gone".

Alarm messages display error states on the machine, which will usually lead to the machine being stopped. Where several errors occur in rapid succession, it is important to be able to distinguish their order of occurrence for troubleshooting purposes. This is indicated, on the one hand, by the order in which they are entered in the diagnostic buffer and on the other, by the time stamp, which is assigned to every entry.

If the cause of the error disappears, the associated alarm message is only deleted if the user has acknowledged it (e.g. by pressing a key on the MCP). In response to this signal, the "Message acquisition" FC examines which of the reported errors have disappeared and enters these in the diagnostic buffer with the entry "Alarm gone". This enables the HMI to also maintain an up-to-date log of pending alarm messages. The time of day indicating the time at which the error occurred is maintained for messages, which are still pending (in contrast to a received interrogation).

STEP 7

A tool can be started in the SIMATIC Manager via menu item "Target system" > "CPU messages". Alarms and messages can be displayed by number using this tool. To do this, acivate the "Alarm" tab and enter a check mark under "A" in the upper half of the screen.

User program

The user PLC program only has to call the basic program block FC10 with appropriate parameter settings in the cyclic program section and set or reset the bit fields in DB2. Everything else is handled by the basic program and the HMI.

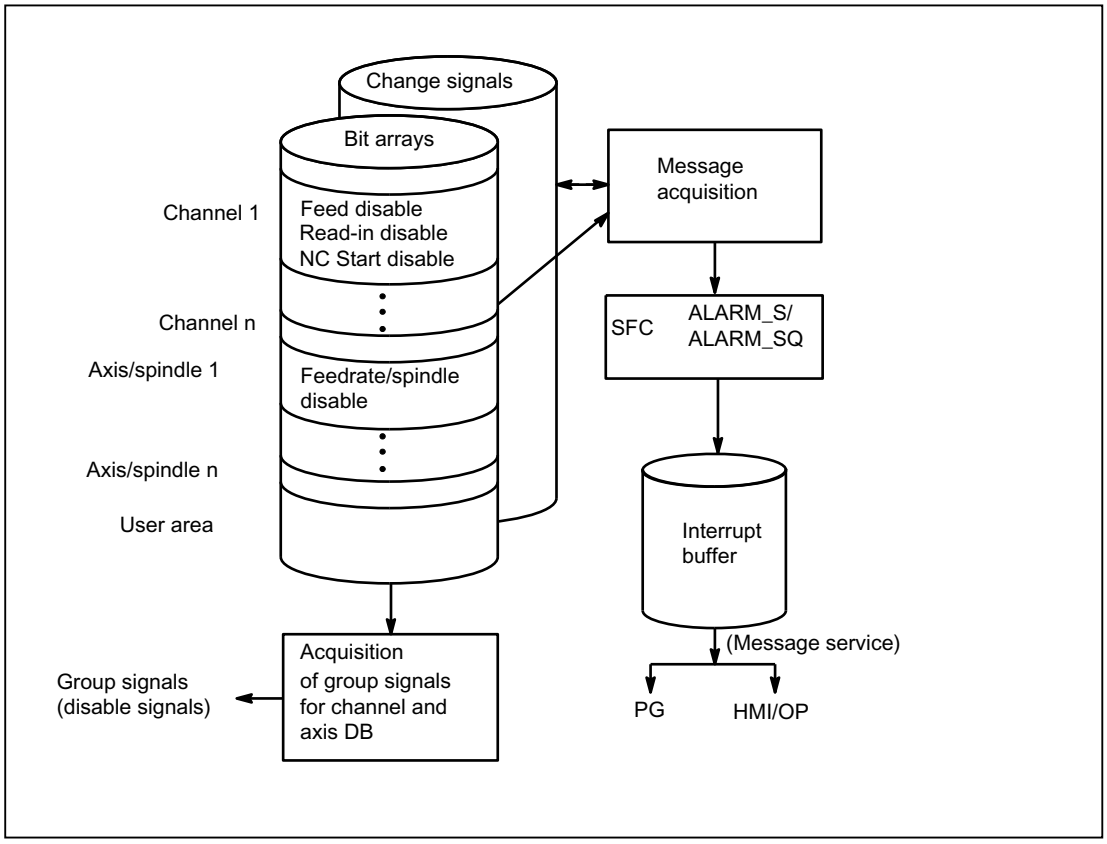


Figure 8-6 Acquisition and signaling of PLC events

Extensions of the PLC alarms via block FC10

The FB1 parameter "ExtendAlMsg" selects the PLC alarm mechanism.

If "ExtendAlMsg:= FALSE" the earlier process of the FC10 with the DB2 is active as bit array data block. The known restrictions regarding the number of channels and axes are applicable.

On the other hand, in case of "ExtendAlMsg:= TRUE" the extension of the FC10 becomes active. DB2 and DB3 are created just as before. The user must set or reset the bits in DB2. The parameter setting via message and alarm and a parameter setting of the numeric value of the 2nd decade of the user alarms are contained in DB5.

The extensions are:

- Support for 10 channels, 31 axes, 64 user areas (the number of user areas should be entered in the FB1 parameter "MsgUser").
- Areas for feed stop, read-in disable, etc. are available without messages. The information from this area is stored on the interface in DB21, DB31 depending on FC10 parameter "ToUserIF" together with the related message bits as group signals. As such, the previous cumbersome handling of the signals is omitted.
- The alarms / messages also get the 16-bit integer additional value (%Z parameter in the alarm text) in addition to the alarm number for the user area 0. The user must write the 16-bit integer values in the DB2 in the array variable ZInfo0 parallel to setting an alarm bit. An integer value is available for each bit in the user area 0, see UDT1002 in the basic program.
- The user messages can be parameteized in the second decade of the message number in the numerical range 0 to 9. The display value of the second decade must be written by the user in the DB5 in the array variable UserDek2No. A number can be defined for each user area, see DB5 in the basic program.
  The value 0 is set by default for the second decade.

The structuring of the DB2 in UDT1002 can be recognized (basic program). In case of new alarm functions, the UDT1002 must be assigned symbolically to the DB2.

At the start of DB2 there are bit arrays for signals without alarm generation. This is followed by an array of size 64 integer values for additional info about the user area 0.

Thereafter follow the areas, which also trigger alams / messages (see List manual) These areas are extended to 10 channels, 31 axes.

Switching over a user program from the small DB2 layout to the large DB2 layout (UDT2 → UDT1002)

With the small DB2 layout, i.e. if DB2 is derived from UDT2 - and FB1 parameter "ExtendAlMsg = FALSE" is used - then a max. of 8 channel areas, max. of 18 axis areas and a max. of 31 user areas (depending on FB1 parameter "MsgUser") can be used. As default setting, 10 user areas are defined in the UDT2. When using more than 10 user areas (FB1 parameter "MsgUser" > 10), then users must extend UDT2 if DB2 is to be symbolically accessed.

With the large DB2 layout, i.e. if DB2 is derived from UDT1002 - and FB1 parameter "ExtendAlMsg = TRUE" is used - then you can generate alarms for 10 channels, 31 axes and 64 user areas.

If it is no longer possible to generate alarms with the restrictions of the small DB2 layout, then it may be necessary to switch over an already existing project to the large layout. The switchover procedure depends on how DB2 is accessed in the project. The various procedures are described in the following.

Switchover for symbolic programming

The source container of the basic program contains the "udt2_for_Convert.awl" file, which contains the following structural element from UDT1002:

- ChanA as array of 1 ... 8 for switching over the 8 channel ranges.
- AxisA as array of 1 ... 18 for switching over the 18 axis/spindle ranges
- UserA as array from 0 ... 31 for switching over user areas

From this STL file, a UDT can be generated that assigns the offsets of the small DB2 version to the symbolic names of the large DB2 version.

Table 8-1 Symbolic signal designator from the channel area

DB2 without alarm extension	DB2 with alarm extension
C**j**.FDD_5**j**00xx[i]	ChanA[**j**].FDD_5C00xx[i]
C**j**.FDD_RID_5**j**01xx[i]	ChanA[**j**].FDD_RID_5C01xx[i]
C**j**.RID_5**j**02xx[i]	ChanA[**j**].RID_5C02xx[i]
C**j**.NCSD_5**j**03xx[i]	ChanA[**j**].NCSD_5C03xx[i]
C**j**.FdStop_1_5**j**11xx[i]	ChanA[**j**].FdStop_1_5C11xx[i]
C**j**.FdStop_2_5**j**12xx[i]	ChanA[**j**].FdStop_2_5C12xx[i]
C**j**.FdStop_3_5**j**13xx[i]	ChanA[**j**].FdStop_3_5C13xx[i]

j (shown bold): Channel number (1 … 8)

Table 8-2 Symbolic signal designator from the axis/spindle area

DB2 without alarm extension	DB2 with alarm extension
_**j**FdStop600**j**xx[i]	AxisA[**j**].AAFdStop60AAxx[i]
_**jj**FdStop60**jj**xx[i]	AxisA[**jj**].AAFdStop60AAxx[i]

j or **jj** (shown bold): Axis/spindle number (1 ... 9 or 10 … 18)

Table 8-3 Symbolic signal designator from the user area

DB2 without alarm extension	DB2 with alarm extension
A70**jj**xx[i]	UserA[**jj**].A70UUxx[i]

jj (shown bold): User range (00 … 31)

Procedure:

1. Select the block folder and under "Object properties" > "Operand priority" ensure that "Absolute value has priority" is active.
2. Compile source "udt2_for_Convert.awl".
   The new symbols of the large layout are now already available in UDT2.
3. Generate sources of blocks, which contain symbolic assignments to the DB2. To do this, in the LAD/STL/FBD KOP/AWL/FUP editor, under "File" > "Generate source", select the required blocks and create an STL source.
4. In the symbol table, assign UDT1002 to the DB2.
5. In the SIMATIC Manager under "Sources" compile the currently generated source.

Now all the alarm allocations are assigned to the new data areas in the DB2 and now only the parameter "ExtendAlMsg" at FB1 must be set to True.

Switchover for direct absolute access to the DB2

For instance, this is the case if the user first opens the DB2 in the program code, and then directly accesses it (at "ALMSG_DB"; S DBX180.0;).

Procedure:

1. Using the cross reference list (menu command "Tools" > "Reference data" > "Display"), search for blocks in the user program that's contain DB2 access operations.
2. Open the program locations involved one after the other and switchover the offset addresses of the relevant DB2 signals.
   The new offset address results from the sum of the previous address and the "offset".

Note

Searching for DB2 accesses in the user program

Not all DB2 access operations are found via the reference data. Accesses via BLK_DB or pointer are not detected.

A comparison of the particular offset addresses and/or the address offset are listed in the following table.

Table 8-4 Addresses in the DB2 versions

Alarm number	DB2 without alarm extension		Off-set	DB2 with alarm extension	
	Area	Offset		Offset	Area
-				0	Chan (Channel 1 ... 10)
-				120	Axis (Axis/spindle 1 ... 31)
Addition-al value				182	As of V6.0: AddVa-lUserA0 (AddValU-serA0 0 ... 63) (Up to V4.8: ZInfo0 0 ... 63)
5C0000 ... 5C1315	(C1 ... C8) Channel 1 ... 8	Byte 0 ... <last byte for channel area>	**310**	Byte 310 ... <last byte for channel area>	ChanA Channel 1 ... 10
60AA00 ... 60AA15	(Axis areas) Axis/spindle 1 ... 18	Byte 144 ... <last byte for axis area>	**346**	Byte 490 ... <last byte for axis area>	AxisA Axis/spindle 1 ... 31
70UU00 ... 70UU63	(User area) User area 0 ... 31	Byte 180 ... <last byte for user area>	**374**	Byte 554 ... <last byte for user area>	UserA User area 0 ... 63

Conversion for indirect access via BLKMOV

The following adaptations are required if DB2 is not directly accessed, but instead BLKMOV is used, for example to copy memory areas of its own data blocks to DB2.

Within the individual DB2 areas, the offset addresses of both DB2 versions are offset by a specific value. This offset is, however, different for each DB2 area.

A check has to be made whether BLKMOV copy operations are used that copy a memory area that is larger than a DB2 area. If this is the case, then copying must be split up over several operations. The new offset address must then be considered and adapted depending on the particular DB2 area.

8.4 PLC/MCP/HHU interface

General

There are different connection options for the machine control panel (MCP) and the handheld unit (HHU). This is in part due to the history of the MCP and HHU. This description focuses primarily on the connection of the Ethernet components.

On the SINUMERIK 840D sl, the machine control panel (MCP) and handheld unit (HHU) are connected via the Ethernet bus, which also links the TCU to the NCU. The advantage of this is that only one bus cable is required to connect the operating unit.

Topology SINUMERIK 840D sl

On the 840 D, the machine control panel and the handheld unit are connected to the CP 840D sl Ethernet bus (see Figure below). Where the connection of further keys and displays is required for customized operator panels, an additional keyboard interface (machine control panel without operating unit) can be used. For each keyboard interface, 64 pushbuttons, switches, etc. and 64 display elements can be connected via ribbon cable.

The signals sent from the MCP are copied to the PLC's DPR (Dual-Port RAM) by the integrated Ethernet CP-840D sl. The basic program of the PLC enters the incoming signals in the input image configured at FB1. The NC-related signals are generally distributed by the basic program to the NC/PLC interface. If required, the signals can be modified by the user.

The signals from the PLC to the MCP (displays) are transferred in the opposite direction.

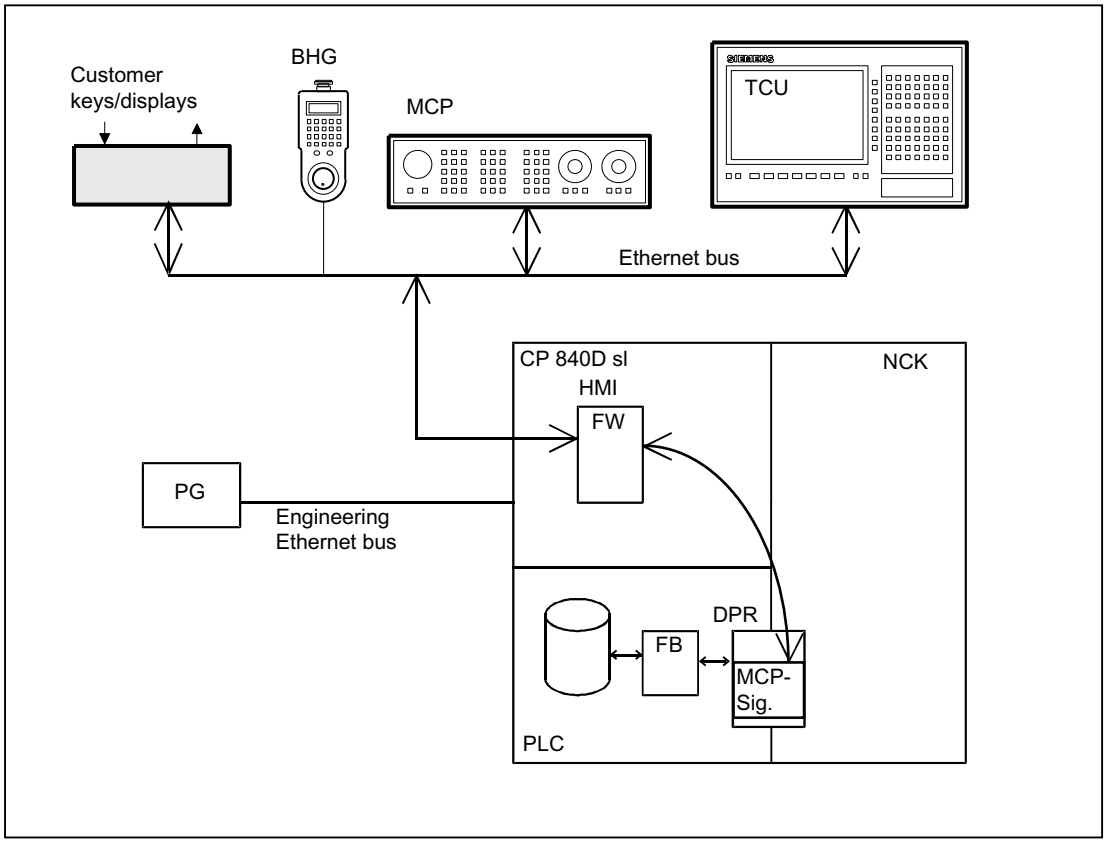


Figure 8-7 Connection of the machine control panel on 840D sl

Bus addresses

On Ethernet components, MAC and IP addresses or logic names are determining factors in respect of communication. The control system's system programs convert logic names into MAC or IP addresses. On the PLC, the numeric component of the logic name is used for communication. This numeric part is specified by the user to the FB1 using parameter "MCPxBusAdr".

The logical name of an MCP or HHU always begins with "DIP". This is followed by a number corresponding to the switch position of the MCP component (e.g. DIP192, DIP17).

MCP interface in the PLC

The signals from the machine control panel are routed by default via the I/O interface to the PLC area. A distinction must be made between NC and machine-specific signals. NC-specific key signals are distributed to the relevant mode-group-, NC-, axis- and spindle-specific interface by FC19 (or FC24, FC25, FC26, depending on the type of MCP) by default. The reverse applies to the associated status signals which are routed to the MCP interface. For this purpose, FC19 or the other blocks mentioned above must be called in the **user program**.

Customized keys, which can be used to trigger a wide range of machine functions, must be evaluated directly by the user program. The user program also routes the status signals to the output area for the LEDs.

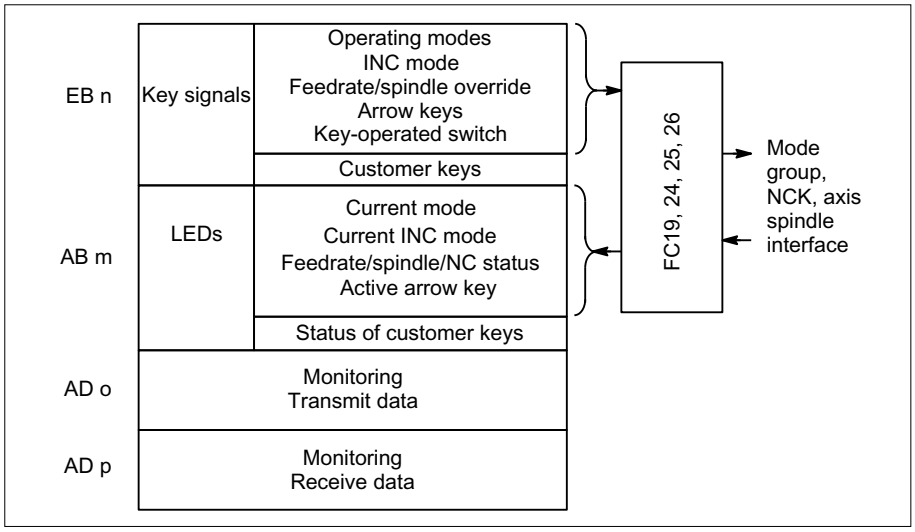


Figure 8-8 Interface to and from machine control panel

Structure and functions of the basic program

9

General

The PLC program has a modular structure. The organization blocks (OB) form the interface between the operating system and the basic and user programs.

- Restart (warm restart) with start-up and synchronization (OB100)
- Cyclic operation (OB1)
- Process alarms (OB40)
- Asynchronous errors: Diagnostics alarm (OB82), module failure (OB86)

The calls of the function blocks of the basic and user programs must be programmed by the user in the organization blocks (OB).

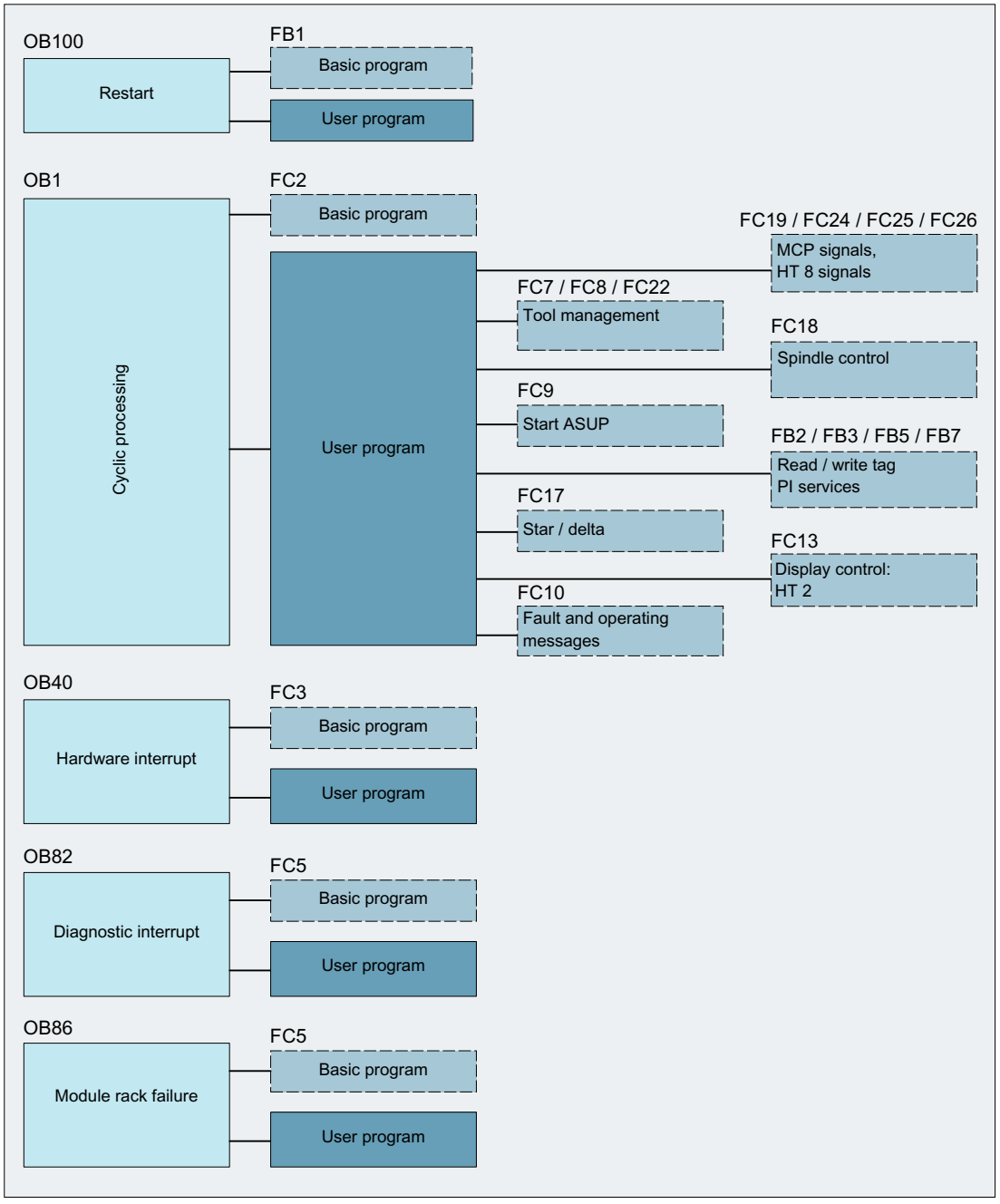


Figure 9-1 Structure of the basic program (principle)

9.1 Startup and synchronization of NCK PLC

Loading the basic program

The basic program must be loaded with the S7 tool when the PLC is in the Stop state. This ensures that all blocks in the basic program will be initiated correctly the next time they are called. Otherwise, undefined states can occur in the PLC (e.g. blinking of all PLC LEDs).

Startup,

The synchronization of NC and PLC is performed during start-up. The system and user data blocks are checked for integrity and the most important basic program parameters are verified for plausibility. In cases of errors, the basic program produces an alarm (visible on HMI) and switches the PLC to the Stop state.

A warm restart is not provided, i.e. following system initialization, the operating system runs organization block OB100 and always commences cyclic execution at the start of OB1.

Synchronization

The PLC is synchronized with the HMI, NC and CP during power-up.

Sign-of-life

After a correct initial start and the first complete OB1 cycle (basic setting cycle) the PLC and NC continuously exchange sign-of-life signals. If the sign of life from the NC fails to materialize, the PLC/NC interface is initialized and the signal "NC CPU ready" in DB10 is set to FALSE.

9.2 Cyclic operation (OB1)

General

The NC/PLC interface is processed completely in cyclic mode. From a chronological viewpoint, the basic program runs ahead of the user program. In order to minimize the execution time of the basic program, only the control/status signals are transferred cyclically; transfer of the auxiliary commands and G commands takes place only on request.

The following functions are performed in the cyclic part of the basic program:

- Transmission of the control/status signals
- Distribution of the auxiliary functions
- M decoding (M00 - M99),
- M, S, F distribution
- Transfer the MCP signals via NC
- Acquisition and conditioning of the user errors and operational messages.

Control/status signals

A shared feature of the control and status signals is that they are bit fields. The basic program updates them at the start of OB1.

The signals can be subdivided into the following groups:

- General signals
- Mode-group-specific signals (such as modes)
- Channel-specific signals (such as program and feed modifications)
- Axis- and spindle-specific signals (such as feed disable)

Auxiliary and G commands

The auxiliary and G commands have the following characteristics:

- Transfer to the PLC is block-synchronous (referred to a part program block)
- Transfer is acknowledge-driven.
- The acknowledgment times have an immediate effect on the execution time of NC blocks containing auxiliary functions requiring acknowledgment.

The value range is presented in the table below:

Function	Structure		Value range		Data type	
	1st value	2nd value	1st value	2nd value	1st value	2nd value
G command		G command		255[1)]		Byte
M word	M group	M word	99	99.999.999	Word	DWord
S word	Spindle no.	S word	6	Floating point[2)]	Word	DWord
T word	Magazine no.	T word	99	65535	Word	Word
D word	-	D word	99	255	Byte	Byte
H word	H group	H word	99	Floating point	Word	DWord
F word	Axis no.	F word	18	Floating point	Word	DWord

1) relative number, transferred for each G group

2) corresponding STEP 7 format (24-bit mantissa, 8-bit exponent)

The M, S, T, H, D and F values sent by the NC are output together with the accompanying change signals to the **CHANNEL DB** interface via the auxiliary/G command distributor (see List Manual). The two values of the auxiliary function are transferred to the appropriate data word. The accompanying change signal is activated to 1 for one PLC cycle. When the change signal is reset, the acknowledgment is passed to the NC. The acknowledgment of high-speed auxiliary functions is performed when the basic program detects the auxiliary function.

In addition to distribution of the auxiliary and G commands, selected signals are processed as described below.

M decoder

M functions can be used to transfer switching commands and fixed-point values. Decoded dynamic signals are output to the **CHANNEL DB** interface for standard M functions (range M00 - M99) (signal duration = 1 cycle time).

G group decoders

In the case of G commands sent by the NC, the related groups are decoded and the current G number is entered in the corresponding interface byte of the CHANNEL DB, i.e. all active G commands are entered in the channel DBs. The entered G commands are retained even after the NC program has terminated or aborted.

Note

During system start-up, all G group bytes are initialized with the value "0".

M, S, F distributor

The M, S, F, distributor is used to enter spindle-specific M words M(1...6)=[3,4,5], S words and F words for axial feeds in the appropriate **spindle and axis data blocks**. The criterion for distribution is the extended address which is passed to the PLC for M words, S words and axial F words.

MCP signal transmission

Depending on the bus connection, the MCP signals are either transferred directly to the PLC or indirectly to the parameterized I/O areas via an internal procedure using the basic program.

User messages

The acquisition and processing of the user error messages and operational messages is performed by an FC in the basic program.

9.3 Time-interrupt processing (OB35)

The user must program **OB35** for time-interrupt processing. The default time base setting of OB 35 is 100 ms. A different time base can be selected using the STEP7 "HW Config" tools. However, the OB35 time setting must be at least 3 ms in order to avoid a PLC CPU stop. The stop is caused by reading of the HMI system state list during powerup of the HMI. This reading process blocks priority class control for approx. 2 ms. The OB35 with a time base set to a rather lower value is then no longer processed correctly.

9.4 Process-interrupt processing (OB 40)

A process interrupt **OB40** (interrupt) can, for example, be triggered by appropriately configured I/Os or by certain NC functions. Due to the different origin of the interrupt, the PLC user program must first interpret the cause of the interrupt in OB40. The cause of the interrupt is contained in the local data of OB40.

Further information
SIMATIC STEP 7 Description or Online Help of STEP 7

9.5 Diagnostic alarm, module failure processing (OB82, OB86)

General

In the event of a diagnostic alarm or failure of an I/O module, basic program block OB82 or OB86 is called. The basic program block FC5 "Diagnostic alarm and module failure" is called from these blocks.

Bus diagnostics

The status of the DP slave modules at PROFIBUS connections MPI/DP, DP1 - or the PROFINET connection PN - is signaled to the user program by the basic program using the "Slaves OK" group signal of the particular bus system:

- DB10.DBX92.0 == 1 (MPI/DP bus: Slaves OK)
- DB10.DBX92.1 == 1 (DP1 bus: Slaves OK)
- DB10.DBX92.2 == 1 (PN bus: Slaves OK)

The group signal is derived from the LED status of the respective bus system (system state list SZL 0x174).

Alarm output

If an error or a failure of a slave is detected at the particular bus system, the following alarm is displayed:

- Alarm 400551 (MPI/DP bus)
- Alarm 400552 (DP1 bus)
- Alarm 400553 (PN bus)

The alarm is automatically deleted again when the error is removed.

Suppression of the alarm output 400551, 400552, 400553

By setting one of the following signals, the alarm is suppressed for the particular bus system:

- DB10.DBX92.4 = 1 (suppress alarm 400551)
- DB10.DBX92.5 = 1 (suppress alarm 400552)
- DB10.DBX92.6 = 1 (suppress alarm 400553)

The alarm is immediately suppressed as soon as the signal is set. Before the fault occurs or while a fault is already active.

When the signal is reset, in the case of a fault, the corresponding alarm is displayed.

The signals are reset when the control system powers up.

Note

By setting the signal, the fault monitoring for the complete bus line is deactivated!

9.6 Response to NCK failure

General

During cyclic operation, the PLC continuously monitors NC availability by querying the sign-of-life. If the NC is no longer reacting, the NC/PLC interface is initialized, and the **NC-CPU ready** signal in the area of the **signals from NC (DB 10.DBX 104.7)** is reset. Furthermore, the signals sent from the NC to the PLC and vice versa are set to an initial state.

The PLC itself remains active so that it can continue to control machine functions. However, it remains the responsibility of the user program to set the machine to a safe state.

NC → PLC signals

The signals sent by the NC to the PLC are divided into the following groups:

- Status signals from the NC, channels, axes and spindles
- Modification signals of the auxiliary functions
- Values of the auxiliary functions
- Values of the G commands

Status signals:

The status signals from the NC, channels, axes, and spindles are reset.

Auxiliary function change signals:

Auxiliary function change signals are also reset.

Auxiliary function values:

Auxiliary function values are retained so that it is possible to trace the last functions triggered by the NC.

G command values:

G command values are reset (i.e. each initialized with the value 0).

PLC → NC signals

The signals sent by the PLC to the NC are divided into control signals and tasks that are transferred by FCs to the NC.

Control signals:

The control signals from the PLC to the NC are frozen; cyclic updating by the PLC basic program is suspended.

Jobs from PLC to NC:

The FCs and FBs, which are used to pass jobs to the NC, must no longer be processed by the PLC user program, as this could lead to incorrect checkback signals. During power-up of the control, a job (e.g. read NC data) must not be activated in the user program until the **NC-CPU ready** signal is set.

9.7 Functions of the basic program called from the user program

General

In addition to the modules of the basic program, which are called at the start of OB1, OB40 and OB100, functions are also provided which have to be called and supplied with parameters at a suitable point in the user program.

These functions can be used, for example, to pass the following jobs from the PLC to the NC:

- Traversing concurrent axes (FC18)
- Start asynchronous subprograms (ASUPs) (FC9),
- Select NC programs (FB4)

- Control spindle (FC18),
- Read and write variables (FB2, FB3)

Note

Checking and diagnostics of a function call of the basic PLC program

To simplify the checking and diagnostics of a function call (FB or FC) of the basic PLC program that is controlled via a trigger (e.g. via Req, Start parameters) and that provide an execution acknowledgment as output parameter (e.g. via Done, NDR, Error parameters), proceed as follows.

A variable compiled of other signals which produce the trigger for the function call should be set. Start conditions may be reset only as a function of the states of parameters Done, NDR and Error.

The appropriate control mechanism can be placed in front of or behind the function call. If the mechanism is placed after the call, the output variables can be defined as local variables (advantage: Reduction of global variables, markers, data variables and time-related advantages over data variables).

The trigger parameter must be a global variable (e.g. marker, data variable).

Jobs that are still active must be reset from the user program in OB100 (Req, Start, parameters,
etc. from TRUE ⇒ FALSE). A POWER OFF/ON could result in a state in which jobs are still active.

Concurrent axes

The distinguishing features of concurrent axes are as follows:

- They must be defined as such via the NC machine data.
- They can be traversed either from the PLC or from the NC by means of the JOG keys.
- Starting from the PLC is possible in the NC operating modes MDI and AUTOMATIK via FC.
- The start is independent of NC block boundaries.

Function calls are available for positioning axes, indexing axes and spindles (FC18).

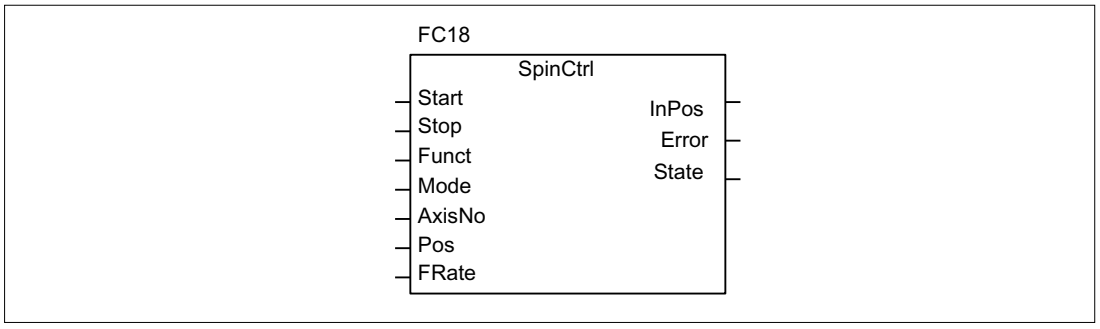
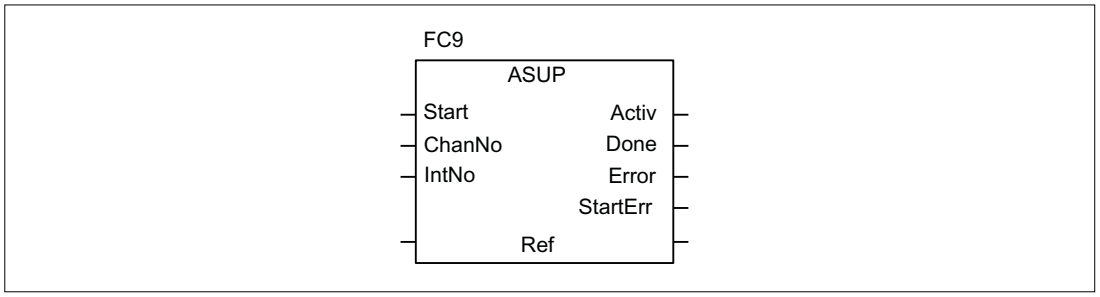


Figure 9-2 FC18 input/output parameters

Asynchronous subprograms (ASUPs)

The ASUP can be used to trigger any functions in the NC. Before an asynchronous subprogram can be started from the PLC, it must be ensured that it is available and prepared by the NC program or by FB4 PI services (ASUP).

Once prepared in this way, it can be started at any time from the PLC. The NC program running in one of the parameterized channels of FC9 is interrupted by the asynchronous subprogram. An ASUP is started by calling FC9 from the user program by setting the start parameter to 1.



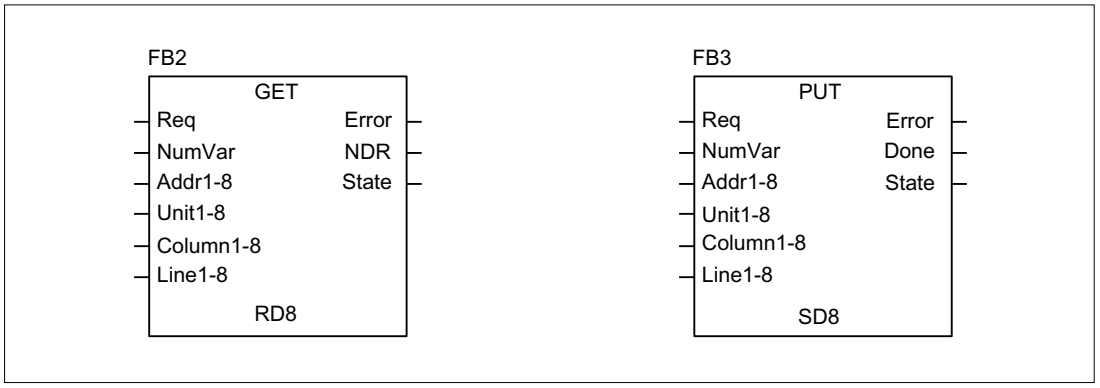
Note

If an asynchronous subprogram has not been prepared by an NC program or by FB4 (ASUP) (e.g. if no interrupt no. has been assigned), a start error is output (StartErr = TRUE).

Read/Write NC variables

NC variables can be read with FB GET while values can be entered in NC variables with FB PUT. The NC variables are addressed via identifiers at inputs Addr1 to Addr8. The identifiers (symbols) point to address data which must be stored in a global DB. To allow generation of this DB, a PC software (NC-Var-Selector) is supplied with the basic program with which the required variables can be selected from a table, which is also supplied. The selected variables are first collected in a second, project-related list. Command **Generate DB** creates a "*.AWL" file which must be linked to the program file for the machine concerned and compiled together with the machine program.

1 to 8 values can be read or written with a read or write job. If necessary, the values are converted [e.g. NC floating-point values (64-bit) are converted to PLC format (32-bit with 24-bit mantissa and 8-bit exponent) and vice versa]. A loss of accuracy results from the conversion from 64-bit to 32-bit REAL. The maximum precision of 32-bit REAL numbers is approximately 10 to the power of 7.



AG_SEND/AG_RECV functions

The AG_SEND/AG_RECV functions correspond to the functions of the library "SIMATIC_NET_CP" of the S7-300 CPU in STEP 7. Generally, the online help is valid for these functions.
The AG_SEND/AG_RECV functions can be used for data exchange with another station via the integrated "CP 840D sl". A description of the functions is provided in Section "Block descriptions (Page 149)".

Note

Other communication blocks (e.g. BSEND, USEND) which possess a CP343-1 are not supported in SINUMERIK 840D sl.

9.8 Symbolic programming of user program with interface DB

General

Note

The basic program library on the CD supplied with the Toolbox for the 840D contains files NST_UDTB.AWL and TM_UDTB.AWL.

The compiled UDT blocks from these two files are stored in the CPU program of the basic program.

A UDT is a data type defined by the user that can, for example, be assigned to a data block generated in the CPU.

Symbolic names of virtually all the interface signals are defined in these UDT blocks.

The UDT numbers 2, 10, 11, 19, 21, 31, 71, 72, 73, 77, 1002, 1071, 1072, 1073 are used.

The assignments have been made as follows:

UDT assignments		
UDT number	**Assignment to interface DB**	**Meaning**
UDT2	DB2	Alarms/messages
UDT10	DB10	NCK signals
UDT11	DB11	Mode group signals
UDT19	DB19	HMI signals
UDT21	DB21 to DB30	Channel signal
UDT31	DB31 to 61	Axis/spindle signals
UDT71	DB71	Tool management: Load/unload locations
UDT72	DB72	Tool management: Change in spindle
UDT73	DB73	Tool management: Change in revolver
UDT77	DB77	MCP and HHU signals with standard SDB 210
UDT1002	DB2	Extended alarms / messages (FB1 parameter "ExtendAlMsg:=TRUE")
UDT 1071	DB 1071	Tool management: Loading/unloading points (multitool)
UDT 1072	DB 1072	Tool management: Change in spindle (multitool)
UDT 1073	DB 1073	Tool management: Change in turret (multitool)

To symbolically program the interface signals, the interface data blocks must first be symbolically assigned using the symbol editor.

For example, symbol "AxisX" is assigned to operand DB31 with data type UDT31 in the symbol file.

After this input, the STEP 7 program can be programmed in symbols for this interface.

Note

Programs generated with an earlier software version that utilize the interface DBs described above can also be converted into symbol programs. A fully qualified command for data access e.g. "U DB31.DBX60.0" (spindle / rotary axis) is necessary in the program previously created. This command is converted upon activation of the symbolics in the editor "AxisX.E_SpKA".

Description

Abbreviated symbolic names of the interface signals are defined in the two STL files NST_UDTB.AWL and TM_UDTB.AWL.

In order to create the reference to the names of the interface signals, the name is included in the comment after each signal.

The names are based on the English language. The comments are in English.

The symbolic names, commands and absolute addresses can be viewed by means of a STEP 7 editor command when the UDT block is opened.

Note

Unused bits and bytes are listed, for example, with the designation "f56_3".

- "56": Byte address of the relevant data block
- "3": Bit number in this byte

9.9 M decoding acc. to list

Function description

Up to 256 M functions with extended address can be decoded from the basic program using the "M decoding according to list" function. The function is activated using FB1 parameter "ListMDecGrp" (number of M groups for decoding). The assignment of the M function with extended address and a signal in the signal list is defined in the decoding list. The signals are also grouped for this purpose.

Decoding list (DB75)

The source file for the decoding list (MDECLIST.AWL) is supplied with the basic program. Data block DB75 is created when the STL source is compiled. Before the function is activated, the decoding list (DB75) must be transferred to the PLC followed by a restart.

An M function is decoded if it is in the decoding list. When decoding the M function, the corresponding signal is set in the signal list as a function of the specific group. When setting a signal in the signal list, the interface signal "Read in inhibit" is set by the basic program in the associated channel of the NC. The interface signal is reset again for the channel as soon as the user resets all of the signals output from this channel in the signal list; i.e. after they have been acknowledged.

Signal list (DB76)

When activating the function in data block DB76, the basic program creates the signal list. From then, for each M signal decoded according to the list, a signal is set in the signal list (DB76) in the corresponding group. At the same time, the "Readin inhibit" interface signal is set in the channel in which the M function has been output. The interface signal is reset again for the channel as soon as the user resets all of the signals output from this channel in the signal list; i.e. after they have been acknowledged.

Highspeed auxiliary functions

When an M function contained in the decoding list is output as "fast help function", no read-in inhibit is set for the corresponding channel of the NC.

The figure below shows the structure of the **M decoding according to list**:

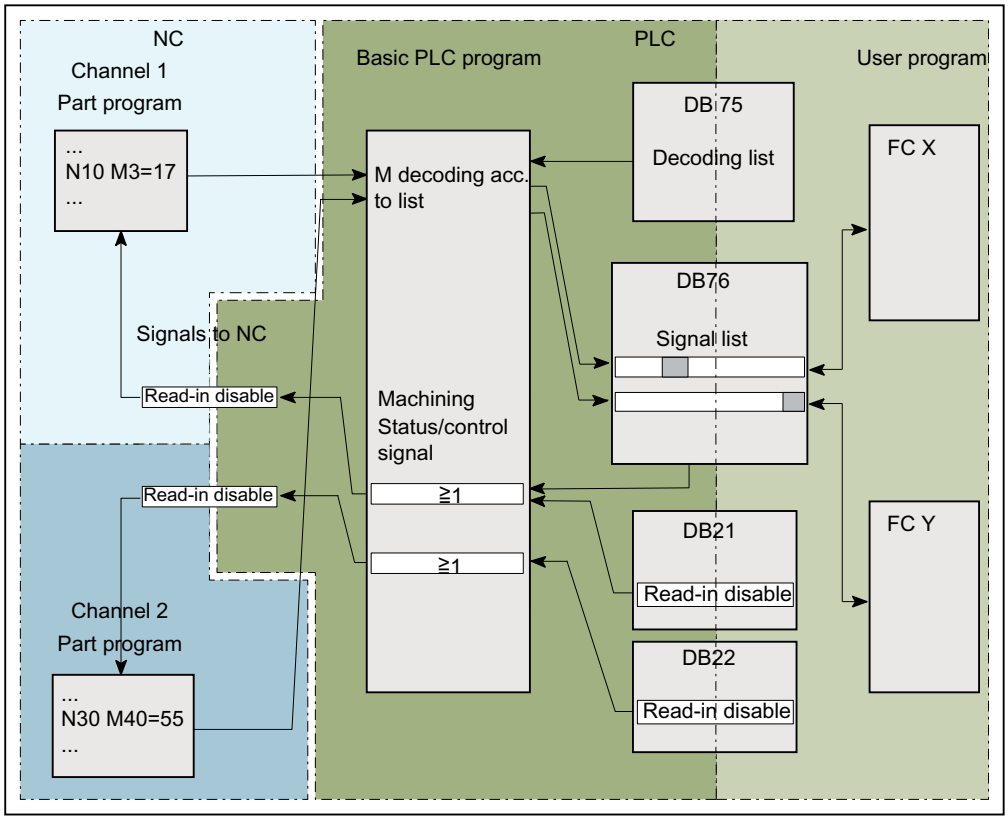


Figure 9-3 M decoding acc. to list

Activation

M decoding is activated using FB1 parameter "**ListMDecGrp**"

The number of M groups to the evaluated and/or decoded is specified using the appropriate parameter. The function is active for a parameter value = 1 ... 16.

- Basic program, OB100, FB1 parameter **ListMDecGrp** = <number of M groups> (also see " FB1: RUN_UP - basic program, start section (Page 149) ").

Properties and structure of the decoding list (DB75)

Properties of the decoding list (DB75):

- There is only one decoding list independent of the channel.
- The decoding list can include a maximum of 16 groups.
- A group has a maximum of 16 signals

- There must be an entry in the decoding list for every group of M functions to be decoded.
- The assignment between the M function with extended address and the signal to be set in the signal list is specified in the decoding list using the first and last M function of the associated group.
  - First M function: Parameter: "MFirstAdr" ≙ signal or bit 0
  - Last M function: Parameter: "MLastAdr" ≙ dependent on the difference to the first M function maximum signal or bit 15

Structure of the decoding list (DB75):

An entry in the decoding lists consists of 3 parameters, each of which is assigned to a group.

Group	Extended M address	First M address of the group	Last M address of the group
1	MSigGrp[1].MExtAdr	MSigGrp[1].MFirstAdr	MSigGrp[1].MLastAdr
2	MSigGrp[2].MExtAdr	MSigGrp[2].MFirstAdr	MSigGrp[2].MLastAdr
...	...	...	...
16	MSigGrp[16].MExtAdr	MSigGrp[16].MFirstAdr	MSigGrp[16].MLastAdr

Type and value range of the signals:

Signal	Type	Value range	Meaning
MExtAdr	INT	0 ... 99	Extended M address
MFirstAdr	DINT	0 to 99.999.999	First M address in group
MLastAdr	DINT	0 to 99.999.999	Last M address in group

Properties of the signal list (DB76)

The signal list (DB76) has the following properties:

- There is only one signal list independent of the channel.
- The signal list can include a maximum of 16 signals for each M group.

Example

3 groups of M functions are to be decoded:

- Group 1: M2 = 1 to M2 = 5
- Group 2: M3 = 12 to M3 = 23
- Group 3: M40 = 55

Structure of the decoding and signal list

Group	Decoding list (DB75)			Signal list (DB76)
	Extended M address	First M address of the group	Last M address of the group	
1	2	1	5	DB76.DBX0.0 ... DBX0.4
2	3	12	23	DB76.DBX2.0 ... DBX3.3
3	40	55	55	DB76.DBX4.0

Program code

```
DATA_BLOCK DB 75
TITLE =
VERSION : 0.0
   STRUCT
      MSigGrp : ARRAY [1 .. 16 ] OF STRUCT
         MExtAdr : INT;
         MFirstAdr : DINT;
         MLastAdr : DINT;
      END_STRUCT ;
BEGIN
   MSigGrp[1].MExtAdr   :=    2; extended M address of the 1st group
   MSigGrp[1].MFirstAdr :=  L#1; first M address of the group
   MSigGrp[1].MLastAdr  :=  L#5; last M address of the group
   MSigGrp[2].MExtAdr   :=    3; extended M address of the 2nd group
   MSigGrp[2].MFirstAdr := L#12; first M address of the group
   MSigGrp[2].MLastAdr  := L#23; last M address of the group
   MSigGrp[3].MExtAdr   :=   40; extended M address of the 3rd group
   MSigGrp[3].MFirstAdr := L#55; first M address of the group
   MSigGrp[3].MLastAdr  := L#55; last M address of the group
END_DATA_BLOCK
```

Structure of FB1 in OB100

To activate the function, insert the parameter for the number of M groups to be decoded "ListMDecGrp".

```
Call FB 1, DB 7(
   ...
   ListMDecGrp := 3;     //M decoding of three groups
   ...
);
```

Description

A restart must be performed after the entry has been made in OB100 and the decoding list (DB75) has been transferred to the PLC. The basic program creates the signal list (DB76) when it restarts.

An NC program is then started, for instance in the 1st channel. An extended M function is included in this (M3=17). When decoding the M function (M3 ≙ group 2), the associated signal (DBW1.5) is set in the signal list (DB76) and the interface signal "Read-inhibit" in the 1st channel. The execution of the NC program is stopped. Further, the "Extended address M function" and the "M function number" are displayed in the channel DB of the 1st channel.

The "Read-in inhibit" signal in the 1st channel is reset once the user has reset all of the signals output from this channel in the signal list (DB76), and has therefore acknowledged them.

9.10 PLC machine data

General information

The user has the option of storing PLC-specific machine data in the NC. These machine data can then be processed during power-up of the PLC (OB100). This enables, for example, user options, machine expansion levels, machine configurations, etc., to be implemented.

The interface to read this data is in DB20. However, DB20 is only created by the basic program during power-up when user machine data is used, i.e. sum of GP parameters "UDInt", "UDHex" and "UDReal" is greater than ZERO.

Size of the data areas

The sizes of the individual areas, and thus the total length of the DB20, is set by the following PLC machine data:

- MD14504 $MN_MAXNUM_USER_DATA_INT
- MD14506 $MN_MAXNUM_USER_DATA_HEX
- MD14508 $MN_MAXNUM_USER_DATA_FLOAT

User-relevant basic PLC program parameters

The machine data settings are provided to the user via the following basic PLC program parameters:

- "UDInt"
- "UDHex"
- "UDReal"

Note

User machine data as ARRAY

The user machine data is always handled internally as an ARRAY. ARRAY variables occupy the memory in the PLC up to the next word limit, i.e. at a BYTE with an even address.

A filler byte is therefore added internally when there is an odd number in MD14506 $MN_MAXNUM_USER_DATA_HEX. This filler byte is included in the FB1 output parameter "UDHex" (number of hexadecimal machine data).

Example:

MD14506 $MN_MAXNUM_USER_DATA_HEX = 3 ⇒ UDHex = 4, although only three hex values can be read.

Data storage

The data is seamlessly stored in the DB20 by the basic PLC program in the following sequence:

1. INT values
2. HEX values (bit arrays)
3. FLOAT values

INT and FLOAT values are saved in S7 format.

The hexadecimal values are stored in DB20 in the order in which they are input (use as bit arrays).

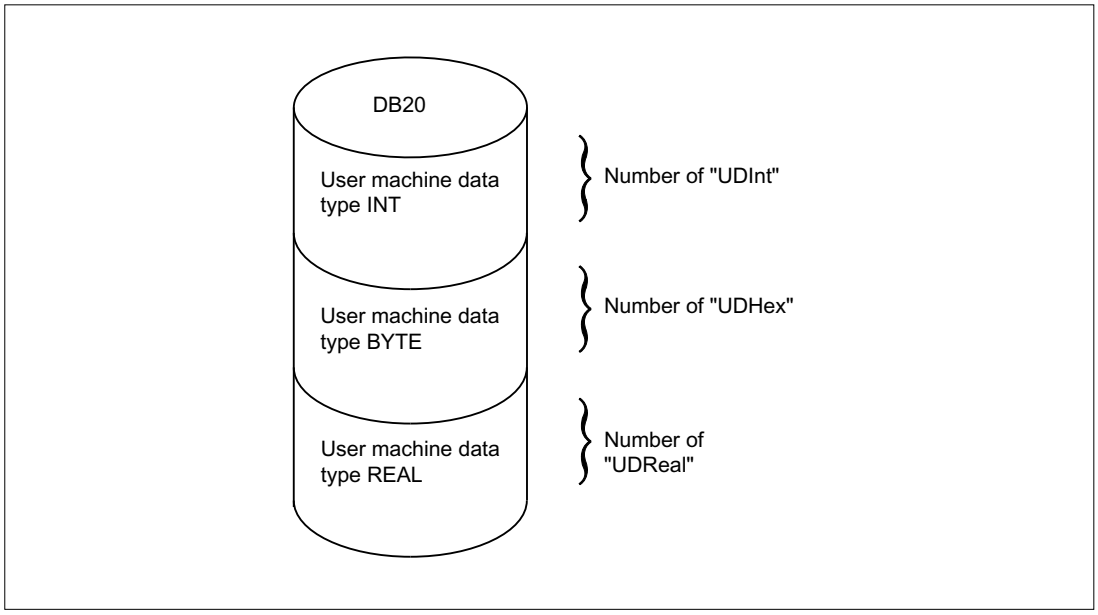


Figure 9-4 User data in DB20

Note

If the number of PLC machine data used is increased later, then DB20 must be deleted beforehand. To prevent such extensions in use having any effect on the existing user program, the data in DB20 should be accessed in symbolic form wherever possible, e.g. by means of a structure definition in the UDT.

Example

For the project in the example, four INT values, two HEX values for bit information, and one FLOAT value are needed.

Machine data:

```
MD14510 $MN_USER_DATA_INT[0] =    123
MD14510 $MN_USER_DATA_INT[1] =    456
MD14510 $MN_USER_DATA_INT[2] =    789
MD14510 $MN_USER_DATA_INT[3] =    1011
...
MD14512 $MN_USER_DATA_HEX[0] =    12
MD14512 $MN_USER_DATA_HEX[1] =    AC
...
MD14514 $MN_USER_DA-              123.456
TA_FLOAT[0] =
```

Basic PLC program parameters (OB100):

```
CALL FB1, DB7(
           MCPNum :=           1,
           MCP1In :=           P#E0.0,
           MCP1Out :=          P#A0.0,
           MCP1StatSend :=     P#A8.0,
           MCP1StatRec :=      P#A12.0,
           MCP1BusAdr :=       6,
           MCP1Timeout :=      S5T#700MS,
           MCP1Cycl :=         S5T#200MS,
           NCCyclTimeout :=    S5T#200MS,
           NCRunupTimeout :=   S5T#50S;
```

Basic PLC program parameters (scan at runtime):

```
          l gp_par.UDInt;      //=4,
          l gp_par.UDHex;      //=2,
          l gp_par.UDReal;     //=1 )
```

During PLC power-up, DB20 was generated with a length of 28 bytes:

DB20	
Address	**Data**
0.0	123
2.0	456
4.0	789
6.0	1011
8.0	b#16#12
9.0	b#16#AC
10.0	1.234560e+02

The structure of the machine data used is specified in a UDT:

```
TYPE UDT20
   STRUCT
      UDInt :   ARRAY [0 .. 3] OF INT;
      UDHex0 :  ARRAY [0 .. 15]OF BOOL;
      UDReal :  ARRAY [0 .. 0] OF REAL;     //Description as field, for
                                            later expansions
   END_STRUCT;
END_TYPE
```

Note

ARRAY OF BOOL are always sent to even-numbered addresses. For this reason, an array range of 0 to 15 must generally be selected in the UDT definition or all Boolean variables specified individually.

Although only a REAL value is used initially in the example, a field (with one element) has been created for the variable. This ensures that extensions can be made easily in the future without the symbolic address being modified.

Symbolic accesses

An entry is made in the symbol table to allow data access in symbolic form:

Symbol	Operand	Data type
UData	DB20	UDT20

Access operations in user program (list includes only symbolic read access):

```
...
         L      "UData".UDInt[0];
         L      "UData".UDInt[1];
         L      "UData".UDInt[2];
         L      "UData".UDInt[3];

         U      "UData".UDHex0[0];
         U      "UData".UDHex0[1];
         U      "UData".UDHex0[2];
         U      "UData".UDHex0[3];
         U      "UData".UDHex0[4];
         U      "UData".UDHex0[5];
         U      "UData".UDHex0[6];
         U      "UData".UDHex0[7];
         ...    ...
         U      "UData".UDHex0[15];

         L      "UData".UDReal[0];
...
```

9.11 Configuration machine control panel, handheld unit, direct keys

General

Up to two machine control panels and one handheld unit can be in operation at the same time. There are various connection options (Ethernet/PROFINET, PROFIBUS) for the machine control panel (MCP) and handheld unit (HHU). It is possible to connect two MCPs to different bus systems (mixed operation is only possible on Ethernet and PROFIBUS). This can be achieved using FB1 parameter "MCPBusType". In this parameter, the right-hand decade (units position) is responsible for the first MCP and the left-hand decade (tens position) for the second MCP.

Parameterization of components is always performed by calling the basic program block FB1 in OB100. FB1 saves its parameters in the associated instance data block (DB7, symbolic "gp_par"). Separate parameter sets are provided for each machine control panel and the handheld unit. The input/output addresses of the user must be defined in these parameter sets. These input and output addresses are also used in FC19, FC24, FC25, FC26 and FC13. Further, the addresses for status information, PROFIBUS or Ethernet/PROFINET are also to be defined. The default time settings for timeout and cyclic forced retriggering should not be changed. Please refer to the Operator Components manual for further information on MCP and HHU components.

Activation

Each component is activated either via the number of machine control panels ("MCPNum" parameter) or, in the case of the handheld unit, via the "BHG" parameter. The MCP and HHU connection settings are entered in FB1 parameters "MCPMPI", "MCPBusType" or "BHG", "BHGMPI".

Handheld unit (HT 2)

In the handheld unit the addressing is done via a parameter of the GD parameter set. This was necessary for reasons of compatibility of the parameter names.

Configuration

Essentially, there are various communication mechanisms for transferring data between the MCP/HHU and PLC. These mechanisms are characterized by the bus connection of the MCP and HHU. In one case (Ethernet), data is transported via the "CP 840D sl".

Parameterization is completely realized via the MCP/HHU parameters in FB1.

In the other case the transmission is via the PLC operating system through the PROFIBUS configuration.

The parameterization is performed via STEP 7 in HW-Config. To enable the basic program to access this data and failure monitoring of MCP/HHU, the addresses set in the FB1 parameters must be made known to the basic program.

An overview of the various coupling mechanisms is shown below. Mixed operation can also be configured.

If an error is detected due to a timeout, an entry is made in the alarm buffer of the PLC CPU (alarms 400260 to 400262). In this case, the input signals from the MCP or from the handheld unit (MCP1In/MCP2In or BHGIn) are reset to 0. If it is possible to resynchronize the PLC and MCP/HHU, communication is resumed automatically and the error message deleted by the GP.

Note

The abbreviation "(n.r.)" in the tables below means "not relevant".

Ethernet connection (MCPBusType = 5)

Without further configuration settings being made, communication takes place directly from the PLC GP via the CP 840D sl. The FB1 parameters listed below are used for parameterization.

The numeric part of the logical name of the component must be entered in "MCP1 BusAdr", "MCP2 BusAdr" or "BHGRecGDNo" (corresponds to the bus address of the node). The logical name is defined via switches on the MCP or terminal box.

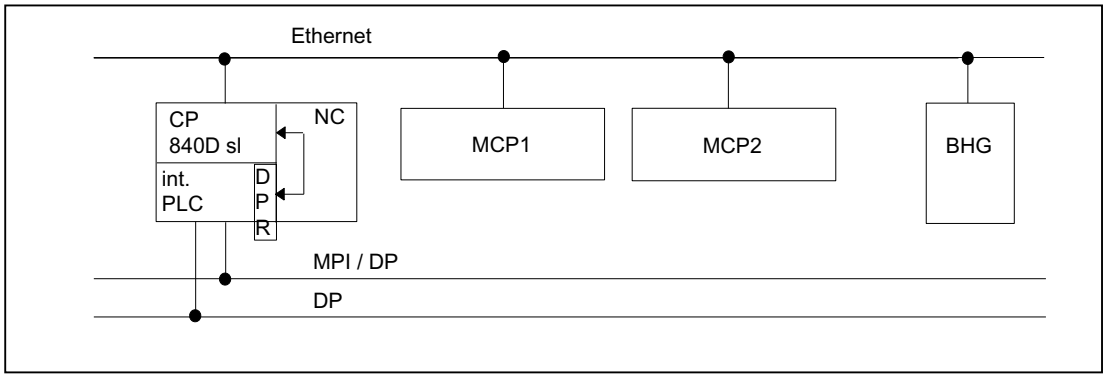


Figure 9-5 Ethernet connection

Relevant parameters (FB1)		
MCP		**HHU**
MCPNum=1 or 2 (number of MCPs)		HHU = 5 (via CP 840D sl)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend	MCP2StatSend	BHGStatSend
MCP1StatRec (n.r.)	MCP2StatRec (n.r.)	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen (n.r.)
MCP1Timeout (n.r.)	MCP2Timeout (n.r.)	BHGOutLen (n.r.)
MCP1Cycl (n.r.)	MCP2Cycl (n.r.)	BHGTimeout (n.r.)
MCPMPI = FALSE		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCP1NotSend	MCP2NotSend	BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPBusType = b#16#55 (via CP 840D sl)		BHGSendGDNo (n.r.)
		BHGSendGBZNo (n.r.)

Relevant parameters (FB1)	
MCP	**HHU**
MCPSDB210 = FALSE	BHGSendObjNo (n.r.)
MCPCopyDB77 = FALSE	**BHGMPI = FALSE**
	BHGStop
	BHG NotSend

An error entry is also made in the PLC alarm buffer for timeouts. As a result, the following error messages are output at the HMI:

- 400260: MCP 1 failure
  or
- 400261: MSST 2 failed
- 400262: HHU failure

An MCP or HHU failure is detected immediately after a cold restart even if no data has yet been exchanged between the MCP/HHU and PLC.
The monitoring function is activated as soon as all components have signaled "Ready" after power-up.

OP with direct keys

The direct keys of the OPs at the Ethernet bus should be transferred to the PLC. Previously, the direct keys have been transferred to the PLC via the PROFIBUS or via a special cable connection between OP and MCP.

For connecting the direct keys via the Ethernet, this concerns e.g. the "OP 08T", there is a parameterization in the basic program for activating the data transport. The associated parameters lie in the instance DB of the FB 1 (OpKeyNum to OpKeyBusType, see data table). The parameters are provided by the user in the start OB100 by connecting the parameter at the FB1 call. The bus address and Op1/2KeyStop can also be modified in the cyclic operation by writing the FB1 instance DB DB 7.

The transport of the user data of the direct keys runs in the same way as in the case of Ethernet MCP. The data transport can also be stopped and restarted by writing the DB7-parameter "Op1/2KeyStop". During the Stop phase the address of the direct key module (TCU-index or the MCP-address) can also be changed.

After resetting the Stop signal, a connection to the new address is established.

The status of the respective direct-key interfaces can be read in the interface signal:

DB10.DBX104.3 (OP1Key ready)

or

DB10.DBX104.4 (OP2Key ready)

Address direct keys

For the parameter Op1/2KeyBusAdr, the TCU index is normally to be used. This affects the OPs, such as OP 08T, OP 12T, which for the direct keys **do not** have special cable connection to an Ethernet MCP.

If OPs with direct keys have a special cable connection and these are connected to an Ethernet-MCP, then for the parameter Op1/2KeyBusAdr the address of the MCP (DIP-switch setting of

the MCP) is to be used. Only the data stream of the direct keys (2 bytes) is transferred via the direct key interface.

Warning of connection termination

To ensure a quick response to an impending termination of the connection between PLC and direct keys, signals are set shortly after the cyclic communicating between PLC and direct keys has failed. If there is still no cyclic communication between PLC and direct keys, the connection is terminated.

The signals for warning against termination of a connection are set in DB10.

- DB10.DBX94.3 (alarm direct keys 1 communication error)
- DB10.DBX94.4 (alarm direct keys 2 communication error)

The signals are reset after a connection is terminated or when there is a new connection.

Alarm direct keys

An error entry is also made in the PLC alarm buffer for timeouts. As a result, the following error messages are output at the HMI:

- 400274: Direct key 1 failed
  or
- 400275: Direct key 2 failed

Control unit switching for direct keys

The user connects Op1/2KeyBusAdr with 0xFF and stop = TRUE in startup block OB100. The direct key address of the M-to-N interface is connected to parameter "Op1KeyBusAdr" via the M-to-N block FB9.

Relevant parameters (FB1)		
Direct keys		**e.g. direct keys OP 08T**
OpKeyNum = 1 or 2 (number of OPs with direct keys)		
Op1KeyIn	Op2KeyIn	
Op1KeyOut	Op2KeyOut	
OpKey1BusAdr	Op2KeyBusAdr	Address: TCU index
Op1KeyStop	Op2KeyStop	
Op1KeyNotSend	Op2KeyNotSend	
OpKeyBusType = b#16#55 (via CP 840D sl)		

MCP identification

Via the identify interface in DB7 it is possible to query the type of the Ethernet component (MCP, HT 2, HT 8 or direct keys) with the relevant parameters at the input/output in cyclic operation:

- Relevant parameters at the input:
  "IdentMcpBusAdr", "IdentMcpProfilNo", "IdentMcpBusType", "IdentMcpStrobe"
- Relevant parameters at the output:
  "IdentMcpType", "IdentMcpLengthIn", "IdentMcpLengthOut"

Here the DIP device address or the TCU index at the parameter "IdentMcpBusAdr" is activated by the user program together with setting of the Strobe signal.

The input parameter "IdentMcpProfilNo" is normally to be set to the value 0. This parameter is to be set to the value 1 only in the identification of the direct keys. The parameter "IdentMcpBusType" currently has no significance for a user program and is to be left in its default value.

After resetting the Strobe signal by the basic program, valid output information becomes available to the user. The resetting of the Strobe signals by the basic program can last for several PLC cycles (up to two seconds).

The output parameters should show the user the size of the data areas for the addressed device. Furthermore, it can be defined here whether an HT 2 or an HT 8 or no device is connected to the terminal box. With this information, the MCP channel or the HHU channel can be activated. In cyclic operation the parameters can be written symbolically by the user program and read via the symbol names of the DB7 (gp_par).

Relevant parameters (FB1)		
MCP device identification		**Input parameters, e.g. OP 08T**
Input	**Output**	**Values for direct keys**
IdentMcpBusAdr	IdentMcpType	IdentMcpBusAdr = TCU index
IdentMcpBusProfilNo	IdentMcpLengthIn	IdentMcpBusProfilNo = Value 1
IdentMcpBusType	IdentMcpLengthOut	IdentMcpBusType = Default value
IdentMcpStrobe		
IdentMcpBusProfilNo		**Value**
MCP, HHU, HT 8, HT 2		B#16#0
Direct keys such as e.g. OP 08T, OP 12T		B#16#1
IdentMcpType (Mcp-Type)		
No device connected		0
MCP 483C IE (Compact)		B#16#80
MCP 483C IE		B#16#81
MCP 310		B#16#82
MCP OEM		B#16#83
MCP DMG		B#16#84
HT 8		B#16#85
TCU_DT (direct keys)		B#16#86
MCP_MPP		B#16#87
HT 2		B#16#88
OP 08T (direct keys)		B#16#89

PROFIBUS connection on the DP port (MCPBusType = 3)

In case of PROFIBUS connection of the MCP, this component must be considered in the hardware configuration setting of STEP 7. The MCP is connected to the standard DP bus of the PLC (**not to MPI/DP**). The addresses must be stored in the input and output mapping area. These start addresses must also be stored in the pointer parameters of the FB1. The FB1 parameters listed below are used for further parameterization.

There is no PROFIBUS variant of the HHU. For this reason, an Ethernet connection is shown for the HHU in this figure. The PROFIBUS slave address must be stored in the parameters "MCP1BusAdr" and "MCP2BusAdr". Enter the pointer to the configured diagnostic address (e.g. P#A8190.0) in "MCPxStatRec".

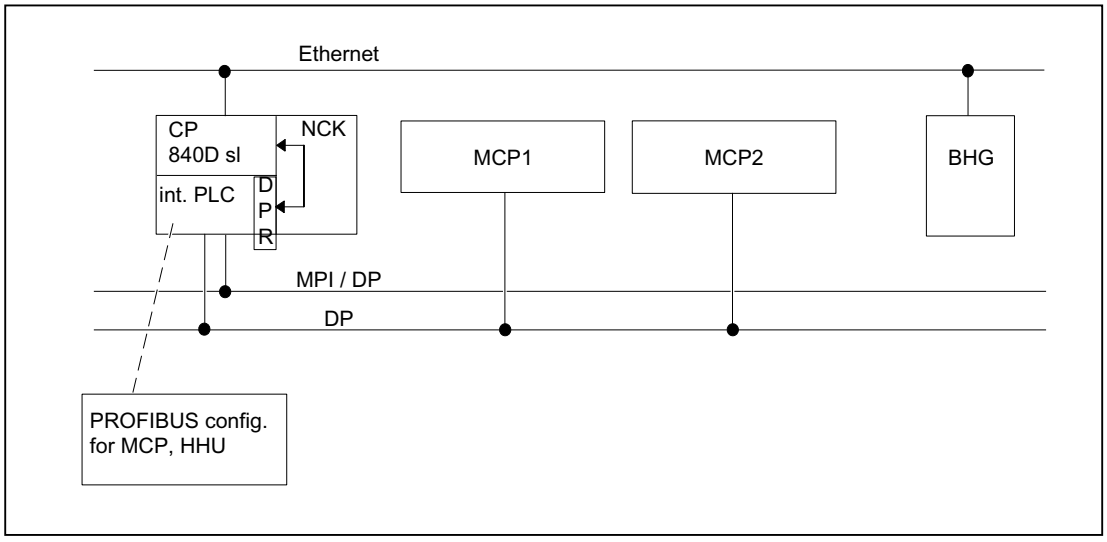


Figure 9-6 PROFIBUS connection

Relevant parameters (FB1)		
MCP		**HHU**
MCPNum = 1 or 2 (number of MCPs)		HHU = 5 (via CP 840D sl)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl (n.r.)	MCP2Cycl (n.r.)	BHGTimeout (n.r.)
MCPMPI = FALSE		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCPBusType = b#16#33		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPSDB210 = FALSE		BHGSendGDNo (n.r.)
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)
		BHGSendObjNo (n.r.)

Relevant parameters (FB1)		
MCP		**HHU**
		BHGMPI = FALSE
		BHGStop

MCP failure normally switches the PLC to the STOP state. If this is undesirable, OB 82, OB 86 can be used to avoid a stop. The basic program has, as standard, the OB82 and OB86 call. FC5 is called in these OBs. This FC5 checks whether the failed slave is an MCP. If this is the case, no PLC stop is triggered. Setting "MCPxStop" := TRUE causes the basic program to deactivate the MCP as a slave via SFC12. If the PLC does not switch to the stop state following the failure or fault of the MCP, an alarm message will be generated via the basic program. The interrupt is deleted when the station recovers.

PROFIBUS connection on the MPI/DP port (MCPBusType = 4)

With the PROFIBUS connection of the MCP, this component must be considered in the STEP 7 hardware configuration. The MCP is connected on the MPI/DP bus of the PLC.

The addresses must be stored in the input and output mapping area. These start addresses must also be stored in the pointer parameters of the FB1. The FB1 parameters listed below are used for further parameterization. There is no PROFIBUS variant of the HHU. For this reason, an Ethernet connection is shown for the HHU in this diagram. The PROFIBUS slave address must be stored in the parameters MCP1BusAdr and MCP2BusAdr. Enter the pointer to the configured diagnostic address (e.g. P#A8190.0) in MCPxStatRec.

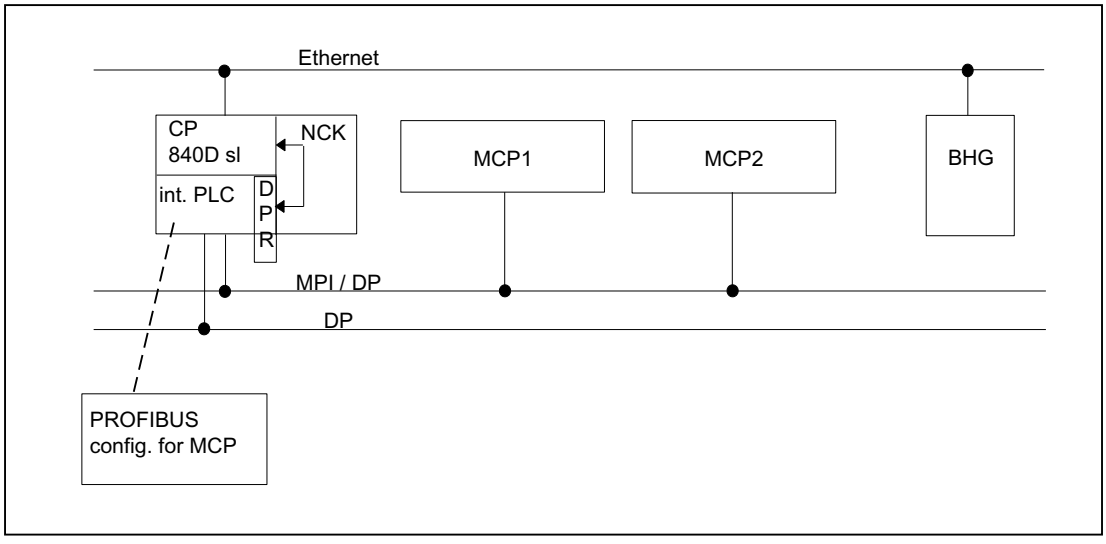


Figure 9-7 PROFIBUS connection on the MPI/DP port

Relevant parameters (FB1)		
MCP		**HHU**
MCPNum = 1 or 2 (number of MCPs)		HHU = 5 (via CP 840D sl)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec

Relevant parameters (FB1)		
MCP		**HHU**
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl (n.r.)	MCP2Cycl (n.r.)	BHGTimeout (n.r.)
MCPMPI = FALSE		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCPBusType = b#16#44		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPSDB210 = FALSE		BHGSendGDNo (n.r.)
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)
		BHGSendObjNo (n.r.)
		BHGMPI = FALSE
		BHGStop

MCP failure normally switches the PLC to the STOP state. If this is undesirable, then OB82 and OB86 can be used to avoid a PLC stop. The basic program has, as standard, the OB82 and OB86 call. FC5 is called in these OBs. This FC5 checks whether the failed slave is an MCP. If this is the case, no PLC stop is triggered. Setting MCPxStop:= TRUE causes the basic program to deactivate the MCP as a slave via SFC12. If the PLC does not switch to the stop state following the failure or fault of the MCP, an alarm message will be generated via the basic program. The alarm is deleted when the station returns.

PROFINET connection (MCPBusType = 6)

In case of PROFINET connection of the MCP, this component must be parameterized in the hardware configuration setting of STEP 7. The MCP is coupled with the PROFINET module of the CPU.

When parameterizing the MCP in HW Config, the addresses should be placed in the input and output mapping area. These start addresses must also be stored in the pointer parameters (MCPxIn and MCPxOut) of FB1. This is because signals are transferred between the MCP and basic program via these parameters. The MCP is also monitored using parameter MCPxIn. This is the reason why parameter MCPxBusAdr is not relevant for this MCP variant.

Enter the pointer to the configured diagnostic address (e.g. P#A8190.0) in MCPxStatRec.

The PROFINET MCP has its own type which should be applied for parameter MCPBusType.

The FB1 parameters listed below are used for further parameterization. There is no PROFIBUS variant of the HHU. An Ethernet port for the HHU is shown in the diagram.

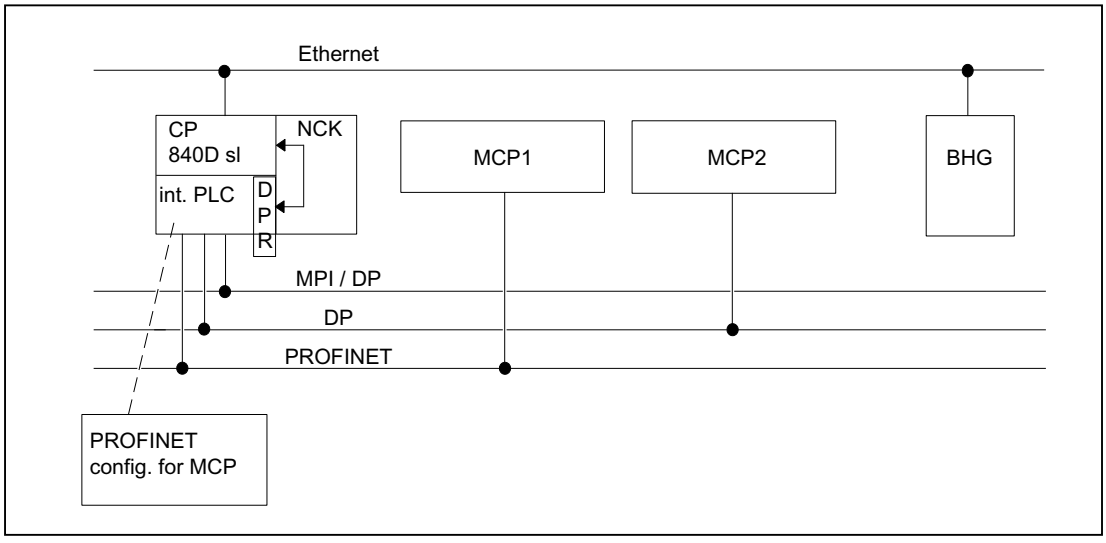


Figure 9-8 PROFINET connection

Relevant parameters (FB1)		
MCP		**HHU**
MCPNum = 1 or 2 (number of MCPs)		HHU = 5 (via CP 840D sl)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr (n.r.)	MCP2BusAdr (n.r.)	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl	MCP2Cycl	BHGTimeout (n.r.)
MCPMPI = FALSE		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCPBusType = b#16#36 (as in the figure as example) (6 = PROFINET for MCP1) (3 = PROFIBUS for MCP2)		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPSDB210 = FALSE		BHGSendGDNo (n.r.)
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)
		BHGSendObjNo (n.r.)
		BHGMPI = FALSE
		BHGStop

MCP failure normally switches the PLC to the STOP state. If this is undesirable, then OB82 and OB86 can be used to avoid a PLC stop. The basic program has, as standard, the OB82 and OB86 call. FC5 is called in these OBs. This FC5 checks whether the failed slave is an MCP. If this is the case, no PLC stop is triggered. The input address at parameter MCPxIn is of significance when monitoring for MCPxIn failure.

Setting MCPxStop := True causes the basic program to deactivate the MCP as a slave via SFC12. If the PLC does not switch to the stop state following the failure or fault of the MCP, an alarm message will be generated via the basic program. The alarm is deleted when the station returns.

9.12 Switchover of machine control panel, handheld unit

Only Ethernet variants support switchover/deactivation of an operator component (MCP or HHU) as standard.

PROFIBUS variant

With PROFIBUS variants, this functionality is only possible to a limited extent and with additional user effort.

With the PROFIBUS variant of the MCP, the data area of the DB77 for specified MCP1, MCP2 or HHU can be used for the MCP pointer on FB1. The MCP slave bus address must be set correctly under `MCPxBusAdr` as this is used as the basis for monitoring. A user program copy routine must copy the signals of the active MCP from the I/O area configured in HW Config to DB77. This enables a number of MCPs on the PROFIBUS to be switched via signals. Set the `MCPxStop` parameter to TRUE for the switchover phase from one MCP to another.

Control signals

Parameters `MCP1Stop`, `MCP2Stop` and `BHGStop` can be used to stop communication with individual components (parameter setting = 1). This stop or activation of communication can be applied in the current cycle. However, the change in value must be implemented through the symbolic notation of the parameters and not by means of another FB1 call.

Example: Stopping the transfer from the 1st machine control panel:

```
SET;
S gp_par.MCP1Stop;
```

Setting parameters `MCP1Stop`, `MCP2Stop`, `BHGStop` also results in a suppression or deletion of alarms 400260 to 400262.

Switchover of the bus address

If an existing communication connection to an operator component (MCP or HHU) is to be cancelled and a new communication connection established to a different component (MCP or HHU) with a different communication address, proceed as follows:

1. Stop the communication of the operator component to be disconnected:
   Parameter `MCP1Stop`, `MCP2Stop` or `BHGStop` = 1
2. The communication is stopped when the following applies: DB10, DBX104.0, .1 or .2 == 0

3. Change the bus address:
   - MCP: FB1 parameter `MCP1BusAdr` or `MCP2BusAdr` = <bus address of the new operator component>
   - HHU (Ethernet variant): FB1 parameter `BHGRecGDNo` = <bus address of the new operator component>
4. Enable the communication (possible in the same PLC cycle as point 3):
   Parameter `MCP1Stop`, `MCP2Stop` or `BHGStop` = 0
5. The communication with the new component is active when the following applies: : DB10, DBX104.0, .1 or .2 == 1

Switching off the LED flashing of an Ethernet MCP

After a Power On, MCPs generally indicate the completion of the power-up and waiting for a connection to be established by flashing LEDs. The flashing of the LEDs can be switched off as described in the following. Presently, this behavior **cannot** be retentively stored on the MCP.

Requirement

MCP firmware as of V02.02.04

Setting for switching off the flashing

The Send status must be set in `MCPxStop` before the start of communication with the MCP. Before the start of communication means either during power-up (OB100) or during cyclic operation (OB1) before the setting of DB7 parameter `MCPxStop` = FALSE

Setting the Send status: FB1 parameter `MCPxStatSend`, bit 30 = 0 and bit 31 = 1

There is no feedback of the current status.

Example

Extract from OB100: (based on the example for MCP1)

```
CALL "RUN_UP" , "gp_par"
   ...
MCP1StatSend := P#E 8.0
   ...
//deactivate MCP flashing
SET
R   A 11.6
S     A     11.7
   ...
```

SPL for Safety Integrated

10

Rather than being a function of the basic program, SPL is a user function. The basic program makes a data block (DB18) available for Safety SPL signals and runs a data comparison to ensure the consistency of SPL program data in the NC.

Further information
Function Manual Safety Integrated

Assignment overview

11

11.1 Block address range for PLC user program

From the following block number assignment overview, you can find out whether a specific block number (or block address) is assigned or reserved by the PLC basic program, or whether it can be used for the PLC user program.

Table 11-1 Assignment overview of the DB numbers

DB number	Availability	Occupied by
1	Reserved	Siemens
2 … 3	Occupied	Siemens (PLC basic program)
4 … 7	Reserved	Siemens
8	Occupied	Siemens (PLC basic program)
9	Reserved	Siemens
10 … 11	Occupied	Siemens (PLC basic program)
12	Occupied	Computer link and transport system interface
13 … 15	Reserved	Siemens
16	Occupied	Siemens (PLC basic program)
17 … 18	Reserved	Siemens (PLC basic program)
19	Occupied	Siemens (PLC basic program)
20	Reserved	Siemens
21 … 30	Occupied[1]	Siemens (PLC basic program: interface for NC channels)
31 … 61	Occupied[1]	Siemens (PLC basic program: interface for axes/spindles)
62 … 70	Freely available (user program)	-
71...73	Occupied[1]	Siemens (PLC basic program: tool management)
74	Reserved	Siemens
75 … 76	Occupied	Siemens (M group decoding)
77 … 80	Reserved	Siemens
81 … 999	Occupied or reserved	Siemens (ShopMill, ManualTurn)
1000 … 1019	Reserved	Siemens
1020	Occupied or reserved	Siemens (PLC basic program)
1021 … 1099	Reserved	Siemens
1100 … 24999	Freely available (user program)	-
25000 … 29999	Reserved	Siemens
30000 … 59999	Reserved[2]	Standard number range for generated F-system blocks
60000 … 60999	Occupied or reserved	CPU DBs

[1]Not recommended for use, but possible with limitations. Data blocks of channels, axes/ spindles and tool management functions that have not been activated can be used by the user, but this can result in conflicts when upgrading or migrating the PLC basic program.

[2]If you use SINUMERIK Safety Integrated plus (F-PLC), this number range is used for automatically generated F-blocks by default. You can adapt the number range used by the F-system, but you must select a sufficiently large range of numbers.

Table 11-2 Assignment overview of the FB numbers

FB number	Availability	Occupied by
0	Reserved	Siemens
1	Reserved	Siemens (PLC basic program)
2 … 3	Occupied	Siemens (PLC basic program)
4	Reserved	Siemens
5	Occupied	Siemens (PLC basic program)
6	Reserved	Siemens
7	Occupied	Siemens (PLC basic program)
8	Reserved	Siemens
9	Occupied	Siemens (PLC basic program)
10 … 29	Reserved	Siemens
15	Reserved	Siemens (PLC basic program)
16…28	Reserved	Siemens
29	Reserved	Siemens (PLC basic program)
30…999	Freely available (user program)	-
1000 … 1009	Reserved	Siemens
1010 … 1019	Reserved	Exclusively available for F blocks of Safety Integrated plus (F-PLC)
1020	Occupied	Siemens (PLC basic program)
1021 … 1023	Reserved	Siemens
1024 … 24999	Freely available (user program)	-
25000 … 32766	Reserved	Siemens (PLC basic program)
32767 … 65535	Reserved[2]	Standard number range for generated F-system blocks

[2]If you use SINUMERIK Safety Integrated plus (F-PLC), this number range is used for automatically generated F-blocks by default. You can adapt the number range used by the F-system, but you must select a sufficiently large range of numbers.

Table 11-3 Assignment overview of the FC numbers

FC number	Availability	Occupied by
0	Reserved	Siemens
1 … 3	Occupied	Siemens (PLC basic program)
4	Reserved	Siemens

FC number	Availability	Occupied by
5	Reserved	Siemens (PLC basic program)
6 … 29	Reserved	Siemens
30 … 999	Freely available (user program)	-
1000 … 1023	Reserved	Siemens
1024 … 24999	Freely available (user program)	-
25000	Occupied	Siemens (PLC basic program)
25001 … 32766	Reserved	Siemens (PLC basic program)
32767 … 65535	Reserved[2]	Standard number range for generated F-system blocks

[2]If you use SINUMERIK Safety Integrated plus (F-PLC), this number range is used for automatically generated F-blocks by default. You can adapt the number range used by the F-system, but you must select a sufficiently large range of numbers.

PLC functions for HMI (DB19) 12

12.1 Channel selection

Function

The channel displayed on the HMI, e.g. in the machine start screen, can be selected from the PLC user program via the HMI/PLC interface.

Requirement

More than one channel is parameterized in the NC.

Job and acknowledgment interface

DB19	Meaning	
DBX32.0 - .5	PLC → HMI	Function number: 1 = channel selection
DBX32.6	PLC → HMI	Function request
DBX32.7	HMI → PLC	Status: 1 = "function being executed"
DBB33	PLC → HMI	Channel number: 1, 2, 3, ... maximum number of channels Next channel: FF_H
DBB36	HMI → PLC	Error identification: • 0: No error • 1: Invalid function number (DBX32.0 - .5) • 2: Invalid parameter (DBB33 - DBB35) • 3: Error when writing the HMI-internal variable • 10: Channel not present (DBB33)

Functional sequence

PLC → HMI

The PLC user program must maintain the following execution sequence:

1. Check whether the interface is free for a new job:
   - `DB19.DBX32.6 == 0` (function request)
   - `DB19.DBX32.7 == 0` (status)
2. If the interface is free, the job data must be entered and the function request set:
   - `DB19.DBB33 = <channel number>`
   - `DB19.DBX32.0 - .5 = 1` (function number)
   - `DB19.DBX32.6 = 1` (function request)

HMI → PLC

The HMI makes the following responses for **error-free** parameterization:

1. Once the HMI has recognized the function request for channel selection, the status is set to "function being performed" and the function request reset:
   - `DB19.DBX32.7 = 1` (status)
   - `DB19.DBX32.6 = 0` (function request)
2. Once the channel selection has been performed, the status is reset again and value 0 is set as error identification:
   - `DB19.DBX32.7 = 0` (status)
   - `DB19.DBX36 = 0` (error identification)

The HMI makes the following responses for **faulty** parameterization:

- The function request is reset and the appropriate error identification is set:
  - `DB19.DBX32.6 = 0` (function request)
  - `DB19.DBX36 = <error identification>`

12.2 Program selection

Function

Preselected programs/workpieces can be selected for machining by the NC via the PLC/HMI interface.

The preselection is implemented by entering programs/workpiece in files, these are known as PLC program lists (*.ppl).

Requirements

The following machine data must be set to allow the HMI to process tasks:

MD9106 $MM_SERVE_EXTCALL_PROGRAMS

In order to activate a sector-specific PLC program list, you must set the appropriate machine data and at least the protection level password:

- Area **User**
  - MD51041 $MN_ENABLE_PROGLIST_USER = 1
  - Protective level password: 3 (users)
  - Program list: /user/sinumerik/hmi/plc/programlist/plc_proglist_user.ppl
- Area **Manufacturer (OEM)**
  - MD51043 $MN_ENABLE_PROGLIST_MANUFACT = 1
  - Protective level password: 1 (manufacturer)
  - Program list: /oem/sinumerik/hmi/plc/programlist/plc_proglist_manufacturer.ppl

Structure of a program list

A program list is a text file. Each line contains the following information:

<program number> <program path><program name> [CH=<channel number>]

- Program number
  The program numbers which may be used in a program list depend on the sector:
  - user: 1 - 100
  - Manufacturer (OEM): 201 - 255
- Program path
  The program path must be completely specified in absolute terms.
  For specifying the program path, see:
  Further information
  Programming Manual NC Programming, Section "File and program management" > " Program memory" > "Addressing of program memory files"
- Channel number
  Specifying the channel number "CH=<channel number>" is optional. It is only required if the NC has more than one channel.

The following excerpt as example shows the structure of the user program list:

Program list: plc_proglist_user.ppl

```
1 //DEV2:/MPFDir/PROG_01.MPF CH=1
2 //DEV2:/MPFDir/PROG_01.MPF CH=2
```

Generating entries in a program list

The entries in a program list (*.ppl) can be directly edited in the file or entered in screen forms in the user interface.

- Via the user interface for the **user** area
  Operating area "Program Manager" > "ETC key (">")" > "Prog. list"
- Via the user interface for the **Manufacturer** area
  Operating area "Commissioning" > "System data" > "ETC key (">")" > "Prog. list"

Program selection: Job interface

Note

The PLC may only request a new job if the last job has been acknowledged by the HMI: DB19.DBB26 == 0

Program list

DB19.DBB16 = <number of the program list>

Number	Program list
129	/user/sinumerik/hmi/plc/programlist/**plc_proglist_user.ppl**
131	/oem/sinumerik/hmi/plc/programlist/**plc_proglist_manufacturer.ppl**

Program number

The program number refers to the programs contained in the selected program list.

DB19.DBB17 = <program number>

- user area: 1 - 100
- oem area: 201 - 255

Requesting program selection

DB19.DBX13.7 = 1

Program selection: Acknowledgment interface

Job acknowledgment

- DB19.DBX26.7 == 1 (selection identified)
- DB19.DBX26.3 == 1 (program is selected)
- DB19.DBX26.2 == 1 (error when selecting the program, see error ID DB19.DBB27)
- DB19.DBX26.1 == 1 (job completed)

Error code

DB19.DBB27 == <value>

<Value>	Meaning
0	No error
1	Invalid program list number (DB19.DBB16)
3	User-specific program list plc_proglist_main.ppl not found (for DB19.DBB16 ≠ 129, 131 only)
4	Invalid program number (DB19.DBB17)
5	The job list in the selected workpiece could not be opened
6	Error in job list (Job list Interpreter returns error)
7	No instructions were found in the job list
8	HMI could not execute the selection (SELECT)
9	An error with no further details occurred upon selection
10	An error occurred during analysis of the job list
11	The instruction is not available
12	Invalid instruction (LOAD, UNLOAD, SELECT only)
13	Not all channels have the Reset status
14	The channel number is invalid

Program selection: Job processing

A job to select a program is executed as follows:

1. Checking the acknowledgment byte: DB19.DBB26 == 0
   If the acknowledgment byte is not 0, then the last job has still not been completed.
2. Specifying the program list: DB19.DBB16

3. Specifying the program number: DB19.DBB17
4. Setting the request to select a program: DB19.DBX13.7 = 1
5. Evaluating the acknowledgment and error interface: DB19.DBB26 and DBB27
   The order is still not completed on the HMI side as long as: DB19.DBX26.3 == 1 (active)
   The order has been completed on the HMI side if one of the two signals has been set:
   - DB19.DBX26.1 == 1 (OK)
   - DB19.DBX26.2 == 1 (error)
6. To complete the order, the program selection request must be reset: DB19.DBX13.7 = 0
7. The HMI signals that it is ready to accept a new order by resetting the acknowledgment byte: DB19.DBB26 == 0

12.3 Activating the key lock

The operator panel keyboard and a keyboard directly connected to the HMI can be locked using the following interface signal

- 1. HMI: DB19.DBX0.2 = <value>
- 2. HMI: DB19.DBX50.2 = <value>

Value	Meaning
0	Key lock inactive
1	Key lock active

12.4 Operating area numbers

The number of the active operating area is normally displayed in: DB19.DBB21

If the HMI monitor is active, the number of the active operating area is no longer displayed in DB19.DBB21, but instead in the user-specific configured area of the HMI monitor (Page 123).

Operating area	Number
Machine	201
Parameters	205
Programming	203
Program Manager	202
Diagnostics	204
Commissioning	206

12.5 Screen numbers

The current screen number is normally displayed in: DB19.DBW24

If the HMI monitor is active, the current screen number is no longer displayed in DB19.DBW24, but instead in the user-specific configured area of the HMI monitor (Page 123).

Screen number ranges

The following screen number ranges are available:

- JOG, manual machine (Page 114)
- Reference point approach (Page 119)
- MDA (Page 119)
- AUTOMATIC (Page 120)
- Parameters operating area (Page 120)
- Program operating area (Page 122)
- Program manager operating area (Page 122)
- Diagnostics operating area (Page 122)

12.5.1 Screen numbers: JOG, manual machine

JOG mode

Screen	Number
Turning technology	
Cycle start screen for all screens that can be taken over	81
Milling technology	
Cycle start screen for all screens that can be taken over	3
Turning/milling	
Start screen	19
T,S,M	2
Set WO	21
Positioning	4
Face milling	18
Stock removal	80
Cycle start screen for all user screens	91
General settings	1
Multi-channel function settings	106
Collision avoidance settings	107
Measurement log settings	108
Swiveling	60
All G commands	100
Actual zoom value (MCS/WCS)	101
Thread synchronizing	102
Retract	103

Screen	Number
Handwheel	104
Action synchronization	105
Turning technology: Workpiece zero	
Workpiece zero (main menu)	30
User screen	31
User screen	34
User screen	35
User screen	36
User screen	37
User screen	38
User screen	40
Measure edge Z	5
Turning technology: Workpiece, measurement	
Measure tool (main menu)	50
Manual X or user screen	51
Manual Y	71
Manual Z or user screen	52
Zoom or user screen	53
User screen	54
User screen	55
Probe calibration X or user screen	56
Probe calibration Z or user screen	57
Automatic length in Z	58
Automatic length in Y	73
Automatic length in X	59
Milling technology: Workpiece zero	
Workpiece zero (main menu)	30
Measure edge X	5
Measure edge X	22
Measure edge Z	23
User screen	7
Align edge or user screen	31
Distance 2 edges or user screen	32
Right-angled corner	33
Any corner or user screen	8
Rectangular pocket	34
1 hole or user screen	9
2 holes	35
3 holes	36
4 holes	37
Rectangular spigot	38
1 circular spigot or user screen	10
2 circular spigots	39

Screen	Number
3 circular spigots	40
4 circular spigots	41
Set up level	42
Probe length calibration or user screen	11
Probe radius calibration	12
Milling technology: Workpiece, measurement	
Measure tool (main menu)	50
Measure length, manual (with milling tool) or measure length in X, manual (with turning tool) or user screen	16
Measure length in Y, manual (with turning tool)	74
Measure length in Z, manual (with turning tool)	24
Measure diameter, manual or user screen	17
Measure length, automatic (with milling tool) or measure length in X, automatic (with turning tool) or user screen	13
Measure length in Y, automatic (with turning tool)	75
Measure length in Z, automatic (with turning tool)	25
Measure diameter, automatic or user screen	14
User screen	51
Probe calibration or user screen	15
Fixed point calibration or user screen	52
RunMyScreens (only for set JobShopIntegration)	
User screen for the 1st horizontal softkey	96
User screen for the 2nd horizontal softkey	98
User screen for the 3rd horizontal softkey	99
User screen for the 4th horizontal softkey	94
User screen for the 5th horizontal softkey	95
User screen for the 6th horizontal softkey	92
User screen for the 7th horizontal softkey	97
User screen for the 8th horizontal softkey	90
User screen for the 9th horizontal softkey	83
User screen for the 10th horizontal softkey	82
User screen for the 11th horizontal softkey	93
User screen for the 12th horizontal softkey	84
User screen for the 13th horizontal softkey	85
User screen for the 14th horizontal softkey	86
User screen for the 15th horizontal softkey	87
User screen for the 16th horizontal softkey	88

JOG mode, manual machine

DB19.DBB24	
Screen	**Screen number**
Turning/milling	
Taper turning	61
Angle milling	62
Stop	63
Straight line	1300
Straight line all axes	1330
Straight line X alpha	1340
Straight line Z alpha	1350
Circle	1360
Drilling	1400
Center drilling	1410
Drilling, thread centered	1420
Drilling, centering	1433
Drilling, drilling	1434
Drilling, reaming	1435
Drilling, boring	1436
Drilling, deep hole drilling	1440
Drilling, deep hole drilling 2	1441
Drilling, tapping	1453
Drill thread milling	1455
Positions	1473
Position row	1474
Position grid	1477
Position frame	1478
Position circle	1475
Position pitch circle	1479
Obstacle	1476
Turning	1500
Turning, stock removal 1	1513
Turning, stock removal 2	1514
Turning, stock removal 3	1515
Turning, groove 1	1523
Turning, groove 2	1524
Turning, groove 3	1525
Turning, undercut form E	1533
Turning, undercut form F	1534
Turning, undercut thread DIN	1535
Turning, undercut thread	1536
Turning, thread, longitudinal	1543
Turning, thread, taper	1544

DB19.DBB24	
Screen	**Screen number**
Turning, thread, facing	1545
Turning, thread, chain	1546
Turning, cut-off	1550
Milling	1600
Milling, face milling	1610
Milling, rectangular pocket	1613
Milling, circular pocket	1614
Milling, rectangular spigot	1623
Milling, circular spigot	1624
Milling, longitudinal groove	1633
Milling, circumferential groove	1634
Milling, open groove	1635
Milling, multi-edge	1640
Milling, thread milling	1454
Milling, engraving	1670
Contour turning	1200
Contour turning, new contour / last contour	1210
Contour turning, stock removal along contour	1220
Contour turning, contour grooving	1230
Contour turning, contour plunge turning	1240
Contour milling	1100
Contour milling, new contour / last contour	1110
Contour milling, path milling	1120
Contour milling, centering	1130
Contour milling, rough drilling	1140
Contour milling, contour pocket	1150
Turning technology: Simulation	
Side view	1740
Front view	1750
3D view	1760
2-window view	1770
Half section	1780
Turning technology: Simultaneous recording	
Side view	1741
Front view	1751
3D view	1761
2-window view	1771
Machine space	1791
Half section	1781
Milling technology: Simulation	
Top view	1742
3D view	1760

DB19.DBB24	
Screen	**Screen number**
From the front	1744
From the rear	1746
From the Left	1748
From the right	1752
Half section	1780
Turning view	1782
Milling technology: Simultaneous recording	
Top view	1743
3D view	1761
From the front	1745
From the rear	1747
From the Left	1749
From the right	1753
Machine space	1791
Half section	1781
Turning view	1783

12.5.2 Screen numbers: Reference point approach

Screen	Number
Actual zoom value MCS/WCS	101

12.5.3 Screen numbers: MDA

Screen	Number
MDI	20
All G commands	100
Actual zoom value MCS/WCS	101
Handwheel	104
Action synchronization	105
Program control	210
Settings	250

12.5.4 Screen numbers: AUTOMATIC

Screen	Number
Automatic	200
Overstore	202
Program control	210
Block search	220
General settings	250
Multi-channel function settings	106
Collision avoidance settings	107
All G commands	100
Actual zoom value MCS/WCS	101
Handwheel	104
Action synchronization	105
Turning technology: Simultaneous recording	
Side view	243
Front view	244
3D view	245
2-window view	246
Machine space	247
Half section	253
Milling technology: Simultaneous recording	
Top view	242
3D view	244
From the front	248
From the rear	249
From the Left	251
From the right	252
Machine space	247
Half section	253
Turning view	254

12.5.5 Screen numbers: Parameters operating area

Screen	Number
Tool list	600
Tool wear	610
User tool list	620
Magazine	630
Work offset	
Work offset, active	642
Work offset, overview	643

Screen	Number
Work offset, basic	644
Work offset, G54 - G509	645
Details of work offset, active, overview, basic or G54 - G509	647
User variable	
R parameters	650
Global GUD 1 (SGUD)	660
Global GUD 2 (MGUD)	661
Global GUD 3 (UGUD)	662
Global GUD 4	663
Global GUD 5	664
Global GUD 6	665
Global GUD 7	666
Global GUD 8	667
Global GUD 9	668
Channel GUD 1 (SGUD)	690
Channel GUD 2 (MGUD)	691
Channel GUD 3 (UGUD)	692
Channel GUD 4	693
Channel GUD 5	694
Channel GUD 6	695
Channel GUD 7	696
Channel GUD 8	697
Channel GUD 9	698
Local LUD	681
Local LUD/PUD	684
Setting data	
Working area limitation	671
Spindle data	670
Spindle chuck data	672
Ctrl-Energy	
Ctrl-Energy, main menu	6170
Ctrl-Energy, analysis	6171
Ctrl-Energy, profiles	6172
Ctrl-Energy, analysis graphic	6176
Ctrl-Energy, analysis long-term measurement	6177
Ctrl-Energy, analysis details	6179
Ctrl-Energy, compare measurements	6178

12.5.6 Screen numbers: Program operating area

Screen	Number
Turning technology: Simulation	
Side view	413
Front view	414
3D view	415
2-window view	416
Half section	423
Milling technology: Simulation	
Top view	412
3D view	414
From the front	418
From the rear	419
From the Left	421
From the right	422
Half section	423
Turning view	424

12.5.7 Screen numbers: Program manager operating area

Screen	Number
NC directory	300
Local drive	325
USB / configured drive1	330
Configured drive2	340
Configured drive3	350
Configured drive4	360
Configured drive5	383
Configured drive6	384
Configured drive7	385
Configured drive8	386

12.5.8 Screen numbers: Diagnostics operating area

Screen	Number
Alarm list	500
Messages	501
Alarm log	502
NC/PLC variable	503

12.6 HMI monitor

Function

The HMI monitor is an 8-byte data area in a freely selectable data block, in which the HMI can provide the following data for the PLC user program:

- Operating area numbers (Page 113)
- Screen numbers (Page 113)

Parameterization

The data area is configured using the following display machine data:

MD9032 $MM_HMI_MONITOR = "string"

with "string" = "DB<DB number>.DBB<byte address>"

Note

Even byte address

The data area must start at an even byte address.

Structure of the data area

Byte	Meaning
EB n + 0	Active SINUMERIK operating area
EB n + 1	Reserved
EB n + 2	Current screen number
EB n + 3	
EB n + 4	Reserved
...	...
EB n + 7	Reserved

Supplementary conditions

When the HMI monitoring is active, the following PLC/HMI interface signals are no longer processed:

- DB19.DBB10 (PLC hardkeys)
- DB19.DBB21 (active SINUMERIK operating area)
- DB19.DBW24 (current screen number)

Example

Assumptions

- Current operating area: "Machine", number: 201
- Actual screen: "AUTOMATIC" start screen, number: 200
- PLC data area: DB60.DBB10

Parameterization

- MD9032 $MM_HMI_MONITOR = "DB60.DBB10"

Values in the data area

- DB60.DBB10: 201
- DB60.DBW12: 200

PLC functions for drive components on the integrated PROFIBUS

13

13.1 Overview

Using the function described below, input and output data from drive components on the integrated PROFIBUS can be consistently, cyclically read and written from the PLC user program of the **hardware PLC**. The following boundary conditions must be observed:

- For reading / writing input/output data, the system functions **SFC14 / SFC15** must be used.
- The reading / writing of input/output data is always over the **entire** slot length.
- The function only supports cyclical, **non-equidistant** data transfer.
- An output slot may not already be occupied on the NC side; (e.g. output slots of drives).

13.2 Performing a start-up

Preconditions

Before the function is put into operation on the NC side, the following preconditions must be satisfied:

- The drive components on the integrated PROFIBUS of the NCU must be fully configured using SIMATIC STEP 7, HW Config.
- The configuration is loaded into the PLC.

NC machine data

The start addresses of the slots to be transferred cyclically are to be entered into the following machine data:

- Input slots:
  MD10520 $MN_PLCINTERN_LOGIC_ADDRESS_IN[<Index>] = <Slot address>
- Output slots:
  MD10525 $MN_PLCINTERN_LOGIC_ADDRESS_OUT[<Index>] = <Slot address>

<Index>: 0, 1, 2, ... (Max no. of slots) – 1

<Slot address>: The slot address parameterized in HW Config

Note

Maximum quantity of data

The sum of all data to be cyclically transferred in inward and outward directions may not currently exceed **2,048** bytes.

13.3 Example

Determining slot addresses

After selecting the DP Slave "SINAMICS_Integrated" on the integrated PROFIBUS "PROFIBUS Integrated: DP master system (3)" in the station window of HW Config, its PROFIdrive message frame and associated slot addresses are displayed in the detailed view.

- Message frame 136: Drive
- Message frame 391: Control Unit
- Message frame **370**: Infeed

The infeed slot addresses required for parameterization in the NC machines are:

- Input slot: Slot 31, address **6514**
- Output slot: Slot 32, address **6514**

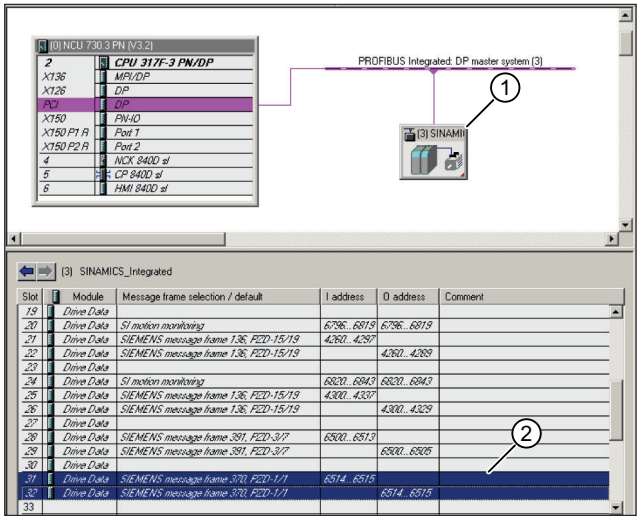


Figure 13-1 Infeed PROFIBUS message frames

Setting NC machine data

- Input slots:
  MD10520 $MN_PLCINTERN_LOGIC_ADDRESS_IN[0] = 6514
- Output slots:
  MD10525 $MN_PLCINTERN_LOGIC_ADDRESS_OUT[0] = 6514

Controlling the ALM using FB390 "ALM_Control"

The SINUMERIK hardware PLC is connected to the CU320 of SINAMICS S120 through the internal PROFIBUS. The ALM is connected to the CU via DRIVE-CLiQ.

In SINAMICS S120, a control and status message frame can be defined for each module (the CU, ALM, motor modules, etc.). If this is performed for an ALM, this can be switched on and off from the PLC user program.

The SIMATIC S7 block FB390 "ALM_Control" checks the status of the ALM and enables the user to switch it on or off.

A description of the block and an example project are available for download under the following link to Industry Online Support:

http://support.automation.siemens.com/WW/view/de/49515414

Memory requirements of the basic PLC program 14

The basic program consists of basic and optional functions. The **basic functions** include cyclic signal exchange between the NC and PLC. The **Options** include e.g. the FCs, which can be used, if needed.

The table below lists the memory requirements for the basic functions and the options. The data quoted represent guide values, the actual values depend on the current software version.

Memory requirements of blocks for SINUMERIK 840D sl			
Block Type, No.	**Function**	**Remark**	**Block size (bytes)**
	Working memory		
Basic functions in basic program			
FB1, FB15		Must be loaded / on CompactFlash Card	52182
FC2, 3, 5, 12		Must be loaded	470
DB4, 5, 7, 8		Must be loaded	1006
DB2, 3, 17		Are generated by the BP	632
OB1, 40, 100, 82, 86		Must be loaded	398
		Total	55698
PLC/NC, PLC/HMI interface			
DB10	PLC/NC signals	Must be loaded	262
DB11	PLC / mode group signals	Is generated by BP	56
DB19	PLC/HMI signals	Is generated by BP	434
DB21 to 30	PLC/channel signals	Are generated by BP as a function of NCMD: for each DB	416
DB31 to 61	PLC / axis or spindle signals	Are generated by BP as a function of NCMD: for each DB	148
Basic program options			
	Machine control panel		
FC19	Transfer of MCP signals, M variant	Must be loaded when M variant of MCP is installed	92
FC25	Transfer of MCP signals, T variant	Must be loaded when T variant of MCP is installed	92
FC24	Transfer of MCP signals, slim variant	Must be loaded when slim variant of MCP is installed	100
FC26	Transfer of MCP signals, HT8 variant	Must be loaded for HT8	68

Basic program options			
	Handheld unit		
FC13	Display control HHU	Can be loaded for handheld units	144
	Error/operating messages		
FC10	Acquisition FM/BM	Load when FM / BM is used	66
	ASUP		
FC9	ASUP start	Load when PLC ASUPs are used	128

Basic program options			
	Star/delta changeover		
FC17	Star/delta switchover of MSD	Load for star/delta switchover	114
	Spindle control		
FC18	Spindle control	Load for spindle control from PLC	132
	PLC/NC communication		
FB2	Read NC variable	Load for Read NC variable	76
DBn	Read NC variable	One instance DB per FB2 call:	270
FB3	Write NC variable	Load for Write NC variable	76
DBm	Write NC variable	One instance DB per FB3 call:	270
FB4	PI services	Load for PI services	76
DBo	PI services	One instance DB per FB4 call:	130
DB16	PI services description	Load for PI services	618
FB5	Read GUD variables	Load for PI services	76
DBp	Read GUD variables	One instance DB per FB5 call:	166
DB15	General communication	Global data block for communication	146
FB7	PI services 2	Load for PI services	76
DBo	PI services 2	One instance DB per FB4 call: every	144
FC21	Transfer	Load with dual-port RAM, ...	164
	M to N		
FB9	Switchover M to N	Load with M to N	58
	Safety Integrated		
FB10	Safety relay	Load with Safety option	74
FB11	Brake test	Load with Safety option	76
DB18	Safety data	DB for Safety	308
	Tool management		
FC7	Transfer function turret	Load for tool management option	84
FC8	Transfer function	Load for tool management option	132
FC22	Direction selection	Load, when direction selection is needed	138
DB71	Loading locations	Generated by BP as a function of NC MD	40+30*B
DB72	Spindles	Generated by BP as a function of NC MD	40+48*Sp

Basic program options			
DB73	Revolver	Generated by BP as a function of NC MD	40+44*R
DB74	Basic function	Generated by BP as a function of NC MD	100+(B+ Sp+R)*22
	Compile cycles		
DB9	Interface PLC compile cycles	Is generated by BP as a function of NC option	436

Example:

Based on the memory requirements in the table above, the memory requirements have been determined for two sample configurations (see table below).

Block Type, No.	Function	Remark	Block size (bytes)
	Working memory		
Minimum configuration (1 spindle, 2 axes and T MCP)			
see above	Basic program, base		54688
	Interface DBs		1612
	MCP		92
		Total	56392

Block Type, No.	Function	Remark	Block size (bytes)
	Working memory		
Maximum configuration (2 channels, 4 spindles, 4 axes, T MCP)			
see above	Basic program, base		54688
see above	Interface DBs		2768
see above	MCP		92
see above	Error/operating messages		66
see above	ASUPs	1 ASUP initiation	128
see above	Concurrent axis	For 2 turrets	132
see above	PLC/NC communication	1 x read variable and 1 x write variable	838
see above	Tool management	2 turrets with one loading point each	674
see above	Compile cycles		436
		Total	59822

NC VAR selector

15

15.1 Overview

General

The PC application "NC VAR selector" retrieves the addresses of required NC variables and prepares them for access in the PLC program (FB2/FB3). This enables the PLC programmer to select NC and drive variables from the entire range of NC and drive variables, to store this selection of variables, and to prepare them with a code generator for the STEP 7 compiler. The data is then stored as ASCII file (*.awl) in the machine CPU program. The "NC VAR Selector" graphic shows this process.

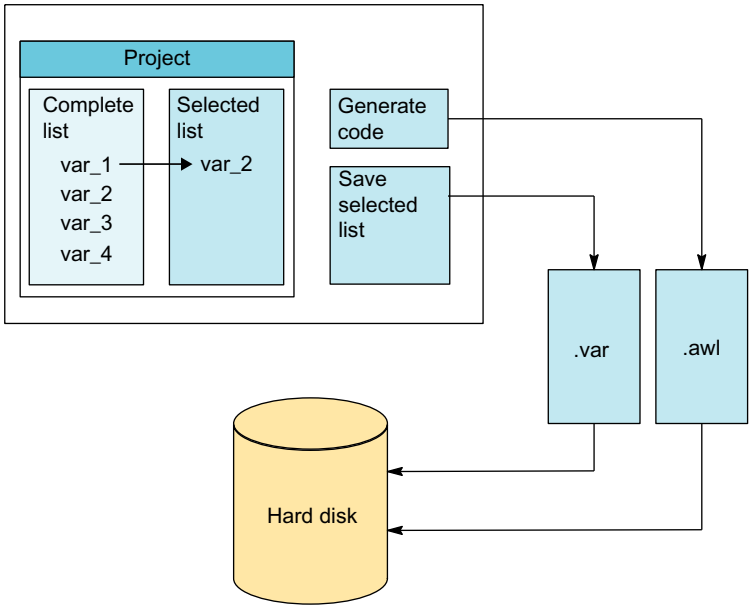


Figure 15-1 NC VAR selector

To store the files created by NC VAR selector, create via Windows Explorer an arbitrary folder where the selected data of NC VAR selector can be stored (data.VAR and data.AWL files). Then transfer and compile the data in a STEP 7 project with "Code - in STEP 7 project". As alternative, insert the "data.awl" file into the STEP 7 machine project via SIMATIC Manager -

Program LAD/STL/FBD blocks with "Insert - External source". The file is stored in the project structure. Once the file has been transferred, compile the AWL (STL) files with STEP 7.

Note

The latest NC VAR selector can be used for each NC software release (even older versions). For older NC software releases, the variables can also be selected from the latest complete list. The data content in DB120 (default DB for variables) does not depend on the software version. This means, variables selected in an older software release need not be reselected when the software is upgraded.

After starting the "NC VAR selector" application, the variables of the current software version are displayed.

Transfer the variables to the project list (another window). These selected variables can then be stored in an ASCII file, prepared as a STEP 7 source file (.awl) and stored.

After creating a PLC data block by means of the STEP 7 compiler, the PLC programmer is able to read or write NC variables via the basic program function blocks "PUT" and "GET" using the created STEP 7 file.

The project list is also stored as an ASCII file (file extension .var).

The variable list supplied with the "NC VAR selector" tool is adapted to the current software release. This list does not contain any variables (GUD variables) defined by the user. These variables are processed by the function block FB5 in the basic program.

System features, supplementary conditions

The PC application "NC VAR selector" requires Windows XP or a later operating system.

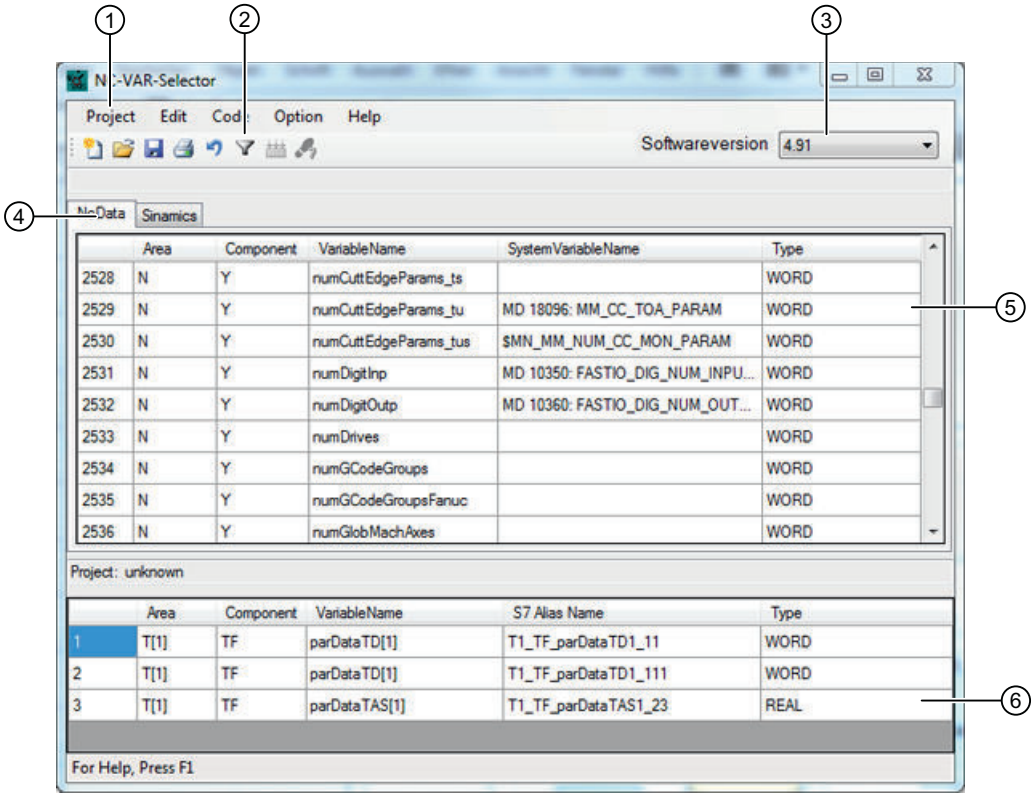
The assignment of names to variables is described in:
Further information:
/List sl (Book1); Section: Variables
and also in the variables help file (integrated in NC VAR selector)

15.2 Using NC VAR selector

15.2.1 NC VAR selector user interface

Software user interface structure

The user interface is structured as follows:



① Menu bar
② Available buttons
③ Drop-down list with the available software versions
④ Tab for switching between NC variables and SINAMICS parameters
⑤ List of the NC variables (variable list)
⑥ List of the selected NC variables (project list)

Figure 15-2 Software user interface

Buttons

Symbol	
	Creates a new project. The file selection for the NC variable list opens (applies only when the NC variable list is not already open).
	Opens an existing project (*.var). The project already contains a stored list of selected variables
	Stores the list of selected variables as project in the *.var format.
	Prints the project list.
	Undoes the last edit step.
	Opens the filter dialog to filter the available variables.
	Generated STL code with the set properties.
	Transfers the generated NCVar data blocks to a STEP 7 project.

Main menu

The main menu contains the following entries:

Menu entry	Menu item	Description
Project	New	Creates a new project and closes a previously opened project.
	Open	Opens an existing project.
	Save	Saves a project in *.var format.
	Save as	Saves a project with a new name, with the option of changing the path.
	Open database	Opens an NC variable list in "mdb" database format.
	Print preview	Shows the print preview of the selected NC variables.
	Print	Prints the selected NC variables as list.
	Exit	Exits NC VAR selector.
Edit	Undo	Undoes the last edit step.
	Transfer NC variables	Transfers the selected NC variables to the project list.
	Delete in project	Deletes selected NC variables from the project.
	Change alias name in the project	Opens a dialog for changing the alias name of the selected variable in the project list.
	Change variable parameters	Opens a dialog for changing the parameters of a variable in the project list.
	Search for / filter NC variables	Opens a dialog for searching for / filtering variables.
Code	Settings	Opens a dialog with which the code generation can be configured.
	Generate	Initiates the generation of STL code.
Help	Online helps for various SINUMERIK data and drive data.	Contains a list of online help for various SINUMERIK data and drive data.

15.2.2 Description of functions

Overview

The figure below illustrates how NC VAR selector is used within the STEP 7 environment.

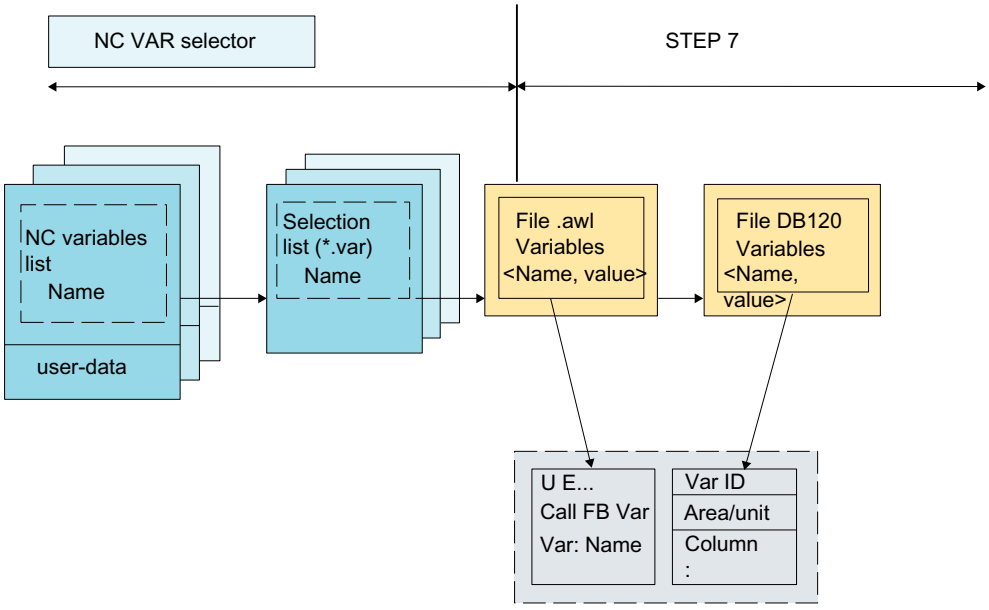


Figure 15-3 Application of NC VAR selector in the STEP 7 environment

NC VAR selector generates a list of selected variables from a list of variables and then creates an **.awl** file that can be compiled by the STEP 7 compiler.

- An *.awl file contains the names and alias names of the NC variables, as well as information about their address parameters. A data block created from this file only contains the address parameters (10 bytes per parameter).
- The created data blocks must always be stored in the machine-specific file storage according to STEP 7 specifications.
- To ensure that the parameterization of the FB2 and FB3 blocks with regard to NC addresses can be implemented with symbols, the freely assignable, symbolic name of the created data block must be included in the STEP 7 symbol list.

15.2.3 Editing projects

Create a project

The data relevant for a project is stored in *.var file format. When the NC VAR selector is opened a project is created implicitly and an empty variable list is shown.

1. Perform "Project - New" to create a new project and to close any currently open project.
2. If a project with a variable list already exists, a query opens:
   – Discard the settings of the opened project; all settings are deleted.
   – Save the project settings. The "Save as" dialog opens.
   Enter a name for the project

The open project closes. The newly created project does not have any name. To do this, save the project with a new name with "Project - Save as".

Save project

After creating a project, only the "Save as" menu item is active.

1. Perform "Project - Save as" to save the project with a new name.
2. If the data of a project has been changed, perform "Project - Save" to save the changes.

Open project

Existing projects can be opened in NC VAR selector.

1. Perform "Project - Open". The "Open" dialog opens.
2. Navigate to the path of the project to be opened.
3. Select the project (*.var) and click "Open".
   The project with the stored variable list opens, the open project closes.

Print project

The selected variables can be printed. The layout of the printed documents can be adapted with the print preview.

1. Perform "Project - Print" to send the variable list to the selected printer.
2. Perform "Project - Print preview" to check the print-out before printing and to adapt the layout.

15.2.4 Filtering the variable list

Structure of the variable list

The variable list is created as table. The structure of the table is oriented on the structure of the NC variables:

- Area; the scope of the NC variables is subdivided into areas similar to machines:
  - [A]; axis-specific basic settings
  - [B]; operating mode groups
  - [C]; channels
  - [M]; HMI data
  - [N]; NC data
  - [T]; tool data
  - [V]; drive parameters
- Blocks; each area contains several blocks to which the NC variables are assigned.
- Variable name; shows the name of the NC variable.
- System variable / machine data item; shows the name of the corresponding system variable / machine data item.
- Type; shows the data type of the variable.

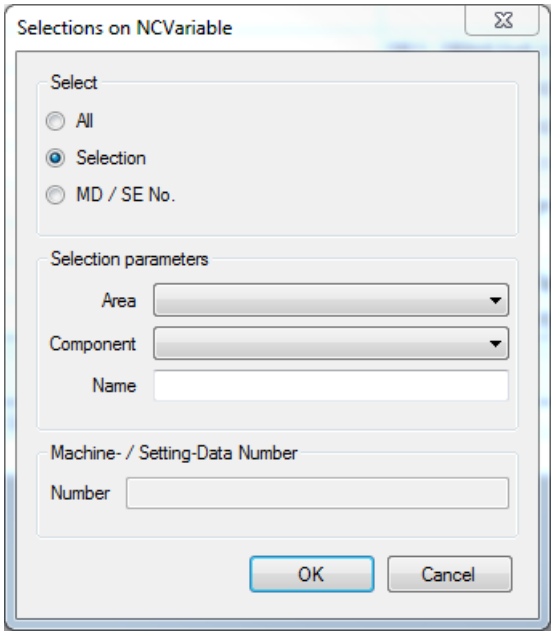
Select software version

The available NC variables differ for each software version. The "Software version" drop-down list lists all versions contained in NC VAR selector.

1. Select in the "Software version" drop-down list, the software version to be used for the project.
   After the selection, a query opens to make the decision whether the currently selected variables should be discarded.
2. Click "Yes" to discard the settings and to display the variables for the specified version.

Filtering the variable list

To restrict the number of displayed variables, the variables can be filtered:



To display the subsets of NC variables, filter criteria can be defined in the "Select NC variables" dialog:

Filter area	Filter criterion	Description
Selection	All	Displays all variables/parameters
	Filter	Activates the filter criteria
	MD / SE No.	Activates the "Number" input field in order to enter the number of a variable / system variable / machine data item. For drive data, this is the parameter number
Filter criteria	Area	Permits the selection of the area for NC variables and the display of all variables of the area.
	Block	Permits the selection of the block and the display of all variables of the block. If the area is already selected, but the selected block is not available there, no result is returned.
	Name	Full text search for variables. Fragments of the variable name can also be entered. All hits with this fragment are displayed.
Machine / setting data number	Number	Input field for the number of the variable / system variable / machine data item. Only the variable with this number is displayed.

To reset the filters, the new filter setting must be performed.

Example search criteria

Name search criterion: CHAN*	Found:	CHAN_NAME
		chanAlarm
		chanStatus

channelName
chanAssignment

15.2.5 Specifying the project list

Structure of the project list

The project list table has the following structure:

- Area; the scope of the NC variables is subdivided into areas similar to machines.
- Blocks; each area contains several blocks to which the NC variables are assigned.
- Variable name; shows the name of the NC variable.
- S7 alias name; the name that can be used in STEP 7 instead of the variable name.
- Type; shows the data type of the variable.

Transferring variables to the project list

All variables required for creating the data block must be added to the project list.

1. Search for the variables to be added to the project list.
2. Double-click the variable.
   - OR -
3. Perform "Transfer NC variables" via the context menu.
   The "Input Area No., Row, Column" dialog opens.

"Input Area No., Row, Column" dialog

The variable parameters are entered in the "Input Area No., Row, Column" dialog

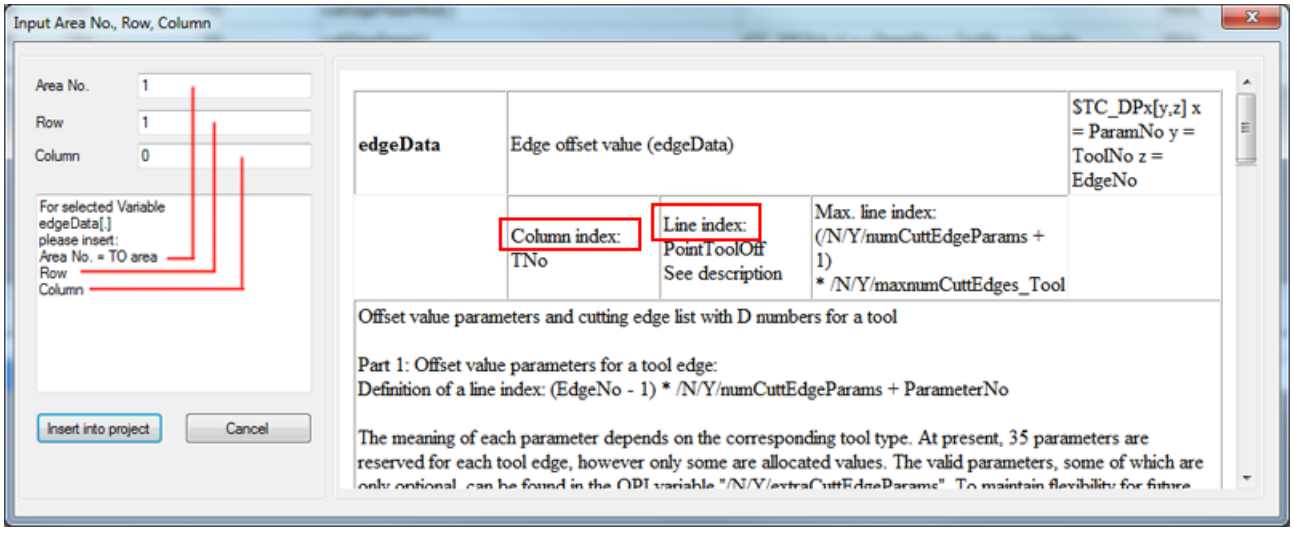


Figure 15-4 Rows, columns and area numbers

The dialog shows an excerpt from the variable description. Fields in which parameters can be entered are active. The description how the various parameters can be entered correctly is shown below the input fields.

- The value that can be entered in the "Area number" field to control the correct variable is described in this field.
- The meaning of row and column is contained in the variable description. The "Column index" and "Row index" fields there describe the associated inputs.

Parameterizing variables

The associated required parameters must be entered for the various variable types:

- Single-line variable; only the area number must be entered to access variables of this type. If this is also fixed, the variable can be used without further parameterization.
- Multi-line variable: A variable of this type is a one-dimensional field. The "area number" and a "line" must be entered for access.
- Multi-line and multi-column NC variables: A variable of this type is a two-dimensional field. The "area number", a "line" and a "column number" must be entered for access.

Special case "pointer"

By entering a zero (0) as area number, row or column index, it is possible to use the variable in the S7 context as a pointer to this data. When reading or writing this data via the functions "PUT" and "GET", the optional parameters "UnitX", "ColumnX" and "LineX" must be filled with the necessary information.

15.2.6 Deleting a variable from the project list

Delete variable

To delete a variable from the project list, proceed as follows:

1. Select the variable.
2. Perform "Delete in the project" via the context menu.
   The variable is deleted from the list.

This action can also be undone with the "Edit" menu item.

Note

Deleting of variables results in a change of the absolute addresses of the pointer structures to the variables. When changing the variable selection, it is absolutely necessary to **create** one or more **text files of all user blocks prior to the change**. This is the only way to ensure that the assignment of the variables in FB "GET" or FB "PUT" remains correct, even after recompilation.

15.2.7 Alias name

Description

The variable names provided can be up to 32 characters in length. To make variables unique in the data block to be generated, several ASCII characters are added to the selected name. However, the STEP 7 compiler recognizes maximum 24 ASCII characters as a unique S7 variable. Since it cannot be precluded that variable names can only be differentiated by the last 8 character positions, **ALIAS** names are used for names that are too long. Consequently, the length of the S7 name to be used is checked when a variable is selected. If the name is longer than 24 characters, the user must enter an additional name, which is then used as the alias name.

In this case, the user must ensure that the alias name is unique.

Configuring alias input

1. Activate the adaptation of the alias name for the transfer of the variables to the project list:
   - "Specify the alias name only when the name is longer than 24 characters"; adapt the alias name only when necessary.
   - "Always specify the alias name"; this allows the alias name always to be adapted.
2. An alias name can then be entered every time a variable is transferred.

Editing alias names

To edit the alias name subsequently, perform "Change alias name in the project" in the context menu of the project list. This action can also be undone with the "Edit" menu item.

15.2.8 Generating code

Generating STL code

To use the block in STEP 7, the project list must be converted to STL. To do this, generate the STL code with the following settings.

Perform settings

Enter in the "Code - Settings" menu item, the DB number and the symbol for the DB number for which the code is generated.

- Select in the "Measuring system" tab how the measuring system variables are processed in the PLC.
- The creation for the associated target system is specified in the "Generate" tab.

Perform generation

Generate the STEP 7 file from the selected variable list with extension *.awl:

1. Save the project list as project (*.var).
2. Perform "Code - Generate". The "Save as" dialog opens.
3. Enter a name for the file and specify the path for the *.awl file to be created.

The *.awl file is stored with this name.

15.2.9 Transfer to STEP 7

Transferring a project list to STEP 7

The generated AWL file is transferred to a selectable SIMATIC project (program path) and compiled. Furthermore, the symbol can also be transferred. This function is available as of STEP 7 Version 5.1. Before transferring a new AWL file, the file window of the AWL file must be closed in the LAD/FBD/STL editor.

1. First save the project list.
2. Perform "Code - In STEP 7 project".
   The "STEP 7 projects" dialog opens.

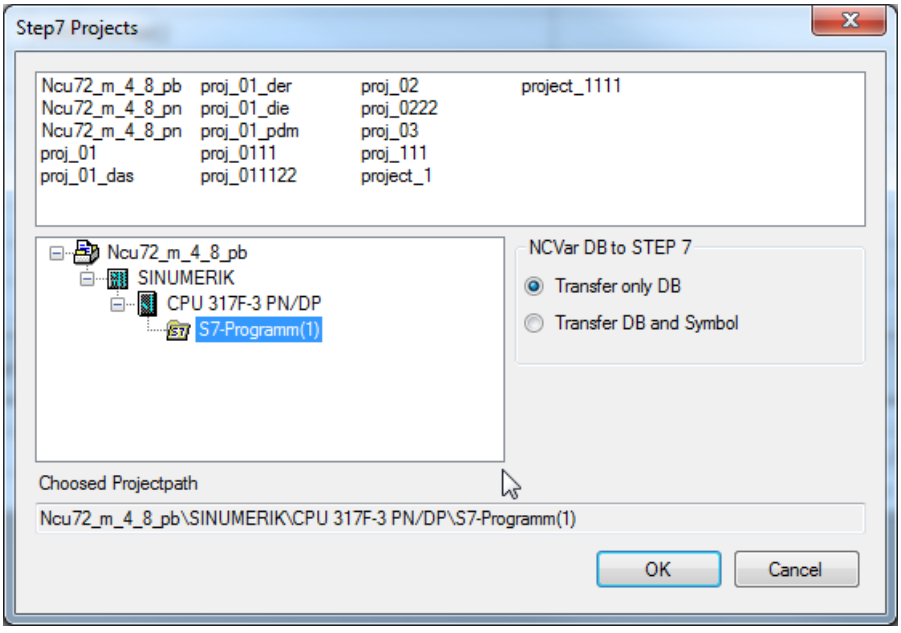


Figure 15-5 Transfer to STEP 7

3. Select the STEP 7 project that contains your program.
4. Activate the "Transfer only DB" checkbox when only the data block should be transferred.
5. Activate the "Transfer DB and Symbol" checkbox when not only the data block, but also symbols should be transferred. Symbols allow the symbolic addressing of variables.
6. Click "OK" to start the operation.
   SIMATIC Manager opens and the block is stored at "Sources".
   The data block can be displayed in the "LAD/FBD/STL" program editor.
7. Double-click the data block at "Blocks" in SIMATIC Manager, e.g. DB120.
   The program editor opens and displays the variables.

Importing a generated AWL file into STEP 7

As alternative to the method described above, the stored STL source can be inserted in a project in SIMATIC Manager.

1. Open the project in SIMATIC Manager and navigate to the "Sources" folder.
2. Perform "Insert - External source".
   The "Insert external source" dialog opens.
3. Select the AWL file and click "Open".
   The file is added to the sources of the project.
4. Compile the file so it can be used as block. The block is then displayed in the "Blocks" folder.

15.2.10 Using TIA Portal

STL source for TIA Portal

If TIA Portal is used rather than SIMATIC Manager, the "in STEP 7 project" function can be used.

The procedure for TIA Portal follows:

1. Specify the project list.
2. Save the list.
3. Generate an AWL file.
4. Import the AWL file into TIA Portal.

For more information on this topic, refer to the online help of TIA Portal.

15.2.11 Using SINAMICS parameters

SINAMICS parameters

In addition to the "NCData" tab and the variable list of the NC variables, a list of SINAMICS parameters can also be shown.

The list contains the parameters of the "Servo" drive object:

- r parameters: display parameters with which values of the drive axis can be displayed.
- p parameters: read and write parameters with which values can be transferred to the drive axis.

Similar to NC variables, SINAMICS parameters can also be added to the project list and transferred to STEP 7.

Selecting parameters

To select drive parameters, proceed as follows:

1. Search for the variables to be added to the project list.
2. Double-click the variable.
   - OR -
3. Perform "Transfer NC variables" via the context menu.
   The "Input Area No., Row, Column" dialog opens.

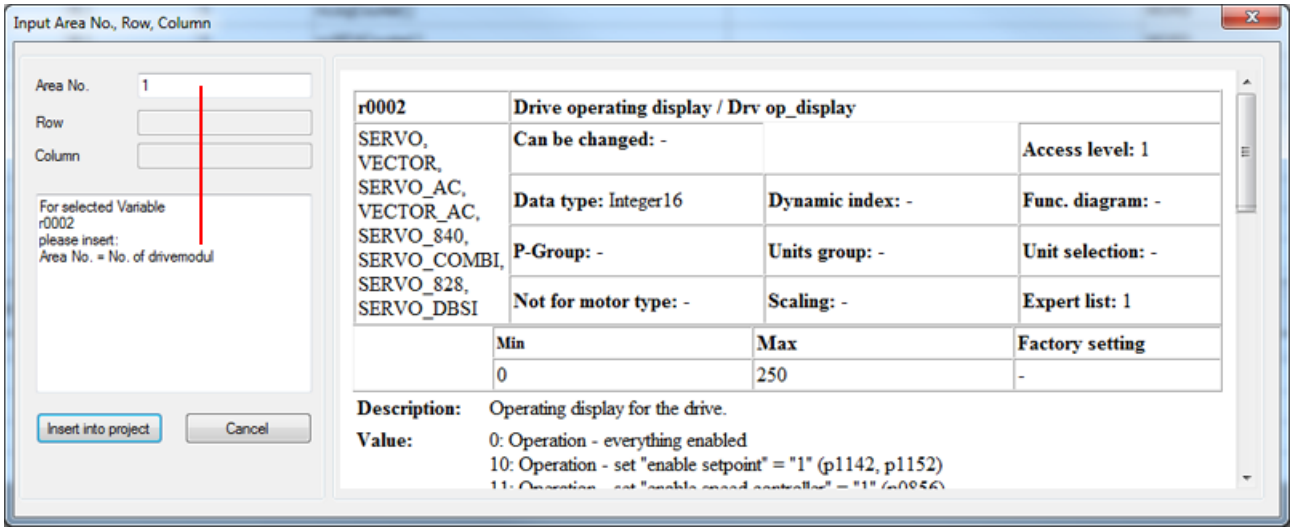


Figure 15-6 Drive parameters

4. Enter the number of the drive module that corresponds to the deployed drive.

   For drives, the area number is the number of the drive module. This corresponds to the component number.

   The component number can be found in the topology view (SINUMERIK Operate).

15.2.12 Importing variables and parameters

Importing variable and parameter database files

For older versions of NC VAR selector, the variable lists of an NC variant exist as ncv*.mdb files. When the lists are opened, all available NC variables / SINAMICS parameters are displayed in the variable list in separate tabs.

The following ncv*.mdb variable list is available:

Variables	List
NC variables including machine and setting data:	ncv_NcData.mdb
Parameters of the drive:	ncv_SinamicsServo.mdb

1. Perform "Project - Open database".
   The "Select NC Variable List" dialog opens.
2. Select the appropriate file (ncv*.mdb) and confirm with "Open".
3. The list opens in NC VAR selector.

15.2.13 Startup, installation

The "NC VAR selector" Windows application is installed by the supplied SETUP program.

The system prerequisites depend on the deployed SIMATIC Manager.

Block descriptions 16

16.1 FB1: RUN_UP - basic program, start section

Function

The synchronization of NC and PLC is performed during start-up. The data blocks for the NC/PLC user interface are created based on the NC configuration defined via machine data, and the basic program parameters are verified for plausibility. In the event of an error, FB1 passes an error identifier to the diagnostic buffer and switches the PLC to the stop state.

"Restart" startup mode

The integrated PLC only supports the startup mode "Restart". After the basic system initialization, the organization block OB100 "Restart" is always run through first, followed by OB1 "Cyclic mode".

Input parameters

Only the respectively relevant parameters of function block FB1 must be described with user-specific values for parameterization of the basic program. The preset values in the instance data block DB7 of the FB1 do not need to be assigned. The function block FB1 must only be called in the organization block OB100.

Output parameters

The output parameters of function block FB1 can also be read from the cyclical part of the basic program. Two options are available for this purpose:

1. Direct access to the instance data block DB7 of FB1 in symbolic form.
   Example: "L gp_par.MaxChan", with "gp_par" as the symbolic name of DB7
2. A bit memory is assigned to an output parameter when FB1 is parameterized. The bit memory is then read in the basic program in order to determine the value of the output parameter.
   Example: "MaxChan":= MW 20

Note

For assigning the FB1 parameters for MCP and HHU, see "Configuration machine control panel, handheld unit, direct keys (Page 91) ".

Declaration of the function

```
FUNCTION_BLOCK FB1
VAR_INPUT
```

```
MCPNum:           INT:=1;             // 0: No MCP
                                      // 1: 1 MCP (default)
                                      // 2: 2 MCPs
MCP1In:           POINTER;            // Start addr. MCP1 input signals
MCP1Out:          POINTER;            // Start addr. MCP1 output signals
MCP1StatSend:     POINTER;            // Status DW for sending MCP1
MCP1StatRec:      POINTER;            // Status DW for receiving MCP1
MCP1BusAdr:       INT:=6;             // default
MCP1Timeout:      S5TIME:= S5T#700MS;
MCP1Cycl:         S5TIME:= S5T#200MS;
MCP2In:           POINTER;            // Start addr. MCP2 input signals
MCP2Out:          POINTER;            // Start addr. MCP2 output signals
MCP2StatSend:     POINTER;            // Status DW for sending MCP2
MCP2StatRec:      POINTER;            // Status DW for receiving MCP2
MCP2BusAdr:       INT;
MCP2Timeout:      S5TIME:= S5T#700MS;
MCP2Cycl:         S5TIME:= S5T#200MS;
MCPMPI:           BOOL:= FALSE;
MCP1Stop:         BOOL:= FALSE;
MCP2Stop:         BOOL:= FALSE;
MCP1NotSend:      BOOL:= FALSE;
MCP2NotSend:      BOOL:= FALSE;
MCPSDB210:        BOOL:= FALSE;
MCPCopyDB77:      BOOL:= FALSE;
MCPBusType:       BYTE=B#16#0;
HHU:              INT:=0;             // Handheld unit interface
                                      // 0: No HHU
                                      // 1: HHU on MPI
                                      // 2: HHU on OPI
BHGIn :           POINTER;            // Transmit data of the handheld
                                      unit
BHGOut :          POINTER;            // Receive data of the handheld
                                      unit
BHGStatSend :     POINTER;            // Status DW for sending HHU
BHGStatRec :      POINTER;            // Status DW for receiving HHU
BHGInLen :        BYTE:= B#16#6;      // Input 6 bytes
BHGOutLen :       BYTE:= B#16#14;     // Output 20 bytes
BHGTimeout :      S5TIME:= S5T#700MS;
BHGCycl :         S5TIME:= S5T#100MS;
BHGRecGDNo :      INT:=2;
BHGRecGBZNo :     INT:=2;
BHGRecObjNo :     INT:=1;
BHGSendGDNo :     INT:=2;
BHGSendGBZNo :    INT:=1;
BHGSendObjNo :    INT:=1;
BHGMPI :          BOOL:= FALSE;
```

```
   BHGStop :          BOOL:= FALSE;
   BHGNotSend :       BOOL:= FALSE;
   NCCyclTimeout :    S5TIME:= S5T#200MS;
   NCRunupTimeout :   S5TIME:= S5T#50S;
   ListMDecGrp :      INT:=0;
   NCKomm:            BOOL:= FALSE;
   MMCToIF:           BOOL:=TRUE;
   HWheelMMC:         BOOL:=TRUE;          // Handwheel selection via HMI
   ExtendAlMsg :      BOOL;
   MCP_IF_TCS:        BOOL;
   ExtendChanAxMsg:   BOOL;
   MsgUser:           INT:=10;             // Number of user areas in DB 2
   UserIR:            BOOL:= FALSE;        // User programs in OB40,
                                           // Observe local data expansion!
   IRAuxfuT:          BOOL:= FALSE;        // Evaluate T function in OB40
   IRAuxfuH:          BOOL:= FALSE;        // Evaluate H function in OB40
   IRAuxfuE:          BOOL:= FALSE;        // Evaluate DL function in OB40
   UserVersion:       POINTER;             // Pointer to string variable,
                                           which is displayed in the version
                                           display of the user interface
   OpKeyNum :         INT;
   Op1KeyIn :         POINTER;
   Op1KeyOut :        POINTER;
   Op1KeyBusAdr :     INT;
   Op2KeyIn :         POINTER;
   Op2KeyOut :        POINTER;
   Op2KeyBusAdr :     INT;
   Op1KeyStop :       BOOL;
   Op2KeyStop :       BOOL;
   Op1KeyNotSend :    BOOL;
   Op2KeyNotSend :    BOOL;
   OpKeyBusType :     BYTE ;
   IdentMcpBusAdr :   INT;
   IdentMcpProfilNo   BYTE ;
   :
   IdentMcpBusType :  BYTE ;
   IdentMcpStrobe :   BOOL;
END_VAR

VAR_OUTPUT
   MaxBAG:            INT;
   MaxChan:           INT;
   MaxAxis:           INT;
   ActivChan:         ARRAY[1..10] OF
                      BOOL;
```

```
    ActivAxis:          ARRAY[1..31] OF
                        BOOL;
    UDInt :             INT;
    UDHex:              INT;
    UDReal :            INT;
    IdentMcpType :      BYTE ;
    IdentMcpLengthIn    BYTE ;
    :
    IdentMcpLengthOut   BYTE ;
     :
END_VAR
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
MCPNum:	I	INT	0, 1, 2	Number of active MCPs 0: No MCPs available
MCP1In: MCP2In:	I	POINTER	E0.0 ... E120.0 or M0.0 ... M248.0 or DBn DBX0.0 ... DBXm.0	Start address for the input signals of the relevant machine control panel
MCP1Out: MCP2Out:	I	POINTER	A0.0 ... A120.0 or M0.0 ... M248.0 or DBn DBX0.0 ... DBXm.0	Start address for the output signals of the relevant machine control panel
MCP1StatSend: MCP2StatSend:	I	POINTER	A0.0 ... A124.0 or M0.0 ... M252.0 or DBn DBX0.0 ... DBXm.0	Only for Ethernet MCP: Switch off flashing (see Section "Switch-over of machine control panel, handheld unit (Page 100)")
MCP1StatRec: MCP2StatRec:	I	POINTER	A0.0 ... A124.0 or M0.0 ... M252.0 or DBn DBX0.0 ... DBXm.0	Currently no significance
MCP1BusAdr:	I	INT	1, 2, 3 ... 126	DP slave: PROFIBUS address
MCP2BusAdr:			192, 193, 194 .. 223	Ethernet MCP: DIP setting
MCP1Timeout: MCP2Timeout:	I	S5time	Recommendation: 700 ms	Cyclic sign-of-life monitoring for machine control panel
MCP1Cycl: MCP2Cycl:	I	S5time	Recommendation: 200 ms	Relevant only for PROFIBUS
MCPMPI:	I	BOOL	FALSE	Available owing to compatibility

Signal	Type	Type	Value range	Meaning
MCP1Stop: MCP2Stop:	I	BOOL	0 (FALSE), 1 (TRUE)	0: Start transfer of the machine control panel signals 1: Stop transfer of machine control panel signals
				DP slave: Slave deactivated
MCP1NotSend : MCP2NotSend:	I	BOOL	0 (FALSE), 1 (TRUE)	0: Send and receive operation activated 1: Receive machine control panel signals only
MCPSDB210:	I	BOOL	false	Available owing to compatibility
MCPCopyDB77:	I	BOOL	false	Available owing to compatibility
MCPBusType:	I	BYTE	3, 4, 5, 6	b#16#yx: • Bus type MCP1: Lower nibble (x) • Bus type MCP2: Upper nibble (y) 3: PROFIBUS 4: PROFIBUS on the MPI/DP port 5: Ethernet 6: PROFINET Mixed mode is possible (see Section "Configuration machine control panel, handheld unit, direct keys (Page 91)")
HHU:	I	INT	0, 5	Handheld unit interface: 0: No HHU 5: HHU on Ethernet
BHGIn:	I	POINTER	E0.0 ... E124.0 or M0.0 ... M252.0 or DBn DBX0.0 ... DBXm.0	Start address PLC receive data from handheld unit
BHGOut:	I	POINTER	A0.0 ... A124.0 or M0.0 ... M252.0 or DBn DBX0.0 ... DBXm.0	Start address PLC transmit data to handheld unit
BHGStatSend:	I	POINTER	A0.0 ... A124.0 or M0.0 ... M252.0 or DBn DBX0.0 ... DBXm.0	Available owing to compatibility
BHGStatRec:	I	POINTER	A0.0 ... A124.0 or M0.0 ... M252.0 or DBn DBX0.0 ... DBXm.0	Available owing to compatibility
BHGInLen:	I	BYTE	HHU default: B#16#6 (6 Byte)	Available owing to compatibility
BHGOutLen:	I	BYTE	HHU default: B#16#14 (20 byte)	Available owing to compatibility

Signal	Type	Type	Value range	Meaning
BHGTimeout:	I	S5time	Recommendation: 700 ms	Available owing to compatibility
BHGCycl:	I	S5time	Recommendation: 100 ms	Available owing to compatibility
BHGRecGDNo:	I	INT	HHU default: 2	Ethernet DIP switch
BHGRecGBZNo:	I	INT	HHU default: 2	Available owing to compatibility
BHGRecObjNo:	I	INT	HHU default: 1	Available owing to compatibility
BHGSendGDNo:	I	INT	HHU default: 2	Available owing to compatibility
BHGSendGBZNo:	I	Int	HHU default: 1	Available owing to compatibility
BHGSendObjNo:	I	INT	HHU default: 1	Available owing to compatibility
BHGMPI:	I	BOOL	0 (FALSE)	Available owing to compatibility
BHGStop:	I	BOOL	0 (FALSE), 1 (TRUE)	0: Start transmission of the handheld unit signals 1: Stop transmission of handheld unit signals
BHGNotSend:	I	BOOL	0 (FALSE), 1 (TRUE)	0: Send and receive operation activated 1: Receive handheld unit signals only
NCCyclTimeout:	I	S5time	Recommendation: 200 ms	Cyclic sign-of-life monitoring NC
NCRunupTimeout:	I	S5time	Recommendation: 50 s	Power-up monitoring NC
ListMDecGrp:	I	INT	0, 1, 2 ... 16	Activation of the expanded M group decoding 0: Not active 1...16: Number of M groups
NCKomm:	I	BOOL	0 (FALSE), 1 (TRUE)	PLC NC communication services FB2, 3, 4, 5, 7 1: Active
MMCToIF:	I	BOOL	0 (FALSE), 1 (TRUE)	Transmission of HMI signals to interface (modes, program control, etc.) 1: Active
HWheelMMC:	I	BOOL	0 (FALSE), 1 (TRUE)	0: Handwheel selection via HMI 1: Handwheel selection via user program
ExtendAlMsg :	I	BOOL	0 (FALSE), 1 (TRUE)	Activation of extensions, error and operational messages of the FC10 (see Section "Extensions of the PLC alarms via block FC10" in Section "Interface PLC/HMI (Page 62)")
MCP_IF_TCS	I	BOOL	0 (FALSE), 1 (TRUE)	FC19: Evaluation R11 key 0: No evaluation 1: R11 key acts as manual traversing in tool orientation (see Section "Cartesian manual travel" in Section "FC19: MCP_IFM - transfer of MCP signals to interface (Page 267)")
ExtendChanAxMsg	I	BOOL	0 (FALSE), 1 (TRUE)	Activation of all areas for error- and operational messages of the FC10 (see Section "FC10: AL_MSG - error and operating messages (Page 247)")

Signal	Type	Type	Value range	Meaning
MsgUser:	I	INT	0, 1, 2 ... 64	Number of user areas (DB2) – depends on parameter "ExtendAlMsg"
UserIR:	I	BOOL	0 (FALSE), 1 (TRUE)	Local data extension OB40 required for processing of signals from the user
IRAuxfuT:	I	BOOL	0 (FALSE), 1 (TRUE)	Evaluate T function in OB40
IRAuxfuH:	I	BOOL	0 (FALSE), 1 (TRUE)	Evaluate H function in OB40
IRAuxfuE:	I	BOOL	0 (FALSE), 1 (TRUE)	Evaluate DL function in OB40
UserVersion:	I	POINTER	P#DBn.DBXx.0	Pointer to string variable, which is displayed in the version display of the user interface The string variable has the following notation (max. 41 characters): "**<Name>** <version **xx.xx.xx**> <date **yy/mm/dd>**" Example: "Test version 07.06.02 13/06/04"
OpKeyNum :	I	INT	0, 1, 2	Number of active direct key modules 0: No Ethernet direct control keys available
Op1KeyIn: Op2KeyIn:	I	POINTER	P#Ex.0 or P#Mx.0 or P#DBn.DBXx.0.	Start address for the input signals of the affected direct control key modules
Op1KeyOut : Op2KeyOut:	I	POINTER	P#Ax.0 or P#Mx.0 or P#DBn.DBXx.0.	Start address for the output signals of the affected direct control key modules
Op1KeyBusAdr : Op2KeyBusAdr:	I	INT	1, 2, 3 ... 191	Direct control keys via Ethernet: TCU index
Op1KeyStop : Op2KeyStop:	I	BOOL	0 (FALSE), 1 (TRUE)	0: Start transmission of the direct key signals 1: Stop transmission of the direct key signals
Op1KeyNotSend: Op2KeyNotSend:	I	BOOL	0 (FALSE), 1 (TRUE)	0: Send and receive operation activated 1: Receive direct control key signals only
OpKeyBusType :	I	BYTE	b#16#55	Ethernet
IdentMcpBusAdr :	I	INT	1, 2, 3 ... 254	Only IE devices
IdentMcpProfilNo :	I	BYTE	0, 1	Profile of a device 0: Complete device 1: Only direct keys
IdentMcpBusType:	I	BYTE	b#16#5	Only IE devices
IdentMcpStrobe:	I	BOOL	0 (FALSE), 1 (TRUE)	1: Activate query
MaxBAG:	O	INT	1, 2, 3 ... 10	Number of mode groups
MaxChan:	O	INT	1, 2, 3 ... 10	Number of channels
MaxAxis:	O	INT	1, 2, 3 ... 31	Number of axes

Signal	Type	Type	Value range	Meaning
ActivChan:	O	ARRAY[1...10] OF BOOL	0 (FALSE), 1 (TRUE)	Bit string for active channels
ActivAxis:	O	ARRAY [1..31] OF BOOL	0 (FALSE), 1 (TRUE)	Bit string for active axes
UDInt :	O	INT	0,	Quantity of INTEGER machine data in DB20
UDHex:	O	INT	---	Quantity of hexadecimal machine data in DB2, where applicable including the filler byte
UDReal :	O	INT	---	Quantity of REAL machine data in DB20
IdentMcpType :	O	BYTE	0, B#16#80, B#16#81, B#16#82 ... B#16#89	MCP type
IdentMcpLengthIn:	O	BYTE	---	Length of MCP input data (MCP → PLC)
IdentMcpLengthOut :	O	BYTE	---	Length of MCP output data (PLC → MCP)

If an error occurs during communication with a machine control panel (MCP) or handheld unit (HHU), the following alarm is displayed at the HMI and the input signals (MCP1In, MCP2In or BHGIn) are set to the zero value:

Error case: MCP/HHU

- Alarm 400260: "MCP 1 failure"
- Alarm 400261: "MCP 2 failure"
- Alarm 400262: "HHU failure"

If a resynchronization is possible between PLC and MCP/HHU, communication is resumed, the error message at the HMI is deleted by the basic program, and process values transferred into the input signals (MCP1In, MCP2In or BHGIn) once again.

Example: FB1 call in the OB100

```
ORGANIZATION_BLOCK OB100
VAR_TEMP
    OB100_EV_CLASS :              BYTE ;
    OB100_STRTUP :                BYTE ;
    OB100_PRIORITY :              BYTE ;
    OB100_OB_NUMBR :              BYTE ;
    OB100_RESERVED_1 :            BYTE ;
    OB100_RESERVED_2 :            BYTE ;
    OB100_STOP :                  WORD;
    OB100_RESERVED_3 :            WORD;
    OB100_RESERVED_4 :            WORD;
    OB100_DATE_TIME :             DATE_AND_TIME;
END_VAR
BEGIN
```

```
    CALL FB1, DB7(                  // FB1 call, instance DB: DB7
          MCPNum :=                 1,
          MCP1In :=                 P#E0.0,
          MCP1Out :=                P#A0.0,
          MCP1StatSend :=           P#A8.0,
          MCP1StatRec :=            P#A12.0,
          MCP1BusAdr :=             6,
          MCP1Timeout :=            S5T#700MS,
          MCP1Cycl :=               S5T#200MS,
          NC-CyclTimeout :=         S5T#200MS,
          NC-RunupTimeout :=        S5T#50S);
// INSERT USER PROGRAM HERE
END_ORGANIZATION_BLOCK
```

16.2 FB2: GET - read NC variable

Function

The FB2 "GET" function block is used to read variables from the NC area.

In order to reference the NC variables, they are first selected with the "NC VAR selector" tool and generated as STL source in a data block. A name must then be assigned to this data block in the S7 symbol table. When calling FB2, the variable addresses are transferred in the following form: Parameter "Addr1" to "Addr8" = "<DB name>.<S7 name>"

Request for reading NC variables

Call of FB2 with positive edge change, parameter "Req" = 0 → 1

S7 names of the NC variables: Parameter "Addr1" to "Addr8" = "NCVAR".<S7 name>"

Pointer for writing the variable values: Parameters "RD1" to "RD8" = "P#<Address>"

Completion of the read request

Read request successfully completed: Parameter "Done" == 1.

Read request terminated with error: Parameter "Error" == 1, error cause in parameter "State"

Requirements

- Release of the NC/PLC communication by OB100, FB1 parameter "NCKomm" = 1
- For the data block DB120 (data interface), the S7 Symbol Editor must be used to assign a symbol (default: NCVAR) in the S7 symbol table of the S7 project. Via this icon you then specify the NC variable e.g. "ADDR1: = "NCVAR".<NC-Variable>" in the FB2 parameters "Addr<x>"

Boundary conditions

- FB2 has multi-instance capability.
- A separate instance or separate instance DB must be assigned from the user area to each call of the FB2.
- When **channel-specific** variables are read, only variables from exactly **one** channel may be addressed via "Addr1" to "Addr8" if FB2 is called.
- When **drive-specific** variables are read, only variables from exactly **one** SERVO drive object may be addressed via "Addr1" to "Addr8" if FB2 is called. The SERVO drive object must be assigned to a machine axis of the NC. The line index corresponds to the logical drive number.
- In a read job, only variables from the same area, channel, or drive object can be read.

Note

Error case
When reading variables from different channels or drive objects, or simultaneously from a channel and a drive object, an error message is output:

- "Error" == TRUE
- "State" == W#16#02

Variable addressing

For some NC variables, it is necessary to select "Area no." and/or "Line" or "Column" from the NC VAR selector. It is possible to select a basic type, i.e. "Area no.", "Line" and "Column" are preassigned "0". The values of the "Area no.", "Line" and "Column" specified by the NC VAR selector are checked for a "0" in FB2. If an NC-VAR selector value == "0", the corresponding value of the FB2 parameter is adopted. To do this, the FB2 parameters "Unit<x>", "Column<x>" and "Line<x>", with <x> = 1 - 8, must be written before FB2 is called.

Table 16-1 Parameter match

FB2 parameter	NC VAR selector
Unit	Area no.
Column	Column
Line	Line

Variables within **one** group can be combined in a job:

Group	Area				
1	C[1]	N	B	A	T
2	C[2]	N	B	A	T
3	V[.]	H[.]	---	---	---
The same rules apply for channels 3 to 10 as for group 1 and group 2 shown in the example.					

Note

The number of usable variables can be less than eight when simultaneously reading several variables of the "String" type.

Declaration of the function

```
FUNCTION_BLOCK FB2
VAR_INPUT
    Req :                BOOL;
    NumVar :             INT;
    Addr1 :              ANY;
    Unit1 :              BYTE ;
    Column1 :            WORD;
    Line1 :              WORD;
    Addr2 :              ANY;
    Unit2 :              BYTE ;
    Column2 :            WORD;
    Line2 :              WORD;
    Addr3 :              ANY;
    Unit3 :              BYTE ;
    Column3 :            WORD;
    Line3 :              WORD;
    Addr4 :              ANY;
    Unit4 :              BYTE ;
    Column4 :            WORD;
    Line4 :              WORD;
    Addr5 :              ANY;
    Unit5 :              BYTE ;
    Column5 :            WORD;
    Line5 :              WORD;
    Addr6 :              ANY;
    Unit6 :              BYTE ;
    Column6 :            WORD;
    Line6 :              WORD;
    Addr7 :              ANY;
    Unit7 :              BYTE ;
    Column7 :            WORD;
    Line7 :              WORD;
    Addr8 :              ANY;
    Unit8 :              BYTE ;
    Column8 :            WORD;
    Line8 :              WORD;
END_VAR
```

```
VAR_OUTPUT
   Error :              BOOL;
   NDR :                BOOL;
   State :              WORD;
END_VAR


VAR_IN_OUT
   RD1 :                ANY;
   RD2 :                ANY;
   RD3 :                ANY;
   RD4 :                ANY;
   RD5 :                ANY;
   RD6 :                ANY;
   RD7 :                ANY;
   RD8 :                ANY;
END_VAR
```

Description of formal parameters

Parameter	Type	Type	Value range	Meaning
Req:	I	BOOL	---	Job start with positive signal edge
NumVar:	I	INT	1 ... 8	Number of variables to be read: "Addr1" - "Addr8"
Addr1 - Addr8:	I	ANY	"DBName".<VarName>	Variable identifiers from **NC Var selector**
Unit1 - Unit8:	I	BYTE	---	Area address, optional for variable addressing
Column1 - Column8:	I	WORD	---	Column address, optional for variable addressing
Line1 - Line8:	I	WORD	---	Line address, optional for variable addressing
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1: Negative acknowledgement of job or execution of job impossible
NDR :	O	BOOL	0 (FALSE), 1 (TRUE)	1: Job successfully executed. Data is available
State:	O	WORD	---	See paragraph "Error identifiers"
RD1 - RD8:	I/O	ANY	P#Mm.n BYTE x ... P#DBnr.dbxm.n BYTE x	Target area for read data

Error identifiers

NC variables

State		Meaning	Note
High byte [1]	Low byte		
1 - 8	1	Access error	---
0	2	Error in job	Incorrect compilation of variables in a job
0	3	Negative acknowledgement, job not executable	Internal error, possible remedy: • Check job data • NC reset
1 - 8	4	Insufficient local user memory available	Data type of the read variable is greater than specified in "RD1" - "RD8"
0	5	Format conversion error	Error on conversion of var. type double: Variable is not within the S7 REAL area
0	6	FIFO full	Job must be repeated since queue is full
0	7	Option not set	FB1 parameter "NCKomm" is not set
1 - 8	8	Incorrect target area (RD)	"RD1" - "RD8" must not be local data
0	9	Transmission occupied	Job must be repeated
1 - 8	10	Error in variable addressing	"Unit" or "Column"/"Line" contains value 0
0	11	Address of variable invalid	Check "Addr" (or variable name), "Area", "Unit"
0	12	NumVar == 0	Check parameter "NumVar"
1 - 8	13	ANY data reference incorrect	Requested "NcVar" data has not been parameterized
1) High byte > 0 ⇒ Number of the variable in which the error occurred			

Drive-specific variables

If an error occurs while reading/writing a drive-specific variable (DB1200.DBX3000.1 == 1), an error number is displayed in the access result which is based on the error numbers defined in the PROFIdrive profile.

State	Meaning
x	<Error number of PROFIdrive profile> + 20_H or 36_D

Determining the significance of the access result:

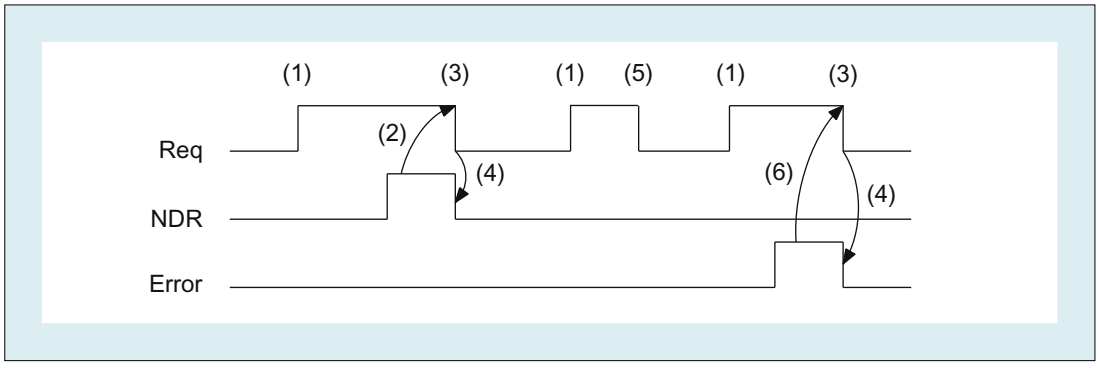
1. Calculation of the error number of the PROFIdrive profile
   <Error number of PROFIdrive profile> = result of access - 20_H or 36_D.
2. Determining the significance of the error number of the PROFIdrive profile
   The error numbers of the PROFIdrive profile are described in:
   Further information
   Function Manual, SINAMICS S120 Drive Functions; Section "Communication" > "Communication according to PROFIdrive" > "Acyclic communication" > "Structure of the requests and responses" > Subsection "Error values in parameter responses"

Configuration steps

Proceed as follows to read NC variables:

- Select variables with the NC VAR selector
- Save selected variables in a *.VAR file
- Generate a STEP 7 *.STL source file
- Generate a DB with the associated address data
- Enter the symbol for the generated DB in the symbol table so that it is possible to access the address parameters symbolically in the user program
- Parameterizing the FB2

Pulse diagram



① User: Set request, Req = 0 → 1

② FB2 successfully completed, NDR = 1
User: Reset request, IF NDR == 1 THEN Req = 0

③ User: IF NDR == 1 THEN reset request: 1 → 0

④ FB2 reset job confirmation, NDR = 0

⑤ User: IF NDR == 0 AND Error == 0 THEN reset request Req = 1 → 0 **not permissible**

⑥ FB2 completed with errors, Error = 1
User: Reset request, IF NDR == 1 OR Error == 1 THEN Req = 0, possible further error handling

Call example

Reading three channel-specific machine data from channel 1, whose address specifications are stored in DB120.

Specification of data

The data is selected with the NC VAR selector and saved in the DB120.**VAR** file. Then the DB120.**AWL** file is created from this.

S7 (ALIAS) names are selected.

To adopt the channel designation into the variable name and to delete the characters "[" and "]", which are not permitted in a STEP 7 symbol, new S7 names are selected:

Area	Block	Name	Type	No.	Byte	S7 Name
C[1]	M	MD20070 $MC_AXCONF_MA-CHAX_USED[1]	CHAR	20070	1	C1AxConfMachAxUsed1
C[1]	M	MD20070 $MC_AXCONF_MA-CHAX_USED[2]	CHAR	20070	1	C1AxConfMachAxUsed2
C[1]	M	MD20090 $MC_SPIND_DEF_MAS-TER_SPIND	INT	20090	1	C1SpindDefMasterSpind

S7 symbol table

"NCVAR" is entered in the S7 symbol table as a symbolic name for the data block DB120:

Symbol	Operand	Data type
NCVAR	DB120	DB120

File DB120.AWL must be compiled and transferred to the PLC.

Parameterization of FB2 with instance DB110:

```
DATA_BLOCK DB110         // Unassigned user DB, as instance for FB2
FB2
BEGIN
END_DATA_BLOCK
Function FC "VariablenCall" : VOID
   U      I 7.7;                    // Unassigned machine control panel key
   S      M 100.0;                  // Activate req.
   U      M 100.1;                  // NDR completed message
   R      M 100.0;                  // Terminate job
   U      I 7.6;                    // Manual error acknowledgment
   U      M 102.0;                  // Error pending
   R      M 100.0;                  // Terminate job
   CALL FB2, DB110(
      Req :           M 100.0,
      NumVar :        3,           // Read three variables
      Addr1 :         "NCVAR".C1AxConfMachAxUsed1,
      Addr2 :         "NCVAR".C1AxConfMachAxUsed2,
      Addr3 :         "NCVAR".C1SpindDefMasterSpind,
      Error :         M102.0,
      NDR :           M100.1,
      State :         MW104,
      RD1 :           P#DB99.DBX0.0 BYTE 1,
      RD2 :           P#DB99.DBX1.0 BYTE 1,
      RD3 :           P#M110.0 INT 1);
```

Example: Variable addressing

Reading two R parameters from channel 1, whose address specifications are stored in DB120 as the basic type. The R parameter number is parameterized via parameter "Line<x>".

```
DATA_BLOCK DB120
VERSION : 0.0
STRUCT
   C1_RP_rpa0_0:
   STRUCT
   SYNTAX_ID :                BYTE := B#16#82;
   area_and_unit :            BYTE := B#16#41;
   column :                   WORD := W#16#1;
   line :                     WORD := W#16#0;
   block type :               BYTE := B#16#15;
   NO. OF LINES :             BYTE := B#16#1;
   type :                     BYTE := B#16#F;
   length :                   BYTE := B#16#8;
   END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK
CALL FB2, DB110(
   Req :=                              M 0.0,
   NumVar :=                           2,
   Addr1 :=                            "NCVAR".C1_RP_rpa0_0,
   Line1 :=                            W#16#1,
   Addr2 :=                            "NCVAR".C1_RP_rpa0_0,
   Line2 :=                            W#16#2,
   Error :=                            M 1.0,
   NDR :=                              M 1.1,
   State :=                            MW 2,
   RD1 :=                              P#M 4.0 REAL 1,
   RD2 :=                              P#M 24.0 REAL 1);
```

Classification of data types

Table 16-2 Classification of data types

NC-internal or BTSS data type	S7 data type
DOUBLE	REAL
DOUBLE	REAL2
REAL	REAL
LONG	DINT
INTEGER	DINT
UINT_32	DWORD

NC-internal or BTSS data type	S7 data type
INT_16	INT
UINT_16	WORD
UNSIGNED	WORD
CHAR	CHAR or BYTE
STRING	STRING
BOOL	BOOL
DATETIME	DATE_AND_TIME

Example

For example, to be able to read an NC variable of the type DOUBLE without adapting the format, an ANY pointer with REAL2 type must be specified in the destination area "RDx" (e.g.: P#M100.0 REAL2). If the basic PLC program identifies a variable of the type DOUBLE as REAL2 target type when reading, the data is adopted as a 64 bit floating point number in the PLC data area.

16.3 FB3: PUT - write NC variables

Function

The FB3 "PUT" function block is used to write variables from the NC area.

In order to reference the NC variables, they are first selected with the "NC VAR selector" tool and generated as STL source in a data block. A name must then be assigned to this data block in the S7 symbol table. When calling FB3, the variable addresses are transferred in the following form: Parameter "Addr1" to "Addr8" = "<DB name>".<S7 name>

Request for writing NC variables

Call of FB3 with positive edge change, parameter "Req" = 0 → 1

S7 names of the NC variables: Parameter "Addr1" to "Addr8" = "NCVAR".<S7 name>"

Pointer for writing the variable values: Parameters "RD1" to "RD8" = "P#<Address>"

Completion of the write request

Write request successfully completed: Parameter "Done" == 1.

Write request terminated with error: Parameter "Error" == 1, error cause in parameter "State"

Requirements

- Release of the NC/PLC communication by OB100, FB1 parameter "NCKomm" = 1
- For the data block DB120 (data interface), the S7 Symbol Editor must be used to assign a symbol (default: NCVAR) in the S7 symbol table of the S7 project. Via this symbol, the NC variable is then specified in the FB3 parameters "Addr<x>", e.g. "ADDR1":= "NCVAR".<NC-Variable>"

Boundary conditions

- FB3 has multi-instance capability.
- A separate instance or a separate instance DB must be assigned from the user area to each call of the FB3.
- In order to define machine data and GUD without a password, the protection level of the data you want to access must be redefined to the lowest level.
  Further information
  - Commissioning Manual; Section: "Protection levels concept"
  - Programming Manual NC Programming; Section: "Define protection levels for user data"
- When **channel-specific** variables are written, only variables from exactly **one** channel may be addressed via "Addr1" to "Addr8" if FB2 is called.
- When **drive-specific** variables are written, only variables from exactly **one** SERVO drive object may be addressed via "Addr1" to "Addr8" if FB2 is called. The SERVO drive object must be assigned to a machine axis of the NC. The line index corresponds to the logical drive number.
- In a write job, only variables from the same area, channel, or drive object can be written.

Note

Error case
When writing variables from different channels or drive objects, or simultaneously from a channel and a drive object, an error message is output:

- "Error" == TRUE
- "State" == W#16#02

Variable addressing

For some NC variables, it is necessary to select "Area no." and/or "Line" or "Column" from the NC VAR selector. It is possible to select a basic type, i.e. "Area no.", "Line" and "Column" are preassigned "0". The values of the "Area no.", "Line" and "Column" specified by the NC VAR selector are checked for a "0" in FB2. If an NC-VAR selector value == "0", the corresponding value of the FB3 parameter is adopted. To do this, the FB3 parameters "Unit<x>", "Column<x>" and "Line<x>", with <x> = 1 - 8, must be written before FB3 is called.

FB3 parameter	NC VAR selector
Unit<x>	Area no.
Column<x>	Column
Line<x>	Line

NC variables within **one** group can be combined in a job:

Group	Area				
1	C[1]	N	B	A	T
2	C[2]	N	B	A	T

Group	Area				
3	V[.]	H[.]	---	---	---
The same rules apply for channels 3 to 10 as for group 1 and group 2 shown in the example.					

Note

The number of usable variables can be less than eight when simultaneously writing several variables of the "String" type.

Declaration of the function

```
FUNCTION_BLOCK FB3
VAR_INPUT
```

```
    Req :               BOOL;
    NumVar :            INT;
    Addr1 :             ANY;
    Unit1 :             BYTE ;
    Column1 :           WORD;
    Line1 :             WORD;
    Addr2 :             ANY;
    Unit2 :             BYTE ;
    Column2 :           WORD;
    Line2 :             WORD;
    Addr3 :             ANY;
    Unit3 :             BYTE ;
    Column3 :           WORD;
    Line3 :             WORD;
    Addr4 :             ANY;
    Unit4 :             BYTE ;
    Column4 :           WORD;
    Line4 :             WORD;
    Addr5 :             ANY;
    Unit5 :             BYTE ;
    Column5 :           WORD;
    Line5 :             WORD;
    Addr6 :             ANY;
    Unit6 :             BYTE ;
    Column6 :           WORD;
    Line6 :             WORD;
    Addr7 :             ANY;
    Unit7 :             BYTE ;
    Column7 :           WORD;
    Line7 :             WORD;
    Addr8 :             ANY;
    Unit8 :             BYTE ;
    Column8 :           WORD;
    Line8 :             WORD;
END_VAR
VAR_OUTPUT
    Error :             BOOL;
    Done :              BOOL;
    State :             WORD;
END_VAR
VAR_IN_OUT
```

```
    SD1 :                ANY;
    SD2 :                ANY;
    SD3 :                ANY;
    SD4 :                ANY;
    SD5 :                ANY;
    SD6 :                ANY;
    SD7 :                ANY;
    SD8 :                ANY;
END_VAR
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
Req:	I	BOOL	-	Job start with positive signal edge
NumVar:	I	INT	1 ... 8	Number of variables to be written: "Addr1" - "Addr8"
Addr1 - Addr8:	I	ANY	"DBName".<VarName>	Variable identifiers from **NC Var selector**
Unit1 - Unit8:	I	BYTE	-	Area address, optional for variable addressing
Column1 - Column8:	I	WORD	-	Column address, optional for variable addressing
Line1 - Line8:	I	WORD	-	Line address, optional for variable addressing
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	Negative acknowledgment of job or execution of job impossible
Done:	O	BOOL	0 (FALSE), 1 (TRUE)	Job successfully executed.
State:	O	WORD	-	See paragraph "Error identifiers"
SD1 - SD8:	I/O	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	Data to be written

Error identifiers

NC variables

State		Meaning	Note
High byte [1]	Low byte		
1 - 8	1	Access error	---
0	2	Error in job	Incorrect compilation of variables in a job
0	3	Negative acknowledgement, job not executable	Internal error, try: • Check job • NC reset
1 - 8	4	Data areas or data types do not match or string is empty	Check data in "SD1" - "SD8"

State		Meaning	Note
High byte [1]	Low byte		
0	6	FIFO full	Job must be repeated since queue is full
0	7	Option not set	FB1 parameter "NCKomm" is not set
1 - 8	8	Incorrect target area (SD)	"SD1" - "SD8" must not be local data
0	9	Transmission occupied	Job must be repeated
1 - 8	10	Error in variable addressing	"Unit" or "Column"/"Line" contains value 0
0	11	Variable address invalid or variable is read-only	Check "Addr" (or variable name), "Area", "Unit"
0	12	NumVar == 0	Check parameter "NumVar"
1 - 8	13	ANY data reference incorrect	Requested "NcVar" data has not been parameterized
1 - 8	15	User data too long	Remedy: Write fewer variables per job or use shorter string variables
1) High byte > 0 ⇒ Number of the variable in which the error occurred			

Drive-specific variables

If an error occurs while reading/writing a drive-specific variable (DB1200.DBX3000.1 == 1), an error number is displayed in the access result which is based on the error numbers defined in the PROFIdrive profile.

State	Meaning
x	<Error number of PROFIdrive profile> + 20_H or 36_D

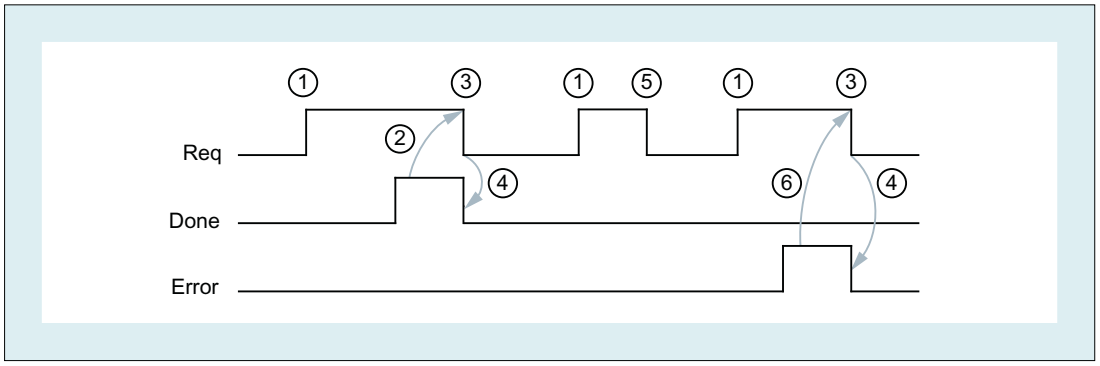
Determining the significance of the access result:

1. Calculation of the error number of the PROFIdrive profile
   <Error number of PROFIdrive profile> = result of access - 20_H or 36_D.
2. Determining the significance of the error number of the PROFIdrive profile
   The error numbers of the PROFIdrive profile are described in:
   Further information
   Function Manual, SINAMICS S120 Drive Functions; Section "Communication" > "Communication according to PROFIdrive" > "Acyclic communication" > "Structure of the requests and responses" > Subsection "Error values in parameter responses"

Configuration steps

To write NC variables, the same configuration steps are required as for reading NC variables. It is useful to store the address data of all NC variables to be read or written in a DB.

Pulse diagram



① User: Set request, Req = 0 → 1
② FB3 successfully completed, Done = 1
User: Reset request, IF Done == 1 THEN Req = 0
③ User: IF Done == 1 THEN reset request: 1 → 0
④ FB3 reset job confirmation, Done = 0
⑤ User: IF Done == 0 AND Error == 0 THEN reset request Req = 1 → 0 **not permissible**
⑥ FB3 completed with errors, Error = 1
User: Reset request, IF Done == 1 OR Error == 1 THEN Req = 0, possible further error handling

Call example

Writing of three channel-specific machine data items of channel 1:

Selection of the three data items with NC VAR selector and storage in file DB120.VAR

S7 (ALIAS) names are selected in order to adopt the block designation into the name and to remove the characters [ ], which are not permitted in a STEP 7 symbol.

Area	Block	Name	Type	Byte	S7 Name
C[1]	RP	rpa[5]	DOUBLE	4	rpa_5C1RP
C[1]	RP	rpa[11]	DOUBLE	4	rpa_11C1RP
C[1]	RP	rpa[14]	DOUBLE	4	rpa_14C1RP

Entry NCVAR for DB120 with the S7 SYMBOL Editor

Symbol	Operand	Data type
NCVAR	DB120	DB120

File DB120.AWL must be compiled and transferred to the PLC.

Calling and parameterizing the FB3 with instance DB111

```
DATA_BLOCK DB111        // Unassigned user DB, as instance for FB3
FB3
BEGIN
Function FC "VariablenCall" : VOID
```

```
END_DATA_BLOCK
   U      I 7.7;        // Unassigned machine control panel key
   S      M 100.0;      // Activate req.
   U      M 100.1;      // Done completed message
   R      M 100.0;      // Terminate job
   U      I 7.6;        // Manual error acknowledgment
   U      M 102.0;      // Error pending
   R      M 100.0;      // Terminate job
   CALL FB3, DB111(
      Req :           M 100.0,
      NumVar :        3,                      // Write three variables
      Addr1 :         "NCVAR".rpa_5C1RP,
      Addr2 :         "NCVAR".rpa_11C1RP,
      Addr3 :         "NCVAR".rpa_14C1RP,
      Error :         M102.0,
      Done :          M100.1,
      State :         MW104,
      SD1 :           P#DB99.DBX0.0 REAL 1,
      SD2 :           P#DB99.DBX4.0 REAL 1,
      SD3 :           P#M110.0 REAL 1);
```

Example: Variable addressing

Writing two R parameters from channel 1, whose address specifications are stored in DB120 as the basic type. The R parameter number is parameterized via parameter LineX.

```
DATA_BLOCK DB120
VERSION : 0.0
STRUCT
   C1_RP_rpa0_0:
   STRUCT
   SYNTAX_ID :             BYTE := B#16#82;
   area_and_unit :         BYTE := B#16#41;
   column :                WORD := W#16#1;
   line :                  WORD := W#16#0;
   block type :            BYTE := B#16#15;
   NO. OF LINES :          BYTE := B#16#1;
   type :                  BYTE := B#16#F;
   length :                BYTE := B#16#8;
   END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK
CALL FB3, DB122(
```

```
Req :=                  M 10.0,
NumVar :=               2,
Addr1 :=                "NCVAR".C1_RP_rpa0_0,
Line1 :=                W#16#1,
Addr2 :=                "NCVAR".C1_RP_rpa0_0,
Line3 :=                W#16#2
Error :=                M 11.0,
Done :=                 M 11.1,
State :=                MW 12,
SD1 :=                  P#M 4.0 REAL 1,
SD2 :=                  P#M 24.0 REAL 1);
```

Classification of data types

See table "Assignment of the data types" in Section "FB2: GET - read NC variable (Page 157)".

16.4 FB4: PI_SERV - request PI service

Function

The function block FB4 "PI_SERV" is used to start PI services.

The available PI services are described in the following chapters with their specific parameters. An overview of the available PI services can be found in: List of available PI services (Page 177)

Note

Due to the large number of "WVar" parameters, it is recommended that you use the function block FB7 instead of FB4. See Section "FB7: PI_SERV2 - request PI service (Page 210)".

Start of a PI service

Request to start a PI service: Call of the FB4 with positive edge, parameter "Req" = 0 → 1

Completion of a PI service

PI service successfully completed: Parameter "Done" == 1.

PI service completed with error: Parameter "Error" == 1, error cause in parameter "State"

Requirements

- Release of the NC/PLC communication by OB100, FB1 parameter "NCKomm" = 1
- For the data block DB16 (data interface of the PI services), the S7 Symbol Editor must be used to assign a symbol (default: PI) in the S7 symbol table of the S7 project. The requested PI service is then specified via this symbol in the FB4 parameter "PIService", e.g. "PIService:= "PI".<PI service>"

Boundary conditions

- A separate instance or a separate instance DB must be assigned from the user area to each call of the FB4.
- The start of a PI service (FB4 call with "Req" = 1) is only permitted in the cyclic part of the PLC basic program (OB1).
  If the PI service is not started (FB4 call with "Req" = 0), the parameters can also be written in the start-up part of the PLC basic program (OB100). The PI service can then be started using the already written parameters in the cyclic part of the PLC basic program (OB1) by calling FB4 with "Req" = 1.
- The execution of a PI service generally extends over several PLC cycles.

Declaration of the function

```
FUNCTION_BLOCK FB4
VAR_INPUT
    Req :               BOOL;
    PIService:          ANY;
    Unit:               INT;
    Addr1 :             ANY;
    Addr2 :             ANY;
    Addr3 :             ANY;
    Addr4 :             ANY;
    WVar1:              WORD;
    WVar2:              WORD;
    WVar3:              WORD;
    WVar4:              WORD;
    WVar5:              WORD;
    WVar6:              WORD;
    WVar7:              WORD;
    WVar8:              WORD;
    WVar9:              WORD;
    WVar10:             WORD;
END_VAR
VAR_OUTPUT
    Error :             BOOL;
    Done :              BOOL;
    State :             WORD;
END_VAR
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
Req:	I	BOOL	0 (FALSE), 1 (TRUE)	Job request
PIService:	I	ANY	"<DBName>".<PI service>	Requested PI service • <DBName>: symbol name for DB16, default: "PI" • <PI service>: List of available PI services (Page 177)
Unit:	I	INT	1, 2, 3, ...	Area number
Addr1 to Addr4:	I	ANY	"<DBName>".<VarName>	Reference to a string Significance depends on selected PI service
WVar1 to WVar10:	I	WORD	-32768 ... 32767 8000_H ... $7FFF_H$	INTEGER or WORD variable Significance depends on selected PI service
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	Error status 1: Negative acknowledgment of job or job could not be executed (see the "State" parameter for the cause of the error)
Done:	O	BOOL	0 (FALSE), 1 (TRUE)	Job status 1: Job successfully executed
State:	O	WORD	1)	Error code Only relevant if "Error" == 1

1) State	Meaning	Note
3	Negative acknowledgement, job not executable	Internal error, possibly remedied by an NC RESET
6	FIFO full	Repeat the command
7	Option not set	FB1 parameter "NCKomm" is not set
13	Addr1 ... Addr4: Reference invalid	Specify missing string
14	PI service is unknown	The PI service specified in the "PIService" parameter is unknown. → check the notation/spelling.
15	Addr1 ... Addr4: String too long	Check the string lengths

Call example

Function: Program selection in channel 1 (main program and workpiece program)

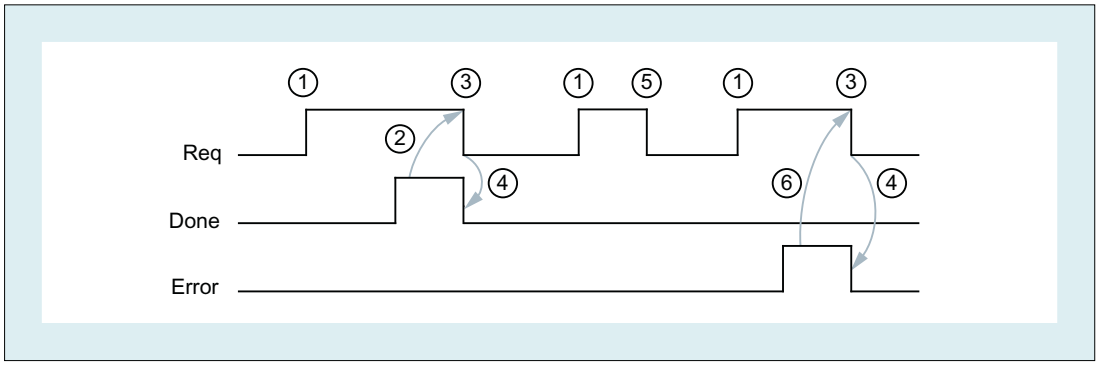
Entry of PI service for DB16 and STR for DB124 using the S7 SYMBOL editor:

Parameterization		
Symbol	Operand	Data type
PI	DB16	DB16
STR	DB124	DB124

```
DATA_BLOCK DB126         // Instance for FB4, unassigned user DB
FB4
BEGIN
END_DATA_BLOCK
// ---------------------------------------------------------------------------
DATA_BLOCK DB124
   STRUCT
      PName:        string[32]:= '_N_TEST_MPF   // Main program
                    ';
      Path:         string[32]:= '/             // Main program path
                    _N_MPF_DIR/';
      PName_WST:    string[32]:= '_N_ABC_MPF';  // Workpiece program
      Path_WST:     string[32]:= '/_N_WCS_DIR/  // Workpiece program path
                    _N_ZYL_WPD';
   END_STRUCT
BEGIN
END_DATA_BLOCK
//
---------------------------------------------------------------------------
Function FC "PICall": VOID
   U   I 7.7;        // Unassigned machine control panel key
   S   M 0.0;        // Activate req.
   U   M 1.1;        // Done completed message
   R   M 0.0;        // Terminate job
   U   I 7.6;        // Manual error acknowledgment
   U   M 1.0;        // Error pending
   R   M 0.0;        // Terminate job

   CALL FB4, DB126 (
      Req:=         M0.0,
      PIService:=   "PI".SELECT                // PI service: SELECT
      Unit:=        1,                         // Channel 1
      Addr1:=       "STR".Path,                // Main program: Path
      Addr2:=       "STR".PName,               // Main program: Program
   //  Addr1:=       "STR".Path_WST,            // Workpiece: Path
   //  Addr2:=       "STR".PName_WST,           // Workpiece: Program
      Error:=       M1.0,
      Done:=        M1.1,
      State:=       MW2
   );
```

Flow diagram



① User: Set request, Req = 0 → 1
② FB4: PI service successfully completed, Done = 1
User: Reset request, IF Done == 1 THEN Req = 0
③ User: IF Done == 1 THEN reset request: 1 → 0
④ FB4: Reset job confirmation, Done = 0
⑤ User: IF Done == 0 AND Error == 0 THEN reset request Req = 1 → 0 **not permissible**
⑥ FB4: PI service completed with errors, Error = 1
User: Reset request, IF Done == 1 OR Error == 1 THEN Req = 0, possible further error handling

16.4.1 List of available PI services

General PI services

PI service	Function
ASUP (Page 178)	Assign interrupt
CANCEL (Page 179)	Execute cancel
CONFIG (Page 180)	Reconfiguration of tagged machine data
DIGION (Page 180)	Digitizing on
DIGIOF (Page 180)	Digitizing off
FINDBL (Page 181)	Activate block search
LOGIN (Page 181)	Activate password
LOGOUT (Page 181)	Reset password
NCRES (Page 182)	Trigger NC-RESET
SELECT (Page 182)	Select program for processing for one channel
SETUDT (Page 183)	Sets the current user data to active
SETUFR (Page 183)	Activate user frame
RETRAC (Page 183)	Retraction of the tool in the tool direction

PI services of tool management

PI service	Function
CRCEDN (Page 184)	Create a tool cutting edge with specification of the T number
CREACE (Page 185)	Create a tool cutting edge with the next higher/free D number
CREATO (Page 185)	Create a tool with specification of a T number.
DELECE (Page 186)	Delete a tool cutting edge
DELETO (Page 186)	Delete tool
MMCSEM (Page 186)	Semaphores for various PI services
TMCRTO (Page 188)	Create a tool with specification of a name, a duplo number
TMFDPL (Page 189)	Empty location search for loading
TMFPBP (Page 190)	Empty location search
TMGETT (Page 191)	T number for the specified tool name with duplo number
TMMVTL (Page 192)	Prepare magazine location for loading, unload tool
TMPOSM (Page 193)	Position magazine location or tool
TMPCIT (Page 194)	Set increment value for workpiece counter
TMRASS (Page 195)	Reset active status
TRESMO (Page 195)	Reset monitoring values
TSEARC (Page 196)	Complex search using search screen forms
TMCRMT (Page 199)	Create multitool
TMDLMT (Page 200)	Delete multitool
POSMT (Page 200)	Position multitool
FDPLMT (Page 201)	Search/check an empty location within the multitool

16.4.2 PI service: ASUP

Function: Assign interrupt

An interrupt number is assigned in the specified channel to a part program that is present in the control and is identified by path and program name.

The PI service ASUB has the same effect as the program instruction SETINT (or CLRINT if the PI service was called without specifying the path name).

Unlike SETINT (or CLRINT), the PI service ASUB remains active even after the end of the program (M30 or channel reset). An assignment activated by the PI service ASUB is not cleared until after a warm restart.

For detailed information about program management, path and file names, see:

Further information

Programming Manual NC Programming; Job Preparation; Section: "File and Program Management" > "Program Memory"

Description of formal parameters

Signal	Type	Value range	Meaning
PIService:	ANY	"PI".ASUP	Assign interrupt
Unit:	INT	1, 2, 3, ... 10	Channel number
Addr1:	STRING		Path name, e.g. "/_N_MPF_DIR/"
Addr2	STRING		Program name, e.g. "_N_TST_FC9ASUP_MPF"
WVar1:	WORD	1, 2, 3, ... 8	Interrupt number
WVar2:	WORD	1, 2, 3, ... 8	Priority
WVar3:	WORD	0, 1	LIFTFAST [1] Fast retraction from the contour
WVar4:	WORD	0, 1	BLSYNC [2] Processing of interrupt routine is only to start with the next block change

[1] **Further information**
Programming Manual NC Programming; Job Preparation, Section: "Flexible NC programming" > "Interrupt routine (ASUB)" > "Fast retraction from the contour (SETINT, **LIFTFAST**, ALF)"

[2] **Further information**
Programming Manual NC Programming; Job Preparation, Section: "Flexible NC programming" > "Interrupt routine (ASUB)" > "Assign and start interrupt routine (SETINT, PRIO, **BLSYNC**)"

Note

The ASUB PI service may only be executed when the specified channel is in the reset state. An ASUB prepared with FB7 can be subsequently initiated with FC9.

Further information

Programming Manual NC Programming; Job Preparation, Section: "Flexible NC Programming" > "Interrupt routine (ASUB)"

16.4.3 PI service: CANCEL

Function: Execute cancel

Triggers the "Cancel" function equivalent to the corresponding "Cancel alarm" button on the user interface (operator panel front).

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".CANCEL	Cancel

16.4.4 PI service: CONFIG

Function: Reconfiguration

The reconfiguration command activates machine data which has been entered sequentially by the operator or the PLC, almost in parallel.

The command can only be activated when the controller is in RESET state or the program is interrupted (NC stop at block limit). An FB4 error checkback signal is output if this condition is not fulfilled (state = 3).

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".CONFIG	Reconfiguration
Unit:	INT	1	
WVar1:	INT	1	Classification

16.4.5 PI service: DIGION

Function: Digitizing on

Selecting digitizing in the parameterized channel.

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".DIGION	Digitizing on
Unit:	INT	1, 2, 3 ... 10	Channel

16.4.6 PI service: DIGIOF

Function: Digitizing off

Deactivating digitizing in the parameterized channel.

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".DIGIOF	Digitizing off
Unit:	INT	1, 2, 3 ... 10	Channel

16.4.7 PI service: FINDBL

Function: Activate block search

A channel is switched to block search mode and the appropriate acknowledgment is then transmitted. The block search is then executed immediately by the NC. The block search pointer must already be in the NC at this point in time. The block search can be interrupted at any time by an NC RESET. Once the block search is successfully completed, the normal processing mode is reactivated automatically. NC start then takes effect from the located search target.

It is the sole responsibility of the operator to ensure a collision-free approach path.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".FINDBL	Block search
Unit:	INT	1 ... 10	Channel
WVar1:	WORD	1, 2	Preprocessing mode 1: Without calculation 2: With calculation

16.4.8 PI service: LOGIN

Function: Create password

Transfers the parameterized password to the NC. The password consists of up to 32 characters. For shorter passwords, the string can be supplemented with empty spaces to attain 8 characters

Example

Password: String[32]:= 'SafePassword.123';

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".LOGIN	Create password
Unit:	INT	1	NC
Addr1:	STRING	8 ... 32 characters	Password

16.4.9 PI service: LOGOUT

Function: Reset password

The password last transferred to the NC is reset.

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".LOGOUT	Reset password
Unit:	INT	1	NC

16.4.10 PI service: NCRES

Function: Trigger NC-RESET

Triggers an NC-RESET. Parameters "Unit" and "WVar1" must always be set to 0.

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".NCRES	Trigger NC-RESET
Unit:	INT	0	-
WVar1:	WORD	0	-

16.4.11 PI service: SELECT

Function: Select processing for a channel

A program stored on the NC is selected for one channel for execution. The program must be executable. The path and program name must be stated in full.

For detailed information, please refer to:

Further information

Programming Manual NC Programming; Job Preparation; Section: "File and Program Management" > "Program Memory"

Possible block types

Block types	
Workpiece directory	WPD
Main program	MPF
Subprogram	SPF
Cycles	CYC
Asynchronous subprograms	ASP
Binary files	BIN

Description of formal parameters

Signal	Type	Value range	Meaning
PIService:	ANY	"PI".SELECT	Program selection
Unit:	INT	1, 2, 3 ... 10	Channel

Signal	Type	Value range	Meaning
Addr1:	STRING	-	Path name
Addr2:	STRING	-	Program name

16.4.12 PI service: SETUDT

Function: Set current user data active

The current user data, such as tool offsets, basic frames and settable frames are set to active in the next NC block (only in stop state).

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".SETUDT	Activate user data
Unit:	INT	1, 2, 3 ... 10	Channel
WVar1:	WORD	1, 2, 3, 4, 5	User Data Type 1: Active tool offset 2: Active basic frame 3: Active settable frame 4: Active global basic frame 5: Active global settable frame
WVar2:	WORD	0	Reserved
Wvar3:	WORD	0	Reserved

16.4.13 PI service: SETUFR

Function: Activate user frames

User frames are loaded to the NC. All necessary frame values must be transferred to the NC first with FB3 "Write variables".

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".SETUFR	Activate user frames
Unit:	INT	1, 2, 3 ... 10	Channel

16.4.14 PI service: RETRAC

Function: Select JOG retract

Selects the JOG retract mode. The retraction axis, i.e. the geometry axis, with which the retraction is executed can be determined by the NC automatically or specified explicitly.

The mode remains active until it is ended with RESET.

Note

The RETRAC PI service can only be activated in JOG mode in the "Reset" state.

Automatic determination

For automatic determination, the geometric axis is selected as a retraction axis, which is perpendicular (orthogonal) to the currently selected working plane:

- G17: Retraction axis ⇒ 3rd geometry axis (Z)
- G18: Retraction axis ⇒ 2nd geometry axis (Y)
- G19: Retraction axis ⇒ 1st Geometry axis (X)

Note

OPI variable retractState

The active retraction axis can be read via the OPI variable retractState.Bit 2/3.

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".RETRAC	Select JOG retract mode
Unit:	INT	1, 2, 3 ... 10	Channel
WVar1:	WORD	0, 1, 2, 3	Retraction axis 0: Automatic determination of the retraction axis by the NC. 1: Retraction axis is the 1st geometry axis of the WCS 2: Retraction axis is the 2nd geometry axis of the WCS 3: Retraction axis is the 3rd geometry axis of the WCS
WVar2:	WORD	0	Reserved. The value must be pre-assigned with 0.

16.4.15 PI service: CRCEDN

Function: Creates new cutting edge

If the T number of an existing tool is entered in the "T number" parameter of the PI service, then a tool cutting edge with the stated D number is created for this tool.

If a T number is specified as a parameter and the tool for the entered T number does not exist, the PI service is aborted.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".CRCEDN	Create new cutting edge
Unit:	INT	1 … 10	TOA
WVar1:	INT		T number of the tool for which the tool cutting edge must be created.
WVar2:	INT	0, 1, 2, 3, ...	Edge number of tool cutting edge 0: D number is assigned by the NC

16.4.16 PI service: CREACE

Function: Create tool cutting edge

Creation of the cutting edge with the next higher / next unassigned D number for the tool with the transferred T number in TO, TS (if present). The cutting edge for the OEM cutting edge data is set up simultaneously in the TUE block (if one is present). If the tool for the stated T number does not exist, the PI service is aborted.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".CREACE	Create tool cutting edge
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T number

16.4.17 PI service: CREATO

Function: Create tool

Create a tool with specification of a T number. The tool is entered as existing in the tool directory area (TV). The first cutting edge "D1" (with zero content) is created for tool offsets in the TO block. The first "D1" (with zero content) is also created for the OEM cutting edge data in the TUE block (if present). If a TU block exists, it will contain the data set for the tool. If a tool with the stated T number already exists, the PI service is aborted.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".CREATO	Create tool
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T number

16.4.18 PI service: DELECE

Function: Deleting a tool cutting edge

If the T number of an existing tool is entered in the "T number" parameter of the PI service, the cutting edge with the specified D number is deleted from this tool. If a T number is entered as a parameter and the tool for the specified T number does not exist, the PI service is aborted. If the specified cutting edge does not exist, the PI service is aborted.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".DELECE	Delete cutting edge
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T number of the tool from which the cutting edge is to be deleted.
WVar2:	INT	0, 1, 2, 3, ...	D number of the tool cutting edge that must be deleted 0: The tool is deleted

16.4.19 PI service: DELETO

Function: Delete tool

Deletes the tool assigned to the transferred T number with all cutting edges (in TO, in some cases TU, TUE and TG (type 4xx), TD and TS blocks). If the tool for the stated T number does not exist, the PI service is aborted.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".DELETO	Delete tool
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T number

16.4.20 PI service: MMCSEM

Function

The PI service is used for testing, setting and resetting of channel-specific semaphores from HMI through the PLC. 10 semaphores are available per channel for protecting critical data areas.

Functions (PI services) are assigned to semaphores 1 to 6. The semaphores 7 to 10 can be freely used.

FB4 return values

- "Done" == TRUE
  The semaphore was set, the critical function can be called.
- "Error" == TRUE AND "State" == 3
  The semaphore was already set, currently the critical function cannot be called.

Schematic sequence for free semaphore

```
Testing and setting the semaphore
IF semaphore == FREE
THEN
   Writing/reading of critical data
   Resetting the semaphore
ELSE // Semaphore is blocked
...
ENDIF
```

NOTICE

Resetting the semaphore

After blocking the critical data area by setting the semaphore and subsequent reading or writing of the data, the critical data area must be enabled once again by resetting the semaphore, otherwise subsequent blocking will not be possible.

Schematic sequence for blocked semaphore

```
Testing and setting the semaphore
IF semaphore == FREE
THEN
   ...
ELSE // Semaphore is blocked
   Set bit memory for "Function could not be executed, repeat
necessary"
ENDIF
```

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".MMCSEM	Edit semaphore
Unit:	INT	1, 2, 3 ... 10	Channel
WVar1:	INT	[1)]	PI service-specific number of the semaphore
WVar2:	WORD	0, 1	Job type 0: Reset semaphore 1: Test and set semaphore

Signal	Type	Value range	Description
1)	Number	PI service	
	1	TMCRTO (create tool)	
	2	TMFDPL (search for empty location for loading)	
	3	TMMVTL (prepare magazine location for loading, unload tool)	
	4	TMFPBP (search for location)	
	5	TMGETT (search for tool number)	
	6	TSEARC (search for tool)	
	7 ... 10	Freely usable	

16.4.21 PI service: TMCRTO

Function: Create tool

Creation of a tool by specifying a name, a duplo number, e.g. with $TC_TP1[y] = <Duplo number> or $TC_TP2[y] = "<Tool name>". Or optionally by a T number, e.g. with y = <T number>.

The tool is entered in the tool directory area as available.

The first cutting edge D1 (with zero content) is created in the TO block for offsets.

The first cutting edge D1 (with zero content) is created in the TS block for monitoring data.

In the TUE block, if it is available, the cutting edge D1 is created for the OEM cutting edge data.

The TD block contains identifiers, duplo numbers, and the number of cutting edges (=1) for the T number that is optionally specified or assigned by the NC.

If a TU block exists, it will contain the data block for the tool.

After execution of the PI service, the T number of the tool created is available in the TV block under **TnumWZV**.

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" must be called with function number 1 for TMCRTO. See Chapter "PI service: MMCSEM (Page 186)".

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".TMCRTO	Create tool
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT	>0 → the stated T number is specified -1 → the NC assigns the T number	T number
WVar2:	INT		Duplo number
Addr1:	STRING	Max. 31 characters	Tool name

16.4.22 PI service: TMFDPL

Function

Search for empty location for loading according to the parameter assignment:

- LocationNumber_To = -1 AND MagazineNumber_To = -1
  Searches all magazines in the specified area (= channel) for an empty location for the tool specified by a T number. After execution of the PI service, the magazine and locations numbers found during the search are listed in the configuration block of the channel (component **magCMCmdPar1** (magazine number) and **magCMCmdPar2** (location number)). LocationNumber_ID and MagazineNumber_ID can be set as search criteria or not (= -1). The PI service is positively or negatively acknowledged according to the search result.
- LocationNumber_To = -1 AND MagazineNumber_To = magazine number
  An empty location for the tool specified with a T number is searched for in the specified magazine. Location number (reference) and MagazineNumber_Ref can be allocated as search criteria or with -1. The PI service is positively or negatively acknowledged according to the search result.
- LocationNumber_To = location number AND MagazineNumber_To = magazine number
  The specified location is checked to ensure that it is free for loading the tool. LocationNumber_ID and MagazineNumber_ID can be set as search criteria or -1. The PI service is acknowledged positively or negatively according to the search result.

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" must be called with function number 2 for TMFDPL.
See Chapter "PI service: MMCSEM (Page 186)".

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".TMFDPL	Empty location for loading
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T number
WVar2:	INT		LocationNumber_To (target)
WVar3:	INT		MagazineNumber_To (target)
WVar4:	INT		Location_number_ID; Number of the reference location (opt.).[1)]
WVar5:	INT		Magazine_number_ID; Number of the reference magazine (opt.).[1)]

[1)] : The reference location/reference magazine is a buffer location/buffer store, i.e. a tool location from the buffer magazine or from the internal loading magazine, e.g. swapping station, gripper, loading station.

16.4.23 PI service: TMFPBP

Function: Empty location search

The PI service searches the specified magazine(s) for an empty location which meets the specified criteria such as tool size and location type.

If the search is successful, the result can be read from the following OPI variables:

- magCMCmdPar1 (magazine number)
- magCMCmdPar2 (location number)

Note

The PI service can only be requested using FB7. See Chapter "FB7: PI_SERV2 - request PI service (Page 210)".

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" must be called with function number 4 for TMFPBP. See Chapter "PI service: MMCSEM (Page 186)".

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".TMFPBP	Empty location search
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		**Magazine number** of the magazine **from** which the search is to be performed
WVar2:	INT		**Location number** of the location in the magazine from "WVar1"
WVar3:	INT		**Magazine number** of the magazine up to which the search is to be performed
WVar4:	INT		**Location number** of the location in the magazine from "WVar3"
WVar5:	INT		Location_number_des; Number of the reference location (opt.).[1)]
WVar6:	INT		Magazine_number_des; Number of the reference magazine (opt.).[1)]
WVar7:	INT	0, 1, 2 ... 7	Number of required half locations to left
WVar8:	INT	0, 1, 2 ... 7	Number of required half locations to right
WVar9:	INT	0, 1, 2 ... 7	Number of required half locations in upward direction
WVar10:	INT	0, 1, 2 ... 7	Number of required half locations in downward direction

Signal	Type	Value range	Description
WVar11:	INT		Number of required location type
WVar12:	INT	0, 1, 2, 3, 4	Specifies the required search direction 0: Search strategy as set in $TC_MAMP2 1: Forward 2: Backward 3: Symmetrical

[1] : The reference location/reference magazine is a buffer location/buffer store, i.e. a tool location from the buffer magazine or from the internal loading magazine, e.g. swapping station, gripper, loading station.

Examples: Setting the search range

From	Location	To	Location	Description
WVar1	WVar2	WVar3	WVar4	
#M1	#P1	#M1	#P1	Only location #P1 in magazine #M1 is checked
#M1	#P1	#M2	#P2	Locations starting at magazine #M1, location #P1 up to magazine #M2, location #P2 are searched
#M1	-1	#M1	-1	All locations in magazine #M1 - and no others - are searched
#M1	-1	-1	-1	All locations starting at magazine #M1 are searched
#M1	#P1	-1	-1	All locations starting at magazine #M1 and location #P1 are searched
#M1	#P1	#M1	-1	Locations in magazine #M1 starting at magazine #M1 and location #P1 in this magazine are searched
#M1	#P1	#M2	-1	Locations starting at magazine #M1, location #P1 up to magazine #M2 are searched
#M1	-1	#M2	#P2	Locations starting at magazine #M1 up to magazine #M2, location #P2 are searched
#M1	-1	#M2	-1	Locations starting at magazine #M1 up to and including magazine #M2 are searched
-1	-1	-1	-1	All magazine locations are searched

16.4.24 PI service: TMGETT

Function: Determine T number for the specified tool name with duplo number

The PI service determines the T number of a tool via the tool name and duplo number.

The result is written in OPI variables in the TF block (parameterization, return parameters from TMGETT, TSEARC):

- resultNrOfTools
  - resultNrOfTools == 0: The specified tool was not found
  - resultNrOfTools == 1: The specified tool was found
- resultToolNr: T number of the specified tool with "resultNrOfTools" == 1

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" must be called with function number 5 for TMGETT. See Chapter "PI service: MMCSEM (Page 186)".

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".TMGETT	Determining the T number
Unit:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING	Max. 31 characters	Tool name
WVar1:	INT		Duplo number

16.4.25 PI service: TMMVTL

Function: Prepare magazine location for loading, unload tool

The PI service loads, unloads and relocates tools:

1. Loading and unloading: Loading point ↔ magazine
2. Loading and unloading: Loading point ↔ buffer storage, e.g. spindle
3. Relocation within a magazine
4. Relocation between different magazines
5. Relocation between magazine and buffer storage
6. Relocation within buffer storage

Cases 1, 3, 4 and 5: The following OPI variables of the TM block (magazine data: general data) are written:

- magCmd (area no. = TO unit, line = magazine number)
- magCmdState ← "acknowledgment"

Cases 2 and 6: The following OPI variables of the TMC block (magazine data: configuration data) are written:

- magCBCmd (area no. = TO unit)
- magCBCmdState ← "acknowledgment"

Loading

- "WVar2" LocationNumber_From, "WVar3" MagazineNumber_From
  The tool location of the magazine is moved to the loading station/location for loading and the tool is loaded.
- "WVar4" LocationNumber_To== -1
  First, an empty location for the tool is searched for in the magazine. The empty location of the magazine is then moved to the loading station/location for loading and the tool is loaded.

Unload

- "WVar4" LocationNumber_To, "WVar5" MagazineNumber_To
- The tool location of the magazine is moved to the loading station for loading and the tool is unloaded.

In the OPI block TP (magazine data: location data), the magazine location of the removed tool is designated as free.

Addressing the tool

The tool can be addressed either by T number or by the location and magazine numbers. The value -1 is to be assigned to unused parameters.

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" must be called with function number 3 for TMMVTL. See Chapter "PI service: MMCSEM (Page 186)".

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".TMMVTL	Make magazine location ready for loading, unload tool
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T number
WVar2:	INT		LocationNumber_From (source)
WVar3:	INT		MagazineNumber_From (source)
WVar4:	INT		LocationNumber_To (target)
WVar5:	INT		MagazineNumber_To (target)

16.4.26 PI service: TMPOSM

Function: Position magazine location or tool according to the parameter assignment

This PI service can traverse a magazine to position a magazine location at a specified position (e.g. at a loading position). The magazine location can be specified directly or via a tool at the location.

The destination, e.g. the loading location, is specified in the parameters:

- "WVar5" LocationNumber_Ref
- "WVar6" MagazineNumber_Ref

The magazine location to be positioned is specified in the following parameters according to the respective specification:

- "WVar1" T number of the tool
  The following parameters are irrelevant here:
  - "Addr1" tool name = ""
  - "WVar2" duplo number = -1
  - "WVar3" LocationNumber_From = -1
  - "WVar4" MagazineNumber_From = -1
- "Addr1" tool name, "WVar2" duplo number
  The following parameters are irrelevant here:
  - "WVar1" T number of the tool = -1
  - "WVar3" LocationNumber_From = -1
  - "WVar4" MagazineNumber_From = -1
- "WVar3" LocationNumber_From, "WVar4" MagazineNumber_From
  The following parameters are irrelevant here:
  - "Addr1" tool name = ""
  - "WVar1" T number of the tool = -1
  - "WVar2" duplo number = -1

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".TMPOSM	Position magazine location or tool
Unit:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING	Max. 31 characters	Tool name
WVar1:	INT		T number of the tool
WVar2:	INT		Duplo number
WVar3:	INT		LocationNumber_From (source)
WVar4:	INT		MagazineNumber_From (source)
WVar5:	INT		LocationNumber_Ref
WVar6:	INT		MagazineNumber_Ref

16.4.27 PI service: TMPCIT

Function: Set increment value for workpiece counter

Incrementing the workpiece counter of the spindle tool

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".TMPCIT	Set increment value for workpiece counter
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	WORD	0, 1, 2, ... max.	Spindle number; corresponds to the type index in the location data with "Spindle" location type of the buffer magazine in channel. 0: Main spindle
WVar2:	WORD	0 ... max.	Increment value; indicates the number of spindle revolutions after which the workpiece counter is incremented

16.4.28 PI service: TMRASS

Function: Resetting the tool status "active"

The PI service sets the status to "inactive" for all tools with a tool status "active" and "blocked".

The following events are sensible times to reset the tool status:

- A negative edge of the NC/PLC interface signal "tool disable ineffective"
- End of program
- Channel reset.

The PI service is intended for the PLC, as there it is known when the disabled tool is no longer to be used.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI". TMRASS	Reset active status
Unit:	INT	1, 2, 3 ... 10	TO area

16.4.29 PI service: TRESMO

Function: Reset monitoring values

This PI service resets the monitoring values of the designated cutting edges of the designated tools to their setpoints (initial values).
This is only performed for tools with active monitoring.

See also the RESETMON command.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".TRESMO	Reset monitoring values
Unit:	INT	1, 2, 3 ... 10	TO area
WVar1:	WORD	- max ... max	Tool number 0: Process all tools >0: Process specified tool <0: Process all sister tools
WVar2:	WORD	0, 1, 2, ... max.	D number >0: Reset the monitoring of the specified cutting edge 0: Reset the monitoring of all cutting edges
WVar3:	WORD		Monitoring mode that is to be reset (all combinations are possible):
			Bit 0 = 1: Tool-life monitoring
			Bit 1 = 1: Quantity monitoring
			Bit 2 = 1: Wear monitoring
			Bit 3 = 1: Sum-offset monitoring
			0: Reset all active tool-monitoring functions ($TC_TP9).

16.4.30 PI service: TSEARC

Function: Complex search using search screen form, depending on the parameter assignment

The PI service is used to search for tools with specific properties within a search range in one or more magazines, beginning with a specific location, up to a specific location.

Note

Active tool management

The PI service is only available if tool management is activated.

Specification options

- Search direction
- Search for next tool with the specified property
- Search for all tools with the specified property

Result

As a result, a list with the internal T-numbers of the tools that are found is created.

Logic operations

For filtering properties, only one AND link is available as a linking option. An OR link must be achieved by the user via several calls of the PI service and subsequent evaluation of the individual results.

Parameterization of the tool properties

The properties of the searched for tools are set in the BTSS block TF (parameterization, return parameters from TMGETT, TSEARC) via the following variables:

- "parMask<X>" parameterization mask
- "parData<X>" comparison values

With <X> = TAD, TAO, TAS, TD

Result list

After completion of the PI service without errors, the search result is located in the BTSS block TF (parameterization, return parameter from TMGETT, TSEARC) in the following variables:

- "resultCuttingEdgeNrUsed" D numbers of cutting edges used since last quantity count
- "resultNrOfCutEdgesUsed" Number of cutting edges since last quantity count
- "resultNrOfTools" Number of found tools
- "resultToolNr" T numbers of found tools
- "resultToolNrUsed" T numbers of cutting edges used since last quantity count

If no tool was found, the number of found tools is zero ("resultNrOfTools" == 0).

Search range specifications

From mag. number `WVar1`	From location number `WVar2`	To mag. number `WVar3`	To location number `WVar4`	Description The following magazine locations are searched:
#M<a>	#L<b>	#M<x>	#L<y>	From: Magazine #M<a>, location #L<b> to: Magazine #M<x>, location #L<y>
#M<a>	-1	#M<a>	-1	From: Magazine #M<a>, first location to: Magazine #M<a>, last location
#M<a>	-1	-1	-1	From: Magazine #M<a>, first location to: Last magazine, last location
#M<a>	#L<b>	-1	-1	From: Magazine #M<a>, location #L<b> to: Last magazine, last location
#M<a>	#L<b>	#M<a>	-1	From: Magazine #M<a>, location #L<b> to: Magazine #M<a>, last location
#M<a>	#L<b>	#M<x>	-1	From: Magazine #M<a>, location #L<b> to: Magazine #M<b>, last location
#M<a>	-1	#M<x>	#L<y>	From: Magazine #M<a>, first location to: Magazine #M<b>, location #L<y>

From mag. number WVar1	From location number WVar2	To mag. number WVar3	To location number WVar4	Description The following magazine locations are searched:
#M<a>	-1	#M<x>	-1	From: Magazine #M<a>, first location to: Magazine #M<b>, last location
-1	-1	-1	-1	From: First magazine, first location to: Last magazine, last location

Symmetrical search

A symmetrical search, relative to a magazine location, can only be performed if the following conditions have been met:

- The search range must encompass only one magazine: "WVar1" (from: Magazine number) == "WVar3" (to: magazine number)
- Specification of a reference location, i.e. a magazine location for which a symmetrical search is to be performed: "WVar5" (number of the reference magazine) and "WVar6" (number of the reference location)
- For the reference location, a multiple assignment to the magazine to be searched must have been configured in the TPM block.
- "WVar7" (search direction) = 3

The reference location is a buffer location, i.e. a tool location from the buffer magazine or from the internal loading magazine, e.g. swapping station, gripper, loading station. The symmetrical search is made in relation to the magazine location in front of the reference location.

If the magazine location is upstream of the reference location outside of the search range, the PI service will respond as if no suitable location had been found.

Note

Before and after this PI service, the MMCSEM PI service must be called up with the associated parameter WVar1 for this PI service. See Chapter "PI service: MMCSEM (Page 186)".

Description of formal parameters

Signal	Type	Value range	Description
PIService:	ANY	"PI".TSEARC	Complex search using search screen forms
Unit:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT	-1, 1, …	From: Magazine number
WVar2:	INT	-1, 1, …	From: Location number
WVar3:	INT	-1, 1, …	to: Magazine number
WVar4:	INT	-1, 1, …	to: Location number
WVar5:	INT	-1, 1, …	Number of the reference magazine (only relevant for symmetrical search: Search direction ==3)

Signal	Type	Value range	Description
WVar6:	INT	-1, 1, …	Number of the reference location (only relevant for symmetrical search: Search direction ==3)
WVar7:	INT	1, 2, 3	Search direction: 1: Forwards from the first location of the search domain 2: Backwards from the last location of the search domain 3: Symmetrical to the real magazine location, which is located **upstream** of the location specified under "WVar5" (number of the reference magazine) and "WVar6" (number of the reference location)
WVar8:	INT	0, 1, 2, 3	Properties of the tools: 0: all tools, cutting edge-specific 1: first tool, cutting edge-specific 2: all tools, via all cutting edges 3: first tool, via all cutting edges

16.4.31 PI service: TMCRMT

Function: Create multitool

The PI service creates a new multitool with a defined identifier, an optionally specifiable multitool number, the number of tool locations, and the type of distance coding.

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" =1 (TMCRTO) must be called. See Chapter: "PI service: MMCSEM (Page 186)".

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".TMCRMT	Create multitool
Unit:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING		Multitool identifier (max. 31 characters)
WVar1:	INT	0	Reserved
WVar2:	INT	-1, 1, 2, ... 32000	Multitool number -1: Automatic assignment of the multitool number by NC 1, 2, 3 ... 32000: Multitool number, **note**: The multitool number must be unique

Signal	Type	Range of values	Meaning
WVar3:	INT	2, 3, 4, ... MAX	Number of tool locations MAX = Parameterized number in MD17504 $MN_MAX_TOOLS_PER_MULTITOOL
WVar4:	INT	1, 2, 3	Type of distance coding

16.4.32 PI service: TMDLMT

Function: Delete multitool

The PI service deletes a multitool in all the data blocks in which it is saved. Tools loaded onto the multitool are not subsequently equipped or loaded, however, continue to be defined if they are not also to be deleted.

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".TMDLMT	Delete multitool
Unit:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING		Multitool identifier (max. 31 characters)
WVar1:	INT	0	Reserved, always 0
WVar2:	INT	-1, 1, 2, ... 32000	Multitool number -1: Delete the multitool with the name specified in "Addr1" 1, 2, 3 ... 32000: Delete the multitool with the specified multitool number; **Note:** Parameter "Addr1" is not evaluated
WVar3:	INT	0, 1	Tools contained in the multitool: 0: Do not delete 1: Delete

16.4.33 PI service: POSMT

Function: Position multitool

The PI service positions a multitool at the programmed location or alternatively at the programmed tool, which is located in one of the locations of the multitool. The tool itself can either be specified by its T number or by its name and duplo number. A multitool can only be positioned if it is on a tool carrier (e.g. spindle) and if no tool offset is active with regard to this tool carrier.

Position specification

The position specification can be specified as one of three variants:

No.	Addr1	WVar1	WVar2	WVar3	WVar4
1	Empty string	Number of the tool carrier	Tool number	-1	-1
2	Tool name	Number of the tool carrier	-1	Duplo number	-1
3	Empty string	Number of the tool carrier	-1	-1	Multitool location number

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".POSMT	Position multitool
Unit:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING		Tool name of the tool to be positioned in the multitool (max. 31 characters)
WVar1:	INT	1, 2, 3, ...	Number of the tool carrier
WVar2:	INT	-1, 1, 2, 3, ... 32000	Tool number (T number) of the tool to be positioned in the multitool 1, 2, 3, ... 32000: Tool number -1: "Addr1" (tool name) and "WVar3" (duplo number) are used
WVar3:	INT	-1, 1, 2, 3, ... 32000	Duplo number of the tool to be positioned in the multitool 1, 2, 3, ... 32000: Duplo number -1: "WVar2" (T number) is used
WVar4:	INT	-1, 1, 2, 3, ...	Multitool location number of the tool carrier location at which the system should be positioned -1: "WVar2" or "Addr1" and "WVar3" are used

16.4.34 PI service: FDPLMT

Function: Searching for or checking an empty location within the multitool

The PI service searches for a free tool location in a multitool to hold the specified tool or to check whether the specified tool location in the multitool is free to hold the specified tool. The tool can be specified by the T number or the identifier and the duplo number.

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" = 2 (TMFDPL) must be called. See Chapter "PI service: MMCSEM (Page 186)".

Position specification

The tool to be positioned in the multitool can be specified as one of the following variants:

- "Addr1" = <Tool name>, "WVar1" = -1 and "WVar2" = <Duplo number>
- "Addr1" = <Empty string>, "WVar1" = <Tool number> and "WVar2" = -1

The position specification can be assigned via:

- "WVar4" = <Multitool location number>

Description of formal parameters

Signal	Type	Range of values	Meaning
PIService:	ANY	"PI".FDPLMT	Search/check an empty tool location within a multitool
Unit:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING		Tool name of the tool to be positioned in the multitool (max. 31 characters)
WVar1:	INT	-1, 1, 2, 3, ... 32000	Tool number (T number) of the tool to be positioned in the multitool -1: "Addr1" (tool name) and "WVar2" (duplo number) are used. 1, 2, 3 ... 32000: Tool number
WVar2:	INT	-1, 1, 2, 3, ... 32000	Duplo number of the tool to be positioned in the multitool 1, 2, 3, ... 32000: Duplo number -1: "WVar1" (T number) is used
WVar3:	INT	-1, 1, 2, 3, ... 32000	Number of the multitool 1, 2, 3, ... 32000: Number of the multitool -1: Search across all multitools for an empty location or check across all multitools whether in one of them the specified tool location is free to hold the tool.
WVar4:	INT	-1, 1, 2, 3, ...	Multitool location number of the tool location at which the system should be positioned >0: Multitool location number -1: Search for any empty tool location within the multitool

16.5 FB5: GETGUD - read GUD variable

Function

The function block FB5 "GETGUD" is used for reading global user data (GUD) in the NC or channel area.

Request for reading NC variables

Call of FB5 with positive edge change, parameter "Req" = 0 → 1

Parameter "Addr": The pointer to the name of the GUD variables, symbolically with "<Data block>".<Variable name>

Parameters "Area", "Unit", "Index1", and "Index2": Additional information for addressing the variables

When the parameter "CnvtToken" is activated, the user receives a token (variable pointer) for the GUD variable to be read. Using this, the GUD variables can then be read or written via FB2 and FB3 with parameter "Addr1" ... "Addr8" = "<Token>". Addressing by means of a token is mandatory for **writing** GUD variables. If the token is used to address GUD variable arrays, the parameter "Line1" ... "Line8" = "<array index>" of the FB2/FB3 must also be provided with values.

To read a GUD variable of the DOUBLE type without adapting the format, an ANY pointer of the REAL2 type must be specified in the target area. E.g. P#M100.0 REAL 2. The value of the GUD variables of the DOUBLE type is then adopted in the PLC data area as a 64-bit floating point number.

Completion of the read request

Read request successfully completed: Parameter "Done" == 1.

Read request completed with error:"Parameter "Error" == 1, error cause in parameter "State"

Prerequisites

- Release of the NC/PLC communication by OB100, FB1 parameter "NCKomm" = 1
- For the data block that will contain the string with the name of the GUD variable, the S7 Symbol Editor must be used to assign a symbol in the S7 symbol table of the S7 project (cf. call example 1 below: DB_GUDVAR). In the data block, a string of suitable length must be created to contain the name of the GUD variable (cf. call example 1 below: "DB_GUDVAR".GUDVar1). This symbol is then passed on to parameter "Addr" of FB5, e.g. Addr := "DB_GUDVAR".GUDVar1. If the token for the GUD variable is required for the following calls of the FB2/FB3, the 10-byte token structure must be created in a DB, e.g. the same DB (cf. call example 1 below: "DB_GUDVAR".GUDVar1Token). This structure is specified in parameter VarToken of the FB5, cf. call example 1 below: FB5, VarToken := "DB_GUDVAR".GUDVar1Token. This token is then specified in parameters "Addr1" … "Addr8" when FB2/FB3 is called, cf. call example 1 below: FB3, Addr1 := "GUDVar1Token".

General conditions

- FB5 has multi-instance capability.
- Every call of FB5 must be assigned a separate instance DB from the user area.
- Reading of a GUD variable (FB5 call with Req = 1) is only permitted in the cyclic part of the PLC basic program (OB1). If the job is not started (FB5 call with Req = 0), the parameters can also be written in the start-up part of the PLC basic program (OB100). The job can then be executed using the already written parameters in the cyclic part of the PLC basic program (OB1) by calling FB5 with Req = 1.
- Only capital letters may be used for the names of GUD variables.
- Reading of a GUD variable generally extends over several PLC cycles.

Note

Error case
When variables from different channels are read, the following feedback message is output:

- "Error" == TRUE
- "State" == W#16#02

Note

To read a variable of the DOUBLE type from NC without adapting the format, an ANY pointer of the REAL2 type must be specified in the target area (e.g.: P#M100.0 REAL 2). If the basic program recognizes REAL 2 as the target type when reading a variable of the DOUBLE type, the data is applied to the PLC data area as a 64-bit floating-point number.

Declaration of the function

```
FUNCTION_BLOCK FB5          //Server name
   KNOW_HOW_PROTECT
   VERSION : 3.0
VAR_INPUT
   Req :          BOOL;
   Addr:          ANY;       // Variable name
   Area           BYTE ;     //Area: NCK = 0, channel = 2
   Unit :         BYTE ;
   Index1:        INT;       // Array index 1
   Index2:        INT;       // Array index 2
   CnvtToken:     BOOL;      // Conversion into 10-byte token
   VarToken       ANY;       // Struct with 10 bytes for the variable token
END_VAR

VAR_OUTPUT
   Error :        BOOL;
   Done :         BOOL;
   State :        WORD;
END_VAR

VAR_IN_OUT
   RD:            ANY;
END_VAR

BEGIN
END_FUNCTION_BLOCK
```

Description of formal parameters

Signal	Type	Type	Value range	Description
Req:	I	BOOL		Job start with positive signal edge
Addr:	I	ANY	"<DBName>".<Var-Name>	Variable name in a variable of the type STRING
Area:	I	BYTE	0, 2	Range: 0: NC 2: Channel
Unit:	I	BYTE	1, 2, 3 ... 10	Area == NC: Unit:=1 Area == Channel: Channel number
Index1:	I	INT	0, 1, 2, ... < max. Array index>	Array index 1 For variables without array index: Index1 = 0
Index2:	I	INT	0, 1, 2, ... < max. Array index>	Array index 2 For variables without 2nd array index: Index2 = 0
CnvtToken:	I	BOOL	0 (FALSE), 1 (TRUE)	Activate generation of a 10 byte variable token
VarToken:	I	ANY	"<DBName>".<Var-Name>	Address of a 10byte token (see example)
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	Negative acknowledgement of job or execution of job impossible
Done:	O	BOOL	0 (FALSE), 1 (TRUE)	Job successfully executed
State:	O	WORD	---	See paragraph "Error identifiers"
RD:	I/O	ANY	P#Mm.n BYTE x ... P#DBnr.dbxm.n BYTE x	Target area for read data

Error identifiers

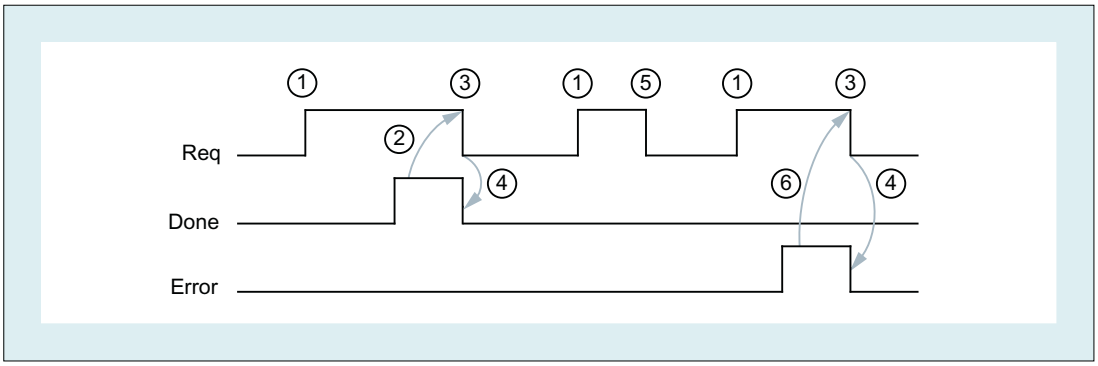
State		Description	Note
WORD H [1)]	WORD L		
0	1	Access error	
0	2	Error in job	Incorrect compilation of Var. in a job
0	3	Negative acknowledgement, job not executable	Internal error, try: NC RESET
0	4	Data areas or data types do not tally	Check data to be read in RD
1	4	Insufficient local user memory available	read variable is longer than specified in RD
0	6	FIFO full	Job must be repeated, since queue is full
0	7	Option not set	BP parameter "NCKomm" is not set
0	8	Incorrect target area (SD)	RD may not be local data
0	9	Transmission occupied	Job must be repeated

State		Description	Note
WORD H [1)]	WORD L		
0	10	Error in addressing	Unit contains value 0
0	11	Address of variable invalid	Check addr (or variable name), area, unit
1 - 8	13	ANY data reference incorrect	String/NcVar data required has not been parameterized
0	15	String more than 32 characters	GUD variable name too long
1) High byte > 0 ⇒ Number of the variable in which the error occurred			

Configuration steps

To be able to read a GUD variable, its name must be stored in a string variable. The data block with this string variable must be defined in the symbol table so that the "Addr" parameter can be assigned symbolically for FB GETGUD. A structure variable can be defined optionally in any data area of the PLC to receive the variable pointer (see specification in following example).

Pulse diagram



① User: Set request, Req = 0 → 1
② FB5 successfully completed, Done = 1
User: Reset request, IF Done == 1 THEN Req = 0
③ User: IF Done == 1 THEN reset request: 1 → 0
④ FB5 reset job confirmation, Done = 0
⑤ User: IF Done == 0 AND Error == 0 THEN reset request Req = 1 → 0 **not permissible**
⑥ FB5 completed with errors, Error = 1
User: Reset request, IF Done == 1 OR Error == 1 THEN Req = 0, possible further error handling

Call example 1

Read a GUD variable from channel 1:

- Name "GUDVAR1"
- Type: INTEGER

Conversion to a 10 byte variable pointer. See table "Assignment of the data types" in Chapter "FB2: GET - read NC variable (Page 157)"

Reading the GUD variables: FB5 with instance DB111

```
// Data block for GUD variable
DATA_BLOCK DB_GUDVAR    // Assignment in symbol table

STRUCT
   GUDVar1 : STRING[32] := 'GUDVAR1';               // Name is defined by
                                                    user
   GUDVar1Token :
   STRUCT
      SYNTAX_ID : BYTE ;
      area_and_unit : BYTE ;
      column : WORD;
      line : WORD;
      block type : BYTE ;
      NO. OF LINES : BYTE ;
      type : BYTE ;
      length : BYTE ;
   END_STRUCT;
END_STRUCT;

BEGIN
END_DATA_BLOCK

// Unassigned user DB, as instance for FB5
DATA_BLOCK DB111

        FB5
BEGIN
END_DATA_BLOCK

// Unassigned user DB, as instance for FB3
DATE_BLOCK DB112

        FB3
BEGIN
END_DATA_BLOCK

// Reading of a channel-specific GUD variable from channel 1, with
conversion to a variable pointer
Function FC "VariablenCall" : VOID
U     I 7.7;           // Unassigned machine control panel key
S     M 100.0;         // Activate req.
U     M 100.1;         // Done completed message
```

```
R     M 100.0;          // Terminate job
U     I 7.6;            // Manual error acknowledgment
U     M 102.0;          // Error pending
R     M 100.0;          // Terminate job
CALL FB5, DB111(
           Req         := M 100.0,             // Starting edge for reading
           Addr        := "DB_GUDVAR".GUDVar1, // Name of the GUD
           Area        := B#16#2,              // Channel variable
           Unit        := B#16#1,              // Channel 1
           Index1      := 0,                   // No array index 1
           Index2      := 0,                   // No array index 2
           CnvtToken   := TRUE,                // Request: Conversion into
                                               10-byte token
           VarToken    := "DB_GUDVAR".GUDVar1Token, // Address of the token
           Error       := M 102.0,
           Done        := M 100.1,
           State       := MW 104
           RD          := P#DB99.DBX0.0 DINT 1 // free memory area
);
```

Writing the GUD variables: FB3 with instance DB112

GUD variable token from FB5, parameter: "VarToken" for writing with FB3, parameter "Addr1"

```
CALL FB3, DB112(
           Req         := M 200.0,
           NumVar      := 1,                     // one GUD variable
           Addr1       :=                        // Token
                      "DB_GUDVAR".GUDVar1Token,
           Error       := M 102.0,
           Done        := M 100.1,
           State       := MW 104
           SD1         := P#DB99.DBX0.0 DINT 1
);
```

Call example 2

Read a GUD variable from channel 1:

- Name "GUD_STRING"
- Type: STRING[32]

Conversion to a 10 byte variable pointer.

Reading the GUD variables: FB5 with instance DB111

```
// Data block for GUD variable
DATA_BLOCK DB_GUDVAR    // Assignment in symbol table
```

```
STRUCT
    GUDVarS : STRING[32] := 'GUD_STRING';    // Name defined by user
    GUDVarSToken :
    STRUCT
       SYNTAX_ID : BYTE ;
       area_and_unit : BYTE ;
       column : WORD;
       line : WORD;
       block type : BYTE ;
       NO. OF LINES : BYTE ;
       type : BYTE ;
       length : BYTE ;
    END_STRUCT;

    string_of_GUD : STRING[30];     // must at least be so long as
                                    // the definition of 'GUD_STRING'!
    new_name : STRING[30] := 'GUD_123';
END_STRUCT;

BEGIN
END_DATA_BLOCK

// Unassigned user DB, as instance for FB5
DATA_BLOCK DB111

        FB5
BEGIN
END_DATA_BLOCK

// Unassigned user DB, as instance for FB3
DATE_BLOCK DB112

        FB3
BEGIN
END_DATA_BLOCK

// Reading of a channel-specific GUD variable from channel 1, with
conversion to a variable pointer
Function FC "VariablenCall" : VOID
U     I 7.7;           // Unassigned machine control panel key
S     M 100.0;         // Activate req.
U     M 100.1;         // Done completed message
R     M 100.0;         // Terminate job
U     I 7.6;           // Manual error acknowledgment
U     M 102.0;         // Error pending
```

```
R     M 100.0;          // Terminate job
CALL FB5, DB111(
           Req          := M 100.0,     // Starting edge for reading
           Addr         := "DB_GUDVAR".GUDVarS,
           Area         := B#16#2,      // Channel variable
           Unit         := B#16#1,      // Channel 1
           Index1       := 0,           // No array index
           Index2       := 0,           // No array index
           CnvtToken    := TRUE,        // Request: Conversion into 10-byte
                                        token
           VarToken     := "DB_GUDVAR".GUDVarSToken, // Address of the token
           Error        := M 102.0,
           Done         := M 100.1,
           State        := MW 104
           RD           := "DB_GUDVAR".string_of_GUD
);
```

Writing the GUD variables: FB3 with instance DB112

GUD variable token from FB5, parameter: "VarToken" for writing with FB3, parameter "Addr1"

```
CALL FB3, DB112(
           Req          := M 200.0,
           NumVar       := 1,                // one GUD variable
           Addr1        := "DB_GUDVAR".GUDVarSToken, // Token
           Error        := M 102.0,
           Done         := M 100.1
           State        := MW 104
           SD1          := "DB_GUDVAR".new_name
);
```

Classification of data types

See table "Assignment of the data types" in Chapter "FB2: GET - read NC variable (Page 157)"

16.6 FB7: PI_SERV2 - request PI service

Function

With the exception of a higher number of "WVar" parameters ("WVar11" - "WVar16"), function block FB7 has the same functionality as function block FB4. It is recommended that function block FB7 be used instead of function block FB4.

For a detailed description, refer to Chapter "FB4: PI_SERV - request PI service (Page 173)".

Declaration of the function

```
FUNCTION_BLOCK FB7
Var_INPUT
    Req:           BOOL;
    PIService:     ANY;
    Unit:          INT;
    Addr1:         ANY;
    Addr2:         ANY;
    Addr3:         ANY;
    Addr4:         ANY;
    WVar1:         WORD;
    WVar2:         WORD;
    WVar3:         WORD;
    WVar4:         WORD;
    WVar5:         WORD;
    WVar6:         WORD;
    WVar7:         WORD;
    WVar8:         WORD;
    WVar9:         WORD;
    WVar10:        WORD;
    WVar11:        WORD;
    WVar12:        WORD;
    WVar13:        WORD;
    WVar14:        WORD;
    WVar15:        WORD;
    WVar16:        WORD;
END_VAR
VAR_OUTPUT
    Error :        BOOL;
    Done :         BOOL;
    State :        WORD;
END_VAR
```

16.7 FB9: MtoN - operator panel switchover

Function

The function block FB9 "MzuN" is used for switching over operating components (MCP/OP), which are connected via a bus system to one or more NCU control modules.

The interface between the individual operating units (operator panels) and the NCU (PLC) is the M : N interface in data block DB19. FB9 uses the signals of these interfaces.

Apart from initialization, sign-of-life monitoring and error routines, the following basic functions are also performed by the block for control unit switchover:

Tabulated overview of functions:	
Basic function	**Description**
HMI queuing	HMI wants to go online with an NCU
HMI coming	HMI is connecting to an NCU
HMI going	HMI is disconnecting from an NCU
Forced break	HMI must break connection with an NCU
Operating focus changeover to server mode	Change operating focus from one NCU to the other
Active/passive operating mode:	Operator control and monitoring/monitoring only
MCP switchover	As an option, MCP can be switched over with the HMI

Note

The block must be called by the user program. The user must provide an instance DB with any number for this purpose. The call is multi-instance-capable.

Brief description of a few important functions

Active/passive operating mode:

An online HMI can operate in two different modes:

Active mode:	Operator can control and monitor
Passive mode:	Operator can monitor (HMI header only)

After switchover to an NCU, this initially requests active operating mode in the PLC of the online NCU. If two control units are linked online simultaneously to an NCU, one of the two is always in active mode and the other in passive mode. The operator can request active mode on the passive HMI at the press of a button.

MCP switchover

As an option, an MCP assigned to the HMI can be switched over at the same time. To achieve this, the MCP address must be entered in the "mstt_adress" parameter of the NETNAMES.INI configuration file on the HMI and "MCPEnable" must be set to TRUE. The MCP of the passive HMI is deactivated so that there is only ever one active MCP on an NCU at one time.

Boot condition

In order to prevent, that for an NCU restart, the previously selected MCP is activated, when calling FB1 in OB100, input parameter "MCP1BusAdr" must be set = **255** (address 1st MCP) and "MCP1Stop" must be set **= TRUE** (switch off 1st MCP).

Approvals

When one MCP is switched over to another, any active feed or axis enables will be retained.

Note

Keys actuated at the moment of switchover remain operative until the new MCP is activated (by the HMI, which is subsequently activated). The override settings for feedrate and spindle also remain valid. To deactivate actuated keys, the input image of the machine control signals must be switched to nonactuated signal level on a falling edge of DB10.DBX104.0 (MCP 1 ready). The override settings should remain unchanged. Measures for deactivating keys must be implemented in the PLC user program (see example "Override Changeover").

The call is permitted only in cyclic program OB1.

Declaration of the function

```
FUNCTION_BLOCK FB9
VAR_INPUT
    Ackn:           BOOL;           // Acknowledge interrupts
    OPMixedMode:    BOOL:= FALSE;   // Mixed operation with non-M-to-N-
                                    enabled OP // deactivated
    ActivEnable:    BOOL:=TRUE;     // Not supported
    MCPEnable:      BOOL:=TRUE;     // Activate MCP switchover
END_VAR
VAR_OUTPUT
    Alarm1:         BOOL;           // Interrupt: Error in HMI bus address,
                                    bus type!
    Alarm2:         BOOL;           // Interrupt: No confirmation HMI 1
                                    offline!
    Alarm3 :        BOOL;           // Interrupt: HMI 1 is not going
                                    offline!
    Alarm4 :        BOOL;           // Interrupt: No confirmation HMI 2
                                    offline!
    Alarm5 :        BOOL;           // Interrupt: HMI 2 is not going
                                    offline!
    Alarm6 :        BOOL;           // Interrupt: Queuing HMI is not going
                                    online!
    Report :        BOOL;           // Message: Signoflife monitoring
    ErrorMMC :      BOOL;           // Error detection HMI
END_VAR
```

Description of formal parameters

Signal	Type	Type	Value range	Description
Ackn:	I	BOOL	0 (FALSE), 1 (TRUE)	Acknowledge interrupts
OPMixedMode:	I	BOOL	0 (FALSE), 1 (TRUE)	Mixed operation deactivated for OP without M:N capability
ActivEnable:	I	BOOL	0 (FALSE), 1 (TRUE)	Function is not supported. Operator panel switchover Interlocking using MMCx_SHIFT_LOCK in DB19
MCPEnable:	I	BOOL	0 (FALSE), 1 (TRUE)	Activate MCP switchover 1: MCP is switched over with operator panel. 0: MCP is not switched over with operator panel. This can be used to permanently link an MCP. See also MMCx_MSTT_SHIFT_LOCK in DB19.
Alarm1:	O	BOOL	0 (FALSE), 1 (TRUE)	Interrupt: Error in HMI bus address, bus type!
Alarm2:	O	BOOL	0 (FALSE), 1 (TRUE)	Interrupt: No confirmation HMI 1 offline!
Alarm3 :	O	BOOL	0 (FALSE), 1 (TRUE)	Interrupt: HMI 1 is not going offline!
Alarm4 :	O	BOOL	0 (FALSE), 1 (TRUE)	Interrupt: No confirmation HMI 2 offline!
Alarm5 :	O	BOOL	0 (FALSE), 1 (TRUE)	Interrupt: HMI 2 is not going offline!
Alarm6 :	O	BOOL	0 (FALSE), 1 (TRUE)	Interrupt: Queuing HMI is not going online!
Report :	O	BOOL	0 (FALSE), 1 (TRUE)	Message: Sign-of-life monitoring HMI
ErrorMMC :	O	BOOL	0 (FALSE), 1 (TRUE)	Error detection HMI

Example of calling FB9

```
CALL FB9, DB109(
Ackn            := Error_ack,       //e.g., MCP RESET
OPMixedMode     := FALSE,
ActivEnable     := TRUE,
MCPEnable       := TRUE);           // Enable for MCP switchover
```

Note

Input parameter “MCPEnable” must be set to TRUE to enable the MCP switchover. The default value of these parameters is set in this way and need not be specially assigned when the function is called.

Interrupts, errors

The output parameters "Alarm1" to "Alarm6" and "Report" exist as information in the PLC and are output in the event of M:N errors visualized on the HMI by the appearance of alarms 410900 - 410906.

If execution of an HMI function has failed (and an appropriate error message cannot be displayed), status parameter "ErrorMMC" is set to 'logical 1' (e.g., booting error, when no connection is made).

Example of a call for FB (call in the OB100)

```
CALL "RUN_UP", "gp_par" (
    MCPNum                := 1,
    MCP1In                := P#I 0.0,
    MCP1Out               := P#Q 0.0,
    MCP1StatSend          := P#Q 8.0,
    MCP1StatRec           := P#Q 12.0,
    MCP1BusAdr            := 255,             // address 1st MCP
    MCP1Timeout           := S5T#700MS,
    MCP1Cycl              := S5T#200MS,
    MCP1Stop              := TRUE,            // MCP switched off
    NCCyclTimeout         := S5T#200MS,
    NCRunupTimeout        := S5T#50S);
```

Example: Override switchover

```
// Auxiliary flags used M100.0, M100.1, M100.2, M100.3
//Edge positive of MCP1Ready must check the override
//and measures for activation
// Initiate MCP block
//This example applies to the feedrate override;
//The interface and input bytes must be exchanged for spindle override.
U     DB10.DBX104.0;      //MCP1Ready
EN    M   100.0;          // Edge memory bit 1
JCN smth1;
S     M   100.2;          // Set auxiliary memory bit 1
R     M   100.3;          //Reset auxiliary flag 2


// Save override
      L DB21.DBB4;        //Feed override interface
      T EB28;             //Buffer storage (freely input
                          // or flag byte)

wei1:
U     M   100.2;          //Switchover takes place
```

```
O    DB10.DBX104.0;      //MCP1Ready
JCN smth2;
U    DB10.DBX104.0;      //MCP1Ready
FP   M    100.1;         // Edge memory bit 2
JC smth2;
U    M    100.2;         //Switchover takes place
R    M    100.2;         //Reset auxiliary flag 1
JC smth2;
U    M    100.3;         //Comparison has taken place
SPB MCP;                 //Call MCP program
// Route the stored override to the interface of the switched MCP
// until the override values match
     L EB28;             //Buffer storage open
     T DB21.DBB4;        //Route override interface
     L EB3;              //Override input byte for feed
     <>i;                //Match?
JC smth2;                //No, jump
S    M    100.3;         //Yes, set auxiliary flag 2
// When override values match, call the MCP program again
MCP: CALL "MCP_IFM"(     //FC19
     BAGNo        := B#16#1,
     ChanNo       := B#16#1,
     SpindleIFNo  := B#16#0,
     FeedHold     := M 101.0,
     SpindleHold  := M 101.1);
wei2: NOP         0:
```

16.8 FB10: Safety relay (SI relay)

Function

The SPL function block FB10 "Safety relay" for "Safety Integrated" is the PLC equivalent of the NC function of the same name. The standard SPL "Safety relay" block is designed to support the implementation of an emergency stop function with safe programmable logic. However, it can also be used to implement other similar safety functions, e.g., control of a protective door.

The function contains 3 input parameters ("In1", "In2", "In3"). On switchover of one of these parameters to the value 0, the output "Out0" is deactivated without delay and outputs "Out1", "Out2", and "Out3" are deactivated via the parameterized timer values (parameters "TimeValue1", "TimeValue2", "TimeValue3"). The outputs are activated again without delay if inputs "In1" to "In3" take on value 1 and a positive edge change is detected at one of the acknowledgment inputs "Ack1", "Ack2".

Call bit memory

To bring the outputs to their basic setting (values = 0) after booting, the parameter "FirstRun" must be configured as follows. The parameter "FirstRun" must be switched to the value TRUE

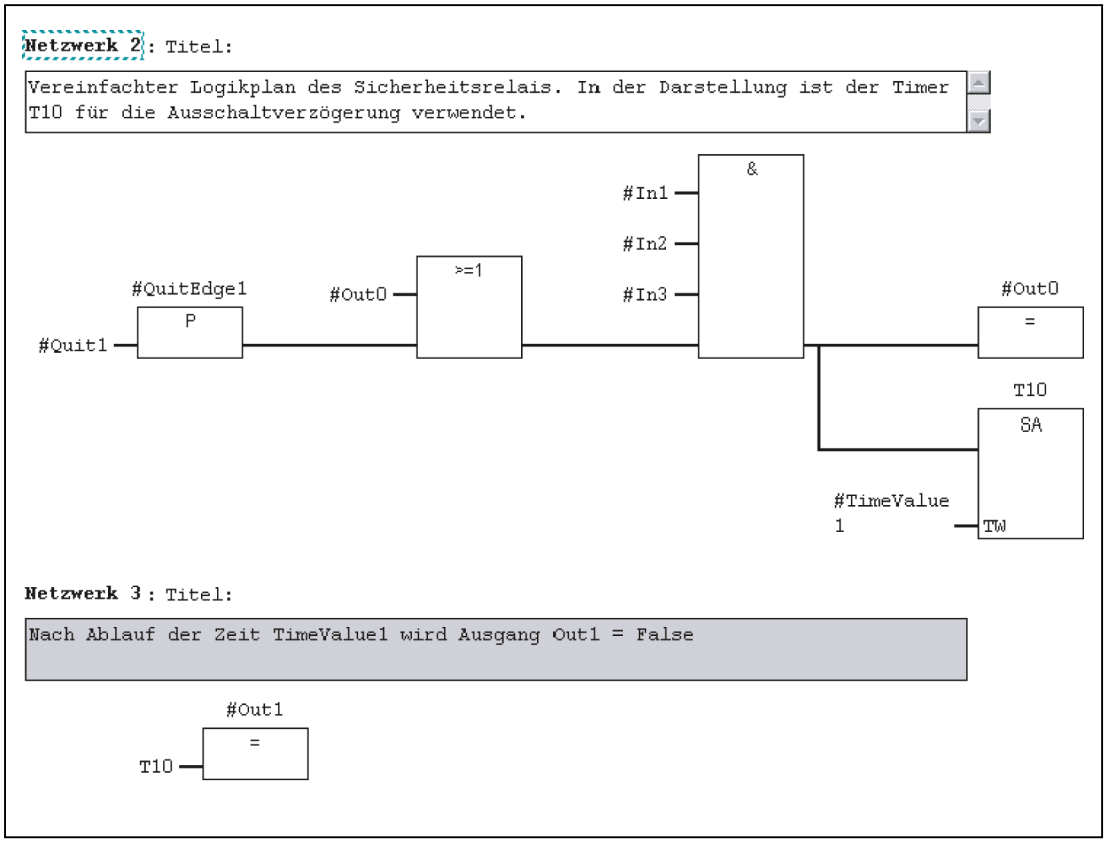
via a retentive data (memory bit, bit in data block) on the 1st run after control booting. This data can be preset, e.g., in OB100. The parameter "FirstRun" is reset to FALSE when FB10 is executed for the first time. Separate data must be used for parameter "FirstRun" for each call with its own instance.

General conditions

- FB10 has multi-instance capability
- After the start of the SPL program, FB10 is called once per SI relay in the cyclic part of the PLC basic program (OB1).
- Every call of FB10 must be assigned a separate instance DB from the user area.

Simplified block diagram in CSF

The figure below shows only one acknowledgment input "Ack1" and one delayed deactivation output "Out1". The circuit for "Ack2" and the other delayed outputs are identical. The parameter "FirstRun" is also missing in the function diagram. The mode of operation is described above.



Declaration of the function

```
FUNCTION_BLOCK FB10
VAR_INPUT
    In1 :           BOOL;
```

```
    In2 :           BOOL;
    In3 :           BOOL;
    Ackn1 :         BOOL;
    Ackn2 :         BOOL;
    TimeValue1 :    TIME;
    TimeValue2 :    TIME;
    TimeValue3 :    TIME;
END_VAR
VAR_OUTPUT
    Out0 :          BOOL;
    Out1 :          BOOL;
    Out2 :          BOOL;
    Out3 :          BOOL;
END_VAR
VAR_INOUT
    FirstRun:       BOOL;
END_VAR
```

Description of formal parameters

Parameter	Type	Type	Value range	Description
In1:	I	BOOL	0 (FALSE), 1 (TRUE)	Input 1
In2:	I	BOOL	0 (FALSE), 1 (TRUE)	Input 2
In3:	I	BOOL	0 (FALSE), 1 (TRUE)	Input 3
Ackn1 :	I	BOOL	0 (FALSE), 1 (TRUE)	Acknowledgment input 1
Ackn2 :	I	BOOL	0 (FALSE), 1 (TRUE)	Acknowledgment input 2
TimeValue1 :	I	S5TIME		Time value 1 for OFF delay
TimeValue2 :	I	S5TIME		Time value 2 for OFF delay
TimeValue3 :	I	S5TIME		Time value 3 for OFF delay
Out0 :	O	BOOL	0 (FALSE), 1 (TRUE)	Output, instantaneous (no delay)
Out1 :	O	BOOL	0 (FALSE), 1 (TRUE)	Output, delayed by TimeValue1
Out2 :	O	BOOL	0 (FALSE), 1 (TRUE)	Output, delayed by TimeValue2
Out3 :	O	BOOL	0 (FALSE), 1 (TRUE)	Output, delayed by TimeValue3
FirstRun:	I/O	BOOL	0 (FALSE), 1 (TRUE)	Call bit memory

16.9 FB11: Brake test

Function

The braking operation check should be used for all axes, which must be prevented from moving in an uncontrolled manner by a holding brake. "Vertical axes" are the main application.

The machine manufacturer can use the PLC user program to regularly close the brake at a suitable instant (e.g. every 8 hours) and have the drive exert torque/force in addition to the weight of the axis. In errorfree operation, the brake can produce the necessary braking torque/ braking force. The axis will hardly move during this.

When an error occurs, the actual position value exits the parameterizable monitoring window. The position controller prevents sagging of the axis. The function test of the brake mechanical system is negatively acknowledged.

Further information

A detailed description of the parameterization of the NC and drive within the framework of the "Safety Integrated" function can be found in:

Function Manual for Safety Integrated

Starts the brake test

The brake test must always be started when the axis is at a standstill. For the entire duration of the brake test, the enable signals of the parameterized axis must be set to enable (e.g. controller inhibit, feed enable). Furthermore, the signal at the axis/spindle DB31, ...DBX28.7 (PLC-controlled axis) is to be set to status 1 by the user program for the complete duration of the test.

Before activating the NC/PLC interfaces DB31, ...DBX28.7 (axis controlled from the PLC program), the axis must be switched as "neutral axis". For example, DB31, ...DBX8.0 - 8.3 (assign NC axis channel) can be set to channel 0 - and DB31, ...DBX8.4 can be set (activation signal when this byte changes).

The return message:

- about the current status can be queried in DB31, ... DBB68.
- the Nc via the signal DB31, ...DBX63.1 (PLC controls axis) is to be awaited before the block is started. The direction in which the drive must produce its torque/force is specified by the PLC in the form of a "traversing motion" (e.g., via FC18).

The axis must be able to reach the destination of this movement without risk of collision if the brake is unable to produce the necessary torque/force.

Note

Note on FC18

If FC18 is called over the further course of the user program for the same axis, the calls must be mutually interlocked. For example, this can be achieved via a common call of this function with an interlocked common data interface for the FC18 parameters. A 2nd option is to call the FC18 repeatedly, in which case the inactive FC18 will not be processed by the program. A multiple-use interlock must be provided.

Structure of a brake test

Step		Expected feedback	Monitoring time value
1	Start brake test	DBX71.0 = 1	TV_BTactiv
2	Close brake	Bclosed = 1	TV_Bclose
3	Output traversing command	DBX64.6 OR DBX64.7	TV_FeedCommand

Step		Expected feedback	Monitoring time value
4	Issue test travel command	DBX62.5 = 1	TV_FXSreached
5	Wait for the holding time	DBX62.5 = 1	TV_FXShold
6	Deselect brake test / open brake	DBX71.0 = 0	TV_BTactiv
7	Output test ok	---	---

Boundary conditions

- FB2 has multi-instance capability.
- Every call of FB11 must be assigned a separate instance DB from the user area.

Declaration of the function

```
Function_BLOCK FB11
VAR_INPUT
    Start             BOOL;
    Ackn :            BOOL;
    Bclosed :         BOOL;
    Axis :            INT;
    TimerNo :         TIMER ;
    TV_BTactiv :      S5TIME ;
    TV_Bclose :       S5TIM;
    TV_FeedCommand :  S5TIME ;
    TV_FXSreached :   S5TIME ;
    TV_FXShold :      S5TIME ;
END_VAR
VAR_OUTPUT
    CloseBrake : BOOL;
    MoveAxis :   BOOL;
    Done :       BOOL;
    Error :      BOOL;
    State :      BYTE ;
END_VAR
```

Description of formal parameters

Signal	Type	Type	Meaning
Start	I	BOOL	Starts the brake test
Ackn	I	BOOL	Acknowledge fault
Bclosed	I	BOOL	Checkback input whether Close Brake is activated (singlechannel - PLC)
Axis	I	INT	Axis number of axis to be tested
TimerNo	I	TIMER	Timer from user program

Signal	Type	Type	Meaning
TV_BTactiv	I	S5TIME	Monitoring time value → brake test active, check of axis signal DBX71.0
TV_Bclose	I	S5TIME	Monitoring time value → close brake. Check of input signal Bclosed after output CloseBrake has been set
TV_FeedCommand	I	S5TIME	Monitoring time value → Travel command given Check travel command after MoveAxis has been set
TV_FXSreched	I	S5TIME	Monitoring time value → fixed stop reached
TV_FXShold	I	S5TIME	Monitoring time value → test brake
CloseBrake	O	BOOL	Request, close brake
MoveAxis	O	BOOL	Request, initiate traversing motion
Done	O	BOOL	Test successfully completed
Error	O	BOOL	An error occurred
State	O	BYTE	See paragraph "Error identifiers"

Fault IDs

State	Meaning
0	No error
1	Start conditions not fulfilled, e.g. the axis is not in closed-loop control / brake closed / axis inhibited
2	No NC checkback in "Brake test active" signal on selection of brake test
3	No "Brake applied" checkback by input signal Bclosed
4	No travel command output (e.g., axis motion has not been started)
5	Fixed end stop will not be reached → axis RESET was initiated.
6	Traversing inhibit/Approach too slow → fixed stop cannot be reached. TV FXSreached monitoring timeout.
7	Brake is not holding at all (the end position is reached)/approach speed is too high
8	Brake opens during the holding time
9	Error when deselecting the brake test
10	Internal error
11	"PLC-controlled axis" signal not enabled in the user program

Example of calling FB11:

```
UN    M       111.1;   //Request to close brake, Z axis of FB
=     O       85.0;    //Brake control, Z axis
AUF           Axis3";  //Brake test, Z axis
O     I       73.0;    //Brake test trigger, Z axis
O     M       110.7;   //Brake test running
FP    M       110.0;
UN    M       111.4;   //Error has occurred
S     M       110.7;   //Brake test running
S     M       110.6;   //Next step
```

```
        JCN   m001
        L     DBB      68;
        AW    W#16#F;
        T     MB       115;     //flag channel state
        L     B#16#10
        T     DBB      8;       //Request neutral axis

m001:   U     DBX      68.6;    //Checkback signal, axis is neutral
        U     M        110.6;
        FP    M        110.1;
        R     M        110.6;
        S     M        110.5;   //Next step
        S     DBX      28.7;    //Request PLC-monitored axis

        U     DBX      63.1;    //Checkback signal, axis monitored by PLC
        U     M        110.5;
        FP    M        110.2;
        R     M        110.5;
        S     M        111.0;   //Start brake test for FB

        CALL FB11, DB211 (//brake test block
             Start            :=M   111.0,   //Start brake test
             Ackn             := I  3.7,     //Acknowledge error with RESET
                                             key
             Bclosed          := I  54.0,    //Return message close brakes
                                             //controlled
             Axis             := 3,          //Axis number of axis to be
                                             tested
                                             //Z axis
             TimerNo          := T  110,     //Timer number
             TV_BTactiv       := S5T#200MS,  //Monitoring time value:
                                             //Brake test active DBX71.0
             TV_Bclose        := S5T#1S,     //Monitoring time value:
                                             //Brake closed
             TV_FeedCommand   := S5T#1S,     //Monitoring time value:
                                             //Traversing command output
             TV_FXSreache     := S5T#1S,     //Monitoring time value:
                                             //Fixed stop reached
             TV_FXShold       := S5T#2S,     //Monitoring time value:
                                             //Brake test time
             CloseBrake       :=M   111.1,   //Request to close brake
             MoveAxis         :=M   111.2,   Initiate //Request traversing
                                             motion //
             Done             :=M   111.3,   //Test successfully completed
             Error            :=M   111.4,   //Error has occurred
             State            := MB 112);    //Error status
```

```
      AUF           "Axis3"  //Brake test, Z axis
                    ;

      U     M       111.2;   //Moveaxis
      FP    M       111.5;   //FC18 Start
      S     M       111.7;   //Start FC18

      O     M       111.3;   //Test successfully completed
      O     M       111.4;   //Error has occurred
      FP    M       110.3;
      R     DBX     28.7;    //Request, PLC-monitored axis

      UN    DBX     63.1;    //Checkback signal, axis monitored by PLC
      U     M       111.0;   //Start brake test for FB
      U     M       110.7;   //Brake test running
      FP    M       110.4;
      R     M       111.0;   //Start brake test for FB
      R     M       110.7;   //Brake test running

//optional begin
      JCN   m002;
      L     MB      115;     //old channel status
      OW    W#16#10
            ;
      T     DBB     8;       //Request channel axis
m002: NOP   0;
//optional end

      CALL "SpinCtrl" (//Traverse Z axis
            Start   :=M     111.2,    //Start traversing motion
            Stop    := FALSE,
            Funct   := B#16#5,        //Mode: Axis mode
            Mode    := B#16#1,        //Procedure: Incremental
            AxisNo  := 3,             //Axis number of axis to be
                                      traversed
                                      //axis Z-axis
            Pos     := -5.000000e+000, //Traversing distance: Minus 5 mm
            FRate   := 1.000000e+003,  //Feedrate: 1000 mm/min
            InPos   :=M     113.0,    //Position reached
            Error   :=M     113.1,    //Error has occurred
            State   := MB   114);     //Error status

      AUF           "Axis3"  //Brake test, Z axis
                    ;
      U     M       113.0;   //Position reached
```

```
O     M       113.1;   //Error has occurred
FP    M       113.2;
R     M       111.7;   //Start FC18
```

16.10 FB29: Signal recorder and data trigger diagnostics

Function

Signal recorder

FB29 "Diagnostics" allows various diagnostic routines to be performed on the PLC user program. A diagnostic routine logs signal states and signal changes. In this diagnostic routine, function number 1 is assigned to the "Func" parameter. Up to 8 signals of the parameters "Signal_1" to "Signal_8" are recorded in a ring buffer each time one of the signals changes. The current information of parameters "Var1" as BYTE value, and "Var2" and "Var3" as INTEGER values are also stored in the ring buffer.

The number of past OB1 cycles is also stored in the buffer as additional information. This information enables the graphical evaluation of signals and values in OB1 cycle grid.

Call rule

First call of FB29 in OB1 cycle: Parameter "NewCycle" = 1

All other calls of FB29 in the same OB1 cycle: Parameter "NewCycle" = 0

Ring buffer

The ring buffer, which must be defined by the user, must have an ARRAY structure specified as in the source code. The array can have any number of elements. A size of 250 elements is recommended. The "ClearBuf" parameter is used to clear the ring buffer and set the "BufAddr" pointer to the start. The instance DB related to the FB29 is a DB from the user area and is to be transferred to the FB Diagnostics with the parameter "BufDB".

Data trigger

The data trigger function is intended to allow triggering on specific values (or bits) at any permissible memory cell. The cell to be triggered is "rounded" with a bit mask ("AndMask" parameter) before the "TestVal" parameter is compared in the diagnostic block.

Note

The source code for the function is available in the source container of the basic-program library under the name "Diagnose.awl". The instance DB and the ring buffer DB are also defined in this source block. The function call is also described in the function. The DB numbers and the call must be modified.

Declaration of the function

```
FUNCTION_BLOCK FB29
VAR_INPUT
Func               : INT;
    Signal_1       : BOOL;
    Signal_2       : BOOL;
    Signal_3       : BOOL;
    Signal_4       : BOOL;
    Signal_5       : BOOL;
    Signal_6       : BOOL;
    Signal_7       : BOOL;
    Signal_8       : BOOL;
    NewCycle       : BOOL;
    Var1           : BYTE ;
    Var2           : INT;
    Var3           : INT;
    BufDB          : INT;
    ClearBuf       : BOOL;
    DataAdr        : POINTER;
    TestVal        : WORD;
    AndMask        : WORD;
END_VAR
VAR_OUTPUT
    TestIsTrue     : BOOL;
END_VAR
VAR_IN_OUT
    BufAddr        : INT;
END_VAR
```

Structure of the ring buffer

```
TITLE =
                                        //Ring buffer DB for FB29
VERSION : 1.0

STRUCT
   Field: ARRAY [0 .. 249 ] OF          //can be any size of this struct
   STRUCT

   Cycle : INT;                         //Delta cycle to last storage in buffer
   Signal_1 : BOOL;                     //Signal names same as FB29
   Signal_2 : BOOL;
   Signal_3 : BOOL;
```

```
   Signal_4 : BOOL;
   Signal_5 : BOOL;
   Signal_6 : BOOL;
   Signal_7 : BOOL;
   Signal_8 : BOOL;
   Var1 : BYTE ;
   Var2 : WORD;
   Var3 : WORD;
   END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK
```

Description of formal parameters

Signal	Type	Type	Value range	Description
Func:	I	INT	0, 1, 2	Function 0: Switch off 1: Signal recorder 2: Data trigger
Parameters for function 1				
Signal_1 ... Signal_8:	I	BOOL	0 (FALSE), 1 (TRUE)	Bit signals checked for change
NewCycle:	I	BOOL	0 (FALSE), 1 (TRUE)	See paragraph "Function" above
Var1 :	I	BYTE		Additional value
Var2 :	I	INT		Additional value
Var3 :	I	INT		Additional value
BufDB:	I	INT		Ring buffer DB no.
ClearBuf:	I	BOOL	0 (FALSE), 1 (TRUE)	Delete ring buffer DB and reset pointer BufAddr
BufAddr:	I/O	INT		Target area for read data
Parameters for function 2				
DataAdr:	I	POINTER		Pointer to word to be tested
TestVal:	I	WORD		Comparison value
AndMask:	I	WORD		See paragraph "Function" above
TestIsTrue:	O	BOOL	0 (FALSE), 1 (TRUE)	Result of comparison

Configuration steps

- Select function of diagnostics block.
- Define suitable data for the recording as signal recorder or data triggering.
- Find a suitable point or points in the user program for calling the diagnostics FB.

- Create a data block for the ring buffer, see call example.
- Call the diagnostics FB with parameters in the user program.

In function 1, it is advisable to clear the ring buffer with the "ClearBuf" parameter. When the recording phase with function 1 is completed, read out the ring buffer DB in STEP7 with the function "opening the data block in the data view". The content of the ring buffer DB can now be analyzed.

Call example

```
FUNCTION FC99: VOID
TITLE =
VERSION : 0.0

BEGIN
NETWORK
TITLE = NETWORK

CALL FB29, DB80(
Func                := 1,
        Signal_1    :=M         100.0,
        Signal_2    :=M         100.1,
        Signal_3    :=M         100.2,
        Signal_4    :=M         100.3,
        Signal_5    :=M         10.4,
        Signal_6    :=M         100.5,
        Signal_7    :=M         100.6,
        Signal_8    :=M         100.7,
        NewCycle    := TRUE,
        Var1        := MB       100,
        BufDB       := 81,
        ClearBuf    :=M         50.0);
END_FUNCTION
```

16.11 FC2 : GP_HP - basic program, cyclic section

Function

The NC/PLC interface is processed by the basic program in cyclic mode (OB1). To keep the runtime to a minimum, only the control and state signals are cyclically transferred. The transfer of auxiliary and G commands is processed only on request from the NC.

Furthermore, the data for handwheel selection, modes, and other operating signals are transferred from the operator panel (HMI) to the NC/PLC interface so that the modes support the selection from the MCP or HMI as required.

The transfer of HMI signals to the NC/PLC interface can be deactivated by setting the value of the parameter "MMCToIF" to "FALSE" in FB1 (DB7).

Handwheel selection signals

Requirement: FB1, parameter "HWheelMMC == "TRUE"

The handwheel selection signals from the HMI are decoded and activated in the respective machine or geometry axis of the respective handwheel.

Declaration

```
FUNCTION FC2: VOID
// No parameters
```

Call example

As far as the time is concerned, the basic program must be executed **before** the user program. It is, therefore, called first in OB1.

The following example contains the standard declarations for OB1 and the calls for the basic program (FC2), the transfer of the MCP signals (FC19), and the acquisition of error and operational messages (FC10).

```
ORGANIZATION_BLOCK OB1
VAR_TEMP
   OB1_EV_CLASS :    BYTE ;
   OB1_SCAN_1 :      BYTE ;
   OB1_PRIORITY :    BYTE ;
   OB1_OB_NUMBR :    BYTE ;
   OB1_RESERVED_1 :  BYTE ;
   OB1_RESERVED_2 :  BYTE ;
   OB1_PREV_CYCLE :  INT;
   OB1_MIN_CYCLE :   INT;
   OB1_MAX_CYCLE :   INT;
   OB1_DATE_TIME :   DATE_AND_TIME;
END_VAR
BEGIN
CALL FC2;                  // Call basic program as 1st FC
//INSERT USER PROGRAM HERE
CALL FC19(                  //MCP signals to interface
BAGNo :=                    B#16#1,    // Mode group no. 1
ChanNo :=                   B#16#1,    // Channel no. 1
SpindleIFNo :=              B#16#4,    //Spindle interface number = 4
                                       // (Number of the associated machine
                                       axis)
FeedHold :=                 m22.0,     //Feed stop signal
                                       //Modal
SpindleHold :=  db2.dbx151.0);         //Spindle stop modal
```

```
                                        //in message DB
CALL FC10(                              // Error and operational messages
               ToUserIF := TRUE,        //Signals transferred from DB2
                                        //to interface
               Ackn :=     I6.1);       //Acknowledgment of error messages
                                        //via I6.1
END_ORGANIZATION_BLOCK
```

16.12 FC3: GP_PRAL - basic program, interruptdriven section

Function

Block-synchronized transfers from the NC to the PLC (auxiliary and G commands) are processed in the alarm-driven part of the basic program. **Auxiliary functions** are subdivided into normal and highspeed auxiliary functions.

The highspeed functions of an NC block are buffered and the transfer acknowledged to the NC. These are transferred to the application interface at the start of the next OB1 cycle.

Highspeed auxiliary functions programmed immediately one after the other, are not lost for the user program. This is ensured by a mechanism in the basic program.

Normal auxiliary functions are acknowledged to the NC only after one completed cycle duration. This allows the application to issue a read disable to the NC.

The **G commands** are evaluated immediately and passed to the application interface.

NC process alarms

If the interrupt is triggered by the NC (possible in each IPO cycle), a bit in the local data of OB40 ("GP_IRFromNCK") is set by the basic program, only when the FB1 parameter "UserIR" is TRUE. This data is not set on other events (process alarms through I/Os). This information makes it possible to branch into the associated interrupt routine in the user program in order to initiate the necessary action.

To be able to implement highspeed, jobdriven processing of the user program for the machine, the following NC functions are available in the interrupt processing routine (OB40 program section) for the PLC user program:

- Selected **auxiliary functions**
- **Tool-change function** for tool-management option
- **Position reached** for positioning axes, indexing axes and spindles with activation via PLC

The functions listed above can or must be evaluated by the user program in OB40 in order to initiate reactions on the machine. As an example, the revolver switching mechanism can be activated when a T command is programmed on a turning machine.

For further details on programming process alarms (time delay, interruptibility, etc.) refer to the corresponding SIMATIC documentation.

Auxiliary functions

Generally, high-speed or acknowledging auxiliary functions are processed with or without interrupt control independently of any assignment.

Basic-program parameters in FB1 can be set to define which auxiliary functions (T, H, DL) must be processed solely on an interruptdriven basis by the user program.

Functions which are not assigned via interrupts are only made available by the cyclic basic program as in earlier versions. The change signals of these functions are available in a PLC cycle.

Even if the selection for the auxiliary function groups (T, H, DL) is made using interrupt control, only one interrupt can be processed by the user program for the selected functions.

A bit is set for a specific channel in the local data "GP_AuxFunction" for the user program (if "GP_AuxFunction[1]" is set, an auxiliary function is available for the 1st channel).

In the related channel-DB the change signal and the function value are available for the user. The request signal of this interrupt-driven function is reset to zero in the cyclic basic program section after the execution of at least one full OB1 cycle (max. approx. two OB1 cycles).

Tool change

With the tool-management option, the tool-change command for revolver and the tool change in the spindle is supported by an interrupt. The local data bit "GP_TM" in OB40 is set for this purpose. The PLC user program can thus check the tool management DB (DB72 or DB73) for the tool change function and initiate the tool change operation.

Position reached

In the bit structure, "GP_InPosition" of the local data of OB40 is specific to the machine axis (each bit corresponds to an axis/spindle, e.g. GP_InPosition[5] corresponds to the 5th axis).

If a function has been activated via FC18 (spindle control, positioning axis, indexing axis) for an axis or spindle, the associated "GP_InPosition" bit can be used to implement instantaneous evaluation of the "InPos" signal of the FCs listed above. This feature can be used, for example, to obtain immediate activation of clamps for an indexing axis.

Declaration

```
FUNCTION FC3: VOID
// No parameters
```

Call example

As far as the time is concerned, the basic program must be executed **before** other alarm-driven user programs. It is, therefore, called first in OB40.

The following example contains the standard declarations for OB40 and the call for the basic program.

```
ORGANIZATION_BLOCK OB40
VAR_TEMP
   OB40_EV_CLASS :      BYTE ;
   OB40_STRT_INF :      BYTE ;
   OB40_PRIORITY :      BYTE ;
   OB40_OB_NUMBR :      BYTE ;
   OB40_RESERVED_1 :    BYTE ;
   OB40_MDL_ID :        BYTE ;
   OB40_MDL_ADDR :      INT;
   OB40_POINT_ADDR :    DWORD;
   OB40_DATE_TIME :     DATE_AND_TIME;

//Assigned to basic program
GP_IRFromNCK : BOOL;                    //Interrupt by NC for user
GP_TM : BOOL;                           //Tool management
GP_InPosition : ARRAY [1..3] OF BOOL;   //Axis-oriented for positioning,
                                        //Indexing axes, spindles
GP_AuxFunction : ARRAY [1..10] OF BOOL; //Channel-oriented for auxiliary
                                        functions
GP_FMBlock : ARRAY [1..10] OF BOOL;     //Currently not used
//Further local user data may be defined from this point onwards
END_VAR
BEGIN
   CALL FC3;
   //INSERT USER PROGRAM HERE
END_ORGANIZATION_BLOCK
```

16.13 FC5: GP_DIAG - basic program, diagnostic alarm and module failure

Function

The block FC5 "GP_DIAG" is used to record assembly disruptions and failures.

A PLC stop can be triggered via the parameter "PlcStop". The PLC stop is only triggered for incoming events. The MCPs connected to the parameterized PROFIBUS (DP1) at FB1 are excluded from this.

Declaration

```
FUNCTION FC5: VOID
   VAR_INPUT
```

```
      PlcStop: BOOL:= TRUE;
   END_VAR
```

Call example

The basic program should be run through after processing of the user programs. This is recommended because a PLC stop can be triggered by the FC5.

The example contains the standard declaration for OB82 and OB86 and the call of the FC5.

```
ORGANIZATION_BLOCK OB82
VAR_TEMP
     OB82_EV_CLASS : BYTE ;
     OB82_FLT_ID : BYTE ;
     OB82_PRIORITY : BYTE ;
     OB82_OB_NUMBR : BYTE ;
     OB82_RESERVED_1 : BYTE ;
     OB82_IO_FLAG : BYTE ;
     OB82_MDL_ADDR : INT ;
     OB82_MDL_DEFECT : BOOL;
     OB82_INT_FAULT : BOOL;
     OB82_EXT_FAULT : BOOL;
     OB82_PNT_INFO : BOOL;
     OB82_EXT_VOLTAGE : BOOL;
     OB82_FLD_CONNCTR : BOOL;
     OB82_NO_CONFIG : BOOL;
     OB82_CONFIG_ERR : BOOL;
     OB82_MDL_TYPE : BYTE ;
     OB82_SUB_NDL_ERR : BOOL;
     OB82_COMM_FAULT : BOOL;
     OB82_MDL_STOP : BOOL;
     OB82_WTCH_DOG_FLT : BOOL;
     OB82_INT_PS_FLT : BOOL;
     OB82_PRIM_BATT_FLT : BOOL;
     OB82_BCKUP_BATT_FLT : BOOL;
     OB82_RESERVED_2 : BOOL;
     OB82_RACK_FLT : BOOL;
     OB82_PROC_FLT : BOOL;
     OB82_EPROM_FLT : BOOL;
     OB82_RAM_FLT : BOOL;
     OB82_ADU_FLT : BOOL;
     OB82_FUSE_FLT : BOOL;
     OB82_HW_INTR_FLT : BOOL;
     OB82_RESERVED_3 : BOOL;
     OB82_DATE_TIME : DATE_AND_TIME;
END_VAR
```

```
     BEGIN
        CALL FC5
            (PlcStop := FALSE) ;
END_ORGANIZATION_BLOCK

ORGANIZATION_BLOCK OB86
VAR_TEMP
     OB86_EV_CLASS : BYTE ;
     OB86_FLT_ID : BYTE ;
     OB86_PRIORITY : BYTE ;
     OB86_OB_NUMBR : BYTE ;
     OB86_RESERVED_1 : BYTE ;
     OB86_RESERVED_2 : BYTE ;
     OB86_MDL_ADDR : WORD;
     OB86_RACKS_FLTD : ARRAY [0 .. 31]OF BOOL;
     OB86_DATE_TIME : DATE_AND_TIME;
END_VAR
     BEGIN
        CALL FC5
            (PlcStop := TRUE) ;
END_ORGANIZATION_BLOCK
```

16.14 FC6: TM_TRANS2 - transfer block for tool management and multitool

Function

The block FC6 "TM_TRANS2" is used for position changes of the tools, state changes, and multitool.

The FC6 block has the same functionality as the FC8 block, plus the **multitool** functionality.

The description of FC6 only contains the **multitool** functionality.

The functionality of FC8 is described in "FC8: TM_TRANS - transfer block for tool management (Page 238)".

Declaration of the function

```
FUNCTION FC6: VOID
VAR_INPUT
   Start:           BOOL;
   TaskIdent:       BYTE ;
   TaskIdentNo:     BYTE ;
   NewToolMag:      INT;
   NewToolLoc:      INT;
   OldToolMag:      INT;
```

```
   OldToolLoc:       INT;
   Status:           INT;
   MtoolPlaceNum:    INT;
END_VAR
VAR_OUTPUT
   Ready:            BOOL;
   Error:            INT;
END_VAR
BEGIN
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Description
Start:	I	BOOL	0 (FALSE), 1 (TRUE)	See block description FC8
TaskIdent:	I	BYTE		See block description FC8
TaskIdentNo:	I	BYTE		See block description FC8
NewToolMag:	I	INT		See block description FC8
NewToolLoc:	I	INT		See block description FC8
OldToolMag:	I	INT		See block description FC8
OldToolLoc:	I	INT		See block description FC8
Status:	I	INT		See block description FC8
MtoolPlaceNum:	**I**	**INT**		**Multitool location No.**
Ready:	O	BOOL	0 (FALSE), 1 (TRUE)	See block description FC8
Error:	O	INT		See block description FC8

16.15 FC7: TM_REV - transfer block for tool change with revolver

Function

After a revolver has been changed, the user calls block FC7 "TM_REV". The revolver number corresponding to interface number in DB73 must be specified in parameter "ChgdRevNo" for this purpose. As this block is called, the associated "Interface active" bit in data block DB73.DBW0 of FC7 is reset after parameter "Ready" == TRUE is returned.

Job executed correctly

If the job was executed correctly, then "Ready" == 1. The user must then set the parameter "Start" = 0 or no longer call FC7.

Job executed with errors

If the job was executed with errors, then parameter "Ready" == 0 and parameter "Error" == 1. The job must be repeated in the next PLC cycle. Since the parameter "Start" does not need a

positive edge for a subsequent job, "Start" = 1 remains, because the job has not yet been completed. See "Call example" and "Pulse diagram" below.

Boundary conditions

- Block FC7 may only be started with parameter "Start" = 1 if an activation signal for the associated interface (DB73.DBW0) for this transfer has been supplied by the tool management function.
- A cancellation of a transfer, e.g. by a channel reset, is not permitted.
- Parameter "Start" = 1, until parameter "Ready" == 1 or "Error" == 1

Further information

- For detailed information about tool management, refer to the Function Manual Tool Manager.
- PI services for tool management, see:
  - FB4: PI_SERV - request PI service (Page 173)
  - FC8: TM_TRANS - transfer block for tool management (Page 238)
  - FC22: TM_DIR - direction selection for tool management (Page 283)

Manual revolver switching

If the revolver is rotated in manual operation, neither a tool change nor an offset selection is associated with this operation. The first step is the removal of the tool from the toolholder back to its location in the revolver. An asynchronous transfer must be performed with FC8 (alternative: FC6). The associated parameter settings are shown below:

```
TaskIdent = 4
TaskIdentNo = Channel no.
NewToolMag = Magazine no. of the revolver
NewToolLoc = Original location of the tool
OldToolMag = Magazine no. of the buffer storage (spindle) = 9998
OldToolLoc = Buffer storage no. of the spindle
Status = 1
```

If the revolver is now turned to an arbitrary position at which a tool is located, this tool must be activated. This is done easiest by the new T programming in the part program. However, if this should take place, for example, at the end of revolver switching from the PLC user program, an ASUP must be started for this purpose. The current revolver position must be transferred to the ASUP. In this way, the tool at this location is determined in the ASUP and is selected (see Jobshop example in the toolbox).

Declaration of the function

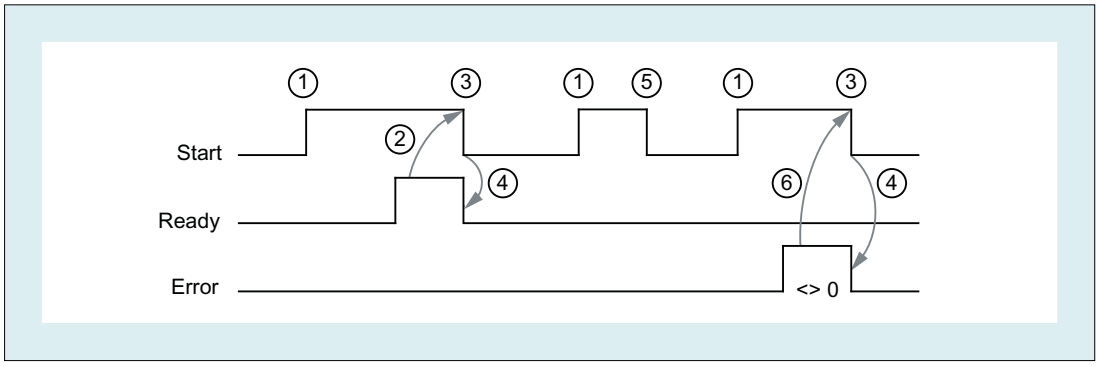
```
FUNCTION FC7:       VOID
//NAME :TM_REV
VAR_INPUT
    Start :         BOOL;
```

```
    ChgdRevNo :     BYTE ;
END_VAR
VAR_OUTPUT
    Ready :         BOOL;
    Error :         INT;
END_VAR
BEGIN
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
Start:	I	BOOL	0 (FALSE), 1 (TRUE)	1 = start transfer
ChgdRevNo:	I	BYTE	1, 2, 3, ...	Number of revolver interface
Ready:	O	BOOL	0 (FALSE), 1 (TRUE)	1 = Transfer complete
Error:	O	INT	0, 1 , 2, 3	Error checkback 0: No error has occurred 1: No revolver present 2: Illegal revolver number in parameter "ChgdRev-No" 3: Illegal job ("interface active" signal for selected revolver = "FALSE")

Pulse diagram



① User: Set request, Start = 0 → 1
② FB4: PI service successfully completed, Ready = 1
User: Reset request, IF Ready == 1 THEN Start = 0
③ User: IF Ready == 1 THEN reset request: 1 → 0
④ FB4: Reset job confirmation, Ready = 0
⑤ User: IF Ready == 0 AND Error == 0 THEN reset request Start = 1 → 0 **not permissible**
⑥ FB4: PI service completed with errors, Error = 1
User: Reset request, IF Ready == 1 OR Error == 1 THEN Start = 0, possible further error handling

Call example

```
CALL     // Tool management: Transfer block for revolver
FC7(
    Start :=          m 20.5,         // Start := "1 " => initiate the
                                      transfer
    ChgdRevNo :=      DB61.DBB1,
    Ready :=          m 20.6,
    Error :=          DB61.DBW12
};
u       m 20.6;                       // Poll ready
r       m 20.5;                       // Reset start
spb     m001;                         // Jump, if everything OK
l       db61.dbw 12;                  // Error information
ow      w#16#0;                       // Evaluate error
spn     error;                        // Jump to troubleshooting, if <> 0
m001:                                 // Start of another program
error :
r       m 20.5;                       // Start reset, if an error has
                                      occurred
```

16.16 FC8: TM_TRANS - transfer block for tool management

Function

The user calls this block FC TM-TRANS when the position of the tool or the status of the transfer operation changes. The parameter "TaskIdent" specifies the transfer job for the block FC8 at the tool management interface:

- For loading/unloading positions
- For spindle change positions
- For revolver change positions as transfer identifier
- Asynchronous transfer
- Asynchronous transfer with location reservation

The interface number is indicated in parameter "TaskIdentNo".

Example for loading point 5:

Parameter "TaskIdent":= 1 and "TaskIdentNo":= 5.

Furthermore, the **current** tool positions and status data (list of "Status" parameter in the following text) are also transferred for this transfer function.

Note

FC8 informs the NC of the current positions of the old tool.

The NC knows where the old and the new tool were located until the position change.

In the case of a transfer without a socalled "old tool" (e.g. on loading), the value 0 is assigned to parameters "OldToolMag", "OldToolLoc".

Block FC TM_TRANS may be started only with "Start" parameter = "TRUE" if an activation signal for the appropriate interface (DB71, DB72, DB73 in word 0) for this transfer has been supplied by the tool management function.

When this job is executed correctly, the output parameter "Ready" contains the value TRUE.

The user must then set the "Start" parameter to FALSE or not call the block again.

If the "Ready" parameter = FALSE, the error code in the "Error" parameter must be interpreted (see Call example FC8 and pulse diagram).

If the error code = 0, then this job must be repeated in the next PLC cycle (e.g. "Start" remains set to "TRUE"). This means that the transfer operation has not yet been completed.

If the user assigns a value of less than 100 to the "Status" parameter, then the associated interface in data block DB71 or DB72 or DB73, word 0 is deactivated (process completed). The appropriate bit for the interface is set to 0 by FC8.

The "Start" parameter does not need a signal edge for a subsequent job. This means that new parameters can be assigned with "Start = TRUE" immediately when "Ready = TRUE" is received.

Asynchronous transfer

To ensure that changes in the position of a tool are automatically signaled from the PLC to the tool management (e.g. power failure during an active command or independent changes in the position by the PLC), FC8 is called with "TaskIdent" = 4 or 5. This call does not require interface activation by the tool management.

If parameter "TaskIdent" = 5, the tool management reserves the location in addition to changing the position. The location is only reserved if the tool has been transported from a real magazine to a buffer storage.

A relevant NC channel must be parameterized in the "TaskIdentNo" parameter.

The previous location of the tool is specified in the parameters "OldToolMag" and "OldToolLoc". The current location of the tool is specified in the parameters "NewToolMag", "NewToolLoc". "Status" = 1 must be specified.

With "Status" = 5, the specified tool remains at location "OldToolMag", "OldToolLoc". This location must be a buffer (e.g. spindle). The real magazine and location must be specified in the parameters "NewToolMag", "NewToolLoc"; the location is at the position of the buffer. This procedure must always be used if the tool management is to be informed of the position of a specific magazine location. This procedure is used for alignment in search strategies.

Boundary conditions

- A cancellation of a transfer, e.g. by a channel reset, is not permitted.
- Parameter "Start" = 1, until parameter "Ready" == 1 or "Error" == 1

Further information

- For detailed information about tool management, refer to the Function Manual Tool Management.
- PI services for tool management
  - FB4: PI_SERV - request PI service (Page 173)
  - FC7: TM_REV - transfer block for tool change with revolver (Page 234)
  - FC22: TM_DIR - direction selection for tool management (Page 283)

Declaration of the function

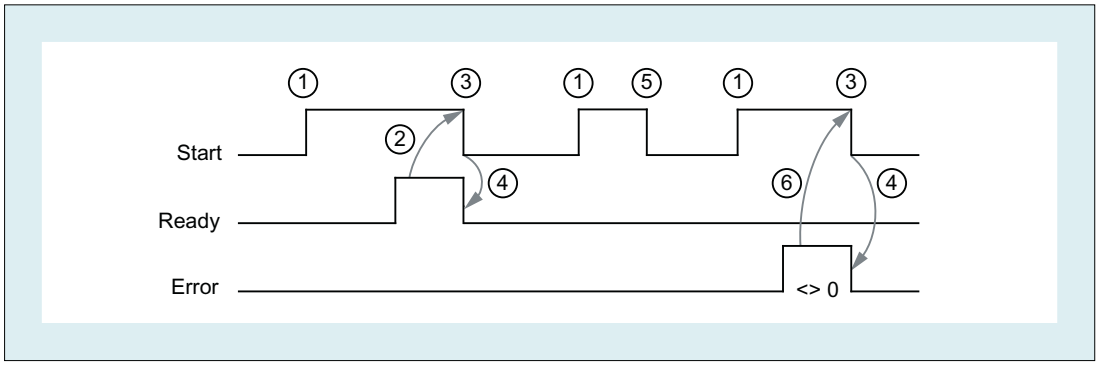
```
FUNCTION FC8: VOID
//NAME :TM_TRANS
VAR_INPUT
    Start :             BOOL;
    TaskIdent:          BYTE ;
    TaskIdentNo:        BYTE ;
    NewToolMag:         INT;
    NewToolLoc:         INT;
    OldToolMag:         INT;
    OldToolLoc:         INT;
    Status:             INT;
END_VAR
```

```
VAR_OUTPUT
    Ready :             BOOL;
    Error :             INT;
END_VAR
BEGIN
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
Start:	I	BOOL	0 (FALSE), 1 (TRUE)	1: Start transfer
TaskIdent:	I	BYTE	1, 2, 3, 4, 5	Interface or task identifier 1: Loading/unloading location 2: Spindle change position 3: Revolver change position 4: Asynchronous transfer 5: Asynchronous transfer with location reservation
TaskIdentNo:	I	BYTE	1, 2, 3 ... 10	Number of the associated interface or channel number.
NewToolMag:	I	INT	-1, 0, 1, 2 ...	Current magazine number of tool to be replaced -1: Tool remains at its location. NewToolLoc = any value Only permissible if TaskIdent = 2.
NewToolLoc:	I	INT	0, 1, 2 … max. location no.	Current location number of new tool
OldToolMag:	I	INT	-1, 0 ...	Current magazine number of tool to be replaced -1: The tool remains at its location. "OldToolLoc" = <any value>. Only permissible if "TaskIdent" = 2.
OldToolLoc:	I	INT	Max. location number	Current location number of tool to be replaced
Status:	I	INT	1, 2, 3 ... 7, 103, 104, 105	Status information about transfer operation
Ready:	O	BOOL	0 (FALSE), 1 (TRUE)	1: Transfer completed
Error:	O	INT	0 ... 65535	Error checkback 0: No error has occurred 1: Unknown "TaskIdent" 2: Unknown "TaskIdentNo" 3: Impermissible job, ("Interface active" signal for selected revolver == 0) Other value: The number corresponds to the error message of the tool management function in the NC caused by this transfer.

Pulse diagram



① Activation of function by means of a positive edge
② Positive acknowledgment: Tool management has been transferred
③ Reset function activation after receipt of acknowledgment
④ Signal change using FC
⑤ This signal chart is not permissible. The job must generally be terminated since the new tool positions must be conveyed to the tool management in the NC.
⑥ Negative acknowledgment: Error occurred, error code in the output parameter Error

Status

Status	Description
1	**The tool management job has been completed** The parameters "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" of the FC8 block should be parameterized to the actual positions of the tools involved. Except in the case of preparing the change, they are normally the specified target positions of the tools of the associated tool management interface, see also "Explanations of the formal parameters". 1. In the case of **loading/unloading/reloading**, the tool has arrived at the required target address. If the bit in the interface in DB 71.DBX (n+0).3 "position at loading point" is enabled, status 1 cannot be used for the function termination. Status 5 must be used for correct termination. 2. In the case of "**Prepare change**", the new tool is now available. The tool may, for example, be positioned in a buffer (gripper). In some cases, the target (magazine, location) of the old tool has been moved to the toolchange position after placement of the new tool in a buffer. However, the old tool still remains in the spindle. The preparations for a tool change are thus complete. After this acknowledgment, the "Change" command can be received. The positions in parameters "NewToolMag", "NewToolLoc", "OldToolMag" and "OldToolLoc" correspond to the current tool positions. 3. In the case of "**Change**" (spindle or revolver), the tools addressed in the interface have now reached the required target addresses. The tool change operation is thus completed.
2	**The "new" tool cannot be made available** This status is only admissible in conjunction with the "Change tool" command. When this status is applied, the PLC must be prevented from making a change with the proposed tool. The proposed (new) tool is disabled by the tool management function in the NC. A new command is then output by the tool management with a duplo tool. The positions in parameters "NewToolMag", "NewToolLoc", "OldToolMag", and "OldToolLoc" correspond to the original tool positions.
3	**An error has occurred** The tool positions must not have been changed. Any changes to the magazine positions of the tool that have taken place in the meantime must be notified beforehand, for example, with status = 105 via FC8 transfer block. Only then will the tool positions be taken into account by the tool management function.
4	**It would be better to position the "old" tool in the magazine position specified in parameters "OldToolMag" and "OldToolLoc"** This status is permissible only in conjunction with preparation for tool change (change into spindle). The magazine location specified for the "old" tool must be free. Once this status has been passed to the tool management in the NC, a new preparation command is generated (Status_4 = final acknowledgment) and output to the DB72, for which the requested magazine position of the old tool is considered. No new tool search is performed explicitly, the positions for "NewToolMag" and "NewToolLoc" are taken from the original preparation command. But this is done only when this position is free. Parameters "NewToolMag" and "NewToolLoc" are not taken into account.

Status	Description
5	**The operation is complete** The "new" tool is in the position specified in parameters "NewToolMag", "NewToolLoc". In this case, the specified tool is not really in this position, but is still in the same magazine location. However, this magazine location has been moved to the position set in the parameters (e.g. tool change position). This status may be used only for revolvers, chain-type magazines and disk magazines. The status enables the tool management function to adjust the current position of a magazine and to improve the search strategy for subsequent commands. This status is permissible only in conjunction with loading, unloading, and reloading operations and with preparations for a tool change. The "OldToolMag" and "OldToolLoc" parameters must be parameterized with the data of a buffer. • **Loading, reloading**: On loading or reloading, a location for the tool is already reserved in the NC. The machine operator must then insert the tool at the target location. Notice: The location reservation is canceled when the control system is switched on again. • **Tool-change preparation**: Tool motions still to be executed are not carried out until after the tool has been changed. • **Positioning to the loading point:** If the bit in the interface in DB 71.DBX (n+0).3 "position at loading point" is enabled, then only status 5 be used for the function termination (not status 1).
6	**The tool management job has been completed** This status has the same function as status 1, but, in addition, a reservation of the source location is carried out. This status is only permitted when reloading. The command is ended and the source location of the tool is reserved if the target location is in a buffer magazine.
7	**Initiate repetition of the command "Prepare Tool"** This status is only admissible in conjunction with the "Change tool" command. This status is intended for use when the "new" tool has changed its position (e.g. via an asynchronous command of the "new" tool). After "Ready = 1" has been provided by FC8, the "Prepare Change" command is repeated automatically with the same tool. For the automatic repetition, a new tool search is carried out. The positions in parameters "NewToolMag", "NewToolLoc", "OldToolMag", and "OldToolLoc" correspond to the original tool positions.
103	**The "new" tool can be inserted** This status is permitted only in the tool change preparation, when the PLC may reject the new tool (e.g. in case of MD20310 $MC_TOOL_MANAGEMENT_MASK, bit 4=1 for the possibility, request changed parameter from PLC once again). The tool positions have remained unchanged. This status is therefore necessary, when the processing is to be continued in the NC without an unnecessary stop.
104	**The "new" tool is in the position specified in parameters "NewToolMag", "NewToolLoc"** This status is only permissible if the tool is still in the magazine in the same location. The "old" tool is in the position (buffer) specified in parameters "OldToolMag", "OldToolLoc". In this case, however, the new tool is not really in this position, but is still in the same magazine location. However, this magazine location has been moved to the position set in the parameters (e.g. tool change position). This status may be used only in conjunction with revolvers, chaintype magazines and disk magazines for the "Tool change preparation" phase. The status enables the tool management to adjust the current position of a magazine and to improve the search strategy for subsequent commands.
105	**The specified buffer location has been reached by all tools involved** Standard case if the operation has not yet been completed. The tools are in the specified tool positions (parameters "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc").

Status definition

A general rule for the acknowledgment status is that the status information 1 to 7 leads to the termination of the command. If FC8 receives a status information, the "Interface active bit" of the interface specified in FC8 is reset to "0" (see also interface lists DB 71 to DB 73). thus completing the operation. The behavior is different in the case of status information 103 to 105. When the FC8 receives a status information, the "Interface active bit" of this interface remains at "1". Further processing is required by the user program in the PLC (e.g. continuation of magazine positioning). This status information is generally used to transfer changes in position of one or both tools while the operation is still in progress.

Call example

```
CALL FC8(           //Tool management transfer block
   Start :          m 20.5,        // Start := "1 " => initiate the transfer
   TaskIdent :      DB61.DBB0,
   TaskIdentNo :    DB61.DBB1,
   NewToolMag :     DB61.DBW2,     // Current position of new tool
   NewToolLoc :     DB61.DBW4,
   OldToolMag :     DB61.DBW6,     // Current position of old tool
   OldToolLoc :     DB61.DBW8,
   Status :         DB61.DBW10,    // Status
   Ready :          m 20.6,
   Error :          DB61.DBW12);
u m 20.6;                          // Poll ready
r m 20.5;                          // Reset start
spb m001;                          // Jump if everything OK
l DB61.dbw12;                      // Error information
ow w#16#0;                         // Evaluate error
JC error;                          // Jump to troubleshooting

m001:                              // Normal branch

error :                            //Troubleshooting
r m 20.5:                          // Reset start
```

16.17 FC9: ASUP - start of asynchronous subprograms

Function

The block FC9 "ASUP" can be used to trigger any functions in the NC. Before an ASUP can be started from the PLC, it must have been selected and parameterized by an NC program or by FB4 (PI service ASUP). In this case, the channel and the interrupt numbers must match the parameters in FC9.

Once prepared in this way, it can be started at any time from the PLC. The NC program running on the channel in question is interrupted by the asynchronous subprogram.

Only one ASUP can be started in the same channel at a time. If several ASUPs are started in **one** PLC cycle, the ASUPs are started in this order in the NC.

Parameter "Start" = 0 must be set by the user if the ASUP has been terminated ("Done" == 1) or an error has occurred ("Error" == 1).

For processing jobs, each FC9 needs its own parameter "Ref" from the global user area. This parameter is for internal use only and must not be changed by the user. The parameter "Ref" is initialized with the value 0 in the first OB1 cycle and, for this reason, every FC9 must be called **absolutely**. Alternatively, the user can initialize parameter "Ref" with a value of 0 during startup. This option makes conditional calls possible. A conditional call requires parameter "Start" = 1 during activation of FC9 until a negative edge change has occurred at parameter "Done" (1 → 0).

General conditions

- The function block FB4 must be terminated before the block FC9 is started.
- The block FC9 cannot be started if DB10, DBX56.1 == 1 (emergency stop).
- Block FC9 must not be started if channel reset is active in the channel in which the ASUP is to be started.

Declaration of the function

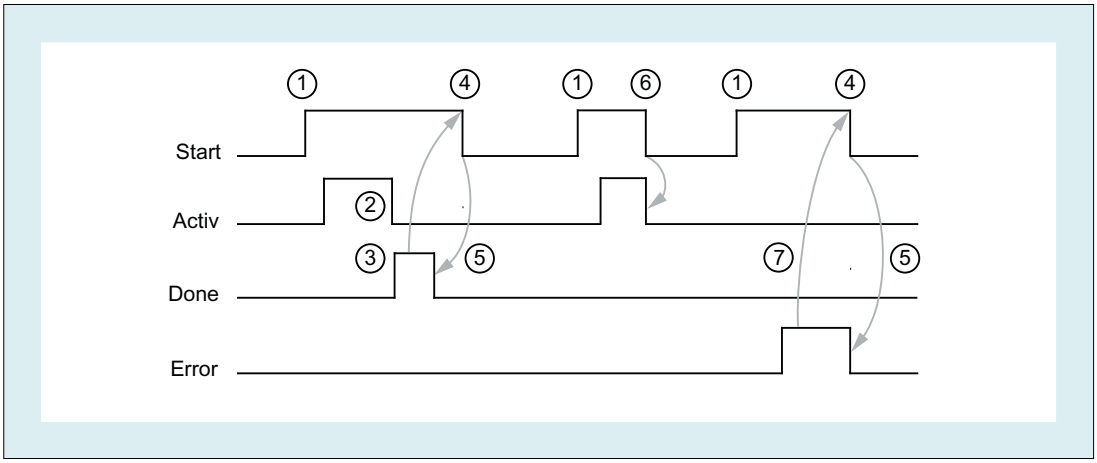
```
FUNCTION FC9: VOID
//NAME :ASUP
VAR_INPUT
    Start :      BOOL;
    ChanNo:      INT;
    IntNo:       INT;
END_VAR
VAR_OUTPUT
    Active:      BOOL;
    Done :       BOOL;
    Error :      BOOL;
    StartErr:    BOOL;
END_VAR
VAR_IN_OUT
    Ref:         WORD;
END_VAR
```

Description of formal parameters

The table below lists all formal parameters of the ASUP function.

Signal	Type	Type	Value range	Description
Start:	I	BOOL	0 (FALSE), 1 (TRUE)	Job start with positive signal edge
ChanNo:	I	INT	1, 2, 3 ... 10	Channel number
IntNo:	I	INT	1, 2, 3 ... 8	Interrupt number
Active:	O	BOOL	0 (FALSE), 1 (TRUE)	1: active
Done:	O	BOOL	0 (FALSE), 1 (TRUE)	1: ASUP completed
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1: Interrupt switched off
StartErr:	O	BOOL	0 (FALSE), 1 (TRUE)	1: Interrupt number not assigned or deleted
Ref:	I/O	WORD	Global variable (MW, DBW,..)	1 word per FC9 (for internal use)

Pulse diagram



(1) Activation of function

(2) ASUP is active

(3) Positive acknowledgment: ASUP completed

(4) Reset function activation after receipt of acknowledgment

(5) Signal change using FC

(6) Not permitted If function activation is reset prior to receipt of acknowledgement, the output signals are not updated without the operational sequence of the activated function being affected.

(7) Negative acknowledgement: An error occurred

Call example

```
CALL FC9(                    //Start an asynchronous subprogram
                             //in channel 1 interrupt number 1
     Start :      I 45.7,
     ChanNo :     1,
     IntNo :      1,
     Active :     M 204.0,
     Done :       M204.1,
     Error :      M 204.4,
     StartErr :   M 204.5,
     Ref :        MW 200);
```

16.18 FC10: AL_MSG - error and operating messages

Function

With block FC10 "AL_MSG", the signals entered in DB2 are evaluated and displayed as incoming or outgoing error messages and operational messages on the user interface.

The incoming signals (positive edge) are displayed immediately in the case of both error and operational messages.

Outgoing signals (negative edge) are deleted immediately only for operational messages. In the event of error messages, the messages that are no longer pending are only deleted with the parameter "Quit", i.e. errors remain displayed on the user interface until they have been acknowledged by the user even if the signals are no longer pending.

The "ToUserIF" parameter can be used to transfer the group signals for the feed, read and NC start disabling signals and feed stop signal to the existing axis, spindle and channel interfaces. The group signals are transferred to the user interface directly from the status information in DB2 irrespective of an alarm acknowledgment.

1. If parameter "ToUserIF" = 0, there is no transfer of the signals to the user interface. In this case, the user must take measures in his PLC program to ensure that these signals are influenced in the interface. The FB1 parameter "ExtendChanAxMsg" is evaluated and so avoids the limitation of the usable message ranges by configuring the NCK machine data.
2. If parameter "ToUserIF" = 1, all signals listed above are sent to the user interface as a group signal in each case. The user PLC program can, therefore, influence these signals only via DB2 in conjunction with a message or alarm output. The appropriate information is overwritten in the user interface.

Alternatively to the response described in paragraph 2, the disable and hold signals can be influenced without a message being output by influencing the interface signals with a disable or stop signal state after FC10 is called.

The following program illustrates this method:

```
CALL FC10(
   ToUserIF :     TRUE,
```

```
   Ackn :         I 6.1);

u m 50.0;                       // Feed disable for channel 1
to DB 21;
s dbx 6.0;                      // Setting the blocking condition
                                // Resetting is done via FC AL_MSG
                                // if M 50.0 outputs the signal "0".
```

FB1 parameter "ExtendAlMsg"

With FB1 parameter "ExtendAlMsg" = TRUE, the new DB2 structure becomes effective (see "Interface PLC/HMI (Page 62)"). For the activation, bit fields for the lock and stop signals are available for 10 channels, 31 axes and maximum 64 user areas (the number of user areas is entered in the FB1 parameter "MsgUser"). The associated functionality is obtained automatically by simply setting/resetting signals in DB2.

The error and operational messages in data block DB2 must be provided in a user-specific way.

FB1 parameter "ExtendChanAxMsg"

With the activation of this parameter, a channel- or axis-number independent acquisition of alarms and messages acts. All DB2 areas are available for users. Group signals cannot be transferred to the user interface. The parameter is evaluated only when the FC10 parameter "ToUserIF" is deactivated.

Display on HMI

In DB2, a "1" signal must be present for several OB1 cycles to ensure that a message can also be displayed on the HMI.

There is an upper limit for the number of alarms and messages that can be pending at the same time. This upper limit is dependent on the PLC CPU. On PLC 317-2DP, the upper limit for messages pending simultaneously is 60.

Further information:
PLC alarms/messages (Page 317)

Declaration of the function

```
FUNCTION FC10:      VOID
    // NAME:        AL_MSG
VAR_INPUT
    ToUserIF :      BOOL;
    Ackn :          BOOL;
END_VAR
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
ToUserIF :	I	BOOL	0 (FALSE), 1 (TRUE)	1: Transfer of the signals to the user interface in each cycle
Ackn:	I	BOOL	0 (FALSE), 1 (TRUE)	1: Acknowledgment of error messages

Call example

```
CALL FC10(                  // Error and operational messages
   ToUserIF :=   TRUE,      // Signals from DB2 are transferred to interface.
   Ackn :=       E6.1       // Acknowledgment of error message via input I6.1
);
```

16.19 FC12: AUXFU - call interface for user with auxiliary functions

Function

The block FC12 "AUXFU" is called on an eventdriven basis in the basic program if the channel transferred in the input parameter contains new auxiliary functions. The PLC user can extend FC AUXFU with program instructions for processing his auxiliary function to avoid cyclic polling of the channel DBs. This mechanism permits auxiliary functions to be processed on a jobdriven basis. FC AUXFU is supplied as a compiled empty block in the basic program. In this case, the basic program supplies parameter "Chan" with the channel number. The PLC user knows which channel has new auxiliary functions available. The new auxiliary functions can be determined by the auxiliary-function change signals in the channel concerned.

Declaration of the function

```
FUNCTION FC12: VOID          // Event control of auxiliary functions
VAR_INPUT
    Chan:     BYTE ;
END_VAR
BEGIN
    BE;
END_FUNCTION
```

Explanation of formal parameters

Signal	Type	Type	Value range	Description
Chan:	I	BYTE	0, 1, 2 ... 9	Index of the channel = channel number -1

Example

```
FUNCTION FC12: VOID        // Event control of auxiliary functions
VAR_INPUT
    Chan:      BYTE ;      // Parameter is supplied by basic program
END_VAR
VAR_TEMP
    ChanDB:    INT;
END_VAR

BEGIN
L Chan;                    // Channel index
+ 21;                      // Channel DB offset
T ChanDB;                  // Save channel DB no.
TO DB[ChanDB];             // Channel DB is opened indirectly
// Auxiliary-function change signals are now scanned, etc.
    BE;
END_FUNCTION
```

16.20 FC13: BHGDisp - display control for handheld unit

Function

Block FC13 "BHGDisp" handles the display control for the handheld unit (HHU or HT 2). The information that is to appear on the display must be saved to a string variable. The pointer to the string is specified in the parameter "ChrArray". To do this, a fixed text assignment of 32 characters (HHU) or 64 characters (HT 2) is needed when the data block for this string is created.

16 characters are sent to the HHU per job. The assignment of the characters in the "ChrArray" for the respective line is unambiguous. For line 1, characters 1 to 16 and for line 2, characters 17 to 32 of the string data ChrArray are transferred. In addition, for HT 2 line 3 with characters 33 to 48 is displayed and line 4 with characters 49 to 64. A job takes several OB1 cycles.

Display

Block FC13 checks whether the necessary minimum length of the "ChrArray" exists for operating the handheld unit. If fewer characters exist in the string variables than should be displayed, the line is filled with blank spaces. If several variables are to be entered in the string in one or more PLC cycles without a display output, the display output can be suppressed by parameter "Row" = 0. The transfer of the characters to the rows takes several OB1 cycles. If several rows are to be updated "simultaneously" (parameter "Row" > 1), the rows are updated successively with 16 characters per row.

Variable portions

Variable components within the string can be inserted using the optional number converter functionality with the parameter "Convert" = 1. The variable to be displayed is referenced via the parameter "Addr". The format of the variables is described in the parameter "DataType". The

number of bytes of the variable is linked to the format description. The address justified to the right within the string is specified by parameter "StringAddr". The number of written characters is shown in the parameter table.

High display resolution

If, for example, the axis value is to be displayed with a higher resolution, the following must be observed:

- The variables are read as before with FB2 or FB5. Instead of anypointer BYTE 8 as criterion for output as 64-bit floating point number, REAL 2 is used (e.g.: P#M100.0 REAL 2).
- When specifying the 64-bit floating point number on the HHU/HT 2, you can select the output format with up to 14 places, distributed freely before and after the decimal point, instead of fixed, specified formats.

HHU output signals

Byte 1 is used by the HHU output signals and the character specifications are used by the block FC13. These may not be written by the PLC user program.

Relevant FB1 parameters

Handheld unit HHU

In OB100, the FB1 parameters must be set for the input and output data of the handheld unit:

- Parameter "BHGIn" corresponds to the input data of the PLC from the handheld unit (data received by PLC)
- Parameter "BHGOut" corresponds to the output data of the PLC to the handheld unit (data transmitted by PLC).

The two pointers must be set to the starting point of the relevant data area, which is also parameterized in SDB 210 with an MPI link.

The FB1 parameter "HHU" = 2 must be set for operating an HHU.

HT 2 handheld terminal

If HT 2 is used, FB1 parameter "HHU" = 5 must be set. The parameters of the input and output data must be set, as described in the above paragraph "Handheld Unit HHU".

The value that was configured at S2 of the DIP-Fix switch (rotary coding switch) of the connecting module of the HT 2 must be assigned to the parameters "BHGRecGDNo" and "BHGRecGBZNo".

Declaration of the function

```
DATA_BLOCK "strdat"
   STRUCT
    disp:      STRING [32]:= 'character_line1 character_line2';
   END_STRUCT;
BEGIN
END_DATA_BLOCK
```

```
FUNCTION FC13: VOID
   VAR INPUT
     Row :        BYTE ;
     ChrArray :   STRING ;
     Convert :    BOOL;
     Addr:        POINTER;
     DataType :   BYTE ;
     StringAddr   INT;
     :
     Digits :     BYTE ;
END VAR
VAR OUTPUT
     Error :      BOOL;
END VAR
```

Description of formal parameters

Signal	Type	Type	Value range	Description
Row :	I	BYTE	0, 1, 2, ... 8 B#16#F	Display line "binary" evaluation 0: no display output 1: Line 1 2: Line 2 3: Line 1 and line 2 to be changed 4: Line 3 5: Line 1 and line 3 to be changed 8: Line 4 B#16#F automatic change of all 4 lines
ChrArray :	I	STRING	"DBName".<VarName>	Display content as pointer to string string[**32**]: not HT 2 string[**64**]: HT 2
Convert :	I	BOOL	0 (FALSE), 1 (TRUE)	Activation of numerical conversion
Addr:	I	Pointer		Pointer to the variable to be converted

Signal	Type	Type	Value range	Description
DataType:	I	BYTE	1, 2, 3 ... 8, B#16#13, B#16#30	Data type of the tag 1: BOOL, 1 character 2: BYTE, 3 characters 3: CHAR, 1 character 4: WORD, 5 characters 5: INT, 6 characters 6: DWORD, 7 characters 7: DINT, 8 characters 8: REAL, 9 characters (7 digits plus a sign and a decimal point; for places after the decimal point, refer to the Digits Parameter) B#16#13: String, up to 32/64 characters, "Addr" must be a pointer to a STRING. B#16#30: REAL64, (12 characters: 10 digits plus a sign and a decimal point; for places after the decimal point, refer to the Digits Parameter)
StringAddr :	I	INT	1 ... 32 / 64	Right-justified address within variable from "ChrArray"
Digits :	I	BYTE	1, 2, 3 ... 9	Number of places after the decimal point: 1 ... 4: DataType REAL 1 ... 9: DataType REAL64
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	Error 1: An error occurred

Ranges of values

Ranges of values of data types	
Data type	Representable numerical range
BOOL	0, 1
BYTE	0 ... 255
WORD	0 ... 65535
INT	- 32768 ... 32767
DWORD	0 ... 9999999
DINT	-9999999 ... 9999999
REAL (Digits := 1)	-999999.9 ... 999999.9
REAL (Digits := 2)	-99999.99 ... 99999.99
REAL (Digits := 3)	-9999.999 ... 9999.999
...	...
REAL (Digits := 9)	-0.9999999 ... 0.9999999

Call example

```
// DB with name strdat in the simple table, data element disp is declared as String[32] (for HT 2:
// String[64]) and completely assigned with characters

CALL FC13(
   Row :=         MB 26,
   ChrArray :=    "strdat".disp
                  ,
   Convert :=     M 90.1,
   Addr :=        P#M 20.0,      // Number to be converted
   DataType :=    MB 28,         // Data type of the variables
   StringAddr :=  MW 30,
   Digits :=      B#16#3,        // 3 decimal places
   Error :=       M 90.2);
```

16.21 FC17: YDelta - star-delta switchover

Function

Block FC17 is used for star-delta changeover for digital main spindle drives. The changeover can be made in both directions (star > delta or delta > star).

Prerequisite

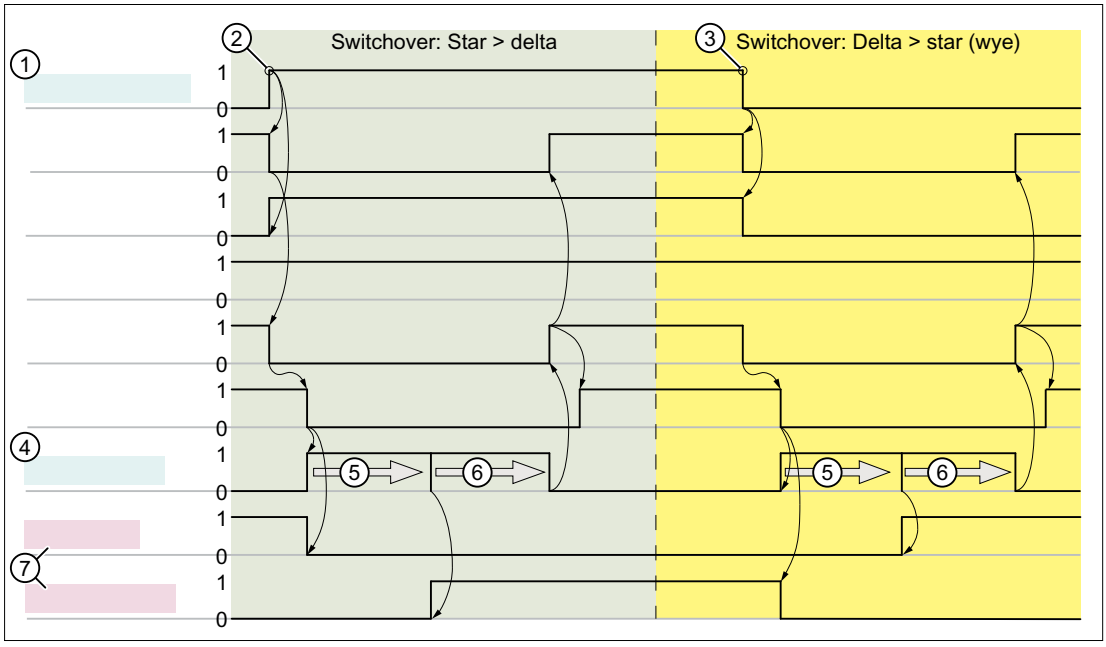
The prerequisite is two isolated contactors. The contactors are controlled via the peripheral output signals configured at the FC17 outputs: "Y" or "Delta".

Internal sequence

The internal sequence when changing over between star and delta after switching over the FC17 control signal is described in the following: "YDelta" displayed.

1. DB31, ...DBX21.5 = 0 (reset feedback signal "motor selected")
   DB31, ...DBX21.x = 1 (set request "2nd motor data set" corresponding to the interface parameterization DB31, ...DBX130.0 - 4)
2. DB31, ...DBX93.7 == 0 (feedback signal "pulses enabled" was reset) ⇒
   - Start FC17 timer
   - FC17: "Y" = 0 (reset output for star contactor)
3. After the FC17 timer elapses (FC17: "TimeVal") ⇒
   - FC17: "Y" = 1 (set output for delta contactor)
4. After the FC17 timer elapses again (FC17: "TimeVal") ⇒
   - The pulses are again internally set
   - DB31, ...DBX93.7 == 1 (feedback signal "pulses enabled" was set)
   - DB31, ...DBX21.5 = 0 (feedback signal "motor selected" was set)

Signal flow



① FC17 input: Signal for star-delta changeover
② - Changeover: Star > delta
③ - Changeover: Delta > star
④ FC17 input: Parameterizable changeover time ""TimeVal"
⑤ - Wait time until the control of the output signal: "Y" or "delta"
⑥ - Wait time until pulse enable
⑦ FC17 output: Signals for contactor control

Further information

For additional explanations of motor speed adjustments, see:

- Function Manual Axes and Spindles; spindles; configurable gear adaptations
- Function Manual Axes and Spindles; speeds, setpoint/target value syst., closed-loop control

---

Note

Drive parameters

The following drive parameters must be considered for a star-delta changeover:

- p833 (dataset changeover configuration)
  - Bit 0 = **1** (contactor switchover via **application**)
  - Bit 1 = **0** (pulse cancellation by **drive**)
- p826 (motor changeover motor number)
- p827 (motor changeover status word bit number)

Boundary conditions

- Block FC17 must be called absolutely and separately for each spindle.
- The drive pulses are deactivated during a star-delta changeover. The feedback signal to the PLC is realized using:
  - DB31, ...DBX93.7 == 0 (pulses enabled)
  - DB31, ...DBX61.7 == 0 (current controller active)
  - DB31, ...DBX61.6 == 0 (speed controller active)
- For a spindle which is located in an axis mode such as M70 or SPOS, a start-delta changeover is not carried out.
- For a closed-loop position controlled spindle (DB31, ...DBX61.5 == 1 (position control active)), while the spindle is moving it is not permissible to carry out a star-delta changeover. In the case of a fault, Alarm 25050 "Contour monitoring" is displayed, and the star-delta changeover is not executed.
- Once the star-delta changeover has been initiated using FC17, it cannot be delayed by the user, e.g. by waiting until the star-delta contactors change over during the course of operation. A delay such as this must be implemented by the user in the PLC user program.

Declaration of the function

```
VAR_INPUT
    YDelta:         BOOL;
    SpindleIFNo:    INT;
    TimeVal:        S5TIME ;
    TimerNo :       INT;
END_VAR
VAR_OUTPUT
    Y:              BOOL;
    Delta:          BOOL;
END_VAR
VAR_IN_OUT
    Ref:            WORD;
END_VAR
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
YDelta:	I	BOOL	0 (FALSE), 1 (TRUE)	Input signal for star-delta changeover The changeover is initiated by a signal change: • 0 (FALSE): Star • 1 (TRUE): Delta
SpindleIFNo:	I	INT	1 ...	Number of the spindle interface (number of the associated machine axis)

Signal	Type	Type	Value range	Meaning	
TimeVal:	I	S5time	0, 50 ms ...	Switchover time	
				Configured	Internally effective
				0	100
				< 50	50
TimerNo:	I	INT	10 ...	Number of the timer being used	
Y:	O	BOOL	0 (FALSE), 1 (TRUE)	Peripheral output for controlling the star contactor	
Delta:	O	BOOL	0 (FALSE), 1 (TRUE)	Peripheral output for controlling the delta contactor	
Ref:	I/O	WORD		Instance for status information. Internal use	

Call example

```
CALL FC17 (
    YDelta :=          I 45.7,          // Accept star/delta changeover from
                                        input 45.7
    SpindleIFNo :=     4,               // Spindle interface number: 4
                                        // (Number of the associated machine
                                        axis)
    TimeVal :=         S5T#150ms,       // Changeover time: 150 ms
    TimerNo :=         10,              // Timer: 10
    Y :=               O 52.3,          // Control of the star contactor:
                                        Output 52.3
    Delta :=           O 52.4,          // Control of the delta contactor:
                                        Output 52.4
    Ref :=             MW 50            // Bit memory word 50
};
```

16.22 FC18: SpinCtrl - spindle control

Function

Block FC18 "SpinCtrl" can be used to control spindles and axes from the PLC. The block supports the following functions:

- Position spindle
- Rotate spindle
- Oscillate spindle
- Traverse indexing axis
- Traverse positioning axis

Each function is activated by the positive edge of the appropriate initiation signal (start, stop). This signal must remain at a logical "1" until the function has been acknowledged positively or

negatively by InPos="1" or Error = "1". The output parameters are deleted when the relevant trigger signal is reset and the function has been completed.

To be able to control an axis or spindle via the PLC, it must be activated for the PLC. This can, for example, be achieved by calling the block with activation of the "Start" or "Stop" parameter. When you do this, the block requests control of the spindle/axis from the NC.

The NC signals the status of the spindle/axis in the associated axis-specific interface DB31, ... DBX68.4 - 7. Once the axis/spindle is operating under PLC control, the travel command for the active status can be evaluated via the relevant axis-specific interface.

Upon completion ("InPos" is True, "Start" changes to zero), the axis/spindle check function is switched to a neutral status using block FC18.

Alternatively, the PLC user program can also request control for the PLC before calling FC18.

By calling this function several times in succession, a better response by the spindle/axis can be obtained as the changeover process in the FC can be omitted.

Activation through the PLC user program is executed in the corresponding spindle interface in byte 8.

After return of the check, the spindle can again be programmed by the NC program.

Further information

- Function Manual Axes and Spindles; spindles
- Function Manual Axes and Spindles; positioning axes
- Function Manual Axes and Spindles; indexing axes

⚠ WARNING

Changed response behavior of the axis/spindle

If several block calls (FC18) have been programmed for the same axis/spindle in the PLC user program, then the functions concerned must be interlocked by conditional calls in the user program. The conditional call of a started block (parameter Start or Stop = TRUE) must be called cyclically until the signal state of output parameter "Active" or "InPos" changes from 1 to 0.

Note

Call note

FC18 must be called cyclically until signal "InPos" or, in the case of an error "Error", produces an edge transition of 1 to 0. An additional "Start" or "Stop" is only possible for this spindle/axis when the "InPos"/"Error" signal has supplied a value of 0. (the system must wait for at least one PLC cycle with the next "Start" or "Stop"). This also applies when the assignment in data byte 8 on the axial interface has been changed.

Abort

The function cannot be aborted by means of parameter "Start" or "Stop", but only by means of the axial interface signals (e.g. delete distance-to-go). The axial interface also returns status signals of the axis that may need to be evaluated (e.g. exact stop, traverse command).

Simultaneity

Several axes can be traversed simultaneously or subject to a delay by FC18 blocks. The upper limit is only limited by the maximum number of axes of the NC.

Axis disable

For a set axis inhibit (DB31, ...DBX1.4 == 1), the axis controlled via FC18 does not move. Only a simulated actual value is generated. Same behavior as when traversing the axis for an axis inhibit by the NC.

Functions

Function 1: Position spindle

Parameter	Meaning
Start :	0 → 1: Start the function
Funct :	1: Function number for "Position spindle"
Mode :	Positioning modes 1, 2, 3, 4 (refer to the paragraph below, "Explanation of the formal parameters"
AxisNo :	Number of machine axis
Pos :	Position
FRate :	FRate ≠ 0: Positioning velocity FRate = 0: Velocity corresponding to MD35300 $MA_SPIND_POSCTRL_VELO
InPos :	1: Position reached with "Exact stop fine"
Error :	1: Positioning error
State :	Error code

Function 2: Rotate spindle

Parameter	Meaning
Start :	0 → 1: Start the function
Stop :	0 → 1: Stop the function
Funct :	2: Function number for "Rotate spindle"

Parameter	Meaning
Mode :	Mode = 5: direction of rotation M4 Mode ≠ 5: direction of rotation M3
AxisNo :	Number of machine axis
FRate :	Spindle speed
InPos :	1: Setpoint speed is output, also see DB31, ... DBX83.5 (spindle in the setpoint range)
Error :	1: Positioning error
State :	Error code

Function 3: Oscillate spindle

Parameter	Meaning
Start :	0 → 1: Start the function
Stop :	0 → 1: Stop the function
Funct :	3: Function number for "Oscillate spindle"
AxisNo :	Number of machine axis
Pos :	Set gear stage
InPos :	1: Setpoint speed is output, also see DB31, ... DBX83.5 (spindle in the setpoint range)
Error :	1: Positioning error
State :	Error code

Parameterized oscillation speed: MD35400 $MA_SPIND_OSCILL_DES_VELO

The function of the parameter "Pos" depends on the setting in MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = <value>

<Value>	Pos	Function
0	0, 1, 2, ... 5	Oscillation
1	0	Oscillation with gear stage change M40
	1	Oscillation with gear stage change M41
	2	Oscillation with gear stage change M42
	3	Oscillation with gear stage change M43
	4	Oscillation with gear stage change M44
	5	Oscillation with gear stage change M45

Function 4: Traverse indexing axes

Note

The modulo conversion can be compared with approaching the indexing position via POS[AX] = CIC (value) in the part program.

Parameter	Meaning
Start :	0 → 1: Start the function
Funct :	4: Function number for "Indexing axis"
Mode :	Positioning mode 0, 1, 2, 3, 4
AxisNo :	Number of machine axis
Pos :	Indexing position
FRate :	FRate ≠ 0: Positioning velocity FRate = 0: Velocity corresponding to MD32060 $MA_POS_AX_VEL
InPos :	1: Position reached with "Exact stop fine"
Error :	1: Positioning error
State :	Error code

Function 5, 6, 7, 8: Position axes

Parameter	Meaning
Start :	0 → 1: Start the function
Funct :	5, 6, 7, 8: Function number for "Position axes"
Mode :	Positioning mode 0, 1, 2, 3, 4
AxisNo :	Number of machine axis
Pos :	Position
FRate :	FRate ≠ 0: Positioning velocity FRate = 0: Velocity corresponding to MD32060 $MA_POS_AX_VELO
InPos :	1: Position reached with "Exact stop fine"
Error :	1: Positioning error
State :	Error code

Function 9: Rotate spindle with automatic gear stage selection:

Parameter	Meaning
Start :	0 → 1: Start the function
Stop :	0 → 1: Stop the function
Funct :	9: Function number for "Rotate spindle with gear stage selection"
Mode :	Mode = 5: direction of rotation M4 Mode ≠ 5: direction of rotation M3
AxisNo :	Number of machine axis
FRate :	Spindle speed
InPos :	1: Setpoint speed is output
Error :	1: Positioning error
State :	Error code

Function 10, 11: Rotate spindle with constant cutting rate

The "Constant cutting rate" function (G96) must be active in the NC.

Parameter	Meaning
Start :	0 → 1: Start the function
Stop :	0 → 1: Stop the function
Funct :	10: Function number for "Constant cutting rate (m/min)" 11: Function number for "Constant cutting rate (feet/min)"
Mode :	Mode = 5: direction of rotation M4 Mode ≠ 5: direction of rotation M3
AxisNo :	Number of machine axis
FRate :	Cutting rate
InPos :	1: Setpoint speed is output
Error :	1: Positioning error
State :	Error code

Declaration of the function

```
FUNCTION FC18: VOID          //SpinCtrl
VAR_INPUT
    Start :                  BOOL;
    Stop :                   BOOL;
    Funct :                  BYTE ;
    Mode :                   BYTE ;
    AxisNo :                 INT;
    Pos :                    REAL;
    FRate :                  REAL;
END_VAR
VAR_OUTPUT
    InPos :                  BOOL;
    Error :                  BOOL;
    State :                  BYTE ;
END_VAR
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
Start:	I	BOOL	0 (FALSE), 1 (TRUE)	0 → 1: Start the function
Stop:	I	BOOL	0 (FALSE), 1 (TRUE)	0 → 1: Stop the function

Signal	Type	Type	Value range	Meaning
Funct:	I	BYTE	1, 2, 3, ... 11	1: Position spindle 2: Rotate spindle 3: Oscillate spindle 4: Indexing axis 5: Positioning axis metric 6: Positioning axis inch 7: PosAxis metric with handwheel override 8: PosAxis inch with handwheel override 9: Rotate spindle with automatic gear stage selection 10: Rotate spindle with constant cutting rate (m/min) 11: Rotate spindle with constant cutting rate (feet/min)
Mode:	I	BYTE	0, 1, 2, ... 5	0: Positioning to absolute position 1: Positioning incremental 2: Positioning along the shortest path 3: Positioning absolute, positive approach direction 4: Positioning absolute, negative approach direction 5: Direction of rotation as for M4
AxisNo:	I	INT	1, 2, 3, ... 31	Number of the axis/spindle to be traversed
Pos:	I	REAL	∓ 0.1469368 l -38 to ∓ 0.1701412 l +39	Rotary axis: Degrees Indexing axis: Indexing position Linear axis: mm or inches
FRate:	I	REAL	∓ 0.1469368 l -38 to ∓ 0.1701412 l +39	Rotary axis and spindle: [rev/min] Linear axes: [m/min] or [ft/min]
InPos:	O	BOOL	0 (FALSE), 1 (TRUE)	1: Position reached or function executed
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1: Error
State:	O	BYTE	0, 1, 2, ... 255	Error code

Error identifiers

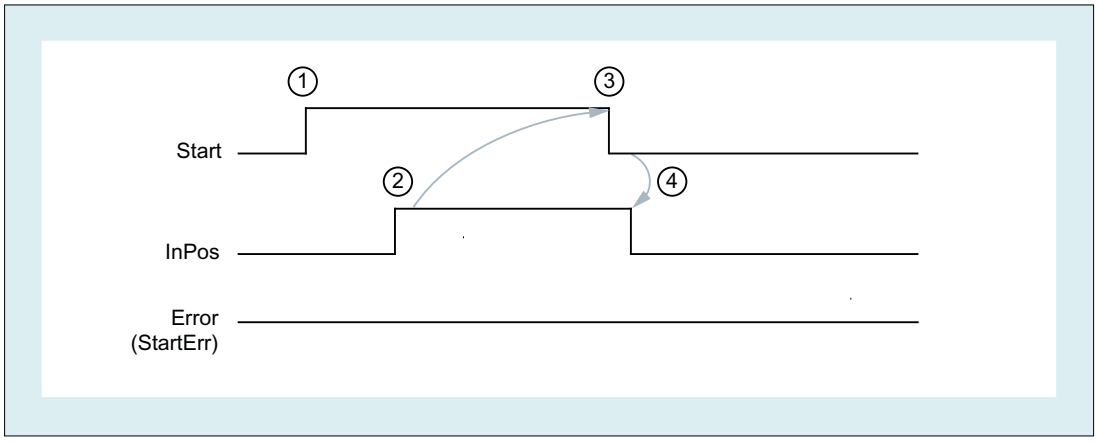
An error is active, if: Parameter "Error" == 1 (TRUE)

The cause of the error is displayed in: Parameter "State"

State	Meaning
Cause of the error on the PLC side	
1	Several functions of the axis/spindle were activated simultaneously
20	A function was started without the position being reached
30	The axis/spindle was transferred to the NC while still in motion

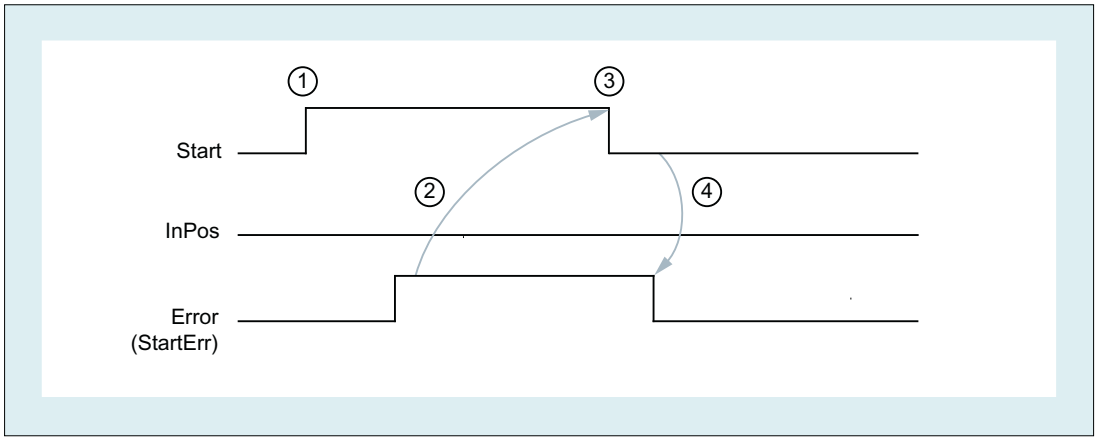
State	Meaning
40	The axis is programmed by the NC program, NC internal error
50	Permanently assigned PLC axis: Traverses (JOG) or references
60	Permanently assigned PLC axis: Channel status does not permit a start
Cause of the error on the NC side	
100	Incorrect position programmed for axis/spindle (corresponds to alarm 16830)
101	Programmed speed is too high
102	Incorrect value range for constant cutting rate (corresponds to alarm 14840)
104	Following spindle: Illegal programming (corresponds to alarm 22030)
105	No measuring system available (corresponds to alarm 16770)
106	Axis positioning still active (corresponds to alarm 22052)
107	Reference mark not found (corresponds to alarm 22051)
108	No transition from speed control to position control (corresponds to alarm 22050)
109	Reference mark not found (corresponds to alarm 22051)
110	Velocity/speed is negative
111	Setpoint speed == zero
112	Invalid gear stage
115	Programmed position has not been reached
117	G96/G961 is **not active** in the NC
118	G96/G961 is **still active** in the NC
120	Axis is not an indexing axis (corresponds to alarm 20072)
121	Indexing position error (corresponds to alarm 17510)
125	DC (shortest distance) not possible (corresponds to alarm 16800)
126	Absolute value minus not possible (corresponds to alarm 16820)
127	Absolute value plus not possible (corresponds to alarm 16810)
128	No transverse axis available for diameter programming (corresponds to alarm 16510)
130	Software limit switch plus (corresponds to alarm 20070)
131	Software limit switch minus (corresponds to alarm 20070)
132	Working area limit plus (corresponds to alarm 20071)
133	Working area limit minus (corresponds to alarm 20071)
134	Frame not permitted for indexing axis
135	Indexing axis with "Hirth joint" is active (corresponds to alarm 17501)
136	Indexing axis with "Hirth joint" is active and axis not referenced (corresponds to alarm 17503)
137	Spindle operation not possible for transformed spindle/axis (corresponds to alarm 22290)
138	Axis: Coordinate system-specific working area plus violated (corresponds to alarm 20082)
139	Axis: Coordinate system-specific working area minus violated (corresponds to alarm 20082)
System error	
200	corresponds to alarm 450007
Alarm numbers: **Further information** Diagnostics Manual	

Signal sequence: Normal case



① PLC user program: Function start using a positive edge: 0 → 1
② NC: Positive acknowledgment, function executed / position reached
③ PLC user program: Reset after detecting the positive acknowledgment
④ FC18: Reset of the positive acknowledgment

Signal sequence: Error case



① PLC user program: Function start using a positive edge: 0 → 1
② NC: Negative acknowledgment, error occurred
③ PLC user program: Reset after detecting the negative acknowledgment
④ FC18: Reset of the negative acknowledgment

Call examples

Example 1: Position spindle:

```
//Positive acknowledgment resets Start:
U M112.0;                    //InPos
```

```
R M 100.0;                  //Start
//Negative acknowledgment, after error evaluation (state: MB114) reset with T12 start
U M113.0;                   // Error
U E 6.4;                    //Key T12
R M 100.0;                  //Start
//Start with T13
U E 6.3;                    //Key T13
UN M 112.0;                 //Restart only when InPos or Error = 0
UN M 113.0;
S M 100.0;

CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,
    Funct :=  B#16#1,       //Position spindle
    Mode :=  B#16#2,        //Shortest path
    AxisNo :=  5,
    Pos :=  MD104,
    FRate :=  MD108,
    InPos :=  M112.0,
    Error :=  M113.0,
    State :=  MB114);
```

Example 2: Start spindle rotation:

```
CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,
    Funct :=  B#16#2,       //Rotate spindle
    Mode :=  B#16#5,        //Direction of rotation as for M4
    AxisNo :=  5,
    Pos :=  0.0,
    FRate :=  MD108,
    InPos :=  M112.0,
    Error :=  M113.0,
    State :=  MB114);
```

Example 3: Start spindle oscillation

```
CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,
    Funct :=  B#16#3,       //Oscillate spindle
    Mode :=  B#16#0,
    AxisNo :=  5,
    Pos :=  0.0,
```

```
    FRate :=  MD108,
    InPos :=  M112.0,
    Error :=  M113.0,
    State :=  MB114);
```

Example 4: Traverse indexing axis

```
CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,          //Not used
    Funct :=  B#16#4,        //Traverse indexing axis
    Mode :=  B#16#0,         //Absolute positioning
    AxisNo :=  4,
    Pos :=  MD104,           //Default setting in REAL: 1.0;2.0;..
    FRate :=  MD108,
    InPos :=  M112.0,
    Error :=  M113.0,
    State :=  MB114);
```

Example 5: Position axes

```
CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,          //Not used
    Funct :=  B#16#5,        //Position axes
    Mode :=  B#16#1,         //Position incrementally
    AxisNo :=  6,
    Pos :=  MD104,
    FRate :=  MD108,
    InPos :=  M112.0,
    Error :=  M113.0,
    State :=  MB114);
```

16.23 FC19: MCP_IFM - transfer of MCP signals to interface

Function

Block FC19 "MCP_IFM" (M version e.g. MCP 483) is used to transfer data from the machine control panel to the NC/PLC interface:

- Modes
- Axis selections
- WCS/MCS switchover
- Traversing keys

- Overrides
- Keyswitch

The following specifications apply to the **feedrate override, axis travel keys** and **INC keys** depending on the active mode or on the coordinate system selected:

- **Feedrate override**
  - The feedrate override is transferred to the interface of the selected channel and to the interface of the axes.
  - The feedrate override signals are transferred to the NC channel in addition to the "Rapid traverse override" (DBB5) interface byte if the "Feedrate override for rapid traverse effective" HMI signal is set (exception: Switch setting "Zero"). "Rapid traverse override effective" is also set with this HMI signal.
- **Machine functions INC and axis travel keys**
  - When the MCS is selected, the signals are transferred to the interface of the selected machine axis.
  - When the WCS is selected, the signals are transferred to the geometry axis interface of the parameterized channel.
  - When the system is switched between MCS and WCS, the active axes are generally deselected.

The LEDs on the machine control panel derived from the selections in the feedback.

Feedrate and spindle Start/Stop are not transferred to the interface, but output modally as a "FeedHold" or "SpindleHold" signal. The user can link these signals to other signals leading to a feed or spindle stop (this can be implemented, e.g. using the appropriate input signals in FC10: AL_MSG). The associated LEDs are activated at the same time.

If the machine control panel fails, the signals it outputs are preset to zero; this also applies to "FeedHold" and "SpindleHold" output signals.

FC19 or also FC24, FC25, FC26 can be called a multiple number of times in a single PLC cycle. In this case, the first call in the cycle drives the LED displays. Furthermore, all actions of the parameterized block are carried out in the first call. In the following calls, only a reduced level of processing of the channel and mode group interface takes place. The geometry axes are supplied with directional data only in the first block call in the cycle.

Single block processing can be selected/deselected only in the first call in the cycle.

The second machine control panel can be processed if parameter "ModeGroupNo" has been increased by B#16#10. When parameterizing, the mode group number is contained in the lower nibble.

"BAGNo" = 0 or B#16#10 ⇒ mode group signals are not processed.

"ChanNo" = 0 ⇒ no processing of the channel signals.

The INC selections are transferred to the mode group interface. The activation for this specification is done via the DB10.DBX57.0 (INC inputs in BAG area active) through this block once after power up.

Furthermore, two machine control panels can still be handled in parallel by the FC19 block. The block call for the 2nd machine control panel in OB1 cycle must be set after the call of the 1st machine control panel. Support of two machine control panels exists to a limited extent in the

machine control panel blocks. Mutual locking of the axis selections for equally assigned axes for two machine control panels is not supported.

Cartesian manual traversing

The R11 direction key on the machine control panel (at the left next to WCS/MCS) makes the "Manual traversing in tool orientation" function available. This requires activation via the FB1 input parameter "MCP_IF_TCS" in DB7.

For "MCP_IF_TCS" = TRUE, the R11 key switches to "Manual traversing in tool orientation". Whereby, the Z key (R3) is permanently selected with FC19. The direction keys are effective for the 3rd geometry axis of the respective channel.

Flexible axis configuration

It is possible to be flexible in the assignment of axis selections or direction keys for machine axis numbers.

Better support is now provided by the MCP blocks for the use of two MCPs, which are to run in parallel, in particular for an application using two channels and two mode groups. Note that the axis-numbers are also specified in the parameterized mode group number of the MCP block in the axis tables of the relevant MCP.

To provide this flexibility, tables for axis numbers are stored in DB10.

For the **first** machine control panel the table starts at byte 8 (symbolic name: MCP1AxisTbl[1..22]) and for the **second** machine control panel starting at byte 32 (symbolic name: MCP2AxisTbl[1..22]). The machine axis numbers must be entered byte-wise here.

It is permissible to enter a value of 0 in the axis table. Checks are not made to find illegal axis numbers, meaning that false entries can lead to a PLC Stop.

For **FC19**, the **maximum possible number of axis selections** can also be restricted. This upper limit is set for the first machine control panel in DB10.DBW30 (symbolic name: MCP1MaxAxis) or for the second machine control panel in DB10.DBW54 (symbolic name: MCP2MaxAxis).

The default setting is 0, corresponding to the maximum number of configured axes. The axis numbers and the limit can also be adapted dynamically. Afterwards, a new axis must be selected for FC19. Axis numbers may not be switched over while the axes are traversing the relevant direction keys.
The compatibility mode is preset with axis numbers **1 to 9** for both MCPs and restricted to the configured number of axes.

Axis selection from the HMI

In addition to the 9 axis selection keys of the MCP, the active axis can also be selected in the HMI. The axis selection from the HMI is then only adopted for the MCP if the selected channel in the HMI correlates with that at FC19. With axis selection from the HMI, an initial attempt is made as with axis selection using the axis selection keys of the MCP, to find an assignment between the selected axis number and the actual machine axis in the axis table of the NC-DB, refer to section "Flexible axis configuration".

This will ensure that the same machine axis is selected with axis selection from the HMI as if the corresponding axis selection key of the MCP were selected. If an assignment is not

possible, then the axis number selected from the HMI is directly used as machine axis. As a consequence it is possible to select axes that cannot be selected via the MCP.

Single block mode

Pressing the single block key activates the single block mode, and if the key is pressed again, the single block mode is deactivated. If the mode is called several times for one MCP, the block transfers single block mode state from the channel of the first call to the channels of the subsequent calls.

Example

More than nine axes are to be controlled with FC19 using a special application. We recommend that you proceed as follows:

- Reserve free key on MCP
- Evaluate this key as a flip-flop
- Evaluate the flip-flop output as positive and negative edge
- For a positive edge write one set of axis numbers in the axis table (DB10) and switch on LED via this key
- For a negative edge write a different set of axis numbers in the axis table (DB10) and switch off LED via this key

Declaration of the function

```
FUNCTION FC19: VOID             //Symbolic name: MCP_IFM

   VAR_INPUT
      BAGNo :          BYTE ;
      ChanNo:          BYTE ;
      SpindleIFNo:     BYTE ;
   END_VAR

   VAR_OUTPUT
      FeedHold :       BOOL;
      SpindleHold :    BOOL;
   END_VAR

BEGIN
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
BAGNo :	I	BYTE	B#16#00 - B#16#0A	**1st MCP**: Number of mode group to which the mode signals are transferred
			B#16#10 - B#16#1A	**2nd MCP**: Number of mode group to which the mode signals are transferred
ChanNo:	I	BYTE	B#16#00 - B#16#0A	Number of the channel to which the channel signals are transferred
SpindleIFNo:	I	BYTE	0 - 31 (B#16#1F)	Number of the axis/spindle to which the spindle data is transferred (number of the associated machine axis)
FeedHold :	O	BOOL	0 (FALSE), 1 (TRUE)	Feed stop from MCP, modal
SpindleHold :	O	BOOL	0 (FALSE), 1 (TRUE)	Spindle stop from MCP, modal

MCP selection signals to the user interface

Table 16-3 Keyswitch

Source: MCP - Switch	Target: Interface DB
Position 0	DB10.DBX56.4
Position 1	DB10.DBX56.5
Position 2	DB10.DBX56.6
Position 3	DB10.DBX56.7

Table 16-4 Operating modes and machine functions

Source: MCP - Key	Target: Interface DB (parameter BAGNo) Display for BAG 1
AUTOMATIC	DB11.DBX0.0
MDI	DB11.DBX0.1
JOG	DB11.DBX0.2
REPOS	DB11.DBX1.1
REF	DB11.DBX1.2
TEACH IN	DB11.DBX1.0
INC 1 ... 10 000, INC Var.	DB11.DBX2.0 - 2.5

Table 16-5 Direction keys rapid traverse override

Source: MCP - Key	Target: Interface DB (parameter ChanNo)
Direction key +	DB21, ...DBX12.7
Direction key -	DB21, ...DBX12.6
Rapid traverse override	DB21, ...DBX12.5
Direction key +	DB21, ...DBX16.7
Direction key -	DB21, ...DBX16.6
Rapid traverse override	DB21, ...DBX16.5
Direction key +	DB21, ...DBX20.7
Direction key -	DB21, ...DBX20.6
Rapid traverse override	DB21, ...DBX20.5

Source: MCP - Key	Target: Interface DB (all axis DBs)
Direction key +	DB31, ...DBX4.7
Direction key -	DB31, ...DBX4.6
Rapid traverse override	DB31, ...DBX4.5

The transfer is dependent upon the selected axis. The associated interface bits are deleted for axes which are not selected.

Table 16-6 Override

Source: MCP - Switch	Target: Interface DB (parameter ChanNo)
Feedrate override	DB21, ...DBB4

Source: MCP - Switch	Target: Interface DB (all axis DBs)
Feedrate override	DB31, ...DBB0 (selected axis number) The feed override of the 1st MCP is applied to all axes.
Spindle override	DB31, ...DBB19 (parameter SpindleIFNo)

Table 16-7 Channel signals

Source: MCP keys	Target: Interface DB (parameter ChanNo)
NC start	DB21, ...DBX7.1
NC stop	DB21, ...DBX7.3
RESET	DB21, ...DBX7.7
Single block	DB21, ...DBX0.4

Table 16-8 Feedrate, spindle signals

Source: MCP keys	Target: FC output parameters
Feed stop Feed enable	Parameter: "FeedHold" linked with memory, LEDs are controlled
Spindle stop Spindle enable	Parameter: "SpindleHold" linked with memory, LEDs are controlled

Table 16-9 Cartesian manual traversing

Source: MCP keys	Target: Interface DB (parameter ChanNo)
R11 direction key	DB21, ...DBB392
Direction key +	DB21, ...DBX20.7
Direction key -	DB21, ...DBX20.6

Checkback signals from user interface for controlling displays

Table 16-10 Operating modes and machine functions

Target: MCP - LED	Source: Interface DB (parameter BAGNo) Display for BAG 1
AUTOMATIC	DB11.DBX6.0
MDI	DB11.DBX6.1
JOG	DB11.DBX6.2
REPOS	DB11.DBX7.1
REF	DB11.DBX7.2
TEACH IN	DB11.DBX7.0

Target: MCP - LED	Source: Interface DB (parameter BAGNo) Display for BAG 1
INC 1 ... 10 000, INC Var.	DB11.DBX8.0 - 8.5

Table 16-11 Channel signals

Target: MCP - LED	Source: Interface DB (parameter ChanNo)
NC start	DB21, ...DBX35.0
NC stop	DB21, ...DBX35.2 or DB21, ...DBX35.3
Single block	DB21, ...DBX0.4

Note

Direction key LEDs are controlled by operating the direction keys.

Axis selection and WCS/MCS LEDs are controlled by operating the relevant key.

Call example

```
CALL FC19(     //Machine control panel M variants Signals to interface
   BAGNo :=          B#16#1,           // Mode group no. 1
   ChanNo :=         B#16#1,           // Channel no. 1
   SpindleIFNo :=    B#16#4,           // Spindle Interface Number = 4
   FeedHold :=       m22.0,            // Feed stop signal modal
   SpindleHold :=    db2.dbx151.0);    //Spindle stop modal in
                                       //message DB
```

With these parameter settings, the signals are sent to the first mode group, the first channel and all axes. In addition, the spindle override is transferred in the 4th axis/spindle interface. The feed hold signal is passed to bit memory 22.0 and the spindle stop signal to data block DB2, data bit 151.0.

Reconnecting the axis selections

To ensure a flexible assignment of the axis selection keys to the appropriate axis or spindle, FC19 **needs not be modified or reprogrammed**. The axis number simply has to be entered in axis table DB10.DBB8 and the following as required. The axis number simply has to be entered in axis table DB10.DBB8 and the following as required.

Example

The spindle is defined as the 4th machine axis and should be selected via axis key 9.

```
Solution:
The value 4 must be entered in DB10 byte (8+(9-1)) for the 4th axis.

CALL FC19(            // Signals to interface
   BAGNo :=           B#16#1,     // Mode group no. 1
   ChanNo :=          B#16#1,     // Channel no. 1
   SpindleIFNo :=     B#16#4,     // Spindle Interface Number = 4

   FeedHold :=        m30.0,      // Feed stop signal modal
   SpindleHold :=     m30.1);     //Spindle stop modal
```

16.24 FC21: Transfer - data exchange NC/PLC

16.24.1 Function

Block FC21 is used to exchange data between the PLC and NC. The data are immediately transferred when FC21 is called – not waiting until the next basic PLC program cycle starts.

The data transfer is activated by calling the block FC21 with parameter "Enable" =1

Functions

The block provides the following functions:

- Synchronized action signals: PLC → NC channel
- Synchronized action signals: NC channel → PLC
- Fast data exchange PLC-NC (read function in NC)
- Fast data exchange PLC-NC (write function in NC)
- Update signals to the NC channel
- Update signals to axes (data byte 2 of the user interface)
- Update signals to axes (data byte 4 of the user interface)

16.24.2 Declaration of the function

Declaration of the function

```
VAR_INPUT
          Enable : BOOL;
          Funct : BYTE ;
          S7Var : ANY ;
          IVar1 : INT ;
          IVar2 : INT ;
END_VAR

VAR_OUTPUT
          Error : BOOL;
          ErrCode : INT ;
END_VAR
```

16.24.3 Explanation of formal parameters

Explanation of formal parameters

Signal	Type	Type	Value range	Description	
Enable:	I	BOOL	0 (FALSE), 1 (TRUE)	1:	Transferring data
Funct:	I	BYTE	1, 2, 3, ... 7	1:	Synchronized actions at channel
				2:	Synchronized actions from channel
				3:	Read data
				4:	Write data
				5:	Control signals to channel
				6:	Control signals to axis
				7:	Control signals to axis
S7Var :	I	ANY	S7 data storage area	Depends on "Funct"	
IVAR1:	I	INT	---	Depends on "Funct"	
IVAR2:	I	INT	---	Depends on "Funct"	
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1:	An error is active
ErrCode:	O	INT	---	Depends on "Funct"	

16.24.4 Function 1, 2: Signals synchronized actions to / from Channel

Synchronized actions can be disabled or enabled by the PLC.

The data area is stored on the user interface in DB21, ...DBB300 ...307 (to channel) and DB21, ...DBB308 ...315 (from channel). The parameter "S7Var" is not evaluated for this function, but must be assigned an actual parameter (see call example). The data are transferred to/from the NC as soon as FC21 is processed.

Signal	Type	Type	Value range	Description	
Enable:	I	BOOL	0 (FALSE), 1 (TRUE)	1:	Transferring data
Funct:	I	BYTE	1, 2	1:	to channel
				2:	from channel
S7Var :	I	ANY	S7 data storage area	Not used	
IVAR1:	I	INT	1, 2, ... max. channel number	Channel number	
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1:	An error is active
ErrCode:	O	INT	1, 10	1:	"Funct" invalid
				10:	Channel number invalid

Call example:

```
FUNCTION FC100: VOID
VAR_TEMP
```

```
        myAny: ANY ;
END_VAR

BEGIN
NETWORK

//Deactivate synchronized actions with ID3, ID10 and ID31 in NC channel 1 :
SYAK:   AUF   DB21;
        SET;
        S     DBX300  //ID3
              .2;
        S     DBX301  //ID10
              .1;
        S     DBX303  //ID31
              .6;
        L     B#16#1
              ;
        T     MB11;
        SPA   TRAN;

//Synchronized actions from NCK channel 1:
SYVK:   L B#16#2;
        T MB11;
TRAN:   CALL FC21 (
             Enable   := M 10.0,      //if TRUE, FC 21 active
             Funct    := MB 11,
             S7Var    := #myAny,      //Not used
             IVAR1    := 1,           //Channel no.
             IVAR2    := 0,
             Error    := M 10.1,
             ErrCode  := MW 12);

END_FUNCTION
```

16.24.5 Function 3, 4: Fast data exchange PLC-NC

General

A separate, internal data area is provided as interface to allow the fast exchange of data between the NC and the PLC. The interface encompasses 4096 bytes. PLC access operations (reading/writing) are realized via FC21. The internal structure of the interface is solely defined by the user, and must have precisely the same definition on the NC and PLC sides.

This data can be accessed by the NC program with the commands $A_DBB[x], $A_DBW[x], $A_DBD[x], $A_DBR[x] (further information: List Manual, System Variables).

The concrete address in the data array is specified by a byte offset (0 to 4095) in parameter "IVAR1". In this case, the alignment must be selected according to the data format, i.e. a DWORD starts at a 4byte limit and a WORD at a 2byte limit. Bytes can be located at any offset within the data field. Individual bits cannot be accessed. FC21 converts them over to a byte access. Data type information and quantity of data are taken from the ANY parameter, transferred via S7Var.

Without additional programming-related measures, data consistency is only ensured for 1 and 2 byte access operations - both from the NCU and from the PLC. For 2-byte consistency this is true only for the data type WORD or INT, but not for the data type BYTE.

In the case of longer data types or transfer of arrays which should be transferred consistently, the semaphore byte must be used in parameter "IVAR2", which can be used by FC21 to determine the validity or consistency of a block. This handling must be supported by the NC, i.e. in the NC program, by writing or deleting the semaphore byte. The semaphore byte is stored in the same data field as the user data.

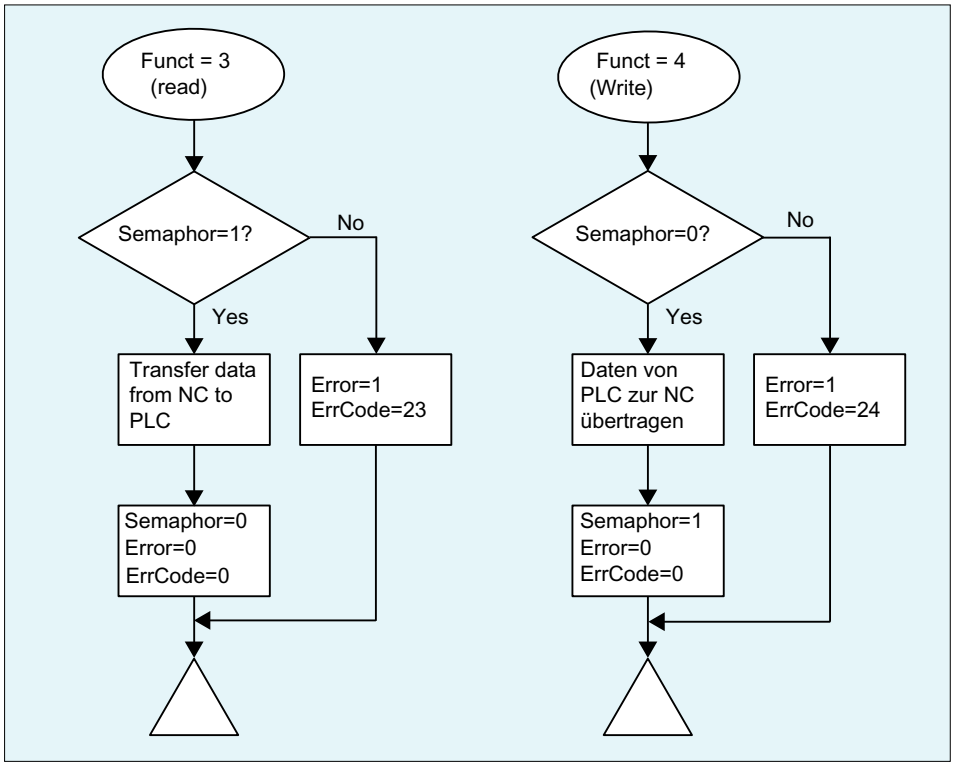
The semaphore byte is identified by a value between 0 and 4095 in "IVAR2".

The PLC reads and writes to the semaphore byte via FC21 in the same call, in which the user data should be transferred. The PLC programmer only has to define the semaphore variable in the interface. For access from the NC via the NC program, the semaphore mechanism must be programmed using individual instructions according to the flow chart shown below. The sequence is different for reading and writing variables.

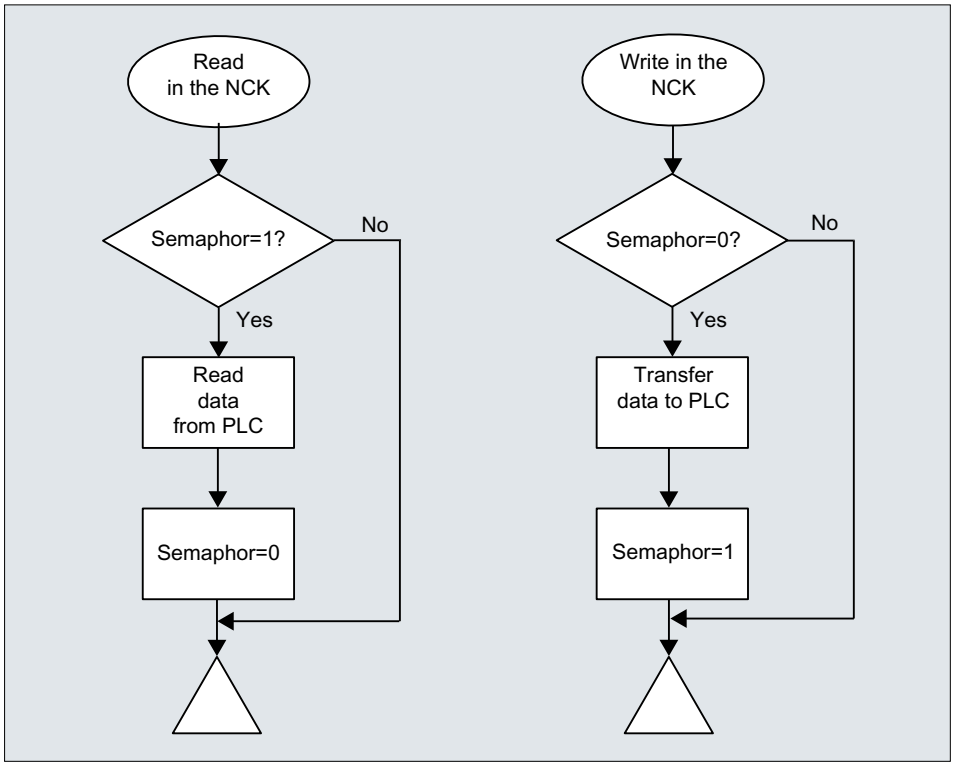
Only individual variables or arrays (fields) can be supported directly using the semaphore technique. Transferring structures must be split up into individual jobs. Here, the user must ensure data consistency of this structure by programming the appropriate semaphore mechanism.

If "IVAR2" = -1 is set, data are transferred without a semaphore.

Data exchange with semaphore in PLC (schematic of FC21)



Basic structure in the NC:

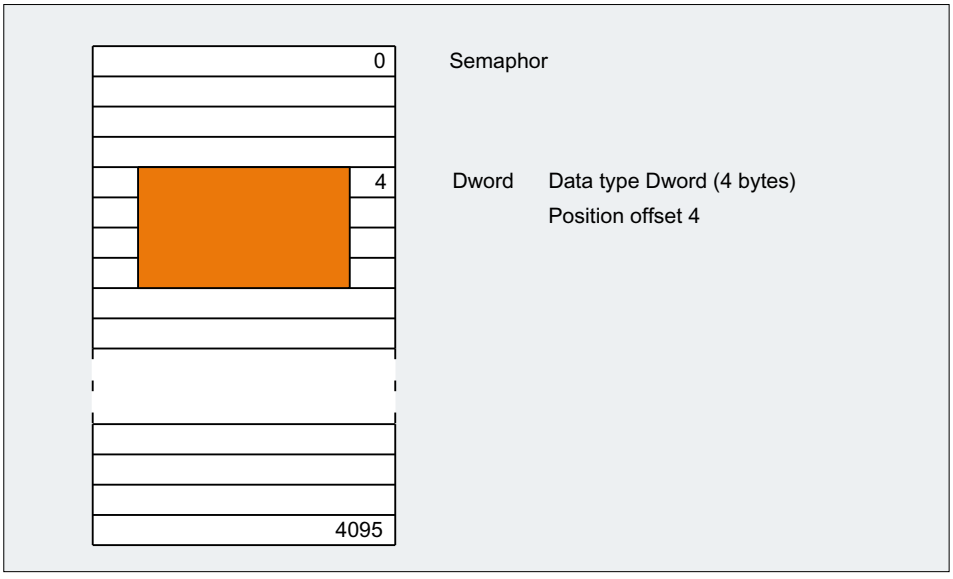


Variable value ranges

Signal	Type	Type	Value range	Meaning	
Enable:	I	BOOL	0 (FALSE), 1 (TRUE)	1:	Transferring data
Funct:	I	BYTE	3, 4	3:	Read data
				4:	Write data
S7Var :	I	ANY	S7 data area, except local data	Source/destination data storage area	
IVAR1:	I	INT	0 ... 4095	Position offset	
IVAR2:	I	INT	-1 ... 4095	Semaphore byte Transfer without semaphore: -1	
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1:	An error is active
ErrCode:	O	INT	20, 21, 22, 23, 24, 25	20:	Alignment error
				21:	illegal position offset
				22:	Illegal semaphore byte
				23:	No new data to be read
				24:	Cannot write data
				25:	Local data parameterized for S7Var

Example 1: Reading a DWORD from position offset 4 using a semaphore in byte 0 - and saving in memory double word 100

- Data type Dword (4 bytes)
- Position offset 4



PLC programming

```
CALL   FC21(
         Enable     := M 10.0,              // if TRUE, FC 21 is active
         Funct      := B#16#3,              //Read data
         S7Var      := P#M 100.0 DWORD 1,
         IVAR1      := 4,
         IVAR2      := 0,
         Error      := M 10.1,
         ErrCode    := MW12);
UN     M10.1;                //Enable while 1, until value is read
R      M10.0;
```

Programming the NC with synchronized actions

- Writing the data to the PLC - byte 0 serves as semaphore:
  ID=1 WHENEVER $A_DBB[0] == 0 DO $A_DBR[4] = $AA_IM[X] $A_DBB[0] = 1
- Reading the data from the PLC - byte 1 serves as semaphore:
  ID=2 WHENEVER $A_DBB[1] == 1 DO $R1 = $A_DBR[12] $A_DBB[1] = 0

Example 2: Reading a WORD from position offset 8 without semaphore, and saving in memory word 104

```
CALL   FC21(
         Enable     :=M 10.0,               // if TRUE, FC 21 is active
         Funct      :=B#16#3,               //Read data
         S7Var      :=P#M 104.0 WORD 1,
         IVAR1      :=8,
         IVAR2      :=-1,
         Error      :=M 10.1,
         ErrCode    :=MW12);
```

16.24.6 Function 5: Update control signals to channel

The purpose of this function is to transmit important control signals at high speed in between cyclic data transfers. Data bytes 6 and 7 of user interfaces DB21, ... are transferred to the NC. The channel is specified in parameter "IVAR1". This enable, for example, the feed disable, read-in disable to be transferred outside of the PLC cycle.

Signal	Type	Type	Value range	Description	
Enable:	I	BOOL	0 (FALSE), 1 (TRUE)	1:	Transferring data
Funct:	I	BYTE	5	5:	Control signals to channel
S7Var :	I	ANY	S7 data storage area	Not used	
IVAR1:	I	INT	1. Maxchannel	Channel number	
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1:	An error is active

Signal	Type	Type	Value range	Description	
ErrCode:	O	INT	1, 10	1:	"Funct" invalid
				10:	Channel no. invalid

16.24.7 Function 6: Update control signals to axes

The purpose of function 6 is to transmit important control signals at high speed in between cyclic data transfers. The **data byte 2** of application interface DB31, ... is transferred to the NC. The transfer is performed for all activated axes. This allows the controller enable to be transferred outside the PLC cycle, for example.

Signal	Type	Type	Value range	Description	
Enable:	I	BOOL	0 (FALSE), 1 (TRUE)	1:	Transferring data
Funct:	I	BYTE	6	6:	Control signals to axes
S7Var :	I	ANY	S7 data storage area	Not used	
IVAR1:	I	INT	0		
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1:	An error is active
ErrCode:	O	INT	1	1:	"Funct" invalid

16.24.8 Function 7: Update control signals to axes

The purpose of function 7 is to transmit important control signals at high speed in between cyclic data transfers. The **data byte 4** of application interface DB31, ... is transferred to the NC. The transfer is performed for all activated axes. This enables, for example, the feed stop to be transferred outside the PLC cycle.

Signal	Type	Type	Value range	Description	
Enable:	I	BOOL	0 (FALSE), 1 (TRUE)		
Funct:	I	BYTE	7	7:	Control signals to axes
S7Var :	I	ANY	S7 data storage area	Not used	
IVAR1:	I	INT	0		
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1:	An error is active
ErrCode:	O	INT	1	1:	"Funct" invalid

16.25 FC22: TM_DIR - direction selection for tool management

NOTICE
Use
Block FC22 "TM_DIR" may only be used in conjunction with the tool management.

Function

Referred to the location numbers, e.g. a tool magazine or revolver (indexing axis) block FC22 "TM_DIR" supplies the shortest path and direction of motion for positioning, based on the target and current position.

Outputs

- Input FC22: "Start" = 1 ⇒ the outputs are cyclically updated.
- Input FC22: "Start" = 0 ⇒ the outputs are undefined.

Special positioning

For direction selection with special positioning (input FC22: "Offset" > 0) a new target position is calculated from the target position, the offset for special positioning as well as the number of magazines locations:

New_target position = (target position - (special position -1)) neg. MODULO number_of_magazine locations

The new target position corresponds to the location number at which the magazine must be positioned so that the target position requested by the user corresponds to the location number of the special position.

The directional optimization is active both with and without special positioning.

Call

The block must be called once with the appropriate parameter settings for each magazine.

Further information

- Further PI services for tool management:
  - FB4: PI_SERV - request PI service (Page 173)
  - FC7: TM_REV - transfer block for tool change with revolver (Page 234)
  - FC8: TM_TRANS - transfer block for tool management (Page 238)
- Function Manual, Tool Management

Declaration of the function

```
FUNCTION FC22: VOID
// NAME:           TM_DIR
VAR_INPUT
```

```
    MagNo:          INT;
    ReqPos:         INT;
    ActPos:         INT;
    Offset:         BYTE ;
    Start :         BOOL;
END_VAR

VAR_OUTPUT
    Cw:             BOOL;
    Ccw:            BOOL;
    InPos :         BOOL;
    Diff:           INT;
    Error :         BOOL;
END_VAR
BEGIN
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
MagNo:	I	INT	1, 2, 3, ...	Magazine number
ReqPos:	I	INT	1, 2, 3, ...	Target position (magazine location number)
ActPos:	I	INT	1, 2, 3, ...	Actual position (magazine location number)
Offset:	I	BYTE	0, 1, 2, ...	Offset for special positioning
Start:	I	BOOL	0 (FALSE), 1 (TRUE)	1 = start of calculation
Cw:	O	BOOL	0 (FALSE), 1 (TRUE)	1 = Move magazine clockwise
Ccw:	O	BOOL	0 (FALSE), 1 (TRUE)	1 = Move magazine counterclockwise
InPos:	O	BOOL	0 (FALSE), 1 (TRUE)	1 = In position
Diff:	O	INT	0, 1, 2, ...	Absolute value of the differential path (shortest path)
Error:	O	BOOL	0 (FALSE), 1 (TRUE)	1 = error

Call example

```
CALL FC22(                  // Tool management direction selection
                            // Inputs
   MagNo :=      2,         // Magazine number
   ReqPos :=     mw 20,     //Target position
   ActPos :=     mw 22,     // Current position
   Offset :=     b#16#0,    // Offset for special positioning
```

```
Start :=      m 30.4,     // Start trigger
                          // Outputs
Cw :=         m 30.0,     // Move magazine clockwise
Ccw :=        m 30.1,     // Move magazine counterclockwise
InPos :=      m 30.2,     // Magazine in position
Diff :=       mw 32,      // Differential path
Error :=      m30.3       // Error has occurred
);
```

16.26 FC24: MCP_IFM2 - transferring MCP signals to the interface

Function

Block FC24 "MCP_IFM2" (M variant, e.g. MCP 310) is used for transferring data from the machine control panel to the NC/PLC interface:

- Modes
- Axis selections
- WCS/MCS switchover
- Traversing keys
- Overrides or override simulation signals
- Key-operated switch position

The following specifications apply to the **feedrate override, axis travel keys** and **INC keys** depending on the active mode or on the coordinate system selected:

- **Feedrate override:**
  - The feedrate override is transferred to the interface of the selected channel and to the interface of the axes.
  - The feedrate override signals are transferred to the NC channel in addition to the "Rapid traverse override" (DBB 5) interface byte if the "Feedrate override for rapid traverse effective" HMI signal is set (exception: Switch setting "Zero"). "Rapid traverse override effective" is also set with this HMI signal.
- **Machine functions for INC and axis travel keys:**
  - When the MCS is selected, the signals are transferred to the interface of the selected machine axis.
  - When the WCS is selected, the signals are transferred to the geometry axis interface of the parameterized channel.
  - When the system is switched between MCS and WCS, the active axes are generally deselected.

The associated LEDs on the machine control panel are derived from the acknowledgments from the relevant selections.

Feedrate and spindle Start/Stop are not transferred to the interface, but output modally as a "FeedHold" or "SpindleHold" signal. The user can link these signals to other signals leading to a feed or spindle stop (this can be implemented, e.g. using the appropriate input signals in FC10: AL_MSG). The associated LEDs are activated at the same time.

The **spindle direction** (+, -) is not switched directly either, but made available as output parameter "SpindleDir". permitting, for example, FC18 to be parameterized. A spindle enable signal is also switched via parameter "SpindleHold". One possible method of moving a spindle directly is to preselect it as an axis so that it can be traversed via (axis) direction keys.

If the machine control panel fails, the signals it outputs are preset to zero; this also applies to "FeedHold" and "SpindleHold" output signals.

FC24 or also FC19, FC25, FC26 can be called a multiple number of times in a single PLC cycle. In this case, the first call in the cycle drives the LED displays. Furthermore, all actions of the parameterized block are carried out in the first call. In the following calls, only a reduced level of processing of the channel and mode group interface takes place. The geometry axes are supplied with directional data only in the first block call in the cycle.

Single block processing can be selected/deselected only in the first call in the cycle.

The second machine control panel can be processed if parameter "ModeGroupNo" has been increased by B#16#10. When parameterizing, the HHU number is contained in the lower nibble (lower 4 bits).

"BAGNo" = 0 or B#16#10 ⇒ mode group signals are not processed.

"ChanNo" = 0 ⇒ no processing of the channel signals.

The INC selections are transferred to the mode group interface. The activation for this specification is done via the DB10.DBX57.0 (INC inputs in BAG area active) through this block once after power up.

Furthermore, two machine control panels can be handled in parallel by this block. The block call for the 2nd machine control panel in OB1 cycle must be set after the call of the 1st MCP. Support for two MCPs is provided in the control panel blocks up to certain limits (support is not provided as standard for mutual interlocking of axis selections with identical assignments on two MCPs).

Key-operated switch position

As of software version 4.5 SP2, the key-operated switch signals in the FC24 are also transferred to the user interface (DBX56.5 to 7). This transfer is made independent of whether a keyswitch is mounted on the MCP.

Note

For further information see "FC19: MCP_IFM - transfer of MCP signals to interface (Page 267)".

Flexible axis configuration

It is possible to be flexible in the assignment of axis selections or direction keys for machine axis numbers.

Better support is now provided by the MCP blocks for the use of two MCPs, which are to run in parallel, in particular for an application using two channels and two mode groups. Note that the

axis-numbers are also specified in the parameterized mode group number of the MCP block in the axis tables of the relevant MCP.

To provide this flexibility, tables for axis numbers are stored in DB10.
For the **1st** machine control panel (MCP) the table starts at byte 8 (symbolic name: MCP1AxisTbl[1..22]) and for the **second** machine control panel (MCP) starting at byte 32 (symbolic name: MCP2AxisTbl[1..22]). The machine axis numbers must be entered byte by byte here. It is permissible to enter a value of 0 in the axis table. Checks are not made to find illegal axis numbers, meaning that false entries can lead to a PLC Stop.

For **FC24**, the **maximum possible number of axis selections** can also be restricted.
This upper limit is set for the first machine control panel in DB10.DBW30 (symbolic name: MCP1MaxAxis) or for the second machine control panel in DB10.DBW54 (symbolic name: MCP2MaxAxis) for the respective MCP.

The default setting is 0, corresponding to the maximum number of configured axes. The axis numbers and the limit can also be adapted dynamically. Afterwards, a new axis must be selected for FC24. Axis numbers may not be switched over while the axes are traversing the relevant direction keys. The compatibility mode is preset with axis numbers **1 to 6** for both MCPs and restricted to the configured number of axes.

Axis selection from the HMI

In addition to the 6 axis selection keys of the MCP, the active axis can also be selected in the HMI.

If an axis is being selected from the HMI, then no axes can be traversed using the direction keys. Axis selection from the HMI is then only adopted for the MCP if the selected channel in the HMI correlates with that at FC25. With axis selection from the HMI, an initial attempt is made as with axis selection using the axis selection keys of the MCP, to find an assignment between the selected axis number and the actual machine axis in the axis table of the NC-DB, refer to section "Flexible axis configuration".

This means that when an axis is selected from the HMI, the same machine axis is selected as if the corresponding axis selection key of the MCP had been selected. If an assignment is not possible, then the axis number selected from the HMI is directly used as machine axis. As a consequence it is possible to select axes that cannot be selected via the MCP.

Single block mode

Pressing the single block key activates the single block mode, and if the key is pressed again, the single block mode is deactivated. If the mode is called several times for one MCP, the block transfers single block mode state from the channel of the first call to the channels of the subsequent calls.

Declaration of the function

```
FUNCTION FC24: VOID
// NAME:                MCP_IFM2

VAR_INPUT
    BAGNo :              BYTE ;
```

```
    ChanNo:             BYTE ;
    SpindleIFNo:        BYTE ;
END_VAR

VAR_OUTPUT
    FeedHold :          BOOL;
    SpindleHold :       BOOL;
    SpindleDir:         BOOL;
END_VAR

BEGIN
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
BAGNo :	I	BYTE	B#16#00 - B#16#0A	**1st MCP**: Number of mode group to which the mode signals are transferred
			B#16#10 - B#16#1A	**2nd MCP**: Number of mode group to which the mode signals are transferred
ChanNo:	I	BYTE	B#16#00 - B#16#0A	Number of the channel to which the channel signals are transferred
SpindleIFNo:	I	BYTE	0 - 31 (B#16#1F)	Number of the axis/spindle to which the spindle data is transferred (number of the associated machine axis)
FeedHold :	O	BOOL	0 (FALSE), 1 (TRUE)	Feed stop from MCP, modal
SpindleHold :	O	BOOL	0 (FALSE), 1 (TRUE)	Spindle stop from MCP, modal
SpindleDir:	O	BOOL	0 (FALSE), 1 (TRUE)	Direction of spindle rotation
				0: Direction of rotation + (left)
				1: Direction of rotation - (right)

Call example

```
CALL FC24(                          // Slimline machine control panel M
                                    variant
                                    // Signals to interface
    BAGNo :=       B#16#1,          // Mode group no. 1
    ChanNo :=      B#16#1,          // Channel no. 1
    SpindleIFNo :  B#16#4,          // Spindle Interface Number = 4
    =
    FeedHold :=    m22.0,           // Feed stop signal modal
    SpindleHold :  db2.dbx151.0,    // Spindle stop modal in message data block
    =
    SpindleDir:=   m22.1);          // Spindle direction return
```

With these parameter settings, the signals are sent to the first mode group, the first channel and all axes. In addition, the spindle override is transferred in the 4th axis/spindle interface. The feed hold signal is passed to bit memory 22.0 and the spindle stop signal to data block DB2, data bit 151.0. The spindle direction feedback signal supplied via parameter "SpindleDir" can be used as a direction input for an additional FC18 call.

16.27 FC25: MCP_IFT - transfer of MCP/OP signals to interface

Function

Block FC25 "MCP_IFT" (T variant, e.g. MCP 483) is used for transferring data from the machine control panel to the NC/PLC interface:

- Modes
- Direction keys of four axes
- WCS/MCS switchover
- Overrides
- Keyswitch

The following specifications apply to the **feedrate override, axis travel keys** and **INC keys** depending on the active mode or on the coordinate system selected:

- **Feedrate override:**
  - The feedrate override is transferred to the interface of the selected channel and to the interface of the axes.
  - The feedrate override signals are transferred to the NC channel in addition to the "Rapid traverse override" (DBB 5) interface byte if the "Feedrate override for rapid traverse effective" HMI signal is set (exception: Switch setting "Zero"). "Rapid traverse override effective" is also set with this HMI signal.
- **Machine functions for INC and axis travel keys:**
  - When the MCS is selected, the signals are transferred to the interface of the selected machine axis.
  - When the WCS is selected, the signals are transferred to the geometry axis interface of the parameterized channel.

The associated LEDs on the machine control panel derived from the acknowledgments of the relevant selections.

Feedrate and spindle Start/Stop are not transferred to the interface, but output modally as a "FeedHold" or "SpindleHold" signal. The user can link these signals to other signals leading to a feed or spindle stop (this can be implemented, e.g. using the appropriate input signals in FC10: AL_MSG). The associated LEDs are activated at the same time.

If the machine control panel fails, the signals it outputs are preset to zero; this also applies to "FeedHold" and "SpindleHold" output signals.

FC25 or also FC19, FC24, FC26 can be called a multiple number of times in a single PLC cycle. In this case, the first call in the cycle drives the LED displays. Furthermore, all actions of the parameterized block are carried out in the first call. In the following calls, only a reduced level

of processing of the channel and mode group interface takes place. The geometry axes are supplied with directional data only in the first block call in the cycle.

Single block processing can be selected/deselected only in the first cycle.

The second machine control panel can be processed if parameter "ModeGroupNo" has been increased by B#16#10. When parameterizing, the HHU number is contained in the lower nibble (lower 4 bits).

"BAGNo" = 0 or B#16#10 ⇒ mode group signals are not processed.

"ChanNo" = 0 ⇒ no processing of the channel signals.

Flexible axis configuration

It is possible to be flexible in the assignment of axis selections or direction keys for machine axis numbers.

Support is now provided by the MCP blocks for the use of two MCPs, which are operated simultaneously, in particular for an application using two channels and two mode groups. The block call for the second machine control panel in OB1 cycle must be set after the call of the 1st MCP. Note that the axis numbers are also specified in the parameterized mode group number of the MCP block in the axis tables of the relevant MCP.

To provide this flexibility, tables for axis numbers are stored in DB10.
For the **1st** machine control panel (MCP), the table starts at byte 8 (symbolic name: MCP1AxisTbl[1..22]) and for the **2nd** machine control panel (MCP), from byte 32 (symbolic name: MCP2AxisTbl[1..22]). The machine axis numbers are entered here byte-by-byte. It is permissible to enter a value of 0 in the axis table. Checks are not made to find illegal axis numbers, meaning that false entries can lead to a PLC Stop.

The restriction of the **possible number of axes at FC25** is realized using the 0-values in the axis table. The axis numbers can also be adapted dynamically. During the manual traversing of axes using the direction keys, the axis numbers must not be switched over. The compatibility mode is preset with axis numbers **1 to 4** for both MCPs and restricted to the configured number of axes.

Note

For further information see "FC19: MCP_IFM - transfer of MCP signals to interface (Page 267) ".

Declaration of the function

```
FUNCTION FC25: VOID
// NAME:              MCP_IFT

VAR_INPUT
    BAGNo :           BYTE ;
    ChanNo:           BYTE ;
    SpindleIFNo:      BYTE ;
END_VAR
```

```
VAR_OUTPUT
    FeedHold :         BOOL;
    SpindleHold :      BOOL;
END_VAR

BEGIN
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
BAGNo :	I	BYTE	B#16#00 - B#16#0A	**1st MCP**: Number of mode group to which the mode signals are transferred
			B#16#10 - B#16#1A	**2nd MCP**: Number of mode group to which the mode signals are transferred
ChanNo:	I	BYTE	B#16#00 - B#16#0A	Number of the channel to which the channel signals are sent
SpindleIFNo:	I	BYTE	B#16#00 - B#16#1F	Number of the axis/spindle to which the spindle data is transferred (number of the associated machine axis)
FeedHold :	O	BOOL	0 (FALSE), 1 (TRUE)	Feed stop from MCP, modal
SpindleHold :	O	BOOL	0 (FALSE), 1 (TRUE)	Spindle stop from MCP, modal

Call example

With this parameter assignment example, the signals are sent to the 1st mode group, the 1st channel and all axes. The spindle override is transferred to the 4th axis/spindle. Feed stop is sent to bit memory 22.0 and spindle stop is sent to the data block DB2, DBX151.0.

```
CALL FC25(                          //Machine control panel T variants
                                    // Signals to interface
  BAGNo :=        B#16#1,           // Mode group no. 1
  ChanNo :=       B#16#1,           // Channel no. 1
  SpindleIFNo :=  B#16#4,           // Spindle Interface Number = 4
  FeedHold :=     m22.0,            // Feed stop signal modal
  SpindleHold :=  db2.dbx151.0)     // Spindle stop modal in message data block
                  ;
```

16.28 FC26: HPU_MCP - Transfer of the HT 8/HT 10 signals to the interface

General description of functions

The function FC26 "HPU_MCP (machine control panel signals of the handheld unit HT 8)" transfers the HT 8-specific or HT 10-specific signals of the following functions between the HT 8 or HT 10 input/output data range parameterized in the function block FB1 (parameter: MCPxIn and MCPxOut) and the NC/PLC-interface:

- Modes
- Machine function INC
- Coordinate system WCS or MCS
- Axial traverse key
- Axis selection
- Feedrate override
- Rapid traverse override
- Keyswitch information

Note

Mode switchover through HT 8 and/or HMI

The function FC2 "GP_HP Basic program, cyclic part" transfers the signals of the mode switchover in such a way that an alternative selection from MCP of HT 8 and of the HMI is possible. The transfer of the HMI signals to the NC/PLC interface can also be deactivated in the function block FB1 with the parameter "MMCToIF" = FALSE .

Active axes

Using HT 8 a maximum of 6 axes can be addressed at the same time. The selection of the axes is to be realized by the user/machine manufacturer in the PLC user program.

Bit 6.7 in the input image is used to make a differentiation between the HT 8 and HT 10. This bit is set for an HT 10.

Flexible axis configuration for the HT 8

The function FC26 enables a flexible assignment of the machine axes to the traversing keys or to the axis selection. 2 tables are available in DB10 for this purpose:

- Machine axis table, 1st MCP: DB10.DBB8 to DBB13 (Table of the machine axis number) Symbolic name: MCP1AxisTbl[1..22]
- Machine axis table, 2nd MCP: DB10.DBB32 to DBB37 (Table of the machine axis number) Symbolic name: MCP2AxisTbl[1..22]

In the tables the axis numbers n (with n = 1, 2, ...) of the active machine axis are to be entered byte-wise. The value 0 must be entered in the unused table locations.

The table length can be specified to the FC26:

- 1st MCP: DB10.DBB30 (upper limit of the machine axis table)
- 2nd MCP: DB10.DBB54 (upper limit of the machine axis table)

A value of 4, for example, means that FC26 takes into account only the first 4 table entries or machine axes. The maximum value for the FC26 is 6. For a value of 0 or values greater than 6 the maximum value is taken implicitly.

Note

Please note the following constraints:

- A check of the permissible machine axis numbers is not done. Invalid machine axis numbers can lead to a PLC stop.
- The machine axis numbers can be changed dynamically. The table may not be written, if currently a machine axis is being moved via a traversing key.

Flexible axis configuration for the HT 10

Contrary to the HT 8, the HT 10 only has one plus and one minus key for traversing. There is also a rapid key for rapid traversing. The active axis, to which these traversing signals refer, must be selected at the HMI.

If an axis is being selected from the HMI, then no axis can be traversed using the direction keys. Axis selection from the HMI is then only adopted for the MCP if the selected channel in the HMI correlates with that at FC26.

If FC26 is called for an HT 10 with ChanNo <= 0, no axes can be traversed as the current axis cannot be determined from the channel DB.

Declaration of the function

```
FUNCTION FC26: VOID
// NAME:         HPU_MCP

VAR_INPUT
    BAGNo :      BYTE ;
    ChanNo:      BYTE ;
END_VAR

BEGIN
END_FUNCTION
```

Description of formal parameters

Signal	Type	Type	Value range	Meaning
BAGNo :	I	BYTE	B#16#00 - B#16#0A	**1st MCP**: Number of mode group to which the mode signals are transferred
			B#16#10 - B#16#1A	**2nd MCP**: Number of mode group to which the mode signals are transferred
ChanNo:	I	BYTE	B#16#00 - B#16#0A	Number of the channel to which the channel signals are transferred 0: No transfer of channel signals (and thus no axis can be traversed in JOG mode for HT 10)

Call examples

Calling FC26 for the first MCP, the first mode group and the first channel of the NC.

```
CALL FC26(                          //Machine control panel of the HT 8
          BAGNo :=    B#16#01,      //1.MCP, 1.BAG
          ChanNo :=   B#16#01);     //Channel 1
```

Calling FC26 for the second MCP, the second mode group and the third channel of the NC.

```
CALL FC26(                          //Machine control panel of the HT 8
          BAGNo :=    B#16#12,      //2.MCP, 2.BAG
          ChanNo :=   B#16#03 );    //Channel 3
```

Transfer of the traversing key signals depending upon the active coordinate system

The traversing key signals for 6 axes lie in the HT 8 input data area below:

- EB n + 2, Bit 0 - Bit 5 (positive traversing direction)
- EB n + 3, Bit 0 - Bit 5 (negative traversing direction)

The switchover of the coordinate system is done via the input signal:

- EB n + 0, Bit 0 (MCS/WCS)

The input signal is evaluated in FC26 with the help of the edge trigger flag. The active coordinate system is shown in the following output signal:

- AB n + 0, Bit 0 (MCS/WCKS) with 0 = MCS, 1 = WCS

In case of active MCS the traversing key signals of the axes 1 - 6 are transferred in the axis-specific interfaces (DB31, ...DBX4.6 and DBX4.7 (traversing key +/-)) of the axes specified in the machine axis tables (DB10.DBB8 to DBB13 or DBB32 to DBB37).

For HT 8 with active WCS, it is assumed that the axes 1 - 3 of the machine axis table are geometry axes. For the traverse key signals with HT 8, the following applies:

- The traverse key signals of axes **1** - **3** (EB n + 2 / 3, bit 0 - bit 2) are transferred into the interface of the geometry axes in DB21, ...DBB12 + (n * 4), with n = 0, 1, 2), bit 6 and bit 7 (traverse keys +/-) of the channel specified with parameter "ChanNo".
  The assignment of the traversing key signals of axes 1, 2 and 3 to the geometric axes 1, 2 and 3 of the channel is permanent and may not be changed.
- The traverse key signals of axes **4** - **6** (EB n + 2 / 3, bit 3 - bit 5) are transferred into the axis-specific interfaces (DB31, ...DBX4.6 and DBX4.7 (traverse keys +/-)) of axes 4 - 6 entered in the machine axis table (DB10.DBB**11** to DBB**13** or DBB**35** to DBB**37**).

For HT 10, only one geometry axis which was selected from the HMI can be traversed respectively.

The axis traverse signals (+/- and rapid traverse) for HT 10 relate to a machine axis or a geometry axis, depending on which axis was selected in the HMI and whether WCS or MCS is active. Refer also to "Flexible axis configuration for HT 10".

No traversing of machine axes in WCS

In case of active WCS (AB n + 0, Bit 0 = 1) the traversing of the machine axes can be locked. For this, the following output signals are to be set in the PLC user program:

- AB n + 3, Bit 7 = 1 (For WCS: no machine axes)
  Requirement to the FC26, not to transfer any traversing key signals for the machine axes. The traversing key signals for the axes 1 - 3 of the machine axis table are transferred to the geometric axes 1 - 3 of the specified channel. The traversing key signals for the axes 4 - 6 of the machine axis table are not transferred.
- AB n + 2, Bit 6 (axes 7 - n selected)
  Requirement to the FC26 not to transfer any traversing key signals, since the axes 1 - 6 of the machine axis table are switched over. The axes 1 - 3 are thus not geometric axes, but instead also machine axes.

Feedrate override

The value of the HT 8 override switch is transferred as feedrate override in the channel-specific interface DB21, ...DBB4 (feedrate override) of the programmed channel (parameter: "ChanNo") and in the axis-specific interfaces DB31, ...DBB0 (feedrate override) of the axes programmed in the table DB10.DBB8 to DBB13 (machine axis number).

Rapid traverse override

If for the programmed channel (parameter: "ChanNo") the signal DB21, ...DBX25.3 = 1 (feedrate override for rapid traverse) is set, the value of the HT 8 override switch is sent as a rapid traverse override to its channel-specific interface in DB21, ...DBB5 (rapid traverse override) and the signal DB21, ...DBX6.6 = 1 (rapid traverse override active) is also set.

Machine function INC

The HT 8 signals of the machine function INC are transferred differently depending upon the active coordinate system MCS or WCS:

- Active coordinate system: MCS
  The selected machine function INC is transferred for all 6 axes in the axis-specific interfaces in DB31, ...DBX5.0 to DBX5.5 (machine function) of the axes programmed in the table in DB10.DBB8 to DBB13 (machine axis numbers) .
- Active coordinate system: WCS
  For the axes 1 to 3 the signals of the machine function INC are transferred in the channel-specific. Interface in DB21, ...DBX13.0 to DBX13.5 (machine function) of the programmed channel (Parameter: "ChanNo").
  For the axes 4 to 6 the signals of the machine function INC are transferred in the channel-specific. interfaces in DB31, ...DBX5.0 to DBX5.5 (machine function) of the axes programmed in the table in DB10.DBB11 to DBB13 (machine axis numbers).

The selection signals of the INC machine functions are transferred in the mode group-specific interface DB11.DBB 2 + (n * 20), bit 0 to bit 5 (with n = 0, 1, 2, ...). The FC26 informs the NC about the activation of the mode group interface for the INC machine functions once after power-up with DB10.DBX57.0 (INC inputs active in the mode group area).

Handwheel selection

The hand-wheel selection signals are evaluated by HMI and transferred to the corresponding NC/PLC interface signals of the machine or geometric axes:

- Geometry axes: DB21, ... DBB 12 + (n * 4), bit 0 to bit 2 (with n = 0, 1, 2)
- Machine axes: DB31, ...DBX4.0 to DBX4.2

Requirement: FB1 parameters: "HWheelMMC" = TRUE

Multiple call in one PLC cycle

FC26 can be called a multiple number of times in a single PLC cycle. Upon the first call in the PLC cycle:

- All actions of the parameterized blocks are executed
- The LED signals are written in the output area
- In case of selected WCS, the traversing key signals of the geometric axes are written
- The signals for the selection and deselection of the individual block are processed

When FC26 is called more often, then the channel and mode group interface are processed at a reduced rate.

Processing of two MCP

If the function FC26 is called twice for two MCP in the cyclic sequence of the PLC program (organization block OB1), the call for the second MCP must be made after the call for the first MCP.

Note

If an axis can be traversed from two MCP, then the implementation of a mutual interlocking is the responsibility of the user (machine manufacturer).

Failure of the MCP of the HT 8 or HT 10

If the MCP of HT 8 or HT 10 fails, all input signals are set to the value 0.

16.28.1 Overview of the NC/PLC interface signals of HT 8

Operating modes and machine functions

Source: MCP	Target: Programmed mode group (Parameter BAGNo) Display for BAG 1
AUTOMATIC	DB11.DBX0.0
MDI	DB11.DBX0.1
JOG	DB11.DBX0.2
REPOS	DB11.DBX1.1
REF	DB11.DBX1.2
TEACH IN	DB11.DBX1.0
INC 1 ... 10 000, INC Var.	DB11.DBX2.0 - DBX 2.5

Traversing keys and rapid traverse override

Source: MCP	Target: Geometry axes of the prog. channel (parameter: ChanNo)
Traversing key +	DB21, ...DBX12.7
Traversing key -	DB21, ...DBX12.6
Rapid traverse override	DB21, ...DBX12.5
Traversing key +	DB21, ...DBX16.7
Traversing key -	DB21, ...DBX16.6
Rapid traverse override	DB21, ...DBX16.5
Traversing key +	DB21, ...DBX20.7
Traversing key -	DB21, ...DBX20.6
Rapid traverse override	DB21, ...DBX20.5

Source: MCP	Target: Prog. Axes according to the table in DB10, DBB8 - 13 (1st MCP) or DBB32 - 37 (2nd MCP)
Traversing key +	DB31, ...DBX4.7
Traversing key -	DB31, ...DBX4.6
Rapid traverse override	DB31, ...DBX4.5; HT 10 only

Override

Source: MCP	Target: Programmed channel (Parameter: ChanNo)
Feedrate override	DB21, ...DBB4

Source: MCP	Target: Prog. Axes according to the table in DB10, DBB8 - 13 (1st MCP) or DBB32 - 37 (2nd MCP)
Feedrate override	DB31, ...DBB0

Channel signals

Source: MCP	Target: Programmed channel (Parameter: ChanNo)
NC start	DB21, ...DBX7.1
NC stop	DB21, ...DBX7.3
RESET	DB21, ...DBX7.7
Single block	DB21, ...DBX0.4

Note

Unlike other MCP models, the HT 8 does not indicate the feedback messages via output signals with LEDs.

Output signals from the axis traversing key pairs are not switched.

Output signals from the WCS and single block are switched after the keys have been pressed.

The output signal from Reset is not switched. (HT 8)

The LEDs of the direction keys +/- and rapid traverse are controlled by pressing the keys. (HT 10)

16.28.2 Overview of the NC/PLC interface signals of HT 8

Operating modes and machine functions

Destination: MCP	Source: Interface-DB (parameter BAGNo) Display for BAG 1
AUTOMATIC	DB11.DBX6.0
MDI	DB11.DBX6.1
JOG	DB11.DBX6.2
REPOS	DB11.DBX7.1
REF	DB11.DBX7.2
TEACH IN	DB11.DBX7.0

16.29 FC1005: AG_SEND - transfers data to Ethernet CP

Function

The AG_SEND function block transfers the data to be transferred via a configured connection to the Ethernet CP. Together with the AG_RECV function block, data exchange can be established with another station via the integrated "CP 840D sl." This station must be configured in STEP 7, "NetPro."

In the basic program, this function is available as a function block FC1005. This is roughly equivalent to function block FC5 in the "SIMATIC_NET_CP" library.

The TCP, UDP, and ISO-on-TCP protocols are supported.

Note

Smaller volume of transmittable data with SINUMERIK CP and UDP or ISO-on-TCP protocol

For SINUMERIK CP, only **240** bytes can be transmitted when function block FC1005 and the UDP or ISO-on-TCP protocol are used.

Description of formal parameters

Signal	Type	Type	Value range	Description
ACT:	I	BOOL	0 (FALSE), 1 (TRUE)	Job initiation [1]
ID:	I	INT	1, 2, 3 ...16	Connection ID
LADDR:	I	WORD	-	Module start address [2]
SEND:	I	ANY	-	Specifies the address and length. The address of the data area alternatively refers to: • Bit memory address area • Data block area

Signal	Type	Type	Value range	Description
LEN:	I	INT	TCP: 1, 2, ... 8192 UDP: 1, 2, ... **240** ISO-on-TCP: 1, 2. ... **240**	Number of bytes that are transmitted
DONE:	O	BOOL	0 (FALSE), 1 (TRUE)	0: Job running 1: Job executed
ERROR:	O	BOOL	0 (FALSE), 1 (TRUE)	0: No error 1: An error is active
STATUS:	O	WORD	-	Status display
1) Parameter ACT must be TRUE long enough until the following applies: (DONE == 1) OR (ERROR == 1) 2) For SINUMERIK 840D sl: Parameter LADDR := **W#16#8110**				

Documentation

A detailed module description can be found in:

- SINUMERIK user interfaces: Online help
- SIMATIC Programming Manual: Program blocks for SIMATIC NET S7-CPs
  Chapter: "Program blocks for Industrial Ethernet" > "Program blocks for open communication services (SEND/RECEIVE interface)" > "AG_SEND / AG_LSEND / AG_SSEND"

16.30 FC1006: AG_RECV - receives data from the Ethernet CP

Function

The function block AG_RECV receives data transferred via a configured connection from the Ethernet CP. Together with the AG_SEND function block, data exchange can be established with another station via the integrated "CP 840D sl." This station must be configured in STEP 7, "NetPro."

In the basic program, this function is available as a function block FC1006. This is roughly equivalent to function block FC6 in the "SIMATIC_NET_CP" library.

The TCP, UDP, and ISO-on-TCP protocols are supported.

Note

Smaller volume of transmittable data with SINUMERIK CP and UDP or ISO-on-TCP protocol

For SINUMERIK CP, only **240** bytes can be transmitted when function block FC1006 and the UDP or ISO-on-TCP protocol are used.

Formal parameters

Signal	Type	Type	Value range	Description
ID:	I	INT	1, 2 ...16	Connection ID
LADDR:	I	WORD	-	Module start address [1]
RECV:	I	ANY	-	Specifies the address and length. The address of the data area alternatively refers to: • Bit memory address area • Data block area
NDR :	O	BOOL	0 (FALSE), 1 (TRUE)	0: Job running 1: New data accepted
ERROR:	O	BOOL	0 (FALSE), 1 (TRUE)	0: No error 1: An error is active
STATUS:	O	WORD	-	Status display
LEN:	O	INT	TCP: 1, 2, ... 8192 UDP: 1, 2, ... **240** ISO-on-TCP: 1, 2. ... **240**	Number of bytes that are transmitted
1) For SINUMERIK 840D sl: Parameter LADDR := **W#16#8110**.				

Documentation

A detailed module description can be found in:

- SINUMERIK user interfaces: Online help
- SIMATIC Programming Manual: Program blocks for SIMATIC NET S7-CPs
  Chapter: "Program blocks for Industrial Ethernet" > "Program blocks for open communication services (SEND/RECEIVE interface)" > "AG_RECV / AG_LRECV / AG_SRECV"

Data lists

17

17.1 Machine data

17.1.1 Display machine data

Number	Identifier: $MM_	Description
9032	HMI_MONITOR	Determining the PLC data for HMI monitor information

17.1.2 NC-specific machine data

Number	Identifier: $MN_	Description
10100	PLC_CYCLIC_TIMEOUT	Cyclic PLC monitoring time
14504	MAXNUM_USER_DATA_INT	Number of user data (INT)
14506	MAXNUM_USER_DATA_HEX	Number of user data (HEX)
14508	MAXNUM_USER_DATA_FLOAT	Number of user data (FLOAT)
14510	USER_DATA_INT	User data (INT)
14512	USER_DATA_HEX	User data (HEX)
14514	USER_DATA_FLOAT[n]	User data (FLOAT)

Note

Machine data in integer/hex format is operated in the NC as DWORD. A machine data in floating point format is managed in the NC as FLOAT (8-byte IEEE). They are stored only in the NC/PLC interface and can be read by the PLC user program from DB20 even when the PLC boots.

17.1.3 Channelspecific machine data

Number	Identifier: $MC_	Description
28150	MM_NUM_VDIVAR_ELEMENTS	Number of elements for writing PLC variables

Interface signals 18

18.1 Interface signals - overview

18.1.1 Overview of the PLC blocks

18.1.1.1 Organization blocks (OBs)

Table 18-1 Assignment of the organization blocks (OBs)

OB No.	Designation	Meaning	Package
1	CYCLE	Cyclic execution	GP
40	ALARM	Process alarms	GP
82	DIAGNOSTIC ALARM	Asynchronous error alarm	GP
86	RACK FAILURE	Asynchronous error alarm	GP
100	RESTART	Startup/restart	GP

18.1.1.2 Function blocks (FBs)

Table 18-2 Assignment of the function blocks (FBs)

Number	Designation	Meaning
0 - 29	---	Reserved for Siemens
1	RUN_UP	Basic program start-up
2	GET	Read NC variables
3	PUT	Write NC variables
4	PI_SERV	PI services
5	GETGUD	Read GUD variable
7	PI_SERV2	General PI services
9	M2N	M to N transition block
10	SI_relay	Safety Integrated relay
11	SI_Braketest	Safety Integrated brake test
29	Diagnostics	Signal recorder and data trigger diagnostics
30 - 999*	---	Free for user assignment

Number	Designation	Meaning
1000 - 1023	---	Reserved for Siemens
1024 - upper limit	---	Free for user assignment

* The actual upper limit of the block number depends on the PLC CPU contained in the selected NCU.

18.1.1.3 Function blocks (FCs)

Table 18-3 Assignment of the function blocks (FCs)

Number	Designation	Meaning
0 - 29	---	Reserved for Siemens
2	GP_HP	Basic program cyclic part
3	GP_PRAL	Basic program alarm-controlled part
5	GP_DIAG	Basic program diagnostic alarm and module failure
6	TM_TRANS2	Transfer block for tool management and multitool
7	TM_REV	Transfer block for tool change with turret
8	TM_TRANS	Transfer block for tool management
9	ASUB	Asynchronous subprograms
10	AL_MSG	Alarms/messages
12	AUXFU	Call interface for user auxiliary functions
13	BHG_DISP	Display control for the handheld unit
17	YDelta	Star-delta changeover
18	SpinCtrl	Spindle control for PLC
19	MCP_IFM	Distribution of MCP and operating software signals at the interface (milling machine)
21	Transfer	PLC-NC data exchange
22	TM_DIR	Direction selection for tool management
24	MCP_IFM2	Transferring MCP signals to the NC/PLC interface
25	MCP_IFT	Transferring MCP/BT signals to the NC/PLC interface
26	HPU_MCP	Transmission of the HT 8 signals to the interface
30 - 999*	---	Free for user assignment
1005	AG_SEND	Transferring data to the Ethernet CP
1006	AG_RECV	Receiving data from the Ethernet CP
1000 - 1023	---	Reserved for Siemens
1024 - upper limit	---	Free for user assignment

* The actual upper limit of the block number depends on the PLC CPU contained in the selected NCU.

18.1.1.4 Data blocks (DBs)

Note

The number of DBs to be set up depends on the parameterization made in the NC machine data.

Note

Data blocks of channels, axes/spindles and tool management that have not been activated are available for the user.

Table 18-4 Overview of the data blocks (DBs)

DB No.	Designation	Interface for
1	---	Reserved for Siemens
2 - 5	PLC-MELD	PLC messages
6 - 8	---	Basic program
9	NC-COMPILE	NC compile cycles
10	NC INTERFACE	Central NC
11	Mode group	Mode group
12	---	Computer link and transport system
13	---	Reserved for Hymnos
14	---	Reserved for basic program
15	---	Basic program
16	---	PI Service definitions
17	---	Version identifier
18	---	Reserved for basic program (SPL interface (Safety Integrated))
19	---	Operating software
20	---	PLC machine data
21 - 30	CHANNEL 1 ... CHANNEL 10	NC channels
31 - 61	AXIS 1 ... AXIS 31	Axes/spindle
62 - 70	---	Free for user assignment
71 - 74	---	User tool management
75 - 76	---	M group decoding
77	---	MCP, HHU signals (for SDB210)
78 - 80	---	Reserved for Siemens
81 - 127	---	Free for user assignment
1000	---	Ctrl-Energy
1001	---	SENTRON PAC
1002 … 1070	---	Reserved for Siemens
1071	---	Multitool: Loading/unloading magazine
1072	---	Multitool: Spindle

DB No.	Designation	Interface for
1073	---	Multitool: Revolver
1074 - 1099	---	Reserved for Siemens

18.1.1.5 Timer block

Table 18-5 Assigned times

Timer No.	Meaning
0 - 512*	Free for user assignment

* The actual upper limit of the timer number (DB) depends on the PLC CPU contained in the selected NCU.

18.1.2 Signals from/to the machine control panel

18.1.2.1 M version, signals from the MCP: Input image

Table 18-6 M version, signals from the MCP: Input image

	Signals from the MCP (keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Spindle override				Operating mode			
	D	C	B	A	JOG	TEACH IN	MDI	AUTO
EB n + 1	Machine function							
	REPOS	REF	INCvar	INC10000	INC1000	INC100	INC10	INC1
EB n + 2	Key-operated switch position 0	Key-operated switch position 2	Spindle start	*Spindle stop	Feedrate start	*Feedrate stop	NC start	*NC stop
EB n + 3		Key-operated switch position 1		Feedrate override				
	Reset		Single block	E	D	C	B	A
EB n + 4	Direction keys			Key-operated switch position 3	Axis selection			
	+R15	-R13	Rapid traverse R14		X R1	4th axis R4	7th axis R7	R10
EB n + 5	Axis selection							
	Y R2	Z R3	5th axis R5	Travel command MCS/WCS	R11	9th axis R9	8th axis R8	6th axis R6
EB n + 6	Freely assignable customer keys							
	T9	T10	T11	T12	T13	T14	T15	
EB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

18.1.2.2 M version, signals to the MCP: Output image

Table 18-7 M version, signals to the MCP: Output image

	Signals to the MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AB n + 0	Machine function				Operating mode			
	INC1000	INC100	INC10	INC1	JOG	TEACH IN	MDI	AUTO
AB n + 1	Feedrate start	*Feedrate stop	NC start	*NC stop	Machine function			
					REPOS	REF	INCvar	INC10000
AB n + 2	Direction key -R13	Axis selection				Single block	Spindle start	*Spindle stop
		X R1	4th axis R4	7th axis R7	R10			
AB n + 3	Axis selection							Direction key +R15
	Z R3	5th axis R5	Travel command MCS/WCS R12	R11	9th axis R9	8th axis R8	6th axis R6	
AB n + 4	Freely assignable customer keys							Y R2
	T9	T10	T11	T12	T13	T14	T15	
AB n + 5	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

18.1.2.3 T version, signals from the MCP: Input image

Table 18-8 T version, signals from the MCP: Input image

	Signals from the MCP (keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Spindle override				Operating mode			
	D	C	B	A	JOG	TEACH IN	MDI	AUTO
EB n + 1	Machine function							
	REPOS	REF	INCvar	INC10000	INC1000	INC100	INC10	INC1
EB n + 2	Key-operated switch position 0	Key-operated switch position 2	Spindle start	*Spindle stop	Feedrate start	*Feedrate stop	NC start	*NC stop
EB n + 3		Key-operated switch position 1		Feedrate override				
	Reset		Single block	E	D	C	B	A
EB n + 4	Direction keys			Key-operated switch position 3	Direction keys			
	R15	R13	R14		+Y R1	-Z R4	-C R7	R10

	Signals from the MCP (keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 5	Direction keys							
	+X R2	+C R3	Rapid traverse override R5	Travel command MCS/WCS R12	R11	-Y R9	-X R8	+Z R6
EB n + 6	Freely assignable customer keys							
	T9	T10	T11	T12	T13	T14	T15	
EB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

18.1.2.4 T version, signals to the MCP: Output image

Table 18-9 T version, signals to the MCP: Output image

	Signals to the MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AB n + 0	Machine function				Operating mode			
	INC1000	INC100	INC10	INC1	JOG	TEACH IN	MDI	AUTO
AB n + 1	Feedrate start	*Feedrate stop	NC start	*NC stop	Machine function			
					REPOS	REF	INCvar	INC10000
AB n + 2	Direction keys				Single block	Spindle start	*Spindle stop	
	R13	+Y R1	-Z R4	-C R7	R10			
AB n + 3	Direction keys							
	R3	R5	Travel command MCS/WCS	R11	-Y R9	-X R8	+Z R6	R15
AB n + 4	Freely assignable customer keys							Direction key +X R2
	T9	T10	T11	T12	T13	T14	T15	
AB n + 5	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

18.1.2.5 Slimline version, signals from the MCP: Input image

Table 18-10 Slimline version, signals from the MCP: Input image

	Signals from the slimline MCP (switches and keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Spindle override				Operating mode			
	*NC stop	SP -	SP 100%	SP +	SINGLEB	JOG	MDI	AUTO
EB n + 1	Spindle				Key-operated switch	Machine function		
	NC start	SP right	*SP stop	SP left	INT 3	REF	REPOS	TEACH IN

	Signals from the slimline MCP (switches and keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 2	Feedrate			Key-operated switch	Machine functions			
	Start	*Stop	INCvar	INT 0	INC1000	INC100	INC10	INC1
EB n + 3		Key-operated switch		Feedrate override				
	Reset	INT 2	INT 1	E	D	C	B	A
EB n + 4	Direction keys			Optional customer keys				
	+R15	-R13	Rapid traverse R14	KT4	KT3	KT2	KT1	KT0
EB n + 5			Axis selection					
	T17	KT5	6	5	4	Z	Y	X
EB n + 6	Freely assignable customer keys				MCS/WCS	Freely assignable customer keys		
	T9	T10	T11	T12		T14	T15	T16
EB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

18.1.2.6 Slimline version, signals to the MCP: Output image

Table 18-11 Slimline version, signals to the MCP: Output image

	Signals to the slimline MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Spindle override				Operating mode			
	NC stop	SP -	SP 100%	SP +	SINGLEB	JOG	MDI	AUTO
EB n + 1	Spindle					Machine function		
	NC start	SP right	SP stop	SP left	Not assigned	REF	REPOS	TEACH IN
EB n + 2	Feedrate				Machine functions			
	Start	Stop	INCvar	Not assigned	INC1000	INC100	INC10	INC1
EB n + 3	Not assigned							
EB n + 4	Direction keys			Optional customer keys				
	+R15	-R13	Rapid traverse R14	KT4	KT3	KT2	KT1	KT0
EB n + 5			Axis selection					
	T17	KT5	6	5	4	Z	Y	X
EB n + 6	Freely assignable customer keys					Freely assignable customer keys		
	T9	T10	T11	T12	MCS/WCS	T14	T15	T16
EB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

18.1.3 Signals from/to the handheld unit HT 2

18.1.3.1 Signals from the handheld unit: Input image

Table 18-12 Signals from the handheld unit: Input image

	Signals from the handheld unit (keys) (HHU / HT 2 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB m + 0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
IB m + 1	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
IB m + 2	Feedrate start	Free T2 key	AUTO	NC stop	Spindle stop	Feed stop	Free T1 key	JOG
IB m + 3	Free T3 key	Handwheel	4th axis	Z	Y	X	NC start	Spindle start
IB m + 4	Direction key	Rapid traverse override	Direction key +	Free T4 key				
IB m + 5	Acknowledgement Digital display	Key-operated switch	Rapid traverse/feedrate override switch					
			E	D	C	B	A	

18.1.3.2 Signals to the handheld unit: Output image

Table 18-13 Signals to the handheld unit: Output image

	Signals to the handheld unit (LED) (PLC → HHU / HT 2)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB m + 0	Always 1				Free T4 key	Free T3 key	Free T2 key	Free T1 key
QB m + 1	New data for selected line						Selecting the line	
							3 and 4	1 and 2
QB m + 2	Feedrate start	Rapid traverse override	AUTO	NC stop	Spindle stop	Feed stop	Direction key +	JOG
QB m + 3	Direction key +	Handwheel	4th axis	Z	Y	X	NC start	Spindle start
	Digital display of the handheld unit							
QB m + 4	Default setting of 1st character (right) of selected line							
QB m + 5	Default setting of 2nd character of selected line							
QB m + 6	Default setting of 3rd character of selected line							
QB m + 7	Default setting of 4th character of selected line							

	Signals to the handheld unit (LED) (PLC → HHU / HT 2)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB m + 8	Default setting of 5th character of selected line							
QB m + 9	Default setting of 6th character of selected line							
QB m + 10	Default setting of 7th character of selected line							
QB m + 11	Default setting of 8th character of selected line							
QB m + 12	Default setting of 9th character of selected line							
QB m + 13	Default setting of 10th character of selected line							
QB m + 14	Default setting of 11th character of selected line							
QB m + 15	Default setting of 12th character of selected line							
QB m + 16	Default setting of 13th character of selected line							
QB m + 17	Default setting of 14th character of selected line							
QB m + 18	Default setting of 15th character of selected line							
QB m + 19	Default setting of 16th character (left) of selected line							

Note

The parameterization or configuration of the various MCP/HHU versions is described in:

Further information

- Manual, Operator Components and Networking

18.1.4 Signals from/to the handheld unit HT 8

18.1.4.1 Signals from the handheld unit HT 8: Input image

Table 18-14 Signals from the handheld unit HT 8: Input image

	Signals from the MCP simulation (HT 8 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	QUIT	Reset	Work/ Machine
EB n + 1	Function key block							
	CPF (U key)	U4	U3	BigFct	U2	U1	INC	REPOS
EB n + 2		Change-over axes (HMI Advanced only)	Traversing keys (JOG) positive direction					
		Ax7-Ax12 instead of Ax1-Ax6	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
EB n + 3	Traversing keys (JOG) negative direction							
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
EB n + 4								
	U9	U10	U11	U12	U13	U14	U15	U16
EB n + 5								
		U8	U7	U6	U5	SBL		
EB n + 6			Start key block					
	Reserved	HT 8	SF2	SF1	SF4	SF3	Start	Stop
EB n + 7	Feedrate override							
				E	D	C	B	A

18.1.4.2 Signals to handheld terminal HT 8: Output image

Table 18-15 Signals from the handheld unit HT 8: Output image

	Signals to the MCP simulation (PLC → HT 8)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	QUIT	Reset	Work/ Machine
AB n + 1	Function key block							
		U4	U3		U2	U1	INC	REPOS

	Signals to the MCP simulation (PLC → HT 8)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AB n + 2		Axes 7 - n selected	Traversing keys (JOG) positive direction					
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
AB n + 3	Traversing keys (JOG) negative direction							
	For WCS: no machine axes		Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
AB n + 4								
	U9	U10	U11	U12	U13	U14	U15	U16
AB n + 5								
		U8	U7	U6	U5	SBL		
AB n + 6	Start key block							
	Display traversing keys		SF2	SF1	SF4	SF3	Start	Stop
AB n + 7								

18.1.5 Signals from/to the handheld unit HT 10

18.1.5.1 Signals from the handheld unit HT 10: Input image

Table 18-16 Signals from the handheld unit HT 10: Input image

	Signals from the MCP simulation (HT 10 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	QUIT	Reset	Work/ Machine
EB n + 1	Function key block							
	CPF (U key)	U4	U3	BigFct	U2	U1	INC	REPOS
EB n + 2		Change-over axes (HMI Advanced only)	Traversing keys (JOG) HT 8, positive direction					
		Ax7-Ax12 instead of Ax1-Ax6	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
EB n + 3	JOG keys HT 10		Traversing keys (JOG) HT 8, negative direction					
	Plus	Minus	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1

	Signals from the MCP simulation (HT 10 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 4								
	U16	U15	U14	U13	U12	U11	U10	U9
EB n + 5								
		U8	U7	U6	U5	SBL		Rapid traverse
EB n + 6			Start key block					
	HT 10	HT 8	SF2	SF1	SF4	SF3	Start	Stop
EB n + 7	Feedrate override							
				E	D	C	B	A

18.1.5.2 Signals to handheld terminal HT 10: Output image

Table 18-17 Signals from the handheld unit HT 10: Output image

	Signals to the MCP simulation (PLC → HT 10)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	QUIT	Reset	Work/ Machine
AB n + 1	Function key block							
		U4	U3		U2	U1	INC	REPOS
AB n + 2	JOG + (HT 10)	Axes 7 - n selected	Traversing keys (JOG) HT 8, positive direction					
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
AB n + 3	Traversing keys (JOG) HT 8, negative direction							
	For WCS: no machine axes	JOG - (HT 10)	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
AB n + 4								
	U9	U10	U11	U12	U13	U14	U15	U16
AB n + 5								
		U8	U7	U6	U5	SBL		Rapid traverse
AB n + 6	Start key block							
	Display traversing keys		SF2	SF1	SF4	SF3	Start	Stop
AB n + 7								

18.1.6 PLC alarms/messages

18.1.6.1 FC 10 alarms in the DB2 (FB1: "ExtendAlMsg" = FALSE)

Message type

- **FM**: A fault message with the associated event number as fault number is triggered by the signal.
- **OM**: An operating message with the associated event number as message number is triggered by the signal.

Further information

A detailed description regarding error and system status messages can be obtained in:
FC10: AL_MSG - error and operating messages (Page 247)

Table 18-18 DB2, channel range 1

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 1							
	Feedrate disable (event no.: 510000-510015)							
0 (FM)	510007	510006	510005	510004	510003	510002	510001	510000
1 (OM)	510015	510014	510013	510012	510011	510010	510009	510008
2 (FM)	Feedrate and read-in disable, byte 1 (event no.: 510100-510107)							
3 (FM)	Feedrate and read-in disable, byte 2 (event no.: 510108-510115)							
4 (OM)	Feedrate and read-in disable, byte 3 (event no.: 510116-510123)							
5 (OM)	Feedrate and read-in disable, byte 4 (event no.: 510124-510131)							
6 (FM)	Read-in disable, byte 1 (event no.: 510200-510207)							
7 (FM)	Read-in disable, byte 2 (event no.: 510208-510215)							
8 (OM)	Read-in disable, byte 3 (event no.: 510216-510223)							
9 (OM)	Read-in disable, byte 4 (event no.: 510224-510231)							
10 (FM)	NC start disable, byte 1 (event no.: 510300-510307)							
11 (OM)	NC start disable, byte 2 (event no.: 510308-510315)							
12 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 511100-511107)							
13 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 511108-511115)							
14 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 511200-511207)							
15 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 511208-511215)							
16 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 511300-511307)							
17 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 511308-511315)							

Table 18-19 DB2, channel range 2

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 2							
	Feedrate disable (event no.: 520000-520015)							
18 (FM)	520007	520006	520005	520004	520003	520002	520001	520000
19 (OM)	520015	520014	520013	520012	520011	520010	520009	520008
20 (FM)	Feedrate and read-in disable, byte 1 (event no.: 520100-520107)							
21 (FM)	Feedrate and read-in disable, byte 2 (event no.: 520108-520115)							
22 (OM)	Feedrate and read-in disable, byte 3 (event no.: 520116-520123)							
23 (OM)	Feedrate and read-in disable, byte 4 (event no.: 520124-520131)							
24 (FM)	Read-in disable, byte 1 (event no.: 520200-520207)							
25 (FM)	Read-in disable, byte 2 (event no.: 520208-520215)							
26 (OM)	Read-in disable, byte 3 (event no.: 520216-520223)							
27 (OM)	Read-in disable, byte 4 (event no.: 520224-520231)							
28 (FM)	NC start disable, byte 1 (event no.: 520300-520307)							
29 (OM)	NC start disable, byte 2 (event no.: 520308-520315)							
30 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 521100-521107)							
31 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 521108-521115)							
32 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 521200-521207)							
33 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 521208-521215)							
34 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 521300-521307)							
35 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 521308-521315)							

Table 18-20 DB2, channel range 3

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 3							
	Feedrate disable (event no.: 530000-530015)							
36 (FM)	530007	530006	530005	530004	530003	530002	530001	530000
37 (OM)	530015	530014	530013	530012	530011	530010	530009	530008
38 (FM)	Feedrate and read-in disable, byte 1 (event no.: 530100-530107)							
39 (FM)	Feedrate and read-in disable, byte 2 (event no.: 530108-530115)							
40 (OM)	Feedrate and read-in disable, byte 3 (event no.: 530116-530123)							
41 (OM)	Feedrate and read-in disable, byte 4 (event no.: 530124-530131)							
42 (FM)	Read-in disable, byte 1 (event no.: 530200-530207)							
43 (FM)	Read-in disable, byte 2 (event no.: 530208-530215)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
44 (OM)	Read-in disable, byte 3 (event no.: 530216-530223)							
45 (OM)	Read-in disable, byte 4 (event no.: 530224-530231)							
46 (FM)	NC start disable, byte 1 (event no.: 530300-530307)							
47 (OM)	NC start disable, byte 2 (event no.: 530308-530315)							
48 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 531100-531107)							
49 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 531108-531115)							
50 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 531200-531207)							
51 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 531208-531215)							
52 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 531300-531307)							
53 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 531308-531315)							

Table 18-21 DB2, channel range 4

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 4							
	Feedrate disable (event no.: 540000-540015)							
54 (FM)	540007	540006	540005	540004	540003	540002	540001	540000
55 (OM)	540015	540014	540013	540012	540011	540010	540009	540008
56 (FM)	Feedrate and read-in disable, byte 1 (event no.: 540100-540107)							
57 (FM)	Feedrate and read-in disable, byte 2 (event no.: 540108-540115)							
58 (OM)	Feedrate and read-in disable, byte 3 (event no.: 540116-540123)							
59 (OM)	Feedrate and read-in disable, byte 4 (event no.: 540124-540131)							
60 (FM)	Read-in disable, byte 1 (event no.: 540200-540207)							
61 (FM)	Read-in disable, byte 2 (event no.: 540208-540215)							
62 (OM)	Read-in disable, byte 3 (event no.: 540216-540223)							
63 (OM)	Read-in disable, byte 4 (event no.: 540224-540231)							
64 (FM)	NC start disable, byte 1 (event no.: 540300-540307)							
65 (FM)	NC start disable, byte 2 (event no.: 540308-540315)							
66 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 541100-541107)							
67 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 541108-541115)							
68 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 541200-541207)							
69 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 541208-541215)							
70 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 541300-541307)							
71 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 541308-541315)							

Table 18-22 DB2, channel range 5

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 5							
	Feedrate disable (event no.: 550000-550015)							
72 (FM)	550007	550006	550005	550004	550003	550002	550001	550000
73 (OM)	550015	550014	550013	550012	550011	550010	550009	550008
74 (FM)	Feedrate and read-in disable, byte 1 (event no.: 550100-550107)							
75 (OM)	Feedrate and read-in disable, byte 2 (event no.: 550108-550115)							
76 (OM)	Feedrate and read-in disable, byte 3 (event no.: 550116-550123)							
77 (OM)	Feedrate and read-in disable, byte 4 (event no.: 550124-550131)							
78 (FM)	Read-in disable, byte 1 (event no.: 550200-550207)							
79 (FM)	Read-in disable, byte 2 (event no.: 550208-550315)							
80 (OM)	Read-in disable, byte 3 (event no.: 550216-550223)							
81 (OM)	Read-in disable, byte 4 (event no.: 550224-550231)							
82 (FM)	NC start disable, byte 1 (event no.: 550300-550307)							
83 (OM)	NC start disable, byte 2 (event no.: 550308-550315)							
84 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 551100-551107)							
85 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 551108-551115)							
86 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 551200-551207)							
87 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 551208-551215)							
88 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 551300-551307)							
89 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 551308-551315)							

Table 18-23 DB2, channel range 6

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 6							
	Feedrate disable (event no.: 560000-560015)							
90 (FM)	560007	560006	560005	560004	560003	560002	560001	560000
91 (OM)	560015	560014	560013	560012	560011	560010	560009	560008
92 (FM)	Feedrate and read-in disable, byte 1 (event no.: 560100-560107)							
93 (FM)	Feedrate and read-in disable, byte 2 (event no.: 560108-560115)							
94 (OM)	Feedrate and read-in disable, byte 3 (event no.: 560116-560123)							
95 (OM)	Feedrate and read-in disable, byte 4 (event no.: 560124-560131)							
96 (FM)	Read-in disable, byte 1 (event no.: 560200-560207)							
97 (FM)	Read-in disable, byte 2 (event no.: 560208-560315)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
98 (OM)	Read-in disable, byte 3 (event no.: 560216-560223)							
99 (OM)	Read-in disable, byte 4 (event no.: 560224-560231)							
100 (FM)	NC start disable, byte 1 (event no.: 560300-560307)							
101 (OM)	NC start disable, byte 2 (event no.: 560308-560315)							
102 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 561100-561107)							
103 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 561108-561115)							
104 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 561200-561207)							
105 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 561208-561215)							
106 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 561300-561307)							
107 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 561308-561315)							

Table 18-24 DB2, channel range 7

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 7							
	Feedrate disable (event no.: 570000-570015)							
108 (FM)	570007	570006	570005	570004	570003	570002	570001	570000
109 (OM)	570015	570014	570013	570012	570011	570010	570009	570008
110 (FM)	Feedrate and read-in disable, byte 1 (event no.: 570100-570107)							
111 (FM)	Feedrate and read-in disable, byte 2 (event no.: 570108-570115)							
112 (OM)	Feedrate and read-in disable, byte 3 (event no.: 570116-570123)							
113 (OM)	Feedrate and read-in disable, byte 4 (event no.: 570124-570131)							
114 (FM)	Read-in disable, byte 1 (event no.: 570200-570207)							
115 (FM)	Read-in disable, byte 2 (event no.: 570208-570315)							
116 (OM)	Read-in disable, byte 3 (event no.: 570216-570223)							
117 (OM)	Read-in disable, byte 4 (event no.: 570224-570231)							
118 (FM)	NC start disable, byte 1 (event no.: 570300-570307)							
119 (OM)	NC start disable, byte 2 (event no.: 570308-570315)							
120 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 571100-571107)							
121 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 571108-571115)							
122 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 571200-571207)							
123 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 571208-571215)							
124 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 571300-571307)							
125 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 571308-571315)							

Table 18-25 DB2, channel range 8

DB2	**Signals for PLC events (PLC → HMI)** **FB1 parameter "ExtendAlMsg" = FALSE**							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Channel 8							
	Feedrate disable (event no.: 580000-580015)							
126 (FM)	580007	580006	580005	580004	580003	580002	580001	580000
127 (OM)	580015	580014	580013	580012	580011	580010	580009	580008
128 (FM)	Feedrate and read-in disable, byte 1 (event no.: 580100-580107)							
129 (FM)	Feedrate and read-in disable, byte 2 (event no.: 580108-580115)							
130 (OM)	Feedrate and read-in disable, byte 3 (event no.: 580116-580123)							
131 (OM)	Feedrate and read-in disable, byte 4 (event no.: 580124-580131)							
132 (FM)	Read-in disable, byte 1 (event no.: 580200-580207)							
133 (FM)	Read-in disable, byte 2 (event no.: 580208-580315)							
134 (OM)	Read-in disable, byte 3 (event no.: 580216-580223)							
135 (OM)	Read-in disable, byte 4 (event no.: 580224-580231)							
136 (FM)	NC start disable, byte 1 (event no.: 580300-580307)							
137 (OM)	NC start disable, byte 2 (event no.: 580308-580315)							
138 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 581100-581107)							
139 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 581108-581115)							
140 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 581200-581207)							
141 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 581208-581215)							
142 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 581300-581307)							
143 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 581308-581315)							
	Channels 9 and **10** not implemented							

Table 18-26 DB2, axis ranges

DB2	**Signals for PLC events (PLC → HMI)** **FB1 parameter "ExtendAlMsg" = FALSE**							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Axis/spindle							
	Feedrate stop / spindle stop for axis/spindle 1 (event no.: 600100-600115)							
144 (FM)	600107	600106	600105	600104	600103	600102	600101	600100
145 (OM)	600115	600114	600113	600112	600111	600110	600109	600108
146 (FM)	Feedrate stop / spindle stop for axis/spindle 2, byte 1 (event no.: 600200-600207)							
147 (OM)	Feedrate stop / spindle stop for axis/spindle 2, byte 2 (event no.: 600208-600215)							
148 (FM)	Feedrate stop / spindle stop for axis/spindle 3, byte 1 (event no.: 600300-600307)							
149 (OM)	Feedrate stop / spindle stop for axis/spindle 3, byte 2 (event no.: 600308-600315)							
150 (FM)	Feedrate stop / spindle stop for axis/spindle 4, byte 1 (event no.: 600400-600407)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
151 (OM)	Feedrate stop / spindle stop for axis/spindle 4, byte 2 (event no.: 600408-600415)							
152 (FM)	Feedrate stop / spindle stop for axis/spindle 5, byte 1 (event no.: 600500-600507)							
153 (OM)	Feedrate stop / spindle stop for axis/spindle 5, byte 2 (event no.: 600508-600515)							
154 (FM)	Feedrate stop / spindle stop for axis/spindle 6, byte 1 (event no.: 600600-600607)							
155 (OM)	Feedrate stop / spindle stop for axis/spindle 6, byte 2 (event no.: 600608-600615)							
156 (FM)	Feedrate stop / spindle stop for axis/spindle 7, byte 1 (event no.: 600700-600707)							
157 (OM)	Feedrate stop / spindle stop for axis/spindle 7, byte 2 (event no.: 600708-600715)							
158 (FM)	Feedrate stop / spindle stop for axis/spindle 8, byte 1 (event no.: 600800-600807)							
159 (OM)	Feedrate stop / spindle stop for axis/spindle 8, byte 2 (event no.: 600808-600815)							
160 (FM)	Feedrate stop / spindle stop for axis/spindle 9, byte 1 (event no.: 600900-600907)							
161 (OM)	Feedrate stop / spindle stop for axis/spindle 9, byte 2 (event no.: 600908-600915)							
162 (FM)	Feedrate stop/spindle stop for axis/spindle 10 byte 1 (event no.: 601000-601007)							
163 (OM)	Feedrate stop/spindle stop for axis/spindle 10 byte 2 (event no.: 601008-601015)							
164 (FM)	Feedrate stop/spindle stop for axis/spindle 11 byte 1 (event no.: 601100-601107)							
165 (OM)	Feedrate stop/spindle stop for axis/spindle 11 byte 2 (event no.: 601108-601115)							
166 (FM)	Feedrate stop/spindle stop for axis/spindle 12 byte 1 (event no.: 601200-601207)							
167 (OM)	Feedrate stop/spindle stop for axis/spindle 12 byte 2 (event no.: 601208-601215)							
168 (FM)	Feedrate stop/spindle stop for axis/spindle 13 byte 1 (event no.: 601300-601307)							
169 (OM)	Feedrate stop/spindle stop for axis/spindle 13 byte 2 (event no.: 601308-601315)							
170 (FM)	Feedrate stop/spindle stop for axis/spindle 14 byte 1 (event no.: 601400-601407)							
171 (OM)	Feedrate stop/spindle stop for axis/spindle 14 byte 2 (event no.: 601408-601415)							
172 (FM)	Feedrate stop/spindle stop for axis/spindle 15 byte 1 (event no.: 601500-601507)							
173 (OM)	Feedrate stop/spindle stop for axis/spindle 15 byte 2 (event no.: 601508-601515)							
174 (FM)	Feedrate stop/spindle stop for axis/spindle 16 byte 1 (event no.: 601600-601607)							
175 (OM)	Feedrate stop/spindle stop for axis/spindle 16 byte 2 (event no.: 601608-601615)							
176 (FM)	Feedrate stop/spindle stop for axis/spindle 17 byte 1 (event no.: 601700-601707)							
177 (OM)	Feedrate stop/spindle stop for axis/spindle 17 byte 2 (event no.: 601708-601715)							
178 (FM)	Feedrate stop/spindle stop for axis/spindle 18 byte 1 (event no.: 601800-601807)							
179 (OM)	Feedrate stop/spindle stop for axis/spindle 18 byte 2 (event no.: 601808-601815)							
	Axes 19 – 31 not realized							

Table 18-27 DB2, user ranges

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	User ranges							
	User range 0 (event no.: 700000-700015)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
180 (FM)	700007	700006	700005	700004	700003	700002	700001	700000
181 (FM)	700015	700014	700013	700012	700011	700010	700009	700008
182 (FM)	User range 0: Byte 3 (event no.: 700016-700023)							
183 (FM)	User range 0: Byte 4 (event no.: 700024-700031)							
184 (OM)	User range 0: Byte 5 (event no.: 700032-700039)							
185 (OM)	User range 0: Byte 6 (event no.: 700040-700047)							
186 (OM)	User range 0: Byte 7 (event no.: 700048-700055)							
187 (OM)	User range 0: Byte 8 (event no.: 700056-700063)							
188 - 191 (FM)	User range 1: Byte 1 - 4 (event no.: 700100-700131)							
192 - 195 (OM)	User range 1, bytes 5 - 8 (event no.: 700132-700163)							
196 - 199 (FM)	User range 2: Byte 1 - 4 (event no.: 700200-700231)							
200 - 203 (OM)	User range 2: Byte 5 - 8 (event no.: 700232-700263)							
204 - 207 (FM)	User range 3: Byte 1 - 4 (event no.: 700300-700331)							
208 - 211 (OM)	User range 3: Byte 5 - 8 (event no.: 700332-700363)							
212 - 215 (FM)	User range 4: Byte 1 - 4 (event no.: 700400-700431)							
216 - 219 (OM)	User range 4: Byte 5 - 8 (event no.: 700432-700463)							
220 - 223 (FM)	User range 5: Byte 1 - 4 (event no.: 700500-700531)							
224 - 227 (OM)	User range 5: Byte 5 - 8 (event no.: 700532-700563)							
228 - 231 (FM)	User range 6: Byte 1 - 4 (event no.: 700600-700631)							
232 - 235 (OM)	User range 6: Byte 5 - 8 (event no.: 700632-700663)							
236 - 239 (FM)	User range 7: Byte 1 - 4 (event no.: 700700-700731)							
240 - 243 (OM)	User range 7: Byte 5 - 8 (event no.: 700732-700763)							
244 - 247 (FM)	User range 8: Byte 1 - 4 (event no.: 700800-700831)							
248 - 251 (OM)	User range 8: Byte 5 - 8 (event no.: 700832-700863)							
252 - 255 (FM)	User range 9: Byte 1 - 4 (event no.: 700900-700931)							
256 - 259 (OM)	User range 9: Byte 5 - 8 (event no.: 700932-700963)							
260 - 263 (FM)	User range 10: Byte 1 - 4 (event no.: 701000-701031)							
264 - 267 (OM)	User range 10: Byte 5 - 8 (event no.: 701032-701063)							
268 - 271 (FM)	User range 11: Byte 1 - 4 (event no.: 701100-701131)							
272 - 275 (OM)	User range 11: Byte 5 - 8 (event no.: 701132-701163)							
276 - 279 (FM)	User range 12: Byte 1 - 4 (event no.: 701200-701231)							
280 - 283 (OM)	User range 12: Byte 5 - 8 (event no.: 701232-701263)							
284 - 287 (FM)	User range 13: Byte 1 - 4 (event no.: 701300-701331)							
288 - 291 (OM)	User range 13: Byte 5 - 8 (event no.: 701332-701363)							
292 - 295 (FM)	User range 14: Byte 1 - 4 (event no.: 701400-701431)							
296 - 299 (OM)	User range 14: Byte 5 - 8 (event no.: 701432-701463)							
300 - 303 (FM)	User range 15: Byte 1 - 4 (event no.: 701500-701531)							
304 - 307 (OM)	User range 15: Byte 5 - 8 (event no.: 701532-701563)							
308 - 311 (FM)	User range 16: Byte 1 - 4 (event no.: 701600-701631)							
312 - 315 (OM)	User range 16: Byte 5 - 8 (event no.: 701632-701663)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = FALSE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
316 - 319 (FM)	User range 17: Byte 1 - 4 (event no.: 701700-701731)							
320 - 323 (OM)	User range 17: Byte 5 - 8 (event no.: 701732-701763)							
324 - 327 (FM)	User range 18: Byte 1 - 4 (event no.: 701800-701831)							
328 - 331 (OM)	User range 18: Byte 5 - 8 (event no.: 701832-701863)							
332 - 335 (FM)	User range 19: Byte 1 - 4 (event no.: 701900-701931)							
336 - 339 (OM)	User range 19: Byte 5 - 8 (event no.: 701932-701963)							
340 - 343 (FM)	User range 20: Byte 1 - 4 (event no.: 702000-702031)							
344 - 347 (OM)	User range 20: Byte 5 - 8 (event no.: 702032-702063)							
348 - 351 (FM)	User range 21: Byte 1 - 4 (event no.: 702100-702131)							
352 - 355 (OM)	User range 21: Byte 5 - 8 (event no.: 702132-702163)							
356 - 359 (FM)	User range 22: Byte 1 - 4 (event no.: 702200-702231)							
360 - 363 (OM)	User range 22: Byte 5 - 8 (event no.: 702232-702263)							
364 - 367 (FM)	User range 23: Byte 1 - 4 (event no.: 702300-702331)							
368 - 371 (OM)	User range 23: Byte 5 - 8 (event no.: 702332-702363)							
372 - 375 (FM)	User range 24: Byte 1 - 4 (event no.: 702400-702431)							
376 - 379 (OM)	User range 24: Byte 5 - 8 (event no.: 702432-702463)							
380 - 383 (FM)	User range 25: Byte 1 - 4 (event no.: 702500-702531)							
384 - 387 (OM)	User range 25: Byte 5 - 8 (event no.: 702532-702563)							
388 - 391 (FM)	User range 26: Byte 1 - 4 (event no.: 702600-702631)							
392 - 395 (OM)	User range 26: Byte 5 - 8 (event no.: 702632-702663)							
396 - 399 (FM)	User range 27: Byte 1 - 4 (event no.: 702700-702731)							
400 - 403 (OM)	User range 27: Byte 5 - 8 (event no.: 702732-702763)							
404 - 407 (FM)	User range 28: Byte 1 - 4 (event no.: 702800-702831)							
408 - 411 (OM)	User range 28: Byte 5 - 8 (event no.: 702832-702863)							
412 - 415 (FM)	User range 29: Byte 1 - 4 (event no.: 702900-702931)							
416 - 419 (OM)	User range 29: Byte 5 - 8 (event no.: 702932-702963)							
420 - 423 (FM)	User range 30: Byte 1 - 4 (event no.: 703000-703031)							
424 - 427 (OM)	User range 30: Byte 5 - 8 (event no.: 703032-703063)							
428 - 431 (FM)	User range 31: Byte 1 - 4 (event no.: 703100-703131)							
432 - 435 (OM)	User range 31: Byte 5 - 8 (event no.: 703132-703163)							

Note

Parameterizing user ranges

The number of user ranges (maximum of 32) can be parameterized using FB 1 "MsgUser".

18.1.6.2 FC 10 alarms in the DB2 (FB1: "ExtendAlMsg" = TRUE)

Description

Message type

- **FM**: A fault message with the associated event number as fault number is triggered by the signal.
- **OM**: An operating message with the associated event number as message number is triggered by the signal.

Further information

FC10: AL_MSG - error and operating messages (Page 247)

Table 18-28 DB2, channel range 1

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Signals WITHOUT the display of a fault/operating message (DBB0 - 309)							
	Channel 1							
0	Feedrate disable							
1	Feedrate disable							
2	Read-in disable							
3	Read-in disable							
4	Start disable							
5	Start disable							
6	Feedrate stop, geo axis 1, byte 1							
7	Feedrate stop, geo axis 1, byte 2							
8	Feedrate stop, geo axis 2, byte 1							
9	Feedrate stop, geo axis 2, byte 2							
10	Feedrate stop, geo axis 3, byte 1							
11	Feedrate stop, geo axis 3, byte 2							
12 - 119	**Channel 2 - channel 10,** see above "Channel 1"							
	Axis/spindle 1							
120	Feedrate stop / spindle stop, byte 1							
121	Feedrate stop / spindle stop, byte 2							
122 - 181	**Axis / spindle 2 - 31,** see above "Axis /spindle 1"							
	Additional values for user range 0							
182	Additional value for event number 700000							
184	Additional value for event number 700001							
...	...							
308	Additional value for event number 700063							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Signals WITH the display of a fault/operating message (from DBB 310)							
	Channel 1							
	Feedrate disable (event no.: 510000-510015)							
310 (FM)	510007	510006	510005	510004	510003	510002	510001	510000
311 (OM)	510015	510014	510013	510012	510011	510010	510009	510008
312 (FM)	Feedrate and read-in disable: Byte 1 (event no.: 510100-510107)							
313 (FM)	Feedrate and read-in disable: Byte 2 (event no.: 510108-510115)							
314 (OM)	Feedrate and read-in disable: Byte 3 (event no.: 510116-510123)							
315 (OM)	Feedrate and read-in disable: Byte 4 (event no.: 510124-510131)							
316 (FM)	Read-in disable: Byte 1 (event no.: 510200-510207)							
317 (FM)	Read-in disable: Byte 2 (event no.: 510208-510215)							
318 (OM)	Read-in disable: Byte 3 (event no.: 510216-510223)							
319 (OM)	Read-in disable: Byte 4 (event no.: 510224-510231)							
320 (FM)	NC start disable: Byte 1 (event no.: 510300-510307)							
321 (OM)	NC start disable: Byte 2 (event no.: 510308-510315)							
322 (FM)	Feedrate stop, geo axis 1: Byte 1 (event no.: 511100-511107)							
323 (OM)	Feedrate stop, geo axis 1: Byte 2 (event no.: 511108-511115)							
324 (FM)	Feedrate stop, geo axis 2: Byte 1 (event no.: 511200-511207)							
325 (OM)	Feedrate stop, geo axis 2: Byte 2 (event no.: 511208-511215)							
326 (FM)	Feedrate stop, geo axis 3: Byte 1 (event no.: 511300-511307)							
327 (OM)	Feedrate stop, geo axis 3: Byte 2 (event no.: 511308-511315)							

Table 18-29 DB2, channel range 2

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 2							
	Feedrate disable (event no.: 510000-520015)							
328 (FM)	520007	520006	520005	520004	520003	520002	520001	520000
329 (OM)	520015	520014	520013	520012	520011	520010	520009	520008
330 (FM)	Feedrate and read-in disable, byte 1 (event no.: 520100-520107)							
331 (FM)	Feedrate and read-in disable, byte 2 (event no.: 520108-520115)							
332 (OM)	Feedrate and read-in disable, byte 3 (event no.: 520116-520123)							
333 (OM)	Feedrate and read-in disable, byte 4 (event no.: 520124-520131)							
334 (FM)	Read-in disable, byte 1 (event no.: 520200-520207)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
335 (FM)	Read-in disable, byte 2 (event no.: 520208-520215)							
336 (OM)	Read-in disable, byte 3 (event no.: 520216-520223)							
337 (OM)	Read-in disable, byte 4 (event no.: 520224-520231)							
338 (FM)	NC start disable, byte 1 (event no.: 520300-520307)							
339 (OM)	NC start disable, byte 2 (event no.: 520308-520315)							
340 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 521100-521107)							
341 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 521108-521115)							
342 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 521200-521207)							
343 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 521208-521215)							
344 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 521300-521307)							
345 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 521308-521315)							

Table 18-30 DB2, channel range 3

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 3							
	Feedrate disable (event no.: 530000-530015)							
346 (FM)	530007	530006	530005	530004	530003	530002	530001	530000
347 (OM)	530015	530014	530013	530012	530011	530010	530009	530008
348 (FM)	Feedrate and read-in disable, byte 1 (event no.: 530100-530107)							
349 (FM)	Feedrate and read-in disable, byte 2 (event no.: 530108-530115)							
350 (OM)	Feedrate and read-in disable, byte 2 (event no.: 530108-530115)							
351 (OM)	Feedrate and read-in disable, byte 4 (event no.: 530124-530131)							
352 (FM)	Read-in disable, byte 1 (event no.: 530200-530207)							
353 (FM)	Read-in disable, byte 2 (event no.: 530208-530215)							
354 (OM)	Read-in disable, byte 3 (event no.: 530216-530223)							
355 (OM)	Read-in disable, byte 4 (event no.: 530224-530231)							
356 (FM)	NC start disable, byte 1 (event no.: 530300-530307)							
357 (OM)	NC start disable, byte 2 (event no.: 530308-530315)							
358 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 531100-531107)							
359 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 531108-531115)							
360 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 531200-531207)							
361 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 531208-531215)							
362 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 531300-531307)							
363 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 531308-531315)							

Table 18-31 DB2, channel range 4

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 4							
	Feedrate disable (event no.: 540000-540015)							
364 (FM)	540007	540006	540005	540004	540003	540002	540001	540000
365 (OM)	540015	540014	540013	540012	540011	540010	540009	540008
366 (FM)	Feedrate and read-in disable, byte 1 (event no.: 540100-540107)							
367 (FM)	Feedrate and read-in disable, byte 2 (event no.: 540108-540115)							
368 (OM)	Feedrate and read-in disable, byte 3 (event no.: 540116-540123)							
369 (OM)	Feedrate and read-in disable, byte 4 (event no.: 540124-540131)							
370 (FM)	Read-in disable, byte 1 (event no.: 540200-540207)							
371 (FM)	Read-in disable, byte 2 (event no.: 540208-540215)							
372 (OM)	Read-in disable, byte 3 (event no.: 540216-540223)							
373 (OM)	Read-in disable, byte 4 (event no.: 540224-540231)							
374 (FM)	NC start disable, byte 1 (event no.: 540300-540307)							
375 (OM)	NC start disable, byte 2 (event no.: 540308-540315)							
376 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 541100-541107)							
377 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 541108-541115)							
378 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 541200-541207)							
379 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 541208-541215)							
380 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 541300-541307)							
381 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 541308-541315)							

Table 18-32 DB2, channel range 5

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 5							
	Feedrate disable (event no.: 550000-550015)							
382 (FM)	550007	550006	550005	550004	550003	550002	550001	550000
383 (OM)	550015	550014	550013	550012	550011	550010	550009	550008
384 (FM)	Feedrate and read-in disable, byte 1 (event no.: 550100-550107)							
385 (FM)	Feedrate and read-in disable, byte 2 (event no.: 550108-550115)							
386 (OM)	Feedrate and read-in disable, byte 3 (event no.: 550116-550123)							
387 (OM)	Feedrate and read-in disable, byte 4 (event no.: 550124-550131)							
388 (FM)	Read-in disable, byte 1 (event no.: 550200-550207)							
389 (FM)	Read-in disable, byte 2 (event no.: 550208-550215)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
390 (OM)	Read-in disable, byte 3 (event no.: 550216-550223)							
391 (OM)	Read-in disable, byte 4 (event no.: 550224-550231)							
392 (FM)	NC start disable, byte 1 (event no.: 550300-550307)							
393 (OM)	NC start disable, byte 2 (event no.: 550308-550315)							
394 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 551100-551107)							
395 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 551108-551115)							
396 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 551200-551207)							
397 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 551208-551215)							
398 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 551300-551307)							
399 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 551308-551315)							

Table 18-33 DB2, channel range 6

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 6							
	Feedrate disable (event no.: 560000-560015)							
400 (FM)	560007	560006	560005	560004	560003	560002	560001	560000
401 (OM)	560015	560014	560013	560012	560011	560010	560009	560008
402 (FM)	Feedrate and read-in disable, byte 1 (event no.: 560100-560107)							
403 (FM)	Feedrate and read-in disable, byte 2 (event no.: 560108-560115)							
404 (OM)	Feedrate and read-in disable, byte 3 (event no.: 560116-560123)							
405 (OM)	Feedrate and read-in disable, byte 4 (event no.: 560124-560131)							
406 (FM)	Read-in disable, byte 1 (event no.: 560200-560207)							
407 (FM)	Read-in disable, byte 2 (event no.: 560208-560215)							
408 (OM)	Read-in disable, byte 3 (event no.: 560216-560223)							
409 (OM)	Read-in disable, byte 4 (event no.: 560224-560231)							
410 (FM)	NC start disable, byte 1 (event no.: 560300-560307)							
411 (OM)	NC start disable, byte 2 (event no.: 560308-560315)							
412 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 561100-561107)							
413 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 561108-561115)							
414 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 561200-561207)							
415 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 561208-561215)							
416 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 561300-561307)							
417 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 561308-561315)							

Table 18-34 DB2, channel range 7

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 7							
	Feedrate disable (event no.: 570000-570015)							
418 (FM)	570007	570006	570005	570004	570003	570002	570001	570000
419 (OM)	570015	570014	570013	570012	570011	570010	570009	570008
420 (FM)	Feedrate and read-in disable, byte 1 (event no.: 570100-570107)							
421 (FM)	Feedrate and read-in disable, byte 2 (event no.: 570108-570115)							
422 (OM)	Feedrate and read-in disable, byte 3 (event no.: 570116-570123)							
423 (OM)	Feedrate and read-in disable, byte 4 (event no.: 570124-570131)							
424 (FM)	Read-in disable, byte 1 (event no.: 570200-570207)							
425 (FM)	Read-in disable, byte 2 (event no.: 570208-570215)							
426 (OM)	Read-in disable, byte 3 (event no.: 570216-570223)							
427 (OM)	Read-in disable, byte 4 (event no.: 570224-570231)							
428 (FM)	NC start disable, byte 1 (event no.: 570300-570307)							
429 (OM)	NC start disable, byte 2 (event no.: 570308-570315)							
430 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 571100-571107)							
431 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 571108-571115)							
432 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 571200-571207)							
433 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 571208-571215)							
434 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 571300-571307)							
435 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 571308-571315)							

Table 18-35 DB2, channel range 8

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 8							
	Feedrate disable (event no.: 580000-580015)							
436 (FM)	580007	580006	580005	580004	580003	580002	580001	580000
437 (OM)	580015	580014	580013	580012	580011	580010	580009	580008
438 (FM)	Feedrate and read-in disable, byte 1 (event no.: 580100-580107)							
439 (FM)	Feedrate and read-in disable, byte 2 (event no.: 580108-580115)							
440 (OM)	Feedrate and read-in disable, byte 3 (event no.: 580116-580123)							
441 (OM)	Feedrate and read-in disable, byte 4 (event no.: 580124-580131)							
442 (FM)	Read-in disable, byte 1 (event no.: 580200-580207)							
443 (FM)	Read-in disable, byte 2 (event no.: 580208-580215)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
444 (OM)	Read-in disable, byte 3 (event no.: 580216-580223)							
445 (OM)	Read-in disable, byte 4 (event no.: 580224-580231)							
446 (FM)	NC start disable, byte 1 (event no.: 580300-580307)							
447 (OM)	NC start disable, byte 2 (event no.: 580308-580315)							
448 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 581100-581107)							
449 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 581108-581115)							
450 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 581200-581207)							
451 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 581208-581215)							
452 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 581300-581307)							
453 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 581308-581315)							

Table 18-36 DB2, channel range 9

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 9							
	Feedrate disable (event no.: 590000-590015)							
454 (FM)	590007	590006	590005	590004	590003	590002	590001	590000
455 (OM)	590015	590014	590013	590012	590011	590010	590009	590008
456 (FM)	Feedrate and read-in disable, byte 1 (event no.: 590100-590107)							
457 (FM)	Feedrate and read-in disable, byte 2 (event no.: 590108-590115)							
458 (OM)	Feedrate and read-in disable, byte 3 (event no.: 590116-590123)							
459 (OM)	Feedrate and read-in disable, byte 4 (event no.: 590124-590131)							
460 (FM)	Read-in disable, byte 1 (event no.: 590200-590207)							
461 (FM)	Read-in disable, byte 2 (event no.: 590208-590215)							
462 (OM)	Read-in disable, byte 3 (event no.: 590216-590223)							
463 (OM)	Read-in disable, byte 4 (event no.: 590224-590231)							
464 (FM)	NC start disable, byte 1 (event no.: 590300-590307)							
465 (OM)	NC start disable, byte 2 (event no.: 590308-590315)							
466 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 591100-591107)							
467 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 591108-591115)							
468 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 591200-591207)							
469 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 591208-591215)							
470 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 591300-591307)							
471 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 591308-591315)							

Table 18-37 DB2, channel range 10

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 10							
	Feedrate disable (event no.: 500000-500015)							
472 (FM)	500007	500006	500005	500004	500003	500002	500001	500000
473 (OM)	500015	500014	500013	500012	500011	500010	500009	500008
474 (FM)	Feedrate and read-in disable, byte 1 (event no.: 500100-500107)							
475 (FM)	Feedrate and read-in disable, byte 2 (event no.: 500108-500115)							
476 (OM)	Feedrate and read-in disable, byte 3 (event no.: 500116-500123)							
477 (OM)	Feedrate and read-in disable, byte 4 (event no.: 500124-500131)							
478 (FM)	Read-in disable, byte 1 (event no.: 500200-500207)							
479 (FM)	Read-in disable, byte 2 (event no.: 500208-500215)							
480 (OM)	Read-in disable, byte 3 (event no.: 500216-500223)							
481 (OM)	Read-in disable, byte 4 (event no.: 500224-500231)							
482 (FM)	NC start disable, byte 1 (event no.: 500300-500307)							
483 (OM)	NC start disable, byte 2 (event no.: 500308-500315)							
484 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 501100-501107)							
485 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 501108-501115)							
486 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 501200-501207)							
487 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 501208-501215)							
488 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 501300-501307)							
489 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 501308-501315)							

Table 18-38 DB2, axis ranges

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Axis/spindle							
	Feedrate stop / spindle stop for axis/spindle 1 (event no.: 600100-600115)							
490 (FM)	600107	600106	600105	600104	600103	600102	600101	600100
491 (OM)	600115	600114	600113	600112	600111	600110	600109	600108
492 (FM)	Feedrate stop/spindle stop for axis/spindle 2 (event no.: 600200-600207)							
493 (OM)	Feedrate stop/spindle stop for axis/spindle 2 (event no.: 600208-600215)							
494 (FM)	Feedrate stop/spindle stop for axis/spindle 3 (event no.: 600300-600307)							
495 (OM)	Feedrate stop/spindle stop for axis/spindle 3 (event no.: 600308-600315)							
496 (FM)	Feedrate stop/spindle stop for axis/spindle 4 (event no.: 600400-600407)							
497 (OM)	Feedrate stop/spindle stop for axis/spindle 4 (event no.: 600408-600415)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
498 (FM)	Feedrate stop/spindle stop for axis/spindle 5 (event no.: 600500-600507)							
499 (OM)	Feedrate stop/spindle stop for axis/spindle 5 (event no.: 600508-600515)							
500 (FM)	Feedrate stop/spindle stop for axis/spindle 6 (event no.: 600600-600607)							
501 (OM)	Feedrate stop/spindle stop for axis/spindle 6 (event no.: 600608-600615)							
502 (FM)	Feedrate stop/spindle stop for axis/spindle 7 (event no.: 600700-600707)							
503 (OM)	Feedrate stop/spindle stop for axis/spindle 7 (event no.: 600708-600715)							
504 (FM)	Feedrate stop/spindle stop for axis/spindle 8 (event no.: 600800-600807)							
505 (OM)	Feedrate stop/spindle stop for axis/spindle 8 (event no.: 600808-600815)							
506 (FM)	Feedrate stop/spindle stop for axis/spindle 9 (event no.: 600900-600907)							
507 (OM)	Feedrate stop/spindle stop for axis/spindle 9 (event no.: 600908-600915)							
508 (FM)	Feedrate stop/spindle stop for axis/spindle 10 (event no.: 601000-601007)							
509 (OM)	Feedrate stop/spindle stop for axis/spindle 10 (event no.: 601008-601015)							
510 (FM)	Feedrate stop/spindle stop for axis/spindle 11 (event no.: 601100-601107)							
511 (OM)	Feedrate stop/spindle stop for axis/spindle 11 (event no.: 601108-601115)							
512 (FM)	Feedrate stop/spindle stop for axis/spindle 12 (event no.: 601200-601207)							
513 (OM)	Feedrate stop/spindle stop for axis/spindle 12 (event no.: 601208-601215)							
514 (FM)	Feedrate stop/spindle stop for axis/spindle 13 (event no.: 601300-601307)							
515 (OM)	Feedrate stop/spindle stop for axis/spindle 13 (event no.: 601308-601315)							
516 (FM)	Feedrate stop/spindle stop for axis/spindle 14 (event no.: 601400-601407)							
517 (OM)	Feedrate stop/spindle stop for axis/spindle 14 (event no.: 601408-601415)							
518 (FM)	Feedrate stop/spindle stop for axis/spindle 15 (event no.: 601500-601507)							
519 (OM)	Feedrate stop/spindle stop for axis/spindle 15 (event no.: 601508-601515)							
520 (FM)	Feedrate stop/spindle stop for axis/spindle 16 (event no.: 601600-601607)							
521 (OM)	Feedrate stop/spindle stop for axis/spindle 16 (event no.: 601608-601615)							
522 (FM)	Feedrate stop/spindle stop for axis/spindle 17 (event no.: 601700-601707)							
523 (OM)	Feedrate stop/spindle stop for axis/spindle 17 (event no.: 601708-601715)							
524 (FM)	Feedrate stop/spindle stop for axis/spindle 18 (event no.: 601800-601807)							
525 (OM)	Feedrate stop/spindle stop for axis/spindle 18 (event no.: 601808-601815)							
526 (FM)	Feedrate stop/spindle stop for axis/spindle 19 (event no.: 601900-601907)							
527 (OM)	Feedrate stop/spindle stop for axis/spindle 19 (event no.: 601908-601915)							
528 (FM)	Feedrate stop/spindle stop for axis/spindle 20 (event no.: 602000-602007)							
529 (OM)	Feedrate stop/spindle stop for axis/spindle 20 (event no.: 602008-602015)							
530 (FM)	Feedrate stop/spindle stop for axis/spindle 21 (event no.: 602100-602107)							
531 (OM)	Feedrate stop/spindle stop for axis/spindle 21 (event no.: 602108-602115)							
532 (FM)	Feedrate stop/spindle stop for axis/spindle 22 (event no.: 602200-602207)							
533 (OM)	Feedrate stop/spindle stop for axis/spindle 22 (event no.: 602208-602215)							
534 (FM)	Feedrate stop/spindle stop for axis/spindle 23 (event no.: 602300-602307)							
535 (OM)	Feedrate stop/spindle stop for axis/spindle 23 (event no.: 602308-602315)							
536 (FM)	Feedrate stop/spindle stop for axis/spindle 24 (event no.: 602400-602407)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
537 (OM)	Feedrate stop/spindle stop for axis/spindle 24 (event no.: 602408-602415)							
538 (FM)	Feedrate stop/spindle stop for axis/spindle 25 (event no.: 602500-602507)							
539 (OM)	Feedrate stop/spindle stop for axis/spindle 25 (event no.: 602508-602515)							
540 (FM)	Feedrate stop/spindle stop for axis/spindle 26 (event no.: 602600-602607)							
541 (OM)	Feedrate stop/spindle stop for axis/spindle 26 (event no.: 602608-602615)							
542 (FM)	Feedrate stop/spindle stop for axis/spindle 27 (event no.: 602700-602707)							
543 (OM)	Feedrate stop/spindle stop for axis/spindle 27 (event no.: 602708-602715)							
544 (FM)	Feedrate stop/spindle stop for axis/spindle 28 (event no.: 602800-602807)							
545 (OM)	Feedrate stop/spindle stop for axis/spindle 28 (event no.: 602808-602815)							
546 (FM)	Feedrate stop/spindle stop for axis/spindle 29 (event no.: 602900-602907)							
547 (OM)	Feedrate stop/spindle stop for axis/spindle 29 (event no.: 602908-602915)							
548 (FM)	Feedrate stop/spindle stop for axis/spindle 30 (event no.: 603000-603007)							
549 (OM)	Feedrate stop/spindle stop for axis/spindle 30 (event no.: 603008-603015)							
550 (FM)	Feedrate stop/spindle stop for axis/spindle 31 (event no.: 603100-603107)							
551 (OM)	Feedrate stop/spindle stop for axis/spindle 31 (event no.: 603108-603115)							

Table 18-39 DB2, user ranges

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	User ranges							
	User range 0 (event no.: 700000-700015)							
554 (FM)	700007	700006	700005	700004	700003	700002	700001	700000
555 (FM)	700015	700014	700013	700012	700011	700010	700009	700008
556 (FM)	User range 0: Byte 3 (event no.: 700016-700023)							
557 (FM)	User range 0: Byte 4 (event no.: 700024-700031)							
558 (OM)	User range 0: Byte 5 (event no.: 700032-700039)							
559 (OM)	User range 0: Byte 6 (event no.: 700040-700047)							
560 (OM)	User range 0: Byte 7 (event no.: 700048-700055)							
561 (OM)	User range 0: Byte 8 (event no.: 700056-700063)							
562 - 565 (FM)	User range 1: Byte 1 - 4 (event no.: 700100-700131)							
566 - 569 (OM)	User range 1: Byte 5 - 8 (event no.: 700132-700163)							
570 - 573 (FM)	User range 2: Byte 1 - 4 (event no.: 700200-700231)							
574 - 577 (OM)	User range 2: Byte 5 - 8 (event no.: 700232-700263)							
578 - 581 (FM)	User range 3: Byte 1 - 4 (event no.: 700300-700331)							
582 - 585 (OM)	User range 3: Byte 5 - 8 (event no.: 700332-700363)							
586 - 589 (FM)	User range 4: Byte 1 - 4 (event no.: 700400-700431)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
590 - 593 (OM)	User range 4: Byte 5 - 8 (event no.: 700432-700463)							
594 - 597 (FM)	User range 5: Byte 1 - 4 (event no.: 700500-700531)							
598 - 601 (OM)	User range 5: Byte 5 - 8 (event no.: 700532-700563)							
602 - 605 (FM)	User range 6: Byte 1 - 4 (event no.: 700600-700631)							
606 - 609 (OM)	User range 6: Byte 5 - 8 (event no.: 700632-700663)							
610 - 613 (FM)	User range 7: Byte 1 - 4 (event no.: 700700-700731)							
614 - 617 (OM)	User range 7: Byte 5 - 8 (event no.: 700732-700763)							
618 - 621 (FM)	User range 8: Byte 1 - 4 (event no.: 700800-700831)							
622 - 625 (OM)	User range 8: Byte 5 - 8 (event no.: 700832-700863)							
626 - 629 (FM)	User range 9: Byte 1 - 4 (event no.: 700900-700931)							
630 - 633 (OM)	User range 9: Byte 5 - 8 (event no.: 700932-700963)							
634 - 637 (FM)	User range 10: Byte 1 - 4 (event no.: 701000-701031)							
638 - 641 (OM)	User range 10: Byte 5 - 8 (event no.: 701032-701063)							
642 - 645 (FM)	User range 11: Byte 1 - 4 (event no.: 701100-701131)							
646 - 649 (OM)	User range 11: Byte 5 - 8 (event no.: 701132-701163)							
650 - 653 (FM)	User range 12: Byte 1 - 4 (event no.: 701200-701231)							
654 - 657 (OM)	User range 12: Byte 5 - 8 (event no.: 701232-701263)							
658 - 661 (FM)	User range 13: Byte 1 - 4 (event no.: 701300-701331)							
662 - 665 (OM)	User range 13: Byte 5 - 8 (event no.: 701332-701363)							
666 - 669 (FM)	User range 14: Byte 1 - 4 (event no.: 701400-701431)							
670 - 673 (OM)	User range 14: Byte 5 - 8 (event no.: 701432-701463)							
674 - 677 (FM)	User range 15: Byte 1 - 4 (event no.: 701500-701531)							
678 - 681 (OM)	User range 15: Byte 5 - 8 (event no.: 701532-701563)							
682 - 685 (FM)	User range 16: Byte 1 - 4 (event no.: 701600-701631)							
686 - 689 (OM)	User range 16: Byte 5 - 8 (event no.: 701632-701663)							
690 - 693 (FM)	User range 17: Byte 1 - 4 (event no.: 701700-701731)							
694 - 697 (OM)	User range 17: Byte 5 - 8 (event no.: 701732-701763)							
698 - 701 (FM)	User range 18: Byte 1 - 4 (event no.: 701800-701831)							
702 - 705 (OM)	User range 18: Byte 5 - 8 (event no.: 701832-701863)							
706 - 709 (FM)	User range 19: Byte 1 - 4 (event no.: 701900-701931)							
710 - 713 (OM)	User range 19: Byte 5 - 8 (event no.: 701932-701963)							
714 - 717 (FM)	User range 20: Byte 1 - 4 (event no.: 702000-702031)							
718 - 721 (OM)	User range 20: Byte 5 - 8 (event no.: 702032-702063)							
722 - 725 (FM)	User range 21: Byte 1 - 4 (event no.: 702100-702131)							
726 - 729 (OM)	User range 21: Byte 5 - 8 (event no.: 702132-702163)							
730 – 733 (FM)	User range 22: Byte 1 - 4 (event no.: 702200-702231)							
734 - 737 (OM)	User range 22: Byte 5 - 8 (event no.: 702232-702263)							
738 - 741 (FM)	User range 23: Byte 1 - 4 (event no.: 702300-702331)							
742 - 745 (OM)	User range 23: Byte 5 - 8 (event no.: 702332-702363)							
746 - 749 (FM)	User range 24: Byte 1 - 4 (event no.: 702400-702431)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
750 - 753 (OM)	User range 24: Byte 5 - 8 (event no.: 702432-702463)							
754 - 757 (FM)	User range 25: Byte 1 - 4 (event no.: 702500-702531)							
758 - 761 (OM)	User range 25: Byte 5 - 8 (event no.: 702532-702563)							
762 - 765 (FM)	User range 26: Byte 1 - 4 (event no.: 702600-702631)							
766 - 769 (OM)	User range 26: Byte 5 - 8 (event no.: 702632-702663)							
770 - 773 (FM)	User range 27: Byte 1 - 4 (event no.: 702700-702731)							
774 - 777 (OM)	User range 27: Byte 5 - 8 (event no.: 702732-702763)							
778 - 781 (FM)	User range 28: Byte 1 - 4 (event no.: 702800-702831)							
782 - 785 (OM)	User range 28: Byte 5 - 8 (event no.: 702832-702863)							
786 - 789 (FM)	User range 29: Byte 1 - 4 (event no.: 702900-702931)							
790 - 793 (OM)	User range 29: Byte 5 - 8 (event no.: 702932-702963)							
794 - 797 (FM)	User range 30: Byte 1 - 4 (event no.: 703000-703031)							
798 - 801 (OM)	User range 30: Byte 5 - 8 (event no.: 703032-703063)							
802 - 805 (FM)	User range 31: Byte 1 - 4 (event no.: 703100-703131)							
806 - 809 (OM)	User range 31: Byte 5 - 8 (event no.: 703132-703163)							
810 - 813 (FM)	User area 32: Byte 1 - 4 (event no.: 703200 - 703231)							
814 - 817 (BM)	User area 32: Byte 5 - 8 (event no.: 703232 - 703263)							
818 - 821 (FM)	User area 33: Byte 1 - 4 (event no.: 703300 - 703331)							
822 - 825 (OM)	User area 33: Byte 5 - 8 (event no.: 703332 - 703363)							
826 - 829 (FM)	User area 34: Byte 1 - 4 (event no.: 703400 - 703431)							
830 - 833 (OM)	User area 34: Byte 5 - 8 (event no.: 703432 - 703463)							
834 - 837 (FM)	User area 35: Byte 1 - 4 (event no.: 703500 - 703531)							
838 - 841 (OM)	User area 35: Byte 5 - 8 (event no.: 703532 - 703563)							
842 - 845 (FM)	User area 36: Byte 1 - 4 (event no.: 703600 - 703631)							
846 - 789 (OM)	User area 36: Byte 5 - 8 (event no.: 703632 - 703663)							
850 - 853 (FM)	User area 37: Byte 1 - 4 (event no.: 703700 - 703731)							
854 - 857 (OM)	User area 37: Byte 5 - 8 (event no.: 703732 - 703763)							
858 - 861 (FM)	User area 38: Byte 5 - 8 (event no.: 703800 -703831)							
862 - 865 (OM)	User area 38: Byte 1 - 4 (event no.: 703832 -703863)							
866 - 869 (FM)	User area 39: Byte 5 - 8 (event no.: 703900 -703931)							
870 - 873 (OM)	User area 39: Byte 1 - 4 (event no.: 703932 -703963)							
874 - 877 (FM)	User area 40: Byte 1 - 4 (event no.: 704000-704031)							
878 - 881 (OM)	User area 40: Byte 5 - 8 (event no.: 704032-704063)							
882 - 885 (FM)	User area 41: Byte 1 - 4 (event no.: 704100-704131)							
886 - 889 (OM)	User area 41: Byte 5 - 8 (event no.: 704132-704163)							
890 - 893 (FM)	User area 42: Byte 1 - 4 (event no.: 704200-704231)							
894 - 897 (OM)	User area 42: Byte 5 - 8 (event no.: 704232-704263)							
898 - 901 (FM)	User area 43: Byte 1 - 4 (event no.: 704300-704331)							
902 - 905 (OM)	User area 43: Byte 5 - 8 (event no.: 704332-704363)							
906 - 909 (FM)	User area 44: Byte 1 - 4 (event no.: 704400-704431)							

DB2	Signals for PLC events (PLC → HMI) FB1 parameter "ExtendAlMsg" = TRUE							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
910 - 913 (OM)	User area 44: Byte 5 - 8 (event no.: 704432-704463)							
914 - 917 (FM)	User area 45: Byte 1 - 4 (event no.: 704500-704531)							
918 - 921 (OM)	User area 45: Byte 5 - 8 (event no.: 704532-704563)							
922 - 925 (FM)	User area 46: Byte 1 - 4 (event no.: 704600-704631)							
926 - 929 (OM)	User area 46: Byte 5 - 8 (event no.: 704632-704663)							
930 - 933 (FM)	User area 47: Byte 1 - 4 (event no.: 704700-704731)							
934 - 937 (OM)	User area 47: Byte 5 - 8 (event no.: 704732-704763)							
938 - 941 (FM)	User area 48: Byte 1 - 4 (event no.: 704800-704831)							
942 - 945 (OM)	User area 48: Byte 5 - 8 (event no.: 704832-704863)							
946 - 989 (FM)	User area 49: Byte 1 - 4 (event no.: 704900-704931)							
950 - 953 (OM)	User area 49: Byte 5 - 8 (event no.: 704932-704963)							
954 - 957 (FM)	User area 50: Byte 1 - 4 (event no.: 705000-705031)							
958 - 961 (OM)	User area 50: Byte 5 - 8 (event no.: 705032-705063)							
962 -965 (FM)	User area 51: Byte 1 - 4 (event no.: 705100-705131)							
966 - 969 (OM)	User area 51: Byte 5 - 8 (event no.: 705132-705163)							
970 - 973 (FM)	User area 52: Byte 1 - 4 (event no.: 705200-705231)							
974 - 977 (OM)	User area 52: Byte 5 - 8 (event no.: 705232-705263)							
978 - 981 (FM)	User area 53: Byte 1 - 4 (event no.: 705300-705331)							
982 - 985 (OM)	User area 53: Byte 5 - 8 (event no.: 705332-705363)							
986 - 989 (FM)	User area 54: Byte 1 - 4 (event no.: 705400-705431)							
990 - 993 (OM)	User area 54: Byte 5 - 8 (event no.: 705432-705463)							
994 - 997 (FM)	User area 55: Byte 1 - 4 (event no.: 705500-705531)							
998 - 1001 (OM)	User area 55: Byte 5 - 8 (event no.: 705532-705563)							
1002 - 1005 (FM)	User area 56: Byte 1 - 4 (event no.: 705600-705631)							
1006 - 1009 (OM)	User area 56: Byte 5 - 8 (event no.: 705632-705663)							
1010 -1013 (FM)	User area 57: Byte 1 - 4 (event no.: 705700-705731)							
1014 - 1017 (OM)	User area 57: Byte 5 - 8 (event no.: 705732-705763)							
1018 - 1021 (FM)	User area 58: Byte 1 - 4 (event no.: 705800-705831)							
1022 - 1025 (OM)	User area 58: Byte 5 - 8 (event no.: 705832-705863)							
1026 - 1029 (FM)	User area 59: Byte 1 - 4 (event no.: 705900-705931)							
1030 - 1033 (OM)	User area 59: Byte 5 - 8 (event no.: 705932-705963)							
1034 - 1037 (FM)	User area 60: Byte 1 - 4 (event no.: 706000-706031)							
1038 - 941 (OM)	User area 60: Byte 5 - 8 (event no.: 706032-706063)							
1042 - 1045 (FM)	User area 61: Byte 1 - 4 (event no.: 706100-706131)							
1046 - 1089 (OM)	User area 61: Byte 5 - 8 (event no.: 706132-706163)							
1050 - 1053 (FM)	User area 62: Byte 1 - 4 (event no.: 706200-706231)							
1054 - 1057 (OM)	User area 62: Byte 5 - 8 (event no.: 706232-706263)							
1058 - 1061 (FM)	User area 63: Byte 1 - 4 (event no.: 706300-706331)							
1062 -1065 (BM)	User area 63: Byte 5 - 8 (event no.: 706332-706363)							

Note

Parameterizing user ranges

The number of user ranges (maximum of 64) can be parameterized using FB 1 "MsgUser".

18.1.7 Signals from/to the NC, PLC and operating software

18.1.7.1 DB10, onboard inputs and outputs of the NC

Table 18-40 DB10, onboard inputs and outputs of the NC

DB10	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Disable of the digital NC inputs  see also (Page 434)							
	Input without hardware				Input onboard			
	8	7	6	5	4	3	2	1
DBB1	Setting of the digital NC inputs from the PLC  see also (Page 435)							
	Input without hardware				Input onboard			
	8	7	6	5	4	3	2	1
DBB2 - DBB3	Not assigned							
DBB4	Disable of the digital NC outputs  see also (Page 435)							
	Output without hardware				Output onboard			
	8	7	6	5	4	3	2	1
DBB5	Overwrite mask of the digital NC outputs  see also (Page 436)							
	Output without hardware				Output onboard			
	8	7	6	5	4	3	2	1
DBB6	Setting value of the digital NC outputs from the PLC  see also (Page 436)							
	Output without hardware				Output onboard			
	8	7	6	5	4	3	2	1
DBB7	Input mask of the digital NC outputs  see also (Page 437)							
	Output without hardware				Output onboard			
	8	7	6	5	4	3	2	1

DB10	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB8 - DBB29	Machine axis numbers table for FC 19, FC 24, FC 25, FC 26 (1st MCP)							
DBW30	Upper limit of the machine axis numbers for FC 19, FC 24 (1st MCP) With 0, the maximum number of machine axis numbers applies							
DBB32 - DBB53	Machine axis numbers table for FC 19, FB 24, FB 25, FB 26 (2nd MCP)							
DBW54	Upper limit of the machine axis numbers for FC 19, FC 24 (2nd MCP) With 0, the maximum number of machine axis numbers applies							

18.1.7.2 DB10, general signals to the NC

Table 18-41 DB10, general signals to the NC

DB10	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB56	Key-operated switch position see also (Page 438)					Acknowledge Emergency Stop see also (Page 438)	Emergency Stop see also (Page 438)	
	Bit 3	Bit 2	Bit 1	Bit 0				
DBB57					Reserved			INC inputs in the mode group range active
DBB58	Collision avoidance: Deactivate protection area group see also (Page 439)							
	For operating mode: JOG				For operating mode: AUTOMATIC			
	Workpieces	Workholder	Tools	Machine	Workpieces	Workholder	Tools	Machine
DBB59								

18.1.7.3 DB10, onboard inputs and outputs from the NC/operating software

Table 18-42 DB10, onboard inputs and outputs from the NC/operating software

DB10	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB60	Actual value of the digital NC inputs				Actual value of the digital onboard inputs of the NC see also (Page 440)			
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
DBB61 - DBB63								
DBB64	Setpoint for the digital outputs of the NC without hardware				Setpoint for the digital onboard outputs of the NC see also (Page 441)			
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB65 - DBB67								
DBB68	Handwheel 1 is operated							
DBB69	Handwheel 2 is operated							
DBB70	Handwheel 3 is operated							
DBB71	Change counter, inch/metric system of units							
DBB72	Status of the displayed actual value screen (1st MCP)							
HT 8 → operating software	Traversing keys shown						MCS / WCS	Display valid
DBB73	Status of the displayed actual value screen (2nd MCP)							
HT 8 → operating software	Traversing keys shown						MCS / WCS	Display valid
DBB74 - DBB79	Machine axis numbers of the displayed axes (1st MCP) MCPT1AxisFromHMI							
HT 8 → operating software								
DBB80 - DBB85	Machine axis numbers of the displayed axes (2nd MCP) MCP2AxisFromHMI							
HT 8 → operating software								
DBW86	Reserved							
DBB88	Reserved							

18.1.7.4 DB10, selection and status signals from the operating software

Table 18-43 DB10, selection and status signals from the operating software

DB10	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB90								
SINUMERIK Integrate → PLC								
DBB91								
PLC → SINUMERIK Integrate								
DBB92		Suppress fault message in case of failure				Slave OK		
GP → PLC		PN bus	DP1 bus	MPI/DP bus		PN bus	DP1 bus	MPI/DP bus
DBB93	Deactivate collision avoidance							
Operating software → PLC	JOG mode				AUTO mode			
	Workpieces	Workholder	Tools	Machine	Workpieces	Workholder	Tools	Machine
DBB94				Alarm Direct keys 2 Communication error	Alarm, direct keys 1 Communication error			
DBB95								
DBB96	Set language ID of the operating software							
Operating software → PLC								
DBB97					Channel number for handwheel 1 see also (Page 442)			
Operating software → PLC					D	C	B	A
DBB98					Channel number for handwheel 2 see also (Page 443)			
Operating software → PLC					D	C	B	A
DBB99					Channel number for handwheel 3 see also (Page 443)			
Operating software → PLC					D	C	B	A

DB10	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB100 Operating software → PLC				Axis number for handwheel 1 see also (Page 444)				
	Machine axis see also (Page 446)	Handwheel 1 selected see also (Page 445)	Define handwheel 1 as contour handwheel see also (Page 444)	E	D	C	B	A
DBB101 Operating software → PLC				Axis number for handwheel 2 see also (Page 446)				
	Machine axis see also (Page 447)	Handwheel 2 selected see also (Page 446)	Define handwheel 2 as contour handwheel see also (Page 446)	E	D	C	B	A
DBB102 Operating software → PLC				Axis number for handwheel 3 see also (Page 447)				
	Machine axis see also (Page 447)	Handwheel 3 selected see also (Page 447)	Define handwheel 3 as contour handwheel see also (Page 447)	E	D	C	B	A
DBB103 Operating software → PLC	Operating software battery alarm see also (Page 448)	Operating software temperature limit see also (Page 448)	AT box ready see also (Page 447)	Operating software fan monitoring	Operating software monitor hard disk			Remote diagnostics active see also (Page 447)

18.1.7.5 DB10, general signals from the NC

Table 18-44 DB10, general signals to the PLC

DB10	Signals to the PLC							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB104 GP → PLC	NC CPU: Ready[1)] see also (Page 448)	1st OB1 cycle		Op2Key ready	Op1Key ready	HHU ready	MCP 2 ready	MCP 1 ready
DBB105 GP → PLC								TOOLMAN: Command Cancel
DBB106 NC → PLC							Emergency Stop: Active see also (Page 449)	Collision avoidance: Protection area monitoring active
DBB107 NC → PLC	Inch system of units	NCU link: Active see also (Page 449)					Measuring input activated see also (Page 449)	
							Pushbutton 2	Pushbutton 1
DBB108 NC → PLC	NC ready see also (Page 450)	Drive ready see also (Page 450)	Drives in cyclic operation see also (Page 450)		Operating software ready see also (Page 450)	Operator panel at MPI: "ready"	Operator panel 2: "ready"	
DBB109 NC → PLC	NC battery alarm see also (Page 452)	Air temperature alarm see also (Page 452)	Heat sink temperature NCU alarm see also (Page 451)	PC operating system error				NC alarm is active see also (Page 451)
DBB110 NC → PLC	Software cam minus see also (Page 452)							
	7	8	5	4	3	2	1	0
DBB111 NC → PLC	Software cam minus see also (Page 452)							
	15	14	13	12	11	10	9	8
DBB112 NC → PLC	Software cam minus see also (Page 452)							
	23	22	21	20	19	18	17	16

DB10	Signals to the PLC							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB113 NC → PLC	Software cam minus see also (Page 452)							
	31	30	29	28	27	26	25	24
DBB114 NC → PLC	Software cam plus see also (Page 453)							
	7	6	5	4	3	2	1	0
DBB115 NC → PLC	Software cam plus see also (Page 453)							
	15	14	13	12	11	10	9	8
DBB116 NC → PLC	Software cam plus see also (Page 453)							
	23	22	21	20	19	18	17	16
DBB117 NC → PLC	Software cam plus see also (Page 453)							
	31	30	29	28	27	26	25	24
DBB118-DBB121 SINUMERIK Integrate → PLC	SINUMERIK Integrate data							

[1] DB10 DBX104.7 (NC CPU: ready) The signal should be included in the machine safety circuit.

18.1.7.6 DB10, external digital NC inputs

Table 18-45 DB10, external digital NC inputs

DB10	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB122	Disable of the external digital NC inputs see also (Page 454)							
	16	15	14	13	12	11	10	9
DBB123	Values from the PLC for the external digital NC inputs see also (Page 454)							
	16	15	14	13	12	11	10	9
DBB124	Disable of the external digital NC inputs see also (Page 454)							
	24	23	22	21	20	19	18	17
DBB125	Values from the PLC for the external digital NC inputs see also (Page 455)							
	24	23	22	21	20	19	18	17

DB10	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB126	Disable of the external digital NC inputs see also (Page 455)							
	32	31	30	29	28	27	26	25
DBB127	Values from the PLC for the external digital NC inputs see also (Page 456)							
	32	31	30	29	28	27	26	25
DBB128	Disable of the external digital NC inputs see also (Page 456)							
	40	39	38	37	36	35	34	33
DBB129	Values from the PLC for the external digital NC inputs see also (Page 456)							
	40	39	38	37	36	35	34	33

18.1.7.7 DB10, external digital NC outputs

Table 18-46 DB10, external digital NC outputs

DB10	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB130	Disable of the external digital NC outputs see also (Page 457)							
	16	15	14	13	12	11	10	9
DBB131	Overwrite mask for the external digital NC outputs see also (Page 457)							
	16	15	14	13	12	11	10	9
DBB132	Value from the PLC for the external digital NC outputs see also (Page 458)							
	16	15	14	13	12	11	10	9
DBB133	Input mask for the external digital NC outputs see also (Page 459)							
	16	15	14	13	12	11	10	9
DBB134	Disable of the external digital NC outputs see also (Page 459)							
	24	23	22	21	20	19	18	17
DBB135	Overwrite mask for the external digital NC outputs see also (Page 460)							
	24	23	22	21	20	19	18	17
DBB136	Value from the PLC for the external digital NC outputs see also (Page 460)							
	24	23	22	21	20	19	18	17

DB10	Signals to the NC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB137	Input mask for the external digital NC outputs see also (Page 461)							
	24	23	22	21	20	19	18	17
DBB138	Disable of the external digital NC outputs see also (Page 461)							
	32	31	30	29	28	27	26	25
DBB139	Overwrite mask for the external digital NC outputs see also (Page 462)							
	32	31	30	29	28	27	26	25
DBB140	Value from the PLC for the external digital NC outputs see also (Page 462)							
	32	31	30	29	28	27	26	25
DBB141	Input mask for the external digital NC outputs see also (Page 463)							
	32	31	30	29	28	27	26	25
DBB142	Disable of the external digital NC outputs see also (Page 464)							
	40	39	38	37	36	35	34	33
DBB143	Overwrite mask for the external digital NC outputs see also (Page 464)							
	40	39	38	37	36	35	34	33
DBB144	Value from the PLC for the external digital NC outputs see also (Page 465)							
	40	39	38	37	36	35	34	33
DBB145	Input mask for the external digital NC outputs see also (Page 465)							
	40	39	38	37	36	35	34	33

18.1.7.8 DB10, external analog NC inputs

Table 18-47 DB10, external analog NC inputs

DB10	Signals to the NC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB146	Analog NC inputs disabled see also (Page 466)							
	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
DBB147	Input analog value input for the NC from the PLC see also (Page 466)							
	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
DBW148	Setpoint from the PLC for analog input 1 of the NC							

DB10	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW150	Setpoint from the PLC for analog input 2 of the NC see also (Page 467)							
DBW152	Setpoint from the PLC for analog input 3 of the NC see also (Page 467)							
DBW154	Setpoint from the PLC for analog input 4 of the NC see also (Page 467)							
DBW156	Setpoint from the PLC for analog input 5 of the NC see also (Page 467)							
DBW158	Setpoint from the PLC for analog input 6 of the NC see also (Page 467)							
DBW160	Setpoint from the PLC for analog input 7 of the NC see also (Page 467)							
DBW162	Setpoint from the PLC for analog input 8 of the NC see also (Page 467)							
DBW164								

18.1.7.9 DB10, external analog NC outputs

Table 18-48 DB10, external analog NC outputs

DB10	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB166	Overwrite mask for the analog NC outputs see also (Page 467)							
	8	7	6	5	4	3	2	1
DBB167	Input mask for the analog NC outputs see also (Page 468)							
	8	7	6	5	4	3	2	1
DBB168	Analog NC outputs disabled see also (Page 468)							
	8	7	6	5	4	3	2	1
DBB169	Reserved							
DBW170	Setpoint from the PLC for analog output 1 of the NC see also (Page 469)							
DBW172	Setpoint from the PLC for analog output 2 of the NC see also (Page 469)							
DBW174	Setpoint from the PLC for analog output 3 of the NC see also (Page 469)							
DBW176	Setpoint from the PLC for analog output 4 of the NC see also (Page 469)							

DB10	Signals to the NC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW178	Setpoint from the PLC for analog output 5 of the NC see also (Page 469)							
DBW180	Setpoint from the PLC for analog output 6 of the NC see also (Page 469)							
DBW182	Setpoint from the PLC for analog output 7 of the NC see also (Page 469)							
DBW184	Setpoint from the PLC for analog output 8 of the NC see also (Page 469)							

18.1.7.10 DB10, external digital NC inputs and outputs

Table 18-49 DB10, external digital NC inputs and outputs

DB10	Signals from the NC (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB186	Actual value of the external digital NC inputs see also (Page 469)							
	16	15	14	13	12	11	10	9
DBB187	Actual value of the external digital NC inputs see also (Page 470)							
	24	23	22	21	20	19	18	17
DBB188	Actual value of the external digital NC inputs see also (Page 470)							
	32	31	30	29	28	27	26	25
DBB189	Actual value of the external digital NC inputs see also (Page 471)							
	40	39	38	37	36	35	34	33
DBB190	NC setpoint for the external digital NC outputs see also (Page 471)							
	16	15	14	13	12	11	10	9
DBB191	NC setpoint for the external digital NC outputs see also (Page 472)							
	24	23	22	21	20	19	18	17
DBB192	NC setpoint for the external digital NC outputs see also (Page 472)							
	32	31	30	29	28	27	26	25
DBB193	NC setpoint for the external digital NC outputs see also (Page 473)							
	40	39	38	37	36	35	34	33

18.1.7.11 DB10, analog NC inputs and outputs

Table 18-50 DB10, analog NC inputs and outputs

DB10	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW194	Actual value of analog input 1 of the NC see also (Page 473)							
DBW196	Actual value of analog input 2 of the NC see also (Page 473)							
DBW198	Actual value of analog input 3 of the NC see also (Page 473)							
DBW200	Actual value of analog input 4 of the NC see also (Page 473)							
DBW202	Actual value of analog input 5 of the NC see also (Page 473)							
DBW204	Actual value of analog input 6 of the NC see also (Page 473)							
DBW206	Actual value of analog input 7 of the NC see also (Page 473)							
DBW208	Actual value of analog input 8 of the NC see also (Page 473)							
DBW210	Setpoint of analog output 1 of the NC see also (Page 473)							
DBW212	Setpoint of analog output 2 of the NC see also (Page 473)							
DBW214	Setpoint of analog output 3 of the NC see also (Page 473)							
DBW216	Setpoint of analog output 4 of the NC see also (Page 473)							
DBW218	Setpoint of analog output 5 of the NC see also (Page 473)							
DBW220	Setpoint of analog output 6 of the NC see also (Page 473)							
DBW222	Setpoint of analog output 7 of the NC see also (Page 473)							
DBW224	Setpoint of analog output 8 of the NC see also (Page 473)							

18.1.7.12 DB10, collision avoidance: Protection area active

Table 18-51 DB10, collision avoidance: Protection zone active

DB10	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB226	Collision avoidance: Protection zone active (bit)  see also (Page 474)							
	7	6	5	4	3	2	1	0
DBB227	Collision avoidance: Protection zone active (bit)  see also (Page 474)							
	15	14	13	12	11	10	9	8
DBB228	Collision avoidance: Protection zone active (bit)  see also (Page 474)							
	23	22	21	20	19	18	17	16
DBB229	Collision avoidance: Protection zone active (bit)  see also (Page 474)							
	31	30	29	28	27	26	25	24
DBB230	Collision avoidance: Protection zone active (bit)  see also (Page 474)							
	39	38	37	36	35	34	33	32
DBB231	Collision avoidance: Protection zone active (bit)  see also (Page 474)							
	47	46	45	44	43	42	41	40
DBB232	Collision avoidance: Protection zone active (bit)  see also (Page 474)							
	55	54	53	52	51	50	49	48
DBB233	Collision avoidance: Protection zone active (bit)  see also (Page 474)							
	63	62	61	60	59	58	57	56

18.1.7.13 DB10, collision avoidance: Activate protection area

Table 18-52 DB10, collision avoidance: Activate protection zone

DB10	Signals from the PLC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB234	Collision avoidance: Activate protection zone (bit)  see also (Page 475)							
	7	6	5	4	3	2	1	0
DBB235	Collision avoidance: Activate protection zone (bit)  see also (Page 475)							
	15	14	13	12	11	10	9	8

DB10	Signals from the PLC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB236	Collision avoidance: Activate protection zone (bit) see also (Page 475)							
	23	22	21	20	19	18	17	16
DBB237	Collision avoidance: Activate protection zone (bit) see also (Page 475)							
	31	30	29	28	27	26	25	24
DBB238	Collision avoidance: Activate protection zone (bit) see also (Page 475)							
	39	38	37	36	35	34	33	32
DBB239	Collision avoidance: Activate protection zone (bit) see also (Page 475)							
	47	46	45	44	43	42	41	40
DBB240	Collision avoidance: Activate protection zone (bit) see also (Page 475)							
	55	54	53	52	51	50	49	48
DBB241	Collision avoidance: Activate protection zone (bit) see also (Page 475)							
	63	62	61	60	59	58	57	56

18.1.7.14 DB10, extension, handwheel signals from the NC

Table 18-53 DB10, extension, handwheel signals from the NC

DB10	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB242	Handwheel 4 is moved							
DBB243	Handwheel 5 is moved							
DBB244	Handwheel 6 is moved							
DBB245			Ethernet handwheel is stationary see also (Page 475)					
			Handwheel 6	Handwheel 5	Handwheel 4	Handwheel 3	Handwheel 2	Handwheel 1
DBB246	Reserved							

18.1.7.15 DB10, interface robot status.

Table 18-54 DB10, signals from the robot

DB10	Signals from the PLC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB248	Robot status byte 0							
DBB249	Robot status byte 1							
DBB250	Robot status byte 2							
DBB251	Robot status byte 3							
DBB252	Robot status byte 4							
DBB253	Robot status byte 5							
DBB254	Robot status byte 6							
DBB255	Robot status byte 7							

18.1.7.16 DB10, interface robot status

Table 18-55 DB10, signals to the robot

DB10	Signals from the NC (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB256	Robot control byte 0							
DBB257	Robot control byte 1							
DBB258	Robot control byte 2							
DBB259	Robot control byte 3							
DBB260	Robot control byte 4							
DBB261	Robot control byte 5							
DBB262	Robot control byte 6							
DBB263	Robot control byte 7							

18.1.8 Mode group-specific signals

18.1.8.1 DB11, mode signals 1 to the NC

The start address of a mode group interface can be calculated as follows:

Start address = 20 * (n - 1), with n = mode group number = 1, 2, 3, ...

Table 18-56 DB11, mode signals mode group 1 to NC

DB11	Signals to mode group 1 (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Mode group reset see also (Page 478)	Mode group stop axes plus spindles see also (Page 478)	Mode group Stop see also (Page 477)	Mode change disable see also (Page 477)		Mode: Requirement		
						JOG see also (Page 476)	MDI see also (Page 476)	AUTO see also (Page 476)
DBB1	Single block					Machine function: Requirement		
	Type A see also (Page 480)	Type B see also (Page 480)				REF see also (Page 480)	REPOS see also (Page 479)	TEACH IN see also (Page 479)
DBB2	Machine function: Requirement **Note**: The interface must be enabled explicitly: DB10 DBX57.0 (Page 340) = 1							
		Continuous traversing	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB3								

18.1.8.2 DB11, mode signals 1 from the NC

Table 18-57 DB11, mode signals mode group 1 from NC

DB11	Signals from mode group 1 (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB4 Operating software → PLC						Mode strobe		
						JOG see also (Page 481)	MDI see also (Page 481)	AUTO see also (Page 481)
DBB5 Operating software → PLC						Machine function strobe		
						REF see also (Page 482)	REPOS see also (Page 482)	TEACH IN see also (Page 482)
DBB6	All channels in the reset state see also (Page 484)		NC internal JOG active /FB-K1/	Mode group reset performed /FB-K1/	Mode group ready see also (Page 483)	Active operating mode		
						JOG see also (Page 483)	MDI see also (Page 483)	AUTO see also (Page 483)

DB11	Signals from mode group 1 (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB7						Active machine function		
						REF see also (Page 485)	REPOS see also (Page 484)	TEACH IN see also (Page 484)
DBB8	Machine functions: Selection **Note**: The interface must be enabled explicitly: DB10 DBX57.0 (Page 340)= 1							
		Continuous traversing active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB9								

Table 18-58 DB11, mode signals mode group 1 from the HMI

DB11	Signals to mode group 1 (HMI → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB10	Machine functions							
Operating software → PLC		Continuous traversing active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB11								

18.1.8.3 DB11, mode signals 2 to the NC

Table 18-59 DB11, mode signals mode group 2 to NC

DB11	Signals to mode group 2 (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB20	Mode group reset	Mode group Stop axes plus spindles	Mode group Stop	Mode change disable		Mode: Requirement		
						JOG	MDI	AUTO
DBB21	Single block					Machine function: Requirement		
	Type A	Type B				REF	REPOS	TEACH IN
DBB22	Machine function: Requirement **Note**: The interface must be enabled explicitly: DB10 DBX57.0 = (Page 340) 1							
		Continuous traversing	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB23								

18.1.8.4 DB11, mode signals 2 from NC

Table 18-60 DB11, mode signals mode group 2 from NC

DB11	**Signals from mode group 2 (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB24						Mode strobe		
Operating software → PLC						JOG	MDI	AUTO
DBB25						Machine function strobe		
Operating software → PLC						REF	REPOS	TEACH IN
DBB26	All channels in state: Reset		JOG is active internally in the NC	Mode group reset performed	Mode group ready	Active operating mode		
						JOG	MDI	AUTO
DBB27						Active machine function		
						REF	REPOS	TEACH IN
DBB28	Active machine function **Note**: The interface must be enabled explicitly: DB10 DBX57.0 (Page 344) = 1							
		Continuous traversing active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB29								

Table 18-61 DB11, mode signals mode group 2 from the HMI

DB11	**Signals to mode group 2 (HMI → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB30	Machine functions							
Operating software → PLC		Continuous traversing	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB31								

18.1.9 Safety Integrated (SPL)

18.1.9.1 DB18, parameterization part

Table 18-62 DB18, parameterization part

DB18	SPL signals (PLC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0 - DBB34								
DBB36							Stop E see also (Page 485)	SPL ready see also (Page 485)
DBB37								

18.1.9.2 DB18, data area / errors

Table 18-63 DB18, data area / errors

DB18 /FBSsl/	SPL signals (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Data area of the SPL inputs/outputs							
DBB38 - DBB41	SPL_DATA.INSEP [1..32] see also (Page 486)							
DBB42 - DBB45	SPL_DATA.INSEP [33..64] see also (Page 486)							
DBB46 - DBB49	SPL_DATA.OUTSEP [1..32] see also (Page 486)							
DBB50 - DBB53	SPL_DATA.OUTSEP [33..64]							
	Data area for user SPL							
DBB54 - DBB57	SPL_DATA.INSIP [1..32]							
DBB58 - DBB61	SPL_DATA.INSIP [33..64]							
DBB62 - DBB65	SPL_DATA.OUTSIP [1..32]							
DBB66 - DBB69	SPL_DATA.OUTSIP [33..64]							

DB18 /FBSsl/	**SPL signals (PLC ⟷ NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB70 - DBB73	SPL_DATA.MARKERSIP [1..32]							
DBB74 - DBB77	SPL_DATA.MARKERSIP [33..64]							
	Difference in signal level NC - PLC for diagnostics							
DBB78 - DBB81	-DBB81 SPL_DELTA.INSEP [1..32]							
DBB82 - DBB85	SPL_DELTA.INSEP [33..64]							
DBB86 - DBB89	SPL_DELTA.OUTSEP [1..32]							
DBB90 - DBB93	SPL_DELTA.OUTSEP [33..64]							
DBB94 - DBB97	SPL_DELTA.INSIP [1..32]							
DBB98 - DBB101	SPL_DELTA.INSIP [33..64]							
DBB102- DBB105	SPL_DELTA.OUTSIP [1..32]							
DBB106 - DBB109	SPL_DELTA.OUTSIP [33..64]							
DBB110 - DBB113	SPL_DELTA.MARKERSIP [1..32]							
DBB114 - DBB117	SPL_DELTA.MARKERSIP [33..64]							
DBB118								CMDSI
DBB119		NC signals a stop to the PLC	Systemerror CDC	Crosswise data comparison error, SPL protection status		PROFIsafe communication error		
DBD120	STATSI 0 = no error Crosswise data comparison error tripped							
DBD124	CDC stack level display (diagnostics capability: How many SPL signals currently have a different level)							

18.1.9.3 DB18, additional data areas

Table 18-64 DB18, additional data areas

DB18 /FBSIsl/	SPL signals (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Data area of the single-channel inputs/outputs							
DBB128	PLC_SI_OUT [1..8]							
NC → PLC								
DBB129	PLC_SI_OUT [9..16]							
NC → PLC								
DBB130	PLC_SI_OUT [17..24]							
NC → PLC								
DBB131	PLC_SI_OUT [25..32]							
NC → PLC								
DBB132	PLC_SI_IN [1..8]							
NC → PLC								
DBB133	PLC_SI_IN [9..16]							
NC → PLC								
DBB134	PLC_SI_IN [17..24]							
NC → PLC								
DBB135	PLC_SI_IN [25..32]							
NC → PLC								
DBB136- DBB137	SPL status							
DBB138	**PROFIsafe module(s) for input byte**							
	8	7	6	5	4	3	2	1
DBB139								
DBB140	**PROFIsafe module(s) for output byte**							
	8	7	6	5	4	3	2	1
DBB141								
DBB142 - DBB149								
DBB150 - DBB157								
DBB158 - DBB188								

18.1.9.4 DB18, F_SENDDP sender

Table 18-65 DB18, F_SENDDP sender

DB18 /FBSIsl/	**SPL signals (PLC ←→ NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	1st F_SENDDP interface: FSDP[1]							
DBW190	ERR_REAC							
DBB192							SUBS_ON	ERROR
DBB193								
DBW194	DIAG							
DBW196	RETVAL14							
DBW198	RETVAL15							
	2nd F_SENDDP interface: FSDP[2]							
DBW200	ERR_REAC							
DBB202							SUBS_ON	ERROR
DBB203								
DBW204	DIAG							
DBW206	RETVAL14							
DBW208	RETVAL15							
	3rd F_SENDDP interface: FSDP[3]							
DBW210	ERR_REAC							
DBB212							SUBS_ON	ERROR
DBB213								
DBW214	DIAG							
DBW216	RETVAL14							
DBW218	RETVAL15							

18.1.9.5 DB18, F_SENDDP receiver

Table 18-66 DB18, F_SENDDP receiver

DB18 /FBSIsl/	**SPL signals (PLC ←→ NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	1st F_RECVDP interface: FRDP[1] (SUBS)							
DBB220	7	6	5	4	3	2	1	0
DBB221	15	14	13	12	11	10	9	8
DBB222	REAC							
DBB224								ACK_REI
DBB225					SEND MODE	ACK_REQ	SUBS_ON	ERROR

DB18 /FBSIsl/	SPL signals (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW226	DIAG							
DBW228	RETVAL14							
DBW230	RETVAL15							
	2nd F_RECVDP interface: FRDP[2] (SUBS)							
DBB232	7	6	5	4	3	2	1	0
DBB233	15	14	13	12	11	10	9	8
DBW234								
DBB236								ACK_REI
DBB237					SEND MODE	ACK_REQ	SUBS_ON	ERROR
DBW238	DIAG							
DBW240	RETVAL14							
DBW242	RETVAL15							
	3rd F_RECVDP interface: FRDP[3] (SUBS)							
DBB244	7	6	5	4	3	2	1	0
DBB245	15	14	13	12	11	10	9	8
DBW246	REAC							
DBB248								ACK_REI
DBB249					SEND MODE	ACK_REQ	SUBS_ON	ERROR
DBW250	DIAG							
DBW252	RETVAL14							
DBW254	RETVAL15							

18.1.9.6 DB18, SPL user data

Table 18-67 DB18, SPL user data

DB18 /FBSsl/	SPL signals (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD256	SPL_USER_DATA[0]							
DBD260	SPL_USER_DATA[1]							
DBD264	SPL_USER_DATA[2]							
DBD268	SPL_USER_DATA[3]							

18.1.9.7 DB18, data area / errors: Extended data area

Table 18-68 DB18, data area / errors: Extended data area

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Data area of the SPL inputs/outputs							
DBD272	SPL_DATA_HF.INSEP [65..96]							
DBD276	SPL_DATA_HF.INSEP [97..128]							
DBD280	SPL_DATA_HF.INSEP [129..160]							
DBD284	SPL_DATA_HF.INSEP [161..192]							
DBD288	SPL_DATA_HF.OUTSEP [65..96]							
DBD292	SPL_DATA_HF.OUTSEP [97..128]							
DBD296	SPL_DATA_HF.OUTSEP [129..160]							
DBD300	SPL_DATA_HF.OUTSEP [161..192]							
	Data area for user SPL							
DBD304	SPL_DATA_HF.INSIP [65..96]							
DBD308	SPL_DATA_HF.INSIP [97..128]							
DBD312	SPL_DATA_HF.INSIP [129..160]							
DBD316	SPL_DATA_HF.INSIP [161..192]							
DBD320	SPL_DATA_HF.OUTSIP [65..96]							
DBD324	SPL_DATA_HF.OUTSIP [97..128]							
DBD328	SPL_DATA_HF.OUTSIP [129..160]							
DBD332	SPL_DATA_HF.OUTSIP [161..192]							
DBD336	SPL_DATA_HF. MARKERSIP [65..96]							

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD340	SPL_DATA_HF. MARKERSIP [97..128]							
DBD344	SPL_DATA_HF. MARKERSIP [129..160]							
DBD348	SPL_DATA_HF. MARKERSIP [161..192]							
	Difference in signal level NC - PLC for diagnostics							
DBD352	SPL_DELTA_HF.INSEP [65..96]							
DBD356	SPL_DELTA_HF.INSEP [97..128]							
DBD360	SPL_DELTA_HF.INSEP [129..160]							
DBD364	SPL_DELTA_HF.INSEP [161..192]							
DBD368	SPL_DELTA_HF.OUTSEP [65..96]							
DBD372	SPL_DELTA_HF.OUTSEP [97..128]							
DBD376	SPL_DELTA_HF.OUTSEP [129..160]							
DBD380	SPL_DELTA_HF.OUTSEP [161..192]							
DBD384	SPL_DELTA_HF. INSIP [65..96]							
DBD388	SPL_DELTA_HF.INSIP [97..128]							
DBD392	SPL_DELTA_HF.INSIP [129..160]							
DBD396	SPL_DATA_HF.INSIP [161..192]							
DBD400	SPL_DELTA_HF.OUTSIP [65..96]							
DBD404	SPL_DELTA_HF.OUTSIP [97..128]							
DBD408	SPL_DELTA_HF.OUTSIP [129..160]							
DBD412	SPL_DELTA_HF.OUTSIP [161..192]							
DBD416	SPL_DELTA_HF.MARKERSIP [65..96]							

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD420	SPL_DELTA_HF. MARKERSIP [97..128]							
DBD424	SPL_DELTA_HF. MARKERSIP [129..160]							
DBD428	SPL_DELTA_HF. MARKERSIP [161..192]							

18.1.9.8 DB18, additional data areas: Extended data area

Table 18-69 DB18, additional data areas: Extended data area

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Data area of the single-channel inputs/outputs							
DBB432	PLCSIOUT_HF [33..40]							
von NC								
DBB433	PLCSIOUT_HF [41..48]							
von NC								
DBB434	PLCSIOUT_HF [49..56]							
von NC								
DBB435	PLCSIOUT_HF [57..64]							
von NC								
DBB436	PLCSIOUT_HF [65..72]							
von NC								
DBB437	PLCSIOUT_HF [73..80]							
von NC								
DBB438	PLCSIOUT_HF [81..88]							
von NC								
DBB439	PLCSIOUT_HF [89..96]							
von NC								
DBB440	PLCSIIN_HF [33..40]							
an NC								
DBB441	PLCSIIN_HF [41..48]							
an NC								
DBB442	PLCSIIN_HF [49..56]							
an NC								
DBB443	PLCSIIN_HF [57..64]							
an NC								

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB444	PLCSIIN_HF [65..72]							
an NC								
DBB445	PLCSIIN_HF [73..80]							
an NC								
DBB446	PLCSIIN_HF [81..88]							
an NC								
DBB447	PLCSIIN_HF [89..96]							
an NC								

18.1.9.9 DB18, F_SENDDP sender

Table 18-70 DB18, F_SENDDP sender

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	4th F_SENDDP interface: FSDP[4]							
DBW448	ERR_REAC							
DBB450							SUBS_ON	ERROR
DBB451								
DBW452	DIAG							
DBW454	RETVAL14							
DBW456	RETVAL15							
	5th F_SENDDP interface: FSDP[5]							
DBW458	ERR_REAC							
DBB460							SUBS_ON	ERROR
DBB461								
DBW462	DIAG							
DBW464	RETVAL14							
DBW466	RETVAL15							
	6th F_SENDDP interface: FSDP[6]							
DBW468	ERR_REAC							
DBB470							SUBS_ON	ERROR
DBB471								
DBW472	DIAG							
DBW474	RETVAL14							
DBW476	RETVAL15							
	7th F_SENDDP interface: FSDP[7]							
DBW478	ERR_REAC							
DBB480							SUBS_ON	ERROR

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB481								
DBW482	DIAG							
DBW484	RETVAL14							
DBW486	RETVAL15							
	8th F_SENDDP interface: FSDP[8]							
DBW488	ERR_REAC							
DBB490							SUBS_ON	ERROR
DBB491								
DBW492	DIAG							
DBW494	RETVAL14							
DBW496	RETVAL15							
	9th F_SENDDP interface: FSDP[9]							
DBW498	ERR_REAC							
DBB500							SUBS_ON	ERROR
DBB501								
DBW502	DIAG							
DBW504	RETVAL14							
DBW506	RETVAL15							
	10th F_SENDDP interface: FSDP[10]							
DBW508	ERR_REAC							
DBB510							SUBS_ON	ERROR
DBB511								
DBW512	DIAG							
DBW514	RETVAL14							
DBW516	RETVAL15							
	11th F_SENDDP interface: FSDP[11]							
DBW518	ERR_REAC							
DBB520							SUBS_ON	ERROR
DBB521								
DBW522	DIAG							
DBW524	RETVAL14							
DBW526	RETVAL15							
	12th F_SENDDP interface: FSDP[12]							
DBW528	ERR_REAC							
DBB530							SUBS_ON	ERROR
DBB531								
DBW532	DIAG							
DBW534	RETVAL14							
DBW536	RETVAL15							
	13th F_SENDDP interface: FSDP[13]							
DBW538	ERR_REAC							

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB540							SUBS_ON	ERROR
DBB541								
DBW542	DIAG							
DBW544	RETVAL14							
DBW546	RETVAL15							
	14th F_SENDDP interface: FSDP[14]							
DBW548	ERR_REAC							
DBB550							SUBS_ON	ERROR
DBB551								
DBW552	DIAG							
DBW554	RETVAL14							
DBW556	RETVAL15							
	15th F_SENDDP interface: FSDP[15]							
DBW558	ERR_REAC							
DBB560							SUBS_ON	ERROR
DBB561								
DBW562	DIAG							
DBW564	RETVAL14							
DBW566	RETVAL15							
	16th F_SENDDP interface: FSDP[16]							
DBW568	ERR_REAC							
DBB570							SUBS_ON	ERROR
DBB571								
DBW572	DIAG							
DBW574	RETVAL14							
DBW576	RETVAL15							

18.1.9.10 DB18, F_RECDP receiver

Table 18-71 DB18, F_SENDDP receiver

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	4th F_RECVDP interface: FRDP_HF[4]							
	(SUBS)							
DBB578	7	6	5	4	3	2	1	0
DBB579	15	14	13	12	11	10	9	8
DBW580	ERR_REAC							
DBB582								ACK_REI

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB583					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW584	DIAG							
DBW586	RETVAL14							
DBW588	RETVAL15							
	5th F_RECVDP interface: FRDP_HF[5]							
	(SUBS)							
DBB590	7	6	5	4	3	2	1	0
DBB591	15	14	13	12	11	10	9	8
DBW592	ERR_REAC							
DBB594								ACK_REI
DBB595					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW596	DIAG							
DBW598	RETVAL14							
DBW600	RETVAL15							
	6th F_RECVDP interface: FRDP_HF[6]							
	(SUBS)							
DBB602	7	6	5	4	3	2	1	0
DBB603	15	14	13	12	11	10	9	8
DBW604	ERR_REAC							
DBB606								ACK_REI
DBB607					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW608	DIAG							
DBW610	RETVAL14							
DBW612	RETVAL15							
	7th F_RECVDP interface: FRDP_HF[7]							
	(SUBS)							
DBB614	7	6	5	4	3	2	1	0
DBB615	15	14	13	12	11	10	9	8
DBW616	ERR_REAC							
DBB618								ACK_REI
DBB619					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW620	DIAG							
DBW622	RETVAL14							
DBW624	RETVAL15							
	8th F_RECVDP interface: FRDP_HF[8]							
	(SUBS)							
DBB626	7	6	5	4	3	2	1	0
DBB627	15	14	13	12	11	10	9	8

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW628	ERR_REAC							
DBB630								ACK_REI
DBB631					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW632	DIAG							
DBW634	RETVAL14							
DBW636	RETVAL15							
	9th F_RECVDP interface: FRDP_HF[9]							
	(SUBS)							
DBB638	7	6	5	4	3	2	1	0
DBB639	15	14	13	12	11	10	9	8
DBW640	ERR_REAC							
DBB642								ACK_REI
DBB643					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW644	DIAG							
DBW646	RETVAL14							
DBW648	RETVAL15							
	10th F_RECVDP interface: FRDP_HF[10]							
DBB650	(SUBS)							
	7	6	5	4	3	2	1	0
DBB651	15	14	13	12	11	10	9	8
DBW652	ERR_REAC							
DBB654								ACK_REI
DBB655					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW656	DIAG							
DBW658	RETVAL14							
DBW660	RETVAL15							
	11th F_RECVDP interface: FRDP_HF[11]							
DBB662	(SUBS)							
	7	6	5	4	3	2	1	0
DBB663	15	14	13	12	11	10	9	8
DBW664	ERR_REAC							
DBB666								ACK_REI
DBB667					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW668	DIAG							
DBW670	RETVAL14							
DBW672	RETVAL15							

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	12th F_RECVDP interface: FRDP_HF[12]							
	(SUBS)							
DBB674	7	6	5	4	3	2	1	0
DBB675	15	14	13	12	11	10	9	8
DBW676	ERR_REAC							
DBB678								ACK_REI
DBB679					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW680	DIAG							
DBW682	RETVAL14							
DBW684	RETVAL15							
	13th F_RECVDP interface: FRDP_HF[13]							
	(SUBS)							
DBB686	7	6	5	4	3	2	1	0
DBB687	15	14	13	12	11	10	9	8
DBW688	ERR_REAC							
DBB690								ACK_REI
DBB691					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW692	DIAG							
DBW694	RETVAL14							
DBW696	RETVAL15							
	14th F_RECVDP interface: FRDP_HF[14]							
	(SUBS)							
DBB698	7	6	5	4	3	2	1	0
DBB699	15	14	13	12	11	10	9	8
DBW700	ERR_REAC							
DBB702								ACK_REI
DBB703					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW704	DIAG							
DBW706	RETVAL14							
DBW708	RETVAL15							
	15th F_RECVDP interface: FRDP_HF[15]							
	(SUBS)							
DBB710	7	6	5	4	3	2	1	0
DBB711	15	14	13	12	11	10	9	8
DBW712	ERR_REAC							
DBB714								ACK_REI
DBB715					SEND-MODE	ACK_REQ	SUBS_ON	ERROR

DB18 /FBSIsl/	Signals for the safety SPL (PLC ⟷ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW716	DIAG							
DBW718	RETVAL14							
DBW720	RETVAL15							
	16th F_RECVDP interface: FRDP_HF[16]							
	(SUBS)							
DBB722	7	6	5	4	3	2	1	0
DBB723	15	14	13	12	11	10	9	8
DBW724	ERR_REAC							
DBB726								ACK_REI
DBB727					SEND-MODE	ACK_REQ	SUBS_ON	ERROR
DBW728	DIAG							
DBW730	RETVAL14							
DBW732	RETVAL15							

18.1.10 Control/status signals to/from operator panel (OP)

18.1.10.1 DB19, signals to the operator panel (OP)

Table 18-72 DB19, signals to the operator panel (OP)

DB19	Signals from the operator panel (OP) (PLC → operating software)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	**Interface of 1st Control (DBB0 - DBB49)**							
	Actual value in the WCS, (1) / MCS (0) see also (Page 489)	Save to action log	HMI Advanced: Shutdown	Delete recall alarms see also (Page 489)	Delete cancel alarms see also (Page 488)	Key lock see also (Page 488)	Darken screen see also (Page 487)	Brighten screen see also (Page 487)
DBB1							Rights for the external viewer	External viewer
DBB2								
DBB4								
DBB6	Analog spindle 1: Utilization as a percentage see also (Page 489)							
DBB7	Analog spindle 2: Utilization as a percentage see also (Page 489)							

DB19	**Signals from the operator panel (OP) (PLC → operating software)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB8	Channel number of the machine control panel to the control  see also (Page 490)							
DBB9	Reserved selection					Autotool measure	OEM2	OEM1
DBB10	Hardkeys  see also (Page 490)							
DBB11	Reserved							
DBB12								
DBB13	Part program			Reserved				Disable teach in transfer /IHsl-IM9/
	Selection  see also (Page 490)	Load  see also (Page 490)	Unload  see also (Page 490)					
DBB14	Active (0) / passive (1) file system  see also (Page 491)	V24 active file system: Index of the file to be transferred from the standard list. V24 passive file system: Number of the control file for user file names.  see also (Page 491)						
DBB15	V24 active file system: Index which specifies the axis, channel or TO No. V24 passive file system: Index of the file to be transferred from the user list.  see also (Page 491)							
DBB16	Always 1 	Program selection from the PLC: Index of the program list  see also (Page 491)						
DBB17	Program selection from the PLC: Program index in the program list  see also (Page 492)							
DBB18								
DBB19	Reserved (message counter)							

18.1.10.2 DB19, signals from the operator panel (OP)

Table 18-73 DB19, signals from the operator panel (OP)

DB19	**Signals from the operator panel (OP)** **(Operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB20	Switch over MCS/WCS see also (Page 494)	Simulation active see also (Page 494)	Language 2 switched over /IHsl-IM9/	Recall alarm deleted see also (Page 493)	Cancel alarm deleted see also (Page 493)	Cancel key actuated /FB-A2/	Screen is dark see also (Page 493)	
DBB21	Current number of the active operating range /FB-P3/; /FB-A2/							
DBB22	Current channel number see also (Page 494)							
DBB23						Control 1		
						Screen change active	Data transfer active	Keyboard has been used
DBW24	Current screen number see also (Page 494)							
DBB25								
DBB26	**Program selection from the PLC**: Status signals							
	Selection see also (Page 497)	Load see also (Page 496)	Unload see also (Page 496)		Active see also (Page 495)	Error see also (Page 495)	Job completed see also (Page 495)	Reserved
DBB27	**Program selection from the PLC**: Error code see also (Page 497)							
DBW28	Screen number for "Supplement user interface" /IHsl-BE2/							
DBB30 PLC → OP	**Screen selection from the PLC:** Control signals							
							Screen deselection	Screen selection
DBB31 Operating software → PLC	**Screen selection from the PLC:** Status signals							
	Inactive			Error, screen selection not possible	Screen is deselected	Screen active	Screen is selected	Screen selection accepted
DBB32 PLC → operating software	Busy function see also (Page 499)	Strobe function see also (Page 499)	Function selection No. from the PLC see also (Page 498)					

DB19	**Signals from the operator panel (OP) (Operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB33 PLC → operating software	Parameter 1 for function selection No. (function selection from DBB32) see also (Page 499)							
DBB34 PLC → operating software	Parameter 2 for function selection No. (function selection from DBB32)							
DBB35 PLC → operating software	Parameter 3 for function selection No. (function selection from DBB32)							
DBB36 Operating software → PLC	Error code for function selection No. (function selection from DBB32) see also (Page 500)							
DBB37 Operating software → PLC	Parameter 1 for function selection No. (function selection from DBB48)							
DBB38 Operating software → PLC	Parameter 2 for function selection No. (function selection from DBB48)							
DBB39 Operating software → PLC	Parameter 3 for function selection No. (function selection from DBB48)							
DBB40 - DBB47	Reserved							
DBB48 Operating software → PLC	PLC busy Function	Operating software strobe Function	Function selection No. from the operating software					
DBB49 PLC → operating software	Error code for function selection No. (function selection from DBB48)							
DBB50 - DBB99	**Interface of 2nd Control** (assignment the same as DBB0 - DBB49)							
DBB100	**Switchover interface to the operating software** **Call waiting interface (operating software announces itself to the NC)** ONL_REQUEST (online request from operating software) /FB-B3/ Operating software writes its client identification as online request (bits 8-15: bus type, bits 0-7: bus address)							

DB19	Signals from the operator panel (OP) (Operating software → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB102	ONL_CONFIRM (acknowledgment from PLC after online request) /FB-B3/ PLC writes as acknowledgment the operating software client identification (bus type, bus address; such as DBB100)							
DBB104	PAR_CLIENT_IDENT /FB-B3/ Operating software writes its client identification (bus type, bus address; such as DBB100).							
DBB106	PAR_MMC_TYP /FB-B3/ Type of operating software acc. to NETNAMES.INI: Main/secondary operator panel / server /...							
DBB107	PAR_MCP_ADR /FB-B3/ Operating software writes address of the MCP to be activated; 255, if no MCP activation							
DBB108	PAR_STATUS /FB-B3/ PLC writes the online enable for the operating software							
DBB109	PAR_Z_INFO /FB-B3/ PLC writes additional info on the status							
DBB110	M_TO_N_ALIVE Sign-of-life from the PLC to the operating software using the M to N block							
DBB112	Res. Bus type MCP							
DBB113	ParOpKeyAdr Direct key index call waiting interface							
DBB114	ParTcuIndex TCU index call waiting interface							
DBB115	ParHt2Index Ht2 index logon interface							
DBB116	Direct key address of 1st Online interface							
DBB117	Direct key address of 2nd Online interface							
DBB118	TCU index of 1st Online interface							
DBB119	TCU index of 2nd Online interface							
DBB120	**Online interface OP 1 (user)** MMC1_CLIENT_IDENT /FB-B3/ PLC writes PAR_CLIENT_IDENT to MMCx_CLIENT_IDENT, if operating software goes online.							
DBB122	MMC1_TYP /FB-B3/ PLC writes PAR_MMC_TYP to MMCx_TYP, if operating software goes online.							
DBB123	MMC1_MCP_ADR /FB-B3/ PLC writes PAR_MCP_ADR to MMCx_MCP_ADR, if operating software goes online.							
DBB124	MMC1_STATUS /FB-B3/ Connection state, operating software and PLC write alternating their requests/acknowledgments.							
DBB125	MMC1_Z_INFO /FB-B3/ Additional information, connection state (pos./neg. acknowledgment, error messages, etc.)							
DBB126	Reserved	TCU1_ SHIFT_ LOCK	MMC1_ CHANGE_ DENIED /FB-B3/	MMC1_ ACTIVE_ CHANGED /FB2B3/	MMC1_ ACTIVE_ PERM /FB-B3/	MMC1_ ACTIVE_ REQ /FB-B3/	MMC1_ MCP_ SHIFT- LOCK /FB-B3/	MMC1_ SHIFT_ LOCK /FB-B3/

DB19	**Signals from the operator panel (OP)** **(Operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB127	Reserved bus type MCP							
DBB128 - DBB129	Reserved Transline (Transline DB number)							
DBB130	**Online interface OP 2 (user)** MMC2_CLIENT_IDENT /FB-B3/ PLC writes PAR_CLIENT_IDENT to MMCx_CLIENT_IDENT, if operating software goes online.							
DBB132	MMC2_TYP /FB-B3/ PLC writes PAR_MMC_TYP to MMCx_TYP, if operating software goes online.							
DBB133	MMC2_MCP_ADR /FB-B3/ PLC writes PAR_MCP_ADR to MMCx_MCP_ADR, if operating software goes online.							
DBB134	MMC2_STATUS /FB-B3/ Connection state, operating software and PLC write alternating their requests/acknowledgments.							
DBB135	MMC2_Z_INFO /FB-B3/ Additional information, connection state (pos./neg. acknowledgment, error messages, etc.)							
DBB136	Reserved	TCU2_ SHIFT_ LOCK	MMC2_ CHANGE_ DENIED /FB-B3/	MMC2_ ACTIVE_ CHANGED /FB-B3/	MMC2_ ACTIVE_ PERM /FB-B3/	MMC2_ ACTIVE_ REQ /FB-B3/	MMC2_ MCP_ SHIFT_ LOCK /FB-B3/	MMC2_ SHIFT_ LOCK /FB-B3/
DBB137	Reserved bus type MCP							
DBB138 - DBB139	Reserved Transline (Transline DB number)							
DBB140 - DBB197	Assignment of the PLC for transfer parameters These data blocks are reserved for the "Tool Ident Connection" option. /FBWsl/							
DBB198 - DBB249	Assignment of the PLC for return values These data blocks are reserved for the "Tool Ident Connection" option. /FBWsl/							
DBB250 - DBB255	Function call of the PLC interface These data blocks are reserved for the "Tool Ident Connection" option. /FBWsl/							
DBB256 - DBB267	Commands for Paramtm.exe							
DBB268	Traffic light status							
DBD270 - DBD394	Counter [1...32]							
DBB398	Handwheel number for simulation override							
DBW400	Simulation override							
DBW402	Simulation state							

18.1.10.3 DB19, sidescreen - interface for the MCP function

Table 18-74 DB19, sidescreen - interface for the MCP function

DB19	Signals to/from the operator panel Signal from the operating software → PLC							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB404	Customer keys							
	8	7	6	5	4	3	2	1
DBB405	Customer keys							
	16	15	14	13	12	11	10	9
DBB406	Customer keys							
	24	23	22	21	20	19	18	17
DBB407	Customer keys							
	32	31	30	29	28	27	26	25
DBB408	Customer keys							
	40	39	38	37	36	35	34	33
DBB409	Customer keys							
	48	47	46	45	44	43	42	41
DBB410	Customer keys							
	56	55	54	53	52	51	50	49
DBB411	Customer keys							
	64	63	62	61	60	59	58	57

18.1.10.4 DB19, sidescreen - interface for the MCP function

Table 18-75 DB19, sidescreen - interface for the MCP function signals to/from the operator panel

DB19	Signals to/from the operator panel [r/w] Signal from PLC → operating software							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB412	Customer LED							
	8	7	6	5	4	3	2	1
DBB413	Customer LED							
	16	15	14	13	12	11	10	9
DBB414	Customer LED							
	24	23	22	21	20	19	18	17
DBB415	Customer LED							
	32	31	30	29	28	27	26	25
DBB416	Customer LED							
	40	39	38	37	36	35	34	33
DBB417	Customer LED							
	48	47	46	45	44	43	42	41

DB19	**Signals to/from the operator panel [r/w]** **Signal from PLC → operating software**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB418	Customer LED							
	56	55	54	53	52	51	50	49
DBB419	Customer LED							
	64	63	62	61	60	59	58	57
DBB420	Customer key deactivated							
	8	7	6	5	4	3	2	1
DBB421	Customer key deactivated							
	16	15	14	13	12	11	10	9
DBB422	Customer key deactivated							
	24	23	22	21	20	19	18	17
DBB423	Customer key deactivated							
	32	31	30	29	28	27	26	25
DBB424	Customer key deactivated							
	40	39	38	37	36	35	34	33
DBB425	Customer key deactivated							
	48	47	46	45	44	43	42	41
DBB426	Customer key deactivated							
	56	55	54	53	52	51	50	49
DBB427	Customer key deactivated							
	64	63	62	61	60	59	58	57

18.1.11 Defining PLC alarms

18.1.11.1 DB20, NC machine data

Table 18-76 DB20, NC machine data

DB20	**NC machine data (PLC → user)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW0	INT values							
DBW								
DBW	INT values							
DBB	Bit arrays							
DBB								

DB20	NC machine data (PLC → user)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB								
DBB	Bit arrays							
DBD	REAL values							
DBD								
DBD	REAL values							

Note

The start and end addresses of the PLC machine data areas are dependent on the particular length data of the partial areas. The range of INTEGER values always starts with data byte 0. The upper limit is defined by the associated length data. The range of the bit arrays starts after the range of the INTEGER values at the next even address. The range of REAL values starts after the range of the bit arrays at the next even address.

18.1.12 Channel-specific signals

18.1.12.1 DB21 - DB30, control signals to the channel (1)

DB21 - DB30	Signals to the channel (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0		Activate dry run feedrate see also (Page 502)	Activate M01 see also (Page 502)	Activate single block see also (Page 501)	Activate DRF see also (Page 501)	RESU: Start retrace support see also (Page 501)	RESU: Backward/ forward see also (Page 500)	
DBB1	Activate program test see also (Page 506)	PLC action completed see also (Page 505)	CLC correction see also (Page 505)	CLC stop see also (Page 504)	Time monitoring active (tool management) see also (Page 504)	Disable all synchronized actions /FBSY/	Enable protection zones /FB-A5/	Activate referencing see also (Page 503)

DB21 - DB30	Signals to the channel (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB2	Activate skip block see also (Page 506)							
	/7	/6	/5	/4	/3	/2	/1	/0
DBB3	Nibbling and punching							
			Manual stroke initiation see also (Page 509)	Stroke inoperative see also (Page 508)	Delayed stroke see also (Page 508)	Stroke suppression see also (Page 508)	Manual stroke initiation see also (Page 507)	Stroke enable see also (Page 507)
DBB4	Path feedrate override see also (Page 509)							
	H	G	F	E	D	C	B	A
DBB5	Path rapid traverse override see also (Page 511)							
	H	G	F	E	D	C	B	A
DBB6	Path feedrate override active see also (Page 516)	Path rapid traverse override active see also (Page 515)		Program level abort see also (Page 515)	Delete UP number of passes	Delete distance-to-go see also (Page 514)	Read-in disable see also (Page 514)	Feedrate disable see also (Page 513)
DBB7	Reset see also (Page 519)	Activate configured STOP see also (Page 519)	Deactivate start disable	NC Stop axes plus spindle see also (Page 518)	NC stop see also (Page 518)	NC Stop at block limit see also (Page 517)	NC start see also (Page 517)	NC start disable see also (Page 516)
DBB8	Activate machine-related protection area see also (Page 520)							
	8	7	6	5	4	3	2	1
DBB9	Activate machine-related protection area see also (Page 520)							
							10	9
DBB10	Activate channel-specific protection area see also (Page 520)							
	8	7	6	5	4	3	2	1
DBB11	Activate channel-specific protection area see also (Page 520)							
							10	9

Note

- Feedrate override active: Even if the feedrate override is not active (= 100%), the 0% position is still effective.
- Feedrate override: Either 31 positions (Gray code) with 31 MD for % evaluation, or 0% to 200% corresponding to the dual value in the byte (201 to 255 = max. 200%)
- Rapid traverse override: Either 31 positions (Gray code) with 31 MD for % evaluation, or 0% to 100% corresponding to the dual value in the byte (101 to 255 = max. 100%)
- Single block: Select the version using "Write variable"
- Delete distance-to-go: Is only active for path axes and not for positioning axes

18.1.12.2 DB21 - DB30, control signals to the geometry axes

Table 18-77 DB21 - DB30, control signals to the geometry axes

DB21 - DB30	**Signals to the channel (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Geometry axis 1							
DBB12	Traversing keys see also (Page 523)		Rapid traverse override see also (Page 523)	Traversing key lock see also (Page 522)	Feedrate stop see also (Page 522)	Activate handwheel see also (Page 521)		
	Plus	Minus				C	B	A
DBB13		Requested machine function see also (Page 525)						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB14	OEM signals							
DBB15								Invert handwheel direction of rotation see also (Page 526)
	Geometry axis 2							

DB21 - DB30	**Signals to the channel (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB16	Traversing keys see also (Page 527)		Rapid traverse override see also (Page 527)	Traversing key lock see also (Page 527)	Feedrate stop see also (Page 522)	Activate handwheel see also (Page 526)		
	Plus	Minus				C	B	A
DBB17		Requested machine function see also (Page 527)						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB18	OEM signals							
DBB19								Invert handwheel direction of rotation see also (Page 527)
	Geometry axis 3							
DBB20	Traversing keys see also (Page 528)		Rapid traverse override see also (Page 528)	Traversing key lock see also (Page 527)	Feedrate stop see also (Page 522)	Activate handwheel see also (Page 527)		
	Plus	Minus				C	B	A
DBB21		Requested machine function see also (Page 528)						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB22	OEM signals							
DBB23								Invert handwheel direction of rotation see also (Page 528)

Note

The NC only evaluates the machine function signals if the signal DB10 DBX57.0 (Page 340)"INC inputs in mode group area active" is not set.

18.1.12.3 DB21 - DB30, HMI signals to channel / OEM signals from/to channel

Table 18-78 DB21 - DB30, control signals from the operating software to the PLC, PLC to the NC and status signal from channel to the PLC

DB21 - DB30	**Signals from the channel/PLC/operating software (operating software → PLC, PLC → NC, NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB24 Operating software → PLC		Dry run feedrate selected see also (Page 530)	M01 selected see also (Page 530)	M01 associated with NC selected see also (Page 529)	DRF selected see also (Page 529)		Activate configured STOP see also (Page 528)	
DBB25 Operating software → PLC	Program test selected /FB-K1/			REPOS mode change /FB-K1/	Feedrate override selected for rapid traverse see also (Page 531)	REPOS mode /FB-K1/ C	B	A
DBB26 Operating software → PLC	Skip block requested, level/x /FB-K1/ /7	/6	/5	/4	/3	/2	/1	/0
DBB27 Operating software → PLC							Skip block requested, level/x /9	/8
DBB28 PLC → NC	OEM signals: Requirement							
DBB29 PLC → NC	Tool do not lock see also (Page 535)	Deactivate wear monitoring see also (Page 534)	Deactivate workpiece counter see also (Page 534)	Activate PTP traversal see also (Page 534)	Activate fixed feedrate see also (Page 533) 4	3	2	1

DB21 - DB30	**Signals from the channel/PLC/operating software (operating software → PLC, PLC → NC, NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB30 PLC → NC	No tool change commands	Jog circles see also (Page 537)	Activate NC-associated M0/1 see also (Page 537)	Contour handwheel simulation, negative direction see also (Page 537)	Contour handwheel simulation on see also (Page 536)	Activate contour handwheel see also (Page 535)		
						C	B	A
DBB31 PLC → NC	Activate skip block		Invert contour handwheel direction of rotation see also (Page 539)	REPOS mode change see also (Page 539)		REPOS mode see also (Page 538)		
	/9	/8				C	B	A
DBB32 NC → PLC		Last action block active see also (Page 541)	M00/M01 active see also (Page 541)	Approach block active see also (Page 540)	Action block active see also (Page 540)	RESU: Retrace support active see also (Page 540)	RESU: Retrace mode active see also (Page 539)	Execution from external active
DBB33 NC → PLC	Program test active see also (Page 544)	Transformation active see also (Page 544)	M02/M30 active see also (Page 543)	Block search active see also (Page 543)	Handwheel override active see also (Page 542)	Revolutional feedrate active /FB-V1/	Orientable tool carrier active	Referencing active see also (Page 542)
DBB34 NC → PLC	OEM signals: Feedback signal							
DBB35 NC → PLC	Channel state			Program status				
	Reset see also (Page 550)	Interrupted see also (Page 549)	Active see also (Page 549)	Canceled see also (Page 548)	Interrupted see also (Page 547)	Stopped see also (Page 547)	Waiting see also (Page 546)	Running see also (Page 545)
DBB36 NC → PLC	NC alarm with machining stop is active see also (Page 552)	Channel-specific NC alarm is active see also (Page 552)	Channel is ready see also (Page 551)	Interrupt handling active see also (Page 551)	All axes stationary see also (Page 551)	All axes that have to be referenced are referenced see also (Page 550)		

DB21 - DB30	Signals from the channel/PLC/operating software (operating software → PLC, PLC → NC, NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB37 NC → PLC	"Stop at the end of block for single block (SBL)" is suppressed see also (Page 555)	Read-in disable is ignored see also (Page 555)	CLC stopped Upper limit see also (Page 554)	CLC stopped Lower limit see also (Page 554)	CLC active see also (Page 553)	Contour handwheel active see also (Page 552)		
						C	B	A
DBB38 NC → PLC	Nibbling and punching /FB-N4/							
							Manual stroke initiation: Acknowledgment see also (Page 556)	Stroke initiation active see also (Page 556)
DBB39 NC → PLC	Stop at end of block due to configured stop see also (Page 558)	Configured stop is activated see also (Page 558)	Contour handwheel direction of rotation inverted active see also (Page 557)	Stop at end of block due to single block see also (Page 557)			NC alarm with program stop see also (Page 557)	Protection zone monitoring not guaranteed /FB-A5/

18.1.12.4 DB21 - DB30, control signals from the geometry axes

Table 18-79 DB21 - DB30, control signals from the geometry axes

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Geometry axis 1							
DBB40	Travel command see also (Page 560)		Traversing requests see also (Page 559)			Handwheel active see also (Page 558)		
	Plus	Minus	Plus	Minus		C	B	A
DBB41		Active machine function see also (Page 561)						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB42	OEM signals							

DB21 - DB30	**Signals from the channel (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB43								Handwheel direction of rotation inverted see also (Page 561)
DBB44 Operating software → PLC								
	Geometry axis 2							
DBB46	Travel command see also (Page 562)		Traversing requests see also (Page 562)			Handwheel active see also (Page 562)		
	Plus	Minus	Plus	Minus		C	B	A
DBB47		Active machine function see also (Page 562)						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB48	OEM signals							
DBB49								Handwheel direction of rotation inverted see also (Page 562)
DBB50 Operating software → PLC								
	Geometry axis 3							
DBB52	Travel command see also (Page 563)		Traversing requests see also (Page 563)			Handwheel active see also (Page 563)		
						C	B	A
DBB53		Active machine function see also (Page 563)						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB54	OEM signals							

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB55								Handwheel direction of rotation inverted see also (Page 563)
DBB56								
Operating software → PLC								

18.1.12.5 DB21 - DB30, change signals for auxiliary function transfer from the channel

Table 18-80 DB21 - DB30, change signals for auxiliary function transfer from the channel

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB58				change see also (Page 563)				
				M fct. 5	M fct. 4	M fct. 3	M fct. 2	M fct. 1
DBB59				Not decoded see also (Page 564)				
				M fct. 5	M fct. 4	M fct. 3	M fct. 2	M fct. 1
DBB60		Quick see also (Page 564)				change see also (Page 563)		
		S fct. 3	S fct. 2	S fct. 1		S fct. 3	S fct. 2	S fct. 1
DBB61		Quick see also (Page 564)				change see also (Page 563)		
		T fct. 3	T fct. 2	T fct. 1		T fct. 3	T fct. 2	T fct. 1
DBB62		Quick see also (Page 564)				change see also (Page 563)		
		D fct. 3	D fct. 2	D fct. 1		D fct. 3	D fct. 2	D fct. 1
DBB63				DL fct. Quick				DL fct.change
DBB64		Quick see also (Page 564)				change see also (Page 563)		
		H fct. 3	H fct. 2	H fct. 1		H fct. 3	H fct. 2	H fct. 1
DBB65			change see also (Page 563)					
			F fct. 6	F fct. 5	F fct. 4	F fct. 3	F fct. 2	F fct. 1

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB66				Quick see also (Page 564)				
				M fct. 5	M fct. 4	M fct. 3	M fct. 2	M fct. 1
DBB67			Quick see also (Page 564)					
			F fct. 6	F fct. 5	F fct. 4	F fct. 3	F fct. 2	F fct. 1

Note

- For 10-decade T numbers, only the signal DBB61, DBX0 "T fct.1 change" is available.
- For 5-decade D numbers, only the signal DBB62, DBX0 "D fct.1 change" is available.

18.1.12.6 DB21 - DB30, transferred M and S functions:

Table 18-81 DB21 - DB30, transferred M and S functions:

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW68	Extended address, M function 1 (16-bit INT) see also (Page 564)							
DBD70	M function 1 (DINT) see also (Page 564)							
DBW74	Extended address, M function 2 (16-bit INT) see also (Page 564)							
DBD76	M function 2 (DINT) see also (Page 564)							
DBW80	Extended address, M function 3 (16-bit INT) see also (Page 564)							
DBD82	M function 3 (DINT) see also (Page 564)							
DBW86	Extended address, M function 4 (16-bit INT) see also (Page 564)							
DBD88	M function 4 (DINT) see also (Page 564)							
DBW92	Extended address, M function 5 (16-bit INT) see also (Page 564)							
DBD94	M function 5 (DINT) see also (Page 564)							
DBW98	Extended address, S function 1 (16-bit INT) see also (Page 565)							

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBD100	S function 1 (REAL format) see also (Page 565)							
DBW104	Extended address, S function 2 (16-bit INT) see also (Page 565)							
DBD106	S function 2 (REAL format) see also (Page 565)							
DBW110	Extended address, S function 3 (16-bit INT) see also (Page 565)							
DBD112	S function 3 (REAL format) see also (Page 565)							

Note

M functions are programmed in the part program in the INTEGER format (8 decades plus leading sign).

18.1.12.7 DB21 - DB30, transferred T/D/DL functions

Table 18-82 DB21 - DB30, transferred T/D/DL functions

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW116	Extended address, T function 1 (16-bit INT)							
DBW118 DBD118	T function 1 (dual) / For 8-decade T No., DBD118 is used as T function 1 (32-bit INT) (see note below) see also (Page 565)							
DBW120	Extended address, T function 2 (16-bit INT)							
DBW122	T function 2 (INT)							
DBW124	Extended address, T function 3 (16-bit INT)							
DBW126	T function 3 (INT)							
DBB128	Extended address D function 1 (8-bit INT)							
DBB129	D function 1 (binary) see also (Page 566)							
DBW130 DBB130	For 5-decade D No., DBW130 is used as D function 1 (16-bit INT) Extended address D function 2 (8-bit INT)							
DBB131	D function 2 (8-bit INT)							
DBB132	Extended address D function 3 (8-bit INT)							
DBB133	D function 3 (8-bit INT)							
DBW134	Extended address DL function (16-bit INT)							
DBD136	DL function (REAL)							

Note

- Programmed T functions are not output to the PLC when tool management is active.
- 8 decade T numbers are only available under DBD118 "T function 1".
- Programmed D functions with names (e.g. D=CUTTING EDGE_1) cannot be output to the PLC in ASCII format.
- 5-decade D numbers are only available as DBW130 "D function 1".
- The REAL format corresponds to the floating-point representation in STEP 7 (24-bit mantissa and 8-bit exponent). This floating point format supplies a maximum of 7 valid places.

18.1.12.8 DB21 - DB30, transferred H/F functions

Table 18-83 DB21 - DB30, transferred H/F functions

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW140	Extended address, H function 1 (16-bit INT) see also (Page 566)							
DBD142	H function 1 (REAL or DINT) see also (Page 566)							
DBW146	Extended address, H function 2 (16-bit INT) see also (Page 566)							
DBD148	H function 2 (REAL or DINT) see also (Page 566)							
DBW152	Extended address, H function 3 (16-bit INT) see also (Page 566)							
DBD154	H function 3 (REAL or DINT) see also (Page 566)							
DBW158	Extended address F function 1 (16-bit INT) see also (Page 566)							
DBD160	F function 1 (REAL format) see also (Page 566)							
DBW164	Extended address F function 2 (16-bit INT) see also (Page 566)							
DBD166	F function 2 (REAL format) see also (Page 566)							
DBW170	Extended address F function 3 (16-bit INT) see also (Page 566)							
DBD172	F function 3 (REAL format) see also (Page 566)							
DBW176	Extended address F function 4 (16-bit INT) see also (Page 566)							

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD178	F function 4 (REAL format) see also (Page 566)							
DBW182	Extended address F function 5 (16-bit INT) see also (Page 566)							
DBD184	F function 5 (REAL format) see also (Page 566)							
DBW188	Extended address F function 6 (16-bit INT) see also (Page 566)							
DBD190	F function 6 (REAL format) see also (Page 566)							

Note

- F functions are programmed in the part program in the REAL data format.
- The extended address of the F function contains an identifier with the following meaning:
  - 0: Path feedrate
  - 1 – 31: Machine axis number for feedrate for positioning axes
- The data type of the H function depends on machine data: MD22110 $MC_AUXFU_H_TYPE_INT

18.1.12.9 DB21 - DB30, decoded M signals

Table 18-84 DB21 - DB30, decoded M signals

DB21 - DB30	Signals from the channel (M0 - M99) (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Dynamic M functions see also (Page 567)							
DBB194								
	M07	M06	M05 #	M04 #	M03 #	M02	M01	M00
DBB195								
	M15	M14	M13	M12	M11	M10	M09	M08
DBB196								
	M23	M22	M21	M20	M19	M18	M17	M16
DBB197								
	M31	M30	M29	M28	M27	M26	M25	M24
DBB198								
	M39	M38	M37	M36	M35	M34	M33	M32

DB21 - DB30	Signals from the channel (M0 - M99) (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB199								
	M47	M46	M45	M44	M43	M42	M41	M40
DBB200								
	M55	M54	M53	M52	M51	M50	M49	M48
DBB201								
	M63	M62	M61	M60	M59	M58	M57	M56
DBB202								
	M71	M70 #	M69	M68	M67	M66	M65	M64
DBB203								
	M79	M78	M77	M76	M75	M74	M73	M72
DBB204								
	M87	M86	M85	M84	M83	M82	M81	M80
DBB205								
	M95	M94	M93	M92	M91	M90	M89	M88
DBB206								
					M99	M98	M97	M96
DBB207								

Note

- #: the M function is not displayed here, if a spindle is parameterized in the channel. In this case, the M function is displayed as extended M function under DB21,... DBB68 ff. and axial under DB31,... DBB86 ff. displayed.
- **Dynamic** M functions (M00 - M99) are decoded by the basic PLC program.
  Static M functions must be generated in the PLC user program from dynamic M functions.

18.1.12.10 DB21 - DB30, active G functions

Table 18-85 DB21 - DB30, active G functions

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB208	Number of the active G function of G function group 1 (8-bit INT) see also (Page 567)							
DBB209	Number of the active G function of G function group 2 (8-bit INT) see also (Page 567)							
DBB210	Number of the active G function of G function group 3 (8-bit INT) see also (Page 567)							

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB211	Number of the active G function of G function group 4 (8-bit INT) see also (Page 567)							
DBB212	Number of the active G function of G function group 5 (8-bit INT) see also (Page 567)							
DBB213	Number of the active G function of G function group 6 (8-bit INT) see also (Page 567)							
DBB214	Number of the active G function of G function group 7 (8-bit INT) see also (Page 567)							
DBB215	Number of the active G function of G function group 8 (8-bit INT) see also (Page 567)							
...	...							
DBB270	Number of the active G function of G function group n-1 (8-bit INT) see also (Page 567)							
DBB271	Number of the active G function of G function group n (8-bit INT) see also (Page 567)							

Note

- The active G functions of the group, for each programming of a G function or a mnemonic identifier (e.g. SPLINE), are updated.
- G functions within a G group are output as binary value, starting with 1. A G function with the value 0 means that for this G group, no G function is active.

18.1.12.11 DB21 - DB30, protection areas from the channel

Table 18-86 DB21 - DB30, signals for the protection zones from the channel

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB272	Machine-related protection zone preactivated see also (Page 568)							
	8	7	6	5	4	3	2	1
DBB273	Machine-related protection zone preactivated see also (Page 568)							
							10	9
DBB274	Channel-specific protection zone preactivated see also (Page 569)							
	8	7	6	5	4	3	2	1
DBB275	Channel-specific protection zone preactivated see also (Page 569)							
							10	9

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB276	Machine-related protection zone violated see also (Page 569)							
	8	7	6	5	4	3	2	1
DBB277	Machine-related protection zone violated see also (Page 569)							
							10	9
DBB278	Channel-specific protection zone violated see also (Page 570)							
	8	7	6	5	4	3	2	1
DBB279	Channel-specific protection zone violated see also (Page 570)							
							10	9

18.1.12.12 DB21 - DB30, synchronous actions, signals from/to the channel

Note

The request signals should be set in the PLC user program. After data transfer, they are reset by the basic PLC program.

Table 18-87 DB21 - DB30, job-controlled signals from/to the channel

DB21 - DB30 /FBSY/	Signals to the channel (PLC ←→ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB280 PLC → NC							Require-ment Disable synchron-ized actions	Reserved
DBB281 NC → PLC							Acknowl-edgment Synchron-ized ac-tions disa-bled	
DBW282 - DBW298	Reserved							
DBB300	Lock synchronized action							
PLC → NC	8	7	6	5	4	3	2	1

DB21 - DB30 /FBSY/	**Signals to the channel (PLC ⟷ NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB301	Lock synchronized action							
PLC → NC	16	15	14	13	12	11	10	9
DBB302	Lock synchronized action							
PLC → NC	24	23	22	21	20	19	18	17
DBB303	Lock synchronized action							
PLC → NC	32	31	30	29	28	27	26	25
DBB304	Lock synchronized action							
PLC → NC	40	39	38	37	36	35	34	33
DBB305	Lock synchronized action							
PLC → NC	48	47	46	45	44	43	42	41
DBB306	Lock synchronized action							
PLC → NC	56	55	54	53	52	51	50	49
DBB307	Lock synchronized action							
PLC → NC	64	63	62	61	60	59	58	57
DBB308	Synchronized action can be locked							
NC → PLC	8	7	6	5	4	3	2	1
DBB309	Synchronized action can be locked							
NC → PLC	16	15	14	13	12	11	10	9
DBB310	Synchronized action can be locked							
NC → PLC	24	23	22	21	20	19	18	17
DBB311	Synchronized action can be locked							
NC → PLC	32	31	30	29	28	27	26	25
DBB312	Synchronized action can be locked							
NC → PLC	40	39	38	37	36	35	34	33
DBB313	Synchronized action can be locked							
NC → PLC	48	47	46	45	44	43	42	41
DBB314	Synchronized action can be locked							
NC → PLC	56	55	54	53	52	51	50	49
DBB315	Synchronized action can be locked							
NC → PLC	64	63	62	61	60	59	58	57

18.1.12.13 DB21 - DB30, control signals from/to the channel

Note

The request signals should be set in the PLC user program. After data transfer, they are reset by the basic PLC program.

Table 18-88 DB21 - DB30, job-controlled signals from/to the channel

DB21 - DB30	**Signals from/to the channel (PLC ⟷ NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB316	Active G functions							
								G00 geo.
DBB317	Tool missing see also (Page 572)	PTP traversing active see also (Page 572)	Drive test travel request				Target number of workpieces reached see also (Page 571)	External language mode active
DBB318	Overstore active /F1-A2/	Dry run feedrate active /FB1-V1/	M01 associated with PLC active see also (Page 574)	Delayed stop	TOFF motion active see also (Page 573)	TOFF active see also (Page 573)	Block search via program test, SERUPRO, is active see also (Page 573)	ASUB stopped see also (Page 572)
DBB319	No tool change commands active	Feedrate stop Delay Suppress	REPOS delay see also (Page 576)	Feedrate stop Delay	Active REPOS mode see also (Page 575)			Acknowledgment of the REPOS mode change see also (Page 574)
					C	B	A	

18.1.12.14 DB21 - DB30, signals to the orientation axes

Table 18-89 DB21 - DB30, signals to the orientation axes

DB21 - DB30	**Signals to the channel (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Orientation axis 1							
DBB320	Traversing keys see also (Page 578)		Rapid traverse override see also (Page 578)	Traversing key disable see also (Page 577)	Feedrate stop	Activate handwheel see also (Page 576)		
	Plus	Minus				C	B	A
DBB321		Requested machine function see also (Page 580)						
		Continuous traversing	var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

DB21 - DB30	Signals to the channel (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB322	OEM signals							
DBB323								Handwheel direction of rotation inverted see also (Page 581)
	Orientation axis 2							
DBB324	Traversing keys see also (Page 582)		Rapid traverse override see also (Page 582)	Traversing key disable see also (Page 581)	Feedrate stop	Activate handwheel see also (Page 581)		
	Plus	Minus				C	B	A
DBB325		Requested machine function see also (Page 582)						
		Continuous traversing	var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB326	OEM signals							
DBB327								Handwheel direction of rotation inverted see also (Page 582)
	Orientation axis 3							
DBB328	Traversing keys see also (Page 583)		Rapid traverse override see also (Page 582)	Traversing key disable see also (Page 582)	Feedrate stop	Activate handwheel see also (Page 582)		
	Plus	Minus				C	B	A
DBB329		Requested machine function see also (Page 583)						
		Continuous traversing	var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

DB21 - DB30	Signals to the channel (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB330	OEM signals							
DBB331								Handwheel direction of rotation inverted see also (Page 583)

18.1.12.15 DB21 - DB30, signals from the orientation axes

Table 18-90 DB21 - DB30, signals from the orientation axes

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Orientation axis 1							
DBB332	Traversing command see also (Page 585)		Travel request see also (Page 584)			Handwheel active see also (Page 583)		
	Plus	Minus	Plus	Minus		C	B	A
DBB333		Active machine function see also (Page 586)						
		Continuous traversing active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB334	OEM signals							
DBB335								Handwheel direction of rotation inversion active
	Orientation axis 2							
DBB336	Traversing command see also (Page 587)		Travel request see also (Page 587)			Handwheel active see also (Page 586)		
	Plus	Minus	Plus	Minus		C	B	A
DBB337		Active machine function see also (Page 587)						
		Continuous traversing active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB338	OEM signals							

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB339								Handwheel direction of rotation inversion active
	Orientation axis 3							
DBB340	Traversing command see also (Page 587)		Travel request see also (Page 587)			Handwheel active see also (Page 587)		
	Plus	Minus	Plus	Minus		C	B	A
DBB341		Active machine function see also (Page 587)						
		Continuous traversing active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB342	OEM signals							
DBB343								Handwheel direction of rotation inversion active

18.1.12.16 DB21 - DB30, tool management functions from the channel

Table 18-91 DB21 - DB30, tool management functions from the channel

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Change signals, tool management functions							
DBB344					Last replacement tool of the tool group see also (Page 589)	Transition to new replacement tool see also (Page 588)	Tool limit value reached see also (Page 588)	Tool prewarning limit reached see also (Page 588)
DBB345 - DBB347								
	Transferred tool management functions							
DBD348	T number for tool prewarning limit (DINT)							
DBD352	T number for tool limit value (DINT)							
DBD356	T number of new replacement tool (DINT)							
DBD360	T number of last replacement tool (DINT)							

18.1.12.17 DB21 - DB30, control signals from/to the channel (2)

Table 18-92 DB21 - DB30, signals from the channel

DB21 - DB30	**Signals from the channel (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB364	CH_CYCLES_SIG_IN (1)							
	8	7	6	5	4	3	2	1
DBB365	CH_CYCLES_SIG_IN (2)							
	16	15	14	13	12	11	10	9
DBB366	CH_CYCLES_SIG_OUT (1)							
	8	7	6	5	4	3	2	1
DBB367	CH_CYCLES_SIG_OUT (2)							
	16	15	14	13	12	11	10	9
DBB368	CH_OEM_TECHNO_SIG_IN (1)							
	8	7	6	5	4	3	2	1
DBB369	CH_OEM_TECHNO_SIG_IN (2)							
	16	15	14	13	12	11	10	9
DBB370	CH_OEM_TECHNO_SIG_IN (3)							
	24	23	22	21	20	19	18	17
DBB371	CH_OEM_TECHNO_SIG_IN (4)							
	32	31	30	29	28	27	26	25
DBB372	CH_OEM_TECHNO_SIG_OUT (1)							
	8	7	6	5	4	3	2	1
DBB373	CH_OEM_TECHNO_SIG_OUT (2)							
	16	15	14	13	12	11	10	9
DBB374	CH_OEM_TECHNO_SIG_OUT (3)							
	24	23	22	21	20	19	18	17
DBB375	CH_OEM_TECHNO_SIG_OUT (4)							
	32	31	30	29	28	27	26	25
DBB376	Display of the triggering event in case of event-driven program call ProgEventDisplay see also (Page 589)							
DBB377		Jog circle active see also (Page 590)	Retraction data available see also (Page 590)	JOG retract active see also (Page 590)			Stop condition	Collision avoidance: Stop see also (Page 590)

DB21 - DB30	Signals from the channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB378							"Silent" ASUB is active  see also (Page 591)	ASUB is active  see also (Page 591)
DBB379								
DBB380	Reserved ASUB							
DBB381	Reserved ASUB							
DBB382	Reserved ASUB							
DBB383	Reserved ASUB							

Table 18-93 DB21 - DB30, signals to the channel

DB21 - DB30	Signals to the channel (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB384								Enable GOTOS  see also (Page 591)
DBB385	Grinding: Input signals 1 ... 8 ($AC_IN_KEY_G[1 ... 8])							
	8	7	6	5	4	3	2	1
DBB386	Grinding: Disable input signals 1 ... 8							
	8	7	6	5	4	3	2	1
DBB387	Grinding: Status of the grinding functions 1 ... 8 ($AC_IN_KEY_G_RUN_IN[1...8])							
	8	7	6	5	4	3	2	1

Table 18-94 DB21 - DB30, signals from/to channel

DB21 - DB30	**Signals from/to the channel (NC ↔ PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW388 NC → PLC	Active transformation number							
DBB390 PLC → NC	Grinding: Enable status of input signals 1 ... 8 ($AC_IN_KEY_G_ISENABLE[1...8])							
	8	7	6	5	4	3	2	1
DBB391 NC → PLC	Grinding: Status of the grinding functions 1 ... 8 ($AC_IN_KEY_G_RUN_OUT[1...8])							
	8	7	6	5	4	3	2	1
DBB392 PLC → NC	Selection of the coordinate system for Cartesian manual traversing and handwheel override in the automatic mode in the tool direction (DRF) see also (Page 593)							
DBB393	Reserved							
DBB394	Reserved							
DBB395	Reserved							
DBB396	H function 1 (DInt), also in DBD142							
DBB400	H function 2 (DInt), also in DBD146							
DBB404	H function 3 (DInt), also in DBD150							
	Interface operating software -> PLC							
DBB 408 HMI -> PLC	Axis selection type: 0: Machine axis, 1: Geo axis							
DBB 409 HMI -> PLC	Axis number: Machine/geo axis							
DBB 410 HMI -> PLC								Key Axis selection
DBB 411								
	Interface PLC -> HMI							
DBB 412 PLC -> HMI	Active axis selection (feedback message to the HMI) 0: Machine axis, 1: Geo axis							
DBB 413 PLC -> HMI	Active axis number (feedback message to the HMI): Machine/geo axis							

18.1.13 Axis/spindle signals

18.1.13.1 DB31 - DB61, signals to the axis/spindle

Table 18-95 DB31 - DB61, signals to the axis/spindle

DB31 - DB61	Signals to the axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Feedrate override, axis-specific see also (Page 594)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB1 Axis and spindle	Override active see also (Page 604)	Position measuring system 1 - 2 see also (Page 602)		Follow-up mode see also (Page 601)	Axis/spindle disable see also (Page 598)	Sensor for fixed stop see also (Page 597)	Acknowledge fixed stop reached see also (Page 597)	Drive test travel enable see also (Page 596)
		Bit 2	Bit 1					
DBB2 Axis and spindle	Reference point value 1 - 4 see also (Page 609)				Clamping in progress see also (Page 609)	Delete distance-to-go / spindle reset see also (Page 607)	Controller enable see also (Page 605)	Cam activation see also (Page 605)
	Bit 4	Bit 3	Bit 2	Bit 1				
DBB3 Axis and spindle	Program test axis/ spindle enable	Velocity / spindle speed limitation see also (Page 611)	Activate fixed feedrate see also (Page 611)				Travel to fixed stop enabled see also (Page 610)	Accept offset external WO see also (Page 610)
			Bit 4	Bit 3	Bit 2	Bit 1		
DBB4 Axis and spindle	Traversing keys see also (Page 614)		Rapid traverse override see also (Page 614)	Traversing key disable see also (Page 614)	Feedrate stop / spindle stop see also (Page 613)	Activate handwheel see also (Page 612)		
	Plus	Minus				Bit 3	Bit 2	Bit 1
DBB5 Axis and spindle		Machine function see also (Page 615)						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB6	OEM signals							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

DB31 - DB61	**Signals to the axis/spindle (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB7	OEM signals							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Invert handwheel direction of rotation see also (Page 616)
DBB8	Request PLC axis/spindle			Channel assignment changed	NC axis / spindle channel assignment see also (Page 617)			
					Bit 3	Bit 2	Bit 1	Bit 0
DBB9					Parameter set change inhibited see also (Page 618)	Controller parameter set see also (Page 618)		
						Bit 2	Bit 1	Bit 0
DBB10								REPOS delay see also (Page 618)
DBB11								SI: Start brake test
DBB12 Axis	Delay reference point approach see also (Page 620)			Modulo rotary axes: Activate traversing range limits see also (Page 620)	2nd software limit switch		Hardware limit switch	
					Plus see also (Page 620)	Minus see also (Page 620)	Plus see also (Page 619)	Minus see also (Page 619)
DBB13 Axis					JOG to position see also (Page 621)	JOG fixed point approach see also (Page 621)		
						Bit 2	Bit 1	Bit 0
DBB14 Axis							Program test	
							Activate see also (Page 623)	Suppress see also (Page 622)
DBB15 Axis								

DB31 - DB61	Signals to the axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB16 Spindle	Delete S value see also (Page 626)	No speed monitoring during gear change	Resynchronize spindle		Gear is changed over see also (Page 625)	Actual gear stage see also (Page 623)		
			Measuring system 2 see also (Page 625)	Measuring system 1 see also (Page 624)		Bit 2	Bit 1	Bit 0
DBB17 Spindle		Invert M3/M4 see also (Page 627)	Resynchronize spindle during positioning					Feedrate correction spindle valid
			Measuring system 2 see also (Page 626)	Measuring system 1 see also (Page 626)				
DBB18 Spindle	Oscillation direction of rotation		Oscillation enable see also (Page 628)	Oscillation controlled by PLC see also (Page 627)				
	Links see also (Page 629)	Right see also (Page 628)						
DBB19 Spindle	Speed override, spindle-specific see also (Page 629)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB20 Drive			Open motor holding brake				Ramp-function generator disable [1] see also (Page 631)	
DBB21 Drive	Pulse enable see also (Page 633)	Integrator disable, speed controller see also (Page 633)	Motor being selected see also (Page 632)	Motor/drive data set: Selection (interface definition: DB31, …DBX130.0 - 4 (Page 408)) see also (Page 632)				
				Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB22 Safety Integrated				SI: SG selection see also (Page 634)		SI: Acknowledgment, communication failure	SI: SBH deselection see also (Page 634)	SI: SBH/SG deselection see also (Page 634)
				Bit 1	Bit 0			

DB31 - DB61	Signals to the axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB23 Safety Integrated	SI: Test stop selection		SINAMICS Close brake	SI: SE selection see also (Page 635)		SI: Gear ratio selection see also (Page 635)		
						Bit 2	Bit 1	Bit 0
DBB24	Master/ slave: switch on see also (Page 637)		Setpoint switchover Accept drive control see also (Page 637)	Master/ slave: Activate torque equalization controller see also (Page 637)	MCS coupling: Activate collision protection see also (Page 636)	MCS coupling: deactivate or do not permit see also (Page 636)	Control axis	Stepping motor: Rotation monitoring
DBB25								Dynam. Activate backlash compensation
DBB26 Grinding				Override Enable see also (Page 638)	Compensatory controller on			
DBB27	Stop				Resume			
Grinding	HIAxMove	Corr.	DEPBCS	DEPMCS	HIAxMove	Corr.	DEPBCS	DEPMCS
DBB28 Grinding: Oscillation	PLC controls axis see also (Page 642)	PLC-controlled axis: Stop along braking ramp see also (Page 641)	PLC-controlled axis: Stop at next reversal point see also (Page 641)	Alter reversal point see also (Page 641)	Set reversal point see also (Page 640)	PLC-controlled axis: Continue see also (Page 639)	PLC-controlled axis: Reset see also (Page 639)	Oscillation reversal from external see also (Page 638)
DBB29 Couplings			Disable automatic synchronization	Start gantry synchronization				
DBB30 Technology				Spindle-start positioning	Select gear stage	Spindle start, counter-clockwise rotation	Spindle start, clockwise rotation	Spindle stop
DBB31 Technology	Delete synchronism correction	Correct synchronous operation	Disable synchronous operation see also (Page 643)	Re-synchronize	REPOS for transformation with PTP			

DB31 - DB61	Signals to the axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB32 Safety Integrated			SI: Deselect external STOP_E	SI: Deselect external STOP_D	SI: Deselect external STOP_C	SI: Deselect external STOP_A		
DBB33	SI: SG override selection/override							
Safety Integrated	Bit 3	Bit 2	Bit 1	Bit 0				
DBB34							Setpoint limitation	
							Bit 1	Bit 0
DBB35								
DBB36 Technology								
DBB37								
DBB38								
DBB39								
DBB40 - DBB55								
DBB56 PLC → HMI						Spindle internal clamping	Spindle speed display	Separate feedrate drive coupled as C axis
DBB57								
DBB58	Reserved							
DBB59								
DBB188								
DBB189							Motor overtemperature: Disable current reduction	DYNEGMA: Disable derating

[1] Only when cyclic interface between NC and drive is operated in the "611U compatibility mode".

Note

DBX8.4 is automatically reset after the assignment is executed

18.1.13.2 DB31 - DB61, signals from the axis/spindle

Table 18-96 DB31 - DB61, signals from the axis/spindle

DB31 - DB61	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB60 Axis / spindle	Position reached with exact stop		Referenced/synchronized Position measuring system		Encoder limit frequency exceeded, measuring system		NCU link: Axis active see also (Page 644)	Spindle / rotary axis see also (Page 644)
	fine see also (Page 647)	coarse see also (Page 647)	2 see also (Page 646)	1 see also (Page 646)	2 see also (Page 645)	1 see also (Page 645)		
DBB61 Axis / spindle	Current controller active see also (Page 651)	Speed controller active see also (Page 651)	Position controller active see also (Page 650)	Axis/spindle stationary ($n < n_{min}$) see also (Page 650)	Follow-up mode active see also (Page 649)	Axis ready see also (Page 649)	Axis-specific alarm see also (Page 648)	Drive test travel request see also (Page 648)
DBB62	Axis container: Rotation active see also (Page 654)	Travel to fixed stop: Force limiting active	Travel to fixed stop: Fixed stop reached see also (Page 654)	Travel to fixed stop: Activating the function see also (Page 653)	Measurement active see also (Page 653)	Revolutional feedrate active see also (Page 653)	Handwheel override active see also (Page 652)	Software cams active see also (Page 652)
DBB63	Stop				Axis/spindle disable active	Axis stop active see also (Page 655)	PLC-controlled axis see also (Page 655)	Reset executed see also (Page 654)
	HIAxMove active	Corr. active	DEPBCS active	DEPMCS active				
DBB64 Axis / spindle	Traversing command see also (Page 657)		Travel request see also (Page 657)			Handwheel active see also (Page 656)		
	Plus	Minus	Plus	Minus		Bit 2	Bit 1	Bit 0
DBB65 Axis / spindle		Active machine function see also (Page 658)						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB66 Axis / spindle								MCS coupling: Collision protection active see also (Page 658)

DB31 - DB61	Signals from the axis/spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB67								Handwheel direction of rotation inversion active  see also (Page 659)
DBB68	Status axis/spindle replacement  see also (Page 659)							
	PLC axis/ spindle	Neutral axis/spindle	Axis interchange possible	New type requested from PLC	NC axis / spindle channel assignment			
					Bit 3	Bit 2	Bit 1	Bit 0
DBB69	NCU number in the NCU link group					Controller parameter set servo  see also (Page 659)		
	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0
DBB70			DRV Safety Integrated active with SIC/SCC	NC Safety Integrated active		REPOS delay acknowledgment  see also (Page 661)	REPOS offset valid  see also (Page 660)	REPOS offset  see also (Page 660)
DBB71	PLC axis permanently assigned		Position restored					Brake test active
			Encoder 2  see also (Page 662)	Encoder 1  see also (Page 662)				
DBB72 HMI → PLC								REPOS delay active  see also (Page 662)
DBB73 HMI → PLC								
DBB74 Axis				Modulo rotary axes: Traversing range limits active  see also (Page 663)				

DB31 - DB61	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB75 Axis	JOG position reached see also (Page 664)	JOG travel to position active see also (Page 664)	JOG approach fixed point reached see also (Page 664)			JOG approach fixed point active see also (Page 663)		
			Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0
DBB76 Axis	Rounding axis in position	Indexing axis in position see also (Page 666)	Positioning axis see also (Page 666)	Path axis see also (Page 665)				Lubrication pulse see also (Page 665)
DBB77 Axis								Collision avoidance: Velocity reduction see also (Page 667)
DBD78 Axis	F value (REAL) for positioning axis see also (Page 667)							
DBB82 Spindle					Change gear stage see also (Page 668)	Setpoint gear stage see also (Page 667)		
						Bit 2	Bit 1	Bit 0
DBB83 Spindle	Actual direction of rotation clockwise see also (Page 672)	Speed monitoring see also (Page 671)	Spindle in setpoint range see also (Page 671)	Support range limits violated	Geometry monitoring see also (Page 671)	Speed setpoint Increased see also (Page 669)	Speed setpoint limited see also (Page 668)	Speed limit exceeded see also (Page 670)
DBB84 Spindle	Active spindle mode Control mode see also (Page 674)	Active spindle mode Oscillation mode see also (Page 674)	Active spindle mode Positioning mode see also (Page 673)	Active spindle mode Synchronous operation see also (Page 673)	Tapping without compensating chuck active see also (Page 672)	CLGON active	GWPS active see also (Page 672)	Const. cutting velocity active
DBB85 Spindle			Spindle actually reached position see also (Page 675)					Tool with dynamic response limitation see also (Page 674)

DB31 - DB61	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW86 Spindle	M function (INT) for spindle (M3, M4, M5, M19, M70 or defined via MD) see also (Page 675)							
DBD88 Spindle	S function (REAL) for spindle see also (Page 676)							
DBB92 Drive	Enable drive operation		Motor holding brake opened	Independent drive motion active[1] see also (Page 677)			Ramp-function-generator disable active see also (Page 676)	
DBB93 Drive	Pulse enable see also (Page 679)	Integrator disable, speed controller see also (Page 678)	Drive ready see also (Page 678)	Motor/drive data set: Display (interface definition: DB31, …DBX130.0 - 4 (Page 644)) see also (Page 677)				
				Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB94 Drive	Variable signaling function[2] see also (Page 683)	$n_{act} = n_{set}$ see also (Page 683)	$\|n_{act}\| < n_x$ see also (Page 682)	$\|n_{act}\| < n_{min}$ see also (Page 682)	$M_d < M_{dx}$ see also (Page 681)	Run-up completed see also (Page 680)	Temperature pre-alarm	
							Heat sink see also (Page 680)	Motor see also (Page 679)
DBB95 Drive	Alarm of alarm class C is active see also (Page 684)				ESR: Generator operation minimum speed fallen below (p2161) see also (Page 684)	ESR: Response triggered or generator operation active (r0887.12) see also (Page 684)	ESR: DC link undervoltage (p1248) see also (Page 683)	
DBB96	Master/ slave: Coupling active see also (Page 686)		Setpoint switchover: drive control active see also (Page 686)	Master-slave:			Control axis active	Stepping motor: Rotation monitoring error
				Compensatory controller active see also (Page 685)	Coarse speed difference see also (Page 685)	Fine speed difference see also (Page 685)		
DBB97					MCS coupling:			
					Offset change see also (Page 688)	Mirroring active see also (Page 687)	Coupling active see also (Page 687)	Following axis see also (Page 687)

DB31 - DB61	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB98 Synchronous spindle	ESR response initiated	Acceleration warning threshold reached see also (Page 691)	Velocity warning threshold reached see also (Page 690)	Superimposed motion see also (Page 690)		Actual value coupling see also (Page 689)	Synchronous operation: coarse see also (Page 689)	Synchronous operation: fine see also (Page 688)
DBB99 Synchronous spindle		Max. acceleration reached	Max. velocity reached	Synchronization in progress	Axis accelerating see also (Page 692)	Synchronism correction implemented	Following spindle active see also (Page 692)	Leading spindle active see also (Page 691)
DBB100 Oscillation grinding	Oscillation active see also (Page 694)	Oscillation motion active see also (Page 693)	Sparking-out active see also (Page 693)	Fault during oscillation motion see also (Page 693)	Oscillation cannot be started work for now I get why you see also (Page 693)	Oscillation reversal from external active see also (Page 692)		
DBB101 Gantry	Gantry axis	Gantry guide axis	Gantry grouping is synchronous	Gantry synchronization ready to start	Gantry alarm limit exceeded	Gantry shutdown limit exceeded		
DBB102		Position measuring system activated: 2 see also (Page 694)	Position measuring system activated: 1 see also (Page 694)		Clamping tolerance exceeded			Dynam. Backlash compensation active
DBB103			Synchronous operation 2: coarse	Synchronous operation 2: fine				Synchronism correction is taken into account
DBB104 Grinding	Active infeed axis see also (Page 695)							
	8	7	6	5	4	3	2	1
DBB105 Grinding	Active infeed axis							
	16	15	14	13	12	11	10	9
DBB106 Grinding	Active infeed axis							
	24	23	22	21	20	19	18	17
DBB107 Grinding	Active infeed axis							
		31	30	29	28	27	26	25

DB31 - DB61	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB108 Safety Integrated	SI: Axis safely referenced			SI: Communication failure not acknowledged	SI: CRC error	SI: Status, pulses are canceled	SI: CRC or sign-of-life error	SI: SBH/SG active
DBB109 Safety Integrated	SI: Output cam position							
	SN 4-	SN 4+	SN 3-	SN 3+	SN 2-	SN 2+	SN 1-	SN 1+
DBB110 Safety Integrated			n < nx	SI: Active SG			SI: SBH active	
				B	A			
DBB111 Safety Integrated	SI: Stop A - E							
	Stop E Active	Stop D Active	Stop C Active	Stop A/B Active				
DBB112 Safety Integrated	Output cam range for cam track 1							
DBB113 Safety Integrated	Output cam range for cam track 2							
DBB114 Safety Integrated	Output cam range for cam track 3							
DBB115 Safety Integrated	Output cam range for cam track 4							
DBB116 Safety Integrated	Reserved							
DBB117 Safety Integrated	Reserved				Output cam track			
					4	3	2	1
DBB118 Safety Integrated	SI: Cam range bit for cam track 1							
	7	6	5	4	3	2	1	0
DBB119 Safety Integrated	SI: Cam range bit for cam track 1							
		14	13	12	11	10	9	8
DBB120 Safety Integrated	SI: Cam range bit for cam track 2							
	7	6	5	4	3	2	1	0
DBB121 Safety Integrated	SI: Cam range bit for cam track 2							
		14	13	12	11	10	9	8

DB31 - DB61	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB122	SI: Cam range bit for cam track 3							
Safety Integrated	7	6	5	4	3	2	1	0
DBB123	SI: Cam range bit for cam track 3							
Safety Integrated		14	13	12	11	10	9	8
DBB124	SI: Cam range bit for cam track 4							
Safety Integrated	7	6	5	4	3	2	1	0
DBB125	SI: Cam range bit for cam track 4							
Safety Integrated		14	13	12	11	10	9	8
DBB126								
DBB127								
DBB128							Program test	
HMI → PLC							Activate  see also (Page 695)	Suppress  see also (Page 695)
DBB129								
DBB130	Motor/drive data set: Formatting is valid			Motor/drive data set: Formatting (interface definition: DB31, …DBX21.0 ... 4 (Page 403), DB 31. ...DBX93.0 ... 4 (Page 677))  see also (Page 696)				
				Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB131								
DBB132	Sensor configuration							
Weiss spindle		Sensor S6 available	Sensor S5 available (angular position, motor shaft)  see also (Page 697)	Sensor S4 available (piston end position)  see also (Page 697)			Sensor S1 available (clamped state)  see also (Page 697)	Sensors available  see also (Page 696)

DB31 - DB61	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB133	Sensor configuration							
Weiss spindle						Status value is generated, speed limitation p5043 is active  see also (Page 698)		
DBW134 Weiss spindle	Clamping state (sensor S1) status value  see also (Page 698)							
DBW136 Weiss spindle	Clamping state (sensor S1) analog value  see also (Page 699)							
DBB138	Status digital sensors							
Weiss spindle			Sensor S5 angular position, motor shaft  see also (Page 699)	Sensor S4, piston end position  see also (Page 699)				
DBB139	Status digital sensors							
Weiss spindle								
DBB190								Encoder servicing required
DBB191	OR operation: Condition fulfilled	AND operation: Condition fulfilled			Pole position identification with encoder successfully carried out	Motor overtemperature		DYNEGMA: Torque limit applied
						Motor temperature model warning pending	Current reduction active	

1) With SINAMICS valid for NC 62.07 and higher when using a 611U telegram type

2) With SINAMICS valid for SW2.6 and higher

18.1.14 Safety Integrated

18.1.14.1 DB31 - DB61, Safety Control Channel (SCC)

Table 18-97 DB31 - DB61, axis signals: Safety Control Channel (SCC)

DB31 - DB61 /FBSIsl/	**Signals to the axis/spindle**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	SCC (PLC → drive)							
DBB140								Test stops for extended functions
DBB141								
DBB142								
DBB143			External brake closed (SBC)	Test sequence 1 or 2	Direction of rotation	Test with brake 1 or 2 (SBT)	Start brake test (SBT)	Select safe brake test (SBT)
DBB144 ... DBB163								

18.1.14.2 DB31 - DB61, Safety Info Channel (SIC)

Table 18-98 DB31 - DB61, axis signals: Safety Info Channel (SIC)

DB31 - DB61 /FBSIsl/	**Signals from axis/spindle**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	SIC (drive → PLC)							
DBB164	Safety error active	ESR requested				Bit 1 limit value Safely limited speed (SLS)	Bit 0 limit value Safely limited speed (SLS)	Safely limited acceleration (SLA)
DBB165	Safety error with Stop A	Safely-limited speed selected (SLS)	Safe operating stop selected (SOS)	Safely-limited speed active (SLS)	Safe operating stop active (SOS)	Safe stop 2 (SS2)	Safe stop 1 (SS1)	Safe torque off (STO)
DBB166			Acceptance test stop requested	Acceptance test stop active			Safe direction negative (SDI)	Safe direction positive (SDI)

DB31 - DB61 /FBSIsl/	**Signals from axis/spindle**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB167	Safely limited position is selected (SLP)			Bit 0 For area Safely limited position (SLP)				
DBD168	Speed limit							
DBB172	Acceptance test: Safely limited position selected (SLP)	Acceptance test: Safely limited position active (SLP)			SS2E_ACTIVE			
DBB173	Load torque negative sign	Close external brake (SBC)	Brake test (SBT)					
			completed	OK	Active	with brake 2	Setpoint input during SBT in the drive	Safe Brake Test (SBT)
DBB174 ... DBB187								

18.1.15 Tool management

18.1.15.1 DB71, interface for loading/unloading the magazine

Table 18-99 DB71, interface for loading/unloading the magazine

DB71	**Loading/unloading positions (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (SS) active see also (Page 700)							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB1								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBB2	Standard end acknowledgement see also (Page 700)							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB3								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9

DB71	Loading/unloading positions (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn + 0	Reserved		Command:					
			Multitool positioning see also (Page 703)	NC progr. positions magazine see also (Page 702)	Positioning to the loading point see also (Page 702)	Relocating see also (Page 701)	Unloading see also (Page 701)	Loading see also (Page 700)
DBBn + 1	Command: Data in extended Area (DB1071 (Page 431)) see also (Page 704)	Reserved						Acknowledgement status = 3 see also (Page 703)
DBBn + 2	Assigned channel (8-bit INT) see also (Page 704)							
DBBn + 3	Tool management no. (8-bit INT) see also (Page 704)							
DBBn + 4	Reserved (free parameter 1 (DWord))							
DBBn + 8	Reserved (free parameter 2 (DWord))							
DBDn + 12	Reserved (free parameter 3 (DWord))							
DBWn + 16	Identifier for loading/unloading point (INT), (fixed value 9999) see also (Page 705)							
DBWn + 18	Location no. of load/unload point (INT) see also (Page 705)							
DBWn + 20	Magazine no. (Source) for unloading/relocating/positioning (INT) see also (Page 705)							
DBWn + 22	Location no. (Source) for unloading/relocating/positioning (INT) see also (Page 706)							
DBWn + 24	Magazine no. (Target) for loading/relocating/positioning (INT) see also (Page 706)							
DBWn + 26	Location no. (Target) for loading/relocating/positioning (INT) see also (Page 706)							
DBWn + 28	Reserved							Loading/ unloading without moving magazine see also (Page 707)
DBWn + 29	Reserved							

Starting addresses of loading/unloading points:

Loading/unloading point 1: n = 4
Loading/unloading point 2: n = 34
Loading/unloading point 3: n = 64
Loading/unloading point 4: n = 94

Loading interface 1 is responsible for loading/unloading in (all) spindles/toolholders and for relocating tools and for positioning at any locations (e.g. buffer location).

Loading and unloading manual tools is always realized via loading interface 1.

18.1.15.2 DB72, interface for the spindle as change position

Table 18-100 DB72, interface for the spindle as change position

DB72	Signals from the spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (SS) active see also (Page 707)							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB1								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBB2	Standard end acknowledgment see also (Page 708)							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB3								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBBn + 0	Command code:							
	Spindle tool remains in the spindle see also (Page 711)	Replace manual tool see also (Page 711)	Insert manual tool see also (Page 710)	Old T in buffer see also (Page 710)	T0 see also (Page 709)	Prepare change see also (Page 709)	Perform change (initiate: M06) see also (Page 708)	Obligatory change see also (Page 708)
DBBn + 1	Data in extended Area (DB1072 (Page 432)) see also (Page 712)	Reserved						Acknowledgment status = 3 see also (Page 712)
DBBn + 2	Assigned channel (8-bit INT) see also (Page 712)							
DBBn + 3	Tool management No. (8-bit INT) see also (Page 713)							
DBDn + 4	User parameter 0 (DWord) see also (Page 713)							

DB72	Signals from the spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBDn + 8	User parameter 1 (DWord) see also (Page 713)							
DBDn + 12	User parameter 2 (DWord) see also (Page 714)							
DBWn + 16	Buffer identifier (INT), (fixed value 9998) (corresponds to "Target position for new tool") see also (Page 714)							
DBWn + 18	Relative location (target) in buffer magazine (INT) see also (Page 714)							
DBWn + 20	Magazine No. (Source) for new tool (INT) see also (Page 715)							
DBWn + 22	Location No. (Source) for new tool (INT) see also (Page 715)							
DBWn + 24	Magazine No. (Target) for old tool (INT) see also (Page 715)							
DBWn + 26	Location No. (Target) for old tool (INT) see also (Page 716)							
DBWn + 28	Tool new: Location type (INT) see also (Page 716)							
DBWn + 30	Tool new: Size left (INT) see also (Page 716)							
DBWn + 32	Tool new: Size right (INT) see also (Page 717)							
DBWn + 34	Tool new: Size top (INT) see also (Page 717)							
DBWn + 36	Tool new: Size bottom (INT) see also (Page 717)							
DBBn + 38	Tool status for new tool see also (Page 718)							
	Manual tool	1:1 exchange	Reserved	Master tool	Tool to be loaded	Tool to be unloaded	Disabled, but ignore	Tool in the buffer
DBBn + 39	Tool status for new tool see also (Page 718)							
	Tool has been in use	Tool fixed-location-coded	Tool being changed	Prewarning limit reached	Measure tool	Tool disabled	Enable tool	Active tool
DBWn + 40	Tool new: Internal T No. (INT) see also (Page 718)							
DBWn + 42	Buffer location of the old tool see also (Page 719)							
DBWn + 44	Original magazine of the new tool see also (Page 719)							

DB72	Signals from the spindle (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBWn + 46	Original location of the new tool see also (Page 719)							

Start addresses of the buffer

Spindle 1: n = 4
Spindle 2: n = 52

18.1.15.3 DB73, interface for the turret

Table 18-101 DB73, interface for the turret

DB73	Signals for the turret (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (SS) active see also (Page 720)							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB1								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBB2	Standard end acknowledgment see also (Page 720)							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB3								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBBn	Reserved	Command code:						
		Unload manual tool	Reserved		T0 see also (Page 721)	Reserved	Perform change see also (Page 721)	Obligatory change see also (Page 721)
DBBn + 1	Data in extended Area (DB1073 (Page 433)) see also (Page 722)	Reserved						Acknowledgment status = 3 see also (Page 722)
DBBn + 2	Assigned channel (8-bit INT) see also (Page 722)							
DBBn + 3	Tool management No. (8-bit INT) see also (Page 723)							

DB73	Signals for the turret (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBDn + 4	User parameter 1 (DWord) see also (Page 723)							
DBDn + 8	User parameter 2 (DWord) see also (Page 723)							
DBDn + 12	User parameter 3 (DWord) see also (Page 724)							
DBWn + 16	Reserved							
DBWn + 18	Reserved							
DBWn + 20	Magazine No. of the turret (INT) see also (Page 724)							
DBWn + 22	Location No. of new tool (INT) see also (Page 724)							
DBWn + 24	Magazine No. of old tool see also (Page 725)							
DBWn + 26	Location No. of old tool (INT) see also (Page 725)							
DBWn + 28	Tool new: Location type (INT) see also (Page 725)							
DBWn + 30	Tool new: Size left (INT) see also (Page 726)							
DBWn + 32	Tool new: Size right (INT) see also (Page 726)							
DBWn + 34	Tool new: Size top (INT) see also (Page 726)							
DBWn + 36	Tool new: Size bottom (INT) see also (Page 727)							
DBBn + 38	Tool status for new tool see also (Page 727)							
	Manual tool	1:1 exchange	Reserved	Master tool	Tool to be loaded	Tool to be unloaded	Disabled, but ignore	Tool in the buffer
DBBn + 39	Tool status for new tool see also (Page 727)							
	Tool has been in use	Tool fixed-location-coded	Tool being changed	Prewarning limit reached	Measure tool	Tool disabled	Enable tool	Active tool
DBWn + 40	Tool new: Internal T No. (INT) see also (Page 728)							
DBWn + 42	Original location of new tool in this circular magazine see also (Page 728)							

Start addresses of the turret:

Turret, 1: n = 4
Turret, 2: n = 48

18.1.16 Signals from/to the machine control panel and the handheld unit

18.1.16.1 DB77, signals from/to the MCP and the HHU

Table 18-102 DB77, signals from/to the MCP and the HHU

DB77	Signals from/to MCP and HHU (GD communication)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0 - DBB7	Input signals from MCP1 to the PLC, MPI bus							
DBB8 - DBB15	Output signals from MCP1 to the PLC, MPI bus							
DBD16	Status, send MCP1, MPI bus							
DBD20	Status, receive MCP1, MPI bus							
DBB24 - DBB31	Input signals from MCP2 to the PLC, MPI bus							
DBB32 - DBB39	Output signals from MCP2 to the PLC, MPI bus							
DBD40	Status, send MCP2, MPI bus							
DBD44	Status, receive MCP2, MPI bus							
DBB48 - DBB53	Input signals from the HHU to the PLC, MPI bus							
DBB60 - DBB79	Output signals from the HHU to the PLC, MPI bus							
DBD80	Status, send HHU, MPI bus							
DBD84	Status, receive HHU, MPI bus							

18.1.17 Signals for Ctrl-Energy

18.1.17.1 DB1000, energy-saving profiles

Table 18-103 DB1000, energy-saving profiles

DB1000 /SCE/	Ctrl-Energy (operating software → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn + 0	Control signals							
							Set time to prewarning limit	Immediately activate energy-saving profile
DBBn + 1	Control signals							
								Directly activate energy-saving profile
DBBn + 2	Signals to check/test the energy-saving profile							
							PLC user signal	Master computer signal
DBBn + 3	Reserved							
DBBn + 4	Status signal							
							Activation time T1 has expired	Energy-saving profile active
DBBn + 5	Reserved							
DBWn + 6	Actual value: Actual value T1							
DBWn + 8	Actual value: Actual value T2							
DBBn + 10	Effectiveness, profile							
							Disable energy-saving profile	Energy-saving profile configured
	State conditions							
DBBn + 11								
						Screen change active	Data transfer active	Keyboard has been used

DB1000 /SCE/	Ctrl-Energy (operating software → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn + 12								
								MP used for operation
DBBn + 13	NC channel in reset							
	8	7	6	5	4	3	2	1
DBBn + 14	NC channel in reset							
							10	9
DBBn + 15								
							PLC user signal	Master computer signal
DBWn + 16	Activation time T1							
DBWn + 18	Pre-warning time T2							

Additional profile instances

Energy profile 2: DB1000 DBB20…DBB39
Energy profile 3: DB1000 DBB40…DBB59
Energy profile 4: DB1000 DBB60…DBB79
Energy profile 5: DB1000 DBB80…DBB99
Energy profile 6: DB1000 DBB100…DBB119
Energy profile 7: DB1000 DBB120…DBB139
Energy profile 8: DB1000 DBB140…DBB159

18.1.18 SENTRON PAC

18.1.18.1 DB1001, SENTRON PAC

Table 18-104 DB1001, signals for SENTRON PAC

DB1001 /SCE/	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0 PLC → operating software			Regenerative energy: Integration using FW	Regenerative energy: Integration using FW	Regenerative energy is read by SENTRON PAC	Energy fed in is read by SENTRON PAC	SENTRON PAC represents the machine	Display manual value
DBB1 GP								GP should perform measurement
DBB2								
Operating software → PLC								Measurement in progress
DBB3 PLC → operating software								Power display on
DBD4	Manual value (REAL) to operating software							
PLC → operating software								
DBD8	Total active power (REAL) to operating software							
PLC → operating software								
DBD12	Measured drawn active energy in kWh (REAL) to the operating software							
PLC → operating software								
DBD16	Measured supplied active energy in kWh (REAL) to the operating software							
PLC → operating software								
DBD20	Total active power in watts (REAL) from SENTRON							
GP								
DBD24	Drawn active energy at tariff 1 (F) in Wh (REAL) from SENTRON							
GP								
DBD28	Supplied active energy at tariff 1 (F) in Wh (REAL) from SENTRON							
GP								

DB1001 /SCE/	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD32	Drawn active energy in kWh per day (REAL) to the operating software							
DBD36	Supplied active energy in kWh day (REAL) to the operating software							
DBD40	Drawn active energy in kWh previous day (REAL) to the operating software							
DBD44	Supplied active energy in kWh previous day (REAL) to the operating software							
DBD48	Drawn active energy in kWh month (REAL) to the operating software							
DBD52	Supplied active energy in kWh month (REAL) to the operating software							
DBD56	Drawn active energy in kWh previous month (REAL) to the operating software							
DBD60	Supplied active energy in kWh previous month (REAL) to the operating software							
DBD64	Drawn active energy in kWh year (REAL) to the operating software							
DBD68	Supplied active energy in kWh year (REAL) to the operating software							
DBD72	Drawn active energy in kWh previous year (REAL) to the operating software							
DBD76	Supplied active energy in kWh previous year (REAL) to the operating software							
DBB80 ... DBB95	Reserved							
DBB96	ProductionAct							
PLC→GP/ operating software								
DBB97						Values invalid in		
GP						DBD28	DBD24	DBD20
DBB98	Values invalid in							
GP	DBD384	DBD344	DBD304	DBD264	DBD224	DBD184	DBD144	DBD104
DBB99							Values invalid in	
GP							DBD464	DBD424

18.1.18.2 DB1001, SENTRON PAC, auxiliary devices

Table 18-105 DB1001, signals for SENTRON PAC

DB1001 /SCE/	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn+100	Control/command bits							
PLC → GP operating software	Reset data structure			Input mode: (energy or power)	Different values when measuring energy	Energy measurement running	Read energy actual value update	Process device
DBBn+101	Reserved							
DBBn+102	Reserved							
DBBn+104	Active power or active energy of the auxiliary unit [kW] or [kWh]							
PLC → GP operating software								
DBDn+108	Drawn active energy of the auxiliary unit [kWh]							
PLC → operating software								
DBDn+112	Active energy supplied by the auxiliary unit [kWh]							
GP → operating software								
DBDn+116	Active energy drawn by the auxiliary unit at the measurement start [kWh]							
GP → operating software								
DBDn+120	Active energy supplied by the auxiliary unit at the measurement start [kWh]							
GP → operating software								
DBDn+124	Active energy drawn by the auxiliary unit at the measurement end [kWh]							
GP → operating software								
DBDn+128	Active energy supplied by the auxiliary unit at the measurement end [kWh]							
GP → operating software								
DBDn+132	Reserved							
DBDn+136	Reserved							

Auxiliary unit instances:

Auxiliary unit 1	(n=0):	DB1001 DBB100…DBB139
Auxiliary unit 2	(n=40):	DB1001 DBB140…DBB179
Auxiliary unit 3	(n=80):	DB1001 DBB180…DBB219
Auxiliary unit 4	(n=120):	DB1001 DBB220…DBB259
Auxiliary unit 5	(n=160):	DB1001 DBB260…DBB299
Auxiliary unit 6	(n=200):	DB1001 DBB300…DBB339
Auxiliary unit 7	(n=240):	DB1001 DBB340…DBB379
Auxiliary unit 8	(n=280):	DB1001 DBB380…DBB419
Auxiliary unit 9	(n=320):	DB1001 DBB420…DBB459
Auxiliary unit 10	(n=360):	DB1001 DBB460…DBB499

18.1.19 Spindle temperature sensor

18.1.19.1 DB1002: spindle temperature sensors

Table 18-106 DB1002, signals for **spindle temperature sensors**

DB1002 /SCE/	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 0	Sensor n Sensor installation location							
DBWn + 2	Reserved							
DBDn + 4	Sensor n Temperature sensor actual value [°C]							
DBDn + 8	Sensor n Temperature sensor warning threshold value [°C]							
DBWn + 12	Sensor n Number of alarm limit value violations							
DBBn + 14	Sensor n Last alarm limit value violation: Year total active power (REAL) to the operating software							
DBBn + 15	Sensor n Last alarm limit value violation: Month							

DB1002 /SCE/	SENTRON PAC							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn + 16	Sensor n Last alarm limit value violation: Day							
DBBn + 17	Sensor n Last alarm limit value violation: Hour							
DBBn + 18	Sensor n Last alarm limit value violation: Minute							
DBBn + 19	Sensor n Last alarm limit value violation: Seconds							
DBBn + 20	Sensor n Duration of the warning limit value violations							
DBBn + 24	Sensor n Temperature sensor fault threshold value [°C]							
DBBn + 28	Sensor n Number of fault limit value violations							
DBBn + 30	Sensor n Last alarm limit value violation: Year							
DBBn + 31	Sensor n Last alarm limit value violation: Month							
DBBn + 32	Sensor n Last alarm limit value violation: Day							
DBBn + 33	Sensor n Last alarm limit value violation: Hour							
DBBn + 34	Sensor n Last alarm limit value violation: Minute							
DBBn + 35	Sensor n Last alarm limit value violation: Seconds							

DB1002 /SCE/	SENTRON PAC							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBDn + 36	Sensor n Duration of the fault limit value violation							

Spindle/temperature sensor instances

Spindle_1, Temperature_sensor_1	(n=0):	DB1002 DBB00…DBB39
Spindle_1, Temperature_sensor_2	(n=40):	DB1002 DBB40…DBB79
Spindle_1, Temperature_sensor_3	(n=80):	DB1002 DBB80…DBB119
Spindle_1, Temperature_sensor_4	(n=120):	DB1002 DBB120…DBB159
Spindle_1, Temperature_sensor_5	(n=160):	DB1002 DBB160…DBB199
Spindle_1, Temperature_sensor_6	(n=200):	DB1002 DBB200…DBB239
Spindle_2, Temperature_sensor_1	(n=240):	DB1002 DBB240…DBB279
Spindle_2, Temperature_sensor_2	(n=280):	DB1002 DBB280…DBB319
Spindle_2, Temperature_sensor_3	(n=320):	DB1002 DBB320…DBB359
Spindle_2, Temperature_sensor_4	(n=360):	DB1002 DBB360…DBB399
Spindle_2, Temperature_sensor_5	(n=400):	DB1002 DBB400…DBB439
Spindle_2, Temperature_sensor_6	(n=440):	DB1002 DBB440…DBB479

18.1.20 Interface to the tool management, extended area

18.1.20.1 DB1071, interface for loading/unloading the magazine Multitool

Table 18-107 DB1071, interface for loading/unloading the magazine Multitool

DB1071 /FBWsl/	Loading/unloading positions (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBWn + 0	Type of distance coding of the multitool (corresponds to $TC_MTP_KD) see also (Page 728)							
DBWn + 2	Multitool location number Number of locations of the multitool see also (Page 729)							
DBWn + 4	Multitool location distance see also (Page 729)							
DBWn + 8	Multitool number see also (Page 730)							

DB1071 /FBWsl/	Loading/unloading positions (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 10	Multitool location number see also (Page 730)							
DBWn + 12	Tool holder see also (Page 730)							
DBWn + 14	Reserved							
DBWn + 16	Reserved							
DBWn + 18	Reserved							

18.1.20.2 DB1072, interface for the spindle: Multitool

Table 18-108 DB1072, interface for the spindle: Multitool

DB1072 /FBWsl/	Spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 0	Distance coding (corresponds to to $TC_MTP_KD) see also (Page 731)							
DBWn + 2	Multitool location number see also (Page 731)							
DBWn + 4	Multitool location distance see also (Page 731)							
DBWn + 8	Multitool number (new tool) see also (Page 732)							
DBWn + 10	Multitool location number (new tool) see also (Page 732)							
DBWn + 12	Multitool number (old tool) see also (Page 732)							
DBWn + 14	Multitool location number (old tool) see also (Page 733)							
DBWn + 16	New tool: Location type see also (Page 733)							
DBWn + 18	New tool: Size left see also (Page 733)							
DBWn + 20	New tool: Size right see also (Page 734)							
DBWn + 22	New tool: Size top see also (Page 734)							
DBWn + 24	New tool: Size bottom see also (Page 734)							
DBWn + 26	Tool status for new tool (corresponds to $TC_TP8[T_Nr]) see also (Page 735)							

DB1072 /FBWsl/	Spindle (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBWn + 28	New tool: Internal T number of the NC see also (Page 735)							
DBWn + 30	Tool holder Spindle or toolholder number to which the change refers see also (Page 736)							
DBWn + 32	Original magazine of new tool (corresponds to NC variables $A_MYMN[T No]) If the new tool is located in the magazine, this value is identical with DB72 DBW(n + 20 (Page 419)). see also (Page 736)							
DBWn + 34	Original location of new tool (corresponds to NC variables $A_MYMLN[T No]) If the new tool is located in the magazine, this value is identical with DB72 DBW(n + 22). see also (Page 736)							
DBWn + 36 - DBWn + 48	Reserved							

18.1.20.3 DB1073, interface for the turret: Multitool

Table 18-109 DB1073, interface for the turret: Multitool

DB1073 /FBWsl/	Turret (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBWn + 0	Distance coding (corresponds to to $TC_MTP_KD) see also (Page 737)							
DBWn + 2	Multitool location number see also (Page 737)							
DBWn + 4	Multitool location distance see also (Page 738)							
DBWn + 8	Multitool number (new tool) see also (Page 738)							
DBWn + 10	Multitool location number (new tool) see also (Page 738)							
DBWn + 12	Multitool number (old tool) see also (Page 739)							
DBWn + 14	Multitool location number (old tool) see also (Page 739)							
DBWn + 16	Location type see also (Page 739)							
DBWn + 18	New tool: Size left see also (Page 740)							

DB1073 /FBWsl/	Turret (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 20	New tool: Size right see also (Page 740)							
DBWn + 22	New tool: Size top see also (Page 740)							
DBWn + 24	New tool: Size bottom see also (Page 741)							
DBWn + 26	Tool status for new tool (corresponds to parameter $TC_TP8[T_No]) see also (Page 741)							
DBWn + 28	New tool: Internal T number of the NC see also (Page 742)							
DBWn + 30	Tool holder see also (Page 742)							
DBWn + 32	Original magazine of new tool (corresponds to NC variables $A_MYMN[T No]) If the new tool is located in the magazine, this value is identical with DB73 DBW (Page 421)(n + 20). see also (Page 742)							
DBWn + 34	Original location of new tool (corresponds to NC variables $A_MYMN[T No]) If the new tool is located in the magazine, this value is identical with DB73 DBW (Page 421)(n + 22). see also (Page 743)							
DBWn + 36 - DBWn + 48	Reserved							

18.2 Interface signals - detailed description

18.2.1 DB10: NC, PLC and HMI

18.2.1.1 DB10 DBX0.0 - 7 (digital NC inputs 1 - 8: Lock)

DB10 DBX0.0 - 7	Digital NC inputs 1 - 8: Lock
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The digital NC input is locked.
Signal state 0	The digital NC input is enabled.

DB10 DBX0.0 - 7	Digital NC inputs 1 - 8: Lock							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 8 [2]	Input 7 [2]	Input 6 [2]	Input 5 [2]	Input 4 [1]	Input 3 [1]	Input 2 [1]	Input 1 [1]
	1) Onboard inputs of the NCU 2) Inputs without hardware							
	Note When read, a locked input supplies a value of 0.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.2 DB10 DBX1.0 - 7 (digital NC inputs 1 - 8: Set)

DB10 DBX1.0 - 7	Digital NC inputs 1 - 8: Set							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	Set NC input value to a defined value of 1.							
Signal state 0	Do not influence NC input value.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 8 [2]	Input 7 [2]	Input 6 [2]	Input 5 [2]	Input 4 [1]	Input 3 [1]	Input 2 [1]	Input 1 [1]
	1) Onboard inputs of the NCU 2) Inputs without hardware							
	If the PLC user program sets the NC input to a value of 1, then the signal state at the onboard input of the NCU - as well as the lock of the NC input - have no effect.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.3 DB10 DBX4.0 - 7 (digital NC outputs 1 - 8: Lock)

DB10 DBX4.0 - 7	Digital NC outputs 1 - 8: Lock
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The digital NC output is locked.
Signal state 0	The digital NC output is enabled.

DB10 DBX4.0 - 7	Digital NC outputs 1 - 8: Lock							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	If the digital NC output is locked, then 0 V (defined) is output at the hardware output. If the digital NC output is not locked, then the value specified in the NC program or by the PLC program is output at the hardware output.							
Corresponds to	DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite screen form) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) DB10 DBB7, 133, 137, 141, 145 (digital NC outputs: Target screen form) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.4 DB10 DBX5.0 - 7 (digital NC outputs 1 - 8: Overwrite)

DB10 DBX5.0 - 7	Digital NC outputs 1 - 8: Overwrite							
Signal flow	PLC → NC							
Update	Cyclic							
Edge change 0 → 1	The associated "setting value" is activated.							
Edge change 1 → 0	No effect.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	For a positive edge change 0 → 1, then for the corresponding output, instead of the value written using system variables $A_OUT, the setting value specified from the PLC user program is used. The value written via system variables $A_OUT is then lost. For a negative edge change 1 → 0, for the corresponding output, the actual value at the hardware output is kept.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.5 DB10 DBX6.0 - 7 (digital NC outputs 1 - 8: Setting value)

DB10 DBX6.0 - 7	Digital NC outputs 1 - 8: Setting value
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Value of the setting value is 1.

DB10 DBX6.0 - 7	Digital NC outputs 1 - 8: Setting value							
Signal state 0	Value of the setting value is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	Using the setting value, the PLC user program can specify a defined output value. In order that the setting value becomes active, it must be activated from the overwrite screen form or the setting screen form.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.6 DB10 DBX7.0 - 7 (digital NC outputs 1 - 8: Target)

DB10 DBX7.0 - 7	Digital NC outputs 1 - 8: Target							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The target is active.							
Signal state 0	The target is **not** active.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	If a bit is set, then for the corresponding output, instead of the NC output value, the setting value specified by the PLC user program is used. The actual NC output value is kept. If a bit is reset, then for the corresponding output, the last NC output value becomes active again.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.7 DB10 DBX56.1 (Emergency Stop)

DB10 DBX56.1	Emergency Stop
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Emergency Stop is requested.
Signal state 0	Emergency Stop is **not** requested.
Further information	All machine axes are braked in the relevant axis-specific parameterized time: MD36610 $MA_AX_EMERGENCY_STOP_TIME
Corresponds to	DB10 DBX56.2 (acknowledge Emergency Stop) DB10 DBX106.1 (Emergency Stop active) MD36610 $MA_AX_EMERGENCY_STOP_TIME (time of braking ramp in event of errors)
Further information	Function Manual Basic Functions

18.2.1.8 DB10 DBX56.2 (acknowledge Emergency Stop)

DB10 DBX56.2	Acknowledge Emergency Stop
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Acknowledgment of the "Emergency Stop" state is requested.
Signal state 0	Acknowledgment of the "Emergency Stop" state is not requested.
Further information	To acknowledge the "Emergency Stop" state of the NC, the following interface signals must remain set until interface signal DB10 DBX106.1 (Emergency Stop active) has been reset: • DB10 DBX56.2 = 1 (acknowledge Emergency Stop) • DB11, ... DBX0.7 = 1 (mode group reset) for all mode groups of the NC
Corresponds to	DB10 DBX56.1 (Emergency Stop) DB10 DBX106.1 (Emergency Stop active) DB11 DBX0.7 (mode group reset)
Further information	Function Manual Basic Functions

18.2.1.9 DB10 DBX56.4 - 7 (key-operated switch position 0 - 3)

DB10 DBX56.4 - 7	Key-operated switch position 0 - 3
Signal flow	PLC → NC
Update	Cyclic

<table>
<tr><th>DB10 DBX56.4 - 7</th><th colspan="5">Key-operated switch position 0 - 3</th></tr>
<tr><td>Additional information</td><td colspan="5">Depending on the key-operated switch position, access to certain elements in the NC can be enabled or disabled.
• Key-operated switch position 0 represents the lowest access rights
• Key-operated switch position 3 represents the highest access rights
The interface signals of key-operated switch positions 1 to 3 can either be directly specified from the key-operated switch on the machine control panel or from the PLC user program.
It is only permissible to set one bit. If several bits are set simultaneously, the control internally activates switch position 3.
<table>
<tr><th>Key-operated switch position</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>2</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>Corresponds with ...</td><td colspan="5">Machine data for access levels: MD11612, MD51044 - MD51064, MD51070 - MD51073, MD51199 - MD51211, MD51215 - MD51225, MD51235
Disabling using a password</td></tr>
</table>

18.2.1.10 DB10 DBX58.0 - 7 (collision avoidance: Deactivate protection area group)

DB10 DBX58.0 - 7	Collision avoidance: Deactivate protection area group
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The deactivation of all protection areas of the protection area type in the selected operating mode is requested.
Signal state 0	The deactivation of all protection areas of the protection area type in the selected operating mode is **not** requested.

DB10 DBX58.0 - 7	Collision avoidance: Deactivate protection area group		
Further information	Bit	Operating mode	Protection area type [1]
	0	AUTOMATIC	Machine
	1		Tool
	2		Workholder
	3		Workpiece
	4	JOG	Machine
	5		Tool
	6		Workholder
	7		Workpiece
	[1] Type of a protection area ($NP_PROT_TYPE) **Note** The deactivation of a protection area group is selected from the SINUMERIK Operate user interface in the operating area "AUTOMATIC", "JOG" or "MDI" > "ETC key (">")" > "Settings" > "Collision avoidance" > "Switch collision avoidance on and off" by setting the HMI/PLC interface signal DB10 DBX93.0 ... 7 of the protection area group. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB10 DBX58.0 ... 7: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE". **Note** If manual traversing has been enabled in AUTOMATIC mode (MD10735 with bit 0 == 1: allow jogging in the automatic mode) the settings for the JOG mode also apply in AUTOMATIC mode during manual traversing.		
Corresponds to	DB10 DBX93.0 - 7 (deactivate collision avoidance) $NP_PROT_TYPE (type of protection area) MD10735 $MN_JOG_MODE_MASK (settings for the JOG mode)		
Further information	• Operating Manual Turning or Milling • Function Manual Monitoring and Compensating • Function Manual Monitoring and Compensating		

18.2.1.11 DB10 DBX60.0 - 7 (digital NC inputs 1 - 8: Actual value)

DB10 DBX60.0 - 7	Digital NC inputs 1 - 8: Actual value
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The actual value is 1.
Signal state 0	The actual value is 0.

DB10 DBX60.0 - 7	Digital NC inputs 1 - 8: Actual value							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
	The actual NC output value can be read in the PLC user program via the actual value.							
	Note The value in the "Actual value" interface can, as a result of the various subsequent influencing possibilities "Lock" and "Set", have a different value than that available at the NC output.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.12 DB10 DBX64.0 - 7 (digital NC outputs 1 - 8: Setpoint)

DB10 DBX64.0 - 7	Digital NC outputs 1 - 8: Setpoint							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The setpoint is 1.							
Signal state 0	The setpoint is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	The actual NC output value can be read in the PLC user program via the setpoint.							
	Note The value in the "Setpoint" interface can, as a result of the various subsequent influencing possibilities "Target" and "Lock", have a different value than that available at the NC output.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.13 DB10 DBX93.0 ... 7 (collision avoidance: Deactivate protection area group)

DB10 DBX93.0 ... 7	Collision avoidance: Deactivate protection area group
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	The deactivation of all protection areas of the protection area type in the selected operating mode is requested.
Signal state 0	The deactivation of all protection areas of the protection area type in the selected operating mode is **not** requested.

<table>
<tr><th>DB10 DBX93.0 ... 7</th><th colspan="3">Collision avoidance: Deactivate protection area group</th></tr>
<tr><td rowspan="11">Further information</td><td>Bit</td><td>Operating mode</td><td>Protection area type [1]</td></tr>
<tr><td>0</td><td rowspan="4">AUTOMATIC</td><td>Machine</td></tr>
<tr><td>1</td><td>Tool</td></tr>
<tr><td>2</td><td>Workholder</td></tr>
<tr><td>3</td><td>Workpiece</td></tr>
<tr><td>4</td><td rowspan="4">JOG</td><td>Machine</td></tr>
<tr><td>5</td><td>Tool</td></tr>
<tr><td>6</td><td>Workholder</td></tr>
<tr><td>7</td><td>Workpiece</td></tr>
<tr><td colspan="3">[1] Type of a protection area ($NP_PROT_TYPE)
Note
The deactivation of a protection area group is selected from the SINUMERIK Operate user interface in the operating area "AUTOMATIC", "JOG" or "MDI" > "ETC key (">")" > "Settings" > "Collision avoidance" > "Switch collision avoidance on and off" by setting the HMI/PLC interface signal DB10 DBX93.0 ... 7 of the protection area group.
The interface signal is transferred according to the value of the FB1 parameter <code>MMCToIf</code> by the basic PLC program into the NC/PLC interface signal DB10 DBX58.0 ... 7:
• "TRUE": Transmission
• "FALSE": No transmission
The default value of the parameter is "TRUE".
Note
If manual traversing has been enabled in AUTOMATIC mode (MD10735 with bit 0 == 1: allow jogging in the automatic mode) the settings for the JOG mode also apply in AUTOMATIC mode during manual traversing.</td></tr>
<tr><td>Corresponds to</td><td colspan="3">DB10 DBX58.0 ... 7 (collision avoidance: Deactivate protection area group)
$NP_PROT_TYPE (type of protection area)
MD10735 $MN_JOG_MODE_MASK (settings for the JOG mode)</td></tr>
<tr><td>Further information</td><td colspan="3">• Operating Manual Turning or Milling
• Function Manual Monitoring and Compensating
• Function Manual Monitoring and Compensating</td></tr>
</table>

18.2.1.14 DB10 DBX97.0 - 3 (channel number geometry axis handwheel 1)

DB10 DBX97.0 - 3	Channel number for geometry axis, handwheel 1
Signal flow	HMI → PLC
Update	Cyclic

<table>
<tr><th>DB10 DBX97.0 - 3</th><th>Channel number for geometry axis, handwheel 1</th></tr>
<tr><td>Further information</td><td>The operator can assign an axis to the handwheel (1, 2, 3) directly at the operator panel. If this axis is a geometry axis (interface signal "Machine axis handwheel <n>" = 0), then the basic PLC program provides the corresponding channel number at the HMI interface as binary-coded value.
Example:
<table>
<tr><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Channel number</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
</table>
Note
For machine axes (interface signal "Machine axis handwheel <n>" = 1), interface signal "Channel number geometry axis handwheel <n>" has no significance.</td></tr>
<tr><td>Corresponds to</td><td>DB10 DBX98.0 - 3 (channel number geometry axis handwheel 2)
DB10 DBX99.0 - 3 (channel number geometry axis handwheel 3)
DB10 DBX100.0 - 4 (axis number handwheel 1)
DB10 DBX101.0 - 4 (axis number handwheel 2)
DB10 DBX102.0 - 4 (axis number handwheel 3)
DB10 DBX100.6 (handwheel 1 selected)
DB10 DBX101.6 (handwheel 2 selected)
DB10 DBX102.6 (handwheel 3 selected)
DB10 DBX100.7 (machine axis handwheel 1)
DB10 DBX101.7 (machine axis handwheel 2)
DB10 DBX102.7 (machine axis handwheel 3)
DB21, ... DBX12.0 - 2 (geometry axis 1: activate handwheel)
DB21, ... DBX16.0 - 2 (geometry axis 2: activate handwheel)
DB21, ... DBX20.0 - 2 (geometry axis 3: activate handwheel)
DB31, ... DBX4.0 - 2 (activate handwheel)</td></tr>
<tr><td>Further information</td><td>Function Manual Axes and Spindles</td></tr>
</table>

18.2.1.15 DB10 DBX98.0 - 3 (channel number geometry axis handwheel 2)

DB10 DBX98.0 - 3	Channel number for geometry axis, handwheel 2
Additional information	See DB10 DBB97 (channel number geometry axis handwheel 1) (Page 442).

18.2.1.16 DB10 DBX99.0 - 3 (channel number geometry axis handwheel 3)

DB10 DBX99.0 - 3	Channel number for geometry axis, handwheel 3
Additional information	See DB10 DBB97 (channel number geometry axis handwheel 1) (Page 442).

18.2.1.17 DB10 DBX100.0 - 4 (axis number handwheel 1)

<table>
<tr><th>DB10 DBX100.0 - 4</th><th>Axis number handwheel 1</th></tr>
<tr><td>Signal flow</td><td>HMI → PLC</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Further information</td><td>The operator can assign an axis to the handwheel (1, 2, 3) directly at the operator panel. To do this, it specifies the requested axis (e.g. X). The basic PLC program provides the axis number, associated with the axis, at the HMI interface as binary-coded value.
Example:
<table>
<tr><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Axis number</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr>
</table>
The following applies when assigning the axis identifier to the axis number:
• Machine axis (interface signal "Machine axis" = 1):
→ The assignment is realized using machine data MD10000.
• Geometry axis (interface signal "Machine axis" = 0):
→ The assignment is realized using machine data MD20060.</td></tr>
<tr><td>Corresponds to</td><td>DB10 DBX101.0 - 4 (axis number handwheel 2)
DB10 DBX102.0 - 4 (axis number handwheel 3)
DB10 DBX97.0 - 3 (channel number geometry axis handwheel 1)
DB10 DBX98.0 - 3 (channel number geometry axis handwheel 2)
DB10 DBX99.0 - 3 (channel number geometry axis handwheel 3)
DB10 DBX100.6 (handwheel 1 selected)
DB10 DBX101.6 (handwheel 2 selected)
DB10 DBX102.6 (handwheel 3 selected)
DB10 DBX100.7 (machine axis handwheel 1)
DB10 DBX101.7 (machine axis handwheel 2)
DB10 DBX102.7 (machine axis handwheel 3)
DB21, ... DBX12.0 - 2 (geometry axis 1: activate handwheel)
DB21, ... DBX16.0 - 2 (geometry axis 2: activate handwheel)
DB21, ... DBX20.0 - 2 (geometry axis 3: activate handwheel)
DB31, ... DBX4.0 - 2 (activate handwheel)
MD10000 $MN_AXCONF_MACHAX_NAME_TAB [<n>] (machine axis name)
MD20060 $MC_AXCONF_GEOAX_NAME_TAB [<n>] (name of the geometry axis in the channel)</td></tr>
<tr><td>Further information</td><td>Function Manual Axes and Spindles</td></tr>
</table>

18.2.1.18 DB10 DBX100.5 (define handwheel 1 as contour handwheel)

DB10 DBX100.5	Define handwheel 1 as contour handwheel
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	The handwheel is defined as contour handwheel via the operator interface.
Signal state 0	The handwheel is not defined as contour handwheel.

DB10 DBX100.5	Define handwheel 1 as contour handwheel
Further information	In order that the handwheel defined via the user interface is also active as contour handwheel, signal "Define handwheel <n> as contour handwheel" must be linked to signal "Activate handwheel <n> as contour handwheel". **Note** Depending on the settings of parameter HWheelMMC in FB1 of the basic PLC program, these signals are either supplied by the basic program or must be supplied by the PLC user program.
Corresponds to	DB10 DBX101.5 (define handwheel 2 as contour handwheel) DB10 DBX102.5 (define handwheel 3 as contour handwheel) DB21 ... DBX30.0 - 2 (activate contour handwheel) FB1 parameter "HWheelMMC"
Further information	Function Manual Axes and Spindles

18.2.1.19 DB10 DBX100.6 (handwheel 1 selected)

DB10 DBX100.6	Handwheel 1 selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Enable handwheel for activation.
Signal state 0	Lock handwheel.
Further information	The basic PLC program provides this interface signal at the HMI interface if, at the operator panel, the operator selects or deselects handwheel 1 for the specified axis, i.e. enables or locks for activation. The information is transferred from the basic PLC program to the PLC (precondition: FB1 parameter "HWheelMMC" == "TRUE") and the interface signal "Activate handwheel" appropriately set for the specified axis. Depending on the setting in the HMI interface signal "machine axis", either the interface for the geometry axis or for the machine axis is used.
Corresponds to	DB10 DBX101.6 (handwheel 2 selected) DB10 DBX102.6 (handwheel 3 selected) DB10 DBX97.0 - 2 (channel number geometry axis handwheel 1) DB10 DBX98.0 - 2 (channel number geometry axis handwheel 2) DB10 DBX99.0 - 2 (channel number geometry axis handwheel 3) DB10 DBX100.0 - 4 (axis number handwheel 1) DB10 DBX101.0 - 4 (axis number handwheel 2) DB10 DBX102.0 - 4 (axis number handwheel 3) DB10 DBX100.7 (machine axis handwheel 1) DB10 DBX101.7 (machine axis handwheel 2) DB10 DBX102.7 (machine axis handwheel 3) DB21, ... DBX12.0 - 2 (geometry axis 1: activate handwheel) DB21, ... DBX16.0 - 2 (geometry axis 2: activate handwheel) DB21, ... DBX20.0 - 2 (geometry axis 3: activate handwheel) DB31, ... DBX4.0 - 2 (activate handwheel) FB1 parameter "HWheelMMC"
Further information	Function Manual Axes and Spindles

18.2.1.20 DB10 DBX100.7 (machine axis handwheel 1)

DB10 DBX100.7	Machine axis handwheel 1
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Axis is a machine axis.
Signal state 0	Axis is a geometry axis.
Further information	The basic PLC program provides this interface signal at the HMI interface if the operator has assigned an axis to the handwheel (1, 2, 3) directly at the operator panel.
Corresponds to	DB10 DBX101.7 (machine axis handwheel 2) DB10 DBX102.7 (machine axis handwheel 3) DB10 DBX97.0 - 2 (channel number geometry axis handwheel 1) DB10 DBX98.0 - 2 (channel number geometry axis handwheel 2) DB10 DBX99.0 - 2 (channel number geometry axis handwheel 3) DB10 DBX100.0 - 4 (axis number handwheel 1) DB10 DBX101.0 - 4 (axis number handwheel 2) DB10 DBX102.0 - 4 (axis number handwheel 3) DB10 DBX100.6 (handwheel 1 selected) DB10 DBX101.6 (handwheel 2 selected) DB10 DBX102.6 (handwheel 3 selected)
Further information	Function Manual Axes and Spindles

18.2.1.21 DB10 DBX101.0 - 4 (axis number handwheel 2)

DB10 DBX101.0 - 4	Axis number handwheel 2
Additional information	See DB10 DBX100.0 - 4 (axis number handwheel 1) (Page 444).

18.2.1.22 DB10 DBX101.5 (define handwheel 2 as contour handwheel)

DB10 DBX101.5	Define handwheel 2 as contour handwheel
Additional information	See DB10 DBX100.5 (define handwheel 1 as contour handwheel) (Page 444).

18.2.1.23 DB10 DBX101.6 (handwheel 2 selected)

DB10 DBX101.6	Handwheel 2 selected
Additional information	See DB10 DBX100.6 (handwheel 1 selected) (Page 445).

18.2.1.24 DB10 DBX101.7 (machine axis handwheel 2)

DB10 DBX101.7	Machine axis handwheel 2
Additional information	See DB10 DBX100.7 (machine axis handwheel 1) (Page 446).

18.2.1.25 DB10 DBX102.0 - 4 (axis number handwheel 3)

DB10 DBX102.0 - 4	Axis number handwheel 3
Additional information	See DB10 DBX100.0 - 4 (axis number handwheel 1) (Page 444).

18.2.1.26 DB10 DBX102.5 (define handwheel 3 as contour handwheel)

DB10 DBX102.5	Define handwheel 3 as contour handwheel
Additional information	See DB10 DBX100.5 (define handwheel 1 as contour handwheel) (Page 444).

18.2.1.27 DB10 DBX102.6 (handwheel 3 selected)

DB10 DBX102.6	Handwheel 3 selected
Additional information	See DB10 DBX100.6 (handwheel 1 selected) (Page 445).

18.2.1.28 DB10 DBX102.7 (machine axis handwheel 3)

DB10 DBX102.7	Machine axis handwheel 3
Additional information	See DB10 DBX100.7 (machine axis handwheel 1) (Page 446).

18.2.1.29 DB10 DBX103.0 (remote diagnosis active)

DB10 DBX103.0	Remote diagnosis active
Signal flow	HMI → PLC
Update	Cyclically
Signal state 1	Remote diagnosis (optional) is active, i.e. the control is operated via an external PC.
Signal state 0	Remote diagnosis is not active.

18.2.1.30 DB10 DBX103.5 (AT box ready)

DB10 DBX103.5	AT box ready
Signal flow	HMI → PLC
Update	Cyclically

DB10 DBX103.5	AT box ready
Signal state 1	The AT box for expansion modules is ready.
Signal state 0	The AT box is not ready. An expansion module conforming to the AT specification has either no functionality or restricted functionality.

18.2.1.31 DB10 DBX103.6 (HMI temperature limit)

DB10 DBX103.6	HMI temperature limit
Signal flow	HMI → PLC
Update	Cyclically
Signal state 1	The temperature is **within** the permissible tolerance range from 5 to 55 °C.
Signal state 0	The temperature is **outside** the permissible tolerance range from 5 to 55 °C. The temperature monitoring has responded and the PCU has been disabled.

18.2.1.32 DB10 DBX103.7 (HMI battery alarm)

DB10 DBX103.7	HMI battery alarm
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	The battery monitor has responded. Power failure can cause the loss of recently changed data and a correct device configuration. An appropriate alarm is issued. The buffer battery should be checked. An insufficient battery voltage also affects the current time on the user interface.
Signal state 0	No HMI battery alarm is present.
Further information	Operator Components Manual (PCU)

18.2.1.33 DB10 DBX104.7 (NC-CPU ready)

DB10 DBX104.7	NC CPU ready
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The NC CPU is ready and registers itself cyclically with the PLC. After a correct initial start and the first complete OB1 cycle, the PLC and NC continuously exchange sign-of-life signals.
Signal state 0	NC CPU is not ready for operation. The following measures are started by the basic PLC program: • The status signals of the NC to the PLC (user interface) are deleted • The change signals of auxiliary functions are deleted • Cyclic processing of the user interface PLC to NC is terminated.
Further information	• Diagnostics Manual

18.2.1.34 DB10 DBX106.1 (Emergency Stop active)

DB10 DBX106.1	Emergency Stop active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Emergency Stop is active.
Signal state 0	Emergency Stop is **not** active.
Further information	To acknowledge the "Emergency Stop" state of the NC, the following interface signals must remain set until interface signal DB10 DBX106.1 (Emergency Stop active) has been reset: • DB10 DBX56.2 = 1 (acknowledge Emergency Stop) • DB11, ... DBX0.7 = 1 (mode group reset) for all mode groups of the NC
Corresponds to	DB10 DBX56.1 (Emergency Stop) DB10 DBX56.2 (acknowledge Emergency Stop) DB11 DBX0.7 (mode group reset)
Further information	Function Manual Basic Functions

18.2.1.35 DB10 DBX107.0 - 1 (probe actuated)

DB10 DBX107.0 - 1	Probe actuated	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	Probe deflected.	
Signal state 0	Probe **not** deflected.	
Further information	Bit 0	Probe 1
	Bit 1	Probe 2
Further information	Function Manual Technologies	

18.2.1.36 DB10 DBX107.6 (NCU link active)

DB10 DBX107.6	NCU link: Active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	NCU link communication is active.
Signal state 0	NCU link communication is **not** active.
Further information	The signal is irrelevant for a system with an NCU.
Further information	• Function Manual Basic Functions • NCU Equipment Manual

18.2.1.37 DB10 DBX108.3 (operating software ready)

DB10 DBX108.3	Operating software ready
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	SINUMERIK Operate is ready and registers itself cyclically with the NC.
Signal state 0	SINUMERIK Operate is not ready.
Further information	Diagnostics Manual

18.2.1.38 DB10 DBX108.5 (drives in cyclic operation)

DB10 DBX108.5	Drives in cyclic operation
Signal flow	NC → PLC
Update	Cyclically
Signal state 1	For all machine axes of the NC, the corresponding drives are in cyclic operation, i.e. they cyclically exchange PROFIdrive telegrams with the NC.
Signal state 0	For at least one machine axis of the NC, the corresponding drive is not in cyclic operation, i.e. it is not cyclically exchanging PROFIdrive telegrams with the NC.

18.2.1.39 DB10 DBX108.6 (drive ready)

DB10 DBX108.6	Drive ready
Signal flow	NC → PLC
Update	Cyclically
Signal state 1	For **all** machine axes of the NC, the corresponding drives are ready: DB31, ... DBX93.5 == 1 (DRIVE ready)
Signal state 0	For at least one machine axis of the NC, the corresponding drive is **not** ready: DB31, ... DBX93.5 == 0 (DRIVE ready)
Corresponds with	DB31, ... DBX93.5 (DRIVE ready)

18.2.1.40 DB10 DBX108.7 (NC ready)

DB10 DBX108.7	NC ready
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The control is ready. The interface signal is an image of the relay contact "NC Ready". The signal is set when: • Relay contact "NC Ready": closed • All internal control voltages have been established • Control state: Cyclic operation

DB10 DBX108.7	NC ready
Signal state 0	The control is not ready. The signal is set when: • Relay contact "NC Ready": open Possible causes: • The undervoltage or overvoltage monitoring function has responded • Individual components are not ready (NC CPU ready) • The NC CPU watchdog has responded If still possible, the following internal control responses occur: • NC: The controller enable signals are withdrawn ⇒ the drives stop • PLC basic program: – The status signals from the NC to the PLC (user interface) are deleted – The change signals of the auxiliary functions are deleted – Cyclic processing of the user interface PLC to NC is terminated Remedy: A warm restart must be initiated to exit the fault state.
Corresponds to	DB10 DBX104.7(NC CPU ready)
Further information	Diagnostics Manual

18.2.1.41 DB10 DBX109.0 (NC alarm active)

DB10 DBX109.0	NC alarm is active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	At least one NC alarm is active.
Signal state 0	No NC alarm is active.
Further information	The interface signal is a combination of interface signals of all of the existing channels: DB21, ... DBX36.6 (channel-specific NC alarm active)
Corresponds to	DB21, ... DBX36.6 (channel-specific NC alarm active) DB21, ... DBX36.7 (NC alarm with machining stop active)
Further information	Diagnostics Manual

18.2.1.42 DB10 DBX109.5 (heat sink temperature alarm, NCU)

DB10 DBX109.5	NCU heat sink temperature alarm
Signal flow	NC → PLC
Update	Cyclically
Signal state 1	The heat sink temperature limit of the NCU has been exceeded. Continuous NCU operation can no longer be guaranteed.
Signal state 0	The heat sink temperature limit of the NCU has not been exceeded.

18.2.1.43 DB10 DBX109.6 (air temperature alarm)

DB10 DBX109.6	Air temperature alarm
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The monitoring of the ambient temperature and/or the fan monitoring has responded. Possible causes: • Ambient temperature too high • The speed monitoring of the 24 VDC fan used to cool the module has responded. Possible remedy: Replace the fan and/or ensure that additional ventilation/cooling is provided.
Signal state 0	The monitoring of the ambient temperature and the fan monitoring has not responded.
Corresponds to	Relay contacts of the infeed/regenerative feedback unit: Terminal 5.1, 5.2 or 5.1, 5.3)
Further information	Diagnostics Manual

18.2.1.44 DB10 DBX109.7 (NC battery alarm)

DB10 DBX109.7	NC battery alarm
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The NC battery voltage monitoring function has responded. This may be due to the following causes: • The battery voltage is within the pre-warning limit range (approx. 2.7 to 2.9 V). • The battery voltage is below the pre-warning limit range (≤ 2.6 V). • When the controller powered-up, it was identified that the battery voltage was below the pre-warning limit range (≤ 2.6 V).
Signal state 0	The battery voltage is higher than the lower limit value (normal situation).
Further information	The NC battery should only be replaced while the NC is switched on to avoid data loss as there is no memory backup.
Further information	• Diagnostics Manual • NCU Equipment Manual

18.2.1.45 DB10 DBX110.0 - 113.7 (software cams: minus cam signal 1 to 32)

DB10 DBX110.0 - 113.7	Software cams: minus cam signal 1 to 32
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	**Linear axis** The minus cam signal switches from 0 to 1 when the axis passes over the minus cam in the negative direction. **Modulo rotary axis** The minus cam signal changes level in response to every positive edge of the plus cam signal.

DB10 DBX110.0 - 113.7	Software cams: minus cam signal 1 to 32
Signal state 0	**Linear axis** The minus cam signal switches from 1 to 0 when the axis passes over the minus cam in the positive axis direction. **Modulo rotary axis** The minus cam signal changes level in response to every positive edge of the plus cam signal.
Additional information	The switching edge of the minus cam signal 1 to 32 is generated depending on the traversing direction of the (round) axis, and is transferred in the IPO clock cycle to the PLC interface.
Corresponds with	DB10 DBX114.0 - 117.7 (software cams: plus cam signal 1 to 32) DB31, ... DBX2.0 (software cams: activation)
Additional information	Function Manual Axes and Spindles

18.2.1.46 DB10 DBX114.0 - 117.7 (software cams: plus cam signal 1 to 32)

DB10 DBX114.0 - 117.7	Software cams: plus cam signal 1 to 32
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	**Linear axis** The plus cam signal switches from 0 to 1 when the axis traverses the plus cam in the positive direction. **Modulo rotary axis** When passing over the minus cam in the positive axis direction, the plus cam signal switches from 0 to 1.
Signal state 0	**Linear axis** The plus cam signal switches from 1 to 0 when the axis traverses the plus cam in the negative direction. **Modulo rotary axis** When passing over the plus cam in the positive axis direction, the plus cam signal switches back from 1 to 0.
Further information	The switching edge of the plus cam signal 1 to 32 is generated depending on the traversing direction of the (round) axis, and is transferred in the IPO clock cycle to the PLC interface. **Note** The described response of the plus cam for modulo rotary axes applies under the condition: plus cam - minus cam < 180 degrees If this condition is not fulfilled or if the minus cam is set to a greater value than the plus cam, then the behavior of the plus cam signal is inverted. The response of the minus cam signal remains unchanged.
Corresponds to	DB10 DBX110.0 - 113.7 (software cams: minus cam signal 1 to 32) DB31, ... DBX2.0 (software cams: activation)
Further information	Function Manual Axes and Spindles

18.2.1.47 DB10 DBX122.0 - 7 (digital NC inputs 9 - 16: Lock)

DB10 DBX122.0 - 7	Digital NC inputs 9 - 16: Lock							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The digital NC input is locked.							
Signal state 0	The digital NC input is enabled.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 16	Input 15	Input 14	Input 13	Input 12	Input 11	Input 10	Input 9
	Note When read, a locked input supplies a value of 0.							
Corresponds to	DB10 DBB1, 123, 125, 127, 129 (digital NC inputs set from the PLC) DB10 DBB60, 186, 187, 188, 189 (actual value of the digital NC inputs) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.48 DB10 DBX123.0 - 7 (digital NC inputs 9 - 16: Set)

DB10 DBX123.0 - 7	Digital NC inputs 9 - 16: Set							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	Set NC input value to a defined value of 1.							
Signal state 0	Do not influence NC input value.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 16	Input 15	Input 14	Input 13	Input 12	Input 11	Input 10	Input 9
	If the PLC user program sets the NC input to a value of 1, then the signal state at the onboard input of the NCU - as well as the lock of the NC input - have no effect.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.49 DB10 DBX124.0 - 7 (digital NC inputs 17 - 24: Lock)

DB10 DBX124.0 - 7	Digital NC inputs 17 - 24: Lock
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The digital NC input is locked.
Signal state 0	The digital NC input is enabled.

DB10 DBX124.0 - 7	Digital NC inputs 17 - 24: Lock							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 24	Input 23	Input 22	Input 21	Input 20	Input 19	Input 18	Input 17
	Note When read, a locked input supplies a value of 0.							
Corresponds to	DB10 DBB1, 123, 125, 127, 129 (digital NC inputs set from the PLC) DB10 DBB60, 186, 187, 188, 189 (actual value of the digital NC inputs) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.50 DB10 DBX125.0 - 7 (digital NC inputs 17 - 24: Set)

DB10 DBX125.0 - 7	Digital NC inputs 17 - 24: Set							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	Set NC input value to a defined value of 1.							
Signal state 0	Do not influence NC input value.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 24	Input 23	Input 22	Input 21	Input 20	Input 19	Input 18	Input 17
	If the PLC user program sets the NC input to a value of 1, then the signal state at the onboard input of the NCU - as well as the lock of the NC input - have no effect.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.51 DB10 DBX126.0 - 7 (digital NC inputs 25 - 32: Lock)

DB10 DBX126.0 - 7	Digital NC inputs 25 - 32: Lock							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The digital NC input is locked.							
Signal state 0	The digital NC input is enabled.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 32	Input 31	Input 30	Input 29	Input 28	Input 27	Input 26	Input 25
	Note When read, a locked input supplies a value of 0.							
Corresponds to	DB10 DBB1, 123, 125, 127, 129 (digital NC inputs set from the PLC) DB10 DBB60, 186, 187, 188, 189 (actual value of the digital NC inputs) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.52 DB10 DBX127.0 - 7 (digital NC inputs 25 - 32: Set)

DB10 DBX127.0 - 7	Digital NC inputs 25 - 32: Set							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	Set NC input value to a defined value of 1.							
Signal state 0	Do not influence NC input value.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 32	Input 31	Input 30	Input 29	Input 28	Input 27	Input 26	Input 25
	If the PLC user program sets the NC input to a value of 1, then the signal state at the onboard input of the NCU - as well as the lock of the NC input - have no effect.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.53 DB10 DBX128.0 - 7 (digital NC inputs 33 - 40: Lock)

DB10 DBX128.0 - 7	Digital NC inputs 33 - 40: Lock							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The digital NC input is locked.							
Signal state 0	The digital NC input is enabled.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 40	Input 39	Input 38	Input 37	Input 36	Input 35	Input 34	Input 33
	Note When read, a locked input supplies a value of 0.							
Corresponds to	DB10 DBB1, 123, 125, 127, 129 (digital NC inputs set from the PLC) DB10 DBB60, 186, 187, 188, 189 (actual value of the digital NC inputs) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.54 DB10 DBX129.0 - 7 (digital NC inputs 33 - 40: Set)

DB10 DBX129.0 - 7	Digital NC inputs 33 - 40: Set
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Set NC input value to a defined value of 1.
Signal state 0	Do not influence NC input value.

DB10 DBX129.0 - 7	Digital NC inputs 33 - 40: Set							
Further information	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Input 40	Input 39	Input 38	Input 37	Input 36	Input 35	Input 34	Input 33
	If the PLC user program sets the NC input to a value of 1, then the signal state at the onboard input of the NCU - as well as the lock of the NC input - have no effect.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.55 DB10 DBX130.0 - 7 (digital NC outputs 9 - 16: Lock)

DB10 DBX130.0 - 7	Digital NC outputs 9 - 16: Lock							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The digital NC output is locked.							
Signal state 0	The digital NC output is enabled.							
Additional information	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
	If the digital NC output is locked, then 0 V (defined) is output at the hardware output. If the digital NC output is not locked, then the value specified in the NC program or by the PLC program is output at the hardware output.							
Corresponds with	DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite screen form) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) DB10 DBB7, 133, 137, 141, 145 (digital NC outputs: Target screen form) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Additional information	Function Manual Basic Functions							

18.2.1.56 DB10 DBX131.0 - 7 (digital NC outputs 9 - 16: Overwrite)

DB10 DBX131.0 - 7	Digital NC outputs 9 - 16: Overwrite
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	The associated "setting value" is activated.
Edge change 1 → 0	No effect.

DB10 DBX131.0 - 7	Digital NC outputs 9 - 16: Overwrite							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
	For a positive edge change 0 → 1, then for the corresponding output, instead of the value written using system variables $A_OUT, the setting value specified from the PLC user program is used. The value written via system variables $A_OUT is then lost. For a negative edge change 1 → 0, for the corresponding output, the actual value at the hardware output is kept.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.57 DB10 DBX132.0 - 7 (digital NC outputs 9 - 16: Setting value)

DB10 DBX132.0 - 7	Digital NC outputs 9 - 16: Setting value							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	Value of the setting value is 1.							
Signal state 0	Value of the setting value is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
	Using the setting value, the PLC user program can specify a defined output value. In order that the setting value becomes active, it must be activated from the overwrite screen form or the setting screen form.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.58 DB10 DBX133.0 - 7 (digital NC outputs 9 - 16: Target)

DB10 DBX133.0 - 7	Digital NC outputs 9 - 16: Target							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The target is active.							
Signal state 0	The target is **not** active.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
	If a bit is set, then for the corresponding output, instead of the NC output value, the setting value specified by the PLC user program is used. The actual NC output value is kept. If a bit is reset, then for the corresponding output, the last NC output value becomes active again.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.59 DB10 DBX134.0 - 7 (digital NC outputs 17 - 24: Lock)

DB10 DBX134.0 - 7	Digital NC outputs 17 - 24: Lock							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The digital NC output is locked.							
Signal state 0	The digital NC output is enabled.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
	If the digital NC output is locked, then 0 V (defined) is output at the hardware output. If the digital NC output is not locked, then the value specified in the NC program or by the PLC program is output at the hardware output.							
Corresponds to	DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite screen form) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) DB10 DBB7, 133, 137, 141, 145 (digital NC outputs: Target screen form) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.60 DB10 DBX135.0 - 7 (digital NC outputs 17 - 24: Overwrite)

DB10 DBX135.0 - 7	Digital NC outputs 17 - 24: Overwrite							
Signal flow	PLC → NC							
Update	Cyclic							
Edge change 0 → 1	The associated "setting value" is activated.							
Edge change 1 → 0	No effect.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
	For a positive edge change 0 → 1, then for the corresponding output, instead of the value written using system variables $A_OUT, the setting value specified from the PLC user program is used. The value written via system variables $A_OUT is then lost. For a negative edge change 1 → 0, for the corresponding output, the actual value at the hardware output is kept.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.61 DB10 DBX136.0 - 7 (digital NC outputs 17 - 24: Setting value)

DB10 DBX136.0 - 7	Digital NC outputs 17 - 24: Setting value							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	Value of the setting value is 1.							
Signal state 0	Value of the setting value is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
	Using the setting value, the PLC user program can specify a defined output value. In order that the setting value becomes active, it must be activated from the overwrite screen form or the setting screen form.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							

DB10 DBX136.0 - 7	Digital NC outputs 17 - 24: Setting value
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
Further information	Function Manual Basic Functions

18.2.1.62 DB10 DBX137.0 - 7 (digital NC outputs 17 - 24: Target)

DB10 DBX137.0 - 7	Digital NC outputs 17 - 24: Target							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The target is active.							
Signal state 0	The target is **not** active.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
	If a bit is set, then for the corresponding output, instead of the NC output value, the setting value specified by the PLC user program is used. The actual NC output value is kept. If a bit is reset, then for the corresponding output, the last NC output value becomes active again.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.63 DB10 DBX138.0 - 7 (digital NC outputs 25 - 32: Lock)

DB10 DBX138.0 - 7	Digital NC outputs 25 - 32: Lock							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The digital NC output is locked.							
Signal state 0	The digital NC output is enabled.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
	If the digital NC output is locked, then 0 V (defined) is output at the hardware output. If the digital NC output is not locked, then the value specified in the NC program or by the PLC program is output at the hardware output.							

DB10 DBX138.0 - 7	Digital NC outputs 25 - 32: Lock
Corresponds to	DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite screen form) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) DB10 DBB7, 133, 137, 141, 145 (digital NC outputs: Target screen form) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
Further information	Function Manual Basic Functions

18.2.1.64 DB10 DBX139.0 - 7 (digital NC outputs 25 - 32: Overwrite)

<table>
<tr><th>DB10 DBX139.0 - 7</th><th colspan="8">Digital NC outputs 25 - 32: Overwrite</th></tr>
<tr><td>Signal flow</td><td colspan="8">PLC → NC</td></tr>
<tr><td>Update</td><td colspan="8">Cyclic</td></tr>
<tr><td>Edge change 0 → 1</td><td colspan="8">The associated "setting value" is activated.</td></tr>
<tr><td>Edge change 1 → 0</td><td colspan="8">No effect.</td></tr>
<tr><td rowspan="3">Further information</td><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>Output 32</td><td>Output 31</td><td>Output 30</td><td>Output 29</td><td>Output 28</td><td>Output 27</td><td>Output 26</td><td>Output 25</td></tr>
<tr><td colspan="8">For a positive edge change 0 → 1, then for the corresponding output, instead of the value written using system variables $A_OUT, the setting value specified from the PLC user program is used. The value written via system variables $A_OUT is then lost.
For a negative edge change 1 → 0, for the corresponding output, the actual value at the hardware output is kept.</td></tr>
<tr><td colspan="8">Note
The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by:
• DB10 DBB5, ... (overwrite) for edge change 0 → 1
• DB10 DBB7, ... (target) for signal state 1
Activating the two interfaces simultaneously should be avoided.</td></tr>
<tr><td>Corresponds to</td><td colspan="8">DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock)
DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite)
DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC)
MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS</td></tr>
<tr><td>Further information</td><td colspan="8">Function Manual Basic Functions</td></tr>
</table>

18.2.1.65 DB10 DBX140.0 - 7 (digital NC outputs 25 - 32: Setting value)

DB10 DBX140.0 - 7	Digital NC outputs 25 - 32: Setting value
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Value of the setting value is 1.
Signal state 0	Value of the setting value is 0.

DB10 DBX140.0 - 7	Digital NC outputs 25 - 32: Setting value							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
	Using the setting value, the PLC user program can specify a defined output value. In order that the setting value becomes active, it must be activated from the overwrite screen form or the setting screen form.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.66 DB10 DBX141.0 - 7 (digital NC outputs 25 - 32: Target)

DB10 DBX141.0 - 7	Digital NC outputs 25 - 32: Target							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The target is active.							
Signal state 0	The target is **not** active.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
	If a bit is set, then for the corresponding output, instead of the NC output value, the setting value specified by the PLC user program is used. The actual NC output value is kept. If a bit is reset, then for the corresponding output, the last NC output value becomes active again.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.67 DB10 DBX142.0 - 7 (digital NC outputs 33 - 40: Lock)

DB10 DBX142.0 - 7	Digital NC outputs 33 - 40: Lock							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The digital NC output is locked.							
Signal state 0	The digital NC output is enabled.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33
	If the digital NC output is locked, then 0 V (defined) is output at the hardware output. If the digital NC output is not locked, then the value specified in the NC program or by the PLC program is output at the hardware output.							
Corresponds to	DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite screen form) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) DB10 DBB7, 133, 137, 141, 145 (digital NC outputs: Target screen form) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.68 DB10 DBX143.0 - 7 (digital NC outputs 33 - 40: Overwrite)

DB10 DBX143.0 - 7	Digital NC outputs 33 - 40: Overwrite							
Signal flow	PLC → NC							
Update	Cyclic							
Edge change 0 → 1	The associated "setting value" is activated.							
Edge change 1 → 0	No effect.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33
	For a positive edge change 0 → 1, then for the corresponding output, instead of the value written using system variables $A_OUT, the setting value specified from the PLC user program is used. The value written via system variables $A_OUT is then lost. For a negative edge change 1 → 0, for the corresponding output, the actual value at the hardware output is kept.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.69 DB10 DBX144.0 - 7 (digital NC outputs 33 - 40: Setting value)

DB10 DBX144.0 - 7	Digital NC outputs 33 - 40: Setting value							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	Value of the setting value is 1.							
Signal state 0	Value of the setting value is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33
	Using the setting value, the PLC user program can specify a defined output value. In order that the setting value becomes active, it must be activated from the overwrite screen form or the setting screen form.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.70 DB10 DBX145.0 - 7 (digital NC outputs 33 - 40: Target)

DB10 DBX145.0 - 7	Digital NC outputs 33 - 40: Target							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The target is active.							
Signal state 0	The target is **not** active.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33
	If a bit is set, then for the corresponding output, instead of the NC output value, the setting value specified by the PLC user program is used. The actual NC output value is kept. If a bit is reset, then for the corresponding output, the last NC output value becomes active again.							
	Note The NC/PLC interface DBB6, ... (Digital NC outputs: setting value) is jointly used by: • DB10 DBB5, ... (overwrite) for edge change 0 → 1 • DB10 DBB7, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							

DB10 DBX145.0 - 7	Digital NC outputs 33 - 40: Target
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
Further information	Function Manual Basic Functions

18.2.1.71 DB10 DBX146.0 - 7 (analog NC inputs 1 - 8: Lock)

DB10 DBX146.0 - 7	Analog NC inputs 1 - 8: Lock							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The analog NC input is locked.							
Signal state 0	The analog NC input is enabled.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
	When read, a locked input supplies a value of 0.							
Corresponds to	DB10 DBB147 (analog NC inputs 1 - 8: Target) DB10 DBW148 - 162 (analog NC outputs 1 - 8: Setting value) DB10 DBB194 - 208 (analog NC outputs 1 - 8: Actual value) MD10300 $MN_FASTIO_ANA_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.72 DB10 DBX147.0 - 7 (analog NC inputs 1 - 8: Target)

DB10 DBX147.0 - 7	Analog NC inputs 1 - 8: Target							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The target of the setting value is activated.							
Signal state 0	The target of the setting value is **not** activated.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
	If the target of the setting value is activated, then the setting value is active as NC input value. If the setting value target is de-activated, then as NC input value, the analog value available at the NC input is active or for an active lock, the value of 0.							
Corresponds to	DBX146.0 - 7 (analog NC inputs 1 - 8: Lock) DB10 DBB147 (analog NC inputs 1 - 8: Target) DB10 DBW148 - 162 (analog NC outputs 1 - 8: Setting value) DB10 DBB194 - 208 (analog NC outputs 1 - 8: Actual value) MD10300 $MN_FASTIO_ANA_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.73 DB10 DBW148 - 162 (analog NC outputs 1 - 8: Setting value)

DB10 DBW148 - 162	Analog NC inputs 1 - 8: Setting value							
Signal flow	PLC → NC							
Update	Cyclic							
Further information	DBW162	DBW160	DBW158	DBW156	DBW154	DBW152	DBW150	DBW148
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
	The setting value is transferred as NC input value if input "Target" (DB10 DBB147) was set The setting value must be specified as a fixed point number (16 bit value including sign) in 2's complement.							
Corresponds to	DBX146.0 - 7 (analog NC inputs 1 - 8: Lock) DB10 DBB147 (analog NC inputs 1 - 8: Target) DB10 DBW148 - 162 (analog NC outputs 1 - 8: Setting value) DB10 DBB194 - 209 (analog NC inputs 1 - 8: Actual value) MD10300 $MN_FASTIO_ANA_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.74 DB10 DBX166.0 - 7 (analog NC outputs 1 - 8: Overwrite)

DB10 DBX166.0 - 7	Analog NC outputs 1 - 8: Overwrite							
Signal flow	PLC → NC							
Update	Cyclic							
Edge change 0 → 1	The associated "setting value" is activated.							
Edge change 1 → 0	No effect.							
Further information	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	For a positive edge change 0 → 1, then for the corresponding output, instead of the value written using system variables $A_OUTA, the setting value specified from the PLC user program is used. The value written via system variables $A_OUTA is then lost. For a negative edge change 1 → 0, for the corresponding output, the actual value at the hardware output is kept.							
	Note The NC/PLC interface DBW170, ... (Analog NC output 1 - 8: setting value) is jointly used by: • DB10 DBB166, ... (overwrite) for edge change 0 → 1 • DB10 DBB167, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBX167.0 - 7 (analog NC outputs 1 - 8: Target) DB10 DBX168.0 - 7 (analog NC outputs 1 - 8: Lock) DB10 DBW170 - 184 (analog NC outputs 1 - 8: Setting value) DB10 DBW210 - 224 (analog NC outputs 1 - 8: Setpoint) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.75 DB10 DBX167.0 - 7 (analog NC outputs 1 - 8: Target)

DB10 DBX167.0 - 7	Analog NC outputs 1 - 8: Target							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The target is active.							
Signal state 0	The target is **not** active.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	If a bit is set, then for the corresponding output, instead of the NC output value, the setting value specified by the PLC user program is used. The actual NC output value is kept. If a bit is reset, then for the corresponding output, the last NC output value becomes active again.							
	Note The NC/PLC interface DBW170, ... (Analog NC output 1 - 8: setting value) is jointly used by: • DB10 DBB166, ... (overwrite) for edge change 0 → 1 • DB10 DBB167, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBX166.0 - 7 (analog NC outputs 1 - 8: Overwrite) DB10 DBX168.0 - 7 (analog NC outputs 1 - 8: Lock) DB10 DBW170 - 184 (analog NC outputs 1 - 8: Setting value) DB10 DBW210 - 224 (analog NC outputs 1 - 8: Setpoint) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.76 DB10 DBX168.0 - 7 (analog NC outputs 1 - 8: Lock)

DB10 DBX168.0 - 7	Analog NC outputs 1 - 8: Target							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	The target is active.							
Signal state 0	The target is **not** active.							
Additional information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	If a bit is set, then for the corresponding output, instead of the NC output value, the setting value specified by the PLC user program is used. The actual NC output value is kept. If a bit is reset, then for the corresponding output, the last NC output value becomes active again.							
Corresponds with	DB10 DBX166.0 - 7 (analog NC outputs 1 - 8: Overwrite) DB10 DBX167.0 - 7 (analog NC outputs 1 - 8: Target) DB10 DBW170 - 184 (analog NC outputs 1 - 8: Setting value) DB10 DBW210 - 224 (analog NC outputs 1 - 8: Setpoint) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS							
Additional information	Function Manual Basic Functions							

18.2.1.77 DB10 DBW170 - 184 (analog NC outputs 1 - 8: Setting value)

DB10 DBW170 - 184	**Analog NC outputs 1 - 8: Setting value**							
Signal flow	PLC → NC							
Update	Cyclic							
Further information	**DBW184**	**DBW182**	**DBW180**	**DBW178**	**DBW176**	**DBW174**	**DBW172**	**DBW170**
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	Using the setting value, the PLC user program can specify a defined output value. In order that the setting value becomes active, it must be activated via the interface for "Override" or "Target".							
	Note The NC/PLC interface DBW170, ... (Analog NC output 1 - 8: setting value) is jointly used by: • DB10 DBB166, ... (overwrite) for edge change 0 → 1 • DB10 DBB167, ... (target) for signal state 1 Activating the two interfaces simultaneously should be avoided.							
Corresponds to	DB10 DBX166.0 - 7 (analog NC outputs 1 - 8: Overwrite) DB10 DBX167.0 - 7 (analog NC outputs 1 - 8: Target) DB10 DBX168.0 - 7 (analog NC outputs 1 - 8: Lock) DB10 DBW210 - 224 (analog NC outputs 1 - 8: Setpoint) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.78 DB10 DBX186.0 - 7 (digital NC inputs 9 - 16: Actual value)

DB10 DBX186.0 - 7	**Digital NC inputs 9 - 16: Actual value**							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The actual value is 1.							
Signal state 0	The actual value is 0.							
Additional information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 16	Input 15	Input 14	Input 13	Input 12	Input 11	Input 10	Input 9
	The actual NC output value can be read in the PLC user program via the actual value.							
	Note The value in the "Actual value" interface can, as a result of the various subsequent influencing possibilities "Lock" and "Set", have a different value than that available at the NC output.							
Corresponds with	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Additional information	Function Manual Basic Functions							

18.2.1.79 DB10 DBX187.0 - 7 (digital NC inputs 17 - 24: Actual value)

DB10 DBX187.0 - 7	Digital NC inputs 17 - 24: Actual value							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The actual value is 1.							
Signal state 0	The actual value is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 24	Input 23	Input 22	Input 21	Input 20	Input 19	Input 18	Input 17
	The actual NC output value can be read in the PLC user program via the actual value.							
	Note The value in the "Actual value" interface can, as a result of the various subsequent influencing possibilities "Lock" and "Set", have a different value than that available at the NC output.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.80 DB10 DBX188.0 - 7 (digital NC inputs 25 - 32: Actual value)

DB10 DBX188.0 - 7	Digital NC inputs 25 - 32: Actual value							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The actual value is 1.							
Signal state 0	The actual value is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 32	Input 31	Input 30	Input 29	Input 28	Input 27	Input 26	Input 25
	The actual NC output value can be read in the PLC user program via the actual value.							
	Note The value in the "Actual value" interface can, as a result of the various subsequent influencing possibilities "Lock" and "Set", have a different value than that available at the NC output.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.81 DB10 DBX189.0 - 7 (digital NC inputs 33 - 40: Actual value)

DB10 DBX189.0 - 7	Digital NC inputs 33 - 40: Actual value							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The actual value is 1.							
Signal state 0	The actual value is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Input 40	Input 39	Input 38	Input 37	Input 36	Input 35	Input 34	Input 33
	The actual NC output value can be read in the PLC user program via the actual value.							
	Note The value in the "Actual value" interface can, as a result of the various subsequent influencing possibilities "Lock" and "Set", have a different value than that available at the NC output.							
Corresponds to	DB10 DBX0, 122, 124, 126, 128 (digital NC inputs 1 - 8: Lock) DB10 DBB1, 123, 125, 127, 129 (digital NC inputs 1 - 8: Set) DB10 DBB60, 186, 187, 188, 189 (digital NC inputs 1 - 8: Actual value) MD10350 $MN_FASTIO_DIG_NUM_INPUTS							
Further information	Function Manual, Extended Functions:							

18.2.1.82 DB10 DBX190.0 - 7 (digital NC outputs 9 - 16: Setpoint)

DB10 DBX190.0 - 7	Digital NC outputs 9 - 16: Setpoint							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The setpoint is 1.							
Signal state 0	The setpoint is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
	The actual NC output value can be read in the PLC user program via the setpoint.							
	Note The value in the "Setpoint" interface can, as a result of the various subsequent influencing possibilities "Target" and "Lock", have a different value than that available at the NC output.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.83 DB10 DBX191.0 - 7 (digital NC outputs 17 - 24: Setpoint)

DB10 DBX191.0 - 7	Digital NC outputs 17 - 24: Setpoint							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The setpoint is 1.							
Signal state 0	The setpoint is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
	The actual NC output value can be read in the PLC user program via the setpoint.							
	Note The value in the "Setpoint" interface can, as a result of the various subsequent influencing possibilities "Target" and "Lock", have a different value than that available at the NC output.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.84 DB10 DBX192.0 - 7 (digital NC outputs 25 - 32: Setpoint)

DB10 DBX192.0 - 7	Digital NC outputs 25 - 32: Setpoint							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The setpoint is 1.							
Signal state 0	The setpoint is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
	The actual NC output value can be read in the PLC user program via the setpoint.							
	Note The value in the "Setpoint" interface can, as a result of the various subsequent influencing possibilities "Target" and "Lock", have a different value than that available at the NC output.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.85 DB10 DBX193.0 - 7 (digital NC outputs 33 - 40: Setpoint)

DB10 DBX193.0 - 7	Digital NC outputs 33 - 40: Setpoint							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The setpoint is 1.							
Signal state 0	The setpoint is 0.							
Further information	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33
	The actual NC output value can be read in the PLC user program via the setpoint.							
	Note The value in the "Setpoint" interface can, as a result of the various subsequent influencing possibilities "Target" and "Lock", have a different value than that available at the NC output.							
Corresponds to	DB10 DBB4, 130, 134, 138, 142 (digital NC outputs: Lock) DB10 DBB5, 131, 135, 139, 143 (digital NC outputs 1 - 8: Overwrite) DB10 DBB6, 132, 136, 140, 144 (digital NC outputs: Setting value from the PLC) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.86 DB10 DBW194 - 208 (analog NC inputs 1 - 8: Actual value)

DB10 DBW194 - 208	Analog NC inputs 1 - 8: Actual value							
Signal flow	NC → PLC							
Update	Cyclic							
Further information	**DBW208**	**DBW206**	**DBW204**	**DBW202**	**DBW200**	**DBW198**	**DBW196**	**DBW194**
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
	The actual value must be specified as a fixed point number (16 bit value including sign) in a 2's complement.							
Corresponds to	DBX146.0 - 7 (analog NC inputs 1 - 8: Lock) DB10 DBB147 (analog NC inputs 1 - 8: Target) DB10 DBW148 - 162 (analog NC outputs 1 - 8: Setting value) MD10300 $MN_FASTIO_ANA_NUM_INPUTS							
Further information	Function Manual Basic Functions							

18.2.1.87 DB10 DBW210 - 224 (analog NC outputs 1 - 8: Setpoint)

DB10 DBW210 - 224	Analog NC outputs 1 - 8: Setpoint
Signal flow	NC → PLC
Update	Cyclic

DB10 DBW210 - 224	Analog NC outputs 1 - 8: Setpoint							
Further information	DBW224	DBW222	DBW220	DBW218	DBW216	DBW214	DBW212	DBW210
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
	The actual NC output value can be read in the PLC user program via the setpoint. The setpoint is represented as a fixed point number (16 bit value including sign) in a 2's complement.							
	Note The value in the "Setpoint" interface can, as a result of the various subsequent influencing possibilities, have a different value than that available at the NC output.							
Corresponds to	DB10 DBX166.0 - 7 (analog NC outputs 1 - 8: Overwrite) DB10 DBX167.0 - 7 (analog NC outputs 1 - 8: Target) DB10 DBX168.0 - 7 (analog NC outputs 1 - 8: Lock) DB10 DBW170 - 184 (analog NC outputs 1 - 8: Setting value) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS							
Further information	Function Manual Basic Functions							

18.2.1.88 DB10 DBX226.0 - 233.7 (collision avoidance: Protection area active)

DB10 DBX226.0 - 233.7	Collision avoidance: Protection area active							
Signal flow	NC → PLC							
Update	Cyclic							
Signal state 1	The protection area connected to the interface signal is active. If **several** protection areas are connected to the interface signal: All protection areas are active.							
Signal state 0	The protection area connected to the interface signal is **not** active. If **several** protection areas are connected to the interface signal: At least one protection area is **not** active.							
Further information	The assignment protection area ↔ interface signal (<byte>.<bit>) is realized by appropriately parameterizing system variables $NP_BIT_NO: $NP_BIT_NO[<protection area>] = <bit number>							
	Byte	**Bit**						
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		Bit number						
	226	7	6	5	4	3	2	1	0
	227	15	14	13	12	11	10	9	8
	228	23	22	21	20	19	18	17	16
	229	31	30	29	28	27	26	25	24
	230	39	38	37	36	35	34	33	32
	231	47	46	45	44	43	42	41	40
	232	55	54	53	52	51	50	49	48
	233	63	62	61	60	59	58	57	56
Corresponds to	DB10 DBX234.0 - 241.7 (collision avoidance: Activate protection area) System variable $NP_BIT_NO (number of the interface bit for switchover activated)							
Further information	Function Manual Monitoring and Compensating							

18.2.1.89 DB10 DBX234.0 - 241.7 (collision avoidance: Activate protection area)

DB10 DBX234.0 - 241.7	Collision avoidance: Activate protection zone							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	Request to activate the protection areas that are connected with the interface signal.							
Signal state 0	Request to deactivate the protection areas that are connected with the interface signal.							
Further information	The assignment protection area ↔ interface signal (<byte>.<bit>) is realized by appropriately parameterizing system variables $NP_BIT_NO: $NP_BIT_NO[<protection area>] = <bit number>							
	Byte	**Bit**							
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		Bit number							
	234	7	6	5	4	3	2	1	0
	235	15	14	13	12	11	10	9	8
	236	23	22	21	20	19	18	17	16
	237	31	30	29	28	27	26	25	24
	238	39	38	37	36	35	34	33	32
	239	47	46	45	44	43	42	41	40
	240	55	54	53	52	51	50	49	48
	241	63	62	61	60	59	58	57	56
Corresponds to	DB10 DBX226.0 - 233.7 (collision avoidance: Protection area active) System variable $NP_BIT_NO (number of the interface bit for switchover activated)							
Further information	Function Manual Monitoring and Compensating							

18.2.1.90 DB10 DBX245.0 - 5 (Ethernet handwheel stationary)

DB10 DBX245.0 - 5	Ethernet handwheel is stationary	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	Ethernet handwheel is stationary.	
Signal state 0	Ethernet handwheel is moved.	
Further information	A stationary state is detected by the Ethernet modules to which the handwheel is connected. If a handwheel does not transfer any handwheel pulses for a defined period of time, the module detects this to be a stationary state and transfers it to the NC/PLC interface. A signal is available for every handwheel for this purpose:	
	Bit 0	Handwheel 1
	Bit 1	Handwheel 2
	Bit 2	Handwheel 3
	Bit 3	Handwheel 4
	Bit 4	Handwheel 5
	Bit 5	Handwheel 6
Further information	Function Manual Axes and Spindles	

18.2.2 DB11: Mode group

18.2.2.1 DB11 DBX0.0 (AUTOMATIC mode)

DB11 DBX0.0	AUTOMATIC mode
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request to switch over into the AUTOMATIC mode.
Signal state 0	There is **no** active request to switch over into the AUTOMATIC mode.
Further information	Irrelevant for: DB11 DBX0.4 (operating mode, changeover inhibit) == 1
Corresponds to	DB11 DBX0.2 (JOG mode) DB11 DBX0.1 (MDI mode) DB11 DBX0.4 (mode change inhibit) DB11 DBX6.0 AUTOMATIC mode active)
Further information	Function Manual Basic Functions

18.2.2.2 DB11 DBX0.1 (MDI mode)

DB11 DBX0.1	MDI mode
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request to switch over into the MDI mode.
Signal state 0	There is **no** active request to switch over into the MDI mode.
Additional information	Irrelevant for: DB11 DBX0.4 (operating mode, changeover inhibit) == 1
Corresponds with	DB11 DBX0.0 (AUTOMATIC mode) DB11 DBX0.2 (JOG mode) DB11 DBX0.4 (mode change inhibit) DB11 DBX6.1 (active MDI mode)
Additional information	Function Manual Basic Functions

18.2.2.3 DB11 DBX0.2 (JOG mode)

DB11 DBX0.2	JOG mode
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request to switch over into the JOG mode.
Signal state 0	There is **no** active request to switch over into the JOG mode.
Further information	Irrelevant for: DB11 DBX0.4 (operating mode, changeover inhibit) == 1

DB11 DBX0.2	JOG mode
Corresponds to	DB11 DBX0.0 (AUTOMATIC mode) DB11 DBX0.1 (MDI mode) DB11 DBX0.4 (mode change inhibit) DB11 DBX6.2 (JOG mode active)
Further information	Function Manual Basic Functions

18.2.2.4 DB11 DBX0.4 (mode change inhibit)

DB11 DBX0.4	Mode change inhibit
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The active mode of the mode group **cannot** be changed. **Note** The machine functions that can be selected within a mode group can be changed.
Signal state 0	The mode of the mode group can be changed.
Corresponds to	DB11 DBX0.0 (AUTOMATIC mode) DB11 DBX0.1 (MDI mode) DB11 DBX0.2 (JOG mode)
Further information	Function Manual Basic Functions

18.2.2.5 DB11 DBX0.5 (mode group stop)

DB11 DBX0.5	Mode group stop
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request for "Mode group stop". Responses: • An "NC Stop" is initiated for all the channels of the mode group. • Channel state change as a result of the mode group stop: – "Active" → "Interrupted" – "Reset" → "Reset" • Program state change as a result of the mode group stop – "Running" → "Stopped" • All traversing axes of the mode group are braked without contour violation to zero speed along their acceleration curves Programs that have been stopped can be restarted with "NC Start". The spindles of the mode group are not influenced.
Signal state 0	There is **no** active request for "Mode group stop".
Corresponds to	DB21, ...DBX7.2 (NC start) DB21, ...DBX7.3 (NC stop)
Further information	Function Manual Basic Functions

18.2.2.6 DB11 DBX0.6 (mode group stop, axes plus spindles)

DB11 DBX0.6	Mode group stop axes plus spindles
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request for "Mode group stop axes plus spindles". Responses: • An "NC Stop" is initiated for all the channels of the mode group. • Channel state change as a result of the mode group stop: – "Active" → "Interrupted" – "Reset" → "Reset" • Program state change as a result of the mode group stop – "Running" → "Stopped" • All traversing axes and spindles of the mode group are braked without contour violation to zero speed along their acceleration curves Programs that have been stopped can be restarted with "NC Start".
Signal state 0	There is **no** active request for "Mode group stop axes plus spindles".
Corresponds to	DB21, ...DBX7.2 (NC start) DB21, ...DBX7.3 (NC stop)
Further information	Function Manual Basic Functions

18.2.2.7 DB11 DBX0.7 (mode group reset)

DB11 DBX0.7	Mode group reset
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	A request was initiated for a "Mode group reset". Responses: • In all channels of the mode group, a "channel reset" is initiated in the control system corresponding to DB21, ... DBX 7.7 = 1. • All the channels are then in the channel state "reset" (DB21, ... DBX35.7 == 1) • All active NC programs are then in the "canceled" program state (DB21, ... DBX35.4 == 1) • All traversing axes and spindles are decelerated without contour violation to zero speed according to their acceleration curves. • The channel-specific reset positions become active. • All alarms of the mode group are cleared - with the exception of power on alarms.
Edge change 1 → 0	Reset of the last request for a "Mode group reset". Starting at this time, a new "Mode group reset" can be requested.

DB11 DBX0.7	Mode group reset
Further information	**Special case** If an alarm resets the interface signal DB11 DBX6.3 (mode group ready), all channels of the mode group are no longer in the "reset" channel state. In order that the mode of the mode group can be switched over, initially, a "Mode group reset" must be initiated using DB11 DBX0.7. **Note** Before a new action can be requested in a channel of the mode group from the PLC user program after a mode group reset (e.g. NC start or axis interchange), the system must first wait for either the mode group state "All channels of the mode group in the reset state" (DB11 DBX6.7 == 1) - or the channel state "Reset" (DB21, ... DBX35.7 == 1) of the channel involved.
Corresponds to	DB21, ... DBX7.7 (channel reset) DB11 DBX6.7 (all channels in the reset state)
Further information	Function Manual Basic Functions

18.2.2.8 DB11 DBX1.0 (TEACH IN machine function)

DB11 DBX1.0	TEACH IN machine function
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request to activate the TEACH IN machine function.
Signal state 0	There is no active request to activate the TEACH IN machine function.
Further information	**Note** Only active in the JOG mode.
Corresponds to	DB11 DBX6.2 (JOG mode active) DB11 DBX7.1 (TEACH IN machine function active)
Further information	Function Manual Basic Functions

18.2.2.9 DB11 DBX1.1 (REPOS machine function)

DB11 DBX1.1	REPOS machine function
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request to activate the REPOS machine function.
Signal state 0	There is no active request to activate the REPOS machine function.
Further information	**Note** Only active in the JOG mode.
Corresponds to	DB11 DBX6.2 (JOG mode active) DB11 DBX7.2 (REPOS machine function active)
Further information	Function Manual Basic Functions

18.2.2.10 DB11 DBX1.2 (REF machine function)

DB11 DBX1.2	REF machine function
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request to activate the REF machine function.
Signal state 0	There is no active request to activate the REF machine function.
Further information	**Note** Only active in the JOG mode.
Corresponds to	DB11 DBX6.2 (JOG mode active) DB11 DBX7.3 (REF machine function active)
Further information	Function Manual Basic Functions

18.2.2.11 DB11 DBX1.6 (single block, type B)

DB11 DBX1.6	Single block, type B
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request to activate the "Single block, type B".
Signal state 0	There is **no** active request to activate the "Single block, type B".
Further information	Cross mode group response when activating "Single block, type B": • All channels are stopped • All channels receive a start command • Channel KS stops at the end of the block • The KA channels receive the STOPATEND instruction, comparable with DB21, ... DBX7.2 (NC Stop at the block limit) • All channels are stopped at a block limit (at some point in time). **Note** • DB11 DBX1.6 == 1 AND DBX1.7 == 0 ⇒ single block type B • DB11 DBX1.6 == 0 AND DBX1.7 == 1 ⇒ single block type A • DB11 DBX1.6 == 1 AND DBX1.7 == 1 **OR** DB11 DBX1.6 == 0 AND DBX1.7 == 0 ⇒ No mode group-specific single block selected
Corresponds to	DB11 DBX1.7 (single block, type A)
Further information	Function Manual Basic Functions

18.2.2.12 DB11 DBX1.7 (single block, type A)

DB11 DBX1.7	Single block, type A
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	There is an active request to activate the "Single block, type A".
Signal state 0	There is **no** active request to activate the "Single block, type A".

DB11 DBX1.7	Single block, type A
Further information	Cross mode group response when activating "Single block, type A": • All channels are stopped • All channels receive a start command • Channel KS stops at the end of the block • The KA channels receive the STOPATEND instruction, comparable with DB21, ... DBX7.2 (NC Stop at the block limit) • All channels are stopped at a block limit (at some point in time). **Note** • DB11 DBX1.6 == 1 AND DBX1.7 == 0 ⇒ single block type B • DB11 DBX1.6 == 0 AND DBX1.7 == 1 ⇒ single block type A • DB11 DBX1.6 == 1 AND DBX1.7 == 1 **OR** DB11 DBX1.6 == 0 AND DBX1.7 == 0 ⇒ No mode group-specific single block selected
Corresponds to	DB11 DBX1.6 (single block, type B)
Further information	Function Manual Basic Functions

18.2.2.13 DB11 DBX4.0 (AUTOMATIC mode selected)

DB11 DBX4.0	AUTOMATIC mode selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	There is an active HMI request to select the AUTOMATIC mode.
Signal state 0	There is **no** active HMI request to select the AUTOMATIC mode.
Corresponds to	DB11 DBX0.0 (AUTOMATIC mode)
Further information	Function Manual Basic Functions

18.2.2.14 DB11 DBX4.1 (MDI mode selected)

DB11 DBX4.1	MDI mode selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	There is an active HMI request to select the MDI mode.
Signal state 0	There is **no** active HMI request to select the MDI mode.
Corresponds to	DB11 DBX0.1 (MDI mode)
Further information	Function Manual Basic Functions

18.2.2.15 DB11 DBX4.2 (JOG mode selected)

DB11 DBX4.2	JOG mode selected
Signal flow	HMI → PLC
Update	Cyclic

DB11 DBX4.2	JOG mode selected
Signal state 1	There is an active HMI request to select the JOG mode.
Signal state 0	There is **no** active HMI request to select the JOG mode.
Corresponds to	DB11 DBX0.2 (JOG mode)
Further information	Function Manual Basic Functions

18.2.2.16 DB11 DBX5.0 (TEACH IN machine function selected)

DB11 DBX5.0	TEACH IN machine function selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	There is an active HMI request to select the TEACH IN machine function.
Signal state 0	There is **no** active HMI request to select the TEACH IN machine function.
Corresponds to	DB11 DBX7.1 (active TEACH IN machine function)
Further information	Function Manual Basic Functions

18.2.2.17 DB11 DBX5.1 (REPOS machine function selected)

DB11 DBX5.1	REPOS machine function selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	There is an active HMI request to select the REPOS machine function.
Signal state 0	There is **no** active HMI request to select the REPOS machine function.
Corresponds to	DB11 DBX7.2 (active REPOS machine function)
Further information	Function Manual Basic Functions

18.2.2.18 DB11 DBX5.2 (REF machine function selected)

DB11 DBX5.2	REF machine function selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	There is an active HMI request to select the REF machine function.
Signal state 0	There is **no** active HMI request to select the REF machine function.
Corresponds to	DB11 DBX7.3 (active REF machine function)
Further information	Function Manual Basic Functions

18.2.2.19 DB11 DBX6.0 (AUTOMATIC mode active)

DB11 DBX6.0	AUTOMATIC mode active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The AUTOMATIC mode is active.
Signal state 0	The AUTOMATIC mode is **not** active.
Corresponds to	DB11 DBX0.0 (AUTOMATIC mode)
Further information	Function Manual Basic Functions

18.2.2.20 DB11 DBX6.1 (active MDI mode)

DB11 DBX6.1	Active MDI mode
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The MDI mode is active.
Signal state 0	The MDI mode is **not** active.
Corresponds to	DB11 DBX0.1 (MDI mode)
Further information	Function Manual Basic Functions

18.2.2.21 DB11 DBX6.2 (JOG mode active)

DB11 DBX6.2	Active JOG mode
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The JOG mode is active.
Signal state 0	The JOG mode is **not** active.
Corresponds to	DB11 DBX0.2 (JOG mode)
Further information	Function Manual Basic Functions

18.2.2.22 DB11 DBX6.3 (mode group ready)

DB11 DBX6.3	Mode group ready
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The mode group is ready.
Signal state 0	The mode group **is** ready.

DB11 DBX6.3	Mode group ready
Further information	Response when the signal changes from 0 → 1: • Traversing axes and spindles are braked down to standstill with setpoint 0 or maximum braking current. • The interface signals from the PLC to the NC are brought into an inactive state (reset position). **Note** If an alarm occurs that clears "Mode group ready", before switching over the mode, a mode group reset (DB11 DBX 0.7) must first be initiated.
Corresponds to	DB11 DBX0.7 (mode group reset)
Further information	Function Manual Basic Functions

18.2.2.23 DB11 DBX6.7 (all channels in the "Reset" state)

DB11 DBX6.7	All channels in the "Reset" state
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	All mode group channels are in the "Reset" state.
Signal state 0	At least one channel of the mode group is not in the "Reset" state.
Corresponds to	DB21, ... DBX7.7 (channel state, "Reset")
Further information	Function Manual Basic Functions

18.2.2.24 DB11 DBX7.0 (TEACH IN machine function active)

DB11 DBX7.0	TEACH IN machine function active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The TEACH IN machine function is active.
Signal state 0	The TEACH IN machine function is **not** active.
Corresponds to	DB11 DBX1.0 (TEACH IN machine function) DB11 DBX5.0 (TEACH IN machine function selected)
Further information	Function Manual Basic Functions

18.2.2.25 DB11 DBX7.1 (REPOS machine function active)

DB11 DBX7.1	REPOS machine function active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Machine function REPOS is active.
Signal state 0	Machine function REPOS is **not** active.

DB11 DBX7.1	REPOS machine function active
Corresponds to	DB11 DBX1.1 (REPOS machine function) DB11 DBX5.1 (REPOS machine function selected)
Further information	Function Manual Basic Functions

18.2.2.26 DB11 DBX7.2 (REF machine function active)

DB11 DBX7.2	REF machine function active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Machine function REF is active.
Signal state 0	Machine function REF is **not** active.
Corresponds with	DB11 DBX1.2 (REF machine function) DB11 DBX5.2 (REF machine function selected)
Additional information	Function Manual Basic Functions

18.2.3 DB18: Safety Integrated, SPL

18.2.3.1 DB18 DBB36.0 (SPL_READY)

DB18 DBB36.0	SPL_READY
Signal flow	PLC → PLC
Update	Job-controlled
Signal state 1	The SPL_READY = TRUE signal indicates that the commissioning phase has been completed, i.e. if a CDC error has occurred, the basic program signals a "STOP D/E" to all the axes.
Signal state 0	SPL_READY = FALSE
Further information	Function Manual, SINUMERIK Safety Integrated

18.2.3.2 DB18 DBB36.1 (STOP_E)

DB18 DBB36.1	STOP_E
Signal flow	PLC → PLC
Update	Job-controlled
Signal state 1	STOP E is active. The signal must be used for the forced checking procedure for external STOPs.
Signal state 0	STOP E is active.
Further information	Function Manual, SINUMERIK Safety Integrated

18.2.3.3 DB18 DBB38 - 41 (SPL inputs, SPL_DATA.INSEP[1...32])

DB18 DBB38.0 - 41.7	Data area of the SPL inputs							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	When cyclic F communication is established, the process values received from F_SENDDP are output at the SPL inputs SPL_DATA.INSEP.							
Signal state 0	The process values received from F_SENDDP are not output at the SPL inputs SPL_DATA.INSEP.							
Further information	**Byte**	**Bit**						
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		Bit number							
	38	7	6	5	4	3	2	1	0
	39	15	14	13	12	11	10	9	8
	40	23	22	21	20	19	18	17	16
	41	31	30	29	28	27	26	25	24
Corresponds to	$A_INSE							
Further information	Function Manual, Safety Integrated							

18.2.3.4 DB18 DBB42.0 - 45.7 (SPL inputs, SPL_DATA.INSEP[33...64])

DB18 DBB42.0 - 45.7	Data area of the SPL inputs							
Signal flow	PLC → NC							
Update	Cyclic							
Signal state 1	When cyclic F communication is established, the process values received from F_SENDDP are output at the SPL inputs SPL_DATA.INSEP.							
Signal state 0	The process values received from F_SENDDP are not output at the SPL inputs SPL_DATA.INSEP.							
Further information	**Byte**	**Bit**						
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		Bit number							
	42	39	38	37	36	35	34	33	32
	43	47	46	45	44	43	42	41	40
	44	55	54	53	52	51	50	49	48
	45	63	62	61	60	59	58	57	56
Corresponds to	$A_INSE							
Further information	Function Manual, Safety Integrated							

18.2.3.5 DB18 DBB46.0 - 49.7 (SPL outputs, SPL_DATA.OUTSEP[1...32]

DB18 DBB46.0 - 49.7	Data area of the SPL outputs
Signal flow	PLC → NC
Update	Cyclic

DB18 DBB46.0 - 49.7	Data area of the SPL outputs							
Signal state 1	The received process values are output at the SPL outputs SPL_DATA.OUTSEP.							
Signal state 0	The received process values are not output at the SPL outputs SPL_DATA.OUTSEP.							
Further information	**Byte**	**Bit**						
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		Bit number						
	38	7	6	5	4	3	2	1	0
	39	15	14	13	12	11	10	9	8
	40	23	22	21	20	19	18	17	16
	41	31	30	29	28	27	26	25	24
Corresponds to	$A_OUTSE							
Further information	Function Manual, Safety Integrated							

18.2.4 DB19: Operator panel

18.2.4.1 DB19 DBX0.0 (brighten screen)

DB19 DBX0.0	Brighten screen
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	The screen content is displayed. Screen darkening is ignored.
Signal state 0	Screen darkening is active.
Corresponds with	DB19 DBX0.1 (darken screen)

18.2.4.2 DB19 DBX0.1 (darken screen)

DB19 DBX0.1	Darken screen
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	The screen is darkened. The automatic screen brightening/darkening is therefore inactive. The screen does **not** automatically become bright when a key on the keyboard is pressed. **Notice** If the interface signal is set, and the screen is darkened, then the keyboard of the operator panel front still remains active. It is therefore recommended that the keyboard of the operator panel front is also locked: DB19 DBX0.2 = 1 (key lock)

DB19 DBX0.1	Darken screen
Signal state 0	The screen is brightened. In this state, the control automatically brightens/darkens the screen: • The screen is darkened if no keystroke is made at the keyboard for a time that is defined using the following machine data: MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL (time to darken the screen) The default setting is 15 minutes. • The screen is brightened the next time a key on the operator panel front is pressed.
Corresponds with	DB19 DBX0.0 (brighten screen) DB19 DBX0.2 (key lock) MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL (time to darken the screen)

18.2.4.3 DB19 DBX0.2 (key lock)

DB19 DBX0.2	Key lock
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	The keyboard is locked for the user.
Signal state 0	The keyboard is enabled for the user.
Additional information	If the screen is darkened with the interface signal: DB19 DBX0.1 (darken screen), the keyboard should be locked simultaneously with the interface signal: DB19 DBX0.2 (key lock) to avoid an unintended operation. **Notice** If the screen is darkened with (DB19 DBX0.1 = 1), the operator panel front keyboard still remains functional. It is therefore recommended that the keyboard of the operator panel front is also locked: DB19 DBX0.2 = 1 (key lock)
Corresponds with	DB19 DBX0.0 (brighten screen) DB19 DBX0.1 (darken screen)

18.2.4.4 DB19 DBX0.3 (delete cancel alarms)

DB19 DBX0.3	Delete cancel alarms
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	The delete error key on the operator panel is pressed ⇒ Request to acknowledge all cancel alarms of the NC and operator panel.
Signal state 0	The delete error key is not pressed on the operator panel.
Corresponds with	DB19 DBX20.3 (cancel alarms deleted)

18.2.4.5 DB19 DBX0.4 (delete recall alarms)

DB19 DBX0.4	Delete recall alarms
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	The delete error key is pressed on the operator panel. Request to acknowledge all recall alarms of the NC and operator panel.
Signal state 0	The delete error key is not pressed on the operator panel.
Corresponds with	DB19 DBX20.4 (recall alarms deleted)

18.2.4.6 DB19 DBX0.7 (actual values in the WCS)

DB19 DBX0.7	Actual values in the WCS
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	If the "machine" operating area is **selected** on the user interface, the axis-specific positions and distance-to-go are displayed in the workpiece coordinate system (**WCS**).
Signal state 0	If the "machine" operating area is **selected** on the user interface, the axis-specific positions and distance-to-go are displayed in the previous active coordinate system.
Additional information	**Within** the "machine" operating area, the coordinate system can be displayed in which the axis-specific positions and distances-to-go are switched via the "MCS actual values" and "WCS actual values" softkeys:
Corresponds with	DB19 DBX20.7 (switch MCS/WCS)

18.2.4.7 DB19 DBB6 (analog spindle 1, utilization as a percentage)

DB19 DBB6	Analog spindle 1, utilization in percent
Signal flow	PLC → OP
Update	Cyclic
Additional information	Format: UINT (unsigned, 0 ... 255_D or 0 ... FF_H)

18.2.4.8 DB19 DBB7 (analog spindle 2, utilization as a percentage)

DB19 DBB7	Analog spindle 2, utilization in percent
Signal flow	PLC → OP
Update	Cyclic
Additional information	Format: UINT (unsigned, 0 ... 255_D or 0 ... FF_H)

18.2.4.9 DB19 DBB8 (channel number)

DB19 DBB8	Channel number
Signal flow	PLC → OP
Update	Cyclic
Additional information	Format: UINT (unsigned, 0 ... 255_D or 0 ... FF_H) Valid channel numbers: 1 ... maximum number of NC channels

18.2.4.10 DB19 DBB10 (PLC hardkeys)

DB19 DBB10	PLC hardkeys
Signal flow	PLC → OP
Update	Cyclic
Signal state ≠ 0	1 ... 255: Selects the corresponding programming area
Signal state == 0	0: No selection
Additional information	Format: UINT (unsigned, 0 ... 255_D or 0 ... FF_H)

18.2.4.11 DB19 DBX13.5 (NC program: unload)

DB19 DBX13.5	NC program: Unload
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	Unload active
Signal state 0	Unload not active

18.2.4.12 DB19DBX13.6 (NC program: load)

DB19 DBX13.6	NC program: Load
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	Load active
Signal state 0	Load not active

18.2.4.13 DB19 DBX13.7 (NC program: selection)

DB19 DBX13.7	NC program: Selection
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	Selection active
Signal state 0	Selection not active

18.2.4.14 DB19 DBX14.0 - 6 (PLC index)

DB19 DBX14.0 - 6	PLC index
Signal flow	PLC → OP
Update	Cyclic
Additional information	The V24 interface is controlled with bits 0 - 6. They specify the PLC index for the standard control file, which specifies the axis, channel or TO number. The file system is selected via: • DB19 DBX14.7 == 1 ⇒ **passive** file system: PLC index for the user control file • DB19 DBX14.7 == 0 ⇒ **active** file system: PLC index, which specifies the axis, channel or TO number. Format: UINT (unsigned, 0 ... 127_D or 0 ... $7F_H$)
Corresponds with	DB19 DBX14.7 (selects the file system)

18.2.4.15 DB19 DBX14.7 (selects the file system)

DB19 DBX14.7	Selects the file system
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	Passive file system
Signal state 0	Active file system
Corresponds with	DB19 DBX14.0 - 6 (PLC index)

18.2.4.16 DB19 DBB15 (PLC line offset)

DB19 DBB15	PLC line offset
Signal flow	PLC → OP
Update	Cyclic
Additional information	The byte to control the V24 interface defines the line of the standard or user control file in which the control file to be transferred is specified. The control file is selected using: • DB19 DBX14.7 == 1 ⇒ passive file system: PLC line offset in a user control file • DB19 DBX14.7 == 0 ⇒ active file system: PLC line offset in a standard control file Format: UINT (unsigned, 0 ... 255_D or 0 ... FF_H)
Corresponds with	DB19 DBX14.7 (selects the file system)

18.2.4.17 DB19 DBB16 (program selection from the PLC: Index of the program list)

DB19 DBB16	Program selection from the PLC: Index of the program list
Signal flow	PLC → OP
Update	Cyclic

<table>
<tr><th>DB19 DBB16</th><th colspan="8">Program selection from the PLC: Index of the program list</th></tr>
<tr><td>Further information</td><td colspan="8">Preselected programs/workpieces in the PLC program lists can be selected for machining by the NC via the PLC/HMI interface.
To specify the program list, its number (index) is output in binary code via control byte DB19 DBB16:</td></tr>
<tr><td></td><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td></td><td>Always 1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="8">DB19 DBB16 = <number of the program list></td></tr>
<tr><td></td><td colspan="2">Number
(binary-coded)</td><td colspan="2">Number
(decimal)</td><td colspan="4">Program list</td></tr>
<tr><td></td><td colspan="2">1000 0001</td><td colspan="2">129</td><td colspan="4">User program list (/user/sinumerik/hmi/plc/programlist/plc_proglist_user.ppl)</td></tr>
<tr><td></td><td colspan="2">1000 0011</td><td colspan="2">131</td><td colspan="4">Manufacturer program list (/oem/sinumerik/hmi/plc/programlist/plc_proglist_manufacturer.ppl)</td></tr>
<tr><td>Corresponds to</td><td colspan="8">DB19.DBX13.7 (NC program: Selection)
DB19.DBB17 (program selection from the PLC: program index in the program list)
DB19.DBB26 (program selection from the PLC: Status signals)
DB19.DBB27 (program selection from the PLC: Error detection)
MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls)
MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area)
MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)</td></tr>
<tr><td>Further information</td><td colspan="8">Program selection (Page 110)</td></tr>
</table>

18.2.4.18 DB19 DBB17 (program selection from the PLC: Program index in the program list)

<table>
<tr><th>DB19 DBB17</th><th colspan="2">Program selection from the PLC: Program index in the program list</th></tr>
<tr><td>Signal flow</td><td colspan="2">PLC → OP</td></tr>
<tr><td>Update</td><td colspan="2">Cyclic</td></tr>
<tr><td>Further information</td><td colspan="2">Preselected programs/workpieces in the PLC program lists can be selected for machining by the NC via the PLC/HMI interface.
To specify the program within the program list selected via DB19 DBB16, the program number is output in binary code via control byte DB19 DBB17.</td></tr>
<tr><td></td><td>Area</td><td>Program number</td></tr>
<tr><td></td><td>user</td><td>1 - 100</td></tr>
<tr><td></td><td>oem</td><td>201 - 255</td></tr>
</table>

DB19 DBB17	Program selection from the PLC: Program index in the program list
Corresponds to	DB19.DBX13.7 (NC program: Selection) DB19.DBB16 (program selection from the PLC: index of the program list) DB19.DBB26 (program selection from the PLC: Status signals) DB19.DBB27 (program selection from the PLC: Error detection) MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls) MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area) MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)
Further information	Program selection (Page 110)

18.2.4.19 DB19 DBX20.1 (screen is dark)

DB19 DBX20.1	Screen is dark
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	The screen is darkened.
Signal state 0	The screen is not darkened.
Corresponds with	DB19 DBX0.1 (darken screen) MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL (time for the screen darkening)

18.2.4.20 DB19 DBX20.3 (cancel alarms deleted)

DB19 DBX20.3	Cancel alarm deleted
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Feedback: Cancel alarms were deleted.
Signal state 0	**Note** The interface signal must be reset in the PLC user program.
Corresponds with	DB19 DBX0.3 (delete cancel alarms)

18.2.4.21 DB19 DBX20.4 (recall alarms deleted)

DB19 DBX20.4	Recall alarms deleted
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Feedback: **Recall** alarms were deleted
Signal state 0	**Note** The interface signal must be reset in the PLC user program.
Corresponds with	DB19 DBX0.4 (delete recall alarms)

18.2.4.22 DB19 DBX20.6 (simulation active)

DB19 DBX20.6	Simulation active
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	An NC program has been selected at the user interface and the "Simulation" function activated for this program.
Signal state 0	The "Simulation" function is not active.
Further information	• SINUMERIK Operate Commissioning Manual • Operating Manuals "Turning", "Milling" and "Universal"

18.2.4.23 DB19 DBX20.7 (switch MCS/WCS)

DB19 DBX20.7	Switch over MCS/WCS
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Request to switch over the display of the axis-specific positions and residual distances to go on the user interface of the workpiece coordinate system (WCS) into the machine coordinate system (MCS) or from MCS to WCS. The signal is active for **one** PLC cycle.
Signal state 0	No effect
Corresponds with	DB19, DBX0.7 (actual value in WCS)

18.2.4.24 DB19 DBB22 (actual channel number)

DB19 DBB22	Actual channel number
Signal flow	OP → PLC
Update	Cyclic
Additional information	Format: UINT (unsigned, 0 ... 255D or 0 ... FFH) Valid channel numbers: 1 ... maximum number of NC channels

18.2.4.25 DB19 DBW24 (current screen number)

DB19 DBW24	Actual screen number
Signal flow	OP → PLC
Update	Cyclic
Additional information	Format: UINT (unsigned, 0 ... 255D or 0 ... FFH) Valid channel numbers: 1 ... maximum number of NC channels

18.2.4.26 DB19 DBX26.1 (program selection from the PLC: Job completed)

DB19 DBX26.1	Program selection from the PLC: Job completed
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Program selection from the PLC: Job completed
Signal state 0	Program selection from the PLC: Job not completed
Corresponds to	DB19.DBX13.7 (NC program: Selection) DB19.DBB16 (program selection from the PLC: index of the program list) DB19.DBB17 (program selection from the PLC: program index in the program list) DB19.DBB26 (program selection from the PLC: Status signals) DB19.DBB27 (program selection from the PLC: Error detection) MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls) MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area) MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)
Further information	Program selection (Page 110)

18.2.4.27 DB19 DBX26.2 (program selection from the PLC: Error)

DB19 DBX26.2	Program selection from the PLC: Error
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Program selection from the PLC: Job completed with errors
Signal state 0	Program selection from the PLC: Job completed correctly
Corresponds to	DB19.DBX13.7 (NC program: Selection) DB19.DBB16 (program selection from the PLC: index of the program list) DB19.DBB17 (program selection from the PLC: program index in the program list) DB19.DBB26 (program selection from the PLC: Status signals) DB19.DBB27 (program selection from the PLC: Error detection) MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls) MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area) MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)
Further information	Program selection (Page 110)

18.2.4.28 DB19 DBX26.3 (program selection from the PLC: Active)

DB19 DBX26.3	Program selection from the PLC: Active
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Program selection from the PLC: Job active

DB19 DBX26.3	Program selection from the PLC: Active
Signal state 0	Program selection from the PLC: No job active
Corresponds to	DB19.DBX13.7 (NC program: Selection) DB19.DBB16 (program selection from the PLC: index of the program list) DB19.DBB17 (program selection from the PLC: program index in the program list) DB19.DBB26 (program selection from the PLC: Status signals) DB19.DBB27 (program selection from the PLC: Error detection) MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls) MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area) MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)
Further information	Program selection (Page 110)

18.2.4.29 DB19 DBX26.5 (program selection from the PLC: Unload)

DB19 DBX26.5	Program selection from the PLC: Unload
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Program selection from the PLC: Unload active
Signal state 0	Program selection from the PLC: Unload inactive
Corresponds to	DB19.DBX13.7 (NC program: Selection) DB19.DBB16 (program selection from the PLC: index of the program list) DB19.DBB17 (program selection from the PLC: program index in the program list) DB19.DBB26 (program selection from the PLC: Status signals) DB19.DBB27 (program selection from the PLC: Error detection) MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls) MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area) MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)
Further information	Program selection (Page 110)

18.2.4.30 DB19 DBX26.6 (program selection from the PLC: Load)

DB19 DBX26.6	Program selection from the PLC: Load
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Program selection from the PLC: Load active
Signal state 0	Program selection from the PLC: Load inactive

DB19 DBX26.6	Program selection from the PLC: Load
Corresponds to	DB19.DBX13.7 (NC program: Selection) DB19.DBB16 (program selection from the PLC: index of the program list) DB19.DBB17 (program selection from the PLC: program index in the program list) DB19.DBB26 (program selection from the PLC: Status signals) DB19.DBB27 (program selection from the PLC: Error detection) MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls) MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area) MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)
Further information	Program selection (Page 110)

18.2.4.31 DB19 DBX26.7 (program selection from the PLC: Selection)

DB19 DBX26.7	Program selection from the PLC: Selection
Signal flow	OP → PLC
Update	Cyclic
Signal state 1	Program selection from the PLC: Selection active
Signal state 0	Program selection from the PLC: Selection inactive
Corresponds to	DB19.DBX13.7 (NC program: Selection) DB19.DBB16 (program selection from the PLC: index of the program list) DB19.DBB17 (program selection from the PLC: program index in the program list) DB19.DBB26 (program selection from the PLC: Status signals) DB19.DBB27 (program selection from the PLC: Error detection) MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls) MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area) MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)
Further information	Program selection (Page 110)

18.2.4.32 DB19 DBB27 (program selection from the PLC: Error detection)

DB19 DBB27	Program selection from the PLC: Error code
Signal flow	OP → PLC
Update	Cyclic

DB19 DBB27	Program selection from the PLC: Error code	
Further information	Output byte for the error values of the data transfer.	
	Value	Meaning
	0	No error
	1	Invalid program list number (DB19.DBB16)
	3	User-specific program list plc_proglist_main.ppl not found (for DB19.DBB16 ≠ 129, 131 only)
	4	Invalid program number (DB19.DBB17)
	5	The job list in the selected workpiece could not be opened
	6	Error in job list (Job list Interpreter returns error)
	7	No instructions were found in the job list
	8	HMI could not execute the selection (SELECT)
	9	An error with no further details occurred upon selection
	10	An error occurred during analysis of the job list
	11	The instruction is not available
	12	Invalid instruction (`LOAD`, `UNLOAD`, `SELECT` only)
	13	Not all channels have the Reset status
	14	The channel number is invalid
Corresponds to	DB19.DBX13.7 (NC program: Selection) DB19.DBB16 (program selection from the PLC: index of the program list) DB19.DBB17 (program selection from the PLC: program index in the program list) DB19.DBB26 (program selection from the PLC: Status signals) MD9106 $MM_SERVE_EXTCALL_PROGRAMS (editing EXTCALL calls) MD51041 $MN_ENABLE_PROGLIST_USER (activation of PLC program list USER area) MD51043 $MN_ENABLE_PROGLIST_MANUFACT (activation of PLC program list MANUFACTURER area)	
Further information	Program selection (Page 110)	

18.2.4.33 DB19 DBX32.0 - 5 (function number)

DB19 DBX32.0 - 5	Function selection from the PLC: Function number	
Signal flow	PLC → OP	
Update	Cyclic	
Further information	The function number is entered in DB19 DBX32.0 - 5, which after setting the requesting strobe in DB19 DBX32.6, is executed:	
	Function number	Function
	0	No function
	1	Channel selection
Corresponds to	DB19 DBX32.6 (requesting strobe) DB19 DBX32.7 (status) DB19 DBB33 - 35 (parameter 1 - 3) DB19 DBB36 (error detection)	
Further information	Channel selection (Page 109)	

18.2.4.34 DB19 DBX32.6 (function request)

DB19 DBX32.6	Function selection from the PLC: Request strobe
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	Execution of the set function by the NC is requested
Signal state 0	The function was executed by the NC or no function execution is requested
Corresponds to	DB19 DBX32.0 - 5 (function number) DB19 DBX32.7 (status) DB19 DBB33 - 35 (parameter 1 - 3) DB19 DBB36 - 35 (error detection)
Further information	Channel selection (Page 109)

18.2.4.35 DB19 DBX32.7 (status)

DB19 DBX32.7	Function selection from the PLC: Status
Signal flow	PLC → OP
Update	Cyclic
Signal state 1	The requested function is currently being processed by the NC.
Signal state 0	The processing of the requested function has been completed.
Corresponds to	DB19 DBX32.0 - 5 (function number) DB19 DBX32.6 (requesting strobe) DB19 DBB33 - 35 (parameter 1 - 3) DB19 DBB36 - 35 (error detection)
Further information	Channel selection (Page 109)

18.2.4.36 DB19 DBB33 - 35 (parameter 1 - 3)

DB19 DBB33 - 35	Function selection from the PLC: Parameter 1 - 3
Signal flow	PLC → OP
Update	Cyclic
Further information	Parameters can be specified for the function requested in DB19 DBB32: • Parameter 1: DB19 DBB33 • Parameter 2: DB19 DBB34 • Parameter 3: DB19 DBB35 The number and value range of the parameters depends on the associated function.
Corresponds to	DB19 DBX32.0 - 5 (function number) DB19 DBX32.6 (requesting strobe) DB19 DBX32.7 (status) DB19 DBB36 (error detection)
Further information	Channel selection (Page 109)

18.2.4.37 DB19 DBB36 (error identification)

DB19 DBB36	Error code
Signal flow	OP → PLC
Update	Cyclic
Further information	Error identification after completion of the function execution by the NC: • 0: No error • 1: Invalid function number (DBX32.0 - .5) • 2: Invalid parameter (DBB33 - DBB35) • 3: Error when writing the HMI-internal variable. • 10: Channel not present (DBB33)
Corresponds to	DB19 DBX32.0 - 5 (function number) DB19 DBX32.6 (function request) DB19 DBX32.7 (status) DB19 DBB33 - 35 (parameter 1 - 3)
Further information	Channel selection (Page 109)

18.2.5 DB21, ...: Channel

18.2.5.1 DB21, ... DBX0.1 (RESU: backward/forward)

DB21, ... DBX0.1	RESU: Backward/forward
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Requirement: Reverse travel
Signal state 0	Requirement: Forward travel
Additional information	Precise retracing of contours is possible on all programmed contours comprising straight and circular elements. During retracing, other contour elements such as splines or automatically inserted non-linear contour elements (circle, parable, etc. e.g. through tool radius compensation) are mapped as straight lines between the start and end points of the corresponding contour element, thereby preventing precise retracing of contours. The main RESU program (CC_RESU.MPF) is generated from the logged traversing blocks of the RESU internal block buffer, so corresponding to the interface signal, the next time that the NC starts, the contour can be traversed either backward or forward.
Corresponds with	DB21, ... DBX0.1 (RESU: backward/forward) DB21, ... DBX0.2 (RESU: start retrace support) DB21, ... DBX32.1 (RESU: retrace mode active) DB21, ... DBX32.2 (RESU: retrace support active)
Additional information	Function Manual Technologies

18.2.5.2 DB21, ... DBX0.2 (RESU: start retrace support)

DB21, ... DBX0.2	RESU: Start retrace support
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	"Start retrace support" is requested.
Edge change 1 → 0	No effect.
Additional information	For retrace support, RESU automatically reselects the original machining program and launches a block search with calculation as far as the program continuation point.
Corresponds with	DB21, ... DBX0.1 (RESU: backward/forward) DB21, ... DBX32.1 (RESU: retrace mode active) DB21, ... DBX32.2 (RESU: retrace support active)
Additional information	Function Manual Technologies

18.2.5.3 DB21, ... DBX0.3 (activate handwheel offset (DRF))

DB21, ... DBX0.3	Activate handwheel offset (DRF)
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activation of handwheel offset (DRF) is requested.
Signal state 0	Activation of handwheel offset (DRF) is **not** requested.
Additional information	With the handwheel offset (DRF) an axis-specific offset can be performed in the AUTOMATIC and MDI modes using a handwheel. **Note** Handwheel offset (DRF) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.3. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX0.3: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX24.3 (handwheel offset (DRF) selected)
Additional information	Function Manual Axes and Spindles

18.2.5.4 DB21, ... DBX0.4 (activate single block)

DB21, ... DBX0.4	Activate single block
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to activate the "single block" function:
Signal state 0	A request has **not** been made to activate the "single block" function.

DB21, ... DBX0.4	Activate single block
Additional information	In AUTOMATIC and MDI modes, the operator must enable processing of each individual part program block of the part program selected in the channel by reactivating NC Start.
Additional information	Function Manual Basic Functions

18.2.5.5 DB21, ... DBX0.5 (activate M01)

DB21, ... DBX0.5	Activate M01
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activation of program control "Conditional stop" `M01` is requested.
Signal state 0	Activation of program control "Conditional stop" `M01` is **not** requested.
Additional information	Program control "Conditional stop" M01 is selected from the SINUMERIK Operate user interface: Operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.5. The interface signal is then transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX0.5: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX24.5 (`M01` selected) DB21, ... DBX32.5 (`M0`/`M01` active)
Additional information	Function Manual Basic Functions

18.2.5.6 DB21, ... DBX0.6 (activate dry run feedrate)

DB21, ... DBX0.6	Activate dry run feedrate (DRY)
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	Activation of dry run feedrate (DRY) is requested.
Edge change 1 → 0	Activation of dry run feedrate (DRY) is **not** requested.

DB21, ... DBX0.6	Activate dry run feedrate (DRY)
Additional information	If the channel is in the "Reset" state, for active G function G01, G02, G03 the currently effective feedrate is used with the next NC start: • Edge change 0 → 1: Dry run feedrate • Edge change 1 → 0: Programmed feedrate **Note** If the programmed feed rate within a G33 block is requested, then the programmed feedrate is not activated until the end of the block is reached. No NC stop is executed within a G33 block. **Note** Dry run feedrate (DRY) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.6. The interface signal is transferred according to the value of the FB1 parameter MMCToIf by the basic PLC program into the NC/PLC interface signal DB21, ... DBX0.6: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX24.6 (dry run feedrate (DRY) selected) SD42100 $SC_DRY_RUN_FEED (dry run feedrate)
Additional information	Function Manual Basic Functions Function Manual Axes and Spindles

18.2.5.7 DB21, ... DBX1.0 (activate referencing)

DB21, ... DBX1.0	Activate referencing
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	The "referencing" function is requested.
Edge change 1 → 0	No effect
Additional information	Acknowledgment signal: DB21, ... DBX33.0 (referencing active) **Note** Using the axis-specific machine data, the sequence in which the machine axes are referenced for channel-specific referencing can be defined: MD34110 $MA_REFP_CYCLE_NR For channel-specific referencing, when all axes entered in machine data MD34110 $MA_REFP_CYCLE_NR have reached their reference point, then interface signal DB21, ... DBX36.3 (all axes stationary) is set.
Corresponds with	DB21, ... DBX33.0 (referencing active) DB21, ... DBX36.3 (all axes stationary) MD34110 $MA_REFP_CYCLE_NR (Axis sequence for channel-specific referencing)
Additional information	Function Manual Axes and Spindles

18.2.5.8 DB21, ... DBX2.1 (disable all synchronized actions)

DB21, ... DBX1.3	Disable all synchronized actions
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	All synchronized actions in the channel are locked. No synchronized action is therefore executed.
Signal state 0	Synchronized actions are executed in the channel.
Additional information	Function Manual Synchronized Actions

18.2.5.9 DB21, ... DBX1.3 (time monitoring active)

DB21, ... DBX1.3	Time monitoring active
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	Acquire tool life.
Signal state 0	Do **not** acquire tool life.
Additional information	The tool life is always acquired as standard, if the geometry axes are not traversed with rapid traverse (G0). With the interface signal, acquiring the tool life is deactivated and then activated again.
	Precondition Enable the functionality with MD20310 TOOL_MANAGEMENT_MASK, bit 17 = 1
Corresponds with	$A_MONIFACT (factor for reading tool life monitoring) $TC_MOP1 (prewarning limit for tool life) MD18080 $MN_MM_TOOL_MANAGEMENT_MASK (memory reserved for the tool management) MD20124 $MC_TOOL_MANAGEMENT_TOOLHOLDER (tool holder number) MD20310 TOOL_MANAGEMENT_MASK bit 17 (activation of the tool management functions) MD20320 $MC_TOOL_TIME_MONITOR_MASK (time monitoring for tool in the tool holder)
Additional information	Function Manual Tool Management

18.2.5.10 DB21, ... DBX1.4 (clearance control (CLC): stop)

DB21, ... DBX1.4	Clearance control (CLC): Stop
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activating clearance control (CLC) is requested (analogous to the program instruction CLC_GAIN=0.0).
Signal state 0	Activating clearance control (CLC) is **not** requested.
Corresponds with	CLC_GAIN=0.0 ; deactivate control loop gain DB21, ... DBX37.3 (clearance control (CLC) active)
Additional information	Function Manual Technologies

18.2.5.11 DB21, ... DBX1.5 (clearance control (CLC): Override)

DB21, ... DBX1.5	Clearance control (CLC): Override
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The channel-specific feedrate override (DB21, … DBB4) should be active for clearance control (CLC).
Signal state 0	The channel-specific feedrate override (DB21, … DBB4) should **not** be active for clearance control (CLC).
Additional information	For signal state 1, the channel-specific feedrate override acts on the maximum velocity of the clearance control motion (MD62516): • Override settings < 100% The velocity limit set in MD62516 for clearance control is appropriately reduced. • Override settings > 100% The limit value from MD62516 is active. Signal state 0 is the maximum velocity of the clearance control motion, independent of the override setting.
Corresponds with	DB21, ... DBB4 (feed rate override) DB21, ... DBX6.7 (feed override active) DB21, ... DBX37.3 (clearance control (CLC) active) MD62516 $MC_CLC_SENSOR_VELO_LIMIT (velocity of clearance control motion)
Additional information	Function Manual Technologies

18.2.5.12 DB21, ... DBX1.6 (PLC action completed)

DB21, ... DBX1.6	PLC action completed
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	PLC action completed.
Signal state 0	PLC action is **not** yet completed.
Additional information	**Note** At the end of the block search, concluding action blocks are executed: DB21, ... DBX32.3 (action block active) == 1 AND DB21, ... DBX32.6 (last action block active) == 1 Alarm "10208 Channel <*Channel Number*> Issue NC Start to continue program" notifies that a new NC Start is required to continue the NC program from the target block. If other actions are to be executed by the PLC user program prior to the NC Start (e.g. tool change), by parameterizing the search mode, the output of the alarm can be delayed until the signal is set again: MD11450 $MN_SEARCH_RUN_MODE = 1
Corresponds with	DB21, ... DBX7.1 (NC Start) DB21, ... DBX32.3 (action block active) DB21, ... DBX32.6 (last action block active) DB21, ... DBX33.4 (block search active)
Additional information	Function Manual Basic Functions

18.2.5.13 DB21, ... DBX1.7 (activate program test (PRT))

DB21, ... DBX1.7	Activate program test (PRT)
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activation of program test (PRT) is requested.
Signal state 0	Activation of program test (PRT) is **not** requested.
Additional information	If program test (PRT) is active, the machine axes do not move during the execution of a block or NC program. However, the axis movements are displayed on the user interface with changing setpoint position values. **Note** While program test (PRT) is active, all traversing motions of the **axes**, but **not** the **spindles**, take place under "Axis disable." **Note** PRT (Program test) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX25.7. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX1.7: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE". **Notice** Because of the axis disable, the assignment of a tool magazine is not changed while the program is being tested. The user / machine manufacturer must utilize a suitable PLC user program to ensure that the NC-internal tool management and the actual assignment of the tool magazine remain consistent.
Corresponds with	DB21, ... DBX25.7 (program test (PRT) requested) DB21, ... DBX33.7 (program test (PRT) active)
Additional information	Function Manual Basic Functions

18.2.5.14 DB21, ... DBX2.0 ... 7 (activate "skip block" (SKP))

DB21, ... DBX2.0 ... 7	Activate "skip block" (SKP)
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activation of "Skip block" (SKP) of the skip level is requested.
Signal state 0	Activation of "Skip block" (SKP) of the skip level is **not** requested.

DB21, ... DBX2.0 ... 7	Activate "skip block" (SKP)
Additional information	Bit 0 ... bit 7 are assigned to the skip levels 0 ... 7. **Note** We recommend that the interface signal is already set before the NC program starts. **Note** For multiple consecutive skip blocks, the blocks are only skipped if the interface signal was already present **before** the **first** skip block of the block sequence was decoded. **Note** "Skip block" (SKP) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX26.0 ...7. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX2.0 ... 7: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX26.0 - 7 ("Skip block" (SKP) selected)
Additional information	Function Manual Basic Functions

18.2.5.15 DB21, ... DBX3.0 (stroke enable)

DB21, ... DBX3.0	Stroke enable
Signal flow	PLC → NC
Signal state 1	Execution of punch strokes is enabled.
Signal state 0	Execution of punch strokes is locked.
Additional information	This signal releases the punching strokes via the PLC. It is not permissible that the NC initiates a punch stroke if the signal is not set. The NC waits until the enable signal is available before continuing the part program.
Additional information	Function Manual Technologies

18.2.5.16 DB21, ... DBX3.1 (punch interface 1: Manual stroke initiation)

DB21, ... DBX3.1	Punch interface 1: Manual stroke initiation
Signal flow	PLC → NC
Signal state 1	Execute a manual punch stroke at the 1st punch interface.
Signal state 0	There is no request to execute a manual punch stroke at the 1st punch interface.
Additional information	This signal enables a single stroke to be initiated at the 1st punch interface in the manual mode. In principle, this is possible in every operating mode, assuming that the axes do not move. The signal is ignored if the axes are moved during the manual stroke initiation. During the stroke, the axes are inhibited, i.e. they can only be moved again after the "Stroke initiation active" signal has been withdrawn.

DB21, ... DBX3.1	Punch interface 1: Manual stroke initiation
Corresponds with	DB21, ... DBX3.0 (stroke enable) DB21, ... DBX3.5 (punch interface 2: Manual stroke initiation) DB21, ... DBX38.0 (stroke initiation active) DB21, ... DBX38.1 (manual stroke initiation: Acknowledgment)
Additional information	Function Manual Technologies

18.2.5.17 DB21, ... DBX3.2 (stroke suppression)

DB21, ... DBX3.2	Stroke suppression
Signal flow	PLC → NC
Signal state 1	Activate stroke suppression.
Signal state 0	There is no request to activate stroke suppression.
Additional information	The interface signal suppresses the fast "Initiate stroke" signal. This allows the part program to be processed without initiating a punching operation (dry run). With active path segmentation, the axes traverse in the "Stop and go" mode.
Corresponds with	DB21, ... DBX3.0 (stroke enable)
Additional information	Function Manual Technologies

18.2.5.18 DB21, ... DBX3.3 (delayed stroke)

DB21, ... DBX3.3	Delayed stroke
Signal flow	PLC → NC
Signal state 1	Activate "Delayed stroke" option.
Signal state 0	There is no request to activate option "Delayed stroke".
Additional information	A "delayed stroke" can be activated using this signal. This functionally corresponds to programming `PDELAYON`.
Corresponds with	DB21, ... DBX3.0 (stroke enable)
Additional information	Function Manual Technologies

18.2.5.19 DB21, ... DBX3.4 (stroke inoperative)

DB21, ... DBX3.4	Stroke inoperative
Signal flow	PLC → NC
Signal state 1	Stroke inoperative
Signal state 0	Stroke operative.
Additional information	The NC responds to this interface signal by immediately stopping motion. An alarm is output if motion or another action is to be interrupted due to this signal. In physical terms, the interface signal is identical to the fast signal "Stroke active" for the NC, i.e. the system is wired in such a way that the two signals are taken to the same NC input via an AND gate.

DB21, ... DBX3.4	Stroke inoperative
Corresponds with	DB21, ... DBX3.0 (stroke enable)
Additional information	Function Manual Technologies

18.2.5.20 DB21, ... DBX3.5 (punch interface 2: Manual stroke initiation)

DB21, ... DBX3.5	Punch interface 2: Manual stroke initiation
Signal flow	PLC → NC
Signal state 1	Perform manual punch stroke at the 2nd punch interface.
Signal state 0	There is no request to initiate a manual punch stroke at the 2nd punch interface.
Additional information	This signal enables a single stroke to be initiated at the 2nd punch interface in manual mode. In principle, this is possible in every operating mode, assuming that the axes do not move. The signal is ignored if the axes are moved during the manual stroke initiation. During the stroke, the axes are inhibited, i.e. they can only be moved again after the "Stroke initiation active" signal has been withdrawn.
Corresponds with	DB21, ... DBX3.0 (stroke enable) DB21, ... DBX3.1 (punch interface 1: Manual stroke initiation) DB21, ... DBX38.0 (stroke initiation active) DB21, ... DBX38.1 (manual stroke initiation: Acknowledgment)
Additional information	Function Manual Technologies

18.2.5.21 DB21, ... DBB4 (path feedrate override)

DB21, ... DBB4	Path feedrate override
Signal flow	PLC → NC
Update	Cyclic

DB21, ... DBB4	Path feedrate override
Additional information	**Binary/Gray coding** The override factors can be specified in the binary or Gray-coded format. The control must be informed of the used format via the following machine data: MD12020 $MN_OVR_FEED_IS_GRAY_CODE = <Coding>

Binary coding

With binary coding, the value in the interface corresponds to the override factor.

Binary code	Decimal	Override factor
0000 0000	0	0.00
0000 0001	1	0.01
0000 0010	2	0.02
0000 0011	3	0.03
000 0100	4	0.04
...	...	...
0110 0100	100	1.00
...	...	...
1100 1000	200	2.00

Gray coding

The Gray-coded values of the interface are assigned the override factors via the following machine data:

MD12030 $MN_OVR_FACTOR_FEEDRATE[<Switch position> - 1] = <Override factor>

Switch position	Gray code	Override factor [1)]
1	00001	0.00
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.05
21	11111	1.10
22	11101	1.15

DB21, ... DBB4	Path feedrate override		
	23	11100	1.20
	24	10100	1.20
	25	10101	1.20
	26	10111	1.20
	27	10110	1.20
	28	10010	1.20
	29	10011	1.20
	30	10001	1.20
	31	10000	1.20
	Limitations • The active override factor is limited internally in the control to 2.00 or 200%. • The maximum possible override factor can be limited to a value less than 200% with the following machine data: MD12100 $MN_OVR_FACTOR_LIMIT_BIN		
Corresponds to	DB21, ... DBX6.7 (path feedrate override active)		
Additional information	Function Manual Axes and Spindles		
1) Default values			

18.2.5.22 DB21, ... DBB5 (path rapid traverse override)

DB21, ... DBB5	Path rapid traverse override
Signal flow	PLC → NC
Update	Cyclic

DB21, ... DBB5	Path rapid traverse override
Additional information	**Binary coding** With binary coding, the value in the interface corresponds to the override factor.

Binary-coded	Decimal	Override factor
0000 0000	0	0.00
0000 0001	1	0.01
0000 0010	2	0.02
0000 0011	3	0.03
000 0100	4	0.04
...	...	...
0110 0100	100	1.00
...	...	...
1100 1000	200	2.00

Gray coding

The Gray-coded values of the interface are assigned the override factors via the following machine data:

MD12050 $MN_OVR_FACTOR_RAPID_TRA[<Switch position> - 1] = <Override factor>

Switch position	Gray code	Override factor [1)]
1	00001	0.00
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.00
21	11111	1.00
22	11101	1.00
23	11100	1.00
24	10100	1.00

DB21, ... DBB5	Path rapid traverse override		
	25	10101	1.00
	26	10111	1.00
	27	10110	1.00
	28	10010	1.00
	29	10011	1.00
	30	10001	1.00
	31	10000	1.00
	Limitations • The active override factor is limited internally in the control to 2.00 or 200%. • The maximum possible override factor can be limited to a value less than 200% with the following machine data: MD12100 $MN_OVR_FACTOR_LIMIT_BIN		
	Note The path rapid traverse override is not effective with the following functions when cutting threads: G33, G331, G332, G63		
Corresponds to	DB21, ... DBX6.6 (path rapid traverse override active)		
Additional information	Function Manual Axes and Spindles		
1) Default values			

18.2.5.23 DB21, ... DBX6.0 (feedrate disable)

DB21, ... DBX6.0	Feedrate disable
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Feedrate disable is active in the channel.
Signal state 0	Feedrate disable is **not** active in the channel.

DB21, ... DBX6.0	Feedrate disable		
Additional information	• The interface signal is active for all of the geometry, synchronized and positioning axes traversing in the channel. • If the interface signal is set while traversing an axis, then the axis is braked down to standstill along its braking characteristic. For geometry axes, the axes are braked so that they follow the contour. • After the interface signal has been reset, traversing motion that was stopped is continued. • The position control is retained and the following error is eliminated. • If the interface signal is set, and a request is issued to traverse an axis, then the axis is not traversed. However, the traverse request is kept. When the interface signal is reset, the traverse request is immediately executed, i.e. the axis is traversed. • The interface signal is active in all modes.		
	Thread cutting During various thread cutting operations, the interface signal is not effective:		
		Thread cutting	**Effectiveness**
		G33, G34, G35	**Not** effective
		G331, G332	Effective
		G63	Effective
Additional information	Function Manual Axes and Spindles		

18.2.5.24 DB21, ... DBX6.1 (read-in disable)

DB21, ... DBX6.1	Read-in disable
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to activate the "read-in disable" function.
Signal state 0	A request has **not** been made to activate the "read-in disable" function.
Additional information	If the read-in disable is set, then the data transfer for the next block is enabled in the interpolator. **Note** Only active in the AUTOMATIC and MDI modes. **Application** If the auxiliary functions must be completed (e.g. for the tool change) before the next NC block can be processed, then the automatic block change must be prevented through the read-in disable.
Corresponds with	DB21, ... DBX35.0 (program state running)
Additional information	Function Manual Basic Functions

18.2.5.25 DB21, ... DBX6.2 (delete distance-to-go, channel-specific)

DB21, ... DBX6.2	Delete distance-to-go (channel-specific)
Signal flow	PLC→ NC
Update	Cyclic

DB21, ... DBX6.2	Delete distance-to-go (channel-specific)
Edge change 0 → 1	Delete distance-to-go (channel-specific) is requested. **Geometry axes** For a rising edge of the signal, the geometry axes currently traversing in the block are brought to a standstill along a braking ramp and then the distance-to-go is deleted. Any remaining following error is still corrected. The next traversing block is then loaded. **Path axes** "Delete distance-to-go" for path axes acts only in the AUTOMATIC mode. **Note** "Delete distance-to-go" causes the next traversing block to be prepared with the new positions. After a "Delete distance-to-go", the geometry axes thus follow a different contour to the one originally programmed in the NC program. Traversal to absolute positions (`G90`) in the block after "Delete distance-to-go" causes at least exact approach to the block end position. For incremental traversing (`G91`), the position specified in the NC program is not approached in the following block.
Edge change 1 → 0	No effect
Additional information	**Signal irrelevant for** • Positioning axes • Dwell time **Application example** Termination of the traversing motion on account of an external signal (e.g. probe)
Corresponds with	DB31, ... DBX2.2 (delete distance-to-go (axis-specific))

18.2.5.26 DB21, ... DBX6.4 (program level abort)

DB21, ... DBX6.4	Program level abort
Signal flow	PLC → NC
Update	Cyclic
Signal state 0 → 1	Request to abort program processing of the actual program level (subprogram level, ASUB level, data save routine).
Signal state 1 → 0	Reset of the last request.
Additional information	After a program level abort, the calling program of the next higher program level is processed further from the call location. The main program level cannot be aborted with the program level abort, but only with a channel reset.
Corresponds with	DB21, ... DBX7.7 (reset)
Additional information	Function Manual Basic Functions

18.2.5.27 DB21, ... DBX6.6 (path rapid traverse override active)

DB21, ... DBX6.6	Path rapid traverse override active
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The path rapid traverse override (DB21, ... DBB5) is active.

DB21, ... DBX6.6	Path rapid traverse override active
Signal state 0	The path rapid traverse override is **not** active.
Additional information	If the path rapid traverse override is **not** active, then irrespective of the current switch position, a value of 1.0 ≙ 100% is used as factor in the control. The **first** switch position is an exception. Depending on the selected coding, the following factor is active for the first switch position: • Binary coding: 0 • Gray coding: MD12050 $MN_OVR_FACTOR_RAPID_TRA[0]
Corresponds to	DB21, ... DBB5 (path rapid traverse override) MD12020 $MN_OVR_FEED_IS_GRAY_CODE (path feed override switch Gray-coded) MD12030 $MN_OVR_FACTOR_FEEDRATE (evaluation of the path feedrate override switch)
Additional information	Function Manual Axes and Spindles

18.2.5.28 DB21, ... DBX6.7 (path feedrate override active)

DB21, ... DBX6.7	Path feedrate override active
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The path feedrate override (DB21, ... DBB4) is active.
Signal state 0	The path feedrate override is **not** active.
Additional information	If the path feedrate override is **not** active, then irrespective of the current switch position, a value of 1.0 ≙ 100% is used as override factor in the control. The **first** switch position is an exception. Depending on the selected coding, for the first switch position the following applies: • Binary coding: Override factor = 0 • Gray coding: Override factor = MD12030 $MN_OVR_FACTOR_FEEDRATE[0]
Corresponds to	DB21, ... DBB4 (path feedrate override) MD12020 $MN_OVR_FEED_IS_GRAY_CODE (path feed override switch Gray-coded) MD12030 $MN_OVR_FACTOR_FEEDRATE (evaluation of the path feedrate override switch)
Additional information	Function Manual Axes and Spindles

18.2.5.29 DB21, ... DBX7.0 (NC start disable)

DB21, ... DBX7.0	NC start disable
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to activate the "NC start disable" function.
Signal state 0	A request has **not** been made to activate the "NC start disable" function.
Additional information	If the NC start disable is set, then the start (DB21, ... DBX7.1) of the NC program selected in the channel is ignored. **Note** The NC start inhibit is not active if the part program is started by the part program command `START` in another channel of the mode group. **Application** e.g. to suppress new program execution because there is no lubricant.

DB21, ... DBX7.0	NC start disable
Corresponds with	DB21, ... DBX7.1 (NC Start)
Additional information	Function Manual Basic Functions

18.2.5.30 DB21, ... DBX7.1 (NC Start)

DB21, ... DBX7.1	NC start
Signal flow	PLC → NC
Update	Cyclic
Signal state 0 → 1	Request to start the program selected in the channel.
Signal state 1 → 0	Reset of the last request.
Additional information	**AUTOMATIC** mode: The selected NC program is started or continued, or the auxiliary functions that were saved during the program interruption are output. If data is transferred from the PLC to the NC during program state "Program interrupted," then this data is immediately cleared at NC Start. **MDI** mode: The blocks in the MDI block memory are executed.
Corresponds with	DB21, ... DBX7.0 (NC Start disable) DB21, ... DBX7.3 (NC Stop) DB21, ... DBX7.7 (reset)
Additional information	Function Manual Basic Functions

18.2.5.31 DB21, ... DBX7.2 (NC Stop at the block limit)

DB21, ... DBX7.2	NC Stop at block limit
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	NC stop at the end of the actual block is requested.
Signal state 0	NC stop at the end of the actual block is **not** requested.
Additional information	The actual block is executed to the end of the block. Program execution is then stopped: • Channel state: "Interrupted" • Program state: "Stopped"
Corresponds with	DB21, ... DBX7.1 (NC Start) DB21, ... DBX7.3 (NC Stop) DB21, ... DBX7.4 (NC Stop axes plus spindles) DB21, ... DBX35.2 (program state stopped) DB21, ... DBX35.6 (interrupted channel state)
Additional information	Function Manual Basic Functions

18.2.5.32 DB21, ... DBX7.3 (NC Stop)

DB21, ... DBX7.3	NC stop
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	NC Stop is requested.
Signal state 0	NC Stop is **not** requested.
Additional information	**AUTOMATIC** or **MDI** mode Execution of the active part program in the channel is stopped. The traversing axes (not spindles) are braked down to standstill with the parameterized acceleration rates. • Channel state: Interrupted • Program state: Stopped **JOG** mode In the JOG mode, incompletely traversed incremental paths (INC...) are executed at the next NC Start (DB21, ... DBX7.1). **Note** If data is transferred to the NC after "NC Stop" (e.g. tool offset), the data is taken into account with the next "NC start".
Corresponds with	DB21, ... DBX7.1 (NC Start) DB21, ... DBX7.4 (NC Stop axes plus spindles) DB21, ... DBX35.2 (program state stopped) DB21, ... DBX35.6 (interrupted channel state)
Additional information	Function Manual Basic Functions

18.2.5.33 DB21, ... DBX7.4 (NC Stop axes plus spindles)

DB21, ... DBX7.4	NC Stop axes plus spindles
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	NC Stop is requested.
Signal state 0	NC Stop is **not** requested.
Additional information	**AUTOMATIC** or **MDI** mode Execution of the active part program in the channel is stopped. The traversing axes **and** spindles are braked down to standstill with the parameterized acceleration rates. • Channel state: Interrupted • Program state: Stopped **JOG** mode In the JOG mode, incompletely traversed incremental paths (INC...) are executed at the next NC Start (DB21, ... DBX7.1). **Note** If data is transferred to the NC after "NC Stop" (e.g. tool offset), the data is taken into account with the next "NC start".

DB21, ... DBX7.4	NC Stop axes plus spindles
Corresponds with	DB21, ... DBX7.1 (NC Start) DB21, ... DBX7.3 (NC Stop) DB21, ... DBX35.2 (program state stopped) DB21, ... DBX35.6 (interrupted channel state)
Additional information	Function Manual Basic Functions

18.2.5.34 DB21, ... DBX7.6 (activate configured stop)

DB21, ... DBX7.6	Activate configured stop
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activation of the configured stop is requested.
Signal state 0	Activation of the configured stop is not requested.
Additional information	"Activate configured stop" (CST) can also be selected via the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, … DBX24.1. Depending on the value of FB1 parameter MMCToIf, the interface signal is transferred from the basic PLC program to the NC/PLC interface signal DB21, … DBX7.6: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, … DBX24.1 (activate configured stop)

18.2.5.35 DB21, ... DBX7.7 (reset)

DB21, ... DBX7.7	Reset
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Channel reset is requested.
Signal state 0	Channel reset is **not** requested.
Additional information	Responses: • All traversing axes and spindles are decelerated without contour violation to zero speed according to their acceleration curves. • All channel-specific reset positions become active. • All alarms of the channel, other than Power On alarms, are deleted. • Channel state: "Reset" (DB21, ... DBX35.7 == 1) • Program state of an active NC program: "Aborted" (DB21, ... DBX35.4 == 1) **Note** After a channel reset from the PLC user program, before a new action can be requested in the channel (e.g. NC start or axis interchange), the system must wait for the channel state "reset" to be indicated at the NC/PLC interface. DB21, ... DBX35.7 == 1

DB21, ... DBX7.7	Reset
Corresponds with	DB11, ... DBX0.7 (mode group reset) DB21, ... DBX7.1 (NC Start) DB21, ... DBX7.3 (NC Stop) DB21, ... DBX35.4 (program state "aborted") DB21, ... DBX35.7 (channel state "reset")
Additional information	Function Manual Basic Functions

18.2.5.36 DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10)

DB21, ... DBX8.0 - 9.1	Activate machine-related protection zone 1 - 10
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Request to activate the machine-related protection zone The protection zone is immediately active. **Note** Activating the protection zone is only effective if it was already preactivated via an NC program.
Signal state 0	Request to deactivate the active machine-related protection zone. The protection zone is immediately inactive. **Note** Deactivating the protection zone is only active if it was preactivated via an NC program and was activated via the interface signal.
Corresponds with	DB21, ... DBX1.1 (enable protection zones) DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10) DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10) DB21, ... DBX272.0 - 273.1 (machine-related protection zone 1 - 10 preactivated) DB21, ... DBX274.0 – 275.1 (channel-specific protection zone 1 - 10 preactivated) DB21, ... DBX276.0 - 277.1 (machine-related protection zone 1 - 10 violated) DB21, ... DBX278.0 – 279.1 (channel-specific protection zone 1 - 10 violated)
Additional information	Function Manual Monitoring and Compensating

18.2.5.37 DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10)

DB21, ... DBX10.0 - 11.1	Activate machine-related protection zone 1 - 10
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Request to activate the machine-related protection zone. The protection zone is immediately active. **Note** Activating the protection zone is only effective if it was already preactivated via an NC program.

DB21, ... DBX10.0 - 11.1	Activate machine-related protection zone 1 - 10
Signal state 0	Request to deactivate the machine-related protection zone. The protection zone is immediately inactive. **Note** Deactivating the protection zone is only active if it was preactivated via an NC program and was activated via the interface signal.
Corresponds with	DB21, ... DBX1.1 (enable protection zones) DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10) DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10) DB21, ... DBX272.0 - 273.1 (machine-related protection zone 1 - 10 preactivated) DB21, ... DBX274.0 – 275.1 (channel-specific protection zone 1 - 10 preactivated) DB21, ... DBX276.0 - 277.1 (machine-related protection zone 1 - 10 violated) DB21, ... DBX278.0 – 279.1 (channel-specific protection zone 1 - 10 violated)
Additional information	Function Manual Monitoring and Compensating

18.2.5.38 DB21, ... DBX12.0 - 2 (geometry axis 1: activate handwheel)

DB21, ... DBX12.0 - 2	Geometry axis 1: Activate handwheel
Signal flow	PLC → NC
Update	Cyclic
Additional information	The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.

Bit coded: **maximum 3** handwheels

Bit 2	Bit 1	Bit 0	Number of the assigned handwheel
0	0	0	No handwheel assignment
0	0	1	1
0	1	0	2
1	0	0	3

Binary coded: **maximum 6** handwheels

Bit 2	Bit 1	Bit 0	Number of the assigned handwheel
0	0	0	No handwheel assignment
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6

Note

At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, then the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".

DB21, ... DBX12.0 - 2	Geometry axis 1: Activate handwheel
Corresponds with	DB21, ... DBX16.0 - 2 (geometry axis 2: activate handwheel) DB21, ... DBX20.0 - 2 (geometry axis 3: activate handwheel) DB21, ... DBX40.0 - 2 (geometry axis 1: handwheel active) DB21, ... DBX46.0 - 2 (geometry axis 2: handwheel active) DB21, ... DBX52.0 - 2 (geometry axis 3: handwheel active) MD11324 $MN_HANDWH_VDI_REPRESENTATION (representation of the handwheel number at the VDI interface)
Additional information	Function Manual Axes and Spindles

18.2.5.39 DB21, ... DBX12.3, 16.3, 20.3 (feedrate stop, geometry axes 1 / 2 / 3)

DB21, ... DBX12.3 DB21, ... DBX16.3 DB21, ... DBX20.3	Feedrate stop, geometry axis 1, 2, 3
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Feedrate disable for geometry axis is active.
Signal state 0	Feedrate disable for geometry axis is **not** active.
Additional information	• If the interface signal is set while traversing a geometry axis, then the geometry axis is braked down to standstill along its braking characteristic. • After the interface signal has been reset, traversing motion that was stopped is continued. • The position control is retained and the following error is eliminated. • If the interface signal is set, and a request is issued to traverse an axis, then the axis is not traversed. However, the traverse request is kept. When the interface signal is reset, the traverse request is immediately executed, i.e. the axis is traversed. • The interface signal is **only** active in the **JOG** mode.

18.2.5.40 DB21, ... DBX12.4 (geometry axis 1: traversing key lock)

DB21, ... DBX12.4	Geometry axis 1: Traversing key lock
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The traversing key lock for the geometry axis is active.
Signal state 0	The traversing key lock for the geometry axis is **not** active.
Additional information	**Note** When the traversing key disable is activated while traversing, then traversing is canceled.
Corresponds with	DB21, ... DBX16.4 (geometry axis 2: traversing key lock) DB21, ... DBX20.4 (geometry axis 3: traversing key lock) DB21, ... DBX12.6 - 7 (geometry axis 1: traversing keys "Plus" / "Minus") DB21, ... DBX16.6 - 7 (geometry axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX20.6 - 7 (geometry axis 3: traversing keys "Plus" / "Minus")
Additional information	Function Manual Axes and Spindles

18.2.5.41 DB21, ... DBX12.5 (geometry axis 1: rapid traverse override)

DB21, ... DBX12.5	Geometry axis 1: Rapid traverse override
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	When manually traversing the geometry axis using the traversing keys, when the interface signal is set, rapid traverse velocity becomes active (MD32010).
Signal state 0	When manually traversing the geometry axis using the traversing keys, the entered JOG velocity becomes active (SD41110 or MD32020).
Additional information	The signal is only active for continuous or incremental manual traversing in the JOG mode. The signal is irrelevant: • for reference point approach (JOG mode) • in the AUTOMATIC and MDI modes The rapid traverse velocity can be influenced using the rapid traverse override switch.
Corresponds with	DB21, ... DBX16.5 (geometry axis 2: rapid traverse override) DB21, ... DBX20.5 (geometry axis 3: rapid traverse override) DB21, ... DBX12.6 - 7 (geometry axis 1: traversing keys "Plus" / "Minus") DB21, ... DBX16.6 - 7 (geometry axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX20.6 - 7 (geometry axis 3: traversing keys "Plus" / "Minus") MD32010 $MA_JOG_VELO_RAPID (conventional rapid traverse) MD32020 $MA_JOG_VELO (conventional axis velocity) SD41110 $SN_JOG_SET_VELO (axis velocity for JOG)
Additional information	Function Manual Axes and Spindles

18.2.5.42 DB21, ... DBX12.6 - 7 (geometry axis 1: traversing keys "Plus" / "Minus")

DB21, ... DBX12.6 - 7	Geometry axis 1: traversing keys "Plus" / "Minus"
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to traverse the geometry axis using the traversing key in the positive/negative axis direction.
Signal state 0	A request has **not** been made to traverse the geometry axis using the traversing key in the positive/negative axis direction.

DB21, ... DBX12.6 - 7	Geometry axis 1: traversing keys "Plus" / "Minus"
Additional information	There is a request signal for every traversing key and/or axis direction:
	Bit 6: Traversing key "Minus" (for traversing in the negative axis direction)
	Bit 7: Traversing key "Plus" (for traversing in the positive axis direction)
	Depending on the active machine function, as well as the settings for jog and continuous operation (SD41050 and MD11300), different responses are initiated when the signal changes: 1. Continuous manual travel in the jog mode The geometry axis traverses in the direction concerned to the active limit switch as long as the interface signal is set to 1. 2. Continuous manual travel in continuous operation At the first edge change 0 →1, the geometry axis starts to traverse in the relevant direction. This traversing movement still continues when the edge changes from 1 → 0. Any new signal edge change 0 → 1 (same traversing direction!) stops the traversing movement. 3. Incremental manual travel in the jog mode With signal 1 the geometry axis starts to traverse at the set increment. If the signal changes to the 0 state before the increment is traversed, then traversing is interrupted. When the signal state changes to 1 again, traversing is continued. The geometry axis can be stopped and started several times as described above until it has traversed the complete increment. 4. Incremental manual travel in continuous operation At the first edge change 0 → 1 the geometry axis starts to traverse at the set increment. If another edge change 0 → 1 is performed with the same traverse signal before the geometry axis has traversed the increment, the traversing movement will be cancelled. The increment is no longer traversed to the end.
	Note • The interface signal is irrelevant in operating modes: – AUTOMATIC – MDI • If both traversing signals ("plus" and "minus") are set at the same time there is no traversing or the current traversing is aborted. • In contrast to machine axes, for geometry axes, only one geometry axis can be traversed at any one time using the traversing keys. • Using DB21, ... DBX12.4, 16.4, 20.4 (geometry axes 1, 2, 3: Traversing key lock) traversing using traversing keys can be locked. • In the following cases, the geometry axis cannot be traversed in the JOG operating mode: – The geometry axis is already being traversed via the axis-specific interface as machine axis. – If another geometry axis is already being traversed with the traversing keys.

DB21, ... DBX12.6 - 7	Geometry axis 1: traversing keys "Plus" / "Minus"
Corresponds with	DB21, ... DBX16.6 - 7 (geometry axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX20.6 - 7 (geometry axis 3: traversing keys "Plus" / "Minus") DB21, ... DBX12.4 (geometry axis 1: traversing key lock) DB21, ... DBX16.4 (geometry axis 2: traversing key lock) DB21, ... DBX20.4 (geometry axis 3: traversing key lock) DB31, ... DBX4.6 - 7 (traversing keys "Plus" / "Minus") MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD (INC and REF in jog mode) SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD (jog/continuous operation for JOG, continuous)
Additional information	Function Manual Axes and Spindles

18.2.5.43 DB21, ... DBX13.0 - 6 (geometry axis 1: machine function request)

DB21, ... DBX13.0 - 6	Geometry axis 1: machine function request	
Signal flow	PLC → NC	
Update	Cyclic	
Signal state 1	The machine function to manually traverse the geometry axis has been requested.	
Signal state 0	The machine function to manually traverse the geometry axis has **not** been requested.	
Additional information	There is a request signal for every machine function to manually traverse the geometry axis in the JOG mode.	
	Bit 0	INC1
	Bit 1	INC10
	Bit 2	INC100
	Bit 3	INC1000
	Bit 4	INC10000
	Bit 5	INCvar
	Bit 6	Continuous manual travel
	Incremental manual travel In addition to five fixed increment sizes (default setting in MD11330: INC1, INC10, INC100, INC1000 and INC10000), a variable increment size (INCvar) that can be set via the setting data SD41010 is also available. The distance evaluation of one increment for fixed and variable increment sizes is performed via the axis-specific machine data MD31090. When pressing the "Plus" or "Minus" traversing key, or by rotating the electronic handwheel, the geometry axis starts to traverse in the appropriate direction, corresponding to the number of increments of the active machine function. **Continuous manual travel** For continuous manual travel, the plus and minus traversing keys are selected to move the geometry axis continuously and the appropriate direction. **Notes** • If several requests are set simultaneously, no machine function becomes active. • If a geometry axis is currently being traversed via a machine function, the movement is aborted through deselection or change of the machine function.	

DB21, ... DBX13.0 - 6	Geometry axis 1: machine function request
Corresponds with	DB21, ... DBX17.0 - 6 (geometry axis 2: machine function request) DB21, ... DBX21.0 - 6 (geometry axis 3: machine function request) DB21, ... DBX41.0 - 6 (geometry axis 1: active machine function) DB21, ... DBX47.0 - 6 (geometry axis 2: active machine function) DB21, ... DBX53.0 - 6 (geometry axis 3: active machine function) MD11320 $MN_HANDWH_IMP_PER_LATCH (handwheel pulses per detent position) MD11330 $MN_JOG_INCR_SIZE_TAB (increment size for INC/handwheel) MD31090 $MA_JOG_INCR_WEIGHT (evaluation of an increment for INC/handwheel) SD41010 $SN_JOG_VAR_INCR_SIZE (size of the variable increment for JOG)
Additional information	Function Manual Axes and Spindles

18.2.5.44 DB21, ... DBX15.0 (geometry axis 1: invert handwheel direction of rotation)

DB21, ... DBX15.0	Geometry axis 1: invert handwheel direction of rotation
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to invert the direction of rotation of the handwheel assigned to the geometry axis.
Signal state 0	A request has **not** been made to invert the direction of rotation of the handwheel assigned to the geometry axis.
Additional information	**Note** It is only permissible to change the interface signal when the geometry axis is at a standstill. **Application examples** • The direction of movement of the handwheel does not match the expected direction of the axis. • A handwheel is assigned to several axes with different orientations.
Corresponds with	DB21, ... DBX19.0 (geometry axis 2: invert handwheel direction of rotation) DB21, ... DBX23.0 (geometry axis 3: invert handwheel direction of rotation) DB21, ... DBX43.0 (geometry axis 1: handwheel direction of rotation inversion active) DB21, ... DBX49.0 (geometry axis 2: handwheel direction of rotation inversion active) DB21, ... DBX55.0 (geometry axis 3: handwheel direction of rotation inversion active)
Additional information	Function Manual Axes and Spindles

See also

DB21 - DB30, decoded M signals (Page 391)

18.2.5.45 DB21, ... DBX16.0 - 2 (geometry axis 2: activate handwheel)

DB21, ... DBX16.0 - 2	Geometry axis 2: Activate handwheel
Additional information	See DB21, ... DBX12.0 - 2 (geometry axis 1: activate handwheel) (Page 521).

18.2.5.46 DB21, ... DBX16.4 (geometry axis 2: traversing key lock)

DB21, ... DBX16.4	Geometry axis 2: Traversing key lock
Additional information	See DB21, ... DB21, ... DBX12.4 (geometry axis 1: traversing key lock) (Page 522).

18.2.5.47 DB21, ... DBX16.5 (geometry axis 2: rapid traverse override)

DB21, ... DBX16.5	Geometry axis 2: Rapid traverse override
Additional information	See DB21, ... DBX12.5 (geometry axis 1: rapid traverse override) (Page 523).

18.2.5.48 DB21, ... DBX16.6 - 7 (geometry axis 2: traversing keys "Plus" / "Minus")

DB21, ... DBX16.6 - 7	Geometry axis 2: traversing keys "Plus" / "Minus"
Additional information	See DB21, ... DBX12.6 - 7 (geometry axis 1: traversing keys "Plus" / "Minus") (Page 523).

18.2.5.49 DB21, ... DBX17.0 - 6 (geometry axis 2: machine function request)

DB21, ... DBX17.0 - 6	Geometry axis 2: machine function request
Additional information	See DB21, ... DBX13.0 - 6 (geometry axis 1: machine function request) (Page 525).

18.2.5.50 DB21, ... DBX19.0 (geometry axis 2: invert handwheel direction of rotation)

DB21, ... DBX19.0	Geometry axis 2: invert handwheel direction of rotation
Additional information	See DB21, ... DBX15.0 (geometry axis 1: invert handwheel direction of rotation) (Page 526).

18.2.5.51 DB21, ... DBX20.0 - 2 (geometry axis 3: activate handwheel)

DB21, ... DBX20.0 - 2:	Geometry axis 3: Activate handwheel
Additional information	See DB21, ... DBX12.0 - 2 (geometry axis 1: activate handwheel) (Page 521).

18.2.5.52 DB21, ... DBX20.4 (geometry axis 3: traversing key lock)

DB21, ... DBX20.4:	Geometry axis 3: Traversing key lock
Additional information	See DB21, ... DBX12.4 (geometry axis 1: traversing key lock) (Page 522).

18.2.5.53 DB21, ... DBX20.5 (geometry axis 3: rapid traverse override)

DB21, ... DBX20.5	Geometry axis 3: Rapid traverse override
Additional information	See DB21, ... DBX12.5 (geometry axis 1: rapid traverse override) (Page 523).

18.2.5.54 DB21, ... DBX20.6 - 7 (geometry axis 3: traversing keys "Plus" / "Minus")

DB21, ... DBX20.6 - 7:	Geometry axis 3: traversing keys "Plus" / "Minus"
Additional information	See DB21, ... DBX12.6 - 7 (geometry axis 1: traversing keys "Plus" / "Minus") (Page 523).

18.2.5.55 DB21, ... DBX21.0 - 6 (geometry axis 3: machine function request)

DB21, ... DBX21.0 - 6	Geometry axis 3: machine function request
Additional information	See DB21, ... DBX13.0 - 6 (geometry axis 1: machine function request) (Page 525).

18.2.5.56 DB21, ... DBX23.0 (geometry axis 3: invert handwheel direction of rotation)

DB21, ... DBX23.0	Geometry axis 3: invert handwheel direction of rotation
Additional information	See DB21, ... DBX15.0 (geometry axis 1: invert handwheel direction of rotation) (Page 526).

18.2.5.57 DB21, ... DBX24.1 (activate configured stop (CST) using the operating software)

DB21, ... DBX24.1	Activate configured stop (CST) using the operating software
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Activation of the configured stop is requested.
Signal state 0	Activation of the configured stop is not requested.
Additional information	"Activate configured stop" (CST) is selected via the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, … DBX24.1. Depending on the value of FB1 parameter MMCToIf, the interface signal is transferred from the basic PLC program to the NC/PLC interface signal DB21, … DBX7.6: • "TRUE": Transmission" • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, … DBX7.6 (activate configured stop)

18.2.5.58 DB21, ... DBX24.3 (handwheel offset (DRF) requested)

DB21, ... DBX24.3	Handwheel offset (DRF) selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Handwheel offset (DRF) is selected
Signal state 0	Handwheel offset (DRF) is **not** selected.
Additional information	As soon as handwheel offset (DRF) is active, the handwheel offset can be changed in AUTOMATIC or MDI mode using the handwheel assigned to the axis. **Note** Handwheel offset (DRF) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.3. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX0.3: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX0.3 (activate handwheel offset (DRF))
Additional information	Function Manual Axes and Spindles

18.2.5.59 DB21, ... DBX24.4 (select NC-associated M01)

DB21, ... DBX24.4	Associated auxiliary function selected
Signal flow	HMI → PLC
Update	Job-controlled
Signal state 1	Associated auxiliary function is selected.
Signal state 0	Associated auxiliary function is **not** selected.
Additional information	"Associated auxiliary function" (M-1) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.4. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX30.5: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX30.5 (activate associated auxiliary function) DB21, ... DBX318.5 (associated auxiliary function active) MD22254 $MC_AUXFU_ASSOC_M0_VALUE (additional M function for program stop) MD22256 $MC_AUXFU_ASSOC_M1_VALUE (additional M function for conditional stop)
Additional information	Function Manual Basic Functions

18.2.5.60 DB21, ... DBX24.5 (M01 requested)

DB21, ... DBX24.5	M01 selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Program control "Conditional stop"M01 is selected.
Signal state 0	Program control "Conditional stop"M01 is **not** selected.
Additional information	Program control "Conditional stop" M01 is selected from the SINUMERIK Operate user interface: Operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.5. The interface signal is then transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX0.5: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX0.5 (activate M01) DB21, ... DBX32.5 (M00/M01 active)
Additional information	Function Manual Basic Functions

18.2.5.61 DB21, ... DBX24.6 (dry run feedrate selected)

DB21, ... DBX24.6	Dry run feedrate (DRY) selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Dry run feedrate (DRY) is selected.
Signal state 0	Dry run feedrate (DRY) is **not** selected.
Additional information	**Note** Dry run feedrate (DRY) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.6. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX0.6: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE". **Note** The value for the dry run feedrate is set using: SD42100 $SC_DRY_RUN_FEED
Corresponds with	DB21, ... DBX0.6 (activate dry run (DRY) feedrate) SD42100 $SC_DRY_RUN_FEED (dry run feedrate) SD42101 $SC_DRY_RUN_FEED_MODE (mode for dry run velocity)
Additional information	Function Manual Basic Functions Operating Manual, Turning or Milling or Grinding

18.2.5.62 DB21, ... DBX25.3 (feedrate override selected for rapid traverse)

DB21, ... DBX25.3	Feedrate override selected for rapid traverse
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Feedrate override is selected for rapid traverse
Signal state 0	Feedrate override is **not** selected for rapid traverse
Additional information	If there is no dedicated rapid traverse override switch on the machine control panel, you can then toggle between feedrate override and rapid traverse override. Rapid traverse override can be selected by selecting feedrate override for rapid traverse at the user interface. The interface signal is set: DB21, ... DBX25.3 = 1 The following interface signals are set by the PLC basic program: • Rapid traverse override active = feedrate override for rapid traverse selected DB21, ... DBX6.6 = DB21, ... DBX25.3 • Rapid traverse override = feedrate override DB21, ... DBB5 = DB21, ... DBB4 Override factors> 100% are limited to 100%.
	Note The PLC user program can also initiate the switchover between rapid traverse and feedrate override. To do this, the following interface signals must be set: • Rapid traverse override active DB21, ... DBX6.6 = 1 • Rapid traverse override = feedrate override DB21, ... DBB5 = DB21, ... DBB4
Corresponds with	DB21, ... DBB4 (feedrate override) DB21, ... DBB5 (rapid traverse override) DB21, ... DBX6.6 (rapid traverse override active)
Additional information	Function Manual Basic Functions

18.2.5.63 DB21, ... DBX25.7 (program test (PRT) requested)

DB21, ... DBX25.7	Program test (PRT) selected
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Program test (PRT) is selected
Signal state 0	Program test (PRT) is **not** selected.

DB21, ... DBX25.7	Program test (PRT) selected
Additional information	If program test (PRT) is active, the machine axes do not move during the execution of a block or NC program. However, the axis movements are displayed on the user interface with changing setpoint position values. **Note** While program test (PRT) is **active**, all traversing motions take place in the channel depending on the current mode of the axis: • Axes: "Axis disable" is active ⇒ no real traversing motions • Spindles: "Axis disable" is **not** active ⇒ real traversing motions **Note** Program test (PRT) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX25.7. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX1.7: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE". **Notice** Because of the axis disable, the assignment of a tool magazine is not changed while the program is being tested. The user / machine manufacturer must utilize a suitable PLC user program to ensure that the NC-internal tool management and the actual assignment of the tool magazine remain consistent.
Corresponds with	DB21, ... DBX1.7 (activate program test (PRT)) DB21, ... DBX33.7 (program test (PRT) active)
Additional information	Function Manual Basic Functions

18.2.5.64 DB21, ... DBX26.0 ... 7 ("Skip block" (SKP) selected)

DB21, ... DBX26.0 ... 7	"Skip block" (SKP) selected
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	"Skip block" (SKP) of the skip level is selected.
Signal state 0	"Skip block" (SKP) of the skip level is **not** selected.

DB21, ... DBX26.0 ... 7	"Skip block" (SKP) selected
Additional information	Bit 0 ... bit 7 are assigned to the skip levels 0 ... 7. **Note** We recommend that the interface signal is already set before the NC program starts. **Note** For multiple consecutive skip blocks, the blocks are only skipped if the interface signal was already present **before** the **first** skip block of the block sequence was decoded. **Note** "Skip block" (SKP) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX26.0 ...7. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX2.0 ... 7: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX2.0 ... 7 (activate "skip block")
Additional information	Function Manual Basic Functions

18.2.5.65 DB21, ... DBX29.0 - 3 (activate fixed feedrate 1 - 4, path/geometry axes)

DB 21, ... DBX29.0 - 3	Activate fixed feedrate 1 - 4, path/geometry axes				
Signal flow	PLC → NC				
Update	Cyclic				
Additional information	With the interface signals, in the **AUTOMATIC** mode, instead of the programmed feedrate or the configured JOG velocities, the fixed feedrate - parameterized using machine data - is activated.				
	Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Meaning**
	0	0	0	0	Fixed feedrate not selected
	0	0	0	1	Fixed feedrate 1
	0	0	1	0	Fixed feedrate 2
	0	1	0	0	Fixed feedrate 3
	1	0	0	0	Fixed feedrate 4
	The fixed feedrates are parameterized using the following machine data: • Linear axes: MD12202 $MN_PERMANENT_FEED • Rotary axes: MD12204 $MN_PERMANENT_ROT_AX_FEED				
	Notes • The fixed feedrate is not used for: – Spindles – Positioning axes – Tapping • The fixed feedrate is always interpreted as linear feedrate. Switchover to linear feedrate is performed internally in the control system also when revolutional feedrate is active.				

DB 21, ... DBX29.0 - 3	Activate fixed feedrate 1 - 4, path/geometry axes
Corresponds with	MD12200 $MN_RUN_OVERRIDE_0 (traversing behavior with override 0) MD12202 $MN_PERMANENT_FEED (fixed feedrates for linear axes) MD12204 $MN_PERMANENT_ROT_AX_FEED (fixed feedrates for rotary axes)
Additional information	Function Manual Basic Functions

18.2.5.66 DB21, ... DBX29.4 (activate PTP travel)

DB21, ... DBX29.4	Activate PTP travel
Signal flow	PLC → NC
Edge change 0 → 1	Activate PTP (Point-To-Point) travel
Edge change 1 → 0	Activate CP (Continuous-Path) travel
Additional information	Using this signal, in the the jog mode, you can switch between Cartesian path movement (CP) and the Cartesian PTP travel. **Note** DB21, ... DBX29.4 is only relevant in the JOG mode when the transformation is active.
Corresponds with	DB21, ... DBX317.6 (PTP travel active)
Additional information	Function Manual Transformations

18.2.5.67 DB21, ... DBX29.5 (tool manager: Deactivate workpiece counter)

DB21, ... DBX29.5	TOOLMAN: Deactivate workpiece counter
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	Deactivate workpiece counter monitoring is requested.
Signal state 0	Deactivate workpiece counter monitoring is **not** requested.
Additional information	The workpiece counter monitoring allows workpieces to be counted that depend on the process, the workpiece material or other factors. Workpiece counter monitoring can be deactivated using the interface signal
Corresponds with	SETPIECE (function) TMPCIT (PI service) $TC_MOP4 (remaining unit quantity) $TC_TP9 (type of tool monitoring)
Additional information	Function Manual Tool Management

18.2.5.68 DB21, ... DBX29.6 (tool manager: Deactivating wear monitoring)

DB21, ... DBX29.6	TOOLMAN: Deactivate wear monitoring
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	Deactivate wear monitoring is requested.
Signal state 0	Deactivate wear monitoring is **not** requested.

DB21, ... DBX29.6	TOOLMAN: Deactivate wear monitoring
Corresponds with	$TC_MOP15 (wear setpoint or additive offset value) $TC_MOP5 (wear pre-warning limit or additive offset pre-warning limit) $TC_MOP6 (wear value or additive offset setpoint) $TC_TP9 (type of tool monitoring) MD18080 $MN_MM_TOOL_MANAGEMENT_MASK (memory reserved for the tool management) MD20310 TOOL_MANAGEMENT_MASK bit 17 (activation of the tool management functions)
Additional information	Function Manual Tool Management

18.2.5.69 DB21, ... DBX29.7 (tool manager: Tool lock not active)

DB21, ... DBX29.7	TOOLMAN: Tool lock not active
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	Tool lock **not**active.
Signal state 0	Tool lock is active
Additional information	A tool assumes the "locked" state if the actual value of the active monitoring function (workpiece count, tool life or wear) has reached a value of zero. If the tool is still being used, then it remains in use (for machining) until the next tool change. After that the tool can no longer be used. Using the interface signal, it can be set that the NC does not take the "locked" state into account when searching for a tool that can be used. The interface signal is **not** active if the tool selection is realized using in it blocks (reset and start mode mask).
Corresponds with	MD22562 $MC_TOOL_CHANGE_ERROR_MODE (response for tool change errors)
Additional information	Function Manual Tool Management

18.2.5.70 DB21 ... DBX30.0 - 2 (activate contour handwheel)

DB21, ... DBX30.0 - 2:	Activate contour handwheel
Signal flow	PLC → NC
Update	Cyclic

DB21, ... DBX30.0 - 2:	Activate contour handwheel			
Additional information	The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.			
	Bit coded: maximum 3 handwheels			
	Bit 2	**Bit 1**	**Bit 0**	**Number of the assigned handwheel**
	0	0	0	No handwheel assignment
	0	0	1	1
	0	1	0	2
	1	0	0	3
	Binary coded: maximum 6 handwheels			
	Bit 2	**Bit 1**	**Bit 0**	**Number of the assigned handwheel**
	0	0	0	No handwheel assignment
	0	0	1	1
	0	1	0	2
	0	1	1	3
	1	0	0	4
	1	0	1	5
	1	1	0	6
	Note At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, then the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".			
Corresponds with	DB21, ... DBX37.0 - 2 (contour handwheel active) MD11324 $MN_HANDWH_VDI_REPRESENTATION (representation of the handwheel number at the VDI interface)			
Additional information	Function Manual Axes and Spindles			

18.2.5.71 DB21, ... DBX30.3 (simulation contour handwheel: Activate)

DB21, ... DBX30.3	Simulation contour handwheel: Activation
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activate contour handwheel simulation
Signal state 0	Deactivate contour handwheel simulation
Additional information	During simulation, the feedrate is no longer defined by the contour handwheel, but traversing occurs with the programmed feedrate on the contour. When deactivated, the actual traversing motion is braked along the braking ramp. **Note** Simulation is only effective in AUTOMATIC mode and can only be activated when the contour handwheel is activated.
Corresponds with	DB21, ... DBX30.4 (simulation contour handwheel: Negative direction)
Additional information	Function Manual Axes and Spindles

18.2.5.72 DB21, ... DBX30.4 (simulation contour handwheel: Negative direction)

DB21, ... DBX30.4	Simulation contour handwheel: Negative direction
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	When simulating the contour handwheel, it is traversing on the contour in the opposite direction to the programmed direction.
Signal state 0	When simulating the contour handwheel, it is traversing on the contour in the direction to the programmed direction.
Additional information	When switching over the traversing direction, the actual traversing motion is braked along the braking ramp and then traversed in the opposite direction.
Corresponds with	DB21, ... DBX30.3 (simulation contour handwheel: Activate)
Additional information	Function Manual Axes and Spindles

18.2.5.73 DB21, ... DBX30.5 (activate M0 / M1 associated with NC)

DB21, ... DBX30.5	Activate associated auxiliary function
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	Activation of "Associated auxiliary function" is requested.
Signal state 0	Activation of "Associated auxiliary function" is **not** requested.
Additional information	"Associated auxiliary function" (M-1) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.4. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX30.5: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX24.4 (associated auxiliary function selected) DB21, ... DBX318.5 (associated auxiliary function active) MD22254 $MC_AUXFU_ASSOC_M0_VALUE (additional M function for program stop) MD22256 $MC_AUXFU_ASSOC_M1_VALUE (additional M function for conditional stop)
Additional information	Function Manual Basic Functions

18.2.5.74 DB21, ... DBX30.6 (JOG circular travel)

DB21, ... DBX30.6	JOG circular travel
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The "Circular travel in JOG" function is requested.
Signal state 0	The "Circular travel in JOG" function is **not** requested.

DB21, ... DBX30.6	JOG circular travel
Additional information	As soon as the function is active (see DB21, ... DBX377.6), the machine operator can simultaneously traverse the two geometry axes of the active plane along an arc using the traverse keys or handwheel. **Application** This function is used for machine tools that are exclusively operated manually. **Note** The following preconditions applying for the "Circular travel in JOG" function: • The "Circular travel in JOG" function can be activated only in the JOG mode. The function cannot be activated if the machine functions JOG-REPOS and JOG-REF are active. • The axes that participate in traversing must be referenced. • It is not permissible that the active plane is inclined in space.
Corresponds with	DB21, ... DBX377.6 (JOG circular travel active)
Additional information	Function Manual Axes and Spindles

18.2.5.75 DB21, ... DBX31.0 - 2 (REPOS mode)

DB21, ... DBX31.0 - 2	REPOS mode			
Signal flow	PLC → NC			
Update	Cyclic			
Additional information	Selecting the REPOS mode:			
	Bit 2	**Bit 1**	**Bit 0**	**REPOS mode**
	0	0	0	No REPOS mode active
	0	0	1	RMB: Repositioning to start of block
	0	1	0	RMI: Repositioning to interrupt point
	0	1	1	RME: Repositioning to end of block
	1	0	0	RMN: Repositioning to the nearest point on the path
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.4 (REPOS mode change) DB31, ... DBX10.0 (REPOS delay) DB31, ... DBX72.0 (REPOS delay) DB21, ... DBX319.0 (REPOS mode change acknowledgment) DB21, ... DBX319.1 - 3 (active REPOS mode) DB21, ... DBX319.5 (REPOS acknowledgment delay) DB31, ... DBX70.0 (REPOS offset) DB31, ... DBX70.1 (REPOS offset valid) DB31, ... DBX70.2 (REPOS delay acknowledgment) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)			
Additional information	Function Manual Basic Functions			

18.2.5.76 DB21, ... DBX31.4 (REPOS activation)

DB21, ... DBX31.4	REPOS activation
Signal flow	PLC → NC
Update	Cyclic
Signal state 0 → 1	Request to activate the "REPOS" function for the actual main run block. The following become active: • DB21, ... DBX31.0 - 2 (REPOS mode) • DB31, ... DBX10.0 (REPOS delay)
Signal state 1 → 0	Reset of the last request.
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.0 - 2 (REPOS mode) DB31, ... DBX10.0 (REPOS delay) DB31, ... DBX72.0 (REPOS delay) DB21, ... DBX319.0 (REPOS mode change acknowledgment) DB21, ... DBX319.1 - 3 (active REPOS mode) DB21, ... DBX319.5 (REPOS acknowledgment delay) DB31, ... DBX70.0 (REPOS offset) DB31, ... DBX70.1 (REPOS offset valid) DB31, ... DBX70.2 (REPOS delay acknowledgment) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)
Additional information	Function Manual Basic Functions

18.2.5.77 DB21, ... DBX31.5 (contour handwheel: invert handwheel direction of rotation)

DB21, ... DBX31.5	Contour handwheel: invert handwheel direction of rotation
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to invert the direction of rotation of the contour handwheel.
Signal state 0	A request has **not** been made to invert the direction of rotation of the contour handwheel.
Additional information	**Note** It is only permissible to change the interface signal when the axis is at a standstill.
Corresponds with	DB21, ... DBX39.5 (Contour handwheel: handwheel direction of rotation inversion active)
Additional information	Function Manual Axes and Spindles

18.2.5.78 DB21, ... DBX32.1 (RESU: retrace mode active)

DB21, ... DBX32.1	RESU: Retrace mode active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The retrace mode is active.
Signal state 0	The retrace mode is **not** active.

DB21, ... DBX32.1	RESU: Retrace mode active
Additional information	The interface signal is set, if DB21, ... DBX0.1 == 1 The interface signal is reset, if DB21, ... DBX0.2 == 1
Corresponds with	DB21, ... DBX0.1 (RESU: backward/forward) DB21, ... DBX0.2 (RESU: start retrace support) DB21, ... DBX32.1 (RESU: retrace mode active) DB21, ... DBX32.2 (RESU: retrace support active)
Additional information	Function Manual Technologies

18.2.5.79 DB21, ... DBX32.2 (retrace support active)

DB21, ... DBX32.2	RESU: Retrace support active
Edge evaluation:	No
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Retrace support active
Signal state 0	Retrace support is **not** active.
Additional information	The interface signal is set, if DB21, ... DBX0.2 == 1 is identified. The interface signal is reset if the last action block has been completed.
Corresponds with	DB21, ... DBX0.1 (RESU: backward/forward) DB21, ... DBX0.2 (RESU: start retrace support) DB21, ... DBX32.1 (RESU: retrace mode active) DB21, ... DBX32.2 (RESU: retrace support active)
Additional information	Function Manual Technologies

18.2.5.80 DB21, ... DBX32.3 (action block active)

DB21, ... DBX32.3	Action block active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	An action block is active / is being executed.
Signal state 0	No action block is active.
Corresponds with	DB21, ... DBX33.4 (block search active) DB21, ... DBX32.6 (last action block active)
Additional information	Function Manual Basic Functions

18.2.5.81 DB21, ... DBX32.4 (approach block active)

DB21, ... DBX32.4	Approach block active
Signal flow	NC → PLC
Update	Cyclic

DB21, ... DBX32.4	Approach block active
Signal state 1	The approach block to continue the NC program for block search, type 2 "Block search with calculation at the contour" is active.
Signal state 0	No approach block is active.
Additional information	Function Manual Basic Functions

18.2.5.82 DB21, ... DBX32.5 (M00/M01 active)

DB21, ... DBX32.5	M00/M01 active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Auxiliary function M00 or M01 is active. The block in which the auxiliary function is programmed, is executed and the help functions output. Program state "stopped"
Signal state 0	Auxiliary function M00 or M01 is not active.
Corresponds with	DB21, ... DBX0.5 (activate M01) DB21, ... DBX24.5 (M01 selected)
Additional information	Function Manual Basic Functions

18.2.5.83 DB21, ... DBX32.6 (last action block active)

DB21, ... DBX32.6	Last action block active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The last action block is active / is being executed.
Signal state 0	The last action block is not active.
Additional information	If the last action block is active, this means that all the action blocks on the NC side have been processed and the actions on the PLC side (ASUB, FC) or the operator such as overstore, mode change according to JOG/REPOS are possible. In this way the PLC, for example, can still perform a tool change before the start of movement. Action blocks contain the actions collected during "Block search with computation" such as • Help function outputs H, M00, M01, M.. • Tool programming T, D, DL • Spindle programming S value, M3/M4/M5/M19, SPOS • Feed programming, F
Corresponds with	DB21, ... DBX33.4 (block search active)
Additional information	Function Manual Basic Functions

18.2.5.84 DB21, ... DBX33.0 (referencing active)

DB21, ... DBX33.0	Referencing active
Signal flow	NC → PLC
Update	Cyclic
Edge change 0 → 1	Referencing is active
Edge change 1 → 0	No effect
Additional information	Feedback signal for the request to activate referencing: 1. DB21, ... DBX1.0: 0 → 1 (request) 2. DB21, ... DBX33.0: 0 → 1 (feedback signal)
Corresponds with	DB21, ... DBX1.0 (activate referencing)
Additional information	Function Manual Axes and Spindles

18.2.5.85 DB21, ... DBX33.2 (revolutional feedrate active)

DB31, ... DBX33.2	Revolutional feedrate active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Revolutional feedrate (G95) is active.
Signal state 0	Revolutional feedrate (G95) is **not** active.
Further information	The interface signal indicates that, in AUTOMATIC mode, path or synchronized axes move at revolutional feedrate.
Additional information	Function Manual Axes and Spindles

18.2.5.86 DB21, ... DBX33.3 (handwheel override active)

DB21, ... DBX33.3	Handwheel override active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The "Handwheel override in automatic mode" function is active for the programmed path axes.
Signal state 0	The function "Handwheel override in automatic mode" is **not** active for the programmed path axes.
Additional information	The handwheel pulses of the 1st geometry axis function as a velocity override on the programmed path velocity. In the following cases, the override is inactive: • The path axes have reached the programmed target position. • The distance-to-go has been deleted. • Reset was initiated.
Corresponds with	DB21, ... DBX6.2 (delete distance-to-go)
Additional information	Function Manual Axes and Spindles

18.2.5.87 DB21, ... DBX33.4 (block search active)

DB21, ...DBX33.4	Block search active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The "block search" function is active. It was selected from the user interface and started using the interface signal: DB21, ... DBX7.1 (NC Start)
Signal state 0	The "block search" function is not active - or the search target was found.
Corresponds with	DB21, ... DBX7.1 (NC Start)
Additional information	Function Manual Basic Functions

18.2.5.88 DB21, ... DBX33.5 (M02 / M30 active)

DB21, ... DBX33.5	M02 / M30 active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	End of program M02 / M30 is active. The interface signal is set after: • A program reset has been completed (M02, M30 or M17) **Note:** If traversing motion is programmed in the block of M02, M30 or M17, then the signal is only set after all of the axes have reached their target positions. • Executed the PROG_EVENT program (PROG_EVENT.SPF) after: – end of program reset (M02 / M30) – Channel reset – Warm restart (power on) – Output of the last action block after a block search
Signal state 0	End of program M02 / M30 is **not** active The interface signal is reset or remains reset: • after the start and while executing an NC program, asynchronous subprogram (ASUB) or PROG_EVENT program. **Note:** The last signal state is kept after the start of a block search with calculation in the "Program test" mode (SERUPRO). • After a program cancellation as the result of an alarm. • During and after executing a channel reset **without** executing a PROG_EVENT program. • During and after the control system powers up **without** executing a PROG_EVENT program.

DB21, ... DBX33.5	M02 / M30 active
Additional information	• In the last part program block of an NC program, it is not permissible to program the following functions: – Auxiliary functions, which initiate a read-in stop – Programmed spindle speed (S value), that should be effective beyond the end of program (M02 / M30) • After a program reset has been completed (M02 / M30), then the signal is set to a value of 1 While processing a subsequent PROG_EVENT program, the signal is set to 0, and after completion of the PROG_EVENT program, is again set to 1. **Note** The signal is not suitable to automatically initiate follow-on functions such as tool counting, bar feed etc. In order to identify the completion of the previous processing (NC program, ASUB, PROG_EVENT, etc.) the end of program command (M02 / M30) must be written to its own dedicated part program block. Command (M02 / M30) or the decoded signal of the M function can be used as trigger.
Additional information	• Function Manual Basic Functions • DB21 - DB30, transferred M and S functions: (Page 388) or "DB21 - DB30, decoded M signals (Page 391)"

18.2.5.89 DB21, … DBX33.6 (transformation active)

DB21, ... DBX33.6	Transformation active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	A transformation is not active.
Signal state 0	A transformation is not active.
Additional information	In the channel - in the active NC program - one of the following commands TRANSMIT, TRACYL, TRAANG or TRAORI has been programmed. The corresponding block has been processed and the corresponding transformation is active.
Additional information	• Function Manual Basic Functions • Function Manual Transformations

18.2.5.90 DB21, ... DBX33.7 (program test active)

DB21, ... DBX33.7	Program test (PRT) active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Program test (PRT) is active.
Signal state 0	Program test (PRT) is **not** active.

DB21, ... DBX33.7	Program test (PRT) active
Additional information	If program test (PRT) is active, the machine axes do not move during the execution of a block or NC program. However, the axis movements are displayed on the user interface with changing setpoint position values. **Note** While program test (PRT) is active, all traversing motions of the **axes**, but **not** the **spindles**, take place under "Axis disable." **Note** Program test (PRT) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX25.7. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX1.7: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE". **Notice** Because of the axis disable, the assignment of a tool magazine is not changed while the program is being tested. The user / machine manufacturer must utilize a suitable PLC user program to ensure that the NC-internal tool management and the actual assignment of the tool magazine remain consistent.
Corresponds with	DB21, ... DBX1.7 (activate program test (PRT)) DB21, ... DBX25.7 (program test (PRT) requested)
Additional information	Function Manual Basic Functions

18.2.5.91 DB21, ... DBX35.0 (program state "running")

DB21, ... DBX35.0	Program state "running"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Program state == "running"
Signal state 0	Program state ≠ "running"

DB21, ... DBX35.0	Program state "running"
Additional information	The signal is set after starting the NC program using DB21, ... DBX7.1 = 1. The signal remains set if NC program execution is stopped using DB21, ... DBX6.1 = 1. The signal is **not** reset if NC program execution is **stopped** as a result of the following events: • A feedrate disable or spindle disable is active • DB21, ... DBX6.1 (read-in disable) • Feedrate override: 0 % • A spindle and axis monitoring function responds • Position setpoints are entered in the NC program for axes in the "follow-up mode," for axes without "controller enable," or for "parking axes"
	Program states: • DB21, ... DBX35.4 (program state "aborted") • DB21, ... DBX35.3 (program state "interrupted") • DB21, ... DBX35.2 (program state "stopped") • DB21, ... DBX35.1 (program state "wait") • DB21, ... DBX35.0 (program state "running")
Corresponds with	DB21, ... DBX6.1 = 1 (read-in disable) DB21, ... DBX7.1 = 1 (NC Start) DB21, ... DBX35.4 (program state "aborted") DB21, ... DBX35.3 (program state "interrupted") DB21, ... DBX35.2 (program state "stopped") DB21, ... DBX35.1 (program state "wait")
Additional information	Function Manual Basic Functions

18.2.5.92 DB21, ... DBX35.1 (program state "wait")

DB21, ... DBX35.1	Program state "Wait"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Program state == "wait"
Signal state 0	Program state ≠ "wait"
Additional information	The "Wait" program state is assumed if, in the NC program, with `WAIT_M` or `WAIT_E` the system waits for synchronization with an NC program or another channel.
	Program states: • DB21, ... DBX35.4 (program state "aborted") • DB21, ... DBX35.3 (program state "interrupted") • DB21, ... DBX35.2 (program state "stopped") • DB21, ... DBX35.1 (program state "wait") • DB21, ... DBX35.0 (program state "running")

DB21, ... DBX35.1	Program state "Wait"
Corresponds with	DB21, ... DBX35.4 (program state "aborted") DB21, ... DBX35.3 (program state "interrupted") DB21, ... DBX35.2 (program state "stopped") DB21, ... DBX35.0 (program state "running")
Additional information	Function Manual Basic Functions

18.2.5.93 DB21, ... DBX35.2 (program state "stopped")

DB21, ... DBX35.2	Program state"stopped"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Program state == "stopped"
Signal state 0	Program state ≠ "stopped"
Additional information	Events, which lead to the NC program being stopped: • DB21, ... DBX7.3 (NC Stop) • DB21, ... DBX7.4 (NC Stop axes plus spindles) • DB21, ... DBX7.2 (NC Stop at the block limit) • Programmed command `M00` or `M01` • Single-block mode
	Program states: • DB21, ... DBX35.4 (program state "aborted") • DB21, ... DBX35.3 (program state "interrupted") • DB21, ... DBX35.2 (program state "stopped") • DB21, ... DBX35.1 (program state "wait") • DB21, ... DBX35.0 (program state "running")
Corresponds with	DB21, ... DBX7.3 (NC Stop) DB21, ... DBX7.4 (NC Stop axes plus spindles) DB21, ... DBX7.2 (NC Stop at the block limit) DB21, ... DBX35.4 (program state "aborted") DB21, ... DBX35.3 (program state "interrupted") DB21, ... DBX35.1 (program state "wait") DB21, ... DBX35.0 (program state "running")
Additional information	Function Manual Basic Functions

18.2.5.94 DB21, ... DBX35.3 (program state "interrupted")

DB21, ... DBX35.3	Program state "interrupted"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Program state == "interrupted"
Signal state 0	Program state ≠ "interrupted"

DB21, ... DBX35.3	Program state "interrupted"
Additional information	The signal indicates that the interrupted NC program can be continued with NC start (DB21, ... DBX7.1 = 1). For example, the "interrupted" state is assumed if, in the "stopped" state (DB21, ... DBX35.2 == 1), the system changes from the AUTOMATIC or MDI mode into the JOG mode. After a subsequent operating mode change from JOG to AUTOMATIC or MDI, the NC program can be executed again from the interruption point using an NC start (DB21, ... DBX7.1 = 1).
	Program states: • DB21, ... DBX35.4 (program state "aborted") • DB21, ... DBX35.3 (program state "interrupted") • DB21, ... DBX35.2 (program state "stopped") • DB21, ... DBX35.1 (program state "wait") • DB21, ... DBX35.0 (program state "running")
Corresponds with	DB21, ... DBX35.4 (program state "aborted") DB21, ... DBX35.2 (program state "stopped") DB21, ... DBX35.1 (program state "wait") DB21, ... DBX35.0 (program state "running")
Additional information	Function Manual Basic Functions

18.2.5.95 DB21, ... DBX35.4 (program state "aborted")

DB21, ... DBX35.4	Program state "aborted"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Program state == "aborted"
Signal state 0	Program state ≠ "aborted"
Additional information	The signal is set in the following cases: • The actual NC program was selected in the channel, however it was not started. • The actual NC program was executed in the channel, and aborted using DB21, ... DBX7.7 = 1 (channel reset).
	Program states: • DB21, ... DBX35.4 (program state "aborted") • DB21, ... DBX35.3 (program state "interrupted") • DB21, ... DBX35.2 (program state "stopped") • DB21, ... DBX35.1 (program state "wait") • DB21, ... DBX35.0 (program state "running")
Corresponds with	DB21, ... DBX7.7 (reset) DB21, ... DBX35.3 (program state "interrupted") DB21, ... DBX35.2 (program state "stopped") DB21, ... DBX35.1 (program state "wait") DB21, ... DBX35.0 (program state "running")
Additional information	Function Manual Basic Functions

18.2.5.96 DB21, ... DBX35.5 (channel state "active")

DB21, ... DBX35.5	Channel state "active"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Channel state == "active"
Signal state 0	Channel state ≠ "active"
Additional information	The signal is set in the following cases: • An NC program is being executed in the Automatic or MDI mode • At least one axis is being traversed in the JOG mode.
	Channel states: • DB21, ... DBX35.5 (channel state "active") • DB21, ... DBX35.6 (channel state "interrupted") • DB21, ... DBX35.7 (channel state "reset")
Corresponds with	DB21, ... DBX35.6 (channel state "interrupted") DB21, ... DBX35.7 (channel state "reset")
Additional information	Function Manual Basic Functions

18.2.5.97 DB21, ... DBX35.6 (channel state "interrupted")

DB21, ... DBX35.6	Channel state "interrupted"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Channel state == "interrupted"
Signal state 0	Channel state ≠ "interrupted"
Additional information	An interruption can be initiated when executing an NC program in the AUTOMATIC or MDA mode - or in the JOG mode while an axis is traversing as a result of the following events: • DB21, ... DBX7.3 (NC Stop) • DB21, ... DBX7.4 (NC Stop axes plus spindles) • DB21, ... DBX7.2 (NC Stop at the block limit) • Programmed stop `M00` or `M01` • Single-block mode
	Channel states: • DB21, ... DBX35.5 (channel state "active") • DB21, ... DBX35.6 (channel state "interrupted") • DB21, ... DBX35.7 (channel state "reset")
Corresponds with	DB21, ... DBX7.3 (NC Stop) DB21, ... DBX7.4 (NC Stop axes plus spindles) DB21, ... DBX7.2 (NC Stop at the block limit) DB21, ... DBX35.5 (channel state "active") DB21, ... DBX35.7 (channel state "reset")
Additional information	Function Manual Basic Functions

18.2.5.98 DB21, ... DBX35.7 (channel state "reset")

DB21, ... DBX35.7	Channel state "reset"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Channel state == "reset"
Signal state 0	Channel state ≠ "reset"
Additional information	The signal is set in the following cases: • end of program reset (M02 / M30) • Channel reset • Warm restart (power on)
	Channel states: • DB21, ... DBX35.5 (channel state "active") • DB21, ... DBX35.6 (channel state "interrupted") • DB21, ... DBX35.7 (channel state "reset")
Corresponds with	DB21, ... DBX35.5 (channel state "active") DB21, ... DBX35.6 (channel state "interrupted")
Additional information	Function Manual Basic Functions

18.2.5.99 DB21, ... DBX36.2 (all axes that have to be referenced have been referenced)

DB21, ... DBX36.2	All axes that have to be referenced are referenced
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	All axes that must be referenced (linear axes and rotary axes) of the channel are referenced. The machine data: MD20700 $MC_REFP_NC_START_LOCK (NC start inhibit without reference point) is zero. If two position measuring systems are connected to an axis, that would prevent an NC start, then the active one must be referenced so that the axis is considered to have been referenced. An NC Start command for the part program processing is only accepted when this signal is present. Axes that have to be referenced are axes, if: MD34110 $MA_REFP_CYCLE_NR _ = -1 and the axis is not in the parked position (position measuring system inactive and the controller enable withdrawn).
Signal state 0	One or more axes of the channel that have to be referenced are not referenced.
Additional information	The spindles of the channel have no effect on this interface signal.
Corresponds with	DB31, ... DBX60.4 (referenced / synchronized 1) DB31, ... DBX60.5 (referenced / synchronized 2)
Additional information	Function Manual Axes and Spindles

18.2.5.100 DB21, ... DBX36.3 (all axes stationary)

DB21, ... DBX36.3	All axes stationary
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	All channel axes are stationary - and no other traversing motion is active.
Signal state 0	Not all channel axes are stationary - or other traversing motion is presently active.
Additional information	Criterion for "axis stationary": Interpolator end
Additional information	Function Manual Basic Functions

18.2.5.101 DB21, ... DBX36.4 (interrupt handling active)

DB21, ... DBX36.4	Interrupt handling active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Interrupt handling is active. One or more channels in the mode group are **not** in the desired operating mode as the result of an active interrupt routine.
Signal state 0	Interrupt handling is **not** active. All channels are operating in the requested mode.
Additional information	**Note** The interface signal is **not**set, if an interrupt is handled in a program mode (AUTOMATIC or MDI).
Corresponds with	MD11600 $MN_BAG_MASK (definition of the mode group response)
Additional information	Function Manual Basic Functions

18.2.5.102 DB21, ... DBX36.5 (channel ready)

DB21, ... DBX36.5	Channel is ready
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The channel is ready
Signal state 0	The channel is **not** ready.
Additional information	The channel is ready to execute an NC program and to traverse axes and spindle that are assigned to the channel.
Corresponds with	MD11600 $MN_BAG_MASK (definition of the mode group response)
Additional information	Function Manual Basic Functions

18.2.5.103 DB21, ... DBX36.6 (channel-specific NC alarm active)

DB21, ... DBX36.6	Channel-specific NC alarm active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	At least one NC alarm is active in the channel.
Signal state 0	No NC alarm is active in the channel.
Corresponds with	DB21, ... DBX36.7 (NC alarm with machining stop active) DB10 DBX109.0 (NC alarm active)
Additional information	Diagnostics guide

18.2.5.104 DB21, ... DBX36.7 (NC alarm with machining stop active)

DB21, ... DBX36.7	NC alarm with machining stop is active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	At least one NC alarm with machining stop is active in the channel.
Signal state 0	No NC alarm with machining stop is active in the channel.
Corresponds with	DB21, ... DBX36.6 (channel-specific NC alarm active) DB10 DBX109.0 (NC alarm active)
Additional information	Diagnostics guide

18.2.5.105 DB21, ... DBX37.0 - 2 (contour handwheel active)

DB21, ... DBX37.0 - 2	Contour handwheel active
Signal flow	PLC → NC
Update	Cyclic

<table>
<tr><td>DB21, ... DBX37.0 - 2</td><td colspan="5">Contour handwheel active</td></tr>
<tr><td>Additional information</td><td colspan="5">The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.</td></tr>
<tr><td></td><td colspan="5">Bit coded: maximum 3 handwheels</td></tr>
<tr><td></td><td></td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td><td>Number of the assigned handwheel</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td></td><td></td><td>1</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td></td><td colspan="5">Binary coded: maximum 6 handwheels</td></tr>
<tr><td></td><td></td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td><td>Number of the assigned handwheel</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td></td><td></td><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td></td><td></td><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td></td><td></td><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td></td><td colspan="5">Note
At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, then the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Corresponds with</td><td colspan="5">DB21 ... DBX30.0 - 2 (activate contour handwheel)
MD11324 $MN_HANDWH_VDI_REPRESENTATION (representation of the handwheel number at the VDI interface)</td></tr>
<tr><td>Additional information</td><td colspan="5">Function Manual Axes and Spindles</td></tr>
</table>

18.2.5.106 DB21, ... DBX37.3 (clearance control (CLC): Active)

DB21, ... DBX37.3	**Clearance control (CLC):**
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Clearance control (CLC) is active.
Signal state 0	Clearance control (CLC) is **not** active.
Corresponds with	DB21, ... DBX1.4 (clearance control (CLC): stop) DB21, ... DBX1.5 (clearance control (CLC): Override) DB21, ... DBX37.4 (clearance control (CLC): Stop at the lower motion limit) DB21, ... DBX37.5 (clearance control (CLC): Stop at the upper motion limit) DB21, ... DBX37.4 - 5 (clearance control (CLC): Motion has stopped)
Additional information	Function Manual Technologies

18.2.5.107 DB21, ... DBX37.4 (clearance control (CLC): Stop at the lower motion limit)

DB21, ... DBX37.4	Clearance control (CLC): Stop at the lower motion limit
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The traversing motion of the clearance-controlled axis, generated by the clearance control, has reached the lower motion limit and was stopped.
Signal state 0	The traversing motion of the clearance-controlled axis, generated by the clearance control, has **not** reached the lower motion limit.
Additional information	The lower motion limit of the clearance control is set in machine data: MD62505 $MC_CLC_SENSOR_LOWER_LIMIT The parameterized limit value can be adapted on a block-specific basis by programming `CLC_LIM(...)` in the part program. **Note** If, in addition to DB21, ... DBX37.4, DB21, ... DBX37.5 are also simultaneously set, then signal "DB21, ... DBX37.3 (clearance control (CLC): Active) (Page 553)"is active.
Corresponds with	`CLC_LIM(...)` ; the control range is limited DB21, ... DBX37.5 (clearance control (CLC): Stop at the upper motion limit) DB21, ... DBX37.4 - 5 (clearance control (CLC): Motion has stopped) MD62505 $MC_CLC_SENSOR_LOWER_LIMIT (lower clearance control motion limit)
Additional information	Function Manual Technologies

18.2.5.108 DB21, ... DBX37.5 (clearance control (CLC): Stop at the upper motion limit)

DB21, ... DBX37.5	Clearance control (CLC): Stop at the upper motion limit
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The traversing motion of the clearance-controlled axis, generated by the clearance control, has reached the upper motion limit and was stopped.
Signal state 0	The traversing motion of the clearance-controlled axis, generated by the clearance control, has **not** reached the upper motion limit.
Additional information	The upper motion limit of the clearance control is set in machine data: MD62506 $MC_CLC_SENSOR_UPPER_LIMIT The parameterized limit value can be adapted on a block-specific basis by programming `CLC_LIM(...)` in the part program. **Note** If, in addition to DB21, ... DBX37.5, DB21, ... DBX37.4 are also simultaneously set, then signal "DB21, ... DBX37.3 (clearance control (CLC): Active) (Page 553)"is active.
Corresponds with	`CLC_LIM(...)` ; the control range is limited DB21, ... DBX37.4 (clearance control (CLC): Stop at the lower motion limit) DB21, ... DBX37.4 - 5 (clearance control (CLC): Motion has stopped) MD62506 $MC_CLC_SENSOR_UPPER_LIMIT (upper clearance control motion limit)
Additional information	Function Manual Technologies

18.2.5.109 DB21, ... DBX37.6 (read-in disable is ignored)

DB21, ... DBX37.6	Read-in disable is ignored
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Read-in disable (DB21, ... DBX6.1) is ignored.
Signal state 0	Read-in disable (DB21, ... DBX6.1) is **not** ignored.
Additional information	Read-in disable (DB21, ... DBX6.1) is ignored, if the following applies: DB21, ... DBX6.1 == 1 (read-in disable) **AND** actual block has the status "read-in disable **inactive**"
	Read-in disable (DB21, ... DBX6.1) is **not** ignored, if the following applies: DB21, ... DBX6.1 == 0 (read-in disable) **OR** (DB21, ... DBX6.1 == 1 (read-in disable) **AND** actual block has the status "read-in disable **active**")
	Blocks for which read-in disable is ignored have the status "read-in disable inactive". The following machine data is used to specify that the read-in disable (DB21, ... DBX6.1) is to be ignored: • MD11602 $MN_ASUP_START_MASK, bit 2 • MD20116 $MC_IGNORE_INHIBIT_ASUP • MD20107 $MC_ PROG_EVENT_IGN_INHIBIT
Corresponds with	DB21, ... DBX37.7 (stop at block end is ignored during single block (SBL)) MD11602 $MN_ASUP_START_MASK, bit 2 (start also permitted if read-in disable is active) MD20116 $MC_IGNORE_INHIBIT_ASUP (execute interrupt program in spite of read-in disable) MD20107 $MC_ PROG_EVENT_IGN_INHIBIT (prog events ignore read-in disable)
Additional information	Function Manual Basic Functions

18.2.5.110 DB21, ... DBX37.7 (stop at block end is ignored during single block (SBL))

DB21, ... DBX37.7	Stop at block end is ignored during single block (SBL)
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Stop at block end is ignored during single block (SBL).
Signal state 0	Stop at block end is **not** ignored during single block (SBL).

DB21, ... DBX37.7	Stop at block end is ignored during single block (SBL)
Additional information	Stop at block end is ignored during single block (SBL), if the following applies: DB21, ... DBX0.4 == 1 (single block) **AND** actual block has the status "single block **inactive**"
	Stop at block end is **not** ignored during single block (SBL), if the following applies: DB21, ... DBB0.4 == 0 (single block) **OR** (DB21, ... DBB0.4 == 1 (single block) **AND** actual block has the status "single block active")
	The following machine data and commands are used to specify that the stop at block end during single block (DB21, ... DBX0.4) is to be ignored: • MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (prevent single-block stop) • MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP (execute interrupt program completely in spite of single block) • MD20106 $MC_PROG_EVENT_IGN_SINGLEBLOCK (prog events ignore single block) • `SBLOF` (suppress single block), `SBLON` (neutral single block suppression) Blocks for which stop at block end during single block is ignored are designated as "single block inactive".
Corresponds with	DB21, ... DBX0.4 (single block) DB21, ... DBX37.6 (read-in disable is ignored)
Additional information	Function Manual Basic Functions

18.2.5.111 DB21, ... DBX38.0 (stroke initiation active)

DB21, ... DBX38.0	Stroke initiation active
Signal flow	NC → PLC
Signal state 1	Stroke initiation is active.
Signal state 0	Stroke initiation is **not** active.
Corresponds with	DB21, ... DBX3.0 (stroke enable)
Additional information	Function Manual Technologies

18.2.5.112 DB21, ... DBX38.1 (manual stroke initiation: Acknowledgment)

DB21, ... DBX38.1	Manual stroke initiation: Acknowledgment
Signal flow	NC → PLC
Signal state 1	Manual stroke has been initiated.
Signal state 0	Manual stroke was **not** initiated.
Corresponds with	DB21, ... DBX3.0 (stroke enable) DB21, ... DBX3.1 (manual stroke initiation) DB21, ... DBX3.5 (manual stroke initiation 2) DB21, ... DBX38.0 (stroke initiation active)
Additional information	Function Manual Technologies

18.2.5.113 DB21, ... DBX39.1 (NC alarm with program stop)

DB21, ... DBX39.1	NC alarm with program stop
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The interface signal indicates when the program progress is blocked by the presence of an alarm. This relates to all alarms generated by stop response: • Stop on the path with deletion of the ready signal • Axes are braked • Immediate stop on the path • Interpreter is stopped, IPO buffer is still being executed. • No stop at end of block Alarms with the following alarm response are also taken into account: • NC start disable in this channel • NC start disable in this channel, ASUB starts, where relevant, are permitted For alarms with this response the interface signal is not set until an NC start has been triggered and only if the alarm has not yet been acknowledged. In this case, the interface signal is not set until NC start. The interface signals are set if the start of an ASUB is refused because of an NC start disable in this channel. The interface signal is deleted as soon as alarm responses that activated the signal are no longer active. This depends on the delete conditions of the alarms.
Signal state 0	No NC alarm with program stop is present in the channel.
Corresponds with	DB10 DBX109.0 (NC alarm present) DB21, ... DBX36.6 (channel-specific NC alarm present) DB21, ... DBX36.7 (NC alarm with machining stop present)
Additional information	Diagnostics Manual

18.2.5.114 DB21, ... DBX39.4 (stop at block end due to single block)

DB21, ... DBX39.4	Stop at end of block due to single block
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Stop at end of block due to single block is active.
Signal state 0	Stop at end of block due to single block is not active.

18.2.5.115 DB21, ... DBX39.5 (Contour handwheel: handwheel direction of rotation inversion active)

DB21, ... DBX39.5	Contour handwheel: handwheel direction of rotation inversion active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Inversion of the direction of rotation of the contour handwheel is active.

DB21, ... DBX39.5	Contour handwheel: handwheel direction of rotation inversion active
Signal state 0	Inversion of the direction of rotation of the contour handwheel is **not** active.
Corresponds with	DB31, ... DBX31.5 (contour handwheel: invert handwheel direction of rotation)
Additional information	Function Manual Axes and Spindles

18.2.5.116 DB21, ... DBX39.6 (configured stop is activated)

DB21, ... DBX39.6	Configured stop is activated
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The "Configured stop" function was activated internally.
Signal state 0	The "Configured stop" function was not activated internally.
Corresponds with	DB21, … DBX7.6 (activate configured stop)

18.2.5.117 DB21, ... DBX39.7 (stop at block end due to configured stop)

DB21, ... DBX39.7	Stop at end of block due to configured stop
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Stop at end of block due to configured stop
Signal state 0	Stop at end of block due to configured stop
Corresponds with	DB21, … DBX7.6 (activate configured stop) DB21, ... DBX39.6 (configured stop is activated)

18.2.5.118 DB21, ... DBX40.0 - 2 (geometry axis 1: handwheel active)

DB21, ... DBX40.0 - 2	Geometry axis 1: handwheel active
Signal flow	NC → PLC
Update	Cyclic

<table>
<tr><th>DB21, ... DBX40.0 - 2</th><th>Geometry axis 1: handwheel active</th></tr>
<tr><td>Additional information</td><td>The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.
Bit coded: maximum 3 handwheels
<table>
<tr><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>3</td></tr>
</table>
Binary coded: maximum 6 handwheels
<table>
<tr><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
</table>
Note
At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, then the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Corresponds with</td><td>DB21, ... DBX46.0 - 2 (geometry axis 2: handwheel active)
DB21, ... DBX52.0 - 2 (geometry axis 3: handwheel active)
DB21, ... DBX12.0 - 2 (geometry axis 1: activate handwheel)
DB21, ... DBX16.0 - 2 (geometry axis 2: activate handwheel)
DB21, ... DBX20.0 - 2 (geometry axis 3: activate handwheel)
MD11324 $MN_HANDWH_VDI_REPRESENTATION (representation of the handwheel number at the VDI interface)</td></tr>
<tr><td>Additional information</td><td>Function Manual Axes and Spindles</td></tr>
</table>

18.2.5.119 DB21, ... DBX40.4 - 5 (geometry axis 1: traversing request "Plus" / "Minus")

DB21, ... DBX40.4 - 5	Geometry axis 1: traversing request "Plus" / "Minus"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	There is a traversing request available for the geometry axis.
Signal state 0	There is **no** traversing request available for the geometry axis.

DB21, ... DBX40.4 - 5	Geometry axis 1: traversing request "Plus" / "Minus"	
Additional information	Depending on the mode selected, the traversing request is initiated in different ways: • JOG mode "Plus" or "Minus" traversing key • REF mode Traversing key that initiates traversing motion in the direction of the reference point. • AUTOMATIC or MDI mode A program block with a traversing operation is executed for a geometry axis. There is a signal for each axis direction:	
	Bit 4	Traversing request "Minus" (for traversing in the negative axis direction)
	Bit 5	Traversing key "Plus" (for traversing in the positive axis direction)
Corresponds with	DB21, ... DBX46.4 - 5 (geometry axis 2: traversing request "Plus" / "Minus") DB21, ... DBX52.4 - 5 (geometry axis 3: traversing request "Plus" / "Minus") DB21, ... DBX12.6 - 7 (geometry axis 1: traversing keys "Plus" / "Minus") DB21, ... DBX16.6 - 7 (geometry axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX20.6 - 7 (geometry axis 3: traversing keys "Plus" / "Minus") DB21, ... DBX40.6 - 7 (geometry axis 1: traversing command "Plus" / "Minus") DB21, ... DBX46.6 - 7 (geometry axis 2: traversing command "Plus" / "Minus") DB21, ... DBX52.6 - 7 (geometry axis 3: traversing command "Plus" / "Minus")	
Additional information	Function Manual Axes and Spindles	

18.2.5.120 DB21, ... DBX40.6 - 7 (geometry axis 1: traversing command "Plus" / "Minus")

DB21, ... DBX40.6 - 7	Geometry axis 1: traversing command "Plus" / "Minus"	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	There is a traversing request for the geometry axis or the geometry axis is traversing.	
Signal state 0	There is **not** a traversing request for the geometry axis or the geometry axis is **not** traversing.	
Additional information	Depending on the setting of MD17900, bit 0, the traversing command is already output if a "Traversing request" is active (bit 0 = 0) or only if the axis is actually traversing (bit 0 = 1). There is a signal for each axis direction:	
	Bit 6	Traversing command "Minus" (for traversing in the negative axis direction)
	Bit 7	Traversing command "Plus" (for traversing in the positive axis direction)
	Application example Releasing the axis clamp when the traversing command is identified. **Note** For axes where clamping is not released until a drive command is detected, the continuous-path mode (G64) is not possible.	

DB21, ... DBX40.6 - 7	Geometry axis 1: traversing command "Plus" / "Minus"
Corresponds with	DB21, ... DBX46.6 - 7 (geometry axis 2: traversing command "Plus" / "Minus") DB21, ... DBX52.6 - 7 (geometry axis 3: traversing command "Plus" / "Minus") DB21, ... DBX12.6 - 7 (geometry axis 1: traversing keys "Plus" / "Minus") DB21, ... DBX16.6 - 7 (geometry axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX20.6 - 7 (geometry axis 3: traversing keys "Plus" / "Minus") DB21, ... DBX40.4 - 5 (geometry axis 1: traversing request "Plus" / "Minus") DB21, ... DBX46.4 - 5 (geometry axis 2: traversing request "Plus" / "Minus") DB21, ... DBX52.4 - 5 (geometry axis 3: traversing request "Plus" / "Minus") MD17900 $MN_VDI_FUNCTION_MASK (setting for VDI signals)
Additional information	Function Manual Axes and Spindles

18.2.5.121 DB21, ... DBX41.0 - 6 (geometry axis 1: active machine function)

DB21, ... DBX41.0 - 6	Geometry axis 1: active machine function	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	The machine function to manually traverse the geometry axis is active.	
Signal state 0	The machine function to manually traverse the geometry axis is **not** active.	
Additional information	There is a signal for every machine function to manually traverse the geometry axis in the JOG mode.	
	Bit 0	INC1
	Bit 1	INC10
	Bit 2	INC100
	Bit 3	INC1000
	Bit 4	INC10000
	Bit 5	INCvar
	Bit 6	Continuous manual travel
	Note Depending on the machine function, the response when actuating the traversing key or the handwheel differs.	
Corresponds with	DB21, ... DBX47.0 - 6 (geometry axis 2: active machine function) DB21, ... DBX53.0 - 6 (geometry axis 3: active machine function) DB21, ... DBX13.0 - 6 (geometry axis 2: machine function request) DB21, ... DBX17.0 - 6 (geometry axis 2: machine function request) DB21, ... DBX21.0 - 6 (geometry axis 3: machine function request)	
Additional information	Function Manual Axes and Spindles	

18.2.5.122 DB21, ... DBX43.0 (geometry axis 1: handwheel direction of rotation inversion active)

DB21, ... DBX43.0	Geometry axis 1: handwheel direction of rotation inversion active
Signal flow	NC → PLC
Update	Cyclic

DB21, ... DBX43.0	Geometry axis 1: handwheel direction of rotation inversion active
Signal state 1	Inversion of the direction of rotation of the handwheel assigned to the geometry axis is active.
Signal state 0	Inversion of the direction of rotation of the handwheel assigned to the geometry axis is **not** active.
Corresponds with	DB21, ... DBX49.0 (geometry axis 2: handwheel direction of rotation inversion active) DB21, ... DBX55.0 (geometry axis 3: handwheel direction of rotation inversion active) DB21, ... DBX15.0 (geometry axis 1: invert handwheel direction of rotation) DB21, ... DBX19.0 (geometry axis 2: invert handwheel direction of rotation) DB21, ... DBX23.0 (geometry axis 3: invert handwheel direction of rotation)
Additional information	Function Manual Axes and Spindles

18.2.5.123 DB21, ... DBX46.0 - 2 (geometry axis 2: handwheel active)

DB21, ... DBX46.0 - 2	Geometry axis 2: handwheel active
Additional information	See DB21, ... DBX40.0 - 2 (geometry axis 1: handwheel active) (Page 558).

18.2.5.124 DB21, ... DBX46.4 - 5 (geometry axis 2: traversing request "Plus" / "Minus")

DB21, ... DBX46.4 - 5	Geometry axis 2: traversing request "Plus" / "Minus"
Additional information	See DB21, ... DBX40.4 - 5 (geometry axis 1: traversing request "Plus" / "Minus") (Page 559).

18.2.5.125 DB21, ... DBX46.6 - 7 (geometry axis 2: traversing command "Plus" / "Minus")

DB21, ... DBX46.6 - 7:	Geometry axis 2: traversing command "Plus" / "Minus"
Additional information	See DB21, ... DBX40.6 - 7 (geometry axis 1: traversing command "Plus" / "Minus") (Page 560).

18.2.5.126 DB21, ... DBX47.0 - 6 (geometry axis 2: active machine function)

DB21, ... DBX47.0 - 6	Geometry axis 2: active machine function
Additional information	See DB21, ... DBX41.0 - 6 (geometry axis 1: active machine function) (Page 561).

18.2.5.127 DB21, ... DBX49.0 (geometry axis 2: handwheel direction of rotation inversion active)

DB21, ... DBX49.0	Geometry axis 2: handwheel direction of rotation inversion active
Additional information	See DB21, ... DBX43.0 (geometry axis 1: handwheel direction of rotation inversion active) (Page 561).

18.2.5.128 DB21, ... DBX52.0 - 2 (geometry axis 3: handwheel active)

DB21, ... DBX52.0 - 2	Geometry axis 3: handwheel active
Additional information	See DB21, ... DBX40.0 - 2 (geometry axis 1: handwheel active) (Page 558).

18.2.5.129 DB21, ... DBX52.4 - 5 (geometry axis 3: traversing request "Plus" / "Minus")

DB21, ... DBX52.4 - 5	Geometry axis 3: traversing request "Plus" / "Minus"
Additional information	See DB21, ... DBX40.4 - 5 (geometry axis 1: traversing request "Plus" / "Minus") (Page 559).

18.2.5.130 DB21, ... DBX52.6 - 7 (geometry axis 3: traversing command "Plus" / "Minus")

DB21, ... DBX52.6 - 7	Geometry axis 3: traversing command "Plus" / "Minus"
Additional information	See DB21, ... DBX40.6 - 7 (geometry axis 1: traversing command "Plus" / "Minus") (Page 560).

18.2.5.131 DB21, ... DBX53.0 - 6 (geometry axis 3: active machine function)

DB21, ... DBX53.0 - 6	Geometry axis 3: active machine function
Additional information	See DB21, ... DBX41.0 - 6 (geometry axis 1: active machine function) (Page 561).

18.2.5.132 DB21, ... DBX55.0 (geometry axis 3: handwheel direction of rotation inversion active)

DB21, ... DBX55.0	Geometry axis 3: handwheel direction of rotation inversion active
Additional information	See DB21, ... DBX43.0 (geometry axis 1: handwheel direction of rotation inversion active) (Page 561).

18.2.5.133 DB21, ... DBB58, DBB60 - 65 (M-, S-, T-, D-, H-, F fct. change)

DB21, ... DBB58, DB21, ... DBB60 - 65	M, S, T, D, H, F fct. change
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	M, S, T, or F information was output to the interface with a new value together with the associated change signal at the beginning of an OB1 cycle. In this case, the change signal indicates that the appropriate value is valid.
Signal state 0	The change signals are reset by the PLC basic program at the start of the next OB1 cycle. The value of the respective data is not valid.
Additional information	Function Manual Basic Functions

18.2.5.134 DB21, ... DBX59.0 - 4 (M fct. 1-5 not decoded)

DB21, ... DBX59.0 - 4	M fct. 1-5 not decoded
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	M function is greater than 99 (for extended address = 0) or for extended address > 0, not included in the decoding list. This signal is available - together with the associated M change signal - for one OB1 cycle. Cause: • Incorrect M function programmed • M function not configured in the decoding list of the PLC Remedy e.g.: • PLC sets read-in inhibit • Output of a PLC alarm
Signal state 0	M function less than 99 (for extended address = 0) or for extended address > 0 included in the decoding list.

18.2.5.135 DB21, ... DBB60 - 64, ... DBB66 - 67 (M-, S-, T-, D-, H-, F fct. Additional quick information (quick acknowledgment)

DB21, ... DBB60 - 64, DB21, ... DBB66 - 67	M, S, T, D, H, F fct. Additional quick information (fast acknowledgment)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	M, S, T, D, H or F information has been output to the interface with a new value together with the associated change signal at the beginning of an OB1 cycle. In this case, the additional info "Quick" indicates the quick help function.
Signal state 0	The change signals are reset by the PLC basic program at the start of the next OB1 cycle. The value of the respective data is not valid.

18.2.5.136 DB21, ... DBB68 - 97 (M function 1 - 5 and extended address M function 1 - 5)

DB21, ... DBB68 - 97	M function 1 - 5 and extended address M function 1 - 5
Signal flow	NC → PLC
Update	Job-controlled
Additional information	If an M function change signal (DB21, ... DBB58) is present, then the up to five M functions - that can be programmed in an NC block - together with the number of the M function and the extended address are available here. The M function values remain until they are overwritten by new M functions. The M function values are deleted by the following events: • The PLC powers up (boots). • A new M function is entered.

DB21, ... DBB68 - 97	M function 1 - 5 and extended address M function 1 - 5
Corresponds with	DB21, ... DBX58.0 - 4 (changes signals: M function) MD10715 $MN_M_NO_FCT_CYCLE MD10716 $MN_M_NO_FCT_CYCLE_NAME MD10718 $MN_M_NO_FCT_CYCLE_PAR $C_M (system variable)
Additional information	Function Manual Basic Functions

18.2.5.137 DB21, ... DBB98 - 115 (S function 1 - 3 and extended address S function 1 - 3)

DB21, ... DBB98 - 115	S function 1 - 3 and extended address S function 1 - 3
Signal flow	NC → PLC
Update	Job-controlled
Additional information	If an S function change signal (DB21, ... DBB60) is present, then the up to three S functions - that can be programmed in an NC block - together with the number of the S function and the extended address are available here. The S function values remain until they are overwritten by new S functions. The S function values are deleted by the following events: • The PLC powers up (boots). • A new S function is entered.
Corresponds with	DB21, ... DBX60.0 - 2 (change signals: S function)
Additional information	Function Manual Basic Functions

18.2.5.138 DB21, ... DBB118 (T function 1)

DB21, ... DBW118 or DBD118	T function 1
Signal flow	NC → PLC
Update	Job-controlled
Additional information	If a T function change signal (DB21, ... DBB61) is present, then number of the T functions, programmed in the NC block, is made available here. The T function values remain until they are overwritten by new T functions. The T function values are deleted by the following events: • The PLC powers up (boots). • A new T function is entered.
Corresponds with	MD10717 $MN_T_NO_FCT_CYCLE_NAME (name of the tool change cycle for T function replacement) MD10719 $MN_T_NO_FCT_CYCLE_MODE (parameterization of the T function replacement) MD22220 $MC_AUXFU_T_SYNC_TYPE (output time of the T functions) MD22550 $MC_TOOL_CHANGE_MODE (new tool compensation for M function) $C_T... (system variable)
Additional information	Function Manual Basic Functions

18.2.5.139 DB21, ... DBB129 (D function 1)

DB21, ... DBB129	D function 1
Signal flow	NC → PLC
Update	Job-controlled
Additional information	If a D function change signal (DB21, ... DBB62) is present, then the number of the D functions, programmed in the NC block, is made available. The D function values remain until they are overwritten by new D functions. The D function values are deleted by the following events: • The PLC powers up (boots). • A new D function is entered.
Corresponds with	MD22250 $MC_AUXFU_D_SYNC_TYPE (output time D functions) MD22252 $MC_AUXFU_DL_SYNC_TYPE (output time DL functions)
Additional information	Function Manual Basic Functions

18.2.5.140 DB21, ... DBB140 - 157 (H function 1 - 3 and extended address H function 1 - 3)

DB21, ... DBB140 - 157	H function 1 - 3 and extended address H function 1-3
Signal flow	NC → PLC
Update	Job-controlled
Additional information	If an H function change signal (DB21, ... DBB64) is present, then the up to three H functions - that can be programmed in an NC block - together with the number of the H function and the extended address are available here. The H function values remain until they are overwritten by new H functions. The H function values are deleted by the following events: • The PLC powers up (boots). • A new H function is entered.
Corresponds with	DB21, ... DBX64.0 - 2 (change signals: H function) MD22110 $MC_AUXFU_H_TYPE_INT (type of H-auxiliary function is an integer) MD22230 $MC_AUXFU_H_SYNC_TYPE (output time of the H functions)
Additional information	Function Manual Basic Functions

18.2.5.141 DB21, ... DBB158 - 193 (F function 1 - 6 and extended address F function 1 - 6)

DB21, ... DBB158 - 193	F function 1 - 6 and extended address F function 1 - 6
Signal flow	NC → PLC
Update	Job-controlled

DB21, ... DBB158 - 193	F function 1 - 6 and extended address F function 1 - 6
Additional information	If an F function change signal (DB21, ... DBB65) is present, then the up to six F functions - that can be programmed in an NC block - together with the number of the F function and the extended address are available here. The extended address of the F function is generated from the feedrate type (path feed or axis-specific feed) and the axis names. • Path feedrate 0 • Axis-specific feedrate: Machine axis number of positioning axis 1, 2, 3, ... maximum number of machine axes The F function values remain until they are overwritten by new F functions. The F function values are deleted by the following events: • The PLC powers up (boots). • A new F function is entered.
Corresponds with	MD22240 $MC_AUXFU_F_SYNC_TYPE (output time of F functions)
Additional information	Function Manual Basic Functions

18.2.5.142 DB21, ... DBB194 - 206 (dynamic M functions: M0 - M99)

DB21, ... DBX194.0 - 7 - DBX206.0 - 3	Dynamic M functions M0 - M99
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The associated M function was has been decoded.
Signal state 0	The associated M function has not been decoded or the interface signal has been reset from the PLC basic program as acknowledgment. For a general auxiliary function output, the interface signal is reset by the PLC basic program after OB1 has been completely executed. For a fast auxiliary function output, the interface signal is reset by the PLC basic program in the same OB40 cycle.
Additional information	Function Manual Basic Functions

18.2.5.143 DB21, ... DBB208 - 271 (active G command of group 1 to 60)

DB21, ... DBB208 - 271	Active G command of group 1 to 60
Signal flow	NC → PLC
Update	Cyclic

DB21, ... DBB208 - 271	Active G command of group 1 to 60							
Additional information	The G command displayed in the BCD format or its mnemonic identifier is active in the specified G group.							
	DBB	**Meaning**						
	208	G group 1: Internal number of the active G command						
	209	G group 2: Internal number of the active G command						
	...	...						
	271	G group 64: Internal number of the active G command						
	Application example G group **14** ≙ DBB208 + **14** - 1 = DBB221, Active G commands: DBB221 == 1 ≙ internal number for G command**G90**							
	Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
	DBB221	0	0	0	0	0	0	0	1
	Special case For a value of 0, no G command or mnemonic identifier of the G group is active.							
	Note In contrast to auxiliary functions, G commands are not output to the PLC subject to acknowledgment, i.e. processing of the part program is continued immediately after the G command output.							
Additional information	• Function Manual Basic Functions • A complete list of the G groups, G commands and their internal numbers can be found in: Programming Manual NC Programming							

18.2.5.144 DB21, ... DBX272.0 - 273.1 (machine-related protection zone 1 - 10 preactivated)

DB21, ... DBX272.0 - 273.1	Machine-related protection zone 1 - 10 preactivated
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The machine-related protection zone 1 (...10) is preactivated. Preactivation is realized in the NC program using command `NPROT`.
Signal state 0	The machine-related protection zone 1 (...10) is not preactivated.
Additional information	**Note** Only a preactivated machine-related protection zone can be activated or deactivated using the corresponding interface signal: DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10)
Corresponds with	DB21, ... DBX1.1 (enable protection zones) DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10) DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10) DB21, ... DBX274.0 – 275.1 (channel-specific protection zone 1 - 10 preactivated) DB21, ... DBX276.0 - 277.1 (machine-related protection zone 1 - 10 violated) DB21, ... DBX278.0 – 279.1 (channel-specific protection zone 1 - 10 violated)
Additional information	Function Manual Monitoring and Compensating

18.2.5.145 DB21, ... DBX274.0 – 275.1 (channel-specific protection zone 1 - 10 preactivated)

DB21, ... DBX274.0 - 275.1	Channel-related protection zone 1 - 10 preactivated
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The channel-specific protection zone 1 (...10) is preactivated. Preactivation is realized in the NC program using command `NPROT`.
Signal state 0	The channel-specific protection zone 1 (...10) is not preactivated.
Additional information	**Note** Only a preactivated channel-specific protection zone can be activated or deactivated using the corresponding interface signal: DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10)
Corresponds with	DB21, ... DBX1.1 (enable protection zones) DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10) DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10) DB21, ... DBX272.0 - 273.1 (machine-related protection zone 1 - 10 preactivated) DB21, ... DBX276.0 - 277.1 (machine-related protection zone 1 - 10 violated) DB21, ... DBX278.0 – 279.1 (channel-specific protection zone 1 - 10 violated)
Additional information	Function Manual Monitoring and Compensating

18.2.5.146 DB21, ... DBX276.0 - 277.1 (machine-related protection zone 1 - 10 violated)

DB21, ... DBX276.0 - 277.1	Machine-related protection zone 1 - 10 violated
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The **activated**, machine-related protection zone 1 - 10 **was** violated in the current block or in the current JOG traversing motion. The **preactivated**, machine-related protection zone 1 - 10 **would** be violated in the current block if it would be active.
Signal state 0	The **activated**, machine-related protection zone 1 - 10 **was** not violated in the current block or in the current JOG traversing motion. The **preactivated**, machine-related protection zone 1 - 10 **would** not be violated in the current block if it would be active.
Corresponds with	DB21, ... DBX1.1 (enable protection zones) DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10) DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10) DB21, ... DBX272.0 - 273.1 (machine-related protection zone 1 - 10 preactivated) DB21, ... DBX274.0 – 275.1 (channel-specific protection zone 1 - 10 preactivated) DB21, ... DBX278.0 – 279.1 (channel-specific protection zone 1 - 10 violated)
Additional information	Function Manual Monitoring and Compensating

18.2.5.147 DB21, ... DBX278.0 – 279.1 (channel-specific protection zone 1 - 10 violated)

DB21, ... DBX278.0 - 279.1	Channel-specific protection zone 1 - 10 violated
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The **activated**, channel-specific protection zone 1 - 10 **was**violated in the current block or in the current JOG traversing motion. The **preactivated**, channel-specific protection zone 1 - 10 **would** be violated in the current block if it would be active.
Signal state 0	The **activated**, channel-specific protection zone 1 - 10 **was** not violated in the current block or in the current JOG traversing motion. The **preactivated**, channel-specific protection zone 1 - 10 **would** not be violated in the current block if it would be active.
Corresponds with	DB21, ... DBX1.1 (enable protection zones) DB21, ... DBX8.0 - 9.1 (activate machine-related protection zone 1 - 10) DB21, ... DBX10.0 - 11.1 (activate channel-specific protection zone 1 - 10) DB21, ... DBX272.0 - 273.1 (machine-related protection zone 1 - 10 preactivated) DB21, ... DBX274.0 – 275.1 (channel-specific protection zone 1 - 10 preactivated) DB21, ... DBX276.0 - 277.1 (machine-related protection zone 1 - 10 violated)
Additional information	Function Manual Monitoring and Compensating

18.2.5.148 DB21, ... DBX280.1 (request disabling of synchronized actions)

DB21, ... DBX280.1	Request disabling of synchronized actions
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	Acceptance of the inhibit signals from the interface in the channel of the NC has been requested by the PLC user program.
Signal state 0	After the request to accept the inhibit signals from the interface in the channel, the NC acknowledged the acceptance by resetting the request.
Further information	The signal is irrelevant for DB21, … DBX1.2 == 1 (disable all synchronized actions)
Corresponds to	Inhibit signals: DB21, … DBX300.0 - 307.7 (inhibit synchronized action ID/IDS) DB21, … DBX281.1 (acknowledge synchronized action disabled)
Additional information	Function Manual Synchronized Actions

18.2.5.149 DB21, ... DBX281.1 (acknowledgement of synchronized actions disabled)

DB21, ... DBX281.1	Acknowledgement of synchronized actions disabled
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Acceptance of the inhibit signals from the interface in the channel of the NC has been requested by the PLC user program.
Signal state 0	After the request to accept the inhibit signals from the interface in the channel, the NC acknowledged the acceptance by resetting the request.

DB21, ... DBX281.1	Acknowledgement of synchronized actions disabled
Corresponds to	DB21, … DBX280.1 (disable request for synchronized actions)
Additional information	Function Manual Synchronized Actions

18.2.5.150 DB21, … DBX300.0 - 307.7 (disable synchronized action ID/IDS)

DB21, ... DBX300.0 ... 307.7	Disable synchronized actions ID/IDS
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The synchronized action ID/IDS is inhibited.
Signal state 0	The synchronized action ID/IDS is enabled.
Further information	The signal is irrelevant for DB21, … DBX1.2 == 1 (disable all synchronized actions)
Corresponds to	DB21, … DBX280.1 (disable request for synchronized actions)
Additional information	Function Manual Synchronized Actions

18.2.5.151 DB21, ... DBX308.0 - 315.7 (synchronized actions ID/IDS can be disabled)

DB21, ... DBX308.0 - 315.7	Synchronized action ID/IDS can be inhibited
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The synchronized action ID/IDS associated with the inhibit signal can be inhibited.
Signal state 0	The synchronized action ID/IDS associated with the inhibit signal cannot be inhibited.
Corresponds to	DB21, … DBX280.1 (disable request for synchronized actions)
Additional information	Function Manual Synchronized Actions

18.2.5.152 DB21, ... DBB317.1 (workpiece setpoint reached)

DB21, ... DBX317.1	Target number of workpieces reached
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The number of machined workpieces (actual workpiece total) is **equal** to the number of workpieces to be machined (target [required] number of workpieces): $AC_ACTUAL_PARTS == $AC_REQUIRED_PARTS
Signal state 0	The number of machined workpieces (actual workpiece total) is **not equal** to the number of workpieces to be machined (target [required] number of workpieces): $AC_ACTUAL_PARTS <> $AC_REQUIRED_PARTS
Corresponds with	MD27880 $MC_PART_COUNTER (activation of workpiece counters)
Additional information	Function Manual Basic Functions

18.2.5.153 DB21, ... DBX317.6 (PTP traversing active)

DB21, ... DBX317.6	PTP traversing active
Signal flow	NC → PLC
Edge change 0 → 1	PTP (Point-To-Point) traversing is active.
Edge change 1 → 0	CP (Continuous Path) traversing is active.
Additional information	**Note** DB21, ... DBX317.6 is only relevant in the JOG mode when the transformation is active.
Corresponds with	DB21, ... DBX29.4 (activate PTP travel)
Additional information	Function Manual Transformations

18.2.5.154 DB21, ... DBX317.7 (TOOLMAN: Tool missing)

DB21, ... DBX317.7	TOOLMAN: Tool missing
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The programmed tool is missing.
Signal state 0	Irrelevant.
Additional information	A tool is either missing or cannot be used.

18.2.5.155 DB21, ... DBX318.0 (ASUB is stopped)

DB21, ... DBX318.0	ASUB is stopped
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The ASUB is in the "stopped" state. **Note** The ASUB was stopped by the control system before it was completely executed.
Signal state 0	The ASUB is **not** in the "stopped" state. **Note** For the following events, the signal is always set to 0: • NC start • Channel reset
Additional information	The interface signal is only processed in the following states: • AUTOMATIC or MDI mode: • Channel state "interrupted" • Program state"stopped"
Additional information	Function Manual Basic Functions

18.2.5.156 DB21, ... DBX318.1 (block search via program test is active (SERUPRO))

DB21, ... DBX318.1	Block search via program test is active (SERUPRO)
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Block search via program test is active During processing of the blocks as part of the block search (internal channel state: "Program test"), the interface signal is set up until the target block is changed to in the main run (program state: "stopped") .
Signal state 0	Block search via program test is **not** active When the target block is changed to in the main run (internal channel state "Program test" is deselected; stop condition: "Search target found" is displayed), the interface signal is reset.
Additional information	The block search (SERUPRO) can only be activated in AUTOMATIC mode in program state "Aborted".
Additional information	Function Manual Basic Functions

18.2.5.157 DB21, ... DBX318.2 (online tool length compensation (TOFF) active)

DB21, ... DBX318.2	Online tool length compensation (TOFF) active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Online tool length compensation (TOFF) is active.
Signal state 0	Online tool length compensation (TOFF) is **not** active.
Additional information	Using the online tool length correction, effective tool lengths can be changed in real time so that the length changes are also considered for changes in orientation of the tool. Application areas: • Orientation transformations (TRAORI) • Orientable toolholder (TCARR)
Corresponds with	DB21, ... DBX318.3 (online tool length compensation (TOFF): Compensation motion active)
Additional information	Function Manual Transformations

18.2.5.158 DB21, ... DBX318.3 (online tool length compensation (TOFF): Compensation motion active)

DB21, ... DBX318.3	Online tool length compensation (TOFF): Compensation motion active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Compensation motion active.
Signal state 0	Compensation motion is **not** active.
Additional information	If, for active online tool length compensation (DB21, ... DBX318.2 == 1), compensation motion is active, then the interface signal is set to "1": DB21, ... DBX318.3 == 1

DB21, ... DBX318.3	Online tool length compensation (TOFF): Compensation motion active
Corresponds with	DB21, ... DBX318.2 (online tool length compensation (TOFF) active)
Additional information	Function Manual Transformations

18.2.5.159 DB21, ... DBX318.5 (associated M0 / M1 active)

DB21, ... DBX318.5	Associated auxiliary function is active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	"Associated auxiliary function" is active.
Signal state 0	Associated auxiliary function is **not** active.
Additional information	"Associated auxiliary function" (M-1) is selected from the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal DB21, ... DBX24.4. The interface signal is transferred according to the value of the FB1 parameter `MMCToIf` by the basic PLC program into the NC/PLC interface signal DB21, ... DBX30.5: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Corresponds with	DB21, ... DBX24.4 (associated auxiliary function selected) DB21, ... DBX30.5 (activate associated auxiliary function) MD22254 $MC_AUXFU_ASSOC_M0_VALUE (additional M function for program stop) MD22256 $MC_AUXFU_ASSOC_M1_VALUE (additional M function for conditional stop)
Additional information	Function Manual Basic Functions

18.2.5.160 DB21, ... DBX319.0 (REPOS mode change acknowledgment)

DB21, ... DBX319.0	REPOS mode change acknowledgment
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The acknowledgment of the REPOS mode change is available. The interface signal DB21, ... DBX31.4 (REPOS mode change) identified by the NC is acknowledged if the requested REPOS mode DB21, ... DBX31.0-2 and delay signal DB31, ... DBX10.0 were accepted in the NC. The signal states refer to the actual main run block
Signal state 0	The acknowledgment of the REPOS mode change is **not** available. The SERUPRO-ASUB stops automatically before REPOS and DB21, ... DBX31.4 (REPOS mode change) does not affect the SERUPRO approach.

DB21, ... DBX319.0	REPOS mode change acknowledgment
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.0 - 2 (REPOS mode) DB21, ... DBX31.4 (REPOS mode change) DB31, ... DBX10.0 (REPOS delay) DB31, ... DBX72.0 (REPOS delay) DB21, ... DBX319.1 - 3 (active REPOS mode) DB21, ... DBX319.5 (REPOS acknowledgment delay) DB31, ... DBX70.0 (REPOS offset) DB31, ... DBX70.1 (REPOS offset valid) DB31, ... DBX70.2 (REPOS delay acknowledgment) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)
Additional information	Function Manual Basic Functions

18.2.5.161 DB21, ... DBX319.1 - 3 (active REPOS mode)

DB21, ... DBX319.1 - 3	Active REPOS mode			
Signal flow	NC → PLC			
Update	Cyclic			
Additional information	Active REPOS mode:			
	Bit 3	Bit 2	Bit 1	Active REPOS approach mode
	0	0	0	No REPOS approach mode active
	0	0	1	Reposition to start of block RMBBL
	0	1	0	Reposition to interrupt point RMIBL
	0	1	1	Reposition to end of block RMEBL
	1	0	0	Reposition to nearest path point RMNBL
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.0 - 2 (REPOS mode) DB21, ... DBX31.4 (REPOS mode change) DB31, ... DBX10.0 (REPOS delay) DB31, ... DBX72.0 (REPOS delay) DB21, ... DBX319.0 (REPOS mode change acknowledgment) DB21, ... DBX319.5 (REPOS acknowledgment delay) DB31, ... DBX70.0 (REPOS offset) DB31, ... DBX70.1 (REPOS offset valid) DB31, ... DBX70.2 (REPOS delay acknowledgment) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)			
Additional information	Function Manual Basic Functions			

18.2.5.162 DB21, ... DBX319.5 (REPOS delay)

DB21, ...DBX319.5	REPOS delay
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	A REPOS delay is active.
Signal state 0	A REPOS delay is **not** active.
Additional information	All axes currently controlled by this channel have either no REPOS offset or their REPOS offsets are not traversed through.
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.0 - 2 (REPOS mode) DB21, ... DBX31.4 (REPOS mode change) DB31, ... DBX10.0 (REPOS delay) DB31, ... DBX72.0 (REPOS delay) DB21, ... DBX319.0 (REPOS mode change acknowledgment) DB21, ... DBX319.1 - 3 (active REPOS mode) DB31, ... DBX70.0 (REPOS offset) DB31, ... DBX70.1 (REPOS offset valid) DB31, ... DBX70.2 (REPOS delay acknowledgment) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)
Additional information	Function Manual Basic Functions

18.2.5.163 DB21, ... DBX320.0 - 2 (orientation axis 1: activate handwheel)

DB21, ... DBX320.0 - 2	Orientation axis 1: Activate handwheel
Signal flow	PLC → NC
Update	Cyclic

DB21, ... DBX320.0 - 2	Orientation axis 1: Activate handwheel
Additional information	The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.
	Bit coded: maximum 3 handwheels

Bit 2	Bit 1	Bit 0	Number of the assigned handwheel
0	0	0	No handwheel assignment
0	0	1	1
0	1	0	2
1	0	0	3

Binary coded: maximum 6 handwheels

Bit 2	Bit 1	Bit 0	Number of the assigned handwheel
0	0	0	No handwheel assignment
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6

Note

At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, then the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".

Corresponds with	DB21, ... DBX324.0 - 2 (orientation axis 2: activate handwheel) DB21, ... DBX328.0 - 2 (orientation axis 3: activate handwheel) DB21, ... DBX332.0 - 2 (orientation axis 1: handwheel active) DB21, ... DBX336.0 - 2 (orientation axis 2: handwheel active) DB21, ... DBX340.0 - 2 (orientation axis 3: handwheel active) MD11324 $MN_HANDWH_VDI_REPRESENTATION (representation of the handwheel number at the VDI interface)
Additional information	Function Manual Axes and Spindles

18.2.5.164 DB21, ... DBX320.4 (orientation axis 1: traversing key lock)

DB21, ... DBX320.4	Orientation axis 1: Traversing key lock
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The traversing key lock for the orientation axis is active.
Signal state 0	The traversing key lock for the orientation axis is **not** active.
Additional information	**Note** When the traversing key disable is activated while traversing, then traversing is canceled.

DB21, ... DBX320.4	Orientation axis 1: Traversing key lock
Corresponds with	DB21, ... DBX324.4 (orientation axis 2: traversing key lock) DB21, ... DBX328.4 (orientation axis 3: traversing key lock) DB21, ... DBX320.6 - 7 (orientation axis 1: traversing keys "Plus" / "Minus") DB21, ... DBX324.6 - 7 (orientation axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX328.6 - 7 (orientation axis 3: traversing keys "Plus" / "Minus")
Additional information	Function Manual Axes and Spindles

18.2.5.165 DB21, ... DBX320.5 (orientation axis 1: rapid traverse override)

DB21, ... DBX320.5	Orientation axis 1: Rapid traverse override
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	When manually traversing the orientation axis using the traversing keys, when the interface signal is set, rapid traverse velocity becomes active (MD32010).
Signal state 0	When manually traversing the orientation axis using the traversing keys, the entered JOG velocity becomes active (SD41110 or MD32020).
Additional information	The signal is only active for continuous or incremental manual traversing in the JOG mode. The rapid traverse velocity can be influenced using the rapid traverse override switch.
Corresponds with	DB21, ... DBX324.5 (orientation axis 2: rapid traverse override) DB21, ... DBX328.5 (orientation axis 3: rapid traverse override) DB21, ... DBX320.6 - 7 (orientation axis 1: traversing keys "Plus" / "Minus") DB21, ... DBX324.6 - 7 (orientation axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX328.6 - 7 (orientation axis 3: traversing keys "Plus" / "Minus") MD32010 $MA_JOG_VELO_RAPID (conventional rapid traverse) MD32020 $MA_JOG_VELO (conventional axis velocity) SD41110 $SN_JOG_SET_VELO (axis velocity for JOG)
Additional information	• Function Manual Axes and Spindles

18.2.5.166 DB21, ... DBX320.6 - 7 (Orientierungsachse 1: traversing keys "Plus" / "Minus")

DB21, ... DBX320.6 - 7	Orientation axis 1: traversing keys "Plus" / "Minus"
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to traverse the orientation axis using the traversing key in the positive/negative axis direction.
Signal state 0	A request has **not** been made to traverse the orientation axis using the traversing key in the positive/negative axis direction.

DB21, ... DBX320.6 - 7	Orientation axis 1: traversing keys "Plus" / "Minus"	
Additional information	There is a request signal for every traversing key and/or axis direction:	
	Bit 6	Traversing key "Minus" (for traversing in the negative axis direction)
	Bit 7	Traversing key "Plus" (for traversing in the positive axis direction)
	Depending on the active machine function, as well as the settings for jog and continuous operation (SD41050 and MD11300), different responses are initiated when the signal changes: 1. Continuous manual travel in the jog mode The orientation axis traverses in the direction concerned to the active limit switch as long as the interface signal is set to 1. 2. Continuous manual travel in continuous operation At the first edge change 0 →1, the orientation axis starts to traverse in the appropriate direction. This traversing movement still continues when the edge changes from 1 → 0. Any new signal edge change 0 → 1 (same traversing direction!) stops the traversing movement. 3. Incremental manual travel in the jog mode With signal 1 the orientation axis starts to traverse the set increment. If the signal changes to the 0 state before the increment is traversed, then traversing is interrupted. When the signal state changes to 1 again, traversing is continued. The orientation axis can be stopped and started several times as described above until it has traversed the complete increment. 4. Incremental manual travel in continuous operation At the first edge change 0 → 1 the orientation axis starts to traverse the set increment. If another edge change 0 → 1 is performed with the same traverse signal before the orientation axis has traversed the increment, the traversing movement will be cancelled. The increment is no longer traversed to the end.	
	Note • The interface signal is irrelevant in operating modes: – AUTOMATIC – MDI • If both traversing signals ("plus" and "minus") are set at the same time there is no traversing or the current traversing is aborted. • In contrast to machine axes, for orientation axes, only one orientation axis can be traversed at any one time using the traversing keys. • Using DB21, ... DBX320.4, 324.4, 328.4 (orientation axis 1, 2, 3: Traversing key lock) traversing using traversing keys can be locked. • In the following cases, the orientation axis cannot be traversed in the JOG operating mode: – The orientation axis is already being traversed via the axis-specific interface as machine axis. – If another orientation axis is already being traversed with the traversing keys.	

DB21, ... DBX320.6 - 7	Orientation axis 1: traversing keys "Plus" / "Minus"
Corresponds with	DB21, ... DBX324.6 - 7 (orientation axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX328.6 - 7 (orientation axis 3: traversing keys "Plus" / "Minus") DB21, ... DBX320.4 (orientation axis 1: traversing key lock) DB21, ... DBX324.4 (orientation axis 2: traversing key lock) DB21, ... DBX328.4 (orientation axis 3: traversing key lock) DB31, ... DBX4.6 - 7 (traversing keys "Plus" / "Minus") MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD (INC and REF in jog mode) SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD (jog/continuous operation for JOG, continuous)
Additional information	Function Manual Axes and Spindles

18.2.5.167 DB21, ... DBX321.0 - 6 (orientation axis 1: machine function request)

DB21, ... DBX321.0 - 6	Orientation axis 1: machine function request	
Signal flow	PLC → NC	
Update	Cyclic	
Signal state 1	A request has been made to manually traverse the orientation axis.	
Signal state 0	A request has **not** been made to manually traverse the orientation axis.	
Additional information	There is a request signal for every machine function to manually traverse the orientation axis in the JOG mode.	
	Bit 0	INC1
	Bit 1	INC10
	Bit 2	INC100
	Bit 3	INC1000
	Bit 4	INC10000
	Bit 5	INCvar
	Bit 6	Continuous manual travel
	Incremental manual travel In addition to five fixed increment sizes (default setting in MD11330: INC1, INC10, INC100, INC1000 and INC10000), a variable increment size (INCvar) that can be set via the setting data SD41010 is also available. The distance evaluation of one increment for fixed and variable increment sizes is performed via the axis-specific machine data MD31090. When pressing the "Plus" or "Minus" traversing key, or by rotating the electronic handwheel, the orientation axis starts to traverse in the appropriate direction, corresponding to the number of increments of the active machine function. **Continuous manual travel** For continuous manual travel, the plus and minus traversing keys are selected to move the orientation axis continuously and the appropriate direction. **Notes** • If several requests are set simultaneously, no machine function becomes active. • If an orientation axis is currently being traversed via a machine function, the movement is aborted through deselection or change of the machine function.	

DB21, ... DBX321.0 - 6	Orientation axis 1: machine function request
Corresponds with	DB21, ... DBX325.0 - 6 (orientation axis 2: machine function request) DB21, ... DBX329.0 - 6 (orientation axis 3: machine function request) DB21, ... DBX333.0 - 6 (orientation axis 1: active machine function) DB21, ... DBX337.0 - 6 (orientation axis 2: active machine function) DB21, ... DBX341.0 - 6 (orientation axis 3: active machine function) MD11320 $MN_HANDWH_IMP_PER_LATCH (handwheel pulses per detent position) MD11330 $MN_JOG_INCR_SIZE_TAB (increment size for INC/handwheel) MD31090 $MA_JOG_INCR_WEIGHT (evaluation of an increment for INC/handwheel) SD41010 $SN_JOG_VAR_INCR_SIZE (size of the variable increment for JOG)
Additional information	Function Manual Axes and Spindles

18.2.5.168 DB21, … DBX323.0 (orientation axis 1: invert handwheel direction of rotation)

DB21, ... DBX323.0	Orientation axis 1: invert handwheel direction of rotation
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to invert the direction of rotation of the assigned handwheel.
Signal state 0	A request has **not** been made to invert the direction of rotation of the assigned handwheel.
Additional information	**Note** It is only permissible to change the interface signal when the orientation axis is at a standstill. **Application examples** • The handwheel direction of rotation should match the axis direction of motion. • A handwheel is assigned to several axes with different orientations.
Corresponds with	DB21, … DBX327.0 (orientation axis 2: invert handwheel direction of rotation) DB21, … DBX331.0 (orientation axis 3: invert handwheel direction of rotation) DB21, ... DBX335.0 (orientation axis 1: handwheel direction of rotation inversion active) DB21, ... DBX339.0 (orientation axis 2: handwheel direction of rotation inversion active) DB21, ... DBX343.0 (orientation axis 3: handwheel direction of rotation inversion active)
Additional information	Function Manual Axes and Spindles

18.2.5.169 DB21, ... DBX324.0 - 2 (orientation axis 2: activate handwheel)

DB21, ... DBX324.0 - 2	Orientation axis 2: Activate handwheel
Additional information	See DB21, ... DBX320.0 - 2 (orientation axis 1: activate handwheel) (Page 576).

18.2.5.170 DB21, ... DBX324.4 (orientation axis 2: traversing key lock)

DB21, ... DBX324.4	Orientation axis 2: Traversing key lock
Additional information	See DB21, ... DBX320.4 (orientation axis 1: traversing key lock) (Page 577).

18.2.5.171 DB21, ... DBX324.5 (orientation axis 2: rapid traverse override)

DB21, ... DBX324.5	Orientation axis 2: Rapid traverse override
Additional information	See DB21, ... DBX320.5 (orientation axis 1: rapid traverse override) (Page 578).

18.2.5.172 DB21, ... DBX324.6 - 7 (orientation axis 2: traversing keys "Plus" / "Minus")

DB21, ... DBX324.6 - 7	Orientation axis 2: traversing keys "Plus" / "Minus"
Additional information	See DB21, ... DBX320.6 - 7 (Orientierungsachse 1: traversing keys "Plus" / "Minus") (Page 578).

18.2.5.173 DB21, ... DBX325.0 - 6 (orientation axis 2: machine function request)

DB21, ... DBX325.0 - 6	Orientation axis 2: machine function request
Additional information	See DB21, ... DBX321.0 - 6 (orientation axis 1: machine function request) (Page 580).

18.2.5.174 DB21, … DBX327.0 (orientation axis 2: invert handwheel direction of rotation)

DB21, ... DBX327.0	Orientation axis 2: invert handwheel direction of rotation
Additional information	See DB21, … DBX323.0 (orientation axis 1: invert handwheel direction of rotation) (Page 581).

18.2.5.175 DB21, ... DBX328.0 - 2 (orientation axis 3: activate handwheel)

DB21, ... DBX328.0 - 2	Orientation axis 3: Activate handwheel
Additional information	See DB21, ... DBX320.0 - 2 (orientation axis 1: activate handwheel) (Page 576).

18.2.5.176 DB21, ... DBX328.4 (orientation axis 3: traversing key lock)

DB21, ... DBX328.4	Orientation axis 3: Traversing key lock
Additional information	See DB21, ... DBX320.4 (orientation axis 1: traversing key lock) (Page 577).

18.2.5.177 DB21, ... DBX328.5 (orientation axis 3: rapid traverse override)

DB21, ... DBX328.5	Orientation axis 3: Rapid traverse override
Additional information	See DB21, ... DBX320.5 (orientation axis 1: rapid traverse override) (Page 578).

18.2.5.178 DB21, ... DBX328.6 - 7 (orientation axis 3: traversing keys "Plus" / "Minus")

DB21, ... DBX328.6 - 7	Orientation axis 3: traversing keys "Plus" / "Minus"
Additional information	See DB21, ... DBX320.6 - 7 (Orientierungsachse 1: traversing keys "Plus" / "Minus") (Page 578).

18.2.5.179 DB21, ... DBX329.0 - 6 (orientation axis 3: machine function request)

DB21, ... DBX329.0 - 6	Orientation axis 3: machine function request
Additional information	See DB21, ... DBX321.0 - 6 (orientation axis 1: machine function request) (Page 580).

18.2.5.180 DB21, … DBX331.0 (orientation axis 3: invert handwheel direction of rotation)

DB21, ... DBX331.0	Orientation axis 3: invert handwheel direction of rotation
Additional information	See DB21, … DBX323.0 (orientation axis 1: invert handwheel direction of rotation) (Page 581).

18.2.5.181 DB21, ... DBX332.0 - 2 (orientation axis 1: handwheel active)

DB21, ... DBX332.0 - 2	Orientation axis 1: handwheel active
Signal flow	NC → PLC
Update	Cyclic

DB21, ... DBX332.0 - 2	Orientation axis 1: handwheel active			
Additional information	The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.			
	Bit coded: **maximum 3** handwheels			
	Bit 2	**Bit 1**	**Bit 0**	**Number of the assigned handwheel**
	0	0	0	No handwheel assignment
	0	0	1	1
	0	1	0	2
	1	0	0	3
	Binary coded: **maximum 6** handwheels			
	Bit 2	**Bit 1**	**Bit 0**	**Number of the assigned handwheel**
	0	0	0	No handwheel assignment
	0	0	1	1
	0	1	0	2
	0	1	1	3
	1	0	0	4
	1	0	1	5
	1	1	0	6
	Note At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, then the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".			
Corresponds with	DB21, ... DBX336.0 - 2 (orientation axis 2: handwheel active) DB21, ... DBX340.0 - 2 (orientation axis 3: handwheel active) DB21, ... DBX320.0 - 2 (orientation axis 2: activate handwheel) DB21, ... DBX324.0 - 2 (orientation axis 2: activate handwheel) DB21, ... DBX328.0 - 2 (orientation axis 3: activate handwheel) MD11324 $MN_HANDWH_VDI_REPRESENTATION (representation of the handwheel number at the VDI interface)			
Additional information	Function Manual Axes and Spindles			

18.2.5.182 DB21, ... DBX332.4 - 5 (orientation axis 1: traversing request "Plus" / "Minus")

DB21, ... DBX332.4 - 5	Orientation axis 1: traversing request "Plus" / "Minus"
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	There is a traversing request available for the orientation axis.
Signal state 0	There is **no** traversing request available for the orientation axis.

DB21, ... DBX332.4 - 5	Orientation axis 1: traversing request "Plus" / "Minus"	
Additional information	The traversing request is initiated in different ways depending on the operating mode: • JOG mode "Plus" or "Minus" traversing key • REF mode Traversing key that initiates traversing motion in the direction of the reference point. • AUTOMATIC or MDI mode A program block with a traversing operation is executed for an orientation axis. There is a signal for each axis direction:	
	Bit 4	Traversing request "Minus" (for traversing in the negative axis direction)
	Bit 5	Traversing key "Plus" (for traversing in the positive axis direction)
Corresponds with	DB21, ... DBX336.4 - 5 (orientation axis 2: traversing request "Plus" / "Minus") DB21, ... DBX340.4 - 5 (orientation axis 3: traversing request "Plus" / "Minus") DB21, ... DBX320.6 - 7 (orientation axis 1: traversing keys "Plus" / "Minus") DB21, ... DBX324.6 - 7 (orientation axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX328.6 - 7 (orientation axis 3: traversing keys "Plus" / "Minus") DB21, ... DBX332.6 - 7 (orientation axis 1: traversing command "Plus" / "Minus") DB21, ... DBX336.6 - 7 (orientation axis 2: traversing command "Plus" / "Minus") DB21, ... DBX340.6 - 7 (orientation axis 3: traversing command "Plus" / "Minus")	
Additional information	Function Manual Axes and Spindles	

18.2.5.183 DB21, ... DBX332.6 - 7 (orientation axis 1: traversing command "Plus" / "Minus"

DB21, ... DBX332.6 - 7	Orientation axis 1: traversing command "Plus" / "Minus"	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	There is a traversing request for the orientation axis or the orientation axis is traversing.	
Signal state 0	There is **no** traversing request for the orientation axis or the orientation axis is **not** traversing.	
Additional information	Depending on the setting of MD17900, bit 0, the traversing command is already output if a "Traversing request" is active (bit 0 = 0) or only if the axis is actually traversing (bit 0 = 1). There is a signal for each axis direction:	
	Bit 6	Traversing command "Minus" (for traversing in the negative axis direction)
	Bit 7	Traversing command "Plus" (for traversing in the positive axis direction)
	Application example Releasing the axis clamp when the traversing command is identified. **Note** For axes where clamping is not released until a drive command is detected, the continuous-path mode (G64) is not possible.	

DB21, ... DBX332.6 - 7	Orientation axis 1: traversing command "Plus" / "Minus"
Corresponds with	DB21, ... DBX336.6 - 7 (orientation axis 2: traversing command "Plus" / "Minus") DB21, ... DBX340.6 - 7 (orientation axis 3: traversing command "Plus" / "Minus") DB21, ... DBX320.6 - 7 (orientation axis 1: traversing keys "Plus" / "Minus") DB21, ... DBX324.6 - 7 (orientation axis 2: traversing keys "Plus" / "Minus") DB21, ... DBX328.6 - 7 (orientation axis 3: traversing keys "Plus" / "Minus") DB21, ... DBX332.4 - 5 (orientation axis 1: traversing request "Plus" / "Minus") DB21, ... DBX336.4 - 5 (orientation axis 2: traversing request "Plus" / "Minus") DB21, ... DBX340.4 - 5 (orientation axis 3: traversing request "Plus" / "Minus") MD17900 $MN_VDI_FUNCTION_MASK (setting for VDI signals)
Additional information	Function Manual Axes and Spindles

18.2.5.184 DB21, ... DBX333.0 - 6 (orientation axis 1: active machine function)

DB21, ... DBX333.0 - 6	Orientation axis 1: active machine function	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	The machine function to manually traverse the orientation axis is active.	
Signal state 0	The machine function to manually traverse the orientation axis is **not** active.	
Additional information	There is a signal for every machine function to manually traverse the orientation axis in the JOG mode.	
	Bit 0	INC1
	Bit 1	INC10
	Bit 2	INC100
	Bit 3	INC1000
	Bit 4	INC10000
	Bit 5	INCvar
	Bit 6	Continuous manual travel
	Note Depending on the machine function, the response when actuating the traversing key or the handwheel differs.	
Corresponds with	DB21, ... DBX337.0 - 6 (orientation axis 2: active machine function) DB21, ... DBX341.0 - 6 (orientation axis 3: active machine function) DB21, ... DBX321.0 - 6 (orientation axis 2: machine function request) DB21, ... DBX325.0 - 6 (orientation axis 2: machine function request) DB21, ... DBX329.0 - 6 (orientation axis 3: machine function request)	
Additional information	Function Manual Axes and Spindles	

18.2.5.185 DB21, ... DBX336.0 - 2 (orientation axis 2: handwheel active)

DB21, ... DBX336.0 - 2	Orientation axis 2: handwheel active
Additional information	See DB21, ... DBX332.0 - 2 (orientation axis 1: handwheel active) (Page 583).

18.2.5.186 DB21, ... DBX336.4 - 5 (orientation axis 2: traversing request "Plus" / "Minus")

DB21, ... DBX336.4 - 5	Orientation axis 2: traversing request "Plus" / "Minus"
Additional information	See DB21, ... DBX332.4 - 5 (orientation axis 1: traversing request "Plus" / "Minus") (Page 584).

18.2.5.187 DB21, ... DBX336.6 - 7 (orientation axis 2: traversing command "Plus" / "Minus"

DB21, ... DBX336.6 - 7	Orientation axis 2: traversing command "Plus" / "Minus"
Additional information	See DB21, ... DBX332.6 - 7 (orientation axis 1: traversing command "Plus" / "Minus" (Page 585).

18.2.5.188 DB21, ... DBX337.0 - 6 (orientation axis 2: active machine function)

DB21, ... DBX337.0 - 6	Orientation axis 2: active machine function
Additional information	See DB21, ... DBX333.0 - 6 (orientation axis 1: active machine function) (Page 586).

18.2.5.189 DB21, ... DBX340.0 - 2 (orientation axis 3: handwheel active)

DB21, ... DBX340.0 - 2:	Orientation axis 3: handwheel active
Additional information	See DB21, ... DBX332.0 - 2 (orientation axis 1: handwheel active) (Page 583).

18.2.5.190 DB21, ... DBX340.4 - 5 (orientation axis 3: traversing request "Plus" / "Minus")

DB21, ... DBX340.4 - 5:	Orientation axis 3: traversing request "Plus" / "Minus"
Additional information	See DB21, ... DBX332.4 - 5 (orientation axis 1: traversing request "Plus" / "Minus") (Page 584).

18.2.5.191 DB21, ... DBX340.6 - 7 (orientation axis 3: traversing command "Plus" / "Minus"

DB21, ... DBX340.6 - 7	Orientation axis 3: traversing command "Plus" / "Minus"
Additional information	See DB21, ... DBX332.6 - 7 (orientation axis 1: traversing command "Plus" / "Minus" (Page 585).

18.2.5.192 DB21, ... DBX341.0 - 6 (orientation axis 3: active machine function)

DB21, ... DBX341.0 - 6	Orientation axis 3: active machine function
Additional information	See DB21, ... DBX333.0 - 6 (orientation axis 1: active machine function) (Page 586).

18.2.5.193 DB21, ... DBX344.0 (TM: tool prewarning limit reached)

DB21, ... DBX344.0	TOOLMAN: Tool prewarning limit reached
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	The tool prewarning limit has been reached.
Signal state 0	Irrelevant.
Additional information	The T number of the tool, whose prewarning value has been reached, is located in: DBD348
	The change signal is active for one OB 1 cycle.
Corresponds with	DB21, ... DBX344.1 tool management change signal: T number for tool limit value DB21, ... DBX344.2 tool management change signal: T number for the new replacement tool DB21, ... DBX344.3 tool management change signal: T number of the last replacement tool DBD348 (T number for tool prewarning limit)
Additional information	Function Manual, Tool Management

18.2.5.194 DB21, ... DBX344.1 (TM: Tool limit value reached)

DB21, ... DBX344.1	TOOLMAN: Tool limit value reached
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	The tool limit value has been reached
Signal state 0	Irrelevant.
Additional information	The T number of the tool, whose limit value has been reached, is located in: DBD352
	The change signal is active for one OB 1 cycle.
Corresponds with	DB21, ... DBX344.0 tool management change signal: T number for tool pre-alarm limit DB21, ... DBX344.2 tool management change signal: T number for the new replacement tool DB21, ... DBX344.3 tool management change signal: T number of the last replacement tool DBD352 (T number for tool limit value)
Additional information	Function Manual, Tool Management

18.2.5.195 DB21, ... DBX344.2 (TM: Transition to new replacement tool)

DB21, ... DBX344.2	TOOLMAN: Transition to new replacement tool
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	A transition is made to a new replacement tool.
Signal state 0	Irrelevant.
Additional information	The T number for the new replacement tool is located in: DBD356
	The change signal is active for one OB 1 cycle.

DB21, ... DBX344.2	TOOLMAN: Transition to new replacement tool
Corresponds with	DB21, ... DBX344.0 tool management change signal: T number for tool pre-alarm limit DB21, ... DBX344.1 tool management change signal: T number for tool limit value DB21, ... DBX344.3 tool management change signal: T number of the last replacement tool DBD356 (T number of the replacement tool)
Additional information	Function Manual, Tool Management

18.2.5.196 DB21, ... DBX344.3 (TM: Last replacement tool of the tool group)

DB21, ... DBX344.3	TOOLMAN: Last replacement tool of the tool group
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	The last replacement tool of the tool group has been reached
Signal state 0	Irrelevant.
Additional information	The T number of the last replacement tool of the tool group is located in: DBD360
	The change signal is active for one OB 1 cycle.
	Note The monitoring for the last replacement tool must be activated in MD20310 $MC_TOOL_MANAGEMENT_MASK, bit 18.
Corresponds with	DB21, ... DBX344.0 tool management change signal: T number for tool pre-alarm limit DB21, ... DBX344.1 tool management change signal: T number for tool limit value DB21, ... DBX344.2 tool management change signal: T number for the new replacement tool DB21, ... DBD360 (T-number of the last replacement tool)
Additional information	Function Manual, Tool Management

18.2.5.197 DB21, ... DBB376 (PROG_EVENT trigger event)

DB21, ...DBB376	PROG_EVENT trigger event		
Signal flow	NC → PLC		
Update	Cyclic		
Additional information	The event that initiated the PROG_EVENT is displayed bit-coded.		
	Bit	**Value**	**Event**
	0	1	NC start from channel state "reset"
	1	1	End of program reset (e.g. M30)
	2	1	Channel reset
	3	1	Warm restart (power on)
	4	1	1. NC start after search
	5	1	Safety
	6 - 7	---	Reserved, currently always 0
	Note As a minimum, the interface signal must remain for a complete PLC cycle.		
Additional information	Function Manual Basic Functions		

18.2.5.198 DB21, ... DBX377.0 (collision avoidance: stop)

DB21, ... DBX377.0	Collision avoidance: Stop
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The collision avoidance has triggered a stop of the traversing motion in the channel.
Signal state 0	The collision avoidance has **not** triggered a stop of the traversing motions in the channel.
Additional information	Function Manual Monitoring and Compensating

18.2.5.199 DB21, ... DBX377.4 (JOG retract active)

DB21, ... DBX377.4	JOG retract active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	JOG retract has been selected and is active.
Signal state 0	JOG retract has not been selected.
Corresponds with	DB21, ... DBX377.5 (JOG retract retraction data available)
Additional information	Function Manual Axes and Spindles

18.2.5.200 DB21, ... DBX377.5 (JOG retract retraction data available)

DB21, ... DBX377.5	JOG retract retraction data available
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Retraction data is available for traversing in the tool direction.
Signal state 0	Retraction data is not available for traversing in the tool direction. JOG retract cannot be selected
Additional information	If the signal state is active, then the JOG retract can be selected (user interface or PI service "RETRAC").
Corresponds with	DB21, ... DBX377.4 (JOG retract active)
Additional information	Function Manual Axes and Spindles

18.2.5.201 DB21, ... DBX377.6 (JOG circular travel active)

DB21, ... DBX377.6	JOG circular travel active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The "Circular travel in JOG" function is active.
Signal state 0	The "Circular travel in JOG" function is**not** active.
Corresponds with	DB21, ... DBX30.6 (JOG circular travel)
Additional information	Function Manual Axes and Spindles

18.2.5.202 DB21, … DBX378.0 (ASUB active)

DB21, ... DBX378.0	ASUB is active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	An ASUB is active.
Signal state 0	**No** ASUB is active.
Additional information	**Note** Also outside block FC9, feedback about a running ASUB is realized using the interface signal.
Additional information	Function Manual Basic Functions

18.2.5.203 DB21, … DBX378.1 (stopped ASUB active)

DB21, ... DBX378.1	"Silent" ASUB is active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	A "silent" ASUB is active.
Signal state 0	**No** "silent" ASUB is active.
Additional information	A "silent" ASUB is an ASUB where blocks are not displayed on the screen as this has been suppressed with `DISPLOF`.
Additional information	Function Manual Basic Functions

18.2.5.204 DB21, ... DBX384.0 (enable GOTOS)

DB21, ... DBX384.0	Enable GOTOS
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to enable `GOTOS` (return jump to the program start) in the channel.
Signal state 0	A request has **not** been made to enable `GOTOS` (return jump to the program start) in the channel. NC program execution is continued with the next command following `GOTOS`.
Corresponds with	MD27850 $MC_PROG_NET_TIMER_MODE (impact of the program runtime net counter) MD27860 $MC_PROCESSTIMER_MODE (activation of the program runtime measurement) MD27880 $MC_PART_COUNTER (activation of the workpiece counters)
Additional information	Function Manual Basic Functions

18.2.5.205 DB21, ... DBX385.0 - 7 (input signal 1 - 8)

DB21, ... DBX385.0 - 7	**Input signal 1 - 8**
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The value of the input signal is 1 (TRUE).
Signal state 0	The value of the input signal is 0 (FALSE).
Further information	Application: Grinding
Corresponds to	$AC_IN_KEY_G_ENABLE[1 ... 8] (input signal 1 - 8 enable on the NC side)
Additional information	Function Manual Synchronized Actions

18.2.5.206 DB21, ... DBX386.0 - 7 (disabling of input signal 1 - 8)

DB21, ... DBX386.0 - 7	**Disabling of input signal 1 - 8**
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The input signal is inhibited.
Signal state 0	The input signal is enabled.
Further information	Application: Grinding
Corresponds to	DB21, … DBX385.0 - 7 (input signals 1 - 8) DB21, … DBX387.0 ... 7 (action is enabled) $AC_IN_KEY_G_ENABLE[1 ... 8] (input signal 1 - 8 enable on the NC side)
Additional information	Function Manual Synchronized Actions

18.2.5.207 DB21, ... DBX387.0 - 7 (status of action 1 - 8)

DB21, ... DBX387.0 - 7	**Status of action 1 - 8**
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The action is enabled.
Signal state 0	The action is inhibited.
Further information	Application: Grinding
Corresponds to	DB21, … DBX391.0 - 7 (enable request for the action on the NC side) $AC_IN_KEY_G_RUN_OUT[1 ... 8] (enable request of the action 1 - 8 on the NC side) $AC_IN_KEY_G_RUN_IN[1 ... 8] (enable of the action 1 - 8 on the PLC side)
Additional information	Function Manual Synchronized Actions

18.2.5.208 DB21, ... DBX390.0 - 7 (enable status of input signal 1 - 8)

DB21, ... DBX390.0 - 7	Enable status of input signal 1 - 8
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The input signal is enabled. The enable signal is formed by the NC-side **AND**logic operation of the NC enable signal $AC_IN_KEY_G_ENABLE[1 ... 8] and the PLC enable signal **NOT** (DBX386.0 - 7).
Signal state 0	The input signal is not enabled.
Further information	Application: Grinding
Corresponds to	DB21, … DBX386.0 - 7 (disabling of input signal 1 - 8) $AC_IN_KEY_G_ENABLE[1 ... 8] (input signal 1 - 8 enable on the NC side) $AC_IN_KEY_G_ISENABLE[1 ... 8] (release status of the input signal 1 - 8)
Additional information	Function Manual Synchronized Actions

18.2.5.209 DB21, ... DBX391.0 - 7 (enable request for action 1 - 8)

DB21, ... DBX391.0 - 7	Enable request for action 1 - 8
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	It has been requested that the action is enabled.
Signal state 0	It has not been requested that the action is enabled.
Further information	Application: Grinding
Corresponds to	DB21, … DBX387.0 - 7 (status of the action 1 - 8) $AC_IN_KEY_G_RUN_OUT[1 ... 8] (enable request of the action 1 - 8 on the NC side) $AC_IN_KEY_G_RUN_IN[1 ... 8] (enable of the action 1 - 8 on the PLC side)
Additional information	Function Manual Synchronized Actions

18.2.5.210 DB21, ... DBB392 (selection: Coordinate system for Cartesian manual traversing and handwheel override in automatic mode in the tool direction (DRF))

DB21, ... DBB392	Selection: Coordinate system for Cartesian manual traversing and handwheel override in automatic mode in the tool direction (DRF)
Signal flow	PLC → NC
Update	Cyclic

DB21, ... DBB392	Selection: Coordinate system for Cartesian manual traversing and handwheel override in automatic mode in the tool direction (DRF)	
Additional information	Selection of the coordinate system for: • Cartesian manual traversing of the geometry axes in the tool direction (TCS) • Handwheel override of the geometry axes in the AUTOMATIC mode in the tool direction (DRF)	
	Value	**Description**
	0	The function is not selected
	1	Manual traversing of the geometry axes in the basic coordinate system (**BCS**)
	2	Manual traversing of the geometry axes in the workpiece coordinate system (**WCS**)
	3	Manual traversing or handwheel override (DRF) of the geometry axes in the workpiece coordinate system (**WCS**)
Additional information	• Function Manual Basic Functions • FC19: MCP_IFM - transfer of MCP signals to interface (Page 267) • Function Manual Transformations • Function Manual Axes and Spindles	

18.2.6 DB31, ...: Axis/spindle

18.2.6.1 DB31, ... DBB0 (feedrate override, axis-specific)

DB31, ... DBB0	Feedrate override, axis-specific
Signal flow	PLC → NC
Update	Cyclic

DB31, ... DBB0	Feedrate override, axis-specific
Additional information	**Binary/Gray coding** The override factors can be specified in the binary or Gray-coded format. The control must be informed of the used format via the following machine data: MD12000 $MN_OVR_AX_IS_GRAY_CODE = <Coding>

Binary coding

With binary coding, the value in the interface corresponds to the override factor.

Binary code	Decimal	Override factor
0000 0000	0	0.00
0000 0001	1	0.01
0000 0010	2	0.02
0000 0011	3	0.03
000 0100	4	0.04
...	...	...
0110 0100	100	1.00
...	...	...
1100 1000	200	2.00

Gray coding

The Gray-coded values of the interface are assigned the override factors via the following machine data:

MD12010 $MN_OVR_FACTOR_AX_SPEED[<Switch position> - 1] = <Override factor>

Switch position	Gray code	Override factor [1)]
1	00001	0.00
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.05
21	11111	1.10
22	11101	1.15

DB31, ... DBB0	Feedrate override, axis-specific		
	23	11100	1.20
	24	10100	1.20
	25	10101	1.20
	26	10111	1.20
	27	10110	1.20
	28	10010	1.20
	29	10011	1.20
	30	10001	1.20
	31	10000	1.20
	Limitations • The active override factor is limited internally in the control to 2.00 or 200%. • The maximum possible override factor can be limited to a value less than 200% with the following machine data: MD12100 $MN_OVR_FACTOR_LIMIT_BIN		
	Notes The axis-specific feedrate override is **not effective** with the following functions when cutting threads: G33, G331, G332, G63		
Corresponds to	DB21, ... DBB4 (path feedrate override) DB31, ... DBB19 (speed override, spindle-specific) DB31, ... DBX1.7 (override active)		
Additional information	Function Manual Axes and Spindles		
1) Default values			

18.2.6.2 DB31, ... DBX1.0 (drive test travel enable)

DB31, ... DBX1.0	Drive test travel enable
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The travel enable for the drive test is **set**, i.e. axis motion for the drive test is **enabled**.
Signal state 0	The travel enable for the drive test is **not set**, i.e. axis motion for the drive test is **inhibited**.
Further information	The signal is the feedback signal for the request: DB31, ... DBX61.0 == 1 (drive test travel request) Only the PLC can enable an axis so that it can be traversed.
Corresponds with	DB31, ... DBX61.0 (drive test travel request)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.3 DB31, ... DBX1.1 (acknowledge fixed stop reached).

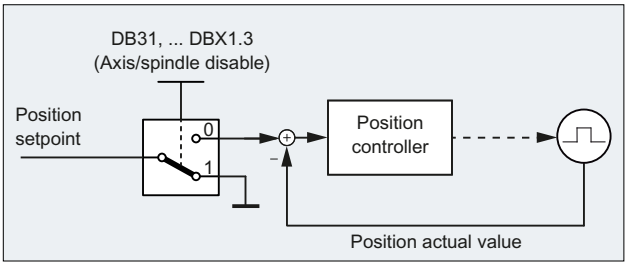
DB31, ... DBX1.1	Acknowledge fixed stop reached
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Significance when the fixed stop is reached: DB31, ... DBX62.5 (fixed stop reached) == 1 • The axis presses against the fixed stop with the clamping torque. • The fixed stop monitoring window is activated. • A block change is executed.
Signal state 0	Significance when the fixed stop is reached: DB31, ... DBX62.5 (fixed stop reached) == 1 • The axis presses against the fixed stop with the clamping torque. • The fixed stop monitoring window is activate • A block change is not executed, and the channel message "Wait: HiFu acknowledgment missing" is displayed • The function is canceled, alarm "20094 axis %1 Function aborted" is output. Significance after deselecting the function `FXS=0` via the part program: • The torque limiting and the monitoring of the fixed stop monitoring window are canceled.
Additional information	Irrelevant for MD37060 $MA_FIXED_STOP_ACKN_MASK == 0 or 2
Corresponds with	DB31, ... DBX1.2 (fixed stop sensor) DB31, ... DBX3.1 (enable travel to fixed stop) DB31, ... DBX62.4 (activate travel to fixed stop) DB31, ... DBX62.5 (fixed stop reached) MD37060 $MA_FIXED_STOP_ACKN_MASK (monitoring PLC acknowledgements for travel to fixed stop)
Additional information	Function Manual Basic Functions

18.2.6.4 DB31, ... DBX1.2 (fixed stop sensor)

DB31, ... DBX1.2	Sensor for fixed stop
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Fixed stop has been reached.
Signal state 0	Fixed stop has not been reached.
Additional information	The signal is only active if: MD37040 $MA_FIXED_STOP_BY_SENSOR == 1
Corresponds with	DB31, ... DBX1.1 (acknowledge fixed stop reached). DB31, ... DBX3.1 (enable travel to fixed stop) DB31, ... DBX62.4 (activate travel to fixed stop) DB31, ... DBX62.5 (fixed stop reached) MD37040 $MA_FIXED_STOP_BY_SENSOR = (fixed stop detection via sensor)
Additional information	Function Manual Axes and Spindles

18.2.6.5 DB31, ... DBX1.3 (axis/spindle disable)

DB31, ... DBX1.3	Axis/spindle disable
Signal flow	PLC → NC
Update	Cyclic

DB31, ... DBX1.3	Axis/spindle disable
Signal state 1	Axis/spindle disable is requested. **Axis** • Effects with stationary axis: – No setpoints are output to the position controller, i.e. the axis traversing movement is disabled. The axis is in closed-loop position control and any remaining following error is compensated. – If the axis is traversed via an NC program or manually, the set position and set velocity are displayed as the actual position and actual velocity on the user interface. With channel reset or end of program (M30/M2), the display of the actual value is set to the actual value of the machine axis. – The travel commands are output to the NC/PLC interface. • Effects with traversing axis: – The axis is stopped in compliance with the currently active braking characteristic and an alarm is displayed. Subsequently, the travel movements of the axis are disabled. Position control remains active and holds the axis in the current position. **Spindle** • Effects with open-loop control mode and stationary spindle: – No setpoints are output to the speed controller, i.e. the spindle traversing movement is disabled. – If the spindle is traversed via an NC program or manually, the speed setpoint is displayed on the user interface as the speed actual value. • Effects with open-loop control mode and rotating spindle: – Speed setpoint zero is immediately output to the speed controller. – The spindle is stopped in compliance with the currently active braking characteristic and an alarm is displayed. Subsequently, the travel movements of the spindle are disabled. Position control remains active and holds the axis in the current position. • Effects with positioning mode: See Stationary axis / Traversing axis **Structure graphic**  Axis/spindle disable
	Note • If one of the following interface signals is set for a traversing axis/spindle for which the signal "Axis/spindle disable" is present, the axis/spindle is **not** stopped: – DB21, ... DBX12.3, 16.3 or 20.3 (feedrate stop for geometry axes 1, 2 and 3) – DB21, ... DBX320.3, 324.3 or 328.3 (feedrate stop for orientation axes 1, 2 and 3)

<table>
<tr><td>DB31, ... DBX1.3</td><td>Axis/spindle disable</td></tr>
<tr><td></td><td>
– DB31, ... DBX4.3 (feedrate stop / spindle stop)
• The axis/spindle can still be switched to the "hold" or "follow up" state with DB31, ... DBX1.4 (follow-up mode).
• An axis cannot be referenced either channel-specifically via DB21, ... DBX1.0 from the NC program (G74) or axis-specifically (REF machine function).
• If the signal "Axis/spindle disable" is set only briefly for a traversing axis, the axis is stopped without an alarm. With the next travel request, the axis is traversed to the new position.
Example:
<code>N10 G0 X0 Y0</code>
<code>N20 G1 F1000 X100</code>
<code>N30 Y100</code>
<code>N40 X200</code>
Regarding N20: At position 20 mm, "Axis/spindle lock" is set briefly for axis X ⇒ axis X is stopped
Regarding N40: Axis X traverses from the last position (approx. 20 mm + braking distance) to position 200 mm.
• Effects of "Axis/spindle disable" for spindle or axis couplings:
<table>
<tr><th>DB31, ... DBX99.0 1)</th><th>DB31, ... DBX99.1 2)</th><th>Coupl. 3)</th><th>Effect</th></tr>
<tr><td>0</td><td>0</td><td>Off</td><td>Setpoints are output</td></tr>
<tr><td>0</td><td>1</td><td>Off</td><td>No setpoint output for following spindle/axis</td></tr>
<tr><td>1</td><td>0</td><td>Off</td><td>No setpoint output for leading spindle/axis</td></tr>
<tr><td>1</td><td>1</td><td>Off</td><td>No setpoint output for leading and following spindle/axis</td></tr>
<tr><td>0</td><td>0</td><td>On</td><td>Setpoints are output</td></tr>
<tr><td>0</td><td>1</td><td>On</td><td>Axis/spindle disable has no effect on the FS / FA</td></tr>
<tr><td>1</td><td>0</td><td>On</td><td>Axis/spindle disable also has an effect on the FS / FA</td></tr>
<tr><td>1</td><td>1</td><td>On</td><td>No setpoint output</td></tr>
</table>
1) Leading spindle/axis active
2) Following spindle/axis active
3) Status of the coupling between leading and following spindle/axis
</td></tr>
<tr><td>Signal state 0</td><td>
There is no request for "axis / spindle disable".
Note
The reset of the signal does not take effect until the axis/spindle is stationary.
• Axis
Setpoints are output to the position controller again immediately after the signal has been reset. If a travel request is pending, it is executed immediately.
• Spindle
After the reset of the signal, the spindle disable which is still effective internally must be canceled by an axis-specific reset, channel reset or end of program (<code>M30</code> / <code>M2</code>). Only then are setpoints output to the speed controller again.
</td></tr>
</table>

DB31, ... DBX1.3	Axis/spindle disable
Corresponds with	DB31, ... DBX14.1 (activate program test) DB31, ... DBX63.3 (axis/spindle lock active) DB21, ... DBX1.7 (activate program test) DB21, ... DBX33.7 (program test active)
Additional information	Response for synchronous operation: Function Manual Axes and Spindles

18.2.6.6 DB31, ... DBX1.4 (follow-up mode)

DB31, ... DBX1.4	Follow-up mode
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Follow-up mode is active: • The position setpoint is continuously tracked: position setpoint = actual position value • Feedback signal: DB31, ... DBX61.3 = 1 (follow-up active) • Standstill and clamping monitoring are not active. • If an NC program is active when the closed-loop control system is switched on again, a control-internal repositioning operation is performed (REPOSA: Linear approach with all axes) to the last programmed position. **Note** Follow-up mode is only useful with a simultaneous reset of the controller enable: DB31, ... DBX2.1 = 0
Signal state 0	Follow-up mode is not active: • The position setpoint is not tracked: Position setpoint = programmed set position • Feedback signal: DB31, ... DBX61.3 = 0 (follow-up not active) • Standstill and clamping monitoring are active. **Note** If the controller enable is reset (DB31, ... DBX2.1 == 0), the axis is in the "hold" state. Whereby the set position of the axis is **not** corrected to the actual position. If the axis is moved in this state, e.g. manually, the difference between the set position and the actual position increases constantly (following error). The following error is suddenly corrected to zero when the "Controller enable" is set (speed setpoint jump).
Additional information	If the controller enable is cancelled in the control because of faults, the "hold" state should be activated for the axis before the NC start after the queued alarms have been successfully deleted, and the accompanying controller enable set: DB31, ... DBX1.4 = 0 (follow-up mode). Otherwise, for an NC Start and active follow-up mode, the traversing distance of the previous NC block would not be executed due to the internal delete distance-to-go. **Notice** During the transition from the "follow-up" state to the "hold" state or when the controller enable is set in position control, delete distance-to-go is activated in the control. As a consequence, a traversing block in which only this axis is traversed, is terminated directly.
Corresponds with	DB31, ... DBX2.1 (controller enable) DB31, ... DBX2.3 (clamping in progress) DB31, ... DBX61.3 (follow-up active)

18.2.6.7 DB31, ... DBX1.5 - 6 (position measuring system 1 (PMS1) / position measuring system 2 (PMS2))

DB31, ... DBX1.5 - 6	Position measuring system 1 (PMS1) / position measuring system 2 (PMS2)
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The position measuring system is active.
Signal state 0	The position measuring system is inactive.

DB31, ... DBX1.5 - 6	Position measuring system 1 (PMS1) / position measuring system 2 (PMS2)		
Additional information	**Bit 6 (PMS 2)**	**Bit 5 (PMS 1)**	**Effect**
	0	1	Position measuring system 1 is active: • Position control of the machine axis via position measuring system 1. • Monitoring functions (measuring system, standstill, clamping monitoring, contour deviation, etc.) of the machine axis via position measuring system 1. • If position measuring system 2 exists (MD30200 $MA_NUM_ENCS == 2), its actual position value is acquired, but not monitored by any of these functions.
	1	0	Position measuring system 2 is active: • Position control of the machine axis via position measuring system 2. • Monitoring functions (measuring system, standstill, clamping monitoring, contour deviation, etc.) of the machine axis via position measuring system 2. • If position measuring system 1 exists (MD30200 $MA_NUM_ENCS == 2), its actual position value is acquired, but not monitored by any of these functions.
	1	1	• Position control of the machine axis via position measuring system 1. • If position measuring system 2 is available (MD30200 $MA_NUM_ENCS == 2), its actual position value is also acquired.
	0	0	Position measuring systems 1 and 2 are inactive ("parking" of the machine axis): • There is no actual value acquisition. • The monitoring of the position measuring system has been deactivated. • The following interface signals are reset: – DB31, ... DBX60.4 / 5 == 0 (referenced/synchronized, encoder 1/2) – DB31, ... DBX61.5 (position controller active) – DB31, ... DBX61.6 (speed controller active) – DB31, ... DBX61.7 (current controller active)
	Notes • If the interface signal of the active position measuring system is reset for a traversing axis, the axis is stopped with a ramp stop without the controller enable being cancelled internally. • If a speed-controlled spindle does not have a position measuring system, the "Controller enable" interface signal must be set: DB31, ... DBX2.1 == 1 (controller enable) • After deactivation of the "parking" state, incremental position measuring systems have to be referenced to achieve the "referenced" encoder status. • If the "parking" state is active, the following interface signal is ignored at NC start for this axis: DB31, ... DBX60.4 / 5 (referenced/synchronized 1/ 2).		
	Application examples		

DB31, ... DBX1.5 - 6	Position measuring system 1 (PMS1) / position measuring system 2 (PMS2)
	1. Switching over from position measuring system 1 to positioning measuring system 2 (and vice versa). If the axis was referenced in both position measuring systems and the limit frequency of the used measured value encoder has not been exceeded in the meantime, i.e. DB31, ... DBX60.4 and 60.5 == 1 (referenced/synchronized 1/2), a new reference point approach is not required after the switchover. On switchover, the current difference between position measuring system 1 and 2 is traversed immediately. A tolerance band in which the deviation between the two actual values may lie at the switchover can be specified in the following machine data: MD36500 $MA_ENC_CHANGE_TOL (maximum tolerance for actual position value switchover) If the actual value difference is greater than the tolerance, switchover is not performed and alarm 25100 "Measuring system switchover not possible" is displayed. 2. Machine axis is parked: The position measuring system monitoring is switched off when the measured value encoder is removed. 3. Switch off position measuring system: When the position measuring system 1 or 2 is switched off, the associated interface signal is reset: DB31, ... DBX60.4/5 (referenced/synchronized 1/2) 4. Reference point approach: Reference point approach of the axis is executed with the selected position measuring system. Every position measuring system must be referenced separately.
Corresponds to	DB31, ... DBX60.4 / .5 (referenced/synchronized 1/2) DB31, ... DBX61.6 (speed controller active) DB31, ... DBX2.1 (controller enable) MD36500 $MA_ENC_CHANGE_TOL (max. tolerance on actual position value switchover) MD30200 $MA_NUM_ENCS (number of encoders)
Additional information	Function Manual Axes and Spindles

18.2.6.8 DB31, ... DBX1.7 (override active)

DB31, ... DBX1.7	Override active
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Feedrate/spindle override is active.
Signal state 0	Feedrate/spindle override is **not** active.
Additional information	The feedrate override factor (DB31, ... DBB0) specified by the NC/PLC interface is activated with the interface signal. If the feedrate/spindle override is **not** active, then independent of the actual switch position, a value of 1.0 ≙ 100 % is used as override factor in the control. The **first** switch position is an exception. Depending on the selected coding, for the first switch position the following factor applies: • Binary coding: Override factor = 0 • Gray coding: Override factor = MD12030 $MN_OVR_FACTOR_FEEDRATE[0]

DB31, ... DBX1.7	Override active
Corresponds to	MD12000 $MN_OVR_AX_IS_GRAY_CODE (axis-specific feedrate override switch Gray-coded) MD12030 $MN_OVR_FACTOR_FEEDRATE (evaluation of the path feedrate override switch) MD12100 $MN_OVR_FACTOR_LIMIT_BIN (limit for binary coded override switch)
Additional information	Function Manual Basic Functions

18.2.6.9 DB31, ... DBX2.0 (software cams: activation)

DB31, ... DBX2.0	Software cams: Activation
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to activate the output of the minus and plus cam signals of an axis to the PLC interface.
Signal state 0	A request has **not** been made to activate the output of the minus and plus cam signals of an axis to the PLC interface.
Corresponds with	DB10 DBX110.0 - 113.7 (software cams: minus cam signal 1 to 32) DB10 DBX114.0 - 117.7 (software cams: plus cam signal 1 to 32) DB31, ... DBX62.0 (software cams active)
Additional information	Function Manual Axes and Spindles

18.2.6.10 DB31, ... DBX2.1 (controller enable)

DB31, ... DBX2.1	Controller enable
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Controller is enabled. The position control loop is closed and the axis/spindle is in closed-loop control. Feedback: DB31, ... DBX61.5 = 1 (position controller active) If the axis/spindle was referenced before resetting the interface signal, the axis/spindle does not have to be re-referenced after the interface signal is set again. Supplementary condition: The limit frequency of the active measuring system must not be exceeded in the meantime. **Note** If the axis/spindle was moved from its position during the time in which the controller enable was not set, the behavior when the controller enable is set depends on the interface signal "follow-up mode": • DB31, ... DBX1.4 == 1 (follow-up mode) Position control is performed at the current position • DB31, ... DBX1.4 == 0 (follow-up mode) Position control is performed at the last position before the controller enable is reset

DB31, ... DBX2.1	Controller enable
Signal state 0	Controller is not enabled. The behavior when the "controller enable" is removed depends on whether the axis/spindle is stationary or traversing at this time: • Axis/spindle stationary: – The position control loop of the axis is opened. – For DB31, ... DBX1.4 == 1 (follow-up mode) ⇒ position setpoint = actual position value – The controller enable on the drive is reset – The following interface signals are reset: DB31, ... DBX61.5 = 0 (position controller active) DB31, ... DBX61.6 = 0 (speed controller active) DB31, ... DBX61.7 = 0 (current controller active) • Axis/spindle traverses – The axis is stopped with rapid stop. – Alarm 21612 "Controller enable VDI signal reset during motion". – The position control loop of the axis/spindle is opened. – Independent of the interface signal DB31, ... DBX1.4 (follow-up mode), the position setpoint is corrected at the end of the braking operation (position setpoint = actual position value) and the feedback signal DB31, ... DBX61.3 = 1 (follow-up mode) is set. – The following interface signals are reset: DB31, ... DBX61.5 (position controller active) DB31, ... DBX61.6 (speed controller active) DB31, ... DBX61.7 (current controller active) **Note** The current set position is retained in the control when "Controller enable" is reset and "Follow-up mode" (DB31, ... DBX1.4 == 0) is not set. If the axis is moved in this state, e.g. manually, the difference between the set position and the actual position increases constantly (following error). The following error is suddenly corrected to zero when the "Controller enable" is set (speed setpoint jump).

DB31, ... DBX2.1	Controller enable
Additional information	**Mechanical clamping of an axis** If the axis is positioned at the clamping position, the clamping mechanism is closed. The controller enable is then reset. Otherwise, the position controller would constantly work against the clamping, if the axis was moved mechanically from its specified position during the clamping operation. When the clamping is removed, the controller enable is set first and then the mechanical clamping is released. **Note** • Travel request for an axis/spindle without controller enable: – The axis/spindle is not traversed – The travel command is output to the interface – As long as the travel request is present, the axis/spindle is traversed immediately when the controller enable is set. • When the controller enable is reset for a traversing geometry axis, this always results in a contour violation.
Corresponds with	DB31, ... DBX61.3 (follow-up active) DB31, ... DBX1.4 (follow-up mode) DB31, ... DBX61.5 (position controller active) DB31, ... DBX61.6 (speed controller active) DB31, ... DBX61.7 (current controller active) MD36620 $MA_SERVO_DISABLE_DELAY_TIME (OFF delay of the controller enable) MD36610 $MA_AX_EMERGENCY_STOP_TIME (braking ramp time when errors occur)

18.2.6.11 DB31, ... DBX2.2 (spindle reset/delete distance to go)

DB31, ... DBX2.2	Delete distance-to-go/spindle reset
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	**Axis**: "Delete-distance-to-go" is requested. **Spindle**: Reset is requested
Edge change 1 → 0	No effect.

DB31, ... DBX2.2	Delete distance-to-go/spindle reset
Additional information	**Axis: Delete distance-to-go** • AUTOMATIC and MDI modes The interface signal is only active if the axis is traversed as positioning axis. A positioning axis is decelerated to standstill along its brake characteristic. The distance-to-go of the axis is deleted. • JOG mode The axis is braked down to standstill alone its deceleration characteristic and then a possible delete distance to go is a deleted. **Note** After the delete distance to go, in an NC program, the following block is prepared again with the actual axis position. As a consequence, a different contour is traversed along than programmed in the NC program. Traversing to absolute position `G90` in the block after "Delete distance-to-go" at least allows the programmed position is approached. When traversing using incremental dimension `G91`, then the programmed position would not be approached.
	Spindle: reset • Control mode: – The spindle is stopped – The NC program is again executed – The spindle continues to rotate with the next M and S value programmed in the NC program. • Oscillation mode: – Oscillation is interrupted – The axes continue to traverse – The NC program continues with the actual gear stage – The spindle continues to rotate with the next M and S value programmed in the NC program. The speed obtained from these values (actual S value and the last active gear stage) could be too high. In this case, the interface signal is set. DB31, ... DBX83.1 • Positioning mode: The spindle is stopped • Axis mode: The spindle is stopped **Note** The response for a spindle reset is independent of the setting in machine data: MD35040 $MA_SPIND_ACTIVE_AFTER_RESET
	Dwell time (`G4`): Delete distance-to-go During a block with dwell time (`G4`), the interface signal has **no** influence on the time that elapses.
Corresponds with	DB21, ... DBX6.2 (delete distance-to-go) DB31, ... DBX83.1 (programmed speed too high) MD35040 $MA_SPIND_ACTIVE_AFTER_RESET (own spindle reset)
Additional information	Function Manual Axes and Spindles

18.2.6.12 DB31, ... DBX2.3 (clamping in progress)

DB31, ... DBX2.3	Clamping in progress
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Clamping in progress. The clamping monitoring is active.
Signal state 0	Clamping completed. Standstill monitoring is active.
Corresponds with	MD36050 $MA_CLAMP_POS_TOL (clamping tolerance)

18.2.6.13 DB31, ... DBX2.4 - 7 (reference point value 1 - 4)

<table>
<tr><th>DB31, ... DBX2.4 - 7</th><th colspan="5">Reference point value 1-4</th></tr>
<tr><td>Signal flow</td><td colspan="5">PLC → NC</td></tr>
<tr><td>Update</td><td colspan="5">Cyclic</td></tr>
<tr><td>Additional information</td><td colspan="5">When the reference cam is reached, the NC is signaled by the interface which coded reference cam has been approached.
Actual value = MD34100 $MA_REFP_SET_POS [<reference point value x>]</td></tr>
<tr><td></td><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Meaning</td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>Reference point value 1</td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>1</td><td>Reference point value 1</td></tr>
<tr><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>Reference point value 2</td></tr>
<tr><td></td><td>0</td><td>1</td><td>0</td><td>0</td><td>Reference point value 3</td></tr>
<tr><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>Reference point value 4</td></tr>
<tr><td></td><td colspan="5">Notes
• The interface signal must remain set until the reference point is reached or until a new coded reference cam is approached.
• If the machine axis has reached the reference point, and noreference point value has been selected, then reference point value 1 is used.
• For length measuring systems with distance-coded reference marks, the interface signal is irrelevant.
Application example
On a machine tool with large traversing paths, up to four coded reference cams can be distributed over the distance traveled by the axis; up to four different reference points approached. This allows the traversing time when referencing to be shortened.</td></tr>
<tr><td>Corresponds with</td><td colspan="5">MD34100 $MA_REFP_SET_POS (reference point value)</td></tr>
<tr><td>Additional information</td><td colspan="5">Function Manual Axes and Spindles</td></tr>
</table>

18.2.6.14 DB31, ... DBX3.0 (accept external work offset)

DB31, ... DBX3.0	Accept external work offset
Signal flow	PLC→ NC
Update	Cyclic
Signal state 0 → 1	Accepting the values for the external work offset is requested.
Signal state 1 → 0	Accepting the values for the external work offset is **not** requested.
Additional information	With the interface signal, the value in system variable $AA_ETRANS is transferred into the frame of the external work offset as external work offset ($P_EXTFRAME, $P_EXTFR): DB31, ... DBX3.0: 0 → 1 ⇒ $P_EXTFRAME[<axis>] = $P_EXTFR[<axis>] = $AA_ETRANS[<axis>]
Corresponds with ...	$AA_ETRANS[<axis>] (external work offset) $P_EXTFRAME[<axis>] (active system frame, external frame) $P_EXTFR[<axis>] (data management frame for external frame) MD28082 $MC_MM_SYSTEM_FRAME_MASK,bit1 (system frame)
Additional information	Function Manual Axes and Spindles

18.2.6.15 DB31, ... DBX3.1 (enable travel to fixed stop)

DB31, ... DBX3.1	Enable travel to fixed stop
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	**Positive** feedback to the request to enable travel to the fixed stop with DB31, ... DBX62.4 = 1.
Signal state 0	**Negative** feedback to the request to enable travel to the fixed stop with DB31, ... DBX62.4 = 1. Meaning **after activating** the function "travel to fixed stop" in the NC program (FXS): • Travel to fixed stop is inhibited. • The axis is stationary at the start position with reduced torque. • The channel message "Wait: HiFu acknowledgment missing" is displayed. Meaning **before reaching** the fixed stop DB31, ...DBX62.5 == 0 (fixed stop reached): • Travel to fixed stop interrupted. • Alarm "20094: Axis%1 function was canceled" is displayed. Meaning **after reaching** the fixed stop DB31, ...DBX62.5 == 1 (fixed stop reached): • The torque limiting and monitoring of the fixed stop monitoring window are canceled.
Additional information	Irrelevant for MD37060 $MA_FIXED_STOP_ACKN_MASK == 0 or 2
Corresponds with	DB31, ... DBX1.1 (acknowledge fixed stop reached). DB31, ... DBX1.2 (fixed stop sensor) DB31, ... DBX62.4 (activate travel to fixed stop) DB31, ... DBX62.5 (fixed stop reached) MD37060 $MA_FIXED_STOP_ACKN_MASK (monitoring PLC acknowledgements for travel to fixed stop)
Additional information	Function Manual Axes and Spindles

18.2.6.16 DB31, ... DBX3.2 - 5 (activate fixed feedrate 1 - 4, machine axes)

<table>
<tr><th>DB 31, ... DBX3.2 - 5</th><th colspan="5">Activate fixed feedrate 1 - 4, machine axes</th></tr>
<tr><td>Signal flow</td><td colspan="5">PLC → NC</td></tr>
<tr><td>Update</td><td colspan="5">Cyclic</td></tr>
<tr><td>Additional information</td><td colspan="5">With the interface signals, in the JOG mode, the fixed feedrate - parameterized using machine data - is activated.</td></tr>
<tr><td></td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Meaning</td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>Fixed feedrate not selected</td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>1</td><td>Fixed feedrate 1</td></tr>
<tr><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>Fixed feedrate 2</td></tr>
<tr><td></td><td>0</td><td>1</td><td>0</td><td>0</td><td>Fixed feedrate 3</td></tr>
<tr><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>Fixed feedrate 4</td></tr>
<tr><td></td><td colspan="5">Instead of the parameterized JOG velocity/ JOG rapid traverse velocity (MD... ...JOG_VELO_...), the axis is traversed with the selected fixed feedrate. The traversing direction is specified via the interface signal:
DB31, ... DBX6 / 7 (traversing keys, minus/plus)</td></tr>
<tr><td></td><td colspan="5">The fixed feedrates are parameterized using the following machine data:
• Linear axes: MD12202 $MN_PERMANENT_FEED
• Rotary axes: MD12204 $MN_PERMANENT_ROT_AX_FEED</td></tr>
<tr><td></td><td colspan="5">Notes
• The fixed feedrate is not used for:
– Spindles
– Positioning axes
– Tapping
• The fixed feedrate is always interpreted as linear feedrate. Switchover to linear feedrate is performed internally in the control system also when revolutional feedrate is active.</td></tr>
<tr><td>Corresponds with</td><td colspan="5">MD12200 $MN_RUN_OVERRIDE_0 (traversing behavior with override 0)
MD12202 $MN_PERMANENT_FEED (fixed feedrates for linear axes)
MD12204 $MN_PERMANENT_ROT_AX_FEED (fixed feedrates for rotary axes)
MD21150 $MC_JOG_VELO_RAPID_ORI (conventional rapid traverse for orientation axes)
MD21155 $MC_JOG_VELO_ORI (conventional velocity for orientation axes)
MD21160 $MC_JOG_VELO_RAPID_GEO (conventional rapid traverse for geometry axes)
MD21165 $MC_JOG_VELO_GEO (conventional velocity for geometry axes)
MD32010 $MA_JOG_VELO_RAPID (conventional rapid traverse)
MD32020 $MA_JOG_VELO (conventional axis velocity)</td></tr>
<tr><td>Additional information</td><td colspan="5">Function Manual Axes and Spindles</td></tr>
</table>

18.2.6.17 DB31, ... DBX3.6 (velocity/spindle speed limiting)

DB31, ... DBX3.6	Velocity/spindle speed limitation
Signal flow	PLC → NC
Update	Cyclic

DB31, ... DBX3.6	Velocity/spindle speed limitation
Signal state 1	The velocity/spindle speed limiting is active.
Signal state 0	The velocity/spindle speed limiting is not active.
Additional information	The control limits the velocity/spindle speed to the parameterized limit value: MD35160 $MA_SPIND_EXTERN_VELO_LIMIT
Corresponds with	MD35100 $MA_SPIND_VELO_LIMIT (maximum spindle speed) MD35160 $MA_SPIND_EXTERN_VELO_LIMIT (spindle speed limiting from the PLC) SD43220 $SA_SPIND_MAX_VELO_G26 (programmable upper spindle speed limitation for G26) SD43230 $SA_SPIND_MAX_VELO_LIMS (spindle speed limitation for G96/G961/G97)

18.2.6.18 DB31, ... DBX4.0 - 2 (activate handwheel)

DB31, ... DBX4.0 - 2	Activate handwheel
Signal flow	PLC → NC
Update	Cyclic
Additional information	The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.

Bit coded: **maximum 3** handwheels

Bit 2	Bit 1	Bit 0	Number of the assigned handwheel
0	0	0	No handwheel assignment
0	0	1	1
0	1	0	2
1	0	0	3

Binary coded: **maximum 6** handwheels

Bit 2	Bit 1	Bit 0	Number of the assigned handwheel
0	0	0	No handwheel assignment
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6

Note

At any one time, the machine axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, then the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".

Corresponds with	DB31, ... DBX64.0 - 2 (handwheel active) MD11324 $MN_HANDWH_VDI_REPRESENTATION (representation of the handwheel number at the VDI interface)
Additional information	Function Manual Axes and Spindles

18.2.6.19 DB31, ... DBX4.3 (feedrate/spindle stop, axis-specific)

<table>
<tr><th>DB31, ... DBX4.3</th><th>Feedrate/spindle stop, axis-specific</th></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Signal state 1</td><td>Feedrate/spindle stop is active.</td></tr>
<tr><td>Signal state 0</td><td>Feedrate/spindle stop is not active.</td></tr>
<tr><td>Additional information</td><td>Axis
• If the interface signal is set while traversing the axis, then the axis is braked down to standstill along its braking characteristic. If the axis is in an interpolatory relationship with other axes, these are also braked.
• After the interface signal has been reset, traversing motion that was stopped is continued.
• The position control is retained and the following error is eliminated.
• If the interface signal is set, and a request is issued to traverse, then the axis is not traversed. However, the traverse request is kept. When the interface signal is reset, the traverse request is immediately executed, i.e. the axis is traversed.
• The interface signal is active in all modes.
<table>
<tr><th>Thread cutting</th><th>Effectiveness</th></tr>
<tr><td>G33, G34, G35</td><td>Effective (contour deviations occur)</td></tr>
<tr><td>G331, G332</td><td>Effective</td></tr>
<tr><td>G63</td><td>Effective</td></tr>
</table>
Spindle
• If the interface signal is set while traversing the spindle, then the spindle is braked down to standstill along its braking characteristic. If the spindle is in an interpolatory relationship with other spindles, these are also braked.
• After the interface signal has been reset, traversing motion that was stopped is continued.
• During positioning, the position control is retained and the following error is eliminated.
• If the interface signal is set, and a request is issued to move during positioning operation, then the spindle is not moved. However, the traverse request is kept. When the interface signal is reset, the traverse request is immediately executed, i.e. the spindle is moved.
• The interface signal is active in all modes.
• The interface signal is not active during tapping (G331, G332).
<table>
<tr><th>Thread cutting</th><th>Effectiveness</th></tr>
<tr><td>G33, G34, G35</td><td>Effective (contour deviations can occur depending on the dynamic performance that has been parameterized)</td></tr>
<tr><td>G331, G332</td><td>Not effective</td></tr>
<tr><td>G63</td><td>Effective</td></tr>
</table>
Note
The interface signal is not active when "Axis/spindle inhibit" (DB31, ... DBX1.3 == 1) is active.</td></tr>
<tr><td>Corresponds with</td><td>DB31, ... DBX1.3 (axis/spindle disable)</td></tr>
<tr><td>Additional information</td><td>Function Manual Axes and Spindles</td></tr>
</table>

18.2.6.20 DB31, ... DBX4.4 (traversing key disable)

DB31, ... DBX4.4	Traversing key lock
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The "Plus" and "Minus" traversing keys are locked.
Signal state 0	The "Plus" and "Minus" traversing keys are enabled.
Additional information	**Note** If the traversing key disable is activated while traversing, the axis is stopped.
Corresponds with	DB31, ... DBX4.6 - 7 (traversing keys "Plus" / "Minus")
Additional information	Function Manual Axes and Spindles

18.2.6.21 DB31, ... DBX4.5 (rapid traverse override)

DB31, ... DBX4.5	Rapid traverse override
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	When manually traversing the machine axis using the traversing keys, when the interface signal is set, rapid traverse velocity becomes active (MD32010).
Signal state 0	When manually traversing the machine axis using the traversing keys, the entered JOG velocity becomes active (SD41110 or MD32020).
Additional information	The signal is only active for continuous or incremental manual traversing in the JOG mode. The signal is irrelevant: • for reference point approach (JOG mode) • in the AUTOMATIC and MDI modes The rapid traverse velocity can be influenced using the rapid traverse override switch.
Corresponds with	DB31, ... DBX4.6 - 7 (traversing keys "Plus" / "Minus") SD41110 $SN_JOG_SET_VELO (axis velocity for JOG) MD32010 $MA_JOG_VELO_RAPID (conventional rapid traverse) MD32020 $MA_JOG_VELO (conventional axis velocity)
Additional information	Function Manual Axes and Spindles

18.2.6.22 DB31, ... DBX4.6 - 7 ("Plus" / "Minus" traversing keys)

DB31, ... DBX4.6 - 7	Traversing keys "Plus" / "Minus"
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to traverse the machine axis using the traversing key in the positive/negative axis direction.
Signal state 0	A request has **not** been made to traverse the machine axis using the traversing key in the positive/negative axis direction.

DB31, ... DBX4.6 - 7	Traversing keys "Plus" / "Minus"	
Additional information	There is a request signal for every traversing key and/or axis direction:	
	Bit 6	Traversing key "Minus" (for traversing in the negative axis direction)
	Bit 7	Traversing key "Plus" (for traversing in the positive axis direction)
	Depending on the active machine function, as well as the settings for jog and continuous operation (SD41050 and MD11300), different responses are initiated when the signal changes: 1. Continuous manual travel in the jog mode The machine axis traverses in the direction concerned to the active limit switch as long as the interface signal is set to 1. 2. Continuous manual travel in continuous operation On the first edge change 0 →1 the machine axis starts to traverse in the relevant direction. This traversing movement still continues when the edge changes from 1 → 0. Any new signal edge change 0 → 1 (same traversing direction!) stops the traversing movement. 3. Incremental manual travel in the jog mode With signal 1 the machine axis starts to traverse at the set increment. If the signal changes to the 0 state before the increment is traversed, then traversing is interrupted. When the signal state changes to 1 again, traversing is continued. The axis can be stopped and started several times as described above until it has traversed the complete increment. 4. Incremental manual travel in continuous operation At the first edge change 0 → 1, the machine axis starts to traverse at the set increment. If another edge change 0 → 1 is performed with the same traverse signal before the geometry axis has traversed the increment, the traversing movement will be cancelled. The increment is no longer traversed to the end.	
	Note • The interface signal is irrelevant in operating modes: – AUTOMATIC – MDI • If both traversing signals ("plus" and "minus") are set at the same time there is no traversing or the current traversing is aborted. • Traversing by means of the traversing keys can be locked via DB31, ... DBX4.4 (traversing key disable).	
Corresponds with	DB31, ... DBX4.4 (traversing key disable) DB31, ... DBX64.6 - 7 ("Plus" and "Minus" traversing command) MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD (INC and REF in jog mode) SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD (jog/continuous operation for JOG, continuous)	
Additional information	Function Manual Axes and Spindles	

18.2.6.23 DB31, ... DBX5.0 - 6 (machine function request)

DB31, ... DBX5.0 - 6	Machine function request
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The machine function to manually traverse the machine axis has been requested.

DB31, ... DBX5.0 - 6	Machine function request	
Signal state 0	The machine function to manually traverse the machine axis has **not** been requested.	
Additional information	There is a request signal for every machine function to manually traverse the machine axis in the JOG mode.	
	Bit 0	INC1
	Bit 1	INC10
	Bit 2	INC100
	Bit 3	INC1000
	Bit 4	INC10000
	Bit 5	INCvar
	Bit 6	Continuous manual travel
	Incremental manual travel In addition to five fixed increment sizes (default setting in MD11330: INC1, INC10, INC100, INC1000 and INC10000), a variable increment size (INCvar) that can be set via the setting data SD41010 is also available. The distance evaluation of one increment for fixed and variable increment sizes is performed via the axis-specific machine data MD31090. When pressing the "Plus" or "Minus" traversing key, or by rotating the electronic handwheel, the axis starts to traverse in the appropriate direction, corresponding to the number of increments of the active machine function. **Continuous manual travel** For continuous manual travel, the plus and minus traversing keys are selected to move the axis continuously and the appropriate direction. **Notes** • If several requests are set simultaneously, no machine function becomes active. • If an axis is currently being traversed via a machine function, the movement is aborted through deselection or change of the machine function.	
Corresponds with	DB31, ... DBX65.0 - 65.6 (active machine function) MD11320 $MN_HANDWH_IMP_PER_LATCH (handwheel pulses per detent position) MD11330 $MN_JOG_INCR_SIZE_TAB (increment size for INC/handwheel) MD31090 $MA_JOG_INCR_WEIGHT (evaluation of an increment for INC/handwheel) SD41010 $SN_JOG_VAR_INCR_SIZE (size of the variable increment for JOG)	
Additional information	Function Manual Axes and Spindles	

18.2.6.24 DB31, ... DBX7.0 (invert handwheel direction of rotation)

DB31, ... DBX7.0	Invert handwheel direction of rotation
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A request has been made to invert the direction of rotation of the handwheel assigned to the machine axis.
Signal state 0	A request has **not** been made to invert the direction of rotation of the handwheel assigned to the machine axis.

DB31, ... DBX7.0	Invert handwheel direction of rotation
Additional information	**Note** It is only permissible to change the interface signal when the machine axis is at a standstill. **Application examples** • The direction of movement of the handwheel does not match the expected direction of the axis. • A handwheel is assigned to several axes with different orientations.
Corresponds with	DB31, ... DBX67.0 (handwheel direction of rotation inversion active)
Additional information	Function Manual Axes and Spindles

18.2.6.25 DB31, ... DBB8 (request axis/spindle replacement)

DB31, ... DBB8	Axis/spindle exchange request				
Signal flow	PLC → NC				
Update	Cyclic				
Edge change 0 → 1	Request from PLC, switchover axis				
Edge change 1 → 0	Acknowledged by the NC				
Additional information	From the PLC user program, via DBB8 an axis/spindle exchange can be requested: • From an NC channel to the PLC • From the PLC to an NC channel • From an NC channel to another NC channel				
	Bit	**Meaning**			
	0 ... 3	Name of the channel to which the axis/spindle is to be assigned (binary code). Example: The axis should be assigned to channel 2.			
		Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Channel number**
		0	0	1	0	2
	4	Request axis/spindle exchange				
		0 → 1	Request from PLC			
		1 → 0	Acknowledged by the NC			
	5	-				
	6	-				
	7	Request "PLC axis"				
Corresponds with	DB31, ... DBB68 (Axis/spindle replacement) MD20070 $MC_AXCONF_MACHAX_USED (machine axis number valid in channel) MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN (initial setting of channel for axis replacement)				
Additional information	Function Manual Axes and Spindles				

18.2.6.26 DB31, ... DBX9.0 - (2 select: position controller parameter set)

DB31, ... DBX9.0 - 2	Selection: Position controller parameter set			
Signal flow	PLC → NC			
Update	Job-controlled			
Additional information	Bit 2	Bit 1	Bit 0	Parameter set
	0	0	0	1
	0	0	1	2
	0	1	0	3
	0	1	1	4
	1	0	0	5
	1	0	1	6
	1	1	0	6
	1	1	1	6
	Note Selecting a position controller parameter set is ignored, for: MD35590 $MA_PARAMSET_CHANGE_ENABLE == 0			
Corresponds with	DB31, ...DBX69.0, .1, ..2 (feedback signal: active position controller parameter set)			
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, drive, "Commissioning NC"			

18.2.6.27 DB31, ... DBX9.3 (parameter set definitions disabled by NC)

DB31, ... DBX9.3	Parameter set definitions disabled by NC
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	The parameter set switchover is disabled.
Signal state 0	The parameter set switchover is enabled.
Corresponds with	DB31, ... DBX9.0, .1, .2

18.2.6.28 DB31, ... DBX10.0 (REPOS delay)

DB31, ... DBX10.0	REPOS delay
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A REPOS delay is requested. After a block search, a REPOS offset is applied for this axis. However it is **not** applied using the **approach block**, but rather using the next **traversing block** in which the axis is **programmed**.
Signal state 0	A REPOS delay is **not** requested. After a block search, a REPOS offset is applied for this axis using the **approach** block.

DB31, ... DBX10.0	REPOS delay
Additional information	**Note** If the machine axes are involved on a path (DB31, ... DBX76.4 == 1 (path axis)), then the interface signal is not active.
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.0 - 2 (REPOS mode) DB21, ... DBX31.4 (REPOS mode change) DB31, ... DBX72.0 (REPOS delay) DB21, ... DBX319.0 (REPOS mode change acknowledgment) DB21, ... DBX319.1 - 3 (active REPOS mode) DB21, ... DBX319.5 (REPOS acknowledgment delay) DB31, ... DBX70.0 (REPOS offset) DB31, ... DBX70.1 (REPOS offset valid) DB31, ... DBX70.2 (REPOS delay acknowledgment) DB31, ... DBX76.4 (path axis) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)
Additional information	Function Manual Basic Functions

18.2.6.29 DB31, ... DBX12.0 (hardware limit switch minus)

DB31, ... DBX12.0	Hardware limit switch minus:
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The hardware limit switch minus was activated.
Signal state 0	The hardware limit switch minus was not activated.
Additional information	Alarm 21614 "Hardware limit switch -" is displayed, and the axis is stopped.
Corresponds with	MD36600 $MA_BRAKE_MODE_CHOICE (deceleration behavior when the hardware limit switch responds)

18.2.6.30 DB31, ... DBX12.1 (hardware limit switch plus)

DB31, ... DBX12.1	Hardware limit switch plus
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Hardware limit switch plus was triggered.
Signal state 0	Hardware limit switch plus was not triggered.
Additional information	Alarm 21614 "Hardware limit switch +" is displayed, and the axis is stopped.
Corresponds with	MD36600 $MA_BRAKE_MODE_CHOICE (deceleration behavior when the hardware limit switch responds)

18.2.6.31 DB31,... DBX12.2 (second software limit switch minus)

DB31, ... DBX12.2	Second software limit switch minus
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Second software limit switch minus
Signal state 0	The second software limit switch minus was not triggered.
Corresponds with	MD36100 $MA_POS_LIMIT_MINUS (1st software limit switch minus) MD36120 $MA_POS_LIMIT_MINUS2 (2nd software limit switch minus)

18.2.6.32 DB31, ... DBX12.3 (second software limit switch plus)

DB31, ... DBX12.3	Second software limit switch plus
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The second software limit switch plus was triggered.
Signal state 0	The second software limit switch plus was not triggered.
Corresponds with	MD36110 $MA_POS_LIMIT_PLUS (1st software limit switch plus) MD36130 $MA_POS_LIMIT_PLUS2 (2nd software limit switch plus)

18.2.6.33 DB31, ... DBX12.4 (modulo rotary axis: activate traversing range limits)

DB31, ... DBX12.4	Modulo rotary axis: activate traversing range limits
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activate traversing range limits (software limit switch, working area limits) for modulo rotary axes.
Signal state 0	Deactivate traversing range limits for modulo rotary axes.
Additional information	**Note** The signal is irrelevant if linear axes/rotary axes do not have modulo functionality. **Application example** Mounted rotary axis with monitoring
Corresponds with	DB31, ... DBX74.4 (modulo rotary axis: traversing range limits active)
Additional information	Function Manual Axes and Spindles

18.2.6.34 DB31, ... DBX12.7 (reference point approach deceleration)

DB31, ... DBX12.7	Deceleration of reference point approach
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The machine axis is positioned **at** the reference cam.

DB31, ... DBX12.7	Deceleration of reference point approach
Signal state 0	The machine axis is positioned **in front** of the reference cam.
Additional information	**Note** Using a suitable reference cam, which extends from the reference point to the end of the traversing range, it is recommended to ensure that before the reference point approach, the machine axis can be located in the range behind the reference cam.
Corresponds with	DB31, ... DBX2.4 - 7 (reference point value 1 to 4)
Additional information	Function Manual Axes and Spindles

18.2.6.35 DB31, ... DBX13.0 - 2 (JOG approach fixed point)

DB31, ... DBX13.0 - 2	JOG fixed point approach
Signal flow	PLC → NC
Update	Cyclic
Additional information	After selecting the "Approach fixed point in JOG" function, the PLC outputs the number of the fixed point to be approached binary coded to the NC using DB31, ... DBX.0 - 2:

Bit 2	Bit 1	Bit 0	Number of fixed point to be approached
0	0	0	-
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4

	As soon as the function is active (see DB31, ... DBX75.0 - 2), then the selected machine axis can be traversed to the corresponding fixed point using either the traversing keys or the handwheel. The fixed points are defined using machine data MD30600.
Corresponds with	DB31, ... DBX75.0 - 2 (JOG approach fixed point active) DB31, ... DBX75.3 - 5 (JOG approach fixed point reached) MD30600 $MA_FIX_POINT_POS[<n>] (fixed value positions of the axis)
Additional information	Function Manual Axes and Spindles

18.2.6.36 DB31, ... DBX13.3 (JOG travel to position)

DB31, ... DBX13.3	JOG travel to position
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The function "Position travel in JOG" is requested:
Signal state 0	The function "Position travel in JOG" is **not** requested:

DB31, ... DBX13.3	JOG travel to position
Additional information	As soon as the function is active (see DB31, ... DBX75.6), then using the traversing keys or the handwheel, the selected machine axis can be traversed to the position specified with setting data SD43320. **Application** The function is used for machine tools that are exclusively operated manually, e.g. to approach drill-holes in precise increments. **Note** The following preconditions applying for the "Position travel in JOG" function: • The function can only be activated in the JOG mode. The function cannot be activated if the machine function JOG-REPOS or JOG-REF is active and in JOG in AUTOMATIC. • The axis to be traversed must be referenced. • A kinematic transformation may not be active. • The axis to be traversed may not be a following axis of an active coupling.
Corresponds with	DB31, ... DBX75.6 (JOG travel to position active) DB31, ... DBX75.7 (JOG position reached) SD43320 $SA_JOG_POSITION (JOG position)
Additional information	Function Manual Axes and Spindles

18.2.6.37 DB31, ... DBX14.0 (suppress program test)

DB31, ... DBX14.0	Suppress program test
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Suppression of the program test (setpoint output locked) is requested.
Signal state 0	Suppression of the program test (setpoint output locked) is **not** requested.
Additional information	If function "Program test" (PRT) is active in the channel to which the axis currently belongs, the axis is traversed with internal axis disable. Setpoints are generated but they not output to the machine axes. Actual value = Setpoint. If the program test is suppressed for the axis, setpoints are output to the machine axis despite the function "program test" being active in the channel to which the axis currently belongs. **Automatic transfer of the interface signals** The HMI request signals DB21, ... DBX128.0 / .1 are only transferred from the basic PLC program to the PLC request signals DB21, ... DBX14.0 / .1 if FB1 parameter **MMCToIF** is set to **TRUE**. If the parameter is not set, the PLC request signals must be set by the PLC user program.
Corresponds with	DB31, ... DBX14.1 (activate program test) DB31, ... DBX128.0 (suppress program test) DB31, ... DBX128.1 (activate program test)
Additional information	Function Manual Basic Functions

18.2.6.38 DB31, ... DBX14.1 (activate program test)

DB31, ... DBX14.1	Activate program test
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activation of the function "program test" (setpoint output locked) is requested.
Signal state 0	Activation of the function "program test" (setpoint output locked) is **not** requested.
Additional information	If the function "program test" is active, the axis is traversed with internal axis disable. Setpoints are generated but they not output to the machine axes. Actual value = Setpoint. **Automatic transfer of the interface signals** The HMI request signals DB21, ... DBX128.0 / .1 are only transferred from the basic PLC program to the PLC request signals DB21, ... DBX14.0 / .1 if FB1 parameter **MMCToIF** is set to **TRUE**. If the parameter is not set, the PLC request signals must be set by the PLC user program.
Corresponds with	DB31, ... DBX14.0 (suppress program test) DB31, ... DBX128.0 (suppress program test) DB31, ... DBX128.1 (activate program test)
Additional information	Function Manual Basic Functions

18.2.6.39 DB31, ... DBX16.0 - 2 (actual gear stage)

DB31, ... DBX16.0 - 2	Actual gear stage
Signal flow	PLC → NC
Update	Cyclic

<table>
<tr><th>DB31, ... DBX16.0 - 2</th><th>Actual gear stage</th></tr>
<tr><td>Additional information</td><td>If the new gear stage is selected at the machine, the PLC user program sets the following interface signals as feedback signal to the NC:
• DB31, ... DBX16.0 - 2 (actual gear stage)
• DB31, ... DBX16.3 (gear stage has been changed)
This signals the NC that the correct stage has been successfully selected. The associated parameter set becomes active in the NC.
<table>
<tr><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Parameter set No.</th><th>Meaning: Parameter set for</th></tr>
<tr><td>---</td><td>---</td><td>---</td><td>0</td><td>Axis mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td rowspan="2">1</td><td>Gear stage 1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Gear stage 1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td><td>Gear stage 2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td><td>Gear stage 3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td><td>Gear stage 4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td rowspan="3">5</td><td>Gear stage 5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Gear stage 5</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Gear stage 5</td></tr>
</table>
Note
If the PLC user program signals back to the NC with a different actual gear stage than issued by the NC as the set gear stage requested by the PLC, the gear change is still considered to have been successfully completed and the actual gear stage that is signaled back is activated in the NC.</td></tr>
<tr><td>Corresponds with</td><td>DB31, ... DBX16.3 (gear stage has been changed)
DB31, ... DBX18.5 (oscillation speed)
DB31, ... DBX82.0 - 2 (set gear stage)
DB31, ... DBX82.3 (change over gear stage)</td></tr>
<tr><td>Additional information</td><td>Function Manual Axes and Spindles</td></tr>
</table>

18.2.6.40 DB31, ... DBX16.3 (gear stage has been changed)

DB31, ... DBX16.3	Gear is changed over
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	Gear is changed over
Edge change 1 → 0	No effect.

DB31, ... DBX16.3	Gear is changed over
Additional information	If the new gear stage is selected at the machine, the PLC user program sets the following interface signals as feedback signal to the NC: • DB31, ... DBX16.0 - 2 (actual gear stage) • DB31, ... DBX16.3 = 0 → 1 (gear stage has been changed) Feedback signal from the NC to the PLC: DB31, ... DBX82.3 = 0 (changeover gear stage) Feedback signal from the PLC to the NC: DB31, ... DBX16.3 = 0 (gear stage has been changed) This completes the gear stage change in the control system. **Note** • The interface signal is irrelevant for all spindle modes **with the exception** of the oscillation mode. • If the PLC user program signals back to the NC with a different actual gear stage than issued by the NC as the set gear stage requested by the PLC, the gear change is still considered to have been successfully completed and the actual gear stage that is signaled back is activated in the NC.
Corresponds with	DB31, ... DBX16.0 - 2 (actual gear stage) DB31, ... DBX82.0 - 2 (set gear stage) DB31, ... DBX82.3 (change over gear stage) DB31, ... DBX18.5 (oscillation speed)
Additional information	Function Manual Axes and Spindles

18.2.6.41 DB31, ... DBX16.4 (resynchronize spindle, measuring system 1)

DB31, ... DBX16.4	Resynchronize spindle, measuring system 1
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	There is an active request to synchronize measuring system 1 of the spindle.
Edge change 1 → 0	No effect.
Additional information	**Note** The interface signal is irrelevant for all spindle modes **with the exception** of the control mode.
Corresponds with	DB31, ... DBX60.4 (referenced / synchronized 1) DB31, ... DBX60.5 (referenced / synchronized 2)
Additional information	Function Manual Axes and Spindles

18.2.6.42 DB31, ... DBX16.5 (resynchronize spindle, measuring system 2)

DB31, ... DBX16.5	Resynchronize spindle, measuring system 2
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	There is an active request to synchronize measuring system 2 of the spindle.
Edge change 1 → 0	No effect.

DB31, ... DBX16.5	Resynchronize spindle, measuring system 2
Additional information	**Note** The interface signal is irrelevant for all spindle modes **with the exception** of the control mode.
Corresponds with	DB31, ... DBX60.4 (referenced / synchronized 1) DB31, ... DBX60.5 (referenced / synchronized 2)
Additional information	Function Manual Axes and Spindles

18.2.6.43 DB31, ... DBX16.7 (delete S value)

DB31, ... DBX16.7	Delete S value
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	There is an active request to delete the S value.
Edge change 1 → 0	No effect.
Additional information	**Control mode:** • The spindle is braked via its parameterized braking ramp down to standstill. **Oscillation mode, axis mode, positioning mode:** • Deleting the actual S value has no effect on the actual function. However, if the control mode is selected again, a new S value must be programmed.
Additional information	Function Manual Axes and Spindles

18.2.6.44 DB31, ... DBX17.4 (resynchronizing the spindle before positioning, measuring system 1)

DB31, ... DBX17.4	Resynchronizing the spindle before positioning, measuring system 1
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	There is an active request to resynchronize measuring system 1 before positioning the spindle.
Edge change 1 → 0	No effect.
Additional information	**Note** The interface signal is irrelevant for all spindle modes **with the exception** of the positioning mode.
Corresponds with	DB31, ... DBX60.4 (referenced/synchronized 1)
Additional information	Function Manual Axes and Spindles

18.2.6.45 DB31, ... DBX17.5 (resynchronizing the spindle before positioning, measuring system 2)

DB31, ... DBX17.5	Resynchronizing the spindle before positioning, measuring system 2
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	There is an active request to resynchronize measuring system 2 before positioning the spindle.

DB31, ... DBX17.5	Resynchronizing the spindle before positioning, measuring system 2
Edge change 1 → 0	No effect.
Additional information	**Note** The interface signal is irrelevant for all spindle modes **with the exception** of the positioning mode.
Corresponds with	DB31, ... DBX60.2 (referenced/synchronized 2)
Additional information	Function Manual Axes and Spindles

18.2.6.46 DB31, ... DBX17.6 (invert M3 / M4)

DB31, ... DBX17.6	Invert M3/M4
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	There is an active request to invert the direction of rotation programmed with M3 / M4.
Edge change 1 → 0	No effect.
Additional information	The direction of rotation of the spindle motor changes for the following functions: • M3 • M4 • M5 • SPOS/M19/SPOSA from the motion; not effective for SPOS/M19/SPOSA from zero speed (stationary).
Additional information	Function Manual Basic Functions

18.2.6.47 DB31, ... DBX18.4 (oscillation controlled by the PLC)

DB31, ... DBX18.4	Oscillation controlled by the PLC
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	The "Oscillation of the machine axis controlled by the PLC user program" function is requested.
Edge change 1 → 0	The "Oscillation of the machine axis controlled by the PLC user program" function is **not** requested.
Additional information	For the "Oscillation of the machine axis controlled by the PLC user program" function the following system data must be set or parameterized. • DB31, ... DBX18.4 = 1 (oscillation controlled by the PLC) • DB31, ... DBX18.5 = 1 (oscillation speed) • MD35400 $MA_SPIND_OSCILL_DES_VELO = <oscillation speed> • MD35410 $MA_SPIND_OSCILL_ACCEL = <oscillation acceleration> • DB31, ... DBX18.6 = ... → 0 → 1 → 0 ... (set direction of rotation clockwise) • DB31, ... DBX18.7 = ... → 1 → 0 → 1 ... (set direction of rotation counterclockwise) The direction of rotation when starting and the traversing duration for each direction of rotation must be specified by the PLC user program.

DB31, ... DBX18.4	Oscillation controlled by the PLC
Corresponds with	DB31, ... DBX18.5 (oscillation speed) DB31, ... DBX18.6 (setpoint direction of rotation, clockwise) DB31, ... DBX18.7 (setpoint direction of rotation, counterclockwise) MD35400 $MA_SPIND_OSCILL_DES_VELO (oscillation speed) MD35410 $MA_SPIND_OSCILL_ACCEL (acceleration when oscillating)
Additional information	• Function Manual Axes and Spindles • Function Manual Technologies

18.2.6.48 DB31, ... DBX18.5 (oscillation enable)

DB31, ... DBX18.5	Oscillation enable
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Traversing the spindle in the oscillation mode (oscillation) has been enabled by the NC.
Signal state 0	Traversing the spindle in the oscillation mode (oscillation) has **not** been enabled by the NC.
Additional information	**Note** The interface signal is irrelevant for all spindle modes **with the exception** of the oscillation mode.
Corresponds with	DB31, ... DBX16.0 - 2 (actual gear stage) DB31, ... DBX16.3 (gear stage has been changed) DB31, ... DBX18.4 (oscillation controlled by the PLC) DB31, ... DBX18.5 (oscillation enable) DB31, ... DBX18.6 (setpoint direction of rotation, clockwise) DB31, ... DBX18.7 (setpoint direction of rotation, counterclockwise) DB31, ... DBX61.4 (spindle stationary) DB31, ... DBX82.0 - 2 (set gear stage) DB31, ... DBX82.3 (change over gear stage) DB31, ... DBX83.5 (spindle in setpoint range) DB31, ... DBX84.6 (active spindle mode: oscillating mode)
Additional information	Function Manual Axes and Spindles

18.2.6.49 DB31, ... DBX18.6 (oscillation: Clockwise direction of rotation)

DB31, ... DBX18.6	Oscillation: Clockwise direction of rotation
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Clockwise oscillation direction of rotation is requested.
Signal state 0	Clockwise oscillation direction of rotation is **not** requested.
Additional information	**Note** The interface signal is irrelevant for all spindle modes **with the exception** of the oscillation mode.

DB31, ... DBX18.6	Oscillation: Clockwise direction of rotation
Corresponds to	DB31, ... DBX16.0 - 2 (actual gear stage) DB31, ... DBX16.3 (gear stage has been changed) DB31, ... DBX18.4 (oscillation controlled by the PLC) DB31, ... DBX18.5 (oscillation enable) DB31, ... DBX18.7 (setpoint direction of rotation, counterclockwise) DB31, ... DBX61.4 (spindle stationary) DB31, ... DBX82.0 - 2 (set gear stage) DB31, ... DBX82.3 (change over gear stage) DB31, ... DBX83.5 (spindle in setpoint range) DB31, ... DBX84.6 (active spindle mode: Oscillation mode)
Additional information	Function Manual Axes and Spindles

18.2.6.50 DB31, ... DBX18.7 (oscillation: Counterclockwise direction of rotation)

DB31, ... DBX18.7	Oscillation: Counterclockwise direction of rotation
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Counterclockwise oscillation direction of rotation is requested.
Signal state 0	Counterclockwise oscillation direction of rotation is **not** requested.
Additional information	**Note** The interface signal is irrelevant for all spindle modes **with the exception** of the oscillation mode.
Corresponds to	DB31, ... DBX16.0 - 2 (actual gear stage) DB31, ... DBX16.3 (gear stage has been changed) DB31, ... DBX18.4 (oscillation controlled by the PLC) DB31, ... DBX18.5 (oscillation enable) DB31, ... DBX18.6 (setpoint direction of rotation, clockwise) DB31, ... DBX61.4 (spindle stationary) DB31, ... DBX82.0 - 2 (set gear stage) DB31, ... DBX82.3 (change over gear stage) DB31, ... DBX83.5 (spindle in setpoint range) DB31, ... DBX84.6 (active spindle mode: Oscillation mode)
Additional information	Function Manual Axes and Spindles

18.2.6.51 DB31, ... DBB19 (speed override, spindle-specific)

DB31, ... DBB19	Speed override, spindle-specific
Signal flow	PLC → NC
Update	Cyclic

DB31, ... DBB19	Speed override, spindle-specific
Additional information	**Binary/Gray coding** The override factors can be specified in the binary or Gray-coded format. The control must be informed of the used format via the following machine data: MD12060 $MN_OVR_SPIN_IS_GRAY_CODE = <Coding>

Binary coding

With binary coding, the value in the interface corresponds to the override factor.

Binary code	Decimal	Override factor
0000 0000	0	0.00
0000 0001	1	0.01
0000 0010	2	0.02
0000 0011	3	0.03
000 0100	4	0.04
...	...	...
0110 0100	100	1.00
...	...	...
1100 1000	200	2.00

Gray coding

The Gray-coded values of the interface are assigned the override factors via the following machine data:

MD12070 $MN_OVR_FACTOR_SPIND_SPEED[<Switch position> - 1] = <Override factor>

Switch position	Gray code	Override factor [1)]
1	00001	0.50
2	00011	0.55
3	00010	0.60
4	00110	0.65
5	00111	0.70
6	00101	0.75
7	00100	0.80
8	01100	0.85
9	01101	0.90
10	01111	0.95
11	01110	1.00
12	01010	1.05
13	01011	1.10
14	01001	1.15
15	01000	1.20
16	11000	1.20
17	11001	1.20
18	11011	1.20
19	11010	1.20
20	11110	1.20
21	11111	1.20
22	11101	1.20

DB31, ... DBB19	Speed override, spindle-specific		
	23	11100	1.20
	24	10100	1.20
	25	10101	1.20
	26	10111	1.20
	27	10110	1.20
	28	10010	1.20
	29	10011	1.20
	30	10001	1.20
	31	10000	1.20
	Limitations • The active override factor is limited internally in the control to 2.00 or 200%. • The effective override factor is limited to 100% in the "oscillation mode" spindle mode. • The maximum possible override factor can be limited to a value less than 200% with the following machine data: MD12100 $MN_OVR_FACTOR_LIMIT_BIN		
	Notes • The spindle-specific speed override is **not effective** with the following function when cutting threads: `G63` • The spindle-specific speed override acts on the programmed values before other limits (e.g. `G26`, `LIMS`...) take effect.		
Corresponds to	MD12080 $MN_OVR_REFERENCE_IS_PROG_FEED (override reference velocity) DB31, ... DBB0 (feedrate override, axis-specific) DB31, ... DBX1.7 (feedrate/speed override active)		
Additional information	Function Manual Axes and Spindles		
1) Default values			

18.2.6.52 DB31, ... DBX20.1 (ramp-function generator disable)

DB31, ... DBX20.1	Ramp-function generator disable
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	A rapid stop with speed setpoint 0 is requested for the drive. The drive is stopped without a ramp function (regenerative braking). Feedback for the triggering of the rapid stop in the drive is via: DB31, ... DBX92.1 == 1 (ramp-function generator disable active)
Signal state 0	No rapid stop with speed setpoint 0 is requested for the drive.
Additional information	Using drive parameters p0922 and p2038, a PROFIdrive telegram compatible with the "SIMODRIVE 611 universal" interface mode must be set in the drive.
Corresponds with	DB31, ... DBX92.1 (ramp-function generator disable active) Driver parameter p0922 (IF1 PROFIdrive PZD telegram selection) Drive parameter p2038 (IF1 PROFIdrive STW/ZSW interface mode)
Additional information	NC: SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive Drive: SINAMICS S120/S150 List Manual

18.2.6.53 DB31, ... DBX21.0 - 4 (motor/drive data set: selection)

DB31, ... DBX21.0 - 4	Motor/drive data set: Selection
Signal flow	PLC → NC
Update	Cyclic
Additional information	The switchover to a new motor (MDS) and/or to drive data sets (DDS) is requested via the interface:
	Formatting Formatting the request interface, i.e. which bits are used to address the motor data sets (MDS) – and which are used to address the drive data sets (DDS) is set via the formatting interface (DB31, …DBX130.0 - 4).
	Main spindle drive For main spindle drives, the following classification applies: • MDS[0] → star operation • MDS[1] → delta operation
	Switchover time In principle it is possible to switch over drive parameter sets at any time. However, as torque jumps can occur when switching over speed controller parameters and motor speed normalization, parameters should only be switched over in stationary states, especially axis standstill. As soon as the request to switch over to another motor data set is identified, the pulses enable is reset.
	Application examples • Main spindle drive For example, it is possible to switch between operating mode 1 (star operation) and operating mode 2 (delta operation) for a main spindle drive (MSD) by switching over the motor data set. • Drive parameter set It may be necessary to switch over the drive parameter set, for: – Gear stage change – Changing the measuring circuit
Corresponds with	DB31, ... DBX93.0 - 4 (motor/drive data set: display) DB31, …DBX130.0 - 4 (motor/drive data set: formatting) DB31, ... DBX21.5 (motor has been selected)
Additional information	• Function Manual Basic Functions • SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.54 DB31, ... DBX21.5 (motor has been selected)

DB31, ... DBX21.5	Motor being selected
Signal flow	PLC → NC
Update	Cyclic

DB31, ... DBX21.5	Motor being selected
Signal state 1	The necessary electrical and/or mechanical switchover operations (e.g. contactor switchover for star-delta switchover) have been completed. • It is permissible that the axis traverses again. • The drive enables the pulses.
Signal state 0	The necessary electrical (e.g. contactor switchover for star-delta switchover) and/or mechanical switchovers have **not** been completed: • It is not permissible that the axis traverses. • The drive does not enable the pulses.
Additional information	The signal must be reset before a request to switch over to a new motor data set (MDS) and/or drive data set (DDS) (DB31, ... DBX21.0 - 4).
Corresponds with	DB31, ... DBX21.0 - 4 (motor/drive data set: request interface) DB31, ... DBX93.0 - 4 (motor/drive data set: display interface) DB31, …DBX130.0 - 4 (motor/drive data set: formatting interface)
Additional information	• Function Manual • SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.55 DB31, ... DBX21.6 (integrator disable, speed controller)

DB31, ... DBX21.6	Integrator disable, speed controller
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The integrator (I component) of the speed controller is disabled or is to be disabled (P instead of PI behavior). **Note:** If the speed controller integrator disable is activated, compensatory action may be taken in certain applications (e.g. if the integrator was already holding a load while stationary). The drive acknowledges the integrator disable to the PLC with the interface signal: DB31, ... DBX93.6 (speed controller integrator disabled).
Signal state 0	The integrator (I component) of the speed controller is enabled (PI behavior).
Additional information	If the speed controller integrator disable is activated, compensatory action may be taken, e.g. if the integrator was already holding a load while stationary. The feedback signal regarding the speed controller disable is realized from the drive using: DB31, ... DBX93.6 (integrator disable, speed controller)
Corresponds with	DB31, ... DBX93.6 (integrator disable, speed controller)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.56 DB31, ... DBX21.7 (pulse enable)

DB31, ... DBX21.7	Pulse enable
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The pulses are enabled for the drive.

DB31, ... DBX21.7	Pulse enable
Signal state 0	The pulses are disabled for the drive.
Additional information	The pulses are only enabled in the drive when the drive signals that it is ready: DB31, ... DBX93.5 == 1 (feedback signal: drive ready)
	If the pulse enable is removed during motion (e.g. emergency stop), the axis/spindle is no longer braked under control. The axis coasts to standstill.
Corresponds with	DB31, ... DBX93.5 (feedback signal: drive ready) DB31, ... DBX93.7 (feedback signal: Pulses are enabled)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.57 DB31, ... DBB22.0 (SI: SBH/SG deselection)

DB31, ... DBX22.0	SBH / SG deselection
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	SBH and SG are deselected
Signal state 0	SBH and SG are deselected
Additional information	Function Manual, SINUMERIK Safety Integrated

18.2.6.58 DB31, ... DBB22.1 (SI: SBH deselection)

DB31, ... DBX22.1	SBH deselection
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	SBH is deselected
Signal state 0	SBH is selected
Additional information	Function Manual, SINUMERIK Safety Integrated

18.2.6.59 DB31, ... DBB22.3 - 4 (SI: SG selection)

DB31, ... DBX22.3 - 4	SG selection		
Signal flow	PLC → NC		
Update	Job-controlled		
Additional information	When the SG function is active, the speed limit value is selected using the interface signals.		
	Bit 1	Bit 0	Meaning
	0	0	Speed limit value for SG1 is selected
	0	1	Speed limit value for SG2 is selected
	1	0	Speed limit value for SG3 is selected
	1	1	Speed limit value for SG4 is selected
Additional information	Function Manual, SINUMERIK Safety Integrated		

18.2.6.60 DB31, ... DBB23.0 - 2 (gear ratio selection)

DB31, ... DBX23.0 - 2	Gear ratio selection			
Signal flow	PLC → NC			
Update	Job-controlled			
Additional information	**Bit 2**	**Bit 1**	**Bit 0**	**Meaning**
	0	0	0	Gearbox stage 1 is selected
	0	0	1	Gearbox stage 2 is selected
	0	1	0	Gearbox stage 3 is selected
	0	1	1	Gearbox stage 4 is selected
	1	0	0	Gearbox stage 5 is selected
	1	0	1	Gearbox stage 6 is selected
	1	1	0	Gearbox stage 7 is selected
	1	1	1	Gearbox stage 8 is selected
Additional information	Function Manual, SINUMERIK Safety Integrated			

18.2.6.61 DB31, ... DBB23.4 (SI: SE selection)

DB31, ... DBX23.4	SE selection	
Signal flow	PLC → NC	
Update	Job-controlled	
Additional information	Selection of the safe software limit switch 1 or 2:	
	Bit 4	**Meaning**
	0	SE2 is selected
	1	SE1 is selected
Additional information	Function Manual, SINUMERIK Safety Integrated	

18.2.6.62 DB31, ... DBB24.1 (control axis)

DB31, ... DBX24.1	Control axis
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	The axis-specific "Control axis" function is set using machine data MD30460_ $MA_BASE_FUNCTION_MASK. Bit 0 = 1: The axis traverses in speed mode
Signal state 0	Bit 0 = 0: Axis control is not allowed
Additional information	SINUMERIK 840D sl List Manual, machine data

18.2.6.63 DB31, …DBX24.2 (MCS coupling: deactivate or do not permit)

DB31, …DBX24.2	MCS coupling: Deactivate or do not permit
Signal flow	PLC → NC
Update	Job-controlled
Signal state 1	Deactivate MCS coupling - or do not permit is requested
Signal state 0	Deactivate MCS coupling - or do not permit is **not** requested.
Additional information	**Notes** • The interface signal is only active if the axis is a slave axis of an MCS coupling • An active coupling is not deactivated until the relevant axes are stationary. • If, with the interface signal set, `CC_COPON` (activate coupling) is programmed, then the MCS coupling is not activated. **No** alarm is displayed.
Corresponds with	DB31, …DBX24.2 (MCS coupling: deactivate or do not permit) DB31, ... DBX24.3 (MCS coupling: activate collision protection) DB31, ... DBX66.0 (MCS coupling: collision protection active) DB31, ... DBX97.0 (MCS coupling: slave axis) DB31, … DBX97.1 (MCS coupling: coupling active) DB31, ... DBX97.2 (MCS coupling: mirroring active) DB31, ... DBX97.3 (MCS coupling: offset change)
Additional information	Function Manual Technologies

18.2.6.64 DB31, ... DBX24.3 (MCS coupling: activate collision protection)

DB31, ... DBX24.3	MCS coupling: Activate collision protection
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Activating the protection function for the axis pair is requested.
Signal state 0	Activating the protection function for the axis pair is **not** requested.
Additional information	**Activation** The protection function is activated if the interface signal for **one** of the two machine axes of the axis pair is set. **Deactivation** The protection function is deactivated if the interface signal for **both** machine axes of the axis pair is reset.
Corresponds with	DB31, …DBX24.2 (MCS coupling: deactivate or do not permit) DB31, ... DBX24.3 (MCS coupling: activate collision protection) DB31, ... DBX66.0 (MCS coupling: collision protection active) DB31, ... DBX97.0 (MCS coupling: slave axis) DB31, … DBX97.1 (MCS coupling: coupling active) DB31, ... DBX97.2 (MCS coupling: mirroring active) DB31, ... DBX97.3 (MCS coupling: offset change)
Additional information	Function Manual Technologies

18.2.6.65 DB31, ... DBX24.4 (master-slave: activate torque compensatory controller)

DB31, ... DBX24.4	Master-slave: activate torque compensatory controller
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	**Act**ivation of the torque compensatory controller is requested.
Edge change 1 → 0	**De**activation of the torque compensatory controller is requested.
Additional information	The torque compensatory controller can be activated for specific axes via the PLC interface signal DB31, ... DBX24.4. Preconditions for activation: • Input of the supplementary speed setpoint is enabled: MD37254 $MA_MS_TORQUE_CTRL_MODE[<slave axis>] = 1 • The "fine" differential speed is reached: DB31, ... DBX96.2 == 1
Corresponds with	DB31, ... DBX96.2 (master-slave: fine differential speed) DB31, ... DBX96.4 (master-slave: compensatory controller active) MD37254 $MA_MS_TORQUE_CTRL_MODE (interconnection torque compensatory controller)
Additional information	Function Manual Technologies

18.2.6.66 DB31, ... DBX24.5 (setpoint switchover: request drive control)

DB31, ... DBX24.5	Setpoint switchover: request drive control
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Accepting the drive control is requested.
Signal state 0	Accepting the drive control is **not** requested.
Additional information	**Notes** • If the request to accept the drive control is pending simultaneously in multiple axes, switchover does not occur. The drive control then remains with the axis in which it is currently active. • If no transfer requests are pending when the control powers up, drive control is assigned by the control to the first of the axes found in the machine data (MD30110 $MA_CTRLOUT_MODULE_NR[<Achse>]) with the same logical drive number.
Corresponds with	DB31, ... DBX96.5 (setpoint switchover: drive control active)
Additional information	Function Manual Axes and Spindles

18.2.6.67 DB31, ... DBX24.7 (master-slave: activate coupling)

DB31, ... DBX24.7	Master-slave: activate coupling
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	**Act**ivation of the master-slave coupling is requested.

DB31, ... DBX24.7	Master-slave: activate coupling
Edge change 1 → 0	**De**activation of the master-slave coupling is requested.
Additional information	A master-slave coupling can be activated/deactivated for specific axes via interface signal DB31, ... DBX24.7. The following conditions must be satisfied for activating/deactivating a master-slave coupling: • MD37262 = 0: The master-slave coupling is **not** continuously active. • DB31, ... DBX61.5 == 1: Master **and** slave axis must be in position control. • DB31, ... DBX61.4 == 1 Master **and** slave axis must be stationary. • DB21, ... DBX35.7 == 1: the channels of the master and slave axes must be in the "Reset" state. If a condition is not fulfilled, the coupling will **not** be activated or deactivated. An alarm is not output and the state of the coupling remains the same. If, at a later point, all conditions are fulfilled, the coupling will be activated or deactivated depending on the state of the signal.
Corresponds with	DB31, ... DBX35.7 (reset channel status) DB31, ... DBX61.4 (axis/spindle stationary (n < nmin)) DB31, ... DBX61.5 (position controller active) DB31, ... DBX96.7 (master-slave: coupling active) MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE (permanent master-slave coupling)
Additional information	Function Manual Technologies

18.2.6.68 DB31, ... DBX26.4 (enable following axis override)

DB31, ... DBX26.4	Enable following axis override
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Additional traversing motion can be overlaid on the following axis.
Signal state 0	The following axis **cannot** be overlaid and traversed.
Additional information	This signal is required for the flying synchronization of master and slave axes. As long as the "Enable following axis overlay" signal stays set to "1", the following axis selected with `EGONSYN` will traverse to synchronization in the EG coupling group. If the EG axis grouping includes modulo axes, their position values are reduced in the modulo, thereby ensuring that they approach the next possible synchronization. If, for the following axis "Enable following axis overlay" has **not**been set, then the axis it does not traverse to synchronization. Instead, the program is stopped at the `EGONSYN` block and the self-clearing alarm 16771 is issued.
Additional information	Function Manual Axes and Spindles

18.2.6.69 DB31, ... DBX28.0 (oscillation reversal from external)

DB31, ... DBX28.0	Oscillation reversal from external
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	Brake oscillation motion and move oscillation axis in the opposite direction with the oscillation axis.

DB31, ... DBX28.0	Oscillation reversal from external
Edge change 1 → 0	Continue oscillation without interruption
Additional information	Function Manual Axes and Spindles

18.2.6.70 DB31, ... DBX28.1 (PLC-controlled axis: Reset)

DB31, ... DBX28.1	PLC-controlled axis: Reset
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	PLC-controlled axis: Reset is requested
Edge change 1 → 0	No effect.
Additional information	NC responses: • The axis is transitioned into the "Single axis in Reset" status: $AA_SNGLAX_STAT == 1 • Stopped sequences are canceled. • Internal axis states are reset. • Axis specific, machine data active at reset become active. • Execution is confirmed: – DB31, ... DBX63.0 (reset executed) == 1 – DB31, ... DBX63.2 (axis stop active) == 0 • Result: The axis is in the reset state. **Note** In conjunction with a channel reset, no axial machine data is active for axes controlled from the PLC.
Corresponds with	DB31, ... DBX28.7 (PLC-controlled axis) DB31, ... DBX63.1 (reset executed) DB31, ... DBX63.2 (axis stop active) System variable: $AA_SNGLAX_STAT (status of the single axis) OPI variable: aaSnglAxStat
Additional information	Function Manual Axes and Spindles

18.2.6.71 DB31, ... DBX28.2 (PLC-controlled axis: Resume)

DB31, ... DBX28.2	PLC-controlled axis: Resume
Signal flow	PLC → NC
Update	Cyclic
Edge change 0 → 1	PLC-controlled axis: Resume uninterrupted traversing motion is requested
Edge change 1 → 0	No effect.

DB31, ... DBX28.2	PLC-controlled axis: Resume
Additional information	Using the signal, the traversing motion of an axis - controlled from the PLC - which is in the "Single axis is interrupted" status ($AA_SNGLAX_STAT = 3), is resumed. NC responses: • Checks whether, for the axis, an axis-specific alarm with delete criterion "CANCELCLEAR" or "NCSTARTCLEAR" is present? If yes, then this is deleted. • Checks whether axis motion can be resumed? If yes, then the axis makes the transition into the "Single axis is active" state. $AA_SNGLAX_STAT == 4 • Execution is confirmed: – DB31,... DBX60.6 (position reached with exact stop coarse) == 0 – DB31, ... DBX60.7 (position reached with exact stop fine) == 0 – DB31, ... DBX63.2 (axis stop active) == 0 – DB31, ... DBX64.6 / 7 (traversing command minus/plus) == 1 • Result: Traversing motion of the axis is resumed. **Note** The signal is ignored in the following cases: • The axis is not controlled by the PLC. • The axis is not in the "Single axis is interrupted" status. • It is not permissible that axis traversing motion resumes due to an active alarm.
Corresponds with	DB31, ... DBX28.6 (PLC-controlled axis: Stop along braking ramp) DB31, ... DBX28.7 (PLC-controlled axis) DB31, ... DBX60.6 (position reached with exact stop coarse) DB31, ... DBX60.7 (position reached with exact stop fine) DB31, ... DBX63.2 (axis stop active) DB31, ... DBX64.6 (traversing command "minus") DB31, ... DBX64.7 (traversing command "plus") System variable: $AA_SNGLAX_STAT (status of the single axis) OPI variable: aaSnglAxStat
Additional information	Function Manual Axes and Spindles

18.2.6.72 DB31, ... DBX28.3 (set reversal point)

DB31, ... DBX28.3	Set reversal point
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Reversal point 2
Signal state 0	Reversal point 1
Additional information	Function Manual Technologies

18.2.6.73 DB31, ... DBX28.4 (change reversal point)

DB31, …DBX28.4	Change reversal point
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	The reversal point position can be changed by manually traversing the axis. In conjunction with DB31, ...DBX28.0 (oscillation reversal from external): The position at which the axis is braked after "oscillation reversal from external", should be the new reversal point.
Signal state 0	The reversal point position **cannot** be changed by manually traversing the axis. In conjunction with DB31, ...DBX28.0 (oscillation reversal from external): No change of the reversal point position.
Additional information	The braking position of the axis can be accepted as the new reversal position by means of the signal DB31, ... DBX28.4 (change reversal position). The signal DB31, ... DBX28.3 (set reversal point) is ignored, but rather the change affects the last initiated "Oscillation reversal from external" command. **Note** For the axis, it is not permissible that changing the reversal points via traversing keys or handwheel is active. In this case, display alarm 20081 "Braking position cannot be used as reversal point - handwheel active "is signaled.
Corresponds with	DB31, ... DBX28.0 (oscillation reversal from external) DB31, ... DBX28.3 (set reversal point)
Additional information	Function Manual Technologies

18.2.6.74 DB31, ... DBX28.5 (PLC-controlled axis: Stop at the next reversal point)

DB31, ... DBX28.5	PLC-controlled axis: Stop at the next reversal point
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Interrupt oscillation motion of the PLC-controlled axis at the next reversal point.
Signal state 0	There is no request to interrupt the oscillation motion of the PLC-controlled axis at the next reversal point.
Corresponds with	DB31, ... DBX28.6 (PLC-controlled axis: Stop along braking ramp) DB31, ... DBX28.7 (PLC-controlled axis)
Additional information	Function Manual Technologies

18.2.6.75 DB31, ... DBX28.6 (PLC-controlled axis: Stop along braking ramp)

DB31, ... DBX28.6	PLC-controlled axis: Stop along braking ramp
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	PLC-controlled axis: Braking the axis along a braking ramp is requested.
Signal state 0	PLC-controlled axis: Braking the axis along a braking ramp is **not** requested.

DB31, ... DBX28.6	PLC-controlled axis: Stop along braking ramp
Additional information	Using the signal, the traversing motion of an axis - controlled from the PLC - can be stopped. NC responses: • The axis is braked down to standstill along its deceleration characteristic. • The axis is transitioned into the "Single axis is interrupted" status: $AA_SNGLAX_STAT == 3 • Execution is confirmed: – DB31,... DBX60.6 (position reached with exact stop coarse) == 1 – DB31, ... DBX60.7 (position reached with exact stop fine) == 1 – DB31, ... DBX63.2 (axis stop active) == 0 – DB31, ... DBX64.6 / 7 ("Plus" / "Minus" traversing command) == 0 • Result: Axis is stopped. **Note** Following axis movements can only be suppressed when the leading axis stops. Retraction motion triggered by the "Extended stop and retract" function cannot be stopped.
Corresponds with	DB31, ... DBX28.5 (PLC-controlled axis: Stop at the next reversal point) DB31, ... DBX28.7 (PLC-controlled axis) DB31, ... DBX60.6 (position reached with exact stop coarse) DB31, ... DBX60.7 (position reached with exact stop fine) DB31, ... DBX63.2 (axis stop active) DB31, ... DBX64.6 (traversing command "minus") DB31, ... DBX64.7 (traversing command "plus") System variable: $AA_SNGLAX_STAT (status of the single axis) OPI variable: aaSnglAxStat
Additional information	Function Manual Axes and Spindles

18.2.6.76 DB31, ... DBX28.7 (request, PLC-controlled axis)

DB31, ... DBX28.7	Request PLC-controlled axis
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Axis control should be relinquished to the PLC.
Signal state 0	There is no request that axis control is transferred to the PLC.
Additional information	NC responses: • Checks whether the axis is a main run axis or a neutral axis. • Checks whether an additional axis may be controlled from the PLC. • Transfer is confirmed: – DB31, ... DBX63.1 (PLC controls axis) == 1 – $AA_SNGLAX_STAT == 1 • Result: PLC controls axis.

DB31, ... DBX28.7	Request PLC-controlled axis
Corresponds with	DB31, ... DBX63.1 (PLC-controlled axis) System variable: $AA_SNGLAX_STAT (status of the single axis) OPI variable: aaSnglAxStat
Additional information	Function Manual Axes and Spindles

18.2.6.77 DB31, ... DBX31.5 (disable synchronization)

DB31, ... DBX31.5	Disable synchronization
Signal flow	PLC → NC
Update	Cyclic
Signal state 1	Synchronous spindle coupling: The PLC blocks synchronization motion for the following spindle specified by offset programming.
Signal state 0	Synchronous spindle coupling: The PLC does **not** block synchronization motion for the following spindle specified by offset programming.
Additional information	When the main run advances to a block with the part program statement `COUPON(<FS>,<LS>,<offset>)`, then interface signal DB31, ... DBX31.5 (block synchronization) is evaluated for the following spindle: • For DB31, ... DBX31.5 == 1, only continuous velocity synchronous operation is established. The following spindle does not execute any additional movement. The coupling then responds analogously to a programmed `COUPON(<FS>,<LS>)`. • For DB31, ... DBX31.5 == 0, the axis only moves through the position offset.
Corresponds with	DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation) DB31, ... DBX98.1 (coarse synchronous operation) DB31, ... DBX98.0 (fine synchronous operation)
Additional information	Function Manual Axes and Spindles

18.2.6.78 DB31, ... DBX34.0 - 1 (setpoint limitation)

DB31, ... DBX34.0 - 1	Setpoint limitation		
Signal flow	PLC → NC		
Update	Cyclic		
Further information	With the interface signals, one of the four setpoint speed limits parameterized in MD36933 $MA_SAFE_DES_VELO_LIMIT[0 ... 3] can be activated, irrespective of the active SG stage.		
	Bit 1	**Bit 0**	**Meaning:** Activation of
	0	0	MD36933 $MA_SAFE_DES_VELO_LIMIT[0]
	0	1	MD36933 $MA_SAFE_DES_VELO_LIMIT[1]
	1	0	MD36933 $MA_SAFE_DES_VELO_LIMIT[2]
	1	1	MD36933 $MA_SAFE_DES_VELO_LIMIT[3]
Corresponds with	MD36933 $MA_SAFE_DES_VELO_LIMIT[0 ... 3]		
Additional information	Function Manual, Safety Integrated		

18.2.6.79 DB31, ... DBX60.0 (spindle / rotary axis)

DB31, ... DBX60.0	Spindle
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The machine axis is operated as a spindle in the following modes: • Control mode • Oscillation mode • Positioning mode • Tapping with compensating chuck • Synchronous mode The interface signals to the axis (DB31, ... DBB12 - DBB15) and from the axis (DB31, ... DBB74 - DBB81) are **invalid**. The interface signals to the spindle (DB31, ... DBB16 - DBB19) and from the spindle (DB31, ... DBB82 - DBB91) are **valid**.
Signal state 0	The machine axis is operated as rotary axis. The interface signals to the axis (DB31, ... DBB12 - DBB15) and from the axis (DB31, ... DBB74 - DBB81) are **valid**. The interface signals to the spindle (DB31, ... DBB16 - DBB19) and from the spindle (DB31, ... DBB82 - DBB91) are **invalid**.
Additional information	If a machine axis is operated alternating as spindle or rotary axis, using the signal it is possible to identify which mode is currently active. • Turning machine: Spindle ↔ C axis • Milling machine: Spindle ↔ rotary axis for rigid tapping
Corresponds with	DB31, ... DBB82 - DBB91
Additional information	Function Manual Axes and Spindles

18.2.6.80 DB31, ... DBX60.1 (NCU link axis active)

DB31, ... DBX60.1	NCU link: Axis active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis is active as NCU link axis.
Signal state 0	The axis is **not** active as NCU link axis.
Additional information	The signal is irrelevant for a system with an NCU.
Additional information	• Function Manual Basic Functions • NCU Equipment Manual

18.2.6.81 DB31, ... DBX60.2 (encoder limit frequency exceeded 1)

DB31, ... DBX60.2	Encoder limit frequency exceeded, measuring system 1
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The encoder frequency limit of measuring system 2 has been exceeded ⇒ • DB31, ...DBX60.4 = 0 (referenced/synchronized position measuring system 1) • DB31, ...DBX61.5 = 0 (position controller active) • DB31, ...DBX61.5 = 1 (speed controller active) only for a spindle • Axis is stopped (fast stop)
Signal state 0	The encoder frequency limit of measuring system 2 has not been exceeded, or after it was exceeded, the encoder frequency limit for encoder resynchronization was fallen below again: MD36302 $MA_ENC_FREQ_LIMIT_LOW
Additional information	The encoder frequency limit to be monitored is set using the following machine data: MD36300 $MA_ENC_FREQ_LIMIT The monitoring of the encoder frequency limit always refers to the active measuring system selected in the NC/PLC interface: DB31, ... DBX1.5 / 6 (position measuring system 1 / 2)
Corresponds with	DB31, ... DBX1.5 (position measuring system 1) DB31, ... DBX1.6 (position measuring system 2) MD36300 $MA_ENC_FREQ_LIMIT (encoder frequency limit) MD36302 $MA_ENC_FREQ_LIMIT_LOW (encoder frequency limit for encoder resynchronization)
Additional information	Function Manual Monitoring and Compensating

18.2.6.82 DB31, ... DBX60.3 (encoder frequency limit exceeded, measuring system 2)

DB31, ... DBX60.3	Encoder frequency limit exceeded, measuring system 2
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The encoder frequency limit of measuring system 2 has been exceeded ⇒ • DB31, ...DBX60.5 = 0 (referenced/synchronized position measuring system 2) • DB31, ...DBX61.5 = 0 (position controller active) • DB31, ...DBX61.5 = 1 (speed controller active) only for a spindle • Axis is stopped (fast stop)
Signal state 0	The encoder frequency limit of measuring system 2 has not been exceeded, or after it was exceeded, the encoder frequency limit for encoder resynchronization was fallen below again: MD36302 $MA_ENC_FREQ_LIMIT_LOW
Additional information	The encoder frequency limit to be monitored is set using the following machine data: MD36300 $MA_ENC_FREQ_LIMIT The monitoring of the encoder frequency limit always refers to the active measuring system selected in the NC/PLC interface: DB31, ... DBX1.5 / 6 (position measuring system 1 / 2)

DB31, ... DBX60.3	Encoder frequency limit exceeded, measuring system 2
Corresponds with	DB31, ... DBX1.5 (position measuring system 1) DB31, ... DBX1.6 (position measuring system 2) MD36300 $MA_ENC_FREQ_LIMIT (encoder frequency limit) MD36302 $MA_ENC_FREQ_LIMIT_LOW (encoder frequency limit for encoder resynchronization)
Additional information	Function Manual Monitoring and Compensating

18.2.6.83 DB31, ... DBX60.4 (referenced/synchronized 1)

DB31, ... DBX60.4	Referenced / synchronized 1
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Position measuring system 1 of the machine axis is referenced/synchronized.
Signal state 0	Position measuring system 1 of the machine axis is **not** referenced/synchronized.
Additional information	**Axis** After successful referencing or synchronizing, the interface signal is set **Spindle** The interface signal is set at the latest after one spindle revolution (360 degrees), if the zero mark was passed or the contactless proximity switch has responded.
Corresponds with	DB31, ... DBX1.6 (position measuring system 2) DB31, ... DBX2.4 - 7 (reference point value 1 to 4) DB31, ... DBX12.7 (reference point approach deceleration) DB31, ... DBX71.4 (position restored encoder 1) DB31, ... DBX71.5 (position restored encoder 2) MD34102 $MA_REFP_SYNC_ENCS (measuring system calibration)
Additional information	Function Manual Axes and Spindles

18.2.6.84 DB31, ... DBX60.5 (referenced/synchronized 2)

DB31, ... DBX60.5	Referenced/synchronized 2
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Position measuring system 2 of the machine axis is referenced/synchronized.
Signal state 0	Position measuring system 2 of the machine axis is **not** referenced/synchronized.
Additional information	**Axis** After successful referencing or synchronizing, the interface signal is set **Spindle** The interface signal is set at the latest after one spindle revolution (360 degrees), if the zero mark was passed or the contactless proximity switch has responded.

DB31, ... DBX60.5	Referenced/synchronized 2
Corresponds with	DB31, ... DBX1.6 (position measuring system 2) DB31, ... DBX2.4 - 7 (reference point value 1 to 4) DB31, ... DBX12.7 (reference point approach deceleration) DB31, ... DBX71.4 (position restored encoder 1) DB31, ... DBX71.5 (position restored encoder 2) MD34102 $MA_REFP_SYNC_ENCS (measuring system calibration)
Additional information	Function Manual Axes and Spindles

18.2.6.85 DB31,... DBX60.6 (position reached with exact stop coarse)

DB31, ... DBX60.6	Position reached with exact stop coarse
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	At least one of the following states applies: • The axis is not traversing, and the actual axis position is within the parameterized exact stop limit MD36000 $MA_STOP_LIMIT_COARSE • The control system is in the "Reset" state • The axis was last programmed as positioning axis or positioning spindle. • Path motion was terminated with an NC stop. • The spindle is in closed-loop position control (SPCON/SPOS) and is stationary • The axis was switched over from closed-loop speed to closed-loop position control
Signal state 0	At least one of the following states applies: • The axis is traversing • Traversing motion is pending for the axis • The actual axis position lies outside the parameterized exact stop limit MD36000 $MA_STOP_LIMIT_COARSE • The spindle is in closed-loop speed control (SPCOF/SPOSA) • The axis is in the "Follow-up mode" state • The axis is in the "Park" state • The axis was switched over from closed-loop position to closed-loop speed control
Corresponds with	MD36000 $MA_STOP_LIMIT_COARSE (exact stop coarse)

18.2.6.86 DB31, ... DBX60.7 (position reached with exact stop fine)

DB31, ... DBX60.7	Position reached with exact stop fine
Signal flow	NC → PLC
Update	Cyclic

DB31, ... DBX60.7	Position reached with exact stop fine
Signal state 1	At least one of the following states applies: • The axis is not traversing, and the actual axis position is within the parameterized exact stop limit MD36010 $MA_STOP_LIMIT_FINE • The control system is in the "Reset" state • The axis was last programmed as positioning axis or positioning spindle. • Path motion was terminated with an NC stop. • The spindle is in closed-loop position control (SPCON/SPOS) and is stationary • The axis was switched over from closed-loop speed to closed-loop position control
Signal state 0	At least one of the following states applies: • The axis is traversing • Traversing motion is pending for the axis • The actual axis position lies outside the parameterized exact stop limit MD36010 $MA_STOP_LIMIT_FINE • The spindle is in closed-loop speed control (SPCOF/SPOSA) • The axis is in the "Follow-up mode" state • The axis is in the "Park" state • The axis was switched over from closed-loop position to closed-loop speed control
Corresponds with	MD36010 $MA_STOP_LIMIT_FINE (exact stop fine)

18.2.6.87 DB31, ... DBX61.0 (drive test travel request)

DB31, ... DBX61.0	Drive test travel request
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The travel enable for the drive test is requested, as the travel conditions for the axis are satisfied: • The mechanical brake of the axis involved was previously released. • Axis disable is not active DB31, ... DBX1.3 == 0 The feedback signal is realized using: DB31, ... DBX1.0 == 1
Signal state 0	The travel enable for the drive test is not requested
Corresponds with	DB31, ... DBX1.0 (drive test travel enable)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.88 DB31, ... DBX61.1 (axis-specific alarm)

DB31, ... DBX61.1	Axis-specific alarm
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	An axis-specific alarm is output.
Signal state 0	No alarm

DB31, ... DBX61.1	Axis-specific alarm
Additional information	Responses: • The axis is braked down to standstill along its deceleration characteristic. • Status of the axis is transitioned into "Single axis alarm is active": $AA_SNGLAX_STAT == 5
Corresponds with	DB31, ... DBX28.7 (request, PLC-controlled axis) System variable: $AA_SNGLAX_STAT (status of the single axis) OPI variable: aaSnglAxStat
Additional information	Function Manual Axes and Spindles

18.2.6.89 DB31, ... DBX61.2 (axis ready)

DB31, ... DBX61.2	NCU link: Axis ready
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis is ready.
Signal state 0	The axis is **not** ready.
Additional information	The signal is processed on the NCU to which the drive of the axis connected. The axis is not ready, if, on this NCU an alarm with system response "NC ..." or "Mode group..." or "Channel not ready" is displayed, which involves this specific axis.
Additional information	Function Manual Basic Functions

18.2.6.90 DB31, ... DBX61.3 (follow-up active)

DB31, ... DBX61.3	Follow-up active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The follow-up mode for the axis/spindle is active. Possible causes: • The controller enable for the drive has been withdrawn. – DB31, ...DBX2.1 == 0 (controller enable) – in the control for faults; see under "Additional information" • Follow-up mode was selected: – DB31, ... DBX1.4 == 1 (follow-up mode). – in the control, e.g. when withdrawing the controller enable for a traversing axis During the follow-up mode: position setpoint = actual position value **Note** The standstill and clamping monitoring are not active.
Signal state 0	The follow-up mode for the axis/spindle is not active. The standstill and clamping monitoring are active.

DB31, ... DBX61.3	Follow-up active
Additional information	**Note** In the "Hold" state, the "Follow-up mode active" signal is not set. **Notice** Delete distance-to-go is executed in the control for the following state transitions: • From the state "Follow-up" into the state "Hold" (DB31, ... DBX1.4 = 1 → 0 (follow-up mode)) • From the state "Follow-up" into the state "Position control" (DB31, ...DBX2.1 = 1 (controller enable))
Corresponds with	DB31, ... DBX2.1 (controller enable) DB31, ... DBX1.4 (follow-up mode)
Additional information	Diagnostics Manual

18.2.6.91 DB31, ... DBX61.4 (axis/spindle stationary (n < nmin))

DB31, ... DBX61.4	Axis/spindle stationary ($n < n_{min}$) (status)
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis/spindle is stationary This means that the actual velocity of the axis or the actual speed of the spindle is less than or equal to the parameterized limit value: MD36060 $MA_STANDSTILL_VELO_TOL
Signal state 0	The axis/spindle is not stationary This means that the actual velocity of the axis or the actual speed of the spindle is greater than the parameterized limit value: MD36060 $MA_STANDSTILL_VELO_TOL If a travel command is present, e.g. for a spindle, then the signal is always = 0 - even if the actual speed lies below that specified in MD36060. If the interface signal: DB31, ... DBX61.4 (axis/spindle stationary) is signaled and there is no closed-loop position control active for the spindle, then at the user interface, an actual speed of zero is displayed and with the system variable $AA_S[n] zero is read.
Additional information	The signal is always 0 if, for the axis/spindle, a travel command is present (DB31, ...DBX64.6 or .7) - or if the actual velocity of the axis or actual speed of the spindle is less than or equal to the parameterized limit value.
Corresponds with	MD36060 $MA_STANDSTILL_VELO_TOL (maximum velocity/speed for signal "Axis/spindle stationary")

18.2.6.92 DB31, ... DBX61.5 (position controller active)

DB31, ... DBX61.5	Position controller active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The position controller is active, i.e. the position control loop of the axis/spindle is closed.

DB31, ... DBX61.5	Position controller active
Signal state 0	The position controller is not active, i.e. the position control loop of the axis/spindle is not closed. The signal is reset if the controller enable (DB31, ... DBX2.1 == 0) is reset by the PLC user program, or as a result of an internal fault.
Additional information	**Spindle without position control** For a spindle without position control, the signal is always 0. **Spindle with position control** For spindles with the option of position control, after activating the position control, e.g.using `SPCON` or `M70`, then the signal is handled the same as for a position-controlled axis. **Vertical axis** For a vertical axis, the holding brake should be activated as soon as the position control is no longer active (DB31, ... DBX61.5 == 0). **Simulation axis** The output of the signal can also be parameterized for a simulation axis: MD30350 $MA_SIMU_AX_VDI_OUTPUT = 1
Corresponds with	DB31, ... DBX2.1 (controller enable) DB31, ... DBX1.4 (follow-up mode) DB31, ... DBX1.5 and .6 (position measuring systems 1 and 2)
Additional information	Diagnostics Manual

18.2.6.93 DB31, ... DBX61.6 (speed controller active)

DB31, ... DBX61.6	Speed controller active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The speed controller is active, i.e. the speed control loop of the axis/spindle is closed.
Signal state 0	The speed controller is not active, i.e. the speed control loop of the axis is not closed. The speed controller output is cleared.
Additional information	**Spindle without position control** If the spindle is not under position control, the signal can be used as a feedback signal for signal DB31, ... DBX2.1 (controller enable). **Simulation axis** The output of the signal can also be parameterized for a simulation axis: MD30350 $MA_SIMU_AX_VDI_OUTPUT = 1
Corresponds with	DB31, ... DBX61.5 (position controller active)

18.2.6.94 DB31, ... DBX61.7 (current controller active)

DB31, ... DBX61.7	Current controller active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The current controller is active, i.e. the current control loop of the axis/spindle is closed.

DB31, ... DBX61.7	Current controller active
Signal state 0	The current controller is not active, i.e. the current control loop of the axis is not closed. The current controller output, including the injection variables on the control voltage, is cleared.
Corresponds with	DB31, ... DBX61.5 (position controller active) DB31, ... DBX61.6 (speed controller active)

18.2.6.95 DB31, ... DBX62.0 (software cams active)

DB31, ... DBX62.0	Software cams active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The output of the minus and plus cam signals of an axis to the PLC interface is active.
Signal state 0	The output of the minus and plus cam signals of an axis to the PLC interface is **not** active.
Corresponds with	DB10 DBX110.0 - 113.7 (software cams: minus cam signal 1 to 32) DB10 DBX114.0 - 117.7 (software cams: plus cam signal 1 to 32) DB31, ... DBX2.0 (software cams: activation)
Additional information	Function Manual Axes and Spindles

18.2.6.96 DB31, ... DBX62.1 (handwheel override active)

DB31, ... DBX62.1	Handwheel override active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The "Handwheel override in AUTOMATIC mode" function is active for the programmed positioning axis (`FDA[<axis>]`).
Signal state 0	The "Handwheel override in automatic mode" function is **not** active for the programmed positioning axis.
Additional information	For the positioning axis, the handwheel pulses either act as path definition (for `FDA[<axis>]=0`) - or as velocity override (for `FDA[<axis>]>0`) on the programmed axis feed. In the following cases, the override is inactive: • The positioning axis has reached the programmed target position. • The distance-to-go has been deleted. • Reset was initiated. **Note** SINUMERIK 840D sl: The interface signal will also be set if "Handwheel override in automatic mode" is active with a concurrent positioning axis with FC18.
Corresponds with	DB31, ... DBX2.2 (delete distance-to-go)
Additional information	Function Manual Axes and Spindles

18.2.6.97 DB31, ... DBX62.2 (revolutional feedrate active)

DB31, ... DBX62.2	Revolutional feedrate active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Revolutional feedrate (G95) is active.
Signal state 0	Revolutional feedrate (G95) is **not** active.
Further information	The interface signal indicates that in JOG mode or AUTOMATIC mode, the axes traverse as a positioning axis with revolutional feedrate.
Corresponds with	SD41100 $SN_JOG_REV_IS_ACTIVE (JOG: Revolutional/linear feedrate) SD41120 $SN_JOG_REV_SET_VELO (revolutional feedrate of the axes JOG) SD42600 $SC_JOG_FEED_PER_REV_SOURCE (control of the revolutional feedrate in JOG) SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE (revolutional feedrate for positioning axes/spindles) MD32040 $MA_JOG_REV_VELO_RAPID (revolutional feed rate for JOG with rapid traverse override) MD32050 $MA_JOG_REV_VELO (revolutional feed rate for JOG mode)
Additional information	Function Manual Axes and Spindles

18.2.6.98 DB31, ... DBX62.3 (measurement active)

DB31, ... DBX62.3	Measurement active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	A measuring function is active.
Signal state 0	A measuring function is **not** active.
Additional information	Interface signal DBX31, ... DBX62.3 shows the instantaneous measurement status of the axis, and can be evaluated for all measuring functions.
Additional information	Function Manual Technologies

18.2.6.99 DB31, ... DBX62.4 (activate travel to fixed stop)

DB31, ...DBX62.4	Activate travel to fixed stop
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Request to enable travel to fixed stop.
Signal state 0	No request
Additional information	The feedback signal is realized via DB31, ... DBX3.1 (enable travel to fixed stop)
Corresponds with	DB31, ... DBX3.1 (enable travel to fixed stop)
Additional information	Function Manual Axes and Spindles

18.2.6.100 DB31, ... DBX62.5 (fixed stop reached)

DB31, ... DBX62.5	Fixed stop reached
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Fixed stop has been reached.
Signal state 0	Fixed stop has not been reached.
Additional information	**Application example** In order that a programmable clamping torque can be specified, for analog drives, the signal is used to switch the actuator drive from closed-speed into closed-loop current or torque controlled operation.
Corresponds with	DB31, ... DBX1.1 (acknowledge fixed stop reached). DB31, ... DBX1.2 (fixed stop sensor) DB31, ... DBX3.1 (enable travel to fixed stop) DB31, ... DBX62.4 (activate travel to fixed stop) DB31, ... DBX62.5 (fixed stop reached)
Additional information	Function Manual Axes and Spindles

18.2.6.101 DB31, ... DBX62.7 (axis container rotation active)

DB31, ... DBX62.7	Axis container: Rotation active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	An axis container rotation is active for the axis.
Signal state 0	An axis container rotation is **not** active for the axis.
Additional information	Function Manual Basic Functions

18.2.6.102 DB31, ... DBX63.0 (reset executed)

DB31, ... DBX63.0	Reset executed
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The PLC-controlled axis is in the reset state.
Signal state 0	The PLC-controlled axis is **not** in the reset state.
Additional information	Reset status: • The machine data of the axis is reloaded. • The axis status is at "Single axis in reset": $AA_SNGLAX_STAT == 1 • DB31, ... DBX63.2 (axis stop active) == 0

DB31, ... DBX63.0	Reset executed
Corresponds with	DB31, ... DBX28.1 (PLC-controlled axis: Reset) DB31, ... DBX63.2 (axis stop active) System variable: $AA_SNGLAX_STAT (status of the single axis) OPI variable: aaSnglAxStat
Additional information	Function Manual Axes and Spindles

18.2.6.103 DB31, ... DBX63.1 (PLC-controlled axis)

DB31, ... DBX63.1	PLC-controlled axis
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Axis control is transferred to the PLC.
Signal state 0	The axis is controlled by the NC.
Corresponds with	DB31, ... DBX28.7 (request, PLC-controlled axis) System variable: $AA_SNGLAX_STAT (status of the single axis)
Additional information	Function Manual Axes and Spindles

18.2.6.104 DB31, ... DBX63.2 (axis stop active)

DB31, …DBX63.2	Axis stop active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The PLC-controlled axis is stopped as a result of DB31, ... DBX28.6 = 1.
Signal state 0	The PLC-controlled axis is stopped.
Additional information	After stopping, the axis is in the "Single axis is interrupted" status. $AA_SNGLAX_STAT == 3
Corresponds with	DB31, ... DBX28.6 (PLC-controlled axis: Stop along braking ramp) DB31, ... DBX28.7 (PLC-controlled axis) DB31, ... DBX60.6 (position reached with exact stop coarse) DB31, ... DBX60.7 (position reached with exact stop fine) System variable: $AA_SNGLAX_STAT (status of the single axis) OPI variable: aaSnglAxStat
Additional information	Function Manual Axes and Spindles

18.2.6.105 DB31, ... DBX63.3 (axis/spindle disable active)

DB31, ... DBX63.3	Axis/spindle disable active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Axis/spindle disable is active.

DB31, ... DBX63.3	Axis/spindle disable active
Signal state 0	The axis/spindle disable is **not** active.
Additional information	An axis / spindle disable is requested via DB31, ... DBX1.3 = 1
Corresponds with	DB31, ... DBX1.3 (axis/spindle disable) DB31, ... DBX14.1 (activate program test) DB21, ... DBX1.7 (activate program test) DB21, ... DBX33.7 (program test active)
Additional information	Response for synchronous operation: Function Manual Axes and Spindles

18.2.6.106 DB31, ... DBX64.0 - 2 (handwheel active)

DB31, ... DBX64.0 - 2	Handwheel active			
Signal flow	NC → PLC			
Update	Cyclic			
Additional information	The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.			
	Bit coded: **maximum 3** handwheels			
	Bit 2	**Bit 1**	**Bit 0**	**Number of the assigned handwheel**
	0	0	0	No handwheel assignment
	0	0	1	1
	0	1	0	2
	1	0	0	3
	Binary coded: **maximum 6** handwheels			
	Bit 2	**Bit 1**	**Bit 0**	**Number of the assigned handwheel**
	0	0	0	No handwheel assignment
	0	0	1	1
	0	1	0	2
	0	1	1	3
	1	0	0	4
	1	0	1	5
	1	1	0	6
	If the assignment is active, the machine axis can be traversed with the handwheel in JOG mode or a DRF offset can be generated in AUTOMATIC or MDI mode. **Note** At any one time, the machine axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, then the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".			
Corresponds with	DB31, ... DBX4.0 - 2 (activate handwheel) MD11324 $MN_HANDWH_VDI_REPRESENTATION (representation of the handwheel number at the VDI interface)			
Additional information	Function Manual Axes and Spindles			

18.2.6.107 DB31, ... DBX64.4 - 5 ("Plus" / "Minus" traversing request)

DB31, ... DBX64.4 - 5	Traversing request "Plus" / "Minus"	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	There is a traversing request for the machine axis.	
Signal state 0	There is **no** traversing request for the machine axis.	
Additional information	Depending on the mode selected, the traversing request is initiated in different ways: • JOG mode "Plus" or "Minus" traversing key • REF mode Traversing key that initiates traversing motion in the direction of the reference point. • AUTOMATIC or MDI mode A program block with a traversing operation is executed for a geometry axis.	
	Bit 4	Traversing command "Minus", travel request in the negative axis direction
	Bit 5	Traversing command "Plus" travel request in the positive axis direction
Corresponds with	DB31, ... DBX4.6 - 7 (traversing keys "Plus" / "Minus") DB31, ... DBX64.6 - 7 ("Plus" / "Minus" traversing command)	
Additional information	Function Manual Axes and Spindles	

18.2.6.108 DB31, ... DBX64.6 - 7 ("Plus" / "Minus" traversing command)

DB31, ... DBX64.6 - 7	Traversing command "Plus" / "Minus"	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	A travel command is pending for the machine axis	
Signal state 0	**No** travel command is pending for the machine axis	
Additional information	The travel command is output depending on MD17900 $MN_VDI_FUNCTION_MASK, bit 0.	
	Bit 6	Travel command "Minus", for traversing in the negative axis direction
	Bit 7	Travel command "Plus", for traversing in the positive axis direction
	Application example Releasing the axis clamp when the traversing command is identified. **Note** For axes where clamping is not released until a drive command is detected, the continuous-path mode (G64) is not possible.	
Corresponds with	DB31, ... DBX4.6 - 7 (traversing keys "Plus" / "Minus") DB31, ... DBX64.4 - 5 (traversing request "Plus" / "Minus") MD17900 $MN_VDI_FUNCTION_MASK (setting for VDI signals)	
Additional information	Function Manual Axes and Spindles	

18.2.6.109 DB31, ... DBX65.0 - 6 (active machine function)

DB31, ... DBX65.0 - 6	Active machine function	
Signal flow	NC → PLC	
Update	Cyclic	
Signal state 1	The machine function to manually traverse the machine axis is active.	
Signal state 0	The machine function to manually traverse the machine axis is **not** active .	
Additional information	There is a signal for every machine function to manually traverse the machine axis in the JOG mode.	
	Bit 0	INC1
	Bit 1	INC10
	Bit 2	INC100
	Bit 3	INC1000
	Bit 4	INC10000
	Bit 5	INCvar
	Bit 6	Continuous manual travel
	Note Depending on the machine function, the response when actuating the traversing key or the handwheel differs.	
Corresponds with	DB31, ... DBX5.0 - 6 (machine function request)	
Additional information	Function Manual Axes and Spindles	

18.2.6.110 DB31, ... DBX66.0 (MCS coupling: collision protection active)

DB31, …DBX66.0	MCS coupling: Collision protection active
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Collision protection is active.
Signal state 0	Collision protection is **not** active.
Additional information	**Note**: The interface signal must be activated with: MD63543 $MD_CC_PROTECT_OPTIONS, bit 7 = 1
Corresponds with	DB31, …DBX24.2 (MCS coupling: deactivate or do not permit) DB31, ... DBX24.3 (MCS coupling: activate collision protection) DB31, ... DBX66.0 (MCS coupling: collision protection active) DB31, ... DBX97.0 (MCS coupling: slave axis) DB31, … DBX97.1 (MCS coupling: coupling active) DB31, ... DBX97.2 (MCS coupling: mirroring active) DB31, ... DBX97.3 (MCS coupling: offset change)
Additional information	Function Manual Technologies

18.2.6.111 DB31, ... DBX67.0 (handwheel direction of rotation inversion active)

DB31, ...DBX67.0	Handwheel direction of rotation inversion active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The inversion of the direction of rotation of the handwheel assigned to the machine axis is active.
Signal state 0	The inversion of the direction of rotation of the handwheel assigned to the machine axis is **not** active.
Corresponds with	DB31, ... DBX7.0 (invert handwheel direction of rotation)
Additional information	Function Manual Axes and Spindles

18.2.6.112 DB31, ... DBB68 (status axis/spindle replacement)

<table>
<tr><th>DB31, ... DBB68</th><th colspan="6">Status axis/spindle replacement</th></tr>
<tr><td>Signal flow</td><td colspan="6">PLC → NC</td></tr>
<tr><td>Update</td><td colspan="6">Cyclic</td></tr>
<tr><td rowspan="9">Additional information</td><td colspan="6">The current status of an axis with regard to axis/spindle replacement can be read by the PLC user program via DBB68.</td></tr>
<tr><td>Bit</td><td colspan="5">Meaning</td></tr>
<tr><td rowspan="3">0 ... 3</td><td colspan="5">Name of the channel to which the axis/spindle is assigned (binary code).
Example:
The axis is assigned to channel 2:</td></tr>
<tr><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td><td>Channel number</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>4</td><td colspan="5">New axis type is requested from the PLC.</td></tr>
<tr><td>5</td><td colspan="5">Axis replacement is possible</td></tr>
<tr><td>6</td><td colspan="5">Axis is "neutral axis/spindle".</td></tr>
<tr><td>7</td><td colspan="5">Axis is "PLC axis/spindle".</td></tr>
<tr><td>Corresponds with</td><td colspan="6">DB31, ... DBB8 (request axis/spindle replacement)
MD20070 $MC_AXCONF_MACHAX_USED (machine axis number valid in channel)
MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN (initial setting of channel for axis replacement)</td></tr>
<tr><td>Additional information</td><td colspan="6">Function Manual Basic Functions</td></tr>
</table>

18.2.6.113 DB31, ... DBX69.0 - 2 (active position controller parameter set)

DB31, ... DBX69.0 - 2	Active position controller parameter set
Signal flow	NC → PLC
Update	Job-controlled

DB31, ... DBX69.0 - 2	Active position controller parameter set			
Additional information	Bit 2	Bit 1	Bit 0	Parameter set
	0	0	0	1
	0	0	1	2
	0	1	0	3
	0	1	1	4
	1	0	0	5
	1	0	1	6
	1	1	0	6
	1	1	1	6
	Note The interface is irrelevant when the switchover is deactivated MD35590 $MA_PARAMSET_CHANGE_ENABLE == 0 In this case, the 1st parameter set is always active.			
Corresponds with	DB31, ...DBX9.0 - DBX9.2 (selection: position controller parameter set)			

18.2.6.114 DB31, ... DBX70.0 (REPOS offset)

DB31, ... DBX70.0	REPOS offset
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	A REPOS offset must be applied for the axis.
Signal state 0	**No** REPOS offset needs to be applied for the axis.
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.0 - 2 (REPOS mode) DB21, ... DBX31.4 (REPOS mode change) DB21, ... DBX319.0 (REPOS mode change acknowledgment) DB21, ... DBX319.1 - 3 (active REPOS mode) DB21, ... DBX319.5 (REPOS acknowledgment delay) DB31, ... DBX10.0 (REPOS delay) DB31, ... DBX70.1 (REPOS offset valid) DB31, ... DBX70.2 (REPOS delay acknowledgment) DB31, ... DBX72.0 (REPOS delay) DB31, ... DBX76.4 (path axis) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)
Additional information	Function Manual Basic Functions

18.2.6.115 DB31, ... DBX70.1 (REPOS offset valid)

DB31, ... DBX70.1	REPOS offset valid
Signal flow	NC → PLC
Update	Cyclic

DB31, ... DBX70.1	REPOS offset valid
Signal state 1	The REPOS offset has been calculated as valid
Signal state 0	The REPOS offset has been calculated as **invalid**.
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.0 - 2 (REPOS mode) DB21, ... DBX31.4 (REPOS mode change) DB21, ... DBX319.0 (REPOS mode change acknowledgment) DB21, ... DBX319.1 - 3 (active REPOS mode) DB21, ... DBX319.5 (REPOS acknowledgment delay) DB31, ... DBX10.0 (REPOS delay) DB31, ... DBX70.0 (REPOS offset) DB31, ... DBX70.2 (REPOS delay acknowledgment) DB31, ... DBX72.0 (REPOS delay) DB31, ... DBX76.4 (path axis) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)
Additional information	Function Manual Basic Functions

18.2.6.116 DB31, ... DBX70.2 (REPOS delay acknowledgment)

DB31, ... DBX70.2	REPOS delay acknowledgment
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The REPOS delay is acknowledged. **Note** A REPOS offset was pending for the axis and "REPOS delay" was active: DB31, ... DBX10.0 == 1 (REPOS delay) The axis was programmed within a traversing block, and the REPOS offset was applied. The interface signal behaves just like: DB21, ... DBX319.1 - 3 (REPOS approach mode acknowledgment)
Signal state 0	The REPOS delay is still not acknowledged - or there is no REPOS offset available.
Corresponds with	DB21, ... DBX25.4 (REPOS mode activation) DB21, ... DBX31.0 - 2 (REPOS mode) DB21, ... DBX31.4 (REPOS mode change) DB21, ... DBX319.0 (REPOS mode change acknowledgment) DB21, ... DBX319.1 - 3 (active REPOS mode) DB21, ... DBX319.5 (REPOS acknowledgment delay) DB31, ... DBX10.0 (REPOS delay) DB31, ... DBX70.0 (REPOS offset) DB31, ... DBX70.1 (REPOS offset valid) DB31, ... DBX72.0 (REPOS delay) DB31, ... DBX76.4 (path axis) MD11470 $MN_REPOS_MODE_MASK (repositioning properties)
Additional information	Function Manual Basic Functions

18.2.6.117 DB31, ... DBX71.4 (position restored measuring system 1)

DB31, ... DBX71.4	Position restored, measuring system 1
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The position of measuring system 1 of the machine axis is restored.
Signal state 0	The position of measuring system 1 of the machine axis is **not** restored.
Additional information	MD34210 $MA_ENC_REFP_STATE == 3 After the control system has powered up, for distance-coded, incremental measuring systems the last axis position buffered before switch off is restored. Referencing does not take place automatically. The position measuring system is in the "position restored" state.
Corresponds with	DB31, ... DBX71.5 (position restored, measuring system 2) MD34102 $MA_REFP_SYNC_ENCS (measuring system calibration) MD34210 $MA_ENC_REFP_STATE (adjustment status absolute encoder)
Additional information	Function Manual Axes and Spindles

18.2.6.118 DB31, ... DBX71.5 (position restored, measuring system 2)

DB31, ... DBX71.5	Position restored, measuring system 2
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The position of measuring system 2 of the machine axis is restored.
Signal state 0	The position of measuring system 2 of the machine axis is **not** restored.
Additional information	MD34210 $MA_ENC_REFP_STATE == 3 After the control system has powered up, for distance-coded, incremental measuring systems the last axis position buffered before switch off is restored. Referencing does not take place automatically. The position measuring system is in the "position restored" state.
Corresponds with	DB31, ... DBX71.4 (position restored measuring system 1) MD34102 $MA_REFP_SYNC_ENCS (measuring system calibration) MD34210 $MA_ENC_REFP_STATE (adjustment status absolute encoder)
Additional information	Function Manual Axes and Spindles

18.2.6.119 DB31, ... DBX72.0 (REPOS delay)

DB31, ... DBX72.0	REPOS delay active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The REPOS delay is active. After a block search, a REPOS offset is applied for this axis. However it is not applied using the **approach block**, but rather using the next **traversing block** in which the axis is **programmed**.
Signal state 0	A REPOS delay is **not** active.

DB31, ... DBX72.0	REPOS delay active
Additional information	**Note** If the machine axes are involved on a path (DB31, ... DBX76.4 == 1 (path axis)), then the interface signal is not active.
Corresponds with	DB31, ... DBX72.0 (REPOS delay)
Additional information	Function Manual Basic Functions

18.2.6.120 DB31, ... DBX74.4 (modulo rotary axis: traversing range limits active)

DB31, ... DBX74.4	Modulo rotary axis: traversing range limits active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Traversing range limits (software limit switch, working area limits) active for modulo rotary axes.
Signal state 0	Traversing range limitation for modulo rotary axes **not** active.
Additional information	**Note** The signal is irrelevant if linear axes/rotary axes do not have modulo functionality. **Application example** Mounted rotary axis with monitoring
Corresponds with	DB31, ... DBX12.4 (modulo rotary axis: activate traversing range limits)
Additional information	Function Manual Axes and Spindles

18.2.6.121 DB31, ... DBX75.0 - 2 (JOG approach fixed point active)

DB31, ... DBX75.0 - 2	JOG approach fixed point active
Signal flow	NC → PLC
Update	Cyclic
Additional information	As soon as the "Approach fixed point in JOG" function is active, the number of the fixed point to be approached is signaled back to the PLC binary-coded using DB31, ... DBX75.0 - 2:

Bit 2	Bit 1	Bit 0	Number of fixed point to be approached
0	0	0	-
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4

	The selected machine axis can be traversed to the corresponding fixed point with the traversing keys or the handwheel. The fixed points are defined using machine data MD30600.
Corresponds with	DB31, ... DBX13.0 - 2 (JOG approach fixed point) DB31, ... DBX75.3 - 5 (JOG approach fixed point reached) MD30600 $MA_FIX_POINT_POS[<n>] (fixed value positions of the axis)
Additional information	Function Manual Axes and Spindles

18.2.6.122 DB31, ... DBX75.3 - 5 (JOG approach fixed point reached)

DB31, ... DBX75.3 - 5	JOG approach fixed point reached
Signal flow	NC → PLC
Update	Cyclic
Additional information	Once the axis for "Approach fixed point in JOG" has reached the fixed point position with "Exact stop fine", using DB31, ... DBX75.3 - 5 the number of the approached fixed point is signaled back to the PLC in binary-coded form:

Bit 2	Bit 1	Bit 0	Number of the approached fixed point
0	0	0	-
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4

	This feedback signal is also issued if the axis reaches the fixed point position in the machine coordinates system via other methods e.g. NC program, FC18 (for 840D sl) or synchronized action on the setpoint side and comes to a standstill on the actual value side within the "Exact stop fine" tolerance window.
Corresponds with	DB31, ... DBX13.0 - 2 (JOG approach fixed point) DB31, ... DBX75.0 - 2 (JOG approach fixed point active) MD30600 $MA_FIX_POINT_POS[<n>] (fixed value positions of the axis) MD36010 $MA_STOP_LIMIT_FINE (exact stop fine)
Additional information	Function Manual Axes and Spindles

18.2.6.123 DB31, ... DBX75.6 (JOG travel to position active)

DB31, ... DBX75.6	JOG travel to position active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The function "Position travel in JOG" is active.
Signal state 0	The function "Position travel in JOG" is **not** active
Additional information	The selected machine axis can be traversed to the position specified with setting data SD43320 using the traversing keys or the handwheel.
Corresponds with	DB31, ... DBX13.3 (JOG travel to position) DB31, ... DBX75.7 (JOG position reached) SD43320 $SA_JOG_POSITION (JOG position)
Additional information	Function Manual Axes and Spindles

18.2.6.124 DB31, ... DBX75.7 (JOG position reached)

DB31, ... DBX75.7	JOG position reached
Signal flow	NC → PLC
Update	Cyclic

DB31, ... DBX75.7	JOG position reached
Signal state 1	For "Position travel in JOG", the axis has reached the position specified with SD43320 with "Exact stop fine".
Signal state 0	For "Position travel in JOG", the axis has still **not** reached the position specified with SD43320.
Additional information	Approaching the position is started with the traversing keys or the handwheel. The axis traverses until it comes to an automatic standstill at the position to be approached.
Corresponds with	DB31, ... DBX13.3 (JOG travel to position) DB31, ... DBX75.6 (JOG travel to position active) SD43320 $SA_JOG_POSITION (JOG position)
Additional information	Function Manual Axes and Spindles

18.2.6.125 DB31, ... DBX76.0 (lubrication pulse)

DB31, ... DBX76.0	Lubrication pulse
Signal flow	NC → PLC
Update	Cyclic
Edge change 0 → 1	The traversing path parameterized in machine data (MD33050 $MA_LUBRICATION_DIST) was traversed.
Edge change 1 → 0	The traversing path parameterized in machine data (MD33050 $MA_LUBRICATION_DIST) was traversed.
Additional information	As soon as the axis/spindle has moved through the traversing distance set in machine data, then the interface signal is inverted. The distance measurement is always restarted after the control powers up.
Corresponds with	MD33050 $MA_LUBRICATION_DIST (lubrication pulse distance)

18.2.6.126 DB31, ... DBX76.4 (path axis)

DB31, ... DBX76.4	Path axis
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis is not a path axis: This means that it is traversed with other axes along a path (path axis).
Signal state 0	The axis is **not** a path axis.
Additional information	**Note** In conjunction with block search type 5 (SERUPRO) in the state: "Target block found" - the interface signal refers to the property of the axis in the target block.
Corresponds with	DB31, ... DBX72.0 (REPOS delay)
Additional information	Function Manual Basic Functions

18.2.6.127 DB31, ... DBX76.5 (positioning axis)

DB31, ... DBX76.5	Positioning axis
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis is a positioning axis.
Signal state 0	The axis is not a positioning axis.
Corresponds with	DB31, ... DBD78 (feedrate, positioning axis)
Additional information	Function Manual Axes and Spindles

18.2.6.128 DB31, ... DBX76.6 (indexing axis in position)

DB31, ... DBX76.6	Indexing axis in position
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The signal is set to "1" in the following cases: • The indexing axis has reached the indexing position with "Exact stop fine". • The indexing axis is located at the indexing position, which was approached in the AUTOMATIC mode with CAC, CACP, CACN, CDC or CIC.
Signal state 0	The signal is "0" in the following cases: • The axis is not defined as an indexing axis. • A traversing command is active, and the indexing axis is traversed. • The indexing axis is located at a position which is not an indexing position. • In the AUTOMATIC mode, the indexing axis was not positioned using CAC, CACP, CACN, CDC or CIC but instead is traversed to any position, for example with AC or DC. • The "Controller enable" of the indexing axis is withdrawn.
Additional information	Signal is irrelevant for axes that have not been defined as indexing axes. **Application example** Tool magazine: Activation of the gripper to remove the tool from the magazine is initiated as soon as the indexing axis is in position. The PLC user program must ensure this happens. **Special cases or errors** The axis positions entered in the indexing position table for the individual divisions can be changed using work offsets (including DRF). The interface signal "Indexing axis in position" is set if the actual position of the indexing axis assumes the position value entered into the indexing table plus the offset. If a DRF offset is applied to an indexing axis in AUTOMATIC mode, then the "Indexing axis in position" interface signal remains active even though the axis is no longer at an indexing position.
Corresponds with	DB31, ... DBX64.6/64.7 (traversing command -/+) DB31, ... DBX2.1 (controller enable) MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB (axis is an indexing axis)
Additional information	Function Manual Axes and Spindles

18.2.6.129 DB31, ... DBX77.0 (collision avoidance: Velocity reduction)

DB31, ... DBX77.0	Collision avoidance: Velocity reduction
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The traversing velocity of the axis is reduced by the collision avoidance.
Signal state 0	The traversing velocity of the axis is **not** reduced by the collision avoidance.
Additional information	Function Manual Monitoring and Compensating

18.2.6.130 DB31, ... DBD78 (feedrate, positioning axis)

DB31, ... DBD78	Feedrate, positioning axis
Signal flow	NC → PLC
Update	Cyclic
Additional information	The feedrate value of the axis can be read via the interface if this axis is traversed as positioning axis. Type: REAL The output time is specified using machine data: MD22240 $MC_AUXFU_F_SYNC_TYPE **Notes** • As default setting, the output is suppressed, because if the feedrate is output (auxiliary function output) in the continuous-path mode, then the velocity can drop. • The last feedrate value is kept until it is overwritten by a new value.
Corresponds with	DB31, ... DBX76.5 (positioning axis) MD22240 $MC_AUXFU_F_SYNC_TYPE (output time of F functions) MD32060 $MA_POS_AX_VELO (initial setting for positioning axis velocity)
Additional information	Function Manual Axes and Spindles

18.2.6.131 DB31, ... DBX82.0 - 2 (set gear stage)

DB31, ... DBX82.0 - 2	Set gear stage
Signal flow	NC → PLC
Update	Cyclic

DB31, ... DBX82.0 - 2	Set gear stage			
Additional information	Gear stage requested from the NC, which must be selected at the machine.			
	Bit 2	Bit 1	Bit 0	Meaning:
	---	---	---	Axis mode
	0	0	0	Gear stage 1
	0	0	1	Gear stage 1
	0	1	0	Gear stage 2
	0	1	1	Gear stage 3
	1	0	0	Gear stage 4
	1	0	1	Gear stage 5
	1	1	0	Gear stage 5
	1	1	1	Gear stage 5
Corresponds with	DB31, ... DBX16.0 - 2 (actual gear stage) DB31, ... DBX16.3 (gear stage has been changed) DB31, ... DBX18.5 (oscillation speed) DB31, ... DBX82.3 (change over gear stage)			
Additional information	Function Manual Axes and Spindles			

18.2.6.132 DB31, ... DBX82.3 (change over gear stage)

DB31, ... DBX82.3	Change gear stage
Signal flow	NC → PLC
Update	Cyclic
Edge change 0 → 1	There is an active requested to changeover the gear to the set gear stage.
Edge change 1 → 0	No effect.
Additional information	**Note** The gear is only changed over to the new set gear stage if the following applies: Set gear stage <> actual gear stage
Corresponds with	DB31, ... DBX16.0 - 2 (actual gear stage) DB31, ... DBX82.0 - 2 (set gear stage)
Additional information	Function Manual Axes and Spindles

18.2.6.133 DB31, ... DBX83.0 (speed limit exceeded)

DB31, ... DBX83.0	Speed limit exceeded
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The speed limit is exceeded.
Signal state 0	The speed limit is **not** exceeded.
Additional information	The speed limit is exceeded if the following applies. Actual speed > (MD35100 $MA_SPIND_VELO_ LIMIT + MD35150 $MA_SPIND_DES_VELO_TOL)

DB31, ... DBX83.0	Speed limit exceeded
Corresponds with	MD35150 $MA_SPIND_DES_VELO_TOL (spindle speed tolerance) MD35100 $MA_SPIND_VELO_LIMIT (maximum spindle speed)
Additional information	Function Manual Axes and Spindles

18.2.6.134 DB31, ... DBX83.1 (speed setpoint limited)

DB31, ... DBX83.1	Setpoint speed limited
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The setpoint speed is limited.
Signal state 0	The setpoint speed is **not** limited.
Additional information	The NC has automatically limited the setpoint speed, as it exceeds the active **maximum** limit value: • MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT • MD35100 $MA_SPIND_VELO_LIMIT • DB31, ... DBX3.6 • `G26` • `LIMS` • `VELOLIM` • **Safety Integrated** MD36931 $MA_SAFE_VELO_LIMIT (limit for safely reduced speed) **Notes** • A possible response in the PLC user program would be to inhibit/lock the path feedrate: DB21, ... DBX6.0 = 1 (feedrate inhibit) • The interface signal DB31, ... DBX83.5 (spindle in the setpoint range) is updated.
	Safety Integrated In addition to the limit value MD36931 $MA_SAFE_VELO_LIMIT, depending on the active safety speed level SG1 ... SGn, the following machine data should be taken into account: • MD36932 $MA_ SAFE_VELO_OVR_FACTOR • MD36933 $MA_SAFE_DES_VELO_LIMIT Example: All standard limit values are greater than 1500 rpm. • SG1 is active • MD36932 $MA_ SAFE_VELO_OVR_FACTOR[<SG1>] = 1111.11111 [rpm] • MD36933 $MA_SAFE_DES_VELO_LIMIT[<SG1>] = 90% Programming: M3 S1500 The speed setpoint is limited to 1000 rpm (MD36932 * MD36933) ⇒ DB31, ... DBX83.1 = 1

DB31, ... DBX83.1	Setpoint speed limited
Corresponds with	DB21, ... DBX6.0 (feedrate disable) DB31, ... DBX4.3 (feedrate/spindle stop) DB31, ... DBX83.2 (speed setpoint increased) DB31, ... DBX83.5 (spindle in setpoint range) DB31, ... DBX3.6 (spindle speed limitation to MD35160 $MA_SPIND_EXTERN_VELO_LIMIT) MD35100 $MA_SPIND_VELO_LIMIT (maximum spindle speed) MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT (maximum speed of gear stage) MD36931 $MA_SAFE_VELO_LIMIT (limit for safely reduced speed) MD36932 $MA_ SAFE_VELO_OVR_FACTOR (SG correction values) MD36933 $MA_SAFE_DES_VELO_LIMIT (SG setpoint speed limit) G26 (upper spindle speed limit) LIMS (speed limit for the master spindle when G96/G961/G97 is active) VELOLIM: programmed spindle speed limit in the open-loop speed controlled mode
Additional information	Function Manual Axes and Spindles

18.2.6.135 DB31, ... DBX83.2 (speed setpoint increased)

DB31, ... DBX83.2	Setpoint speed increased
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The setpoint speed was increased.
Signal state 0	The setpoint speed was **not** increased.
Additional information	The NC has automatically increased the setpoint speed, as it had fallen below the active **minimum** limit value: • MD35120 $MA_GEAR_STEP_MIN_VELO • MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT • G25 **Notes** • A possible response in the PLC user program would be to inhibit/lock the path feedrate: DB21, ... DBX6.0 = 1 (feedrate inhibit) • The interface signal DB31, ... DBX83.5 (spindle in the setpoint range) is updated.
Corresponds with	DB21, ... DBX6.0 (feedrate disable) DB31, ... DBX4.3 (feedrate/spindle stop) DB31, ... DBX83.1 (speed setpoint limited) DB31, ... DBX83.5 (spindle in setpoint range) MD35120 $MA_GEAR_STEP_MIN_VELO (minimum speed for automatic gear stage selection M40) MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT (minimum speed of the gear stage) G25 (lower spindle speed limitation)
Additional information	Function Manual Axes and Spindles

18.2.6.136 DB31, ... DBX83.3 (geometry monitoring)

DB31, ... DBX83.3	Geometry monitoring
Signal flow	NC → PLC
Signal state 1	Error in the grinding wheel geometry The actual wheel radius falls below the value specified in parameter $TC_TPG3 or the actual wheel width ($TC_TPG5) falls below the value defined in parameter $TC_TPG4. **Note** There is no controller-internal error response. Required responses must be programmed by the PLC user.
Signal state 0	No error in the grinding wheel geometry
Additional information	The geometry monitoring is a subfunction of the grinding-specific tool monitoring. The actual wheel radius and the actual wheel width are monitored.
Additional information	Function Manual Tools

18.2.6.137 DB31, ... DBX83.5 (spindle in setpoint range)

DB31, ... DBX83.5	Spindle in setpoint range
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The spindle speed is in the setpoint range.
Signal state 0	The spindle speed is **not** in the setpoint range.
Additional information	The spindle speed is in the setpoint range if the following applies. | Actual speed - setpoint speed | < MD35150 $MA_SPIND_DES_VELO_TOL **Note** It is normal if the spindle speed is not in the setpoint range while accelerating or braking.
Corresponds with	DB21, ... DBX6.0 (feedrate disable) DB31, ... DBX4.3 (feedrate/spindle stop) DB31, ... DBX83.1 (speed setpoint limited) DB31, ... DBX83.2 (speed setpoint increased)
Additional information	

18.2.6.138 DB31, ... DBX83.6 (speed monitoring)

DB31, ... DBX83.6	Speed monitoring
Signal flow	NC → PLC
Signal state 1	Error in grinding wheel speed The speed limit value is reached, the speed is limited to the limit value. **Note** There is no controller-internal error response. Required responses must be programmed by the PLC user.
Signal state 0	No error in grinding wheel speed

DB31, ... DBX83.6	Speed monitoring
Additional information	The speed monitoring is a subfunction of the grinding-specific tool monitoring. The speed monitor checks the grinding wheel peripheral speed in m/s (parameter $TC_TPG7) as well as the maximum spindle speed in rpm (parameter $TC_TPG6). The speed setpoint is monitored against the speed limitation cyclically with allowance for the spindle override.
Additional information	Function Manual Tools

18.2.6.139 DB31, ... DBX83.7 (clockwise actual direction of rotation)

DB31, ... DBX83.7	Actual direction of rotation clockwise
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Actual direction of rotation: Clockwise
Signal state 0	Actual direction of rotation: Counterclockwise
Additional information	The actual direction of rotation is derived from the spindle position measuring encoder. **Note** The interface signal is irrelevant for: • DB31, ... DBX61.4 == 1 (axis/spindle stationary) • Spindles without position encoder
Corresponds with	DB31, ... DBX61.4 (axis/spindle stationary)
Additional information	Function Manual Axes and Spindles

18.2.6.140 DB31, ... DBX84.1 (grinding wheel peripheral speed active)

DB31, ... DBX84.1	Grinding wheel peripheral speed active
Signal flow	NC → PLC
Signal state 1	The function "constant grinding wheel peripheral speed (GWPS)" is active.
Signal state 0	The function "Constant grinding wheel peripheral speed (GWPS)" is **not** active.
Additional information	If the function is active, then all S value inputs from the PLC are interpreted as the grinding wheel peripheral speed.
Additional information	Function Manual Tools

18.2.6.141 DB31, ... DBX84.3 (tapping without compensating check active)

DB31, ... DBX84.3	Tapping without compensating chuck active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Rigid tapping (without compensating chuck) active
Signal state 0	Rigid tapping (without compensating chuck) **not** active

DB31, ... DBX84.3	Tapping without compensating chuck active
Additional information	If "Rigid tapping without compensating chuck" (G331/G332) is active, then internally the spindle is switched over into closed-loop position controlled axis mode. **Notice** Setting the following interface signals while tapping without compensating chuck (rigid tapping) will destroy the thread: • DB11, … DBX0.7 (mode group reset) = 1 • DB21, ... DBX7.7 (channel reset) = 1 • DB31, ... DBX2.1 (controller enable) = 0 • DB31, ... DBX4.3 (feedrate stop) = 1
Corresponds with	DB11, … DBX0.7 (mode group reset) DB21, ... DBX7.7 (channel reset) DB31, ... DBX2.1 (controller enable) DB31, ... DBX4.3 (feedrate stop)
Additional information	Function Manual Axes and Spindles

18.2.6.142 DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation)

DB31, ... DBX84.4	Active spindle mode: Synchronous mode
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The spindle is in the "synchronous mode". **Note** The signal is only set for the machine axis which is active as following spindle: DB31, ... DBX99.1 (following spindle active) == 1
Signal state 0	The spindle is not operated as a following spindle in the "synchronous mode".
Additional information	In the synchronous mode, the following spindle tracks the motion of the leading spindle corresponding to the ratio and the monitoring functions are active for coarse and fine synchronism. When deactivating the coupling (deselection of synchronous mode), the following spindle is switched to the "control mode".
Corresponds with	DB31, ... DBX98.0 (fine synchronous operation) DB31, ... DBX98.1 (coarse synchronous operation) DB31, ... DBX99.1 (following spindle active)
Additional information	Function Manual Axes and Spindles

18.2.6.143 DB31, ... DBX84.5 (active spindle mode: positioning mode)

DB31, ... DBX84.5	Active spindle mode: Positioning mode
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Positioning mode (SPOS or SPOSA) is active.
Signal state 0	Positioning mode (SPOS or SPOSA) is **not** active.

DB31, ... DBX84.5	Active spindle mode: Positioning mode
Corresponds with	DB31, ... DBX84.6 (spindle mode, oscillation mode) DB31, ... DBX84.7 (spindle mode, control mode)
Additional information	Function Manual Axes and Spindles

18.2.6.144 DB31, ... DBX84.6 (active spindle mode: oscillating mode)

DB31, ... DBX84.6	Active spindle mode: Oscillation mode
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Oscillation mode is active.
Signal state 0	Oscillation mode is **not** active.
Additional information	**Note** When the gear stage is changed over, the spindle automatically goes into the oscillation mode.
Corresponds with	DB31, ... DBX82.3 (change over gear stage) DB31, ... DBX84.5 (spindle mode, positioning mode) DB31, ... DBX84.7 (spindle mode, control mode)
Additional information	Function Manual Axes and Spindles

18.2.6.145 DB31, ... DBX84.7 (active spindle mode: control mode)

DB31, ... DBX84.7	Active spindle mode: Control mode
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Control mode is active
Signal state 0	Control mode is **not**active.
Additional information	The spindle is in control mode for the following functions: • Spindle direction of rotation specified by `M3`/`M4` or spindle stop `M5` • `M41`...`M45`, or automatic gear stage change `M40`
Corresponds with	DB31, ... DBX84.5 (spindle mode, positioning mode) DB31, ... DBX84.6 (spindle mode, oscillation mode)
Additional information	Function Manual Axes and Spindles

18.2.6.146 DB31, ... DBX85.0 (tool with dynamic response limitation)

DB31, ... DBX85.0	Tool with dynamic response limitation
Signal flow	NC → PLC
Update	Cyclic

DB31, ... DBX85.0	Tool with dynamic response limitation
Signal state 1	After a tool change on the spindle, it is shown that a tool with dynamic response limitation is in the spindle. To do this, the PLC acknowledgments are evaluated. The assignment state of the spindles is checked, especially if all tools have a maximum tool speed, which may be very high so that there is no effective limiting.
Signal state 0	The tool does not contain a parameterized dynamic response limitation.
Corresponds with	Option "Tool monitoring for maximum speed" Tool data TC_TP_MAX_VELO >0 or TC_TP_MAX_ACC >0
Additional information	Function Manual, Tool Management

18.2.6.147 DB31, ... DBX85.5 (spindle in position)

DB31, ... DBX85.5	Spindle in position
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The spindle is in position.
Signal state 0	The spindle is **not** in position.
Additional information	Precondition for the output of the interface signal: • DB31, ... DBX60.7 == 1 (exact stop fine) **AND** • Programmed setpoint position is reached **Note** • The interface signal is only processed in the positioning mode (DB31, ... DBX84.5 == 1), e.g.: – NC program: SPOS, SPOSA and M19: – Synchronized action: SPOS and M19 – PLC user program: Positioning with FC18 or DB31, ... DBX30.4 • The interface signal remains set if the spindle is already at the programmed setpoint position.
Corresponds with	DB31, … DBX30.4 (spindle positioning start) DB31, ... DBX60.7 (exact stop fine) DB31, ... DBX84.5 (spindle mode, positioning mode) SPOS SPOSA M19
Additional information	Function Manual Axes and Spindles

18.2.6.148 DB31, ... DBW86 (M function for spindle)

DB31, ... DBW86	M function for spindle
Signal flow	NC → PLC
Update	Job-controlled

DB31, ... DBW86	M function for spindle	
Additional information	The M function programmed for the spindle in the NC program is output.	
	Value	M function
	3	M3 (clockwise spindle rotation)
	4	M4 (counterclockwise spindle rotation)
	5	M5 (spindle stop)
	19	M19 (spindle positioning to the position entered in SD43240)
	70	M70 (switchover in axis operation)
Corresponds with	DB21, ... DBX58.0 - 4 (M function: change) DB21, ... DBX59.0 - 4 (M function: not decoded) DB21, ... DBB68 - 97 (M functions for spindles)	

18.2.6.149 DB31, ... DBD88 (S function for spindle)

DB31, ... DBD88	S function for spindle
Signal flow	NC → PLC
Update	Job-controlled
Additional information	The spindle-specific S function programmed in the NC program is output. The following S functions are checked: • S.... as spindle speed in rpm (programmed value) • S .... as constant cutting rate in m/min or ft/min The following S functions are **not** output: • S.... as the programmed spindle speed limiting `G25` • S.... as the programmed spindle speed limiting `G26` • S .... as spindle speed in rpm if a spindle was not defined in the control system • S.... as the dwell time in spindle revolutions
Corresponds with	DB21, ... DBX60.0 - 2 (S function: change) DB21, ... DBX60.4 - 6 (S function: Quick) DB21, ... DBB98 - 115 (S functions for spindles), channel-specific

18.2.6.150 DB31, ... DBX92.1 (ramp-function generator disable active)

DB31, ... DBX92.1	Ramp-function generator disable
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Ramp function generator fast stop is active for the axis drive.
Signal state 0	Ramp function generator fast stop is not active for the axis drive.
Corresponds with	DB31, ... DBX20.1 (ramp-function generator fast stop)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.151 DB31, ... DBX92.4 (drive-autonomous motion active)

DB31, ... DBX92.4	Drive-autonomous motion active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Drive-autonomous motion is active. The axis traverses due to setpoints generated by drive-internal functions. The drive still responds to control signals of the NC, e.g. controller enable. Setpoints from the NC are ignored.
Signal state 0	Drive-autonomous motion is not active.
Additional information	DBX92.4 = 1 **IF** MELDW.11 == 1 (controller enable) **AND** ZSW1.2 == 0 (operation enabled)
	Note • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Additional information is available in the following reference: SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
	Application examples Internal drive functions: • Rotor or pole position identification • Function generator
Additional information	SINAMICS S120 Function Manual

18.2.6.152 DB31, ... DBX93.0 - 4 (motor/drive data set: display)

DB31, ... DBX93.0 - 4	Motor/drive data set: Display
Signal flow	NC → PLC
Update	Cyclic
Additional information	Display interface for the currently active motor (MDS) / drive data set (DDS).
	Formatting Formatting the display interface, i.e. which bits are used to address the motor data sets (MDS) – and which are used to address the drive data sets (DDS) is set via the formatting interface (DB31, …DBX130.0 - 4).
Corresponds with	DB31, ... DBX21.0 - 4 (motor/drive data set: selection) DB31, ... DBX21.5 (motor has been selected) DB31, …DBX130.0 ... 4 (motor/drive data set: formatting)
Additional information	• Function Manual Basic Functions • SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.153 DB31, ... DBX93.5 (drive ready)

DB31, ... DBX93.5	Drive ready
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The drive is ready.
Signal state 0	The drive is **not** ready. If the signal is reset while operational, the drive is stopped (pulse inhibit or fast stop). When powering-up, the pulses are still inhibited (canceled). In addition, the following interface signals are reset: • DB10, DBX108.6 = 0 (drive ready) • DB31, ... DBX61.7 = 0 (current controller active) • DB31, ... DBX61.6 = 0 (speed controller active)
Additional information	Interface signal: DB31, ... DBX93.5 = drive: MELDW.12
	Note • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Additional information is available in the following reference: SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Corresponds with	DB10, DBX108.6 (Drives ready) DB31, ... DBX61.7 (current controller active) DB31, ... DBX61.6 (speed controller active)
Additional information	• Function Manual Basic Functions • SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive • SINAMICS S120/S150 List Manual

18.2.6.154 DB31, ... DBX93.6 (integrator disable, speed controller)

DB31, ... DBX93.6	Integrator disable, speed controller
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The integrator of the speed controller in the drive is inhibited. The speed controller functions as a P controller.
Signal state 0	The integrator of the speed controller in the drive is not disabled. The speed controller functions as a PI controller.
Corresponds with	DB31, ... DBX21.6 (integrator disable, speed controller)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.155 DB31, ... DBX93.7 (pulses enabled)

DB31, ... DBX93.7	Pulses enabled
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The drive pulses are enabled.
Signal state 0	The drive pulses are not enabled. • DB31, ... DBX61.7 = 0 (current controller active) • DB31, ... DBX61.6 = 0 (speed controller active) • DB31, ... DBX61.5 = 0 (position controller active)
Additional information	DB31, ... DBX93.7 = MELDW.13
	Note • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Additional information is available in the following reference: SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Corresponds with	DB31, ... DBX21.7 (pulse enable)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.156 DB31, ... DBX94.0 (motor temperature prewarning)

DB31, ... DBX94.0	Motor temperature prewarning
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The motor temperature has exceeded the warning threshold (p0604) configured in the drive. **Note** If the motor temperature remains too high for longer than the parameterized time (p0606), a fault is output, the drive is stopped and the pulse enable removed. If the motor temperature falls below the warning threshold (p0604) again before the time has expired (p0606), the interface signal is reset.
Signal state 0	The motor temperature is below the warning threshold (p0604).
Additional information	The actual motor temperature is displayed on the user interface at: Operating area "Diagnostics" > "Service display: Axis/spindle"
Corresponds with	DB31, ... DBX94.1 (heat sink temperature prewarning)

18.2.6.157 DB31, ... DBX94.1 (heat sink temperature prewarning)

DB31, ... DBX94.1	Heat sink temperature prewarning
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The heat sink temperature of the power semiconductors has exceeded the parameterized warning threshold (p0294). **Note** The parameterized reaction (p0290) is performed in the drive. If the temperature violation remains, a fault is output after approx. 20 s, the drive is stopped and the pulse enable removed.
Signal state 0	The heat sink temperature is below the warning threshold.
Additional information	The interface signals DB31, ... DBX94.0 and .1 are derived from the following signals of the cyclic drive telegram: • Case 1: Temperature alarm in the message word – DB31, ... DBX94.0 ≙ MELDW, bit 6 (no motor overtemperature warning) – DB31, ... DBX94.1 ≙ MELDW, bit 7 (no thermal overload in power unit warning) • Case 2: Alarm of alarm class B (interface mode "SIMODRIVE 611U", p2038 = 1) DB31, ... DBX94.0 == 1 and DBX94.1 == 1, if the following applies: cyclic drive telegram, ZSW1: bit 11 == 0 and 12 == 1 (alarm class B) The interface signals are derived from the alarm of alarm class B if there is no specific information from the message word. An alarm is displayed. Alarm number = 200.000 + alarm value (r2124)
	Note • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Additional information is available in the following reference: SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Additional information	• S120 Commissioning Manual • S120 Function Manual • S120 List Manual – MELDW, bit 6 ≙ BO: r2135.14 → function diagram: 2548, 8016 – MELDW, bit 7 ≙ BO: r2135.15 → function diagram: 2548, 2452, 2456, 8016 • SINUMERIK Diagnostics Manual

18.2.6.158 DB31, ... DBX94.2 (run-up completed)

DB31, ... DBX94.2	Power up completed
Signal flow	NC → PLC
Update	Cyclic

DB31, ... DBX94.2	Power up completed
Signal state 1	After a new speed setpoint, the speed actual value has reached the speed tolerance band defined using drive parameter p2164, and has not left it for the time specified in p2166. Any subsequent speed fluctuations, also outside the tolerance band, e.g. due to load changes, will not affect the interface signal.
Signal state 0	The power up operation is still active after the speed setpoint has been changed.
Corresponds with	DB31, ... DBX94.6 ("$n_{act} = n_{set}$") DB31, ... DBX94.3 ("$\|M_D\| = M_{dx}$")
Additional information	• SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive • SINAMICS S120 List Manual

18.2.6.159 DB31, ... DBX94.3 (|Md| < Mdx)

DB31, ... DBX94.3	$\|M_d\| < M_{dx}$
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The current torque utilization is below the torque utilization threshold (torque threshold 2, p2194). The power up operation has been completed, the drive is in the steady state and the torque setpoint $\|M_d\|$ drive does not exceed the threshold torque M_{dx}. The torque threshold characteristic is speed-dependent. During the power up operation, DB31, ... DBX94.3 ($\|M_d\| < M_{dx}$) == 1. The interface signal is not updated until the power has been completed (DB31, ... DBX94.2 == 1) and the signal interlocking time for the threshold torque has expired.
Signal state 0	The torque setpoint $\|M_d\|$ is higher than the threshold torque M_{dx}. A motor overload can be determined via the interface signal. An appropriate response can then be initiated in the PLC user program.
Additional information	DB31, ... DBX94.3 = MELDW.1
	Note • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Additional information is available in the following reference: SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Additional information	• SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive • SIMATIC S120 List Manual

18.2.6.160 DB31, ... DBX94.4 (|nact| < nmin)

DB31, ... DBX94.4	$\|n_{act}\| < n_{min}$
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The actual speed value n_{act} is less than n_{min} (speed threshold value 3, p2161).
Signal state 0	The actual speed value is greater than the minimum threshold speed n_{min}.
Additional information	DB31, ... DBX94.4 = MELDW.2
	Note • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Additional information is available in the following reference: SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.161 DB31, ... DBX94.5 (|nact| < nx)

DB31, ... DBX94.5	$\|n_{act}\| < n_x$
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The actual speed value n_{act} is less than n_x (speed threshold value 2, p2155).
Signal state 0	The actual speed value n_{act} is greater than the threshold speed n_x.
Additional information	DB31, ... DBX94.5 = MELDW.3
	Note • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Additional information is available in the following reference: SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.162 DB31, ... DBX94.6 (nact = nset)

DB31, ... DBX94.6	$n_{act} = n_{set}$
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The actual speed value is at least the parameterized time (switch-on delay n_act = n_set, p2167) within the tolerance band around the speed setpoint (speed threshold value 4, p2163).
Signal state 0	The actual speed value is outside the tolerance band around the speed setpoint (speed threshold value 4, p2163).
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.163 DB31, ... DBX94.7 (variable signaling function)

DB31, ... DBX94.7	Variable signaling function
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The parameterized drive variable has exceeded the specified threshold value, including hysteresis.
Signal state 0	The parameterized drive variable has fallen below the specified threshold value, including hysteresis.
Additional information	Using the "Variable signaling" function, BICO interconnections and parameters which have the attribute traceable can be monitored in the drive.
	DB31, ... DBX94.7 = MELDW.5
	Note • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Additional information is available in the following reference: SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Additional information	SINAMICS S120 Function Manual

18.2.6.164 DB31, ... DBX95.1 (ESR: DC link undervoltage)

DB31, ... DBX95.1	ESR: DC link undervoltage
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The drive signals that the DC link voltage $U_{DC\ link}$ is **less** than the lower DC link voltage threshold set with parameter p1248.
Signal state 0	The drive signals that the DC link voltage $U_{DC\ link}$ is **greater** than they lower DC link voltage threshold set with parameter p1248.

DB31, ... DBX95.1	ESR: DC link undervoltage
Additional information	**Note** The PLC user program can initiate measures in order to safely end machining and/or to buffer the DC link voltage, e.g. initiate a drive-based and/or control-managed extended stop and retract (ESR).
Corresponds with	Drive parameter p1248 (lower DC link voltage threshold) Drive telegram MELDW.Bit 4
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, drive

18.2.6.165 DB31, ... DBX95.2 (ESR: Response initiated or generator operation active)

DB31, ... DBX95.2	ESR: Response initiated or generator operation active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The drive signals that the configured ESR response has been initiated or generator operation is active.
Signal state 0	The drive signals that neither the configured ESR response has been initiated nor that generator operation is active.
Corresponds with	Drive telegram MELDW.Bit 9
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, drive

18.2.6.166 DB31, … DBX95.3 (generator operation minimum speed fallen below)

DB31, ... DBX95.3	ESR: Generator operation - minimum speed fallen below
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis is parameterized as generator axis. The drive signals that the actual speed is **less** than the minimum speed set using parameter p2161 (speed threshold value 3).
Signal state 0	The axis is parameterized as generator axis. The drive signals that the actual speed is **greater** than the minimum speed set using parameter p2161 (speed threshold value 3).
Additional information	The PLC user program can initiate measures in order to safely end machining and/or to buffer the DC link voltage, e.g. initiate a drive-based and/or control-managed extended stop and retract (ESR).
Corresponds with	Drive parameter p2161 (speed threshold value 3) Drive telegram MELDW.Bit2
Additional information	SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, drive

18.2.6.167 DB31, ... DBX95.7 (alarm of alarm class C is active)

DB31, ... DBX95.7	Alarm of alarm class C is active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The drive signals that an alarm of alarm class C is active.

DB31, ... DBX95.7	Alarm of alarm class C is active
Signal state 0	The drive signals that no alarm of alarm class C is active.
Additional information	In the drive, an alarm is the response to a detected potential or expected fault condition that does not cause the drive to switch off and does not have to be acknowledged.
Additional information	SINAMICS S120 List Manual, "Faults and Alarms"

18.2.6.168 DB31, ... DBX96.2 (master-slave: fine differential speed)

DB31, ... DBX96.2	Master-slave: Fine speed difference
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The speed difference between the master and slave axis lies **within** the tolerance specified with MD37272.
Signal state 0	The speed difference between the master and slave axis lies **outside** the tolerance specified with MD37272.
Corresponds with	DB31, ... DBX24.4 (master-slave: activate torque compensatory controller) DB31, ... DBX24.7 (master-slave: activate coupling) DB31, ... DBX96.3 (master-slave: coarse speed difference) DB31, ... DBX96.7 (master-slave: coupling active) MD37272 $MA_MS_VELO_TOL_FINE (master-slave: velocity tolerance fine)
Additional information	Function Manual Technologies

18.2.6.169 DB31, ... DBX96.3 (master-slave: coarse speed difference)

DB31, ... DBX96.3	Master-slave: Coarse speed difference
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The speed difference between the master and slave axis lies **within** the tolerance specified with MD37270.
Signal state 0	The speed difference between the master and slave axis lies **outside** the tolerance specified with MD37270.
Corresponds with	DB31, ... DBX24.7 (master-slave: activate coupling) DB31, ... DBX96.2 (master-slave: fine differential speed) DB31, ... DBX96.7 (master-slave: coupling active) MD37270 $MA_MS_VELO_TOL_COARSE (master-slave: Speed tolerance coarse)
Additional information	Function Manual Technologies

18.2.6.170 DB31, ... DBX96.4 (master-slave: compensatory controller active)

DB31, ... DBX96.4	Master-slave: Compensatory controller active
Signal flow	NC → PLC
Update	Cyclic

DB31, ... DBX96.4	Master-slave: Compensatory controller active
Signal state 1	Torque compensatory control is active.
Signal state 0	Torque compensatory control is **not** active.
Corresponds with	DB31, ... DBX24.4 (master-slave: activate torque compensatory controller) MD37254 $MA_MS_TORQUE_CTRL_MODE (interconnection torque compensatory controller)
Additional information	Function Manual Technologies

18.2.6.171 DB31, ... DBX96.5 (setpoint switchover: drive control active)

DB31, ... DBX96.5	Setpoint switchover: drive control active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis has taken over control of the drive.
Signal state 0	The axis has **not** taken over control of the drive.
Additional information	The controller can only be enabled (DB31, ... DBX2.1) if the axis has control of the drive: DB31, ... DBX96.5 == 1 **Note** All axes included in setpoint switchover that currently do not have drive control, are switched to follow-up mode by the control. I.e. they are **not in position control**. As a consequence, we recommend that suspended axes are assigned a brake control.
Corresponds with	DB31, ... DBX2.1 (controller enable) DB31, ... DBX24.5 (setpoint switchover: accept drive control)
Additional information	Function Manual Axes and Spindles

18.2.6.172 DB31, ... DBX96.7 (master-slave: coupling active)

DB31, ... DBX96.7	Master/slave: Active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The coupling is active.
Signal state 0	The coupling is not active.
Additional information	**Note** For a master/slave coupling, for the brake control logic of the slave axes, it is no longer permissible that the interface signal DB31, ... DBX61.5 (position controller active) is evaluated, as it is no longer set for an active master/slave coupling. Instead, this interface signal must be used.
Additional information	Function Manual Technologies

18.2.6.173 DB31, ... DBX97.0 (MCS coupling: slave axis)

DB31, …DBX97.0	MCS coupling: Following axis
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The axis is a following axis:
Signal state 0	The axis is **not** a following axis:
Corresponds with	DB31, …DBX24.2 (MCS coupling: deactivate or do not permit) DB31, ... DBX24.3 (MCS coupling: activate collision protection) DB31, ... DBX66.0 (MCS coupling: collision protection active) DB31, ... DBX97.0 (MCS coupling: slave axis) DB31, … DBX97.1 (MCS coupling: coupling active) DB31, ... DBX97.2 (MCS coupling: mirroring active) DB31, ... DBX97.3 (MCS coupling: offset change)
Additional information	Function Manual Technologies

18.2.6.174 DB31, … DBX97.1 (MCS coupling: coupling active)

DB31, …DBX97.1	MCS coupling: Coupling active
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The coupling is active.
Signal state 0	The coupling is not active.
Corresponds with	DB31, …DBX24.2 (MCS coupling: deactivate or do not permit) DB31, ... DBX24.3 (MCS coupling: activate collision protection) DB31, ... DBX66.0 (MCS coupling: collision protection active) DB31, ... DBX97.0 (MCS coupling: slave axis) DB31, … DBX97.1 (MCS coupling: coupling active) DB31, ... DBX97.2 (MCS coupling: mirroring active) DB31, ... DBX97.3 (MCS coupling: offset change)
Additional information	Function Manual Technologies

18.2.6.175 DB31, ... DBX97.2 (MCS coupling: mirroring active)

DB31, …DBX97.2	MCS coupling: Mirroring active
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The "mirroring" function is active.
Signal state 0	The "mirroring" function is **not** active.
Additional information	Displayed only for the CC_slave axis.

DB31, …DBX97.2	MCS coupling: Mirroring active
Corresponds with	DB31, …DBX24.2 (MCS coupling: deactivate or do not permit) DB31, ... DBX24.3 (MCS coupling: activate collision protection) DB31, ... DBX66.0 (MCS coupling: collision protection active) DB31, ... DBX97.0 (MCS coupling: slave axis) DB31, … DBX97.1 (MCS coupling: coupling active) DB31, ... DBX97.2 (MCS coupling: mirroring active) DB31, ... DBX97.3 (MCS coupling: offset change)
Additional information	Function Manual Technologies

18.2.6.176 DB31, ... DBX97.3 (MCS coupling: offset change)

DB31, …DBX97.3	MCS coupling: Offset change
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The offset has changed since activation instant.
Signal state 0	The offset has **not** changed since the activation instant.
Additional information	The interface signal is set, if, for an active 1:1 coupling, the offset saved at the activation instant has changed.
	Note In the reset phase, the interface signal is not set.
Corresponds with	DB31, …DBX24.2 (MCS coupling: deactivate or do not permit) DB31, ... DBX24.3 (MCS coupling: activate collision protection) DB31, ... DBX66.0 (MCS coupling: collision protection active) DB31, ... DBX97.0 (MCS coupling: slave axis) DB31, … DBX97.1 (MCS coupling: coupling active) DB31, ... DBX97.2 (MCS coupling: mirroring active) DB31, ... DBX97.3 (MCS coupling: offset change)
Additional information	Function Manual Technologies

18.2.6.177 DB31, ... DBX98.0 (fine synchronous operation)

DB31, ... DBX98.0	Synchronous operation fine
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Synchronous spindle coupling: The position deviation or velocity difference between the following spindle and its leading spindle is within the "Fine synchronous operation" tolerance band.
Signal state 0	Synchronous spindle coupling: The position deviation or velocity difference between the following spindle and its leading spindle is **not** within the "Fine synchronous operation" tolerance band.

DB31, ... DBX98.0	Synchronous operation fine
Additional information	**Note** The signal is relevant only for the following spindle in synchronous operation. **Application example** Clamping of the workpiece in the following spindle at the transfer from the leading spindle: The clamping of the workpiece is only initiated by the PLC user program when the spindles are operating in adequate synchronism.
Corresponds with	DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation) MD37210 $MA_COUPLE_POS_TOL_FINE (threshold value for "fine synchronism") MD37230 $MA_COUPLE_VELO_TOL_FINE ("fine" speed tolerance)
Additional information	Function Manual Technologies

18.2.6.178 DB31, ... DBX98.1 (coarse synchronous operation)

DB31, ... DBX98.1	Synchronous operation, coarse
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Synchronous spindle coupling: The position deviation or velocity difference between the following spindle and its leading spindle is within the "Coarse synchronous operation" tolerance band.
Signal state 0	Synchronous spindle coupling: The position deviation or velocity difference between the following spindle and its leading spindle is **not** within the "Coarse synchronous operation" tolerance band.
Additional information	**Note** The signal is relevant only for the following spindle in synchronous operation. **Application example** Clamping of the workpiece in the following spindle at the transfer from the leading spindle: The clamping of the workpiece is only initiated by the PLC user program when the spindles are operating in adequate synchronism.
Corresponds with	DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation) MD37200 $MA_COUPLE_POS_TOL_COARSE (threshold value for "coarse synchronism") MD37220 $MA_COUPLE_VELO_TOL_COARSE ("coarse" speed tolerance)
Additional information	Function Manual Axes and Spindles

18.2.6.179 DB31, ... DBX98.2 (actual value coupling)

DB31, ... DBX98.2	Actual value coupling
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Synchronous spindle coupling: Actual value coupling active
Signal state 0	Synchronous spindle coupling: Setpoint coupling active

DB31, ... DBX98.2	Actual value coupling
Additional information	**Note** The signal is relevant only for the active following spindle in synchronous operation. **Special cases or errors** If faults occur on the following spindle that cause a withdrawal of the "Controller enable" for the following spindle, under certain circumstances, the control may exchange the coupling relationship of the following spindle and leading spindle and switch to actual value coupling.
Corresponds with	DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation) MD21310 $MC_COUPLING_MODE_1 (coupling type in synchr. spindle oper.)
Additional information	Function Manual Axes and Spindles

18.2.6.180 DB31, ... DBX98.4 (overlaid movement)

DB31, ... DBX98.4	Overlaid motion
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Synchronous spindle coupling: The following spindle executes an additional motion component that is superimposed on the motion from the coupling with the leading spindle.
Signal state 0	Synchronous spindle coupling: The following spindle does not traverse through any additional motion component - or this has been completed.
Additional information	**Note** The signal is relevant only for the following spindle in synchronous operation. **Application examples** Example for superimposed motion of the following spindle: • Activating the synchronous mode with a defined angular offset between the following spindle and leading spindle. • Activating the synchronous mode for rotating leading spindle. • Changing the ratio while the synchronous mode is active. • Entering a new defined angular offset when the synchronous mode is active. • Traversing the following spindle in the JOG mode with the traversing keys "Plus" / "Minus" or the handwheel with synchronous operation active
Corresponds with	DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation)
Additional information	Function Manual Axes and Spindles

18.2.6.181 DB31, ... DBX98.5 (speed alarm threshold reached)

DB31, ... DBX98.5	Velocity alarm threshold reached
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Electronic gearbox (EG): The velocity of the following axis has reached or exceeded the parameterized velocity alarm threshold.
Signal state 0	Electronic gearbox (EG): The velocity of the following axis is below the parameterized velocity alarm threshold.

DB31, ... DBX98.5	Velocity alarm threshold reached
Additional information	The signal is only relevant when the "Electronic gearbox (EG)" coupling function is active. It is set if the velocity of the following axis, reaches or exceeds the velocity alarm threshold, defined with MD37550 and MD32000 (= percentage of the maximum axis velocity).
Corresponds with	MD37550 $MA_EG_VEL_WARNING (threshold value velocity alarm threshold) MD32000 $MA_MAX_AX_VELO (maximum axis velocity)
Additional information	Function Manual Axes and Spindles

18.2.6.182 DB31, ... DBX98.6 (acceleration alarm threshold reached)

DB31, ... DBX98.6	Acceleration alarm threshold reached
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Electronic gearbox (EG): The acceleration of the following axis has reached or exceeded the parameterized acceleration alarm threshold.
Signal state 0	Electronic gearbox (EG): The acceleration of the following axis is below the parameterized acceleration alarm threshold.
Additional information	The signal is only relevant when the "Electronic gearbox (EG)" coupling function is active. It is set if the acceleration of the following axis, reaches or exceeds the acceleration alarm threshold, defined with MD37550 and MD32300 (= percentage of the maximum axis acceleration).
Corresponds with	MD37550 $MA_EG_VEL_WARNING (threshold value velocity alarm threshold) MD32300 $MA_MAX_AX_ACCEL (maximum axis acceleration)
Additional information	Function Manual Axes and Spindles

18.2.6.183 DB31, ... DBX99.0 (leading spindle active)

DB31, ... DBX99.0	Leading spindle active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Synchronous spindle coupling: The machine axis is presently active as leading spindle.
Signal state 0	Synchronous spindle coupling: The machine axis is presently **not** active as leading spindle.
Additional information	**Note** The signal is relevant only in synchronous operation. **Special cases or errors** If faults occur on the following spindle that cause a withdrawal of the "Controller enable" for the following spindle, under certain circumstances, the control may exchange the coupling relationship of the following spindle and leading spindle and switch to actual value coupling. In this case, the previous leading spindle becomes the new active following spindle (IS "Following spindle active ").
Corresponds with	DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation) DB31, ... DBX99.1 (following spindle active)
Additional information	Function Manual Axes and Spindles

18.2.6.184 DB31, ... DBX99.1 (following spindle active)

DB31, ... DBX99.1	Following spindle active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Synchronous spindle coupling: The machine axis is currently being operated as a following spindle.
Signal state 0	Synchronous spindle coupling: The machine axis is currently **not** being operated as a following spindle.
Additional information	For DB31, ... DBX99.1 == 1, the following spindle follows the motion of the leading spindle corresponding to the ratio. **Note** The signal is relevant only in synchronous operation. **Special cases or errors** If faults occur on the following spindle that cause a withdrawal of the "Controller enable" for the following spindle, under certain circumstances, the control may exchange the coupling relationship of the following spindle and leading spindle and switch to actual value coupling.
Corresponds with	DB31, ... DBX84.4 (active spindle operating mode: Synchronous operation) DB31, ... DBX99.0 (leading spindle active)
Additional information	Function Manual Axes and Spindles

18.2.6.185 DB31, ... DBX99.3 (axis accelerates)

DB31, ... DBX99.3	Axis accelerates
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Electronic gearbox (EG): The acceleration of the following axis has reached or exceeded the parameterized acceleration level, at which the acceleration of the axis is displayed.
Signal state 0	Electronic gearbox (EG): The acceleration of the following axis lies below the parameterized acceleration level, at which the acceleration of the axis is displayed.
Additional information	The signal is only relevant when the "Electronic gearbox (EG)" coupling function is active. It is set if the acceleration of the following axis, reaches or exceeds the acceleration alarm value, defined with MD37560 and MD32300.
Corresponds with	MD37560 $MA_EG_ACC_TOL (threshold value for "Axis accelerates") MD32300 $MA_MAX_AX_ACCEL (maximum axis acceleration)
Additional information	Function Manual Axes and Spindles

18.2.6.186 DB31, ... DBX100.2 (oscillation reversal from external active)

DB31, ... DBX100.2	External oscillation reversal active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The braking phase after "oscillation reversal from external" is active.
Signal state 0	The braking phase after "oscillation reversal from external" is **not** active.

DB31, ... DBX100.2	External oscillation reversal active
Corresponds with	DB31, ... DBX28.0 (oscillation reversal from external)
Additional information	Function Manual Technologies

18.2.6.187 DB31, ... DBX100.3 (oscillation cannot be started)

DB31, ... DBX100.3	Oscillation cannot be started
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The oscillation axis cannot be started.
Signal state 0	Oscillation motion cannot be started.
Additional information	This signal is set if there is a programming error, for example. **Note** This state can also occur if axis was already traversed.
Additional information	Function Manual Technologies

18.2.6.188 DB31, ... DBX100.4 (error during oscillation)

DB31, ... DBX100.4	Error during oscillation
Signal flow	NC → PLC
Signal state 1	Oscillation motion was canceled as an error occurred while oscillating.
Signal state 0	Oscillation is error free.
Additional information	Function Manual Technologies

18.2.6.189 DB31, …DBX100.5 (sparking-out active)

DB31, ... DBX100.5	Sparking-out active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis executes sparking-out strokes.
Signal state 0	The axis is presently **not** executing any sparking-out strokes.
Corresponds with	DB31, ... DBX100.7 (oscillation active)
Additional information	Function Manual Technologies

18.2.6.190 DB31, ... DBX100.6 (oscillation active)

DB31, ... DBX100.6	Oscillation active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis performs an oscillation movement between two reversal points.

DB31, ... DBX100.6	Oscillation active
Signal state 0	The axis is **not** currently oscillating.
Additional information	The signal is irrelevant: DB31, ... DBX100.7 == 0
Corresponds with	DB31, ... DBX100.7 (oscillation active)
Additional information	Function Manual Technologies

18.2.6.191 DB31, ... DBX100.7 (oscillation active)

DB31, ... DBX100.7	Oscillation active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The axis is presently being traversed as oscillating axis.
Signal state 0	The axis is a positioning axis.
Corresponds with	DB31, …DBX100.5 (sparking-out active) DB31, ... DBX100.6 (oscillation active)
Additional information	Function Manual Technologies

18.2.6.192 DB31, ... DBX102.5 (position measuring system 1 activated)

DB31, ... DBX102.5	Position measuring system 1 activated
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Position measuring system 1 is in the "active" or "passive" state. Monitoring and updating the position measuring system is **activated**.
Signal state 0	Position measuring system 1 is in "Park" state. Monitoring and updating the position measuring system is **deactivated**.
Corresponds with	DB31, ... DBX1.6 (position measuring system 2)
Additional information	Function Manual Monitoring and Compensating

18.2.6.193 DB31, ... DBX102.6 (position measuring system 2 activated)

DB31, ... DBX102.6	Position measuring system 2 activated
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Position measuring system 2 is in the "active" or "passive" state. Monitoring and updating the position measuring system is **activated**.
Signal state 0	Position measuring system 21 is in the "Park" state. Monitoring and updating the position measuring system is **deactivated**.
Corresponds with	DB31, ... DBX1.5 (position measuring system 1)
Additional information	Function Manual Monitoring and Compensating

18.2.6.194 DB31, ... DBX104.0 - 107.6 (active infeed axes)

DB31, ... DBX104.0 - 107.6:	Active infeed axes					
Signal flow	NC → PLC					
Update	Cyclic					
Signal state 1	For the actual axis (oscillating axis), the corresponding axis is an infeed axis.					
Signal state 0	For the actual axis (oscillating axis), the corresponding axis is **not** an infeed axis.					
Additional information	**DBB**	**Bit 7**	**Bit 6**	**...**	**Bit 1**	**Bit 0**
	104	Axis 8	Axis 7	...	Axis 2	Axis 1
	105	Axis 16	Axis 15	...	Axis 10	Axis 9
	106	Axis 24	Axis 23	...	Axis 18	Axis 17
	107	---	Axis 31	...	Axis 26	Axis 25
Corresponds with	DB31, ... DBX100.7 (oscillation active)					
Additional information	Function Manual Technologies					

18.2.6.195 DB31, ... DBX128.0 (suppress program test)

DB31, ... DBX128.0	Suppress program test
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Suppression of the program test (setpoint output locked) is requested by the HMI.
Signal state 0	Suppression of the program test (setpoint output locked) is **not** requested by the HMI.
Additional information	If function "Program test" (PRT) is active in the channel to which the axis currently belongs, the axis is traversed with internal axis disable. Setpoints are generated but they not output to the machine axes. Actual value = Setpoint. If the program test is suppressed for the axis, setpoints are output to the machine axis despite the function "program test" being active in the channel to which the axis currently belongs. **Automatic transfer of the interface signals** The HMI request signals DB21, ... DBX128.0 / .1 are only transferred from the basic PLC program to the PLC request signals DB21, ... DBX14.0 / .1 if FB1 parameter **MMCToIF** is set to **TRUE**. If the parameter is not set, the PLC request signals must be set by the PLC user program.
Corresponds with	DB31, ... DBX14.0 (suppress program test) DB31, ... DBX14.1 (activate program test) DB31, ... DBX128.1 (activate program test)
Additional information	Function Manual Basic Functions

18.2.6.196 DB31, ... DBX128.1 (activate program test)

DB31, ... DBX128.1	Activate program test
Signal flow	HMI → PLC
Update	Cyclic
Signal state 1	Activation of the function "program test" (setpoint output locked) is requested by the HMI.

DB31, ... DBX128.1	Activate program test
Signal state 0	Activation of the function "program test" (setpoint output locked) is **not** requested by the HMI.
Additional information	If the function "program test" is active, the axis is traversed with internal axis disable. Setpoints are generated but they not output to the machine axes. Actual value = Setpoint. **Automatic transfer of the interface signals** The HMI request signals DB21, ... DBX128.0 / .1 are only transferred from the basic PLC program to the PLC request signals DB21, ... DBX14.0 / .1 if FB1 parameter **MMCToIF** is set to **TRUE**. If the parameter is not set, the PLC request signals must be set by the PLC user program.
Corresponds with	DB31, ... DBX14.0 (suppress program test) DB31, ... DBX14.1 (activate program test) DB31, ... DBX128.0 (suppress program test)
Additional information	Function Manual Basic Functions

18.2.6.197 DB31, ... DBX130.0 - 4 (motor/drive data set: formatting)

DB31, ... DBX130.0 - 4	Motor/drive data set: Formatting
Signal flow	NC → PLC
Update	Cyclic
Additional information	Formatting interface for the currently active motor (MDS) / drive data set (DDS).
Corresponds with	DB31, ... DBX21.0 - 4 (motor/drive data set: selection) DB31, ... DBX21.5 (motor has been selected) DB31, ... DBX93.0 - 4 (motor/drive data set: display)
Additional information	• Function Manual Basic Functions • SINUMERIK 840D sl Commissioning Manual, CNC Commissioning: NC, PLC, Drive

18.2.6.198 DB31, ... DBX132.0 (sensors available)

DB31, ... DBX132.0	Sensors available
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The sensor required for spindles with SMI 24 is available.
Signal state 0	The sensor required for spindles with SMI 24 is **not** available.
Additional information	**Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Corresponds with	DB31, … DBX132.1: Sensor S1 (clamped state) is available DB31, … DBX132.4: Sensor S4 is available DB31, … DBX132.5: Sensor S5 is available
Additional information	Function Manual Axes and Spindles

18.2.6.199 DB31, ... DBX132.1 (sensor S1 available (clamped state))

DB31, ... DBX132.1	Sensor S1 available (clamped state)
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Sensor S1 is available.
Signal state 0	Sensor S1 is **not** available.
Additional information	**Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Corresponds with	DB31, … DBW134 (state of the clamping system) DB31, … DBW136 (analog value: clamped state)
Additional information	Function Manual Axes and Spindles

18.2.6.200 DB31, … DBX132.4 (sensor S4 available (piston end position))

DB31, ... DBX132.4	Sensor S4 available (piston end position)
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Sensor S4 is available.
Signal state 0	Sensor S4 is **not** available.
Additional information	**Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Corresponds with	DB31, … DBX138.4 (sensor S4: piston end position)
Additional information	Function Manual Axes and Spindles

18.2.6.201 DB31, ... DBX132.5 (sensor S5 available (angular position of the motor shaft))

DB31, ... DBX132.5	Sensor S5 available (angular position of the motor shaft)
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	Sensor S5 is available.
Signal state 0	Sensor S5 is **not** available.
Additional information	**Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Corresponds with	DB31, … DBX138.5 (sensor S5: angular position of the motor shaft)
Additional information	Function Manual Axes and Spindles

18.2.6.202 DB31, ... DBX133.2 (state value is generated, speed limitation p5043 is active)

DB31, ... DBX133.2	State value is generated, speed limitation p5043 is active
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The status value is generated and the speed limits from drive parameter p5043 are active.
Signal state 0	The status value is not generated and the speed limits from drive parameter p5043 are not active.
Additional information	**Note** • Only relevant for spindles with SMI 24 (Weiss spindle) • When generating the status value, the analog voltage values of sensor S1 are transformed into discrete status values of drive parameter r5001.
Corresponds with	DB31, … DBX134 (clamped state) Drive parameters: r5001 System variable: $VA_MOT_CLAMPING_STATE[<axis>] OPI variable: vaMotClampingState
Additional information	Function Manual Axes and Spindles

18.2.6.203 DB31, … DBW134 (state of the clamping system (sensor S1))

DB31, ... DBW134	State of the clamping system (sensor S1)	
Signal flow	NC → PLC	
Update	Cyclic	
Additional information	**Note** Only relevant for spindles with SMI 24 (Weiss spindle)	
	Depending on the position of the clamping device, sensor S1 supplies an analog voltage value. To simplify the evaluation of the clamped state, the analog voltage of sensor module SMI 24 is converted into a state value. The state values correspond to certain voltage ranges. The voltage ranges can be set via: Drive parameter p5041[0...5].	
	State value	**Clamped state**
	0	Sensor S1 not available or state values inactive
	1	State initialization running
	2	Released with signal (error state)
	3	Released
	4	Clamping with tool
	5	Releasing with tool
	6	Releasing without tool
	7	Clamped with tool AND S4 == 0
	8	Clamped with tool AND S4 == 1
	9	Clamping without tool
	10	Clamped without tool
	11	Clamped with signal (error state)

DB31, ... DBW134	State of the clamping system (sensor S1)
Corresponds with	DB31, … DBW136 (analog value: clamped state) Drive parameters: p5041[0...5], p5043[0...6]
Additional information	Function Manual Axes and Spindles

18.2.6.204 DB31, ... DBW136 (analog measured value: of the clamping system)

DB31, ... DBW136	Analog measured value: of the clamping system
Signal flow	NC → PLC
Update	Cyclic
Additional information	**Note** Only relevant for spindles with SMI 24 (Weiss spindle)
	Sensor S1 supplies an analog voltage value: 0 - 10 V. The analog value of the clamped state is mapped to: 0 - 10000 increments, resolution 1 mV **Note** SIMATIC S7 input module: 0 - 27648 increments, resolution 0.36 mV Adaptation factor if you change to a spindle with SMI 24: 2.7648
Corresponds with	DB31, … DBW134 (clamped state) Drive parameters: p5041[0...5], p5043[0...6]
Additional information	Function Manual Axes and Spindles

18.2.6.205 DB31, … DBX138.4 (sensor S4, piston end position)

DB31, ... DBX138.4	Sensor S4, piston end position
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The piston is in position, i.e. the piston is free to move
Signal state 0	The piston is not in position
Additional information	**Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Corresponds with	DB31, … DBX132.4 (sensor S4 available)
Additional information	Function Manual Axes and Spindles

18.2.6.206 DB31, ... DBX138.5 (sensor S5 angular position of the motor shaft)

DB31, ... DBX138.5	Sensor S5, angular position of the motor shaft
Signal flow	NC → PLC
Update	Cyclic
Signal state 1	The motor shaft is in position (requirement: the spindle is stationary)
Signal state 0	The motor shaft is not aligned

DB31, ... DBX138.5	Sensor S5, angular position of the motor shaft
Additional information	**Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Corresponds with	DB31, … DBX132.5 (sensor S5 available)
Additional information	Function Manual Axes and Spindles

18.2.7 DB71. Tool management, loading/unloading points

18.2.7.1 DB71 DBX0.0 - 1.7 (active status of interfaces 1 - 16)

DB71 DBX0.0 - 1.7	Active status of interfaces 1- 16							
Signal flow	NC → PLC							
Update	Job-controlled							
Signal state 1	The interface is active.							
Signal state 0	The interface is **not** active.							
Additional information	If an interface is active, then a valid data set exists for this interface. Assignment of the interfaces:							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	DBB0	INT 8	INT 7	INT 6	INT 5	INT 4	INT 3	INT 2	INT 1
	DBB1	INT 16	INT 15	INT 14	INT 13	INT 12	INT 11	INT 10	INT 9

18.2.7.2 DB71 DBX2.0 - 3.7 ("auto" acknowledgment of interfaces 1 - 16)

DB71 DBX2.0 - 3.7	"auto" acknowledgment of interfaces 1 - 16
Signal flow	NC → PLC
Update	Job-controlled

Signal state 0 → 1	When setting (edge) the bit, the active command is acknowledged with status_1. Unless it is a relocation operation from a real magazine to a buffer. Then acknowledgment is made with status_6. This reserves the source location.
	e. g.
	`U DB71 DBX0.0`
	`U DB71 DBX4.1 (unload)`
	`S DB71 DBX2.0`

Signal state 1 → 0	Operation for this interface has ended. Is reset by the basic program.

18.2.7.3 DB71 DBX(n+0).0 (command: Loading)

DB71 DBX(n+0).0	Command: Loading
Signal flow	NC → PLC
Update	Job-controlled

DB71 DBX(n+0).0	Command: Loading
Signal state 1	Loading operation for a tool is initiated. The magazine location into which the tool is to be loaded is defined in DBW(n+26). The loading point in question is the location number of the loading point. It is also located in DBW(n+18)
Additional information	The bits in DBB(n+0) (loading, unloading,....) are not updated by the basic program until a new task exists for this interface. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". If required, the user can reset bits DBB(n+0).
Additional information	**Start addresses** The start addresses n of the loading/unloading points are 1 - 4: • Loading/unloading point 1: n = 4 • Loading/unloading point 2: n = 34 • Loading/unloading point 3: n = 64 • Loading/unloading point 4: n = 94
Corresponds with	DB71 DBX(n+16) and (n+18) and (n+26)

18.2.7.4 DB71 DBX(n+0).1 (command: Unload)

DB71 DBX(n+0).1	Command: Unload
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Unloading operation is initiated for a tool; the magazine location from which it is to be unloaded is in DBW (n+20) and DBW (n+22). The number of the unloading point is in DBW(n+18).
Edge evaluation	Signal(s) updated: Conditional
Additional information	The bits in DBB(n+0) (loading, unloading,....) are not updated by the basic program until a new task exists for this interface. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". If required, the user can reset bits DBB(n+0).
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94
Corresponds with	DB71 DBX(n+16) and (n+18) or (n+20) and (n+22)

18.2.7.5 DB71 DBX(n+0).2 (command: Relocate)

DB71 DBX(n+0).2	Command: Relocate
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Relocation operation for a tool is initiated. From magazine/location (n+20, n+22=source) to magazine/location (n+24, n+26=target)

DB71 DBX(n+0).2	Command: Relocate
Signal state 0	Relocation operation for a tool is not initiated.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94 The bits in DBB(n+0) (loading, unloading,....) are not updated by the basic program until a new task exists for this interface. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". If required, the user can reset bits DBB(n+0).

18.2.7.6 DB71 DBX(n+0).3 (command: Position to load point)

DB71 DBX(n+0).3	Command: Position to the loading point
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	A magazine location is to be positioned at the loading point (magazine No. 9999). The magazine location that is to be moved to the loading point is in DB71.DBW(n+20) and (n+22). The loading location is in DB71.DBWn+18.
Signal state 0	A magazine location is not to be positioned at the loading point (magazine No. 9999).
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94 The bits in DBB(n+0) (loading, unloading,....) are not updated by the basic program until a new task exists for this interface. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". If required, the user can reset bits DBB(n+0).

18.2.7.7 DB71 DBX(n+0).4 (command: Job/task comes from the NC program)

DB71 DBX(n+0).4	Command: Task comes from the NC program
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The signal is set if the task is received from the part program or via a cycle. 1. Positioning task with the language command POSM 2. A relocation task or tool transport uses the language command MVTOOL 3. Positioning a multitool using the language command POSMT

DB71 DBX(n+0).4	Command: Task comes from the NC program
Signal state 0	The signal is not set.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94 The bits in DBB(n+0) (loading, unloading,....) are not updated by the basic program until a new task exists for this interface. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". If required, the user can reset bits DBB(n+0).

18.2.7.8 DB71 DBX(n+0).5 (command: Position a multitool)

DB71 DBX(n+0).5	Command: Position a multitool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Positioning a multitool has been initiated. This can be realized using a language command, PI service or the HMI user interface.
Signal state 0	The signal is not set.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94 The bits in DBB(n+0) (loading, unloading,....) are not updated by the basic program until a new task exists for this interface. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". If required, the user can reset bits DBB(n+0).

18.2.7.9 DB71 DBX(n+1).0 ("auto" acknowledgment negative)

DB71 DBX(n+1).0	"auto" acknowledgment negative
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The bit is only evaluated in conjunction with "auto" ackn. If it is set, then the auto acknowledge is negative, i.e. with status_3.
Signal state 0	The bit is not set.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94

18.2.7.10 DB71 DBX(n+1).7 (command: data in the extended range)

DB71 DBX(n+1).7	Command: Data in the extended range
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The signal is set if a multitool is loaded, unloaded or positioned. The multitool data are then available in DB1071.
Signal state 0	The signal is not set.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94

18.2.7.11 DB71 DBB(n+2) (assigned channel)

DB71 DBB(n+2)	Assigned channel
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Number of the channel for which the active interface is valid.
Signal state 0	No channel number
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94

18.2.7.12 DB71 DBB(n+3) (tool management No.)

DB71 DBB(n+3)	Tool management No.
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Associated tool management number; corresponds to the number of the TO unit within a TO area.
Edge evaluation	Signal(s) updated: Conditional
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94

18.2.7.13 DB71 DBW(n+16) (identifier for loading/unloading point (fixed value 9999))

DB71 DBW(n+16)	Identifier for loading/unloading point (fixed value 9999)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The identifier for the loading/unloading point is fixed to the value 9999.
Signal state 0	The identifier is not defined.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94

18.2.7.14 DB71 DBW(n+18) (location number of the loading/unloading point)

DB71 DBW(n+18)	Location No. of the loading/unloading point
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The location No. of the loading/unloading point is displayed.
Signal state 0	The location No. of the loading/unloading point is not displayed.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94

18.2.7.15 DB71 DBW(n+20) (Magazine No. (Source) for unloading/relocating/positioning)

DB71 DBW(n+20)	Magazine No. (Source) for unloading/relocating/positioning
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Unloading: Magazine from which the tool is to be unloaded Relocating: Magazine from which the tool is received Positioning: Magazine that is to be positioned
Signal state 0	No magazine is unloaded, relocated or positioned.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94
Corresponds with	DBW(n+22)

18.2.7.16 DB71 DBW(n+22) (Location No. (Source) for unloading/relocating/positioning)

DB71 DBW(n+22)	Location No. (Source) for unloading/relocating/positioning
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Unloading: Location from which the tool is to be unloaded Relocating: Location from where the tool is received Positioning: Location that should be positioned
Signal state 0	No tool is unloaded, relocated, positioned from any location.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94
Corresponds with	DBW(n+20)

18.2.7.17 DB71 DBW(n+24) (Magazine No. (Target) for unloading/relocating/positioning)

DB71 DBW(n+24)	Magazine No. (Target) for unloading/relocating/positioning
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Loading: Magazine into which the tool is to be loaded Relocating: Magazine into which the tool is loaded: Positioning: Magazine to be positioned to - tool remains in the original magazine Only meaningful for interface 1. If values other than 0 are entered here, the data define the magazine or location for positioning (language command POSM).
Signal state 0	No magazine number specified.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94
Corresponds with	DBW(n+26)

18.2.7.18 DB71 DBW(n+26) (location No. (Target) for unloading/relocating/positioning)

DB71 DBW(n+26)	Location No. (Target) for unloading/relocating/positioning
Signal flow	NC → PLC
Update	Job-controlled

DB71 DBW(n+26)	Location No. (Target) for unloading/relocating/positioning
Signal state 1	Loading: Location into which the tool is to be loaded Relocating: Location into which the tool comes Positioning: Location at which the tool must be positioned. Tool remains at original location Only meaningful for interface 1. If values other than 0 are entered here, the data define the magazine or location for positioning (language command POSM).
Signal state 0	No location number specified for loading, relocating, positioning.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94
Corresponds with	DBW(n+24)

18.2.7.19 DB71 DBX(n+28).0 (loading/unloading without any magazine movement)

DB71 DBX(n+28).0	Loading/unloading without magazine movement
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	HMI / Jobshop sets this signal when requested by the operator. If the bit is active, there must be no traversing motion of the magazine, only a mechanical unlocking/locking of the location. The load/unload command must be acknowledged after the action. For a positioning and relocating request, this signal is not valid for a traversing motion.
Signal state 0	HMI / Jobshop delete this signal when requested by the operator.
Additional information	Starting addresses of loading/unloading points: Loading/unloading point 1: n = 4 Loading/unloading point 2: n = 34 Loading/unloading point 3: n = 64 Loading/unloading point 4: n = 94

18.2.8 DB72: Tool management, change in the spindle

18.2.8.1 DB72 DBX0.0-1.7 (active status of interfaces 1-16)

DB72 DBX0.0-1.7	Active status of interfaces 1-16
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Associated interface has a valid data set, a tool change request or tool preparation has been initiated.

DB72 DBX0.0-1.7	Active status of interfaces 1-16
Signal state 0	Operation for this interface has ended. Is reset by FC8/FC6.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.2 DB72 DBX2.0-3.7 ("auto" acknowledgment of interfaces 1-16)

DB72 DBX2.0-3.7	"auto" acknowledgment of interfaces 1-16
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	When setting (edge) the bit, the active command is acknowledged with status_1.
Signal state 0	Operation for this interface has ended. Is reset by the basic program.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.3 DB72 DBX(n+0).0 (command code: Obligatory change)

DB72 DBX(n+0).0	Command code: Obligatory change
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The new tool is fixed-location-coded.
Signal state 1	The new tool is fixed-location-coded.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100 The bit in DBB (n+0).2 (prepare change) is not reset by the system with a change command. The bits in DBB(n+0) ... are current only if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary. If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

18.2.8.4 DB72 DBX(n+0).1 (command code: Perform change with M06)

DB72 DBX(n+0).1	Command code: Perform change with M06
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	M06 command was programmed for tool change, the tool change can now take place

DB72 DBX(n+0).1	Command code: Perform change with M06
Signal state 0	M06 command for tool change was not programmed.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100 The bit in DBB(n+0).2 (prepare change) is not reset by the system with a change command. The bits in DBB(n+0) are up to date only if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary. If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

18.2.8.5 DB72 DBX(n+0).2 (command code: Prepare change)

DB72 DBX(n+0).2	Command code: Prepare change
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Prepare new tool for change. If necessary, move location for old tool to spindle.
Signal state 0	No new tool is ready.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100 The bit in DBB (n+0).2 (prepare change) is not reset by the system with a change command. The bits in DBB(n+0) ... are current only if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary. If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

18.2.8.6 DB72 DBX(n+0).3 (command code: T0)

DB72 DBX(n+0).3	Command code: T0
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Indicates that T0 has been programmed (no-load spindle traversing).

DB72 DBX(n+0).3	Command code: T0
Signal state 0	Does not indicate that T0 has been programmed.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100 The bit in DBB (n+0).2 (prepare change) is not reset by the system with a change command. The bits in DBB(n+0) ... are current only if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary. If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

18.2.8.7 DB72 DBX(n+0).4 (command code: Old tool in buffer)

DB72 DBX(n+0).4	Command code: Old tool in buffer
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The buffer number of the tool to be changed is in DB72.DBW (n+42)
Signal state 0	The buffer number of the tool to be changed is not in DB72.DBW (n+42)
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100 The bit in DBB(n+0).2 (prepare change) is not reset by the system with a change command. The bits in DBB(n+0) ... are current only if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary. If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

18.2.8.8 DB72 DBX(n+0).5 (command code: Load manual tool)

DB72 DBX(n+0).5	Command code: Load manual tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	A manual tool is to be loaded. The HMI displays the tool which is to be loaded.
Signal state 0	No new manual tool is loaded.

DB72 DBX(n+0).5	Command code: Load manual tool
Edge evaluation	Signal(s) updated: Conditional
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100 The bit in DBB(n+0).2 (prepare change) is not reset by the system with a change command. The bits in DBB(n+0) ... are current only if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary. If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

18.2.8.9 DB72 DBX(n+0).6 (command code: Unload manual tool)

DB72 DBX(n+0).6:	Command code: Unload manual tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The tool is to be changed via manual operation.
Signal state 0	The tool is not to be changed via manual operation.
Edge evaluation	Signal(s) updated: Conditional
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100 The bit in DBB(n+0).2 (prepare change) is not reset by the system with a change command. The bits in DBB(n+0) ... are current only if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary. If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

18.2.8.10 DB72 DBX(n+0).7 (command code: Tool remains in spindle)

DB72 DBX(n+0).7	Command code: Tool remains in spindle
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The bit is set at change from spindle to spindle. Initiated, e.g. by reset and start mode or block search.
Signal state 0	The bit is not set at change from spindle to spindle.

DB72 DBX(n+0).7	Command code: Tool remains in spindle
Edge evaluation	Signal(s) updated: Conditional
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100 The bit in DBB(n+0).2 (prepare change) is not reset by the system with a change command. The bits in DBB(n+0) ... are current only if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary. If DBX(n+0).1 and DBX(n+0).2 are present at the same time, it means that T and M06 were programmed in one block.

18.2.8.11 DB72 DBX(n+1).0 ("auto" acknowledgment negative)

DB72 DBX(n+1).0	"auto" acknowledgment negative
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The bit is only evaluated in conjunction with "auto" ackn. If it is set, the auto acknowledge is negative, i.e. with status_3.
Signal state 0	Bit is not evaluated.
Edge evaluation	Signal(s) updated: Conditional
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.12 DB72 DBX(n+1).7 (command: data in the extended range)

DB72 DBX(n+1).7	Command: Data in the extended range
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The signal is set if a tool is prepared in a multitool or is loaded into a multitool. DB72 contains the data of the multitool, the data of the selected tool is available in DB1072. For T0 the signal is not set.
Signal state 0	The signal is not set.

18.2.8.13 DB72 DBB(n+2) (assigned channel)

DB72 DBB(n+2)	Assigned channel
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Number of the channel for which the active interface applies.

DB72 DBB(n+2)	Assigned channel
Signal state 0	There is no assignment.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.14 DB72 DBB(n+3) (tool management No.)

DB72 DBB(n+3)	Tool management No.
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Associated tool management No. (TO area).
Signal state 0	No associated tool management No.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.15 DB72 DBD(n+4) (user-definable parameter 0 (DInt))

DB72 DBD(n+4)	User-definable parameter 0 (DInt)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	If a value has to be sent to the PLC via the part program, the transfer can be programmed with $P_VDITCP[0]. Values are transferred with a T call.
Signal state 0	No value is transferred to the PLC via the part program.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.16 DB72 DBD(n+8) (user-definable parameter 1 (DInt))

DB72 DBD(n+8)	User-definable parameter 1 (DInt)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[1].

DB72 DBD(n+8)	User-definable parameter 1 (DInt)
Signal state 0	No value is transferred to the PLC via the part program
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.17 DB72 DBD(n+12) (user-definable parameter 2 (DInt))

DB72 DBD(n+12)	User-definable parameter 2 (DInt)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[2]
Signal state 0	No value is transferred to the PLC via the part program
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.18 DB72 DBW(n+16) (buffer magazine number (fixed value 9998))

DB72 DBW(n+16)	Buffer magazine no. (fixed value 9998); target position for new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Magazine no. 9998 (buffer magazine); target magazine for new tool.
Signal state 0	There is no buffer magazine.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.19 DB72 DBW(n+18) (location in the buffer magazine (spindle))

DB72 DBW(n+18)	Location in buffer magazine (spindle)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location no. of buffer magazine to which the new tool must be loaded. This is normally the spindle. The location number defined for this particular buffer during commissioning is output.

DB72 DBW(n+18)	Location in buffer magazine (spindle)
Signal state 0	No location No. is output.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.20 DB72 DBW(n+20) (magazine No. (source) for new tool to be loaded)

DB72 DBW(n+20)	Magazine No. (source) for new tool to be loaded
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	No. of magazine from which the new spindle tool comes.
Signal state 0	No magazine No. is output.
Corresponds with	DBW(n+22)
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.21 DB72 DBW(n+22) (location No. (source) for new tool)

DB72 DBW(n+22)	Location No. (source) for new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location no. of the magazine from which the new spindle tool comes.
Signal state 0	No location No. is output.
Corresponds with	DBW(n+20)
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.22 DB72 DBW(n+24) (magazine No. (target) for old tool to be removed)

DB72 DBW(n+24)	Magazine No. (target) for old tool to be removed
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Number of magazine in which the tool to be removed at change will be placed.
Signal state 0	No magazine No. is output.

DB72 DBW(n+24)	Magazine No. (target) for old tool to be removed
Corresponds with	DBW(n+26)
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.23 DB72 DBW(n+26) (location No. (target) for old tool)

DB72 DBW(n+26)	Location No. (target) for old tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Magazine location for tool that is unloaded at change.
Signal state 1	No magazine location for tool that is unloaded at change.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.24 DB72 DBW(n+28) (new tool: Location type)

DB72 DBW(n+28):	New tool: Location type
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The location type of the new spindle tool is entered here.
Signal state 0	No location type entered.
Corresponds with	Tool size: Left, right, top, bottom.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.25 DB72 DBW(n+30) (new tool: Size left)

DB72 DBW(n+30)	New tool: Size left
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size **left** in half locations for the new spindle tool.

DB72 DBW(n+30)	New tool: Size left
Signal state 0	No tool size is specified.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.26 DB72 DBW(n+32) (new tool: Size right)

DB72 DBW(n+32):	New tool: Size right
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size **right** in half locations for the new spindle tool.
Edge evaluation	Signal(s) updated: Conditional
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.27 DB72 DBW(n+34) (new tool: Size top)

DB72 DBW(n+34)	New tool: Size top
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size **top** in half locations for the new spindle tool.
Signal state 0	No tool size is specified.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.28 DB72 DBW(n+36) (new tool: Size bottom)

DB72 DBW(n+36)	New tool: Size bottom
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size **bottom** in half locations for the new spindle tool.

DB72 DBW(n+36)	New tool: Size bottom
Signal state 0	No tool size is specified.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.29 DB72 DBW(n+38) (tool status for new tool)

DB72 DBW(n+38)	Tool status for new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Bit 0: Tool in the buffer Bit 1: disabled, but ignore Bit 2: Tool to be unloaded Bit 3: Tool to be loaded Bit 4: Master tool Bit 5: Reserved Bit 6: 1:1 exchange Bit 7: Manual tool Bit 8: Active tool Bit 9: Enable tool Bit 10: Tool disabled Bit 11: Measure tool Bit 12: Prewarning limit reached Bit 13: Tool is being changed Bit 14: Tool is fixed-location-coded Bit 15: Tool was in use
Signal state 0	No tool status specified.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.30 DB72 DBW(n+40) (new tool: internal T number of NC)

DB72 DBW(n+40)	New tool: Internal T number of the NC
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Display of internal T number of NC for the new spindle tool.

DB72 DBW(n+40)	New tool: Internal T number of the NC
Signal state 0	No internal T number displayed.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.31 DB72 DBW(n+42) (buffer location of the old tool)

DB72 DBW(n+42)	Buffer location of old tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	If DB72 (n+0.4) = 1, the buffer location of the old tool must be entered here. This can be any buffer magazine (also a gripper).
Signal state 0	No buffer magazine location specified.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.32 DB72 DBW(n+44) (original magazine of the new tool)

DB72 DBW(n+44)	Original magazine of new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Owner magazine of the new tool Corresponds to NC variables $A_MYMN[T no.] If the new tool is located in the magazine, then this value is identical with DB72 DBW(n+20). If the new tool is in the buffer magazine (e.g. gripper) then the magazine no. is entered here from where the tool originally came.
Signal state 0	No original magazine specified.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.8.33 DB72 DBW(n+46) (original location of the new tool)

DB72 DBW(n+46)	Original location of new tool
Signal flow	NC → PLC
Update	Job-controlled

DB72 DBW(n+46)	Original location of new tool
Signal state 1	Owner location of the new tool Corresponds to NC variables $A_MYMLN[T no.]
Signal state 0	No original location specified.
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.9 DB73: Tool management, change in turret

18.2.9.1 DB73 DBX0.0 - 1.7 (active status of interfaces 1 - 16)

DB73 DBX0.0 - 1.7	Active status of interfaces 1-16
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Associated interface has a valid data set.
Signal state 0	Operation for this interface has ended. Is reset by FC7.
Edge evaluation	Signal(s) updated: Conditional
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.2 DB73 DBX2.0 - 3.7 ("auto" acknowledgment of interfaces 1 - 16)

DB73 DBX2.0 - 3.7	"auto" acknowledgment of interfaces 1-16
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	When setting (edge) the bit, the active command is acknowledged with status_1.
Signal state 0	Operation for this interface has ended. Is reset by the basic program.
Edge evaluation	Signal(s) updated: Conditional
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.3 DB73 DBX(n+0).0 (command code: Obligatory change)

DB73 DBX(n+0).0	Command: Obligatory change
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The new tool is fixed-location-coded.
Signal state 0	The new tool is not fixed-location-coded.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92 The bits in DBB(n+0) (obligatory change, execute change, ...) are not reset by the system. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary.
Corresponds with	Position of participating tools

18.2.9.4 DB73 DBX(n+0).1 (command code: Execute change)

DB73 DBX(n+0).1	Command: Perform change
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Execute tool change.
Signal state 0	Do not change tool.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92 The bits in DBB(n+0) (obligatory change, execute change, ...) are not reset by the system. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary.

18.2.9.5 DB73 DBX(n+0).3 (T0))

DB73 DBX(n+0).3	T0
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Indicates that T0 has been programmed.

DB73 DBX(n+0).3	T0
Signal state 0	Does not indicate that T0 has been programmed.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92 The bits in DBB(n+0) (obligatory change, execute change, ...) are not reset by the system. They are only current (up-to-date) if the corresponding interface bit in DBB0 is set to "1". However, the bits can be reset by the user if necessary.

18.2.9.6 DB73 DBX(n+1).0 ("auto" acknowledgment negative)

DB73 DBX(n+1).0	"auto" acknowledgment negative
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The bit is only evaluated in conjunction with "auto" ackn. If it is set, then the auto acknowledge is negative, i.e. with status_3.
Signal state 0	The bit is not set.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.7 DB73 DBX(n+1).7 (command: data in the extended range)

DB73 DBX(n+1).7	Command: Data in the extended range
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The signal is set if a tool is loaded into a multitool. DB73 contains the data of the multitool, the data of the selected tool is available in DB1073. For T0 the signal is not set.
Signal state 0	The signal is not set if no tool is loaded into a multitool.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.8 DB73 DBB(n+2) (assigned channel)

DB73 DBB(n+2)	Assigned channel
Signal flow	NC → PLC
Update	Job-controlled

DB73 DBB(n+2)	Assigned channel
Signal state 1	Number of channel from which the T word was programmed.
Signal state 0	Number of the channel is not assigned.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.9 DB73 DBB(n+3) (tool management No.)

DB73 DBB(n+3)	Tool management number
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Associated tool management number (TO area) of channel.
Signal state 0	Tool management is not assigned.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.10 DB73 DBD(n+4) (user-definable parameter 0 (DInt))

DB73 DBD(n+4)	User-definable parameter 0 (DInt)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[0] = value. Parameters 0-2 are transferred with the T command.
Signal state 0	No value is transferred to the PLC.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.11 DB73 DBD(n+8) (user-definable parameter 1 (DInt))

DB73 DBD(n+8)	User-definable parameter 1 (DInt)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[1] = value.

DB73 DBD(n+8)	User-definable parameter 1 (DInt)
Signal state 0	No value is transferred to the PLC
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.12 DB73 DBD(n+12) (user-definable parameter 2 (DInt))

DB73 DBD(n+12):	User-definable parameter 2 (DInt)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	If you need to send a value to the PLC via the part program, the transfer can be programmed with $P_VDITCP[2] = value.
Signal state 0	No value is transferred to the PLC
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.13 DB73 DBW(n+20) (magazine number of the new tool)

DB73 DBD(n+20)	Magazine no. of new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Magazine no. of the new tool, which should be processed.
Signal state 0	There is no magazine number of the new tool.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92
Corresponds with	DBW(n+22)

18.2.9.14 DB73 DBW(n+22) (location number of the new tool to be loaded)

DB73 DBD(n+22)	Location no. of new tool to be loaded at change
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location no. of the new tool, which should be processed.
Signal state 0	There is no location number of the new tool.

DB73 DBD(n+22)	Location no. of new tool to be loaded at change
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92
Corresponds with	DBW(n+20)

18.2.9.15 DB73 DBW(n+24) (magazine No. (target) for old tool to be removed)

DB73 DBW(n+24)	Magazine No. (target) for old tool to be removed
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Number of magazine in which the tool to be removed at change will be placed.
Signal state 0	No magazine No. is output.
Corresponds with	DBW(n+26)
Additional information	Start address of the spindle: Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

18.2.9.16 DB73 DBW(n+26) (location number of the old tool to be removed)

DB73 DBD(n+26)	Location no. of old tool to be unloaded
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location no. of the old tool (used up to now for machining).
Signal state 0	No location number specified.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.17 DB73 DBW(n+28) (new tool: Location type)

DB73 DBD(n+28)	New tool: Location type
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	The location type of the new tool is entered here.
Signal state 0	No location type of the new entry entered.

DB73 DBD(n+28)	New tool: Location type
Corresponds with	Tool size: Left, right, top, bottom
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.18 DB73 DBW(n+30) (new tool: Size left)

DB73 DBD(n+30)	New tool: Size left
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size **left** in half locations for the new tool.
Signal state 0	No tool size is specified.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.19 DB73 DBW(n+32) (new tool: Size right)

DB73 DBD(n+32)	New tool: Size right
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size **right** in half locations for the new tool
Signal state 1	No tool size is specified.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.20 DB73 DBW(n+34) (new tool: Size top)

DB73 DBD(n+34)	New tool: Size top
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size **top** in half locations for the new tool

DB73 DBD(n+34)	New tool: Size top
Signal state 0	No tool size is specified.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.21 DB73 DBW(n+36) (new tool: Size bottom)

DB73 DBD(n+36)	New tool: Size bottom
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size **bottom** in half locations for the new tool
Signal state 0	No tool size is specified.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.22 DB73 DBW(n+38) (tool status for new tool)

DB73 DBD(n+38)	Tool status for new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Bit 0: Tool in the buffer Bit 1: Locked, but ignore Bit 2: Tool to be unloaded Bit 3: Tool to be loaded Bit 4: Master tool Bit 5: Reserved Bit 6: 1:1 exchange Bit 7: Manual tool Bit 8: Active tool Bit 9: Enable tool Bit 10: Tool disabled Bit 11: Measure tool Bit 12: Prewarning limit reached Bit 13: Tool is being changed Bit 14: Tool is fixed-location-coded Bit 15: Tool was in use

DB73 DBD(n+38)	Tool status for new tool
Signal state 0	No tool status displayed.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.23 DB73 DBW(n+40) (new tool: Internal T no. of the NC)

DB73 DBD(n+40)	New tool: Internal T no. of the NC
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Display internal T no. of the NC for the new tool. Tool management variables can be read/ written via FB2/FB3 using this T no.
Signal state 0	No internal T number displayed.
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.9.24 DB73 DBW(n+42) (original location of the new tool in this turret magazine)

DB73 DBD(n+42)	Original location of new tool in this turret magazine
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	
Signal state 0	
Additional information	Start address of the turret: Turret, 1: n = 4 Turret 2: n = 48 Turret 3: n = 92

18.2.10 DB1071: Tool management, load/unload magazine (multitool)

18.2.10.1 DB1071 DBW(n+0) (distance coding)

DB1071 DBW(n+0)	Distance coding
Signal flow	NC → PLC
Update	Job-controlled

DB1071 DBW(n+0)	Distance coding
Signal state 1	Type of distance coding of the multitool (corresponds to $TC_MTP_KD 1 = location number 2 = distance 3 = angle
Signal state 0	No distance coding
Additional information	Starting address of loading/unloading points: Loading/unloading point 1: n = 0 Loading/unloading point 2: n = 20 Loading/unloading point 3: n = 40 Loading/unloading point 4: n = 60

18.2.10.2 DB1071 DBW(n+2) (multitool location number)

DB1071 DBW(n+2)	Multitool location number
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Number of locations of the multitool
Signal state 0	No locations specified.
Additional information	Starting address of loading/unloading points: Loading/unloading point 1: n = 0 Loading/unloading point 2: n = 20 Loading/unloading point 3: n = 40 Loading/unloading point 4: n = 60

18.2.10.3 DB1071 DBD(n+4) (multitool location distance)

DB1071 DBW(n+4)	Multitool location distance
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Distance of the MT location to be positioned from the reference location (real value), corresponding to the distance coding
Signal state 1	No distance specified.
Additional information	Starting address of loading/unloading points: Loading/unloading point 1: n = 0 Loading/unloading point 2: n = 20 Loading/unloading point 3: n = 40 Loading/unloading point 4: n = 60

18.2.10.4 DB1071 DBW(n+8) (multitool number)

DB1071 DBW(n+8)	Multitool number
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Internal T number of the multitool.
Signal state 1	No internal T number specified.
Additional information	Starting address of loading/unloading points: Loading/unloading point 1: n = 0 Loading/unloading point 2: n = 20 Loading/unloading point 3: n = 40 Loading/unloading point 4: n = 60

18.2.10.5 DB1071 DBW(n+10) (multitool location number)

DB1071 DBW(n+10)	Multitool location number
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location number within the multitool (which the system positions to)
Signal state 0	No location number specified.
Additional information	Starting address of loading/unloading points: Loading/unloading point 1: n = 0 Loading/unloading point 2: n = 20 Loading/unloading point 3: n = 40 Loading/unloading point 4: n = 60

18.2.10.6 DB1071 DBW(n+12) (toolholder)

DB1071 DBW(n+12)	Tool holder
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Spindle or tool holder number
Signal state 0	No spindle or toolholder number specified.
Additional information	Starting address of loading/unloading points: Loading/unloading point 1: n = 0 Loading/unloading point 2: n = 20 Loading/unloading point 3: n = 40 Loading/unloading point 4: n = 60

18.2.11 DB1072: Tool management, spindle (multitool)

18.2.11.1 DB1072 DBW(n+0) (distance coding)

DB1072 DBW(n+0)	Distance coding
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Type of distance coding of the multitool (corresponds to $TC_MTP_KD) 1 = location number 2 = distance 3 = angle
Signal state 0	No distance coding specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.2 DB1072 DBW(n+2) (multitool location number)

DB1072 DBW(n+2)	Multitool location number
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Number of locations of the multitool
Signal state 0	No data regarding the number of locations.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.3 DB1072 DBW(n+4) (multitool location distance)

DB1072 DBW(n+4)	Multitool location distance
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Distance of the MT location to be positioned from the reference location (real value), corresponding to the distance coding
Signal state 1	No data regarding the distance of the MT location to be positioned.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.4 DB1072 DBW(n+8) (multitool number (new tool))

DB1072 DBW(n+8)	Multitool number (new tool)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Internal T number of the (new) multitool.
Signal state 0	No internal T number specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.5 DB1072 DBW(n+10) (multitool location number (new tool))

DB1072 DBW(n+10)	Multitool location number (new tool)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location number within the multitool (where the new tool is located).
Signal state 0	No location number specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.6 DB1072 DBW(n+12) (multitool number (old tool))

DB1072 DBW(n+12)	Multitool number (old tool)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Internal T number of the (old) multitool. The T number is entered here if the preparation or the change to a tool is realized within the same multitool (which, due to a previous change, is located on the tool holder). It is identical with DB1072 DBW(n + 8)
Signal state 0	No internal T number of the multitool specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.7 DB1072 DBW(n+14) (multitool location number (old tool))

DB1072 DBW(n+14)	Multitool location number (old tool)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location number within the multitool (where the old tool is located). The location number - where the old tool is located - is entered here if the preparation or the change to a tool is realized within the same multitool (which, due to a previous change, is located on the tool holder).
Signal state 0	No multitool location number specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.8 DB1072 DBW(n+16) (new tool: Location type)

DB1072 DBW(n+16)	New tool: Location type
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location type of the new tool (of the programmed tool in the multitool).
Signal state 0	No location type specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100
Corresponds with	Tool size: Left, right, top, bottom

18.2.11.9 DB1072 DBW(n+18) (new tool: Size left)

DB1072 DBW(n+18)	New tool: Size left
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size to the **left** in half locations of the new tool (of the programmed tool in the multitool)
Signal state 0	No tool size is specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.10 DB1072 DBW(n+20) (new tool: Size right)

DB1072 DBW(n+20)	New tool: Size right
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size to the **right** in half locations of the new tool (of the programmed tool in the multitool)
Signal state 0	No tool size is specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.11 DB1072 DBW(n+22) (new tool: Size top)

DB1072 DBW(n+22)	New tool: Size top
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size to the **top** in half locations of the new tool (of the programmed tool in the multitool)
Signal state 0	No tool size is specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.12 DB1072 DBW(n+24) (new tool: Size bottom)

DB1072 DBW(n+24)	New tool: Size bottom
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size to the **bottom** in half locations of the new tool (of the programmed tool in the multitool)
Signal state 0	No tool size is specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.13 DB1072 DBW(n+26) (new tool status)

DB1072 DBW(n+26)	Tool status for new tool:
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Tool status of the programmed tool in the multitool. Corresponds to parameter $TC_TP8[T_No] Bit 0: Active tool Bit 1: Enable tool Bit 2: Tool disabled Bit 3: Measure tool Bit 4: Prewarning limit reached Bit 5: Tool is being changed Bit 6: Tool is fixed-location-coded Bit 7: Tool was in use Bit 8: Tool in the buffer Bit 9: Disabled, but ignore Bit 10: Tool to be unloaded Bit 11: Tool to be loaded Bit 12: Master tool Bit 13: Reserved Bit 14: 1:1 exchange Bit 15: Manual tool
Signal state 0	No tool status displayed.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.14 DB1072 DBW(n+28) (new tool: internal T number of the NC)

DB1072 DBW(n+28)	New tool: Internal T number of the NC
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Display the internal T number of the NC for the new spindle tool (of the programmed tool in the multitool).
Signal state 0	No internal T number displayed.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.15 DB1072 DBW(n+30) (tool holder)

DB1072 DBW(n+30)	Tool holder
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Spindle or toolholder number to which the change refers (the multitool that is to be changed).
Signal state 0	No spindle or toolholder number specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.16 DB1072 DBW(n+32) (original magazine of the new tool)

DB1072 DBW(n+32)	Original magazine of new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Magazine that owns the new tool. Corresponds to NC variables $A_MYMN[T no.] If the new tool is located in the magazine, then this value is identical with DB72 DBW(n+20). If the new tool is in the buffer magazine (e.g. gripper) then the magazine No. is entered here from where the tool originally came. The new tool has the same owner location as the multitool onto which it is loaded.
Signal state 0	No original magazine specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.11.17 DB1072 DBW(n+34) (original location of the new tool)

DB1072 DBW(n+34)	Original location of new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location that owns the new tool. Corresponds to NC variables $A_MYMLNT[T no.] If the new tool is located in the magazine, then this value is identical with DB72 DBW(n+20). If the new tool is in the buffer magazine (e.g. gripper) then the magazine No. is entered here from where the tool originally came. The new tool has the same owner location as the multitool onto which it is loaded.

DB1072 DBW(n+34)	Original location of new tool
Signal state 0	No original location specified.
Additional information	Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12 DB1073: Tool management, turret (multitool)

18.2.12.1 DB1073 DBW(n+0) (distance coding)

DB1073 DBW(n+0)	Distance coding
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Type of distance coding of the multitool (corresponds to $TC_MTP_KD) 1 = location number 2 = distance 3 = angle
Signal state 0	No distance coding specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.2 DB1073 DBW(n+2) (multitool location number)

DB1073 DBW(n+2)	Multitool location number
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Number of locations of the multitool
Signal state 0	Number of locations of the multitool not displayed.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.3 DB1073 DBW(n+4) (multitool location distance)

DB1073 DBW(n+4)	Multitool location distance
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Distance of the MT location to be positioned from the reference location (real value). In accordance with the distance coding.
Signal state 0	No distance specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.4 DB1073 DBW(n+8) (multitool number (new tool))

DB1073 DBW(n+8)	Multitool number (new tool)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Internal T number of the (new) multitool.
Signal state 0	No internal T number of the multitool is specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.5 DB1073 DBW(n+10) (multitool location number (new tool))

DB1073 DBW(n+10)	Multitool location number (new tool)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location number within the multitool (where the new tool is located).
Signal state 1	No multitool location number specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.6 DB1073 DBW(n+12) (multitool number (old tool))

DB1073 DBW(n+12)	Multitool number (old tool)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Internal T number of the (old) multitool. The T number is entered here if the preparation or the change to a tool is realized within the same multitool (which, due to a previous change, is located on the tool holder). It is identical with DB1072 DBW(n + 8)
Signal state 0	No internal T number of the multitool specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.7 DB1073 DBW(n+14) (multitool location number (old tool))

DB1073 DBW(n+14)	Multitool location number (old tool)
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location number within the multitool (where the old tool is located). The location number - where the old tool is located - is entered here if the preparation or the change to a tool is realized within the same multitool (which, due to a previous change, is located on the tool holder).
Signal state 0	No location number specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.8 DB1073 DBW(n+16) (new tool: Location type)

DB1073 DBW(n+16)	New tool: Location type
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location type of the new tool (of the programmed tool in the multitool).
Edge evaluation	Signal(s) updated: Conditional
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100
Corresponds with	Tool size: Left, right, top, bottom

18.2.12.9 DB1073 DBW(n+18) (new tool: Size left)

DB1073 DBW(n+18)	New tool: Size left
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size to the **left** in half locations of the new tool (of the programmed tool in the multitool)
Signal state 1	No tool size is specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.10 DB1073 DBW(n+20) (new tool: Size right)

DB1073 DBW(n+20)	New tool: Size right
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size to the **right** in half locations of the new tool (of the programmed tool in the multitool)
Signal state 0	No tool size is specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.11 DB1073 DBW(n+22) (new tool: Size top)

DB1073 DBW(n+22)	New tool: Size top
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size to the **top** in half locations of the new tool (of the programmed tool in the multitool)
Signal state 0	No tool size is specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.12 DB1073 DBW(n+24) (new tool: Size bottom)

DB1073 DBW(n+24)	New tool: Size bottom
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Specification of the tool size to the **bottom** in half locations of the new tool (of the programmed tool in the multitool)
Signal state 0	No tool size is specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.13 DB1073 DBW(n+26) (new tool status)

DB1073 DBW(n+26)	Tool status for new tool:
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Tool status of the programmed tool in the multitool Corresponds to parameter $TC_TP8[T_No] Bit 0: Active tool Bit 1: Enable tool Bit 2: Tool disabled Bit 3: Measure tool Bit 4: Prewarning limit reached Bit 5: Tool is being changed Bit 6: Tool is fixed-location-coded Bit 7: Tool was in use Bit 8: Tool in the buffer Bit 9: Disabled, but ignore Bit 10: Tool to be unloaded Bit 11: Tool to be loaded Bit 12: Master tool Bit 13: Reserved Bit 14: 1:1 exchange Bit 15: Manual tool
Signal state 0	No tool status specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.14 DB1073 DBW(n+28) (new tool: internal T number of the NC)

DB1073 DBW(n+28)	New tool: Internal T number of the NC
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Display the internal T number of the NC for the new spindle tool (of the programmed tool in the multitool).
Signal state 0	No internal T number displayed.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.15 DB1073 DBW(n+30) (tool holder)

DB1073 DBW(n+30)	Tool holder
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Spindle or tool holder number to which the change refers (the multitool that is to be changed)
Signal state 0	No spindle or toolholder number specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.16 DB1073 DBW(n+32) (original magazine of the new tool)

DB1073 DBW(n+32)	Original magazine of new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Magazine that owns the new tool. Corresponds to NC variables $A_MYMN[T no.] If the new tool is located in the magazine, then this value is identical with DB72 DBW(n+20). If the new tool is in the buffer (e.g. gripper), then the magazine No. is entered here from where the tool originally came. The new tool has the same owner location as the multitool onto which it is loaded.
Signal state 0	Magazine that owns the new tool not specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

18.2.12.17 DB1073 DBW(n+34) (original location of the new tool)

DB1073 DBW(n+34)	Original location of new tool
Signal flow	NC → PLC
Update	Job-controlled
Signal state 1	Location that owns the new tool. Corresponds to NC variables $A_MYMLNT[T no.] If the new tool is located in the magazine, then this value is identical with DB72 DBW(n+20). If the new tool is in the buffer magazine (e.g. gripper), the number of the magazine from which the tool originally came is entered here. The new tool has the same owner location as the multitool onto which it is loaded.
Signal state 0	Location that owns the new tool not specified.
Additional information	Start addresses of the turrets: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100

Appendix

A

A.1 Overview

General documentation



Advertising brochure
- SINUMERIK 840D sl
- SINUMERIK 828D
- SINUMERIK 828D BASIC



Catalog NC 62
SINUMERIK 840D sl



Catalog NC 82
SINUMERIK 828D



Catalog PM 21
SIMOTION, SINAMICS S120



System Manual
Intelligent machine tool operation



System Manual
Ctrl-Energy

User documentation



Operating Manual
- Universal
- Turning
- Milling
- Grinding



Programming Manual
- Fundamentals
- Job planning
- Measuring cycles



Programming Manual
- ISO turning
- ISO milling



Diagnostics Manual
Alarms



Diagnostics Manual
Alarms

Manufacturer/Service Documentation



Equipment Manual
- NCU
- Operator components and networking
- ADI4



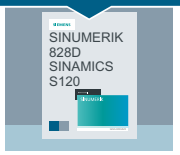
Equipment Manual
Commissioning Manual
Service Manual



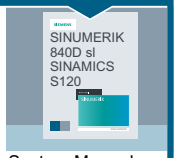
Commissioning Manual
- CNC: NCK, PLC, drive
- Base software and operating software



List Manual
- Machine data
- Interface signals
- Variables



List Manual
- Machine data
- Interface signals
- Parameters
- Variables



System Manual
Guidelines for configuring machines



Function Manual
- Basic functions
- Extended functions
- Special functions
- Synchronized actions
- ISO dialects



Function Manual
Tool management



Function Manual
Safety Integrated



Function Manual
Safety Integrated



Configuration Manual
- EMC installation guideline
- Industrial Security

Information/training



Training documentation
- Basic milling with ShopMill
- Basic turning with ShopTurn



Manual
Tool and mold making

Electronic documentation



DOConCD



Industry Online Support
(SIOS)



Industry Mall

Index

A

Activation
    from machine control panel, hand-held unit, 91
AG_SEND, AG_RECV, 81

C

Concurrent axes, 79
Cyclic operation, 73

D

Data block DB, 307
Data exchange
    with operator panel, 51
Data interface, 55
DB10
    Analog NC inputs and outputs, 350
    Collision avoidance: Activate protection area, 351
    Collision avoidance:protection area active, 351
    DBB194 - 208, 473
    DBW148 - 162, 467
    DBW170 - 184, 469
    DBW210 - 224, 473, 474
    DBX0.0 - 7, 434, 435
    DBX1.0 - 7, 435
    DBX100.0 - 4, 444
    DBX100.5, 444, 445
    DBX100.6, 445
    DBX100.7, 446
    DBX101.0 - 4, 446
    DBX101.5, 446
    DBX101.6, 446
    DBX101.7, 447
    DBX102.0 - 4, 447
    DBX102.5, 447
    DBX102.6, 447
    DBX102.7, 447
    DBX103.0, 447
    DBX103.5, 447, 448
    DBX103.6, 448
    DBX103.7, 448
    DBX104.3, 93
    DBX104.4, 93
    DBX104.7, 77, 448
    DBX106.1, 449
    DBX107.0 - 1, 449
    DBX107.6, 449
    DBX108.3, 450
    DBX108.5, 450
    DBX108.6, 450
    DBX108.7, 450, 451
    DBX109.0, 451
    DBX109.5, 451
    DBX109.6, 452
    DBX109.7, 452
    DBX110.0 - 113.7, 452, 453
    DBX114.0 - 117.7, 453
    DBX122.0 - 7, 454
    DBX123.0 - 7, 454
    DBX124.0 - 7, 454, 455
    DBX125.0 - 7, 455
    DBX126.0 - 7, 455
    DBX127.0 - 7, 456
    DBX128.0 - 7, 456
    DBX129.0 - 7, 456, 457
    DBX130.0 - 7, 457
    DBX131.0 - 7, 457, 458
    DBX132.0 - 7, 458
    DBX133.0 - 7, 459
    DBX134.0 - 7, 459
    DBX135.0 - 7, 460
    DBX136.0 - 7, 460, 461
    DBX137.0 - 7, 461
    DBX138.0 - 7, 461, 462
    DBX139.0 - 7, 462
    DBX140.0 - 7, 462, 463
    DBX141.0 - 7, 463
    DBX142.0 - 7, 464
    DBX143.0 - 7, 464
    DBX144.0 - 7, 465
    DBX145.0 - 7, 465, 466
    DBX146.0 - 7, 466
    DBX147.0 - 7, 466
    DBX166.0 - 7, 467
    DBX167.0 - 7, 468
    DBX168.0 - 7, 468
    DBX186.0 - 7, 469
    DBX187.0 - 7, 470
    DBX188.0 - 7, 470
    DBX189.0 - 7, 471
    DBX190.0 - 7, 471
    DBX191.0 - 7, 472
    DBX192.0 - 7, 472
    DBX193.0 - 7, 473

DBX226.0 - 233.7, 474
DBX234.0 - 241.7, 475
DBX245.0 - 5, 475
DBX4.0 - 7, 435, 436
DBX5.0 - 7, 436
DBX56.1, 438
DBX56.2, 438
DBX56.4 - 7, 438, 439
DBX58.0 - 7, 439, 440
DBX6.0 - 7, 436, 437
DBX60.0 - 7, 440, 441
DBX64.0 - 7, 441
DBX7.0 - 7, 437
DBX92.0, 76
DBX92.1, 76
DBX92.2, 76
DBX92.4, 76
DBX92.5, 76
DBX92.6, 76
DBX93.0 ... 7, 441, 442
DBX97.0 - 3, 442, 443
DBX98.0 - 3, 443
DBX99.0 - 3, 443
External analog NC inputs, 347
External analog NC outputs, 348
External digital NC inputs, 345
External digital NC inputs and outputs, 349
External digital NC outputs, 346
General signals from the NC, 344
General signals to the NC, 340
Handwheel signals from the NC, 352
Onboard inputs and outputs from the NC, 341
Onboard NC inputs/outputs, 339
Signals from the operating software, 342
Signals from the robot, 353
Signals to the robot, 353
DB1000
Energy-saving profiles, 424
DB1001
SENTRON PAC, 426
SENTRON PAC auxiliary devices, 428
DB1002
Spindle temperature sensors, 429
DB1071
DBW(n+0), 728, 729
DBW(n+10), 730
DBW(n+12), 730
DBW(n+2), 729
DBW(n+4), 729
DBW(n+8), 730
Multitool magazine interface, 431
DB1072
DBW(n+0), 731
DBW(n+10), 732
DBW(n+12), 732
DBW(n+14), 733
DBW(n+16), 733
DBW(n+18), 733
DBW(n+2), 731
DBW(n+20), 734
DBW(n+22), 734
DBW(n+24), 734
DBW(n+26), 735
DBW(n+28), 735
DBW(n+30), 736
DBW(n+32), 736
DBW(n+34), 736, 737
DBW(n+4), 731
DBW(n+8), 732
Multitool spindle interface, 432
DB1073
DBW(n+0), 737
DBW(n+10), 738
DBW(n+12), 739
DBW(n+14), 739
DBW(n+16), 739
DBW(n+18), 740
DBW(n+2), 737
DBW(n+20), 740
DBW(n+22), 740
DBW(n+24), 741
DBW(n+26), 741
DBW(n+28), 742
DBW(n+30), 742
DBW(n+32), 742
DBW(n+34), 743
DBW(n+4), 738
DBW(n+8), 738
Multitool turret interface, 433
DB11
DBX0.0, 476
DBX0.1, 476
DBX0.2, 476, 477
DBX0.4, 477
DBX0.5, 477
DBX0.6, 478
DBX0.7, 478, 479
DBX1.0, 479
DBX1.1, 479
DBX1.2, 480
DBX1.6, 480
DBX1.7, 480, 481
DBX4.0, 481

DBX4.1, 481
DBX4.2, 481, 482
DBX5.0, 482
DBX5.1, 482
DBX5.2, 482
DBX6.0, 483
DBX6.1, 483
DBX6.2, 483
DBX6.3, 483, 484
DBX6.7, 484
DBX7.0, 484
DBX7.1, 484, 485
DBX7.2, 485
Mode signals 1 from HMI, 355
Mode signals 1 from the NC, 354
Mode signals 2 from HMI, 356
Mode signals 2 from the NC, 356
Mode signals 2 to the NC, 355
Mode signals to the NC, 354
DB18
Additional data areas, 359
Data area/errors, 357
DBB36.0, 485
DBB36.1, 485
DBB38.0 - 41.7, 486
DBB42.0 - 45.7, 486
DBB46.0 - 49.7, 486, 487
Extended additional data areas, 364
Extended data area / errors, 362
F_SENDDP receiver, 360, 367
F_SENDDP sender, 360, 365
Parameterization part, 357
SPL user data, 361
DB19
DBB10, 490
DBB15, 491
DBB16, 491, 492
DBB17, 492, 493
DBB22, 494
DBB27, 497, 498
DBB33, 109
DBB33 - 35, 499
DBB36, 109, 500
DBB6, 489
DBB7, 489
DBB8, 490
DBW24, 494
DBX0.0, 487
DBX0.1, 487, 488
DBX0.2, 488
DBX0.3, 488
DBX0.4, 489
DBX0.7, 489
DBX13.5, 490
DBX13.6, 490
DBX13.7, 490
DBX14.0 - 6, 491
DBX14.7, 491
DBX20.1, 493
DBX20.3, 493
DBX20.4, 493
DBX20.6, 494
DBX20.7, 494
DBX26, 495, 496
DBX26.1, 495
DBX26.2, 495
DBX26.5, 496
DBX26.6, 496, 497
DBX26.7, 497
DBX32.0 - .5, 109, 498
DBX32.6, 109, 499
DBX32.7, 109, 499
Signals from the operator panel, 373
Signals to the operator panel, 371
Signals to/from operator panel, sidescreen, 377
DB19, signals from the operator panel, 377
DB2
Axis ranges, 322, 333
Channel range 1, 317, 326
Channel range 10, 333
Channel range 2, 318, 327
Channel range 3, 318, 328
Channel range 4, 319, 329
Channel range 5, 320, 329
Channel range 6, 320, 330
Channel range 7, 321, 331
Channel range 8, 322, 331
Channel range 9, 332
User ranges, 323, 335
DB20
NC machine data, 378
DB21
DBB129, 566
DBB140, 566
DB21 - DB30
Change signals from the channel, 387
Control signals from the geometry axes, 385
Control signals from the operating software, 383
Control signals from the PLC, 383
Control signals to the geometry axes, 381
D function signals, 389
DL function signals, 389
F function signals, 390
G function signals, 392

H function signals, 390
Job-controlled signals, 394, 396
M function signals, 388
M function signals, dynamic, 391
Protection zone signals from the channel, 393
S function signals, 388
Signals for the tool management functions, 399
Signals from channel, 400
Signals from the orientation axes, 398
Signals from/to channel, 402
Signals to channel, 401
Signals to the orientation axes, 396
Status signal from the channel, 383
T function signals, 389
DB21, ...
DBB158 - 193, 566, 567
DBB208 - 271, 567, 568
DBB376, 589
DBB392, 593, 594
DBB4, 509, 510, 511
DBB5, 511, 512, 513
DBB58, 563
DBB60 - 64, 564
DBB60 - 65, 563
DBB66 - 67, 564
DBB68 - 97, 564, 565
DBB98 - DBB115, 565
DBW118 or DBD118, 565
DBX0.1, 500
DBX0.2, 501
DBX0.3, 501
DBX0.4, 501, 502
DBX0.5, 502
DBX0.6, 502, 503
DBX1.0, 503
DBX1.2, 504
DBX1.3, 504
DBX1.4, 504
DBX1.5, 505
DBX1.6, 505
DBX1.7, 506
DBX10.0 - 11.1, 520, 521
DBX12.0 - 2, 521, 522
DBX12.3, 522
DBX12.4, 522
DBX12.5, 523
DBX12.6 - 7, 523, 524, 525
DBX13.0 - 6, 525, 526
DBX15.0, 526
DBX16.0 - 2, 526
DBX16.3, 522
DBX16.4, 527
DBX16.5, 527
DBX16.6 - 7, 527
DBX17.0 - 6, 527
DBX19.0, 527
DBX194.0 - 7 - DBX206.0 - 3, 567
DBX2.0 - 7, 506, 507
DBX20.0 - 2, 527
DBX20.3, 522
DBX20.4, 527
DBX20.5, 528
DBX20.6 - 7, 528
DBX21.0 - 6, 528
DBX23.0, 528
DBX24.1, 528
DBX24.3, 529
DBX24.4, 529
DBX24.5, 530
DBX24.6, 530
DBX25.3, 531
DBX25.7, 531, 532
DBX26.0 ... 7, 532, 533
DBX272.0 - 273.1, 568
DBX274.0 - 275.1, 569
DBX276.0 - 277.1, 569
DBX278.0 - 279.1, 570
DBX280.1, 570
DBX281.1, 570, 571
DBX29.0 - 3, 533, 534
DBX29.4, 534
DBX29.5, 534
DBX29.6, 534, 535
DBX29.7, 535
DBX3.0, 507
DBX3.1, 507, 508
DBX3.2, 508
DBX3.3, 508
DBX3.4, 508, 509
DBX3.5, 509
DBX30.0 - 2, 535, 536
DBX30.3, 536
DBX30.4, 537
DBX30.5, 537
DBX30.6, 537, 538
DBX300.0 - 307.7, 571
DBX308.0 - 315.7, 571
DBX31.0 - 2, 538
DBX31.4, 539
DBX31.5, 539
DBX317.1, 571
DBX317.6, 572
DBX317.7, 572
DBX318.0, 572

DBX318.1, 573
DBX318.2, 573
DBX318.3, 573, 574
DBX318.5, 574
DBX319.0, 574, 575
DBX319.1 - 3, 575
DBX319.5, 576
DBX32.1, 539, 540
DBX32.2, 540
DBX32.3, 540
DBX32.4, 540, 541
DBX32.5, 541
DBX32.6, 541
DBX320.0 - 2, 576, 577
DBX320.4, 577, 578
DBX320.5, 578
DBX320.6 - 7, 578, 579, 580
DBX321.0 - 6, 580, 581
DBX323.0, 581
DBX324.0 - 2, 581
DBX324.4, 581
DBX324.5, 582
DBX324.6 - 7, 582
DBX325.0 - 6, 582
DBX327.0, 582
DBX328.0 - 2, 582
DBX328.4, 582
DBX328.5, 582
DBX328.6 - 7, 583
DBX329.0 - 6, 583
DBX33.0, 542
DBX33.3, 542
DBX33.4, 543
DBX33.5, 543, 544
DBX33.6, 544
DBX33.7, 544, 545
DBX331.0, 583
DBX332.0 - 2, 583, 584
DBX332.4 - 5, 584, 585
DBX332.6 - 7, 585, 586
DBX333.0 - 6, 586
DBX336.0 - 2, 586
DBX336.4 - 5, 587
DBX336.6 - 7, 587
DBX337.0 - 6, 587
DBX340.0 - 2, 587
DBX340.4 - 5, 587
DBX340.6 - 7, 587
DBX341.0 - 6, 587
DBX344.0, 588
DBX344.1, 588
DBX344.2, 588, 589
DBX344.3, 589
DBX35.0, 545, 546
DBX35.1, 546, 547
DBX35.2, 547
DBX35.3, 547, 548
DBX35.4, 548
DBX35.5, 549
DBX35.6, 549
DBX35.7, 550
DBX36.2, 550
DBX36.3, 551
DBX36.4, 551
DBX36.5, 551
DBX36.6, 552
DBX36.7, 552
DBX37.0 - 2, 552, 553
DBX37.3, 553
DBX37.4, 554
DBX37.5, 554
DBX37.6, 555
DBX37.7, 555, 556
DBX377.0, 590
DBX377.4, 590
DBX377.5, 590
DBX377.6, 590
DBX378.0, 591
DBX378.1, 591
DBX38.0, 556
DBX38.1, 556
DBX384.0, 591
DBX385.0 - 7, 592
DBX386.0 - 7, 592
DBX387.0 - 7, 592
DBX39.1, 557
DBX39.4, 557
DBX39.5, 557, 558
DBX39.6, 558
DBX39.7, 558
DBX390.0 - 7, 593
DBX391.0 - 7, 593
DBX40.0 - 2, 558, 559
DBX40.4 - 5, 559, 560
DBX40.6 - 7, 560, 561
DBX41.0 - 6, 561
DBX43.0, 561, 562
DBX46.0 - 2, 562
DBX46.4 - 5, 562
DBX46.6 - 7, 562
DBX47.0 -.6, 562
DBX49.0, 562
DBX52.0 - 2, 563
DBX52.4 - 5, 563

DBX52.6 - 7, 563
DBX53.0 - 6, 563
DBX55.0, 563
DBX59.0 - 4, 564
DBX6.0, 513, 514
DBX6.1, 514
DBX6.2, 514, 515
DBX6.4, 515
DBX6.6, 515, 516
DBX6.7, 516
DBX7.0, 516, 517
DBX7.1, 517
DBX7.2, 517
DBX7.3, 518
DBX7.4, 518, 519
DBX7.6, 519
DBX7.7, 519, 520
DBX8.0 - 9.1, 520
DB31
DBX34.0 - 1, 643
DB31 - DB61
Axis signals: Safety Control Channel (SCC), 416
Safety Info Channel (SIC), 416
Signals from axis/spindle, 408
Signals to the axis/spindle, 403
DB31, ...
DBB0, 594, 595, 596
DBB19, 629, 630, 631
DBB68, 659
DBB8, 617
DBD78, 667
DBD88, 676
DBW134, 698, 699
DBW136, 699
DBW86, 675, 676
DBX1.0, 596
DBX1.1, 597
DBX1.2, 597
DBX1.3, 598, 599, 600, 601
DBX1.4, 601
DBX1.5 - DBX1.6, 602, 603, 604
DBX1.7, 604, 605
DBX10.0, 618, 619
DBX100.2, 692, 693
DBX100.3, 693
DBX100.4, 693
DBX100.5, 693
DBX100.6, 693, 694
DBX100.7, 694
DBX102.5, 694
DBX102.6, 694
DBX104.0 / 107.6, 695
DBX12.0, 619
DBX12.1, 619
DBX12.2, 620
DBX12.3, 620
DBX12.4, 620
DBX12.7, 620, 621
DBX128.0, 695
DBX128.1, 695, 696
DBX13.0 - 2, 621
DBX13.3, 621, 622
DBX130.0 - 4, 696
DBX132.0, 696
DBX132.1, 697
DBX132.4, 697
DBX132.5, 697, 698
DBX138.4, 699
DBX138.5, 699, 700
DBX14.0, 622
DBX14.1, 623
DBX16.0 - DBX16.2, 623, 624
DBX16.3, 624, 625
DBX16.4, 625
DBX16.5, 625, 626
DBX16.7, 626
DBX17.4, 626
DBX17.5, 626, 627
DBX17.6, 627
DBX18.4, 627, 628
DBX18.5, 628
DBX18.6, 628, 629
DBX18.7, 629
DBX2.0, 605
DBX2.1, 605, 606, 607
DBX2.2, 607, 608
DBX2.3, 609
DBX2.4 - 7, 609
DBX20.1, 631
DBX21.0 - 4, 632
DBX21.5, 632, 633
DBX21.6, 633
DBX21.7, 633, 634
DBX22.0, 634
DBX22.1, 634
DBX22.3 - 4, 634
DBX23.0 - 2, 635
DBX23.4, 635
DBX24.1, 635
DBX24.2, 636
DBX24.3, 636
DBX24.4, 637
DBX24.5, 637
DBX24.7, 637, 638

DBX26.4, 638
DBX28.0, 638, 639
DBX28.1, 639
DBX28.2, 639, 640
DBX28.3, 640
DBX28.4, 641
DBX28.5, 641
DBX28.6, 641, 642
DBX28.7, 642, 643
DBX3.0, 610
DBX3.1, 610
DBX3.2 - 5, 611
DBX3.6, 611, 612
DBX31.5, 643
DBX33.2, 542
DBX4.0 - 2, 612
DBX4.3, 613
DBX4.4, 614
DBX4.5, 614
DBX4.6 - 7, 614, 615
DBX5.0 - 6, 615, 616
DBX60.0, 644
DBX60.1, 644
DBX60.2, 645
DBX60.3, 645, 646
DBX60.4, 646
DBX60.5, 646, 647
DBX60.6, 647
DBX60.7, 647, 648
DBX61.0, 648
DBX61.1, 648, 649
DBX61.2, 649
DBX61.3, 649, 650
DBX61.4, 650
DBX61.5, 650, 651
DBX61.6, 651
DBX61.7, 651, 652
DBX62.0, 652
DBX62.1, 652
DBX62.2, 653
DBX62.3, 653
DBX62.4, 653
DBX62.5, 654
DBX62.7, 654
DBX63.0, 654, 655
DBX63.1, 655
DBX63.2, 655
DBX63.3, 655, 656
DBX64.0 - 2, 656
DBX64.4 - 5, 657
DBX64.6 - 7, 657
DBX65.0 - 6, 658
DBX66.0, 658
DBX67.0, 659
DBX69.0 - 2, 659, 660
DBX7.0, 616, 617
DBX70.0, 660
DBX70.1, 660, 661
DBX70.2, 661
DBX71.4, 662
DBX71.5, 662
DBX72.0, 662, 663
DBX74.4, 663
DBX75.0 - 2, 663
DBX75.3 - 5, 664
DBX75.6, 664
DBX75.7, 664, 665
DBX76.0, 665
DBX76.4, 665
DBX76.5, 666
DBX76.6, 666
DBX77.0, 667
DBX82.0 - 2, 667, 668
DBX82.3, 668
DBX83.0, 668, 669
DBX83.1, 669, 670
DBX83.2, 670
DBX83.3, 671
DBX83.5, 671
DBX83.6, 671, 672
DBX83.7, 672
DBX84.1, 672
DBX84.3, 672, 673
DBX84.4, 673
DBX84.5, 673, 674
DBX84.6, 674
DBX84.7, 674
DBX85.0, 674, 675
DBX85.5, 675
DBX9.0 - 2, 618
DBX9.3, 618
DBX92.1, 676, 677
DBX93.0 - 4, 677
DBX93.5, 678
DBX93.6, 678
DBX93.7, 679
DBX94.0, 679
DBX94.1, 680
DBX94.2, 680, 681
DBX94.3, 681
DBX94.4, 682
DBX94.5, 682, 684, 685
DBX94.6, 683
DBX94.7, 683

DBX95.1, 683, 684
DBX95.2, 684
DBX95.3, 684
DBX96.2, 685
DBX96.3, 685
DBX96.4, 685, 686
DBX96.5, 686
DBX96.7, 686
DBX97.0, 687
DBX97.1, 687
DBX97.2, 687, 688
DBX97.3, 688
DBX98.0, 688, 689
DBX98.1, 689
DBX98.2, 689, 690
DBX98.4, 690
DBX98.5, 690, 691
DBX98.6, 691
DBX99.0, 691
DBX99.1, 692
DBX99.3, 692
DB71
DBB(n+2), 704
DBB(n+3), 704
DBW(n+16), 705
DBW(n+18), 705
DBW(n+20), 705
DBW(n+22), 706
DBW(n+24), 706
DBW(n+26), 706, 707
DBX(n+0).0, 700, 701
DBX(n+0).1, 701
DBX(n+0).2, 701, 702
DBX(n+0).3, 702
DBX(n+0).4, 702, 703
DBX(n+0).5, 703
DBX(n+1).0, 703
DBX(n+1).7, 704
DBX(n+28).0, 707
DBX0.0 - 1.7, 700
DBX2.0 - 3.7, 700
Magazine interface, 417
DB72
DBB(n+2), 712, 713
DBB(n+3), 713
DBD(n+12), 714
DBD(n+4), 713
DBD(n+8), 713, 714
DBW(n+16), 714
DBW(n+18), 714, 715
DBW(n+20), 715
DBW(n+22), 715
DBW(n+24), 715, 716
DBW(n+26), 716
DBW(n+28), 716
DBW(n+30), 716, 717
DBW(n+32), 717
DBW(n+34), 717
DBW(n+36), 717, 718
DBW(n+38), 718
DBW(n+40), 718, 719
DBW(n+42), 719
DBW(n+44), 719
DBW(n+46), 719, 720
DBX(n+0).0, 708
DBX(n+0).1, 708, 709
DBX(n+0).2, 709
DBX(n+0).3, 709, 710
DBX(n+0).4, 710
DBX(n+0).5, 710, 711
DBX(n+0).6, 711
DBX(n+0).7, 711, 712
DBX(n+1).0, 712
DBX(n+1).7, 712
DBX0.0-1.7, 707, 708
DBX2.0-3.7, 708
Spindle interface, 419
DB73
DBB(n+2), 722, 723
DBB(n+3), 723
DBD(n+12), 724
DBD(n+20), 724
DBD(n+22), 724, 725
DBD(n+26), 725
DBD(n+28), 725, 726
DBD(n+30), 726
DBD(n+32), 726
DBD(n+34), 726, 727
DBD(n+36), 727
DBD(n+38), 727, 728
DBD(n+4), 723
DBD(n+40), 728
DBD(n+42), 728
DBD(n+8), 723, 724
DBW(n+24), 725
DBX(n+0).0, 721
DBX(n+0).1, 721
DBX(n+0).3, 721, 722
DBX(n+1).0, 722
DBX(n+1).7, 722
DBX0.0 - 1.7, 720
DBX2.0 - 3.7, 720
Turret interface, 421

DB77
MCP and HHU, 423
Direct keys
Address, 93
Alarms, 94
OPs at Ethernet bus, 93

E

Ethernet
connection, 92

F

FB1 RUN_UP (basic program, startup section), 149
FB10 Safety relay, 216
FB11 Brake test, 218
FB2 GET (Read NC Variable), 157
FB29 Signal recorder and data trigger diagnostics, 224
FB3 PUT (write NC variables), 165
FB4 PI_SERV (PI services), 173
Available PI services, 177, 178
FB5 GETGUD (read GUD variable), 202
FB7 PI_SERV2 (PI services), 210
FB9 MzuN (operator panel switchover), 211
FC10 AL_MSG, 247
FC1005 AG_SEND, 299
FC1006 AG_RECV, 300
FC12 AUXFU, 249
FC13 BHGDisp, 250
FC17 YDelta, 254
FC18 SpinCtrl, 257
FC19 MCP_IFM, 267
FC2 GP_HP (basic program, cyclic section), 227
FC21 Transfer, 275
FC22 TM_DIR, 283
FC24 MCP_IFM2, 285
FC25 MCP_IFT, 289
FC26 HPU_MCP, 293
FC3 GP_PRAL (basic program, alarmdriven section), 229
FC5 GP_DIAG (basic program, diagnostics), 231
FC6 TM_TRANS2, 233
FC7 TM_REV, 234
FC8 TM_TRANS, 238
FC9 ASUP, 244
Function block FB, 305
Function block FC, 306
Function interface, 56

H

Handheld unit
HHU / HT 2, 312
HT 10, 315, 316
HT 8, 314
Hardware interrupt, 76
HMI monitor, 123

I

Interface
MCP/PLC and HHU/PLC, 53
PLC/HMI, 53, 62
PLC/HMI messages, 62
PLC/MCP, 68
PLC/NC, 55
Interface signals
"auto" acknowledgment negative, 703, 712, 722
"auto" acknowledgment of interfaces 1 - 16, 700, 720
"auto" acknowledgment of interfaces 1-16, 708
"Silent" ASUB is active, 591
Acceleration alarm threshold reached, 691
Accept external work offset, 610
Acknowledge Emergency Stop, 438
Acknowledge fixed stop reached, 597
Acknowledgement of synchronized actions disabled, 570, 571
Action block active, 540
Activate associated auxiliary function, 537
Activate configured stop, 519
Activate configured stop (CST) using the operating software, 528
Activate contour handwheel, 535, 536
Activate dry run feedrate (DRY), 502, 503
Activate fixed feedrate 1 - 4, machine axes, 611
Activate fixed feedrate 1 - 4, path/geometry axes, 533, 534
Activate handwheel, 612
Activate handwheel offset (DRF), 501
Activate M01, 502
Activate machine-related protection zone 1 - 10, 520, 521
Activate program test, 623, 695, 696
Activate program test (PRT), 506
Activate PTP travel, 534
Activate referencing, 503
Activate single block, 501, 502
Activate skip block, 506, 507

Activate travel to fixed stop, 653
Active G command of group 1 to 60, 567, 568
Active infeed axes, 695
Active JOG mode, 483
Active machine function, 658
Active MDI mode, 483
Active position controller parameter set, 659, 660
Active REPOS mode, 575
Active spindle mode, control mode, 674
Active spindle mode, oscillation mode, 674
Active spindle mode, positioning mode, 673, 674
Active spindle mode: Synchronous mode, 673
Active status of interface 1- 16, 720
Active status of interfaces 1- 16, 700
Active status of interfaces 1-16, 707, 708
Actual direction of rotation clockwise, 672
Actual gear stage A to C, 623, 624
Actual screen number, 494
Actual value coupling, 689, 690
Actual values in the WCS, 489
Air temperature alarm, 452
All axes stationary, 551
All axes that have to be referenced are referenced, 550
All channels in the "Reset" state, 484
Analog measured value of the clamping system, 699
Analog NC input 1: Setting value, 467
Analog NC inputs 1 - 8: Actual value, 473
Analog NC inputs 1 - 8: Lock, 466
Analog NC inputs 1 - 8: Target, 466
Analog NC outputs 1 - 8: Overwrite, 467
Analog NC outputs 1 - 8: Setpoint, 441, 473, 474
Analog NC outputs 1 - 8: Setting value, 469
Analog NC outputs 1 - 8: Target, 468
Analog spindle 1, utilization in percent, 489
Analog spindle 2, utilization in percent, 489
Approach block active, 540, 541
Assigned channel, 704, 712, 713, 722, 723
Associated auxiliary function is active, 574
Associated auxiliary function selected, 529
ASUB is active, 591
ASUB is stopped, 572
AT box ready, 447, 448
AUTOMATIC mode, 476
AUTOMATIC mode active, 483
AUTOMATIC mode selected, 481
Axis accelerates, 692
Axis container: Rotation active, 654
Axis number handwheel 1, 444
Axis number handwheel 2, 446
Axis number handwheel 3, 447
Axis stop active, 655
Axis/spindle disable, 598, 599, 600, 601
Axis/spindle disable active, 655, 656
Axis/spindle exchange request, 617
Axis/spindle stationary, 650
Axis-specific alarm, 648, 649
Block search active, 543
Block search via program test is active, 573
Brighten screen, 487
Buffer location of old tool, 719
Buffer magazine no. (fixed value 9998); target position for new tool, 714
Cancel alarm deleted, 493
Change gear stage, 668
Change reversal point, 641
Channel is ready, 551
Channel number, 490
Channel number for geometry axis, handwheel 1, 442, 443
Channel number for geometry axis, handwheel 2, 443
Channel number for geometry axis, handwheel 3, 443
Channel state active, 549
Channel state interrupted, 549
Channel state reset, 550
Channel-specific NC alarm active, 552
Channel-specific protection zone 1 - 10 violated, 570
Clamping in progress, 609
Clearance control (CLC):, 553
Clearance control (CLC): Override, 505
Clearance control (CLC): Stop, 504
Clearance control (CLC): Stop at the lower motion limit, 554
Clearance control (CLC): Stop at the upper motion limit, 554
Collision avoidance: Activate protection zone, 475
Collision avoidance: Deactivate protection area group, 439, 440, 441, 442
Collision avoidance: Protection area active, 474
Collision avoidance: Stop, 590
Collision avoidance: Velocity reduction, 667
Command code: Load manual tool, 710, 711
Command code: Obligatory change, 708
Command code: Old tool in buffer, 710
Command code: Perform change with M06, 708, 709
Command code: Prepare change, 709
Command code: T0, 709, 710

Command code: Tool remains in spindle, 711, 712
Command code: Unload manual tool, 711
Command: Data in the extended range, 704, 712, 722
Command: Loading, 700, 701
Command: Obligatory change, 721
Command: Perform change, 721
Command: Position a multitool, 703
Command: Position to the loading point, 702
Command: Relocate, 701, 702
Command: Task comes from the NC program, 702, 703
Command: Unload, 701
Configured stop is activated, 558
Contour handwheel active, 552, 553
Contour handwheel: handwheel direction of rotation inversion active, 557, 558
Contour handwheel: invert handwheel direction of rotation, 539
Control axis, 635
Controller enable, 605, 606, 607
Current channel number, 494
Current controller active, 651, 652
D function 1, 566
Darken screen, 487, 488
Data area of the SPL inputs, 486
Data area of the SPL outputs, 486, 487
Deceleration of reference point approach, 620, 621
Define handwheel 1 as contour handwheel, 444, 445
Define handwheel 2 as contour handwheel, 446
Define handwheel 3 as contour handwheel, 447
Delayed stroke, 508
Delete cancel alarms, 488
Delete distance-to-go (channel-specific), 514, 515
Delete distance-to-go/spindle reset, 607, 608
Delete recall alarms, 489
Delete S value, 626
Digital NC inputs 1 - 8: Actual value, 440, 441
Digital NC inputs 1 - 8: Lock, 434, 435
Digital NC inputs 1 - 8: Set, 435
Digital NC inputs 17 - 24: Actual value, 470
Digital NC inputs 17 - 24: Lock, 454, 455
Digital NC inputs 17 - 24: Set, 455
Digital NC inputs 25 - 32: Actual value, 470
Digital NC inputs 25 - 32: Lock, 455
Digital NC inputs 25 - 32: Set, 456
Digital NC inputs 33 - 40: Actual value, 471
Digital NC inputs 33 - 40: Lock, 456
Digital NC inputs 33 - 40: Set, 456, 457
Digital NC inputs 9 - 16: Actual value, 469
Digital NC inputs 9 - 16: Lock, 454
Digital NC inputs 9 - 16: Set, 454
Digital NC outputs 1 - 8: Lock, 435, 436
Digital NC outputs 1 - 8: Overwrite, 436
Digital NC outputs 1 - 8: Setting value, 436, 437
Digital NC outputs 1 - 8: Target, 437
Digital NC outputs 17 - 24: Lock, 459
Digital NC outputs 17 - 24: Overwrite, 460
Digital NC outputs 17 - 24: Setpoint, 472
Digital NC outputs 17 - 24: Setting value, 460, 461
Digital NC outputs 17 - 24: Target, 461
Digital NC outputs 25 - 32: Lock, 461, 462
Digital NC outputs 25 - 32: Overwrite, 462
Digital NC outputs 25 - 32: Setpoint, 472
Digital NC outputs 25 - 32: Setting value, 462, 463
Digital NC outputs 25 - 32: Target, 463
Digital NC outputs 33 - 40: Lock, 464
Digital NC outputs 33 - 40: Overwrite, 464
Digital NC outputs 33 - 40: Setpoint, 473
Digital NC outputs 33 - 40: Setting value, 465
Digital NC outputs 33 - 40: Target, 465, 466
Digital NC outputs 9 - 16: Lock, 457
Digital NC outputs 9 - 16: Overwrite, 457, 458
Digital NC outputs 9 - 16: Setpoint, 471
Digital NC outputs 9 - 16: Setting value, 458
Digital NC outputs 9 - 16: Target, 459
Disable all synchronized actions, 504
Disable synchronization, 643
Disable synchronized actions ID/IDS, 571
Disabling of input signal 1 - 8, 592
Distance coding, 728, 729, 731, 737
Drive ready, 450, 678
Drive test travel enable, 596
Drive test travel request, 648
Drive-autonomous motion active, 677
Drives in cyclic operation, 450
Dry run feedrate (DRY) requested, 530
Dynamic M functions M0 - M99, 567
Emergency Stop, 438
Emergency Stop active, 449
Enable following axis override, 638
Enable GOTOS, 591
Enable request for action 1 - 8, 593
Enable status of input signal 1 - 8, 593
Enable travel to fixed stop, 610
Encoder frequency limit exceeded, measuring system 2, 645, 646

Encoder limit frequency exceeded, measuring system 1, 645
Error code, 500
Error during oscillation, 693
ESR DC link undervoltage, 683, 684
ESR: Generator operation - minimum speed fallen below, 684
ESR: Response initiated or generator operation active, 684
Ethernet handwheel is stationary, 475
Extended address, F function 1 - 6, 566, 567
Extended address, M function 1 - 5, 564, 565
Extended address, S function 1 - 3, 565
External oscillation reversal active, 692, 693
Feedrate disable, 513, 514
Feedrate override selected for rapid traverse, 531
Feedrate override, axis-specific, 594, 595, 596
Feedrate stop, geometry axis 1, 2, 3), 522
Feedrate, positioning axis, 667
Feedrate/spindle stop, axis-specific, 613
Fixed stop reached, 654
Following spindle active, 692
Follow-up active, 649, 650
Follow-up mode, 601
Function selection from the PLC: Function number, 498
Function selection from the PLC: Parameter 1 - 3, 499
Function selection from the PLC: Request strobe, 499
Function selection from the PLC: Status, 499
Gear is changed over, 624, 625
Gear ratio selection, 635
Geometry axis 1: Activate handwheel, 521, 522
Geometry axis 1: active machine function, 561
Geometry axis 1: handwheel active, 558, 559
Geometry axis 1: handwheel direction of rotation inversion active, 561, 562
Geometry axis 1: invert handwheel direction of rotation, 526
Geometry axis 1: machine function request, 525, 526
Geometry axis 1: Rapid traverse override, 523
Geometry axis 1: traversing command Plus / Minus, 560, 561
Geometry axis 1: Traversing key lock, 522
Geometry axis 1: traversing keys plus / minus, 523, 524, 525
Geometry axis 1: traversing request Plus / Minus, 559, 560
Geometry axis 2: Activate handwheel, 526
Geometry axis 2: active machine function, 562
Geometry axis 2: handwheel active, 562
Geometry axis 2: handwheel direction of rotation inversion active, 562
Geometry axis 2: invert handwheel direction of rotation, 527
Geometry axis 2: machine function request, 527
Geometry axis 2: Rapid traverse override, 527
Geometry axis 2: traversing command Plus / Minus, 562
Geometry axis 2: traversing keys plus / minus, 527
Geometry axis 2: traversing request Plus / Minus, 562
Geometry axis 3: Activate handwheel, 527
Geometry axis 3: active machine function, 563
Geometry axis 3: handwheel active, 563
Geometry axis 3: handwheel direction of rotation inversion active, 563
Geometry axis 3: invert handwheel direction of rotation, 528
Geometry axis 3: machine function request, 528
Geometry axis 3: Rapid traverse override, 528
Geometry axis 3: traversing command Plus / Minus, 563
Geometry axis 3: Traversing key lock, 527
Geometry axis 3: traversing keys plus / minus, 528
Geometry axis 3: traversing request Plus / Minus, 563
Geometry monitoring, 671
Grinding wheel peripheral speed active, 672
H function 1 - 3 and extended address H function 1-3, 566
Handwheel 1 selected, 445
Handwheel 2 selected, 446
Handwheel 3 selected, 447
Handwheel active, 656
Handwheel direction of rotation inversion active, 659
Handwheel offset (DRF) selected, 529
Handwheel override active, 542, 652
Hardware limit switch minus:, 619
Hardware limit switch plus, 619
Heat sink temperature prewarning, 680
HMI battery alarm, 448
HMI temperature limit, 448
Indexing axis in position, 666
Input signals 1 - 8, 592
Integrator disable, speed controller, 678
Interrupt handling active, 551
Invert handwheel direction of rotation, 616, 617
Invert M3/M4, 627

JOG approach fixed point active, 663
JOG approach fixed point reached, 664
JOG circular travel, 537, 538
JOG circular travel active, 590
JOG fixed point approach, 621
JOG mode, 476, 477
JOG mode selected, 481, 482
JOG position reached, 664, 665
JOG retract active, 590
JOG retract retraction data available, 590
JOG travel to position, 621, 622
JOG travel to position active, 664
Key lock, 488
Key-operated switch position, 438, 439
Last action block active, 541
Leading spindle active, 691
Loading/unloading without magazine movement, 707
Location in buffer magazine (spindle), 714, 715
Location No. (source) for new tool, 715
Location No. (Source) for unloading/relocating/positioning, 706
Location No. (target) for old tool, 716
Location No. (Target) for unloading/relocating/positioning, 706, 707
Location no. of new tool to be loaded at change, 724, 725
Location no. of old tool to be unloaded, 725
Location No. of the loading/unloading point, 705
Lubrication pulse, 665
M fct. 1-5 not decoded, 564
M function for spindle, 675, 676
M(d) less than M(dx), 681
M, S, T, D, H, F fct. Additional quick information (fast acknowledgment), 564
M, S, T, D, H, F fct. change, 563
M00/M01 active, 541
M01 selected, 530
M02/M30 active, 543, 544
Machine axis handwheel 1, 446
Machine axis handwheel 2, 447
Machine axis handwheel 3, 447
Machine function request, 615, 616
Machine-related protection zone 1 - 10 preactivated, 568, 569
Machine-related protection zone 1 - 10 violated, 569
Magazine No. (source) for new tool to be loaded, 715
Magazine No. (Source) for unloading/relocating/positioning, 705
Magazine No. (target) for old tool to be removed, 715, 716, 725
Magazine No. (Target) for unloading/relocating/positioning, 706
Magazine no. of new tool, 724
Manual stroke initiation: Acknowledgment, 556
Master/slave: Active, 686
Master-slave: activate coupling, 637, 638
Master-slave: activate torque compensatory controller, 637
Master-slave: Coarse speed difference, 685
Master-slave: Compensatory controller active, 685, 686
Master-slave: Fine speed difference, 685
MCS coupling: Activate collision protection, 636
MCS coupling: Collision protection active, 658
MCS coupling: Coupling active, 687
MCS coupling: Deactivate or do not permit, 636
MCS coupling: Following axis, 687
MCS coupling: offset change, 688
MDI mode, 476
MDI mode selected, 481
Measurement active, 653
Mirroring active, 687, 688
Mode change acknowledgment, 574, 575
Mode change inhibit, 477
Mode group ready, 483, 484
Mode group reset, 478, 479
Mode group stop, 477
Mode group stop axes plus spindles, 478
Modulo rotary axis: activate traversing range limits, 620
Modulo rotary axis: traversing range limits active, 663
Motor being selected, 632, 633
Motor temperature prewarning, 679
Motor/drive data set: Display, 677
Motor/drive data set: Formatting, 696
Motor/drive data set: Selection, 632
Multitool location distance, 729, 731, 738
Multitool location number, 729, 730, 731, 737
Multitool location number (new tool), 732, 738
Multitool location number (old tool), 733, 739
Multitool number, 730
Multitool number (new tool), 732, 738
Multitool number (old tool), 732, 739
n(act) equals n(set), 683
n(act) less than n(min), 682
n(act) less than n(x), 682, 684, 685
NC alarm is active, 451
NC alarm with machining stop is active, 552
NC alarm with program stop, 557

NC battery alarm, 452
NC CPU ready, 448
NC program: Load, 490
NC program: Selection, 490
NC program: Unload, 490
NC ready, 450, 451
NC start, 517
NC start disable, 516, 517
NC stop, 518
NC Stop at block limit, 517
NC Stop axes plus spindles, 518, 519
NCU heat sink temperature alarm, 451
NCU link: Active, 449
NCU link: Axis active, 644
NCU link: Axis ready, 649
New tool: Internal T no. of the NC, 728
New tool: Internal T number of the NC, 718, 719, 735, 742
New tool: Location type, 716, 725, 726, 733
New tool: Location type), 739
New tool: Size bottom, 717, 718, 727, 734
New tool: Size left, 716, 717, 726, 733
New tool: Size left), 740
New tool: Size right, 717, 726, 734
New tool: Size right), 740
New tool: Size top, 717, 726, 727, 734
New tool: Size top), 740, 741
Online tool length compensation (TOFF) active, 573
Online tool length compensation (TOFF): Compensation motion active, 573, 574
Operating software ready, 450
Orientation axis 1: Activate handwheel, 576, 577
Orientation axis 1: active machine function, 586
Orientation axis 1: handwheel active, 583, 584
Orientation axis 1: invert handwheel direction of rotation, 581
Orientation axis 1: machine function request, 580, 581
Orientation axis 1: Rapid traverse override, 578
Orientation axis 1: traversing command Plus / Minus, 585, 586
Orientation axis 1: Traversing key lock, 577, 578
Orientation axis 1: traversing keys plus / minus, 578, 579, 580
Orientation axis 1: traversing request Plus / Minus, 584, 585
Orientation axis 2: Activate handwheel, 581
Orientation axis 2: active machine function, 587
Orientation axis 2: handwheel active, 586
Orientation axis 2: invert handwheel direction of rotation, 582
Orientation axis 2: machine function request, 582
Orientation axis 2: Rapid traverse override, 582
Orientation axis 2: traversing command Plus / Minus, 587
Orientation axis 2: Traversing key lock, 581
Orientation axis 2: traversing keys plus / minus, 582
Orientation axis 2: traversing request Plus / Minus, 587
Orientation axis 3: Activate handwheel, 582
Orientation axis 3: active machine function, 587
Orientation axis 3: handwheel active, 587
Orientation axis 3: invert handwheel direction of rotation, 583
Orientation axis 3: machine function request, 583
Orientation axis 3: Rapid traverse override, 582
Orientation axis 3: traversing command Plus / Minus, 587
Orientation axis 3: Traversing key lock, 582
Orientation axis 3: traversing keys plus / minus, 583
Orientation axis 3: traversing request Plus / Minus, 587
Original location of new tool, 719, 720, 736, 737, 743
Original location of new tool in this turret magazine, 728
Original magazine of new tool, 719, 736, 742
Oscillation active, 693, 694
Oscillation cannot be started, 693
Oscillation controlled by the PLC, 627, 628
Oscillation enable, 628
Oscillation reversal from external, 638, 639
Oscillation: Clockwise direction of rotation, 628, 629
Oscillation: Counterclockwise direction of rotation, 629
Overlaid motion, 690
Override active, 604, 605
Parameter set definitions disabled by NC, 618
Path axis, 665
Path feedrate override, 509, 510, 511
Path feedrate override active, 516
Path rapid traverse override, 511, 512, 513
Path rapid traverse override active, 515, 516
PLC action completed, 505
PLC hardkeys, 490
PLC index, 491
PLC line offset, 491
PLC-controlled axis, 642, 643, 655
PLC-controlled axis: Reset, 639
PLC-controlled axis: Resume, 639, 640

PLC-controlled axis: Stop along braking ramp, 641, 642
PLC-controlled axis: Stop at the next reversal point, 641
Position controller active, 650, 651
Position measuring system 1 activated, 694
Position measuring system 1/2, 602, 603, 604
Position measuring system 2 activated, 694
Position reached with exact stop coarse, 647
Position reached with exact stop fine, 647, 648
Position restored, measuring system 1, 662
Position restored, measuring system 2, 662
Positioning axis, 666
Power up completed, 680, 681
Probe actuated, 449
PROG_EVENT trigger event, 589
Program level abort, 515
Program selection from the PLC: Active, 495, 496
Program selection from the PLC: Error, 495
Program selection from the PLC: Error code, 497, 498
Program selection from the PLC: Index of the program list, 491, 492
Program selection from the PLC: Job completed, 495
Program selection from the PLC: Load, 496, 497
Program selection from the PLC: Program index in the program list, 492, 493
Program selection from the PLC: Selection, 497
Program selection from the PLC: Unload, 496
Program state aborted, 548
Program state interrupted, 547, 548
Program state running, 545, 546
Program state stopped, 547
Program state wait, 546, 547
Program test (PRT) active, 544, 545
Program test (PRT) selected, 531, 532
PTP traversing active, 572
Pulse enable, 633, 634
Pulses enabled, 679
Punch interface 1: Manual stroke initiation, 507, 508
Punch interface 2: Manual stroke initiation, 509
Ramp-function generator disable, 631, 676
Rapid traverse override, 614
Read-in disable, 514
Read-in disable is ignored, 555
Recall alarms deleted, 493
REF machine function, 480
REF machine function active, 485
REF machine function selected, 482
Reference point value 1-4, 609
Referenced / synchronized 1, 646
Referenced/synchronized 2, 646, 647
Referencing active, 542
Remote diagnosis active, 447
REPOS delay, 576, 618, 619
REPOS delay acknowledgment, 661
REPOS delay active, 662, 663
REPOS machine function, 479
REPOS machine function active, 484, 485
REPOS mode selected, 482
REPOS offset, 660
REPOS offset valid, 660, 661
Request disabling of synchronized actions, 570
Reset, 519, 520
Reset executed, 654, 655
RESU: Backward/forward, 500
RESU: Retrace mode active, 539, 540
RESU: Retrace support active, 540
RESU: Start retrace support, 501
Resynchronize spindle, measuring system 1, 625
Resynchronize spindle, measuring system 2, 625, 626
Resynchronizing the spindle during positioning, measuring system 1, 626
Resynchronizing the spindle during positioning, measuring system 2, 626, 627
Revolutional feedrate active, 542, 653
S function for spindle, 676
SBH / SG deselection, 634
SBH deselection, 634
Screen is dark, 493
SE selection, 635
Second software limit switch minus, 620
Second software limit switch plus, 620
Selection: Coordinate system for Cartesian manual traversing and handwheel override in automatic mode in the tool direction (DRF), 593, 594
Selection: Position controller parameter set, 618
Selects the file system, 491
Sensor for fixed stop, 597
Sensor S1 available (clamped state), 697
Sensor S4 available (piston position), 697
Sensor S4, piston end position, 699
Sensor S5 available (angular position of the motor shaft), 697
Sensor S5, angular position of the motor shaft, 699, 700
Sensors available, 696
Set gear stage, 667, 668
Set reversal point, 640
Setpoint limitation, 643

Setpoint speed increased, 670
Setpoint speed limited, 669, 670
Setpoint switchover: drive control active, 686
Setpoint switchover: request drive control, 637
SG selection, 634
Signal identifier for loading/unloading point, 705
Simulation active, 494
Simulation contour handwheel: Activation, 536
Simulation contour handwheel: Negative direction, 537
Single block, type A, 480, 481
Single block, type B, 480
Skip block selected (SKP), 532, 533
Software cams active, 652
Software cams: Activation, 605
Software cams: minus cam signal 1 to 32, 452, 453
Software cams: plus cam signal 1 to 32, 453
Sparking-out active, 693
Speed controller active, 651
Speed controller integrator disable, 633
Speed limit exceeded, 668, 669
Speed monitoring, 671, 672
Speed override, spindle-specific, 629, 630, 631
Spindle, 644
Spindle in position, 675
Spindle in setpoint range, 671
SPL_READY, 485
State of the clamping system, 698, 699
Status axis/spindle replacement, 659
Status of action 1 - 8, 592
Status value is generated, speed limit p5043 is active, 698
Stop at block end is ignored during single block (SBL), 555, 556
Stop at end of block due to configured stop, 558
Stop at end of block due to single block, 557
STOP_E, 485
Stroke enable, 507
Stroke initiation active, 556
Stroke inoperative, 508, 509
Stroke suppression, 508
Suppress program test, 622, 695
Switch over MCS/WCS, 494
Synchronized action ID/IDS can be inhibited, 571
Synchronous operation fine, 688, 689
Synchronous operation, coarse, 689
T function 1, 565
T0, 721, 722
Tapping without compensating chuck active, 672, 673
Target number of workpieces reached, 571
TEACH IN machine function, 479
TEACH IN machine function active, 484
TEACH IN machine function selected, 482
Time monitoring active, 504
Tool holder, 730, 736, 742
Tool management No., 704, 713
Tool management number, 723
Tool status for new tool, 718, 727, 728
Tool status for new tool:, 735, 741
Tool with dynamic response limitation, 674, 675
TOOLMAN: Deactivate wear monitoring, 534, 535
TOOLMAN: Deactivate workpiece counter, 534
TOOLMAN: Last replacement tool of the tool group, 589
TOOLMAN: Tool limit value reached, 588
TOOLMAN: Tool lock not active, 535
TOOLMAN: Tool missing, 572
TOOLMAN: Tool prewarning limit reached, 588
TOOLMAN: Transition to new replacement tool, 588, 589
Transformation active, 544
Traversing command Plus / Minus, 657
Traversing key lock, 614
Traversing key lock, geometry axis 2, 527
Traversing keys plus / minus, 614, 615
Traversing request Plus / Minus, 657
User-definable parameter 0 (DInt), 713, 723
User-definable parameter 1 (DInt), 713, 714, 723, 724
User-definable parameter 2 (DInt), 714, 724
Variable signaling function, 683
Velocity alarm threshold reached, 690, 691
Velocity/spindle speed limitation, 611, 612

K

Key disable, 113

M

M decoding acc. to list, 83
Machine control panel
M version, 308, 309
Slimline version, 310, 311
T version, 309, 310
MCP identification, 94
MD14504, 87
MD14506, 87
MD14508, 87
MD35400, 260

MD51041, 110
MD51043, 110
MD9106, 110
Memory requirements
of basic PLC program, 129
Mode group, 74
Multitool, 199

N

NC
failure, 77
-Read/write variables, 80
NC VAR selector, 133
Alias name, 143
Commissioning, installation, 148
Compiling, 143
Create a project, 138
Create project list, 141
Filtering the variable list, 139
Open project, 138
Save project, 138
Software version, 139
Transfer to STEP 7, 145
Number range
for user program, 105

O

Organization block OB, 305

P

PLC
Basic program functions, 31
HMI monitor, 123
Key disable, 113
Program list, 109, 110
-Versions, 31
PLC basic program
Assignment overview, 105
PROFIBUS
connection, 96
PROFIBUS connection, 96
PROFIBUS diagnostics, 76
PROFINET
connection, 98
Program
Select via PLC, 110

R

REPOS activation, 539
REPOS mode, 538

S

Signal exchange
Cyclic, 29
Event-controlled, 29
Signals
Compile cycles, 57
NCK/PLC, 58
PLC / mode group, 58, 59
PLC/axes, spindles, 61
PLC/NC, 57
PLC/NCK channels, 59
Star-delta changeover, 254
Startup and synchronization of NCK PLC, 73
Symbolic programming, 81

T

Timer No., 308

U

UDT assignments, 82
UTD-blocks, 81