



Start Up Guide

PA 8000

Edition	10.13
Software Revision	3.x
Copyright	PA
SUBJECT TO TECHNICAL MODIFICATIONS AND ERRORS	

Contents

1 General information.....	6
2 How to set up the PA 8000.....	7
2.1 Preparation	7
2.2 Configure the PLC inputs and outputs.....	8
2.2.1 General.....	8
2.2.2 Structure of the input / output configuration file.....	9
2.2.3 PAMIO components.....	9
2.2.4 PCI Sercos board onboard inputs / outputs	10
2.2.5 PA 8000 with analog axes and manually configuration.....	10
2.2.6 PA 8000 with 4ENC4A and manually configuration	11
2.2.7 PA 8000 with Sercos axes and automatically configuration.....	12
2.3 Configure the drives	14
2.3.1 General.....	14
2.3.2 Structure of analog drives configuration file.....	15
2.3.3 Structure of Sercos drives configuration file	16
2.4 Set-Up the Axes.....	18
2.4.1 Allowed programmable letters and name of the axes	18
2.4.2 Maximum axis velocity - AxisSpeedMaxAppl.....	22
2.4.2.1 Calculation for analog axes.....	22
2.4.2.2 Calculation for SERCOS drives	23
2.4.3 Axis velocity in Manual mode – SAxisFeedAppl	23
2.4.4 Axis acceleration – AxisSlopeTime.....	24
2.4.5 Travel distance per encoder pulse - MachToInternalIncr.....	25
2.4.5.1 Calculation for analog axes.....	25
2.4.5.2 Calculation for SERCOS drives	28
2.4.6 Position loop gain - GainSpeedFactor	28
2.4.6.1 Calculation for analog axes.....	29
2.4.6.2 Calculation for SERCOS drives	31
2.4.7 Position loop location – SercosPositionControl (only for SERCOS drives)	32
2.4.8 Set up the override function	33
2.4.8.1 Hardware location of override switches – AdditionKeyIOAddress	33

2.4.8.2 Override functionality - OverrideAppl	34
2.4.9 First positioning test.....	35
2.4.10 Additional axis machine Parameters.....	37
2.4.10.1 Limit Values for Ramp Funktionen – AxisSlopeSpeedAppl	37
2.4.10.2 Standstill Lag check – StandstillLagPerCent.....	37
2.4.10.3 In position window – InpositioningArea	38
2.4.11 Circle KV Element – CircleKVAppl.....	38
2.4.12 Limit accerelation – CircleSpeedKVAppl	39
2.4.13 Allowed contour error of circles – CircleContourError.....	39
2.5 Homing the axes.....	41
2.5.1 General.....	41
2.5.2 Reference sequence.....	41
2.5.3 Software limit switches	44
2.5.4 Mandatory homing - RefAxesAppl	44
2.5.5 Axes homing sequence - AxisSequence	45
2.5.6 Homing search direction - RefDirectionAppl	46
2.5.7 Homing first search velocity - RefVelocity1Appl.....	47
2.5.8 Homing second search velocity - RefVelocityAppl.....	48
2.5.9 Homing travel distance - RefPositionDistance	49
2.5.10 Homing set position - RefPositionValue.....	49
2.5.11 Distance between two marker pulses - MarkerDistance	49
2.5.12 Maximum input frequency - MaxRMSFrequency	50
2.5.13 Software limit switches – SoftwareLimiPlus and SoftwareLimitMinus.....	51
2.5.14 Special axes properties – RefCycleType	52
2.6 Spindle Set-up	53
2.6.1 Spindle Output Channel – SpindleOutputAppl.....	53
2.6.2 Spindle Feedback.....	55
2.6.3 Spindle Speed – SpindleMaxSpeedAppl	55
2.6.4 Spindle Output Polarity - SpindleReversalAppl.....	56
2.7 PLC program.....	57
2.7.1 General.....	57
2.7.2 The PLC project 'Set_up.pro'	58

2.7.2.1 Variable declaration	59
2.7.2.2 PLC program	60
2.7.3 CNC-PLC variables used by 'Set_up.pro'	61
2.7.3.1 Emergency stop - INEMERGENCn.....	61
2.7.3.2 NC Start - IN_START	62
2.7.3.3 NC Stop - IN_STOPn.....	66
2.7.3.4 Block transfer enable - IN_TRANSF	67
2.7.3.5 Feed enable all axes - IN_ENABLE	69
2.7.3.6 Position loop enable - IN_DRIVEON.....	71
2.7.3.7 Axis feed enable - IN_DRIVEEN.....	74
Appendix 1 Machine Parameter Tool	76
Appendix 1.1 General.....	76
Appendix 1.2 Edit and save	78
Appendix 1.3 Load the machine parameters into the CNC.....	80
Appendix 1.4 Output the actual machine parameters.....	81
Appendix 2 Programmable Logic Controller tool	82
Appendix 2.1 General.....	82
Appendix 2.2 Load project	83
Appendix 2.3 Download project.....	84
Appendix 2.4 Online mode	85
Appendix 2.5 Write and Force variables	87
Appendix 3 Customized push buttons.....	90
Appendix 3.1 Example: Machine Control Panel	90
Appendix 3.2 Corresponding PLC program.....	91
Appendix 3.3 Corresponding machine parameter.....	94
Appendix 3.3.1 Handwheel factors - ExtModeHandwheelFeed	94
Appendix 3.3.2 Increment values - ExtModeJogInkr.....	94
Appendix 3.3.3 Continuous jog speeds - ExtModeManFeed	95
Appendix 3.4 PLC program for customized push buttons	96
Appendix 4 PA cables	107



1 General information

This Manual is a guideline how skilful experienced electrical staff can set up the axes of a CNC machine tool with the PA 8000 CNC control.

This documentation is not especially made for specific manufacturing technologies like milling or turning, but the axes of standard milling and turning machines and 'flat' (2.5D) cutting machines can be set up easily.

It is neither a standard programming nor a standard operating manual. Both are available separate at Power Automation.

2 How to set up the PA 8000

2.1 Preparation

Do the following before set-up of the CNC for your specific machine:

- Become familiar with the Machine Parameter Tool (MP Tool) editor. A procedure for working with this tool can be found in the Appendix of this Manual.
- Make sure that the control is wired according to the connection requirements. This is especially true for correct polarity and for the grounding system. Please follow first the instructions of the chapters 1 to 6.
- Observe safety in moving the machine axis.
- Set-up the drives to operate at maximum velocity in plus (+) direction at 8V (Feed override = 100%)
- Make sure that the emergency stop limit switch and the axis limit switches are functional.



2.2 Configure the PLC inputs and outputs

2.2.1 General

Before using machine input/output modules from PLC program or CNC functions all I/O modules have to be configured for your application.

To modify the I/O configuration file, use the control HMI. Within the control HMI you will find below the menu item **SETUP** the Softkey F1 **PLC**, Softkey F3 'I/O configuration'. Press this Softkey; this will open a text editor to modify the I/O configuration file. After the modifications are done, save the file with the original name – **PAIOCfg.ini**. The modifications will become active with the next start-up of the CNC control.

This I/O configuration file lists all I/O modules used in your application and defines the logical addresses (byte numbers) to access the hardware modules by software (PLC -program and CNC software functions).

The configuration of logical addresses (byte numbers) allows the application programmer (CNC and PLC) to know the hardware addresses of all modules supported by Power Automation.

All controls are shipped with a standard I/O configuration file. You will have to modify this file to your needs (add or remove I/O modules and set up the required logical addresses).

Note:

- The file is called PAIOCfg.ini and can be found under C:\...\Power Automation\User Data.

2.2.2 Structure of the input / output configuration file

The I/O configuration file has to respect following structure :

board number	number of input Bytes	number of output Bytes	address of first input Byte	address of first output Byte	comment (optional)
--------------	-----------------------	------------------------	-----------------------------	------------------------------	--------------------

Each line of the file gives the description of each In / Out-component used in the application.

Note:

- The character "," is used as separator.
- The character ";" has to stand in front of any comment.

2.2.3 PAMIO components

PAMIO 24I16O (24 inputs / 16 outputs)

- Each module has 3 digital input bytes and 2 digital output bytes

PAMIO 4AD4DA (4 analog inputs / 4 analog outputs)

- Each analog input occupies 2 bytes and each analog output occupies 2 bytes

PAMIO 4ENC4A (4 axis-module / 4- analog inputs)

- Each analog input occupies 2 bytes

These components can be specified by following instructions:

- manually by: " **PA-MODULAR-IO Configuration** "
- automatically by: " **PA-MODULAR-IO AUTO Configuration** "

2.2.4 PCI Sercos board onboard inputs / outputs

PCI Sercos board onboard inputs / outputs (24 inputs / 16 outputs)

Each board has 3 digital input bytes and 2 digital output bytes

These inputs / outputs are specified by the instruction:

" PA-FAST-PCI-IO Configuration "

2.2.5 PA 8000 with analog axes and manually configuration

Example:

```

5 PAMIO digital In / Out      (120 in-/80 outputs)
2 PAMIO analog In / Out      (4analog in-/4 analog outputs)

;
;                               PA8000
;                               I/O Bus Configuration
;
PA-MODULAR-IO Configuration
;
1,      3,      2,      1,      101      ; PAMIO 24160
2,      3,      2,      4,      103      ; PAMIO 24160
3,      3,      2,      7,      105      ; PAMIO 24160
4,      3,      2,      10,     107      ; PAMIO 24160
5,      3,      2,      13,     109      ; PAMIO 24160
6,      8,      8,      16,     111      ; PAMIO 4AD4DA
7,      8,      8,      24,     119      ; PAMIO 4AD4DA
;
;                               END of I/O-Bus configuration
;

```

2.2.6 PA 8000 with 4ENC4A and manually configuration

Example:

```
2 PAMIO 4ENC4A      (4 analog IN)
2 PAMIO digital E/A (48 inputs / 32 outputs)
1 PAMIO analog E/A  (4analog in-/4 analog outputs)
```

```
;
;
;          PA8000
;          I/O Bus Configuration
;
PA-MODULAR-IO Configuration
```

;In order to configure the length of the cable insert optional a new line:

```
BUSLENGTH  10
1,      8,      0,      1,      0      ; PAMIO 4ENC4A
2,      8,      0,      9,      0      ; PAMIO 4ENC4A
3,      3,      2,      17,     101     ; PAMIO 2416
4,      3,      2,      20,     103     ; PAMIO 2416
5,      8,      8,      23,     105     ; PAMIO 4AD4DA
;
;
;
;          END of I/O-Bus configuration
;
```

Note:

- As BUSLENGTH have to the length of the Superbus in meters (integer number)
- BUSLENGTH is always in meters (also in USA)

- The practicable speed at the Superbus is depending from the length of the cable and the revision of the connected PAMIO-modules and PCI-board.
- Without the BUSLENGTH definition automatically Slow Speed is active. The line with BUSLENGTH definition has to be between PA-MODULAR-IO Configuration and module configuration.

2.2.7 PA 8000 with Sercos axes and automatically configuration

Example:

```

1 Sercos board          ( 24 in-/16 outputs)
5 PAMIO digital In / Out (120 in-/80 outputs)

;
;          PA8000
;          I/O Bus Configuration
;
PA-FAST-PCI-IO Configuration
;          3 Bytes in 2 Bytes out on PA-PCI-Sercos Board
;
1,      3,      2,      1,      101      ; 3 Bytes In, 2 Bytes out
;
;          PA with PAMIO automatic configuration
;
PA-MODULAR-IO AUTO configuration IN 4, OUT 103
;
;          END of I/O-Bus configuration

```



Note:

- In case of Auto Configuration, if no input / output addresses are specified, than by default the first input byte will have the address 1 and the first output byte will have the address 100.
- But it is possible to specify any start address as follows:

“ PA-MODULAR-IO AUTO Configuration IN 4, OUT 103 ”

In this case the first input byte of the used PAMIO components will have the address 4, and the first output byte will have the address 103.

2.3 Configure the drives

2.3.1 General

The hardware output channels to the CNC-axis and the spindles are assigned in the drive configuration file.

The output channels are assigned with numbers 1 to 8 for all possible output channels and correspond with hardware axes numbers 1 to 8. Axes, spindles and laser power control can be connected to output channels as indicated in the **Drivecfg.ini** file.

All controls are shipped with a standard drives configuration file. You will have to modify this file to your needs (add or remove drives and set up the required hardware addresses).

To modify the drives configuration file, use the control HMI. Within the control HMI you will find below the menu item **SETUP** the Softkey F2 Machine Setup, Softkey F2 'Drive configuration'. Press this Softkey; this will open a text editor to modify the drives configuration file. After the modifications are done, save the file with the original name – DriveCfg.ini. The modifications will become active with the next start-up of the CNC control.

Note:

- The file is called DriveCfg.ini and can be found under C:\...\Power Automation\User Data.
- The hardware output number may be different than the display index!! The display index of the axis is defined by the machine parameters in the "NCAddressFormat" group. This is the order in which the axes are displayed on the HMI. All the axis dependent machine parameters are related to the display index and not to the hardware output number!!!

2.3.2 Structure of analog drives configuration file

The drives configuration file has to respect following structure :

hardware output number	CNC channel number	axis character
------------------------	--------------------	----------------

Each line of the file has to correspond with each drive used in the application.

Note:

- The character "," is used as separator.
- The character ";" has to stand in front of any comment.

Example:

```
; drive configuration file
; analog axes
```

ANALOG

; output #	channel #	axis character
ADDR = 1,	1,	X
ADDR = 2,	1,	Y
ADDR = 3,	1,	Z
ADDR = 4,	1,	C
ADDR = 5,	1,	A
ADDR = 6,	1,	W
;ADDR = 7,	1,	S
;ADDR = 8,	1,	L

Note:

- An axis character 'S' stands for a spindle output.
- An axis character 'L' stands for laser power control output.

2.3.3 Structure of Sercos drives configuration file

The drives configuration file has to respect following structure :

hardware output number	parameter file name	CNC channel number	axis character
---------------------------	------------------------	-----------------------	----------------

Each line of the file has to correspond with each drive used in the application.

The name given under 'parameter file name' is the one that the control will use to save the Sercos ID-data of the corresponding axes, when this is required in the CNC HMI Sercos menu.

Note:

The character "," is used as separator.

The character ";" has to stand in front of any comment.

Example:

```

; drives configuration file
SERCOS
T_SCYC = 1.0 MS      ;SERCOS cycle time in ms,
                    ;has to be the same as the position-loop time,
                    ;this value is pre-set and should not be modified
T_OFFSET = 0.165 MS ;Time-delay for position-controller in ms
                    ;this value is pre-set and should not be modified
T_DEL  = 300         ;jitter on the ring in µs
; first Sercos ring
RING = 1             ;start of the first ring
                    ;all the following axes will be located to the
                    first ring
VAL_TXD = 3          ;transmitter-current
FREQ = 4000          ;frequency of transmitter in kHz
; output #    parameter file name    channel #    axis character
ADDR = 1,     DRIVEX.PAR,            1,            X
ADDR = 2,     DRIVEY.PAR,            1,            Y
ADDR = 3,     DRIVEZ.PAR,            1,            Z
ADDR = 4,     DRIVEA.PAR,            1,            A
; second Sercos ring
RING = 2           ;start of the second ring
                    ;all the following axes will be located to the
                    second ring
VAL_TXD = 3        ;transmitter-current for second ring
FREQ = 4000        ;frequency of transmitter of second ring
; output #    parameter file name    channel #    axis character
; ADDR = 1,    DRIVEB.PAR,            1,            B
; ADDR = 2,    DRIVEC.PAR,            1,            C
; ADDR = 3,    DRIVES.PAR,            1,            S

```

Note:

- An axis character 'S' stands for a spindle output.
- Do not change the first part of the file without consult PA.

2.4 Set-Up the Axes

The following machine parameters are these machine parameter, which are at least required to ensure safe and proper axes movement of machine tools, , regardless manufacturing technology.

2.4.1 Allowed programmable letters and name of the axes

The machine parameter group **NCAddressFormat** consists of 26 different machine parameters '**CharacterAppITab(x)**', each of them giving the format and the assignment of the corresponding letter of the alphabet. This way the allowed programmable characters of the alphabet are defined.

It determines axis names, axis display index and the total number of programmable digits before and after the decimal point as well.

Each character in the alphabet which will be allowed in the programming (including axis names) has to be defined with 6 values (6 words).

The following input definitions are valid:

1st value: metric digits

Maximum admissible total number of digits for metric displays and inputs. Corresponds to the sum of digits before and after the decimal point.

- If MSB (bit 7) = 0, $\Rightarrow$ Sign is not allowed.
- If MSB (bit 7) = 1, $\Rightarrow$ Sign is allowed.

2nd value: inch digits

Maximum admissible total number of digits for inch displays and inputs.
Corresponds to the sum of digits before and after the decimal point.

- If MSB (bit 7) = 0, $\Rightarrow$ Sign is not allowed.
- If MSB (bit 7) = 1, $\Rightarrow$ Sign is allowed.

3rd value: metric decimal digits

Number of programmable digits after the decimal point for metric displays and inputs.

- If MSB (bit 7) = 0, $\Rightarrow$ Number of decimal digits for display and inputs.
- If MSB (bit 7) = 1, $\Rightarrow$ Number of decimal digits for displays only.

4th value: inch decimal digits

Number of programmable digits after the decimal point for inch displays and inputs.

- If MSB (bit 7) = 0, $\Rightarrow$ Number of decimal digits for display and inputs.
- If MSB (bit 7) = 1, $\Rightarrow$ Number of decimal digits for displays only.

5th value: axis or BCD number

Defines axis display index or BCD index assignment.

Axis display index assignment:

- If value = FFFFh, $\Rightarrow$ No axis or BCD.
- If value = 0h, $\Rightarrow$ this is the 1st axis
- If value = 1h, $\Rightarrow$ this is the 2nd axis
- If value = 1Fh $\Rightarrow$ this is the 32nd axis

BCD index assignment:

- If value = 8000h $\Rightarrow$ this is the 1st BCD
- If value = 8001h $\Rightarrow$ this is the 2nd BCD
- If value = 8002h $\Rightarrow$ this is the 3rd BCD
- If value = 8003h $\Rightarrow$ this is the 4th BCD

6th value:

Defines special axes functions:

Bit 1-8	Lead axis index of a parallel axis
Bit 9-11	Round Axis Type
	100h ⇒ normal round axis
	200h ⇒ shortest way round axes
	300h ⇒ modulo round axes
	400h ⇒ resettable round axis
Bit 13 = 1	1000h ⇒ Positioning Axis
Bit 14 = 1	2000h ⇒ Transverse Axis
Bit 15 = 1	4000h ⇒ Longitudinal Axis
Bit 16 = 1	8000h ⇒ Parallel Axis

Example:

- The letter 'Y' should be the name of the 2nd displayed axis, as linear axis with 1 µm resolution in metric and 0.0001 inch resolution in imperial.

CharacterAppITab(Y)

index	data
1	0088
2	0088
3	0003
4	0004
5	0001
6	0000

- The letter 'A' should be the name of the 4th displayed axis, as 'shortest way' round axis with 0.001 degree resolution in metric and 0.001 degree inch resolution in imperial.

CharacterAppITab(A)

index	data
1	0088
2	0088
3	0003
4	0003
5	0003
6	0200

- The letter 'M' should be the name of the first BCD function.

CharacterAppITab(M)

index	data
1	0002
2	0002
3	0000
4	0000
5	8000
6	0000

2.4.2 Maximum axis velocity - AxisSpeedMaxAppl

The machine parameter **AxisSpeedMaxAppl** has 32 indexes; the index number is related to the CNC axes display index. This machine parameter determines the maximum admissible axis velocity for the respective axis.

Unit: 1000 internal increments / min

2.4.2.1 Calculation for analog axes

AxisSpeedMaxAppl has to be set to the velocity, the axis moves at output voltage 8V for analog axes.

Example:

- linear axis

given: motor speed (for 8V set value)	2000 rpm
given: ratio between motor and lead screw	1/2
given: lead screw pitch	20 mm
AxisSpeedMaxAppl = $2000 * 1/2 * 20 = 20000$	

- rotary axis

given: motor speed (for 8V set value)	2000 rpm
given: ratio between motor and first gear	1/10
wheel	
given: ratio between gear wheels	37/63
AxisSpeedMaxAppl = $2000 * 1/10 * 37/63 * 360 = 42285$	

2.4.2.2 Calculation for SERCOS drives

AxisSpeedMaxAppl has to be set to 80% of **SercosMaxCommandValue** (same group).

linear axis: $= (\text{m/min}) * 10^{\text{decimal places}}$

rotary axis: $= \text{rpm} * 360 * 10^{\text{decimal places}}$

Maximum axis velocity - **SercosMaxCommandValue**

The machine parameter **SercosMaxCommandValue** has 32 indexes; the index number is related to the CNC axes display index. This machine parameter gives the maximum command value to the drives.

Unit: Same unit as the one that the drive uses for velocity scaling. The velocity scaling is defined in digital Sercos drives with indent-numbers 44, 45, 46 and 47.

2.4.3 Axis velocity in Manual mode – **SAxisFeedAppl**

The machine parameter **SAxisFeedAppl** has 32 indexes; the index number is related to the CNC axes display index. This machine parameter determines the desired feed rate in the operation mode 'MANUAL'. This velocity correspond to the feed rate potentiometer position 100%.

Unit: internal increments / ms

Calculation:

linear axis: $= (\text{m/min}) * 10^{\text{decimal places}} / 60$

rotary axis: $= \text{rpm} * 360 / 60$



Example:

- linear axis

given: manual feed rate should be **5 m / min**

$$\text{SAxisFeedAppl} = 5 * 1000 / 60 = 83.3333$$

- rotary axis

given: manual feed rate should be **20 rpm**

$$\text{SAxisFeedAppl} = 20 * 360 * / 60 = 120$$

2.4.4 Axis acceleration – AxisSlopeTime

The machine parameter **AxisSlopeTime** has 32 indexes; the index number is related to the CNC axes display index. This machine parameter determines the time in which the axes should accelerate from standstill to rapid traverse (AxisFeedMaxAppl).

Unit: ms

Calculation:

The settings depend on the drive and axis characteristics; this time should be adjusted in such a way, that the drive never comes in current limit during the acceleration or deceleration phase.

For the first start up of the axis it is recommended to increase this time approximately by 5 to reduce the acceleration as long as the drive set up is not properly achieved.

Example:

- linear axis

given: axis acceleration should be **4 m / s²**

given: AxisSpeedMaxAppl **20000 mm/min**
 (=20 /60 m/s)

$$\text{acceleration time} = \text{speed} / \text{acceleration value}$$

$$= 20 / 60 / 4 = 0.08333 \text{ s}$$

AxisSlopeTime = 83.333

- rotary axis

given: axis acceleration should be **25 rad / s²**

given: AxisSpeedMaxAppl **42285 ° / min**

(=42285 *2π / 360 rad / min)

acceleration time= speed / acceleration value

= (42285 *2π / 360) / 60 / 25 = 0.492 s

AxisSlopeTime = 492.0

Note:

- If the acceleration is not known, it is suggested to start with a value of 1 m/s² for linear axes and a value of 6 rad / s² for rotary axes.

2.4.5 Travel distance per encoder pulse - MachToInternallncr

The machine parameter **MachToInternallncr** has 32 indexes; the index number is related to the CNC axes display index. This machine parameter determines the travel distance between two encoder pulses. The sign represents the counting direction.

Unit: internal increments

2.4.5.1 Calculation for analog axes

The CNC receives encoder pulses. These pulses are multiplied by 4 in the hardware input circuit of the control. After that the software multiplies the resulting counts with the factor “MachToInternallncr”. This result in “internal increments” is the base for all (internal) calculation.

The entire CNC internal calculation is based on this machine parameter value; therefore it is necessary to determine and to enter this value with as much digits as possible.

$$\text{MachToInternallncr} = \frac{\text{distance (mm)}}{(\text{pulse number} * 4) * \text{display resolution (mm)}} * i^*$$

i^* = mechanical ratio between encoder and lead screw

Example:

- linear axis

given: encoder pulses / revolution **2500 p / rev**

given: ratio between encoder and lead screw **1/2**

given: lead screw pitch **20 mm**

given: display resolution **0.001 mm**

$$\text{MachToInternallncr} = 20 / [(2500 * 4) * 0.001] * 1/2 = 1$$

- rotary axis

given: encoder pulses / revolution **10000 p / rev**

given: encoder ratio **37/63**

given: 1 load revolution **360°**

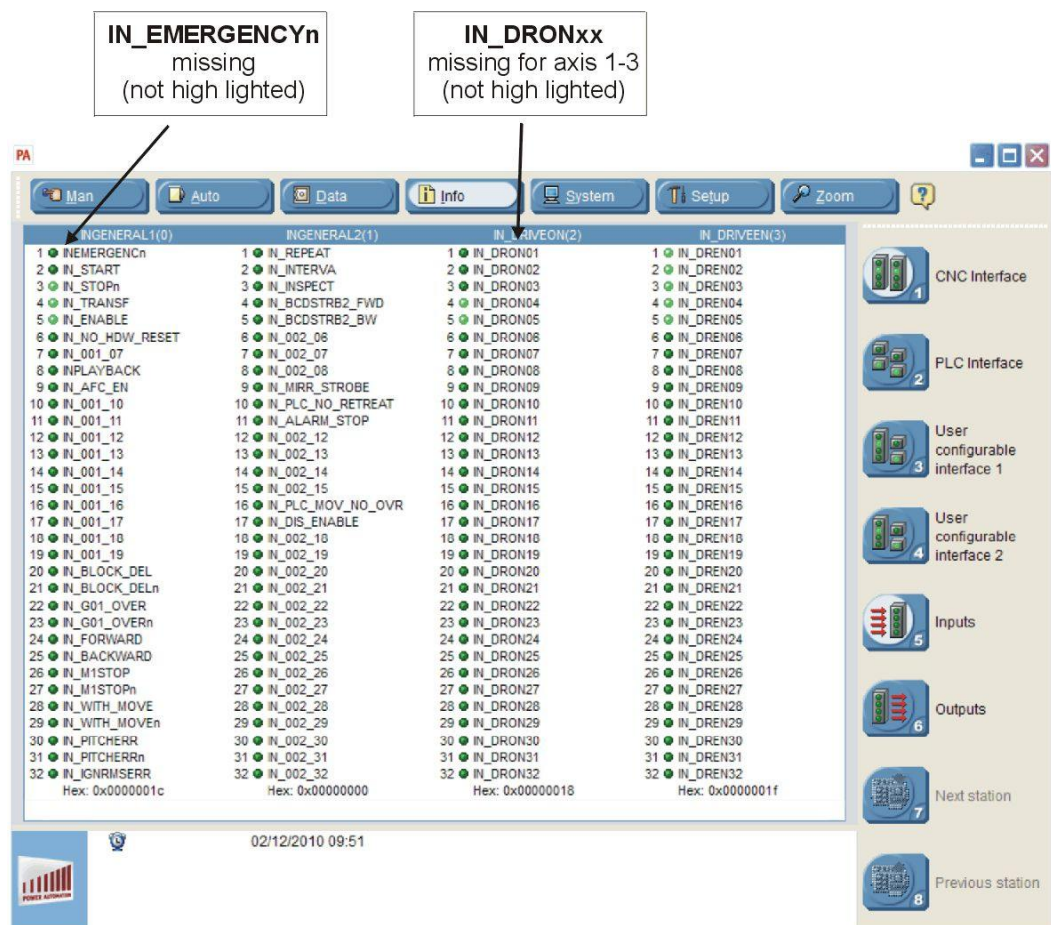
given: display resolution **0.001°**

$$\begin{aligned} \text{MachToInternallncr} &= 360 / [(10000 * 4) * 0.001] * 37/63 \\ &= 5.285714286 \end{aligned}$$

Testing the resolution and the counting direction:

Before executing the following test it has to be ensured, that the modified machine parameters have been activated by downloading the machine parameter file!

If the signal **INEMERGENCn** and / or the signal **IN_DRONxx** for the respective axis are missing (see chapter PLC-Program), then the counting direction and the calculated resolution can be examined by shifting the axes manually. The status of these signals can be checked using the **INFO** mode of the screen and the Softkey F1 'Interface display'.



If the sign of the position display does not correspond with the direction of the axis of the machine, the measuring system counting direction has to be changed; this is achieved by changing the sign of the machine parameter **MachToInternalIncr**. After changing this sign, this test has to be performed again for safety purposes.

The resolution of the measuring system can be roughly verified in the same way; check the axis position display, shift the axis manually and compare the shifted distance with the changing of the axis position display. If there is an obvious difference the calculation of **MachToInternalIncr** should be verified.

2.4.5.2 Calculation for SERCOS drives

In Sercos drives this parameter gives the conversion factor from position values of the drives to the metric internal increments of the controller.

The position scaling in the drives is set up with indent-numbers 76, 77, 78 and 79

Standard: 0.1

2.4.6 Position loop gain - GainSpeedFactor

The machine parameter **GainSpeedFactor** has 32 indexes; the index number is related to the CNC axes display index. This machine parameter determines the maximum following error (lag) admissible for the respective axis and is calculated in relation with the **KV-factor** (position control loop gain).

The **KV-factor** defines the relationship between the feed-rate, the following error (lag) and the axis voltage output to the drives for analog axes.

The smaller the value of **GainSpeedFactor**, the higher the **KV-factor** value, the harder the axis is controlled, and the smaller the allowed following error (lag).

2.4.6.1 Calculation for analog axes

The polarity of the axis voltage output depends on the sign of this value.

The drive is set to maximum speed at 8V at 100% override. When feed override is at 120% the CNC outputs 9,6V; the servo limit is preset CNC side at 10V (125% of 8V).

Definition: KV-factor = Feed rate (m/min) / lag (mm)

Unit: internal increments

The maximum following error for the desired KV-factor must be calculated for each axis according to the following formula and must be entered into the corresponding index of the machine parameter **GainSpeedFactor**.

$$\text{GainSpeedFactor} = \frac{\text{AxisSpeedMaxAppl} * 1,25}{\text{KV-factor}}$$

Example:

- linear axis

given: AxisSpeedMaxAppl **20000 mm/min**

given: desired KV-factor **1**

GainSpeedFactor = (20000 * 1,25) / 1 = 25000

- rotary axis

given: AxisSpeedMaxAppl **42285 mm/min**

given: desired KV-factor **0,5**

GainSpeedFactor = (42285 * 1,25) / 0,5 = 105712,5

Note:

- If the machine and drives characteristics are not known, it is suggested to start with a KV-factor value of 0,5 so that the axes are not controlled to hard (stiff).

Testing the polarity of the axis voltage output

Before executing the following test it has to be ensured, that the modified machine parameters has to be activated by downloading the machine parameter file.

If the **INEMERGENCn** is released and **IN_DRONxx** and **IN_DRENxx** are set (see chapter PLC-Program), the polarity of the axis voltage output can be checked. Once in servo lock, the error message “Lag error during standstill” may occur. If this does happen, the polarity of the D/A output has to be changed.

There are two methods :

- **Hardware solution:**
Swap the plus and minus axis voltage outputs. This will result in a positive axis voltage output for a plus motion.
- **Software solution:**
Change the sign of the machine parameter **GainSpeedFactor**. Changing this sign will result in a plus direction with a plus lag but it could result in a negative axis voltage output.

A further check needs to be done to verify the proper sign of the axis voltage output.

2.4.6.2 Calculation for SERCOS drives

Only if the position loop is closed inside the control, the position loop gain will be defined in the machine parameter **GainSpeedFactor**. Mostly it is located inside the Sercos drive. If any doubt about where the position loop is closed, contact either your Sercos drive supplier or PA technical staff. If the position loop is closed inside the Sercos drives, the KV-factor value used in following calculations for this machine parameter has to be exactly the same as this one, set up in the drive.

Definition: KV-factor = refer to drive set up

Units: internal increments

The maximum following error for the desired KV-factor must be calculated for each axis according to the following formula and must be entered into the corresponding index of the machine parameter **GainSpeedFactor**.

$$\text{GainSpeedFactor} = \frac{\text{AxisSpeedMaxAppl} * 1,25}{\text{KV-factor}}$$

Example:

- linear axis

given: AxisSpeedMaxAppl **20000 mm/min**

given: desired KV-factor **1**

$$\text{GainSpeedFactor} = (20000 * 1,25) / 1 = 25000$$

- rotary axis

given: AxisSpeedMaxAppl **42285 mm/min**

given: desired KV-factor **0,5**

$$\text{GainSpeedFactor} = (42285 * 1,25) / 0,5 = 105712,5$$

Note:

If the machine and drives characteristics are not known, it is suggested to start with a KV-factor value of 0,5 so that the axes are not controlled to hard (stiff).

2.4.7 Position loop location – SercosPositionControl (only for SERCOS drives)

This parameter defines where the position loop is controlled, whether in the CNC control or in the Sercos drives.

Note:

- The position loop is mostly closed inside the Sercos drive. If any doubt about where the position loop is closed, contact either your Sercos drive supplier or PA technical staff.

Each axis number corresponds to a Hex value (see table below). The sum of the Hex values of all axes that shall be set for mandatory homing after power-up must be written into **SercosPositionControl**.

Axis no.	8	7	6	5	4	3	2	1
Hex Value	80	40	20	10	8	4	2	1

Example:

The machine tool has 4 axes with digital Sercos drives.

Calculation:

axis number	Sercos drive	HEX value
1	yes	1
2	yes	2
3	yes	4
4	yes	8
5	no	0
6	no	0
7	no	0
8	no	0
	sum	F

SercosPositionControl = 000F

2.4.8 Set up the override function

In order to be able to move an axis, the override functionality of the CNC must be set up according to the wiring of the override potentiometer. This requires at least the proper definition of following machine parameters in the **FeedOverride** group:

- Hardware location of override switches **AdditionKeyIOAddress**
- Override functionality **OverrideAppl**

2.4.8.1 Hardware location of override switches – **AdditionKeyIOAddress**

AdditionKeyIOAddress = 0

Override switches, jog keys, NC START/NC STOP key are not read in via the I/O bus.

AdditionKeyIOAddress <> 0

- Low byte: The low byte of this machine parameter contains the byte number of the I/O module where the feed rate switch and the Jog-Plus / Jog-Minus switch buttons can be directly read-in.
- High byte: The machine parameter contains the byte number of the I/O module where the spindle rate switch and the NC Start / NC Stop switch buttons can be directly read-in.

The contents of the low - and high byte of **AdditionKeyIOAddress** must correspond to the I/O configuration file.

2.4.8.2 Override functionality - OverrideAppl

Bit 1 - 8: ⇒	Always be set to 00FFh for usage together with bits 9 – 11 and 15 New feature:
Bit 1 = 1: ⇒	Reading feed override value from CNC-PLC interface (low WORD of 15 DWORD IN_OVERRIDE – IN_FEEDOVR)
Bit 2 = 1: ⇒	Reading spindle override value from CNC-PLC interface (high WORD of 15 DWORD IN_OVERRIDE – IN_SPINDLEOVR) Note: the bits 3 – 8, 9 – 11 and 15 should be 0 for the new feature. PLCOverrideByteNo should be FFFFh for the new feature.
Bit 9 = 0: ⇒	The override potentiometer bits are read-in as binary-code.
Bit 9 = 1: ⇒	The override potentiometer bits are read-in as Gray-code.
Bit 10 = 0: ⇒	HMI supplies potentiometer values in binary-code format as far as binary-code is set up.
Bit 10 = 1: ⇒	HMI supplies potentiometer values in Gray-code format in as far as Gray-code is set up.
Bit 11 = 0: ⇒	The switch values are read-in directly.
Bit 11 = 1: ⇒	A value 1 is to be subtracted from the read-in switch values.
Bit 12 = 0: ⇒	The switch values are read-in.
Bit 12 = 1: ⇒	Feed- and spindle-override are permanently set to 100% regardless of the switch position.
Bit 13 = 0: ⇒	Is the feed rate-override in 0% position, then this value is effective regardless of the value programmed in G63.
Bit 13 = 1: ⇒	The value programmed in G63 is effective regardless of the feed rate switch position.
Bit 14 = 0: ⇒	Is the spindle-override in 0% position, then this value is effective regardless of the value programmed in G63.
Bit 14 = 1: ⇒	The value programmed in G63 is effective regardless of the spindle switch position.
Bit 15 = 0: ⇒	When the feed and spindle override per PLC function is active, and if switches are used which delivers values from 0 - 23.
Bit 15 = 1: ⇒	When the feed and spindle override per PLC function is active, and if switches are used which delivers values from 1 - 24.
Default: 01FFh	If feed rate and / or spindle Gray code switches are wired; than this value allows influencing the feed rate and spindling speed with these switches.

Default: 09FFh	If there are no feed rate and / or spindle switches, than this value is necessary to ensure a feed rate override and a spindle override permanently 100%, as otherwise this override would be permanently set to 0 %; no movement would be possible.
-----------------------	--

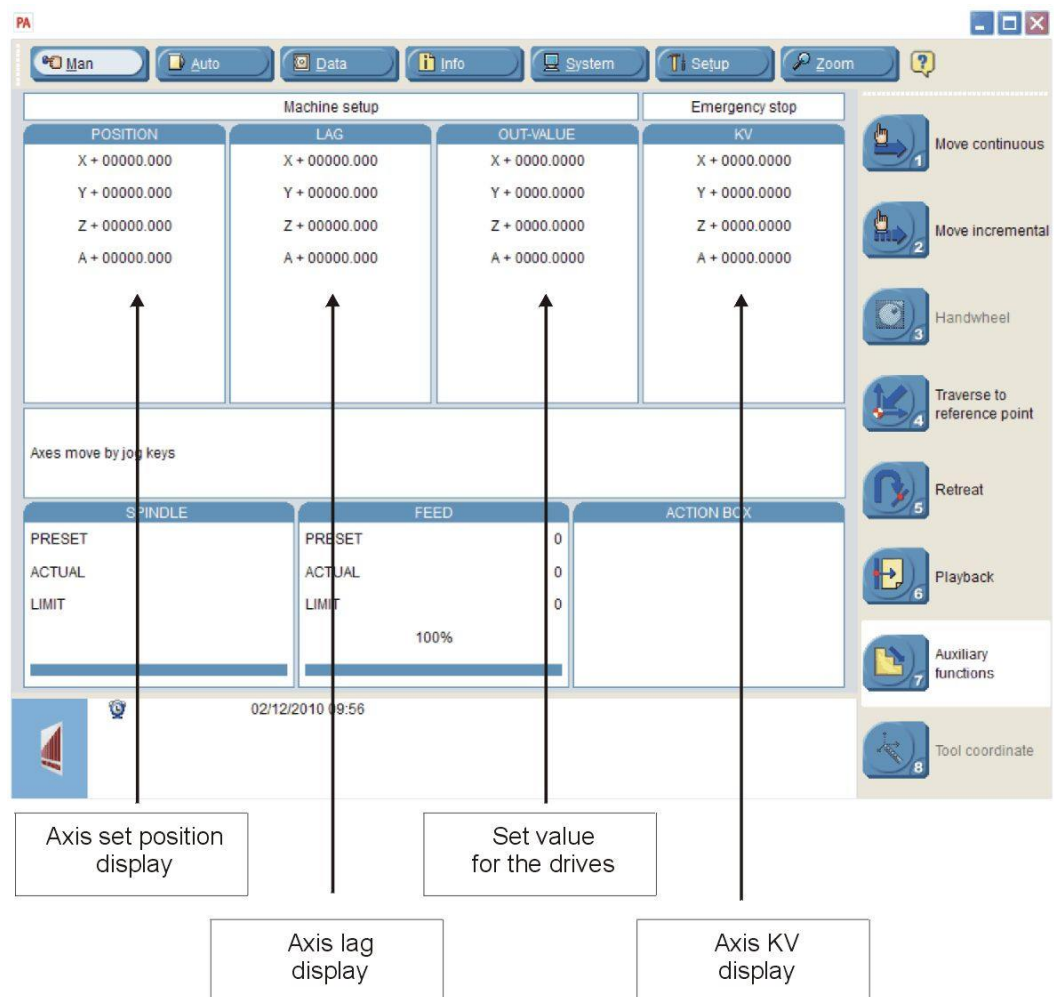
2.4.9 First positioning test

Select the LAG display to monitor the following error.

Set the HMI to **SYStem** mode.

Press F1: Display functions.

Press F2: LAG display.



Axis set position display

Axis lag display

Set value for the drives

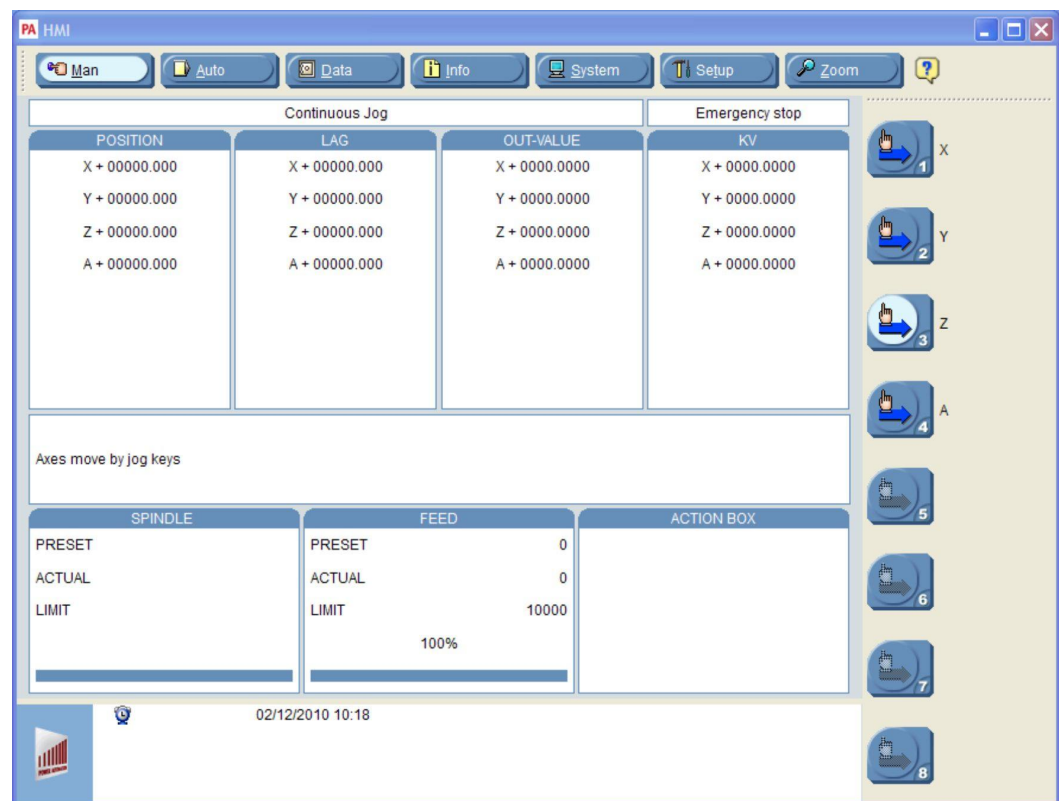
Axis KV display

Select one axis in **MANUAL-Continuous jog**.

Set the HMI to **MANual** mode.

Press **ALT-1: Move continuous**.

Press **ALT-x: Select the axis to be moved**.



Move the selected axis with a reduced feed rate (feed override rotary switch turned down), using the Jog plus / Jog minus keys.

If then this axis travels at 'over speed' during the positioning test, the direction of the motion must be tested again for this axis.

If no error can be detected during the retest, that means that the problem is on the drive side. Then the velocity and the current-control loop, respectively, of the corresponding drive must be checked (signals from the tacho-generator or motor connection may be reversed).

When the axes are in ‘stand still’, all axes displays in the LAG or voltage columns should fluctuate around zero. Any axis offset must be corrected at the drive side.

Note:

- There is no gain or offset adjustment possible inside the CNC

2.4.10 Additional axis machine Parameters

The following machine parameters typically stay with their preset default values. However, in some machines they may need to be modified.

2.4.10.1 Limit Values for Ramp Functions – AxisSlopeSpeedAppl

Unit: 1000 internal increments/min

The velocity at which the axis should ramp up from standstill can be set for each axis in the machine parameter AxisSlopeSpeedAppl. Below this velocity, the CNC outputs a step function of this amount.

2.4.10.2 Standstill Lag check – StandstillLagPerCent

Unit: %

The machine parameter StandstillLagPerCent indicates the allowed lag for each axis of the machine in the standstill mode. The value must be entered in percent of the GainSpeedFactor.

A typical value is 5% to 20%.

0 – turns the function off.

Note:

- If lag exceeds this value, the following error will be generated: “Lag error during standstill on axis x”

2.4.10.3 In position window – InpositioningArea

Unit: internal increments

This machine parameter determines the tolerance with which the axis approaches the pre-defined destination point. With precision positioning (G73), block changes is prevented until the in-position window is reached.

The CNC in-position message can be both monitored by the HMI in the operating mode INFORMATION or by the signal interface.

When entering the value of InpositioningArea the set value of the encoder resolution should be taken into account.

2.4.11 Circle KV Element – CircleKVAppl

Unit: 1/min

This machine parameter is used to check the maximum feed rate of the machine to circles.

This set-up data contains the smallest KV factor of all axes that can participate in the circular interpolation.

Note:

- If this machine parameter is zero, then CNC will use a proper value automatically”

2.4.12 Limit acceleration – CircleSpeedKVAppl

To avoid exceeding the maximum axis acceleration when processing circles, the value for the machine parameter CircleSpeedKVAppl must be calculated according to the following formula and entered:

$$\text{CircleSpeedKVAppl} = \sqrt{\frac{\text{AxisSpeedMaxAppl}(\text{max}) * 60}{\text{AxisSlopeTime}(\text{ms})}}$$

For AxisSpeedMaxAppl take the largest value.

For AxisSlopeTime take the smallest value.

Changing of the value = change of the acceleration and the feed rate of circular interpolation.

Example:

Axis	Rapid Traverse	AxisSpeedMaxAppl
1 st and 2 nd axis	15 m/min	15000 mm/min
3 rd axis	10 m/min	10000 mm/min
4 th axis	25 rpm	9000 degrees/min

$$\text{CircleSpeedKVAppl} = \sqrt{\frac{15000 * 60}{250(\text{standard value})}} = 60$$

Note:

- If this machine parameter is zero, then CNC will use a proper value automatically”

2.4.13 Allowed contour error of circles – CircleContourError

Unit: internal increments



The velocity of circles is limited such that a programmable, admissible contour error is not exceeded.

2.5 Homing the axes

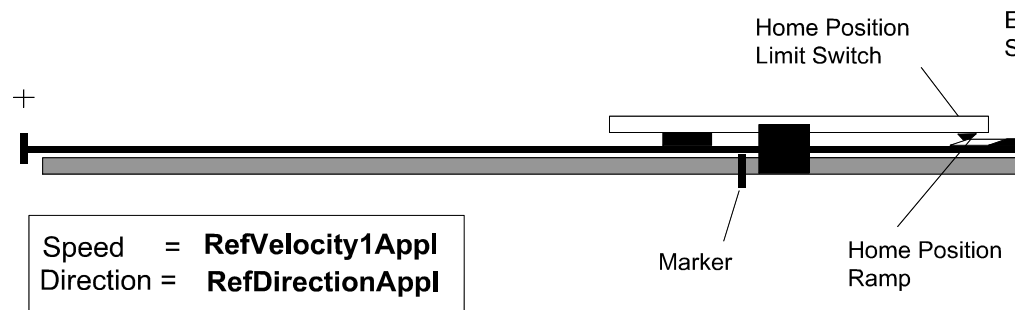
2.5.1 General

If the CNC has to position the axes in a coordinate system fixed in the machine, then it must first define a point on each axis where the coordinate system has its origin. This point is called the home position (or machine zero point). The home position is determined for each axis with mandatory homing by means of a limit switch signal (normally closed contact NC) and the marker signal from the measuring system.

The use of pre-set data allows the machine tool builder to decide whether or not to use mandatory homing. If homing is used, the sequence of machine axis homing can be set-up.

2.5.2 Reference sequence

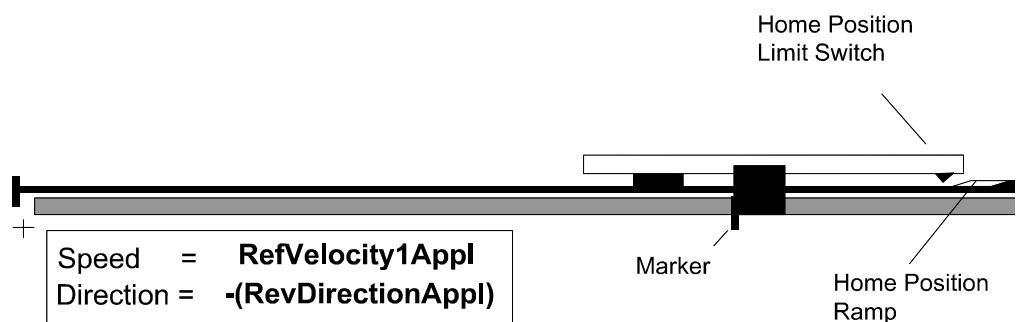
- **Axis travels to the home position**



Each axis travels to its home position limit switch.

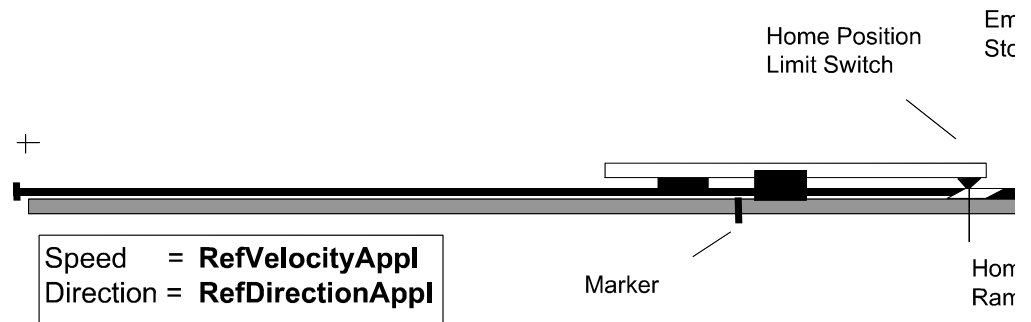
- **Leaving the home position limit switch**

The axis moves away from the limit switch



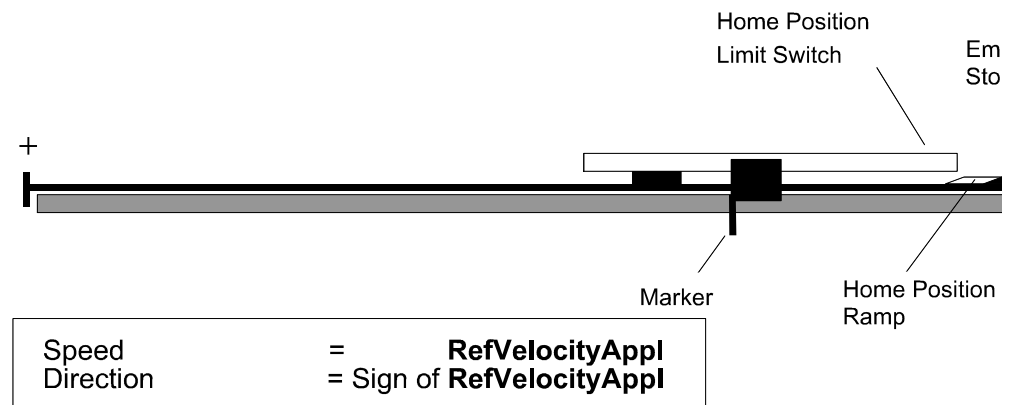
- **Re-approaching the home position limit switch**

The axis approaches the limit switch again.

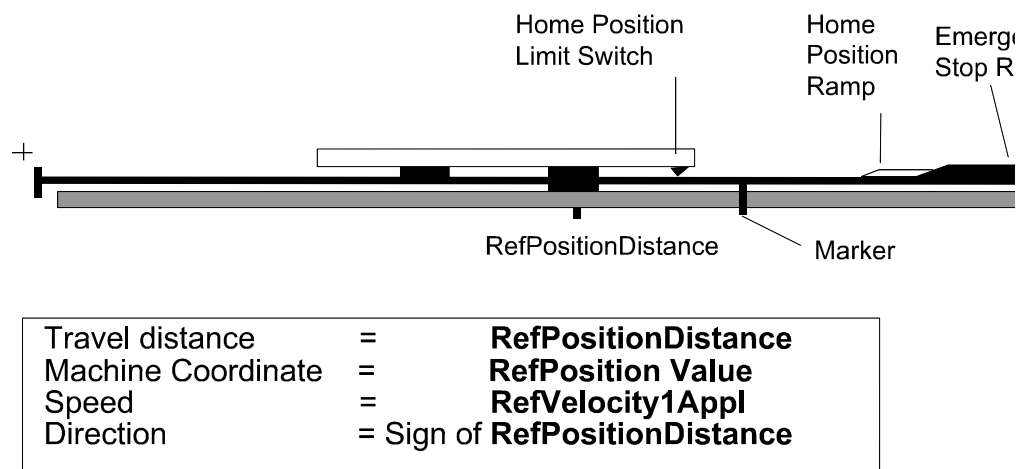


- **Searching for the marker signal**

The axis searches for the marker signal.



- **Travelling by RefPositionDistance, then setting RefPositionValue**



If **RefPositionDistance** = 0, then the axis will not move off of the marker location and the reference cycle will be complete.

When the reference cycle is complete, the axis location is set by **RefPositionValue**.

Note:

- For SDI drives Marker Signal depends on the measuring system type. With Resolver or SinCos measuring systems the system is absolute within one motor resolution or measuring system period. There the zero point of the measuring system period is used instead of marker signal.
With Encoder measuring system or others there may be a real marker, too. In that case and if you want to use the real marker set axis bit in machine parameter **RefUseNullMarker** to 1. Take care that the marker signal is connected to the measuring system input of SDI drive correctly!
- Home position limit switches must be normally closed limit switches. The homing position ramp must intersect with the axis limit emergency stop ramp.
- The CNC only identifies the marker signal from the measuring system (and thus the home position), if the signals A and B from the measuring system have a high level at the same time as the marker signal M (deviation < 100ns). If the measuring system does not comply, the bit 3 in the corresponding index of '**MaxRMSFrequency**' must be set.
- If during the step where the CNC is searching the marker signal this marker signal is not found, the axis travels at the velocity entered in the machine parameter **RefVelocityAppl**, up to the limit set by the machine parameter '**MarkerDistance**' from the **AxisControl** group.

2.5.3 Software limit switches

For axes with mandatory homing, it is possible to monitor the axis limits (travel distance). For safety reasons, it is recommended to set the respective positive and negative limits as soon as the start up of the homing cycle of an axis has been done.

The positive limit of the axis concerned is set in the machine parameter **SoftwareLimitPlus** and the negative in the machine parameter **SoftwareLimitMinus**.

Note:

- Software limits are not active until referencing of all mandatory axes is complete.
- The SoftwareLimitPlus must be larger in absolute terms than the corresponding SoftwareLimitMinus.

2.5.4 Mandatory homing - RefAxesAppl

The axes that have to move to the reference point after the CNC is powered up are determined through the machine parameter **RefAxesAppl**.

Each axis number corresponds to a Hex value (see table below). The sum of the Hex values of all axes that shall be set for mandatory homing after power-up must be written into **RefAxesAppl**.

Axis no.	8	7	6	5	4	3	2	1
Hex Value	80	40	20	10	8	4	2	1

Example:

No mandatory homing:		RefAxesAppl= 00
Mandatory homing only for:	1 st axis	RefAxesAppl = 01
	2 nd axis	RefAxesAppl = 02

3rd axis RefAxesAppl = 04

Example:

Mandatory homing is requested for 1st , 2nd , 4th , and 5th axis:

axis number	mandatory homing	HEX value
1	yes	1
2	yes	2
3	no	0
4	yes	8
5	yes	10
6	no	0
7	no	0
8	no	0
		sum 1B

RefAxesAppl = 001B

2.5.5 Axes homing sequence - AxisSequence

The sequence in which the axes must now move to the home position can be determined in the machine parameter **AxisSequence**. There are a maximum of eight cycles possible.

The cycles are defined through the values of the high and low bytes of the 4 indexes. Each axis number corresponds to a Hex value (see table below). The sum of the Hex values of the entire axis that shall reference in a certain cycle is the value that has to be written in the corresponding high or low byte of the address that reflects this cycle. A value FF in the low byte of **AxisSequence** (index 1) means that all axes reference at the same time.

If the CNC finds a cycle with content = 00, then the homing is terminated. All axes that are set up in the machine parameter **RefAxesAppl** as having mandatory homing have also to be defined in machine parameter **AxisSequence**.

Axis no.	8	7	6	5	4	3	2	1
Hex Value	80	40	20	10	8	4	2	1

Example:

The axis sequence should be first homing the 3rd axis alone, then in second step 1st and 2nd axes simultaneously and in a third step the 4th axis.

Reference Cycle	Address	Hex Value
1 st	3 rd axis	hex value = 04
2 nd	1 st and 2 nd axis	hex value = 01+ 02 = 03
3 rd	4 th axis	hex value = 08

AxisSequence (1) = 0304

AxisSequence (2) = 0008

2.5.6 Homing search direction - RefDirectionAppl

The direction in which the individual axes should search for the home position limit switch is determined in the machine parameter **RefDirectionAppl**.

If the value 0 is present in this address, all axes search for the limit switch in the negative direction (default, in so far all the limit switches are from NC type).

If the search has to be conducted in the positive direction, proceed as follows:

- Each axis number corresponds to a hex value (see table below).

- The sum of the hex values of all axes that shall be for the marker signal in negative direction, is the value of machine parameter **RefDirectionAppl.**

Axis no.	8	7	6	5	4	3	2	1
Hex Value	80	40	20	10	8	4	2	1

2.5.7 Homing first search velocity - **RefVelocity1Appl**

Unit: internal increments / ms

The velocity with which the axes should search for the home position limit switch

is entered for each axis in the corresponding index of the machine parameter **RefVelocity1Appl.**

Example:

The resolution is 0.001 mm, the desired homing velocity should be 6 m/min

$$\frac{6 \text{ m/min} * 1000 \text{ mm/m} * 1000 \text{ increments / mm}}{60 \text{ s/min} * 1000 \text{ ms/s}} = 100 \text{ internal increments/ms}$$

$$\text{RefVelocityAppl} = 100$$

2.5.8 Homing second search velocity - RefVelocityAppl

Unit: internal increments / ms

After receiving the signal from the home position limit switch, the CNC searches the marker signal of the measuring system.

The search velocity of the marker signal is entered for each axis in the corresponding index of the machine parameter **RefVelocityAppl**.

Note:

- The sign of **RefVelocityAppl** determines the search direction.

Example:

1 st axis and 2 nd axis	= 0.5 m/min	searches in positive direction
3 rd axis	= 0.5 m/min	searches in negative direction
4 th rotational axis	= 0.22 RPM	searches in positive direction

$$1^{\text{st}} \text{ and } 2^{\text{nd}} \text{ axis: } \frac{0.5 \text{ m/min} * 1000 \text{ mm/m} * 1000 \text{ internal inc/mm}}{60 \text{ s/min} * 1000 \text{ ms/s}} = 8.333 \text{ mm/ms}$$

$$3^{\text{rd}} \text{ axis } \frac{- 0.5 \text{ m/min} * 1000 \text{ mm/m} * 1000 \text{ internal inc/mm}}{60 \text{ s/min} * 1000 \text{ ms/s}} = - 8.333 \text{ mm/ms}$$

$$4^{\text{th}} \text{ axis } \frac{0.2 \text{ RPM} * 360 \text{ degrees/rev} * 1000 \text{ internal inc/degree}}{60 \text{ s/min} * 1000 \text{ ms/s}} = 1.2 \text{ degrees/ms}$$

2.5.9 Homing travel distance - **RefPositionDistance**

Unit: internal increments

A travel distance can be entered for each axis in the corresponding index of the machine parameter **RefPositionDistance**. The CNC travels this distance immediately after detection of the marker signal. The sign defines the direction.

2.5.10 Homing set position - **RefPositionValue**

Unit: internal increments

A set position that will be displayed after completion of the homing cycle can be entered for each axis in the corresponding index of the machine parameter **RefPositionValue**.

2.5.11 Distance between two marker pulses - **MarkerDistance**

Unit: axis increments

This machine parameter defines the maximum distance between two marker pulses of the encoder, counted in measuring system pulses. The value has to be entered at the corresponding index of the machine parameter '**MarkerDistance**' in the **AxisControl** group.

It is mainly used to determine the maximum search distance during marker search in the homing cycle.

The message "Marker error during home position cycle" appears :

- If the content of this parameter is wrong.
- If the measuring system is defective, that means the CNC could not find a marker pulse within this distance during the search.

The maximum search path emerges from the following formulas:

- Standard Measuring Systems:
Search path = $2 * \text{MarkerDistance} * \text{MachToInternalIncr}$
- Coded Distance Measuring Systems:
Search path = $4 * \text{MarkerDistance} * \text{MachToInternalIncr}$

Example

- linear axis

given: encoder ratio	1/2
given: lead screw pitch	20mm
given: MachToInternalIncr	1
MarkerDistance = $(20000 * \frac{1}{2}) / (2 * 1) = 5000$	
- rotary axis

given: encoder ratio	37/63
given: rotary load	360°
given: MachToInternalIncr	5.2857
MarkerDistance = $(360000 * 37/63) / (2 * 5.2857) = 20000$	

2.5.12 Maximum input frequency - MaxRMSFrequency

The maximum input frequency of the RMS-GATE-ARRAYS (RMS for Rotary Measuring System) has to be entered in the corresponding index of this machine parameter.

If the marker is not being found during a homing cycle, bit 3 can be set to resolve this situation.

Note:

- Before setting this bit, verify that the machine parameter '**MarkerDistance**' is correct.

The possible values are:

1000h	or	1008h	1000,0 kHz
800h	or	808h	500,0 kHz
400h	or	408h	250,0 kHz
200h	or	208h	62,5 kHz

Default: 1000h

2.5.13 Software limit switches – SoftwareLimitPlus and SoftwareLimitMinus

Unit: internal increments

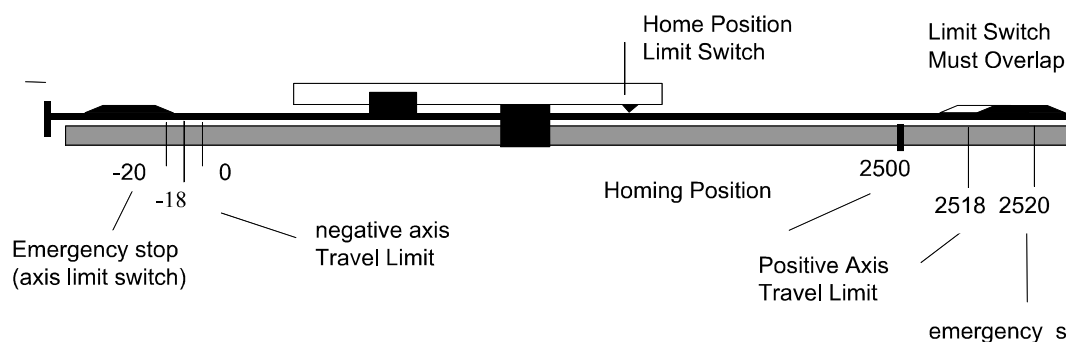
The values for the software travel limitation have to be entered in the corresponding index of respectively the machine parameters '**SoftwareLimitPlus**' and '**SoftwareLimitMinus**'. The **SoftwareLimitPlus** must be larger in absolute terms than the corresponding **SoftwareLimitMinus**.

Example:

- The home position has the position: 2500 mm
- The axis limit switches lie on the positions: +2520 mm and -20 mm
- given: The resolution is 0.001mm:

SoftwareLimitPlus=2518000

SoftwareLimitMinus=-18000



2.5.14 Special axes properties – RefCycleType

Special axes properties could be selected with the machine parameter **RefCycleType**. The value could be entered for each axis with corresponding index. The usual value for this parameter is zero (default).

Bit 1	= 0	This is not a stepper axis (default)
	= 1	This is a stepper axis
Bit 2	= 0	This is a gantry axis without an own homing limit switch (default)
	= 1	This is a gantry axis with an own homing limit switch
Bit 3	= 0	Marker signal is used for the axis (default)
	= 1	Marker signal is not used for the axis
Bit 4	= 0	Homing limit switch is used for the axis (default)
	= 1	Homing limit switch is not used for the axis

Note:

- Bits 2 - 4 of **RefCycleType** could be used for analog axes only.

Example:

- **RefCycleType = 1** An axis with stepper interface
- **RefCycleType = 2** A gantry axis with an own homing limit switch. Such gantry axis can moves independently from the leading axis (in small range – 80% of it's **GantryLagDiff** value) to find it's own home limit switch
- **RefCycleType = 4** Marker signal isn't used for the axis. Limit switch is only used for homing.
- **RefCycleType = 6** A gantry axis with an own homing limit switch, but without marker signal
- **RefCycleType = 8** An axis without homing limit switch, but with marker. This option can be useful for round axes with one marker signal per revolution.

2.6 Spindle Set-up

2.6.1 Spindle Output Channel – SpindleOutputAppl

Output via S axis Channel:

The S axis channels are defined in the drive configuration file. The CNC assigns the first spindle group to the first axis channel, the second spindle group to the second axis channel etc. The low byte defines which group the spindle belongs to.

High-Byte:	=0	
Low-Byte:	=0	group 1
	=1	group 2
	=2	group 3
	=3	group 4
	=4	group 5
	=5	group 6

Output via NC switch-over spindle axis channel:

For output via NC switch-over spindle axis the option switch-over Spindle/Rotational axis is necessary. (see the manual in options)

High-Byte:	=1F	
Low-Byte:	=0	output via 1 st axis channel of the set up axes
	=1	output via 2 nd axis channel of the set up axes
	=2	output via 3 rd axis channel of the set up axes
	=3	output via 4 th axis channel of the set up axes
	=4	output via 5 th axis channel of the set up axes
	=5	output via 6 th axis channel of the set up axes



Output via D/A-components:

High-Byte: = 2FH

Low-Byte: = Byte number of the D/A-component (see I/O configuration)

Note:

- Spindles with feedback must precede spindles without feedback.
- Spindles with output via axis channels must precede spindles with output via D/A-component.
- Spindles using the same output channel must be listed in series. There can be no gaps or spindles with another output channel located between spindles using the same output channel.

Example:

5 spindles in three groups:

- Main spindle via the first free S axis channel
- Spindle 2-4 via the second S axis channel
- spindle 5 via D/A-component, Byte No 4

SpindleOutputAppl(0)	= 0H	main spindle
SpindleOutputAppl(1)	= 01H	spindle 2
SpindleOutputAppl(2)	= 01H	spindle 3
SpindleOutputAppl(3)	= 01H	spindle 4
SpindleOutputAppl(4)	=2F04H	spindle 5

2.6.2 Spindle Feedback

SpindleFeedbackAppl:

SpindleFeedbackAppl specifies if the spindle has feedback. There is a bit for each spindle. If the first spindle has feedback, set bit zero to 1. If the second spindle has feedback, set bit 1 to 1.

It is not allowed to set up spindle 1 and 3 with feedback and spindle 2 without feedback.

SpindleIncrPerRev:

With SpindleIncrPerRev the resolution of each spindle will be set up. SpindleIncrPerRev indicates the number of the feedback pulses (after quadrupling) per revolution.

If the spindle is switchable to axis or if the parameter G33SpindleControlAppl is set, the machine parameters IncrementsPerRev, MachToInternalIncr and MachIncrementsPerRev for the axis channel corresponding to the spindle must be set up. See section in this manual for axis setup.

2.6.3 Spindle Speed – SpindleMaxSpeedAppl

Unit: RPM

Using SpindleMaxSpeedAppl, the maximum number of revolutions (in rpm) is set up for each spindle. A voltage of 10V is output when programming this number of revolutions.

2.6.4 Spindle Output Polarity - SpindleReversalAppl

The machine parameter SpindleReversalAppl determines, whether the spindle output is positive or negative in the case of clockwise rotating spindle (M03).

Bit 1	= 0	Spindle 1 output is positive in the case of M03
	= 1	Spindle 1 output is negative in the case of M03
Bit 2	= 0	Spindle 2 output is positive in the case of M03
	= 1	Spindle 2 output is negative in the case of M03
Bit n	= 0	Spindle n output is positive in the case of M03
	= 1	Spindle n output is negative in the case of M03

2.7 PLC program

2.7.1 General

In a machine tool the PLC is usually the 'master' of the equipment, which is responsible for the adjustment of the machine logic to the CNC and for the observance of certain functional sequences and operation sequence.

For this purpose the PLC uses interfaces to the machine and to the CNC. Via these interfaces the PLC can recognize the current state of the machine and can initiate the necessary action to the CNC and vice versa.

On the PA 8000 the PLC is already built-in to the controller, and communicates via an internal interface, the 'Signal-interface'.

This internal interface can be addressed from the PLC PROGRAM outwards and of course in the same manner, in which I/O boards are addressed for communication with the machine.

The 'conversation partner' of the PLC-program at this internal interface is the operating system of the CNC. The operating system ensures that information about current CNC states is provided and that action initiated by the PLC is executed.

The '**CNC-interface**' corresponds to a parallel, digital interface. Here, mainly single-signals are transferred with a direct state assignment or a function assignment. These single-signals ('bits') are grouped by 32 into 'DWORDs' and can be addressed by the PLC-program individually (as bits) or as a group (as a DWORD).

Typical applications are:

- Start and stop of a NC program processing;
- Closing and opening of the CNC control loops;
- Set the auxiliary functions programmed in NC programs;

This '**CNC-interface**' is described in the **PA-1131-PLC interfaces** manual. The PLC program '**Set_up.pro**' which is proposed as example in this manual, uses only the signals of the CNC interface which are at least necessary in every PLC program.

2.7.2 The PLC project '**Set_up.pro**'

The PLC project called '**Set_up.pro**' can be used for a simple first Start up of the axes.

This program sets all the necessary CNC interface variables for a machine. The PLC project '**Set_up.pro**' is written for a 5 axes machine.

Note:

- This is a test example program that does not take care of any SAFETY functions regarding the machine controlled by the CNC / PLC. If you run a machine with this PLC program, you will have to ensure that nobody stands inside the machine area, because unexpected machine movements could occur during the Start up.
- At least it should be ensured that all the E-stop buttons and all the axis E-stop limits are wired as hardware safety chain and that this chain is in function!
- This program sets all the interface signals to a 'TRUE' value. If wanted a 'FALSE' status of any of them can be set, using the 'Online Force Value' feature.



Only following inputs are effective:

- NC Start
- NC Stop
- Jog Plus
- Jog Minus

Note:

It is supposed that these signals are connected as described and that the necessary machine parameters are set as explained in this manual

2.7.2.1 Variable declaration

```
PROGRAM PLC_PRG
VAR
I_NC_Stop      AT %IX2.7 :    BOOL;
I_NC_Start     AT %IX2.8 :    BOOL;
END_VAR
```

2.7.2.2 PLC program

(* Signals Byte 1*)

```
INEMERGENCn    := TRUE ;  
IN_START        := I_NC_Start  
IN_STOPn        := NOT I_NC_Stop  
IN_TRANSF       := TRUE ;  
IN_ENABLE       := TRUE ;
```

(* Signals Byte 3 Drive on *)

```
IN_DRON01       := TRUE ;  
IN_DRON02       := TRUE ;  
IN_DRON03       := TRUE ;  
IN_DRON04       := TRUE ;  
IN_DRON05       := TRUE ;
```

(* Signals Byte 4 Axes enable *)

```
IN_DREN01       := TRUE;  
IN_DREN02       := TRUE;  
IN_DREN03       := TRUE;  
IN_DREN04       := TRUE;  
IN_DREN05       := TRUE;
```

2.7.3 CNC-PLC variables used by 'Set_up.pro'

2.7.3.1 Emergency stop - INEMERGENCn

This signal informs the CNC that the machine is in an EMERGENCY STOP state, resulting of the drop from the emergency stop safety-relay.

The CNC emergency stop status is self-locking in the CNC. That means, its effects are held until the input signal is TRUE again and confirmed by a CONTROL RESET.

value	meaning
FALSE	EMERGENCY STOP state active
TRUE	EMERGENCY STOP state not active

value	effect in the CNC
FALSE	All D/A outputs are directly set to 0V
	Any current CNC activities are interrupted and reset.
	The signal ON_NO_CNTR becomes TRUE.
	A display 'Emergency stop' appears in the headline of the basic CNC window.
	Actual machine axes positions are displayed instead of set positions.
	The E-stop self-locking cannot be released.
TRUE	No direct effect.
FALSE → TRUE	No direct effect.
TRUE → FALSE	No direct effect.

Note:

- Any drop of the emergency stop safety relay of the machine must be transmitted to the CNC through this signal, as otherwise:
 - Safety functions of the CNC are not activated;
 - The CNC could possibly output error messages, which could result in false interpretations.
- The axis measurement systems are not interrupted when emergency stop is active. Thus an axes homing cycle is not necessary after reset of the emergency stop status.
- The emergency stop status has no influence on the CNC Ready-relay

2.7.3.2 NC Start - IN_START

Any programmed action of the CNC can only be executed by this CNC interface signal, for example:

- Processing of a NC program;
- Homing cycle.

Note:

- Even if the NC Start push button is directly read-in from the CNC, it will have no direct effect. It will be output to the PLC, via output 'DWORD' 1 = ON_STARTNC, and has to be transmitted to the CNC via this signal IN_START.

value	meaning
FALSE	'Start' is not requested
TRUE	'Start' is requested

value	effect in the CNC
FALSE	No direct effect.
TRUE	No direct effect.
FALSE → TRUE	- The feed-hold self -locking of the CNC is released.(-> output DWORD 1 = ON_STOPNCn) - If the start requirements are fulfilled, the programmed CNC action is executed. - If the start requirements are fulfilled, the signal ON_CYCLEON becomes TRUE. - If the start requirements are fulfilled, a display 'Cycle on' appears in the headline of the basic CNC window.
TRUE → FALSE	No direct effect.

Note:

- Before the PLC program initiates a 'start', ensure, by suitable interlocks in the PLC logic, that:
 - All necessary safety loops are sharply controlled and do not report any errors.
 - It is safe to start the action on the machine side.
- The signal IN_START is not processed, if:
 - The CNC is in emergency stop status.
 - The program test mode 'without movement' is active.
 - The CNC is not either in Automatic mode or in homing cycle mode.
 - The signal IN_STOPn is FALSE.

Note:

- It is recommended:
 - To monitor the execution of 'start' via the signal ON_CYCLEON.
 - To realize the signal IN_START as a pulse either by resetting it through the TRUE status of ON_CYCLEON, or simply through a time triggered pulse.

'Start'-conditions in the CNC and effects of the "start" in the CNC:

Conditions	Effects of IN_START
The CNC is in operating mode MANUAL, incremental Jog was active and interrupted by IN_STOPn = FALSE	The interrupted movement of the axis is continued.
The CNC is in operating mode HOMING, not yet started; at least one axis has been selected with homing duty.	The homing cycle of the selected axes is started
The CNC is in operating mode HOMING; the homing cycle was active, and interrupted by IN_STOPn = FALSE	The interrupted homing cycle is continued.
The CNC is in operating mode AUTOMATIC-Sequential blocks, not yet started and an executable NC PROGRAM is selected.	The execution of the NC program is started.

Conditions	Effects of IN_START
The CNC is in operating mode AUTOMATIC-Single block; not yet started and an executable NC PROGRAM is selected.	The execution of the NC program is started; one block transfer is processed, but still no NC block is executed. The execution of the first NC block will only be done after two IN_START.
The CNC is in operating mode MDI; no NC block is active.	An NC block transfer is performed. The execution of the MDI NC block will only be done after two IN_START.
The CNC is in operating mode AUTOMATIC-Sequential blocks, or AUTOMATIC-Single block or MDI; a NC block was active and interrupted by IN_STOPn = FALSE	The execution of the NC block is continued. If the NC block was already terminated, then NC block transfers are performed according to the active CNC-operating mode.
The CNC is in operating mode AUTOMATIC-Sequential blocks, or AUTOMATIC-Single block or MDI; the processing of NC blocks was active and was stopped by the codes M0 or M1	The execution of the NC block is continued. If the NC block was already terminated, then NC block transfers are performed according to the active CNC-operating mode.

Note:

- The described effects of IN_START on NC blocks and their execution relate to pure NC-blocks and not to cycle blocks. If one or more cycle blocks precede a pure NC block then they are treated as belonging to the following NC block.

2.7.3.3 NC Stop - IN_STOPn

This CNC interface signal instructs the CNC for a feed hold. It is used mainly to interrupt actions which were activated with IN_START.

The CNC feed hold status is self-locking in the CNC. That means, its effects are held until the input signal is TRUE again and confirmed:

- by a IN_START if the test mode 'Without movement' is not active
- By the Softkey 'Test start' if the test mode 'Without movement' is active.
- by CONTROL RESET

value	meaning
FALSE	Feed hold requested
TRUE	Feed hold not requested

value	effect in the CNC
FALSE	• The signal IN_START is not processed.
	• The feed hold self-locking cannot be released.
TRUE	• No direct effect.
FALSE → TRUE	• No direct effect.
TRUE → FALSE	• Controlled stop of all axes with regard of the preset deceleration and the programmed path.
	• Feed hold self-locking status is entered.
	• As soon as all axes are on standstill, the signal ON_CYCLEON becomes FALSE.
	• As soon as all axes are on standstill, the signal ON_STAND becomes TRUE.

Note:

- The signal ON_CYCLEON remains TRUE by missing IN_STOPn.
- If thread cutting (G33, G34) is active it is not guaranteed, that the feed is hold as soon as IN_STOPn becomes FALSE:
 - The feed will be hold when the active (the last, if several are programmed in sequence) G33/34-block is terminated. This signal is therefore not suited for a 'stop in any case'.
 - The feed can only be stopped by dropping the signal IN_NULLV01...06 to FALSE; this sets the speed of the spindles to 0.
- If 'feed per revolution' (G95) is active, the axes deceleration is not controlled.
- The IN_STOPn signal is not relevant for axes output used as set value output for a spindle. Has the spindle also to be stopped by IN_STOPn, it has to be ensured that the PLC will then set the signals IN_NULLV01...06 to FALSE; this sets the speed of the spindles to 0.
- IN_STOPn is processed also when the test mode 'Without movement' is active.

2.7.3.4 Block transfer enable - IN_TRANSF

This signal allows the transfer of NC blocks from the preparation level to the execution level in the CNC and thus the execution of the NC blocks is stopped, once the execution of the actual active block is finished.

This signal is used mainly for short-term interruptions of the NC-program, if it is necessary to wait for the result of an external event (i.e. end of tool change).

value	meaning
FALSE	NC block transfer blocked.
TRUE	NC block transfer allowed.

value	effect in the CNC
FALSE	• NC block transfer is blocked.
	• Controlled stop of all axes with regard of the preset deceleration and the programmed path.
	• As soon as all axes are on standstill, the signal ON_STAND becomes TRUE.
TRUE	• No direct effect.
FALSE → TRUE	• If the CNC is in operating mode AUTOMATIC-Sequential blocks, or AUTOMATIC-Single block or MDI, and if the active NC block is terminated, than an NC block transfer is executed.
TRUE → FALSE	• No direct effect.

Note:

- In the operating mode MANUAL (including homing cycle) IN_TRANSF is not processed.
- The signal ON_CYCLEON remains TRUE by missing IN_TRANSF.
- The execution of cycle block is not interrupted by missing IN_TRANSF
- IN_TRANSF is processed also when the test mode 'without movement' is active.

Note:

- Under certain circumstances it is not guaranteed, that the NC block transfer is interrupted as soon as IN_TRANSF becomes FALSE:
If Look Ahead (G09) function is active and the path velocity is such high, that by the moment where IN_TRANSF becomes FALSE it is impossible to decelerate before the end of the active NC block; then further NC block transfers will be done as long it is necessary to decelerate the axes.
If thread cutting (G33, G34) is active and several G33/34-blocks are programmed in direct sequence, then further NC block transfer continue to be executed until the last G33/34-block is terminated.

2.7.3.5 Feed enable all axes - IN_ENABLE

This signal allows axes interpolation in the active NC block. This signal is used mainly for short-term interruptions of axes movements, if it is necessary to wait for the result of an external event (e. g. wait until the spindle has reached its nominal speed

value	meaning
FALSE	Interpolation blocked.
TRUE	Interpolation allowed.

value	effect in the CNC
FALSE	• Interpolation is blocked.
	• Controlled stop of all interpolating axes with regard of the preset deceleration and the programmed path.
	• As soon as all axes are on standstill, the signal ON_STAND becomes TRUE.

value	effect in the CNC
TRUE	No direct effect.
FALSE → TRUE	If the CNC is in operating mode AUTOMATIC-Sequential blocks, or AUTOMATIC-Single block or MDI, and if in the active NC block an interpolation is programmed, than this interpolation will be restarted.
TRUE → FALSE	No direct effect.

Note:

- The signal IN_ENABLE is not processed in the operating mode MANUAL (including homing cycle).
- The signal ON_CYCLEON remains TRUE by missing IN_ENABLE.
- If thread cutting (G33, G34) is active it is not guaranteed, that the interpolation is interrupted as soon as IN_ENABLE becomes FALSE:
 - The interpolation will be interrupted when the active (the last, if several are programmed in sequence) G33/34-block is terminated. This signal is therefore not suited for a 'stop in any case'.
 - The interpolation can only be stopped by dropping the signal IN_NULLV01...06 to FALSE; this sets the speed of the spindles to 0.
- If 'feed per revolution' (G95) is active, the axes deceleration is not controlled.
- IN_ENABLE is processed also when the test mode 'Without movement' is active.

2.7.3.6 Position loop enable - IN_DRIVEON

This DWORD groups the enable-signals for the position loops of all the CNC axes up to
A maximum of 32 axes.

Bit		
1	IN_DRON01	DRIVE ON 1st axis
2	IN_DRON02	DRIVE ON 2nd axis
3	IN_DRON03	DRIVE ON 3rd axis
4	IN_DRON04	DRIVE ON 4th axis
5	IN_DRON05	DRIVE ON 5th axis
6	IN_DRON06	DRIVE ON 6th axis
7	IN_DRON07	DRIVE ON 7th axis
...		...
32	IN_DRON32	DRIVE ON 32nd axis

Each axis position loop is independent of the other (except special cases like gantry axes). Therefore the bit signals allow treating separately each position loop. Following descriptions refer to each bit signal.

These signals are mainly used, if position loops have to be opened without EMERGENCY STOP (e. g. when axes have to be mechanically clamped, or when an axis is currently controlled by an external device)

value	meaning
FALSE	Position loop is open.
TRUE	Position loop is closed.

value	effect in the CNC
FALSE	The D/A output of the corresponding axis is set to 0V.
	A display 'Drive On missing' appears in the headline of the basic CNC window in the operation modes MANUAL and AUTOMATIC, as long as there is no emergency stop.
	The corresponding signals ON_POS_01...32 become TRUE, as long as the test mode 'Without movement' is not active.
	The machine position of the corresponding axes is displayed continuously instead of the set position.
TRUE	No direct effect.
FALSE → TRUE	The position loop of the concerned axis is closed and the CNC set-positions of all axes are synchronized with the last read-in machine positions; if the CNC is not in emergency stop status and the test mode 'Without movement' is not active.
	The geometric calculation for the actual NC program is synchronized with the last read-in machine position of all axes, if the CNC is not in emergency stop status.
TRUE → FALSE	No direct effect.

Note:

- A change from FALSE to TRUE of any IN_DRONxx during a program execution causes the loss of the NC blocks in the geometrical preparation buffer.
- Thus unexpected NC error messages, axes movements and switching operations on the machine will probably occur.
- This is the reason why in any automatic operation such a change may only appear, if no NC blocks have been yet processed.
- Otherwise such a change can only appear, if the selected M-code for closing the position loop is programmed in the active NC block. The number of this M-code is selected by the machine parameter '**DriveOnAppl**' in the **AxisControl** group.
- The signal ON_NO_CNTR (no feedback control) is not influenced by the IN_DRONxx signals.
- Even if the position loop of an axis is opened, this axis can be interpolated in the CNC-operating modes MANUAL and AUTOMATIC. But then the CNC will not proceed with the movements of the concerned axes.
- The IN_DRONxx signal is not relevant for an axis output used as set value output for a spindle.

2.7.3.7 Axis feed enable - IN_DRIVEEN

This DWORD groups the axis enable-signals used for non-interpolated movements for all the CNC axes up to a maximum of 32 axes.

Bit		
1	IN_DREN01	AXIS ENABLE 1st axis
2	IN_DREN02	AXIS ENABLE 2nd axis
3	IN_DREN03	AXIS ENABLE 3rd axis
4	IN_DREN04	AXIS ENABLE 4th axis
5	IN_DREN05	AXIS ENABLE 5th axis
6	IN_DREN06	AXIS ENABLE 6th axis
7	IN_DREN07	AXIS ENABLE 7th axis
...		...
32	IN_DREN32	AXIS ENABLE 32nd axis

In non-interpolated movements, each axis is independent of the other (except special cases like gantry axes). Therefore the bit signals allow treating separately each axis. Following descriptions refer to each bit signal.

These signals are mainly used, to interrupt movements when an axis reaches its hardware limit switches.

value	meaning
FALSE	Axis movement is blocked.
TRUE	Axis movement is enabled.

value	effect in the CNC
FALSE	Axis movement is blocked.
TRUE	No direct effect.
FALSE → TRUE	No direct effect.
TRUE → FALSE	The concerned axis is stopped with regard of its deceleration.

Note:

- In the automatic operating modes the IN_DRENxx signals only work on positioning axes, or if a programmed homing (G74) is active.
- The signal ON_CYCLEON is not influenced by the IN_DRENxx signals.
- The IN_DRENxx signals are processed also when the test mode 'Without movement' is active.
- The IN_DRENxx signals are not relevant for axes output used as set value output for a spindle.

Appendix 1 Machine Parameter Tool

Appendix 1.1 General

Machine Parameter affects a variety of conditions on the mechanic of the machine including items such as e.g. the soft limit switches, homing cycle and speed for the axes and spindle data, as well as specific definition data for different software features. You can think of Machine Parameter as containing parameters that affect and optimize the operation of the entire system CNC - machine tool.

The Machine Parameter Tool is used to edit the content of the different machine parameters and to save them as a PC file. However so far these changes are not valid in the CNC control until this file has been loaded into the CNC.

Note:

- Any changes in Machine Parameter values should only be done by specifically instructed people.
- Changing Machine Parameter values can have a negative impact on the machine tool causing incorrect operation, damage of the machine tool and potential personal injury.

To start the MP Tool by doing the following:

- Start the HMI
- Set the HMI to **SETUP** mode
- Enter the password
- Press ALT-t: Machine setup
- Press ALT-1: machine Parameter
- Press ALT-1: MPTool

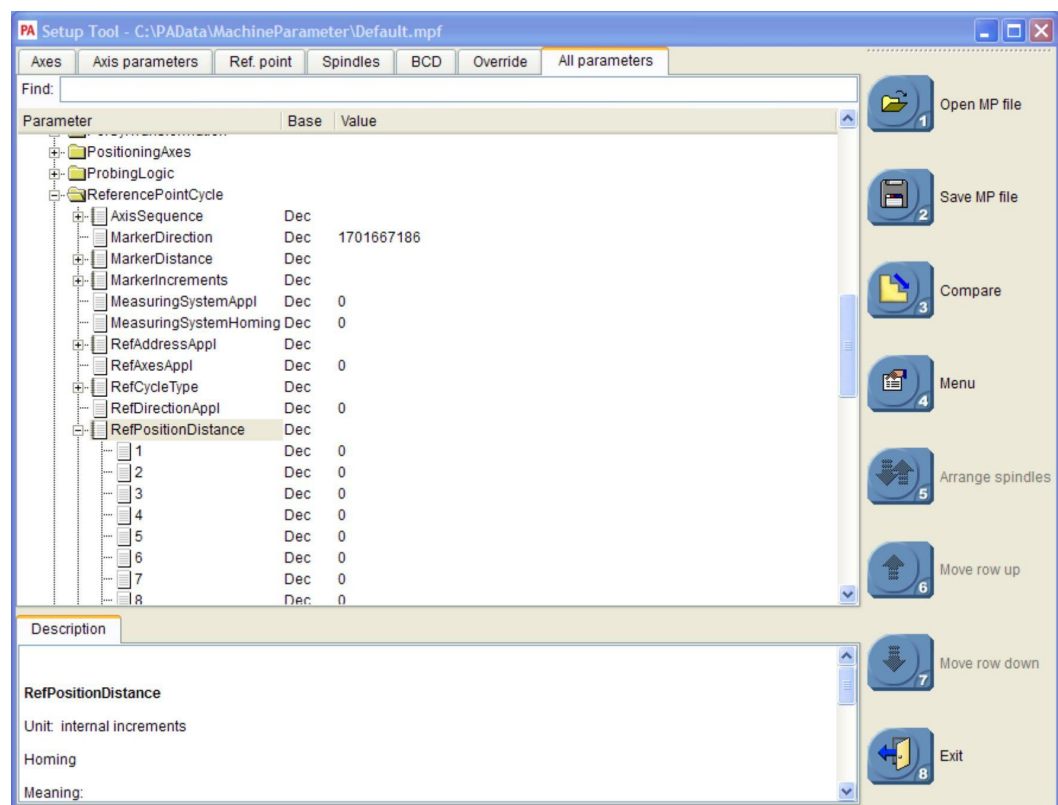
The following screen will appear:



Appendix 1.2 Edit and save

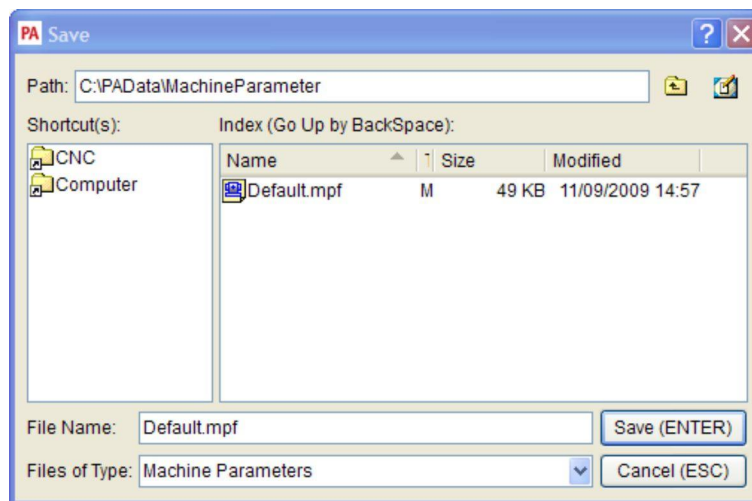
To change an Machine Parameter item, do the following:

- Saving the current state as file onto disk to be on the safe side.
- Select the MP group where the item to be changed is located. To display the content of the group select with the mouse pointer the + box left of the group name and click the left mouse button
- Select the MP item to be changed using the mouse.



- All values may be edited directly. Press the right mouse button to get a context menu for special functions like switching value display from DEC to HEX or BIN.

- All MP changes have to be saved as file onto disk. Once the edit window has been closed, selecting Softkey ALT-2 will open following window.



Key in a different file name if desired. The file will be saved by pressing the **SAVE** button.

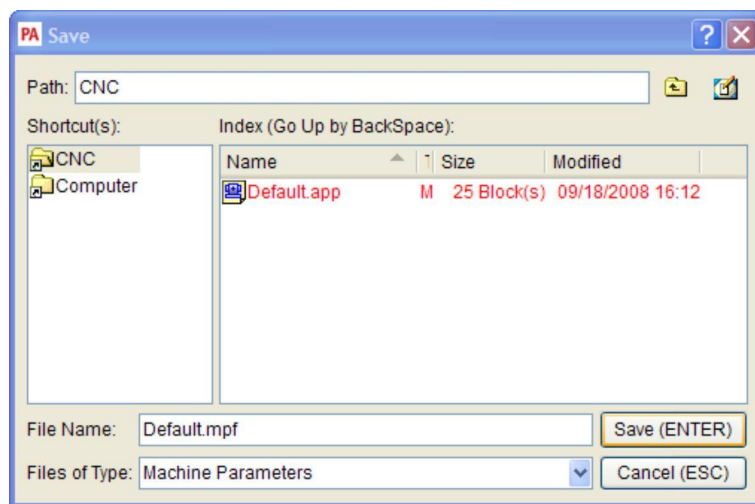
Appendix 1.3 Load the machine parameters into the CNC

Data which have been modified using the MP tool need to be loaded into the memory of the CNC for the CNC to be able to act on the data items.

To load the machine parameters into the CNC, do the following:

The machine parameter file can be saved directly in the CNC.

Select Softkey ALT-2 'Save MP File'. Following window will be opened:



- Select 'CNC' in the left directory window.
- Select the file to be loaded with the mouse pointer or by keying in its file name.
- Press the Emergency Stop button. Emergency stop must be engaged to load machine parameters values.
- Load the file by pressing the 'Save' button. A message should be displayed by the CNC indicating the transfer has been successful.

Appendix 1.4 Output the actual machine parameters

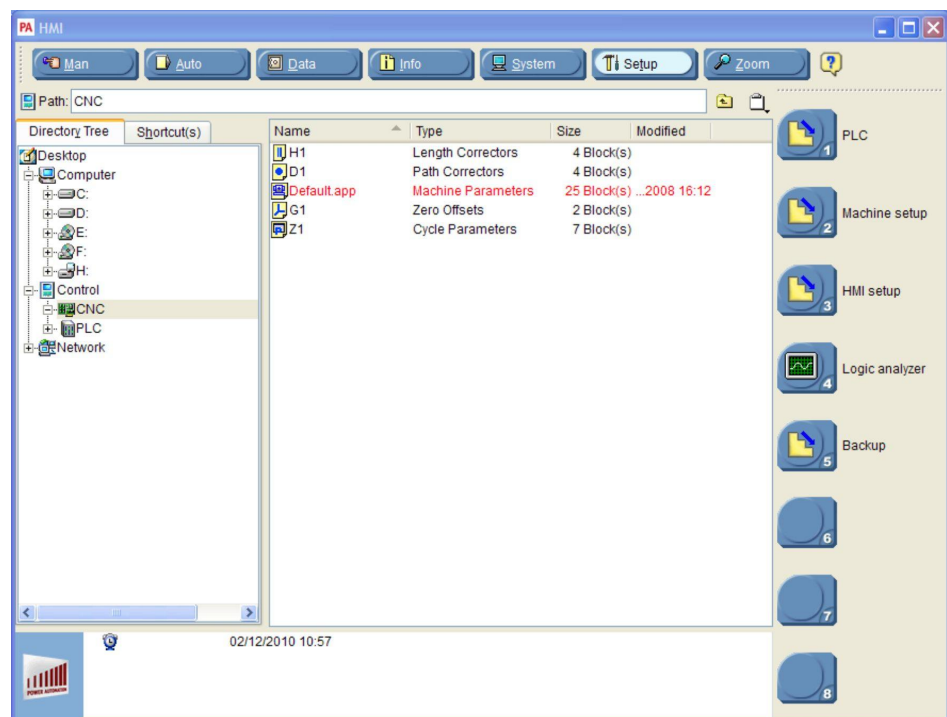
In some cases it is necessary to back up the actual machine parameter values. The output function allows saving the values out of the CNC memory.

This is necessary:

- Before reinstalling the PA CNC software, to make sure that the actual data are not being lost.
- After any procedures where the CNC generates own machine parameter values, e. g. ART optimize procedure.

To output the machine parameters from the CNC, do the following:

Under **SETUP** select 'Control' -> 'CNC' in the directory tree. Following window will be opened:



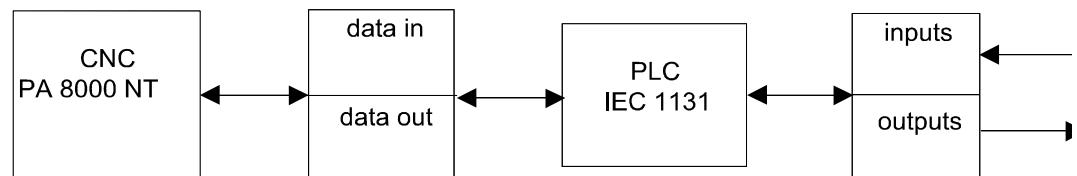
- Select the machine parameter file you wish to output (default.app)
- Press the right mouse button and copy it with the context menu
- Now you can choose any other directory in the left window and paste the machine parameter file with the same context menu.

Appendix 2 Programmable Logic Controller tool

Appendix 2.1 General

The PLC integrated in the PA 8000 CNC control allows the communication between the CNC control and the machine tool. This communication runs in two ways:

- Hardware connection to inputs and outputs modules between the PAMIO PLC modules and the machine tool.
- Software data exchange between the CNC and the PLC



Note:

- The purpose of this chapter is to give a brief, but adequate information, how to use the PLC tool for loading and debugging a PLC program (or 'project'), using the Online features of this tool.
- Any further information about the PLC language and the PLC development system IEC 1131 is described in the **PLC-programming with PA-1131-DS**.

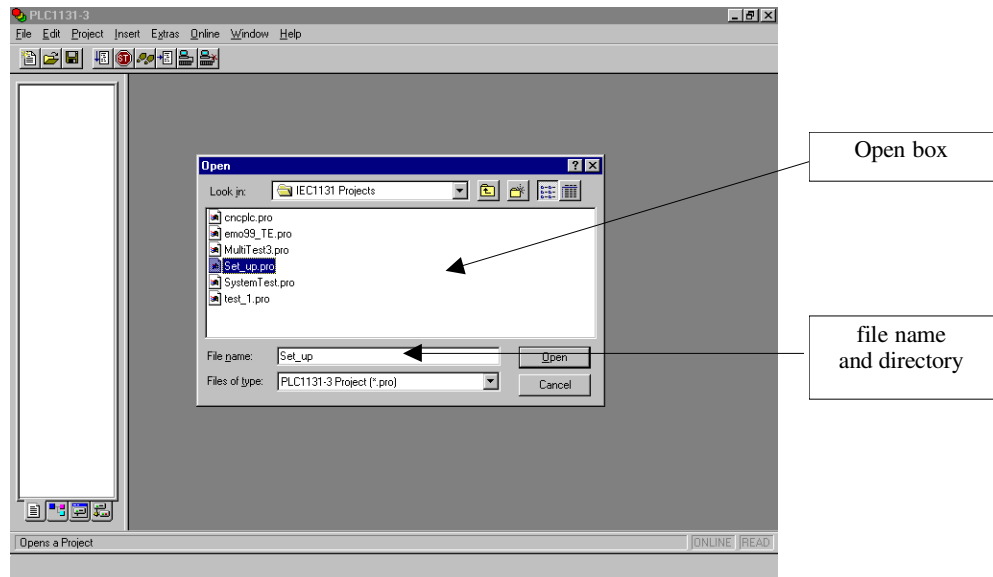
Start the PLC Tool by doing the following:

- Start the HMI
- Set the HMI to **SETUP** mode
- Enter the password (if required).
- Press F1: PLC
- Press F1: PLC programming

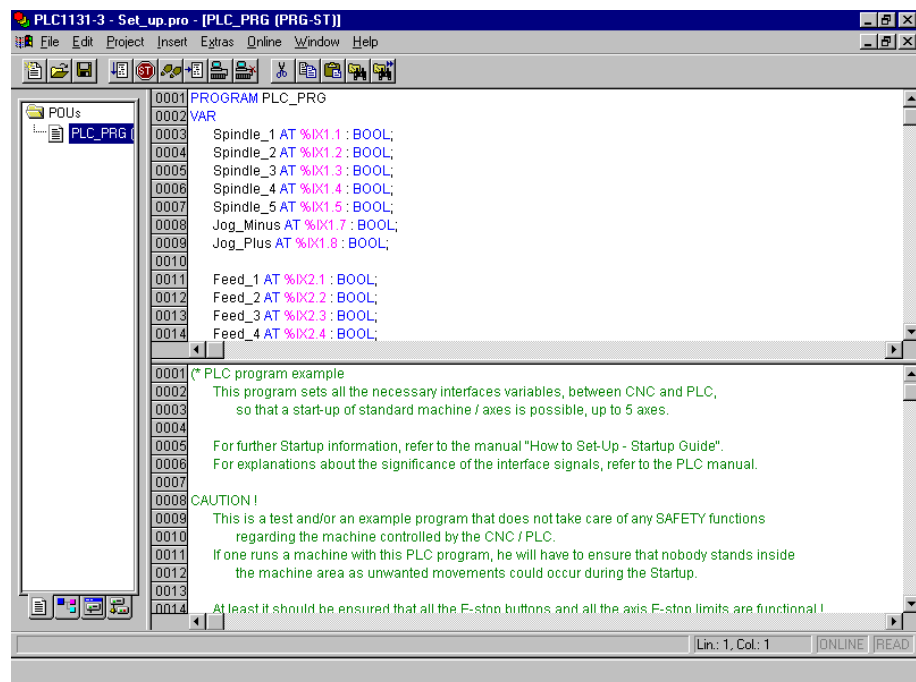
The PLC development system screen will appear.

Appendix 2.2 Load project

To load a different than the existing, or a new PLC program, open the '**F**ile' menu and the '**O**pen' box, select it with the mouse pointer or key-in its name and valid with the **O**pen button



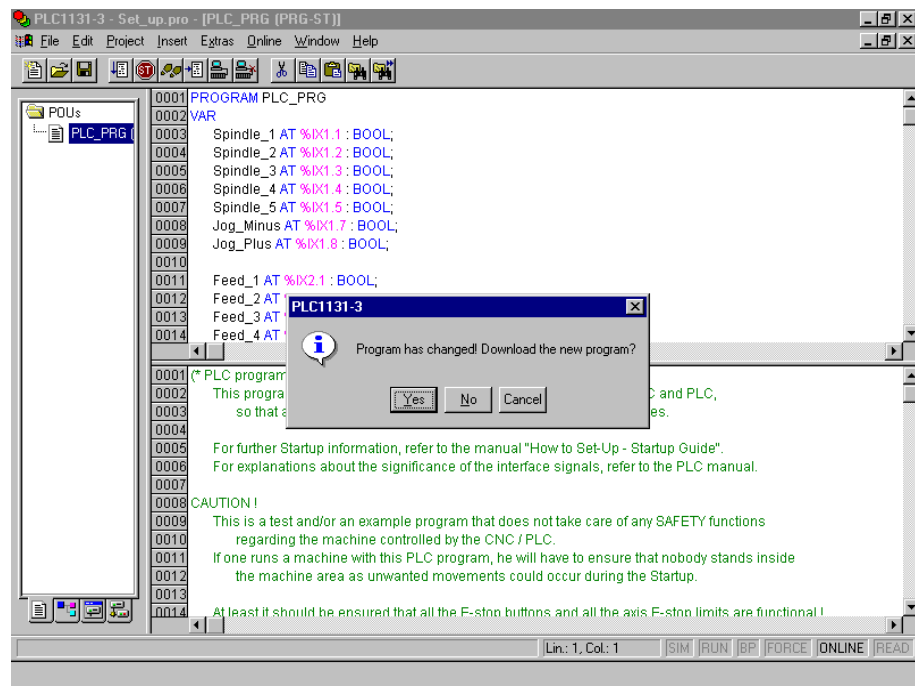
After successful loading the following window will appear:



Appendix 2.3 Download project

The new PLC program is now loaded in the PLC development system, but not yet down-loaded to the CNC. Therefore it is necessary to select under the '**Online**' menu the '**Login**' function. The project will now be rebuild. That means that all syntax and logical connection checks will be done.

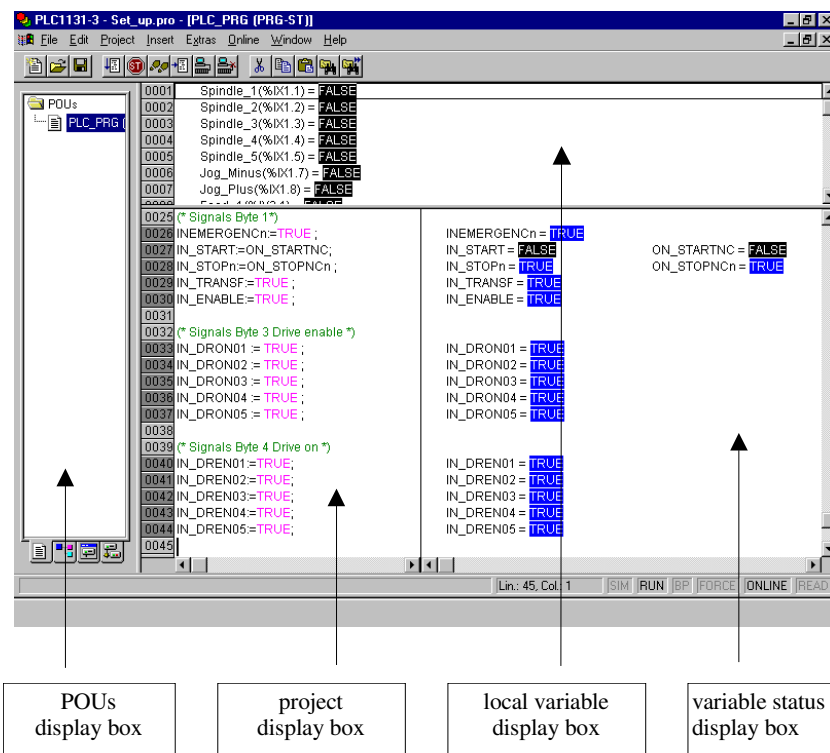
Is the project correct, the software will ask to download it; if not, it is necessary first to correct the faults in the current project.



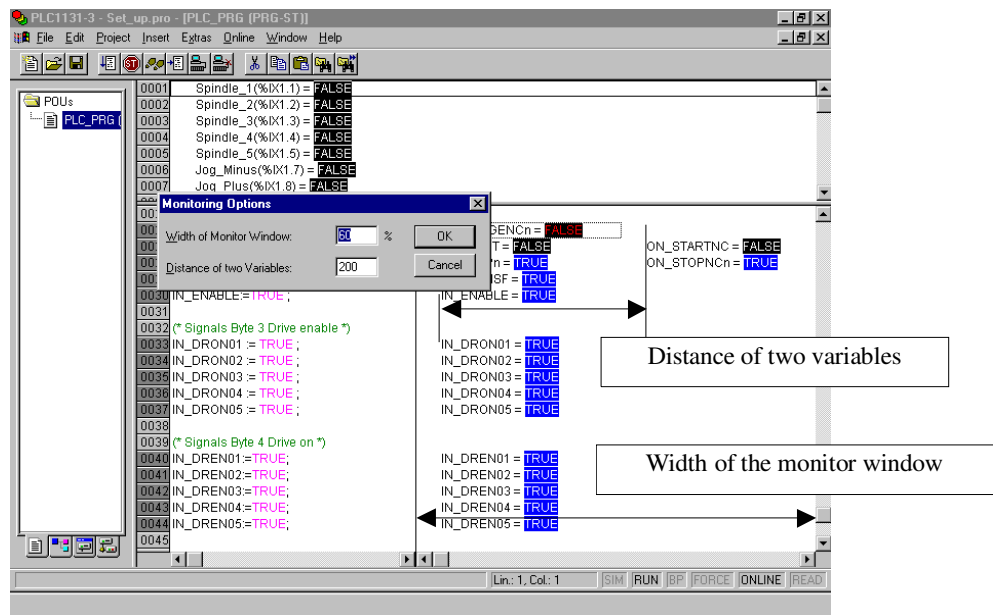
Appendix 2.4 Online mode

The Online screen is roughly split in 4 boxes:

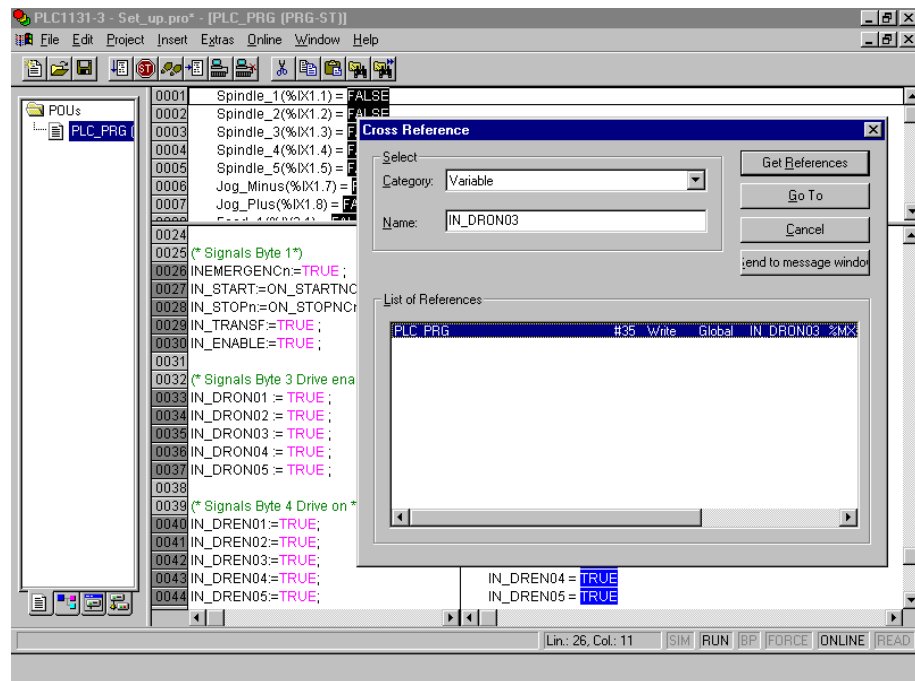
- POUs display box: Here are listed the different POU's of the actual project. To switch the display from one POU to another, select it with the mouse pointer and double click.
- local variable display box: Here are listed the entire variable included in the variable declaration of the actual POU with Online status display.
- project display box: Here is displayed the program of the actual POU.
- variable status display box: Here is shown the status of the variables called in the corresponding program line of the actual POU.



The relative size of the 'project display box' and the 'variable display box' is adjustable. This can be done by using the '**Monitoring Options**' in the '**Extras**' menu. The 'Monitoring Option' dialog allows setting the '**Width of the Monitor Window**' in percentage of the total width of the two boxes, as well as the '**Distance of two Variables**'.



It is possible to search all location in all the POU's of the project where a certain variable is called. This can be done by using the **'Show Cross Reference'** in the **'Project'** menu.

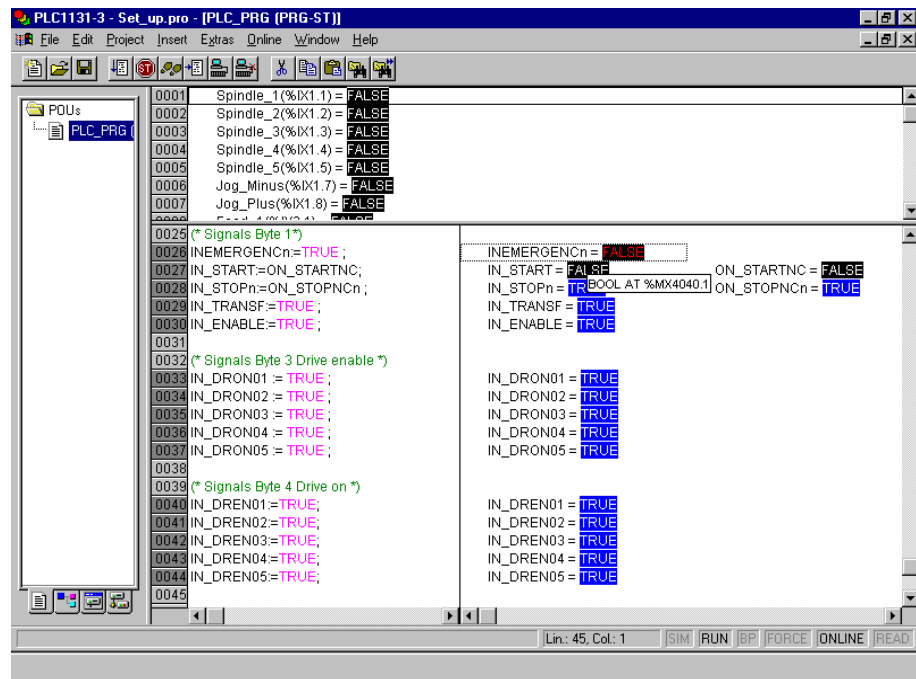


Appendix 2.5 Write and Force variables

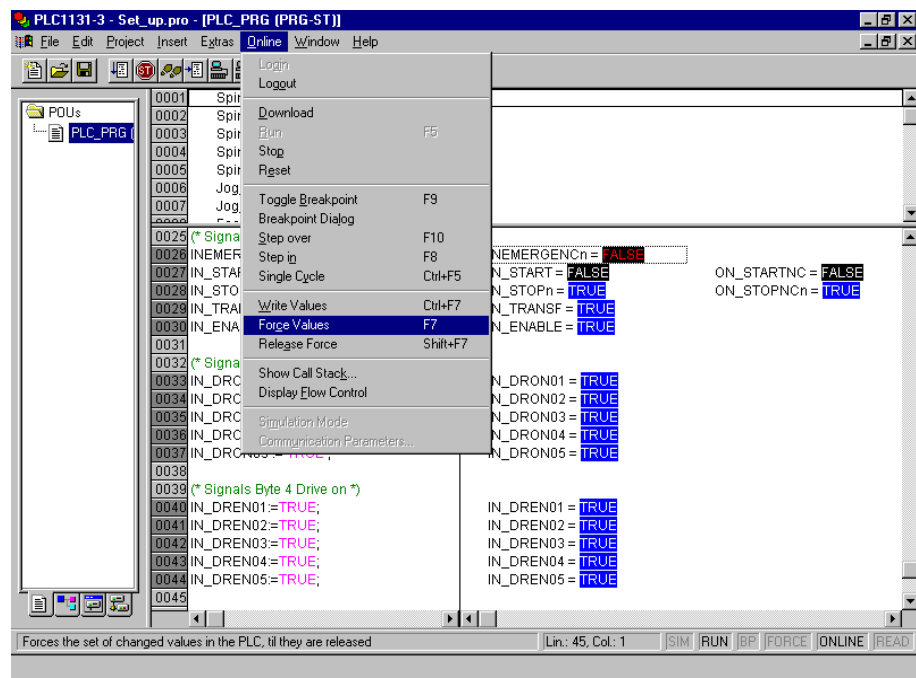
For test purposes it is possible to change the status of any local or global variable to the desired value, regardless the status to which the program would set this variable.

For changing the value of a variable, perform a double mouse click on the variable either in the local variable display box or in the variable status display box. Is the variable from:

- BOOL operand: double click will alternate TRUE and FALSE
- other operands: Double click will open a dialog for changing the value.



After that the value is just marked for changing, but not actually changed in the controller. It is possible to change the value of several variables at once. The changing itself is performed by a **'Write values'** or a **'Force Values'** in the **Online** menu, or respectively by a **'Ctrl+F7'** or a **'F7'**.





With the command “**Write values**”, the new values of all marked variables are just written once to the PLC (and monitored again) and can immediately be changed by the program.

With the command ‘**Force values**’, the new values of all marked variables are written after each controller cycle to the PLC, until ‘**Release Force**’ or ‘**Shift+F7**’ is executed. As long as the values are forced, they are displayed in red colour.

For other Online functionality, refer to the **PA - 1131-DS User’s Guide**

Appendix 3 Customized push buttons

Appendix 3.1 Example: Machine Control Panel

Following an example of a customized machine control panel is being described. The requirements are:

- Push button selection of 3 manual modes:
 - Continuous Jog
 - Incremental Jog
 - Handwheel (MPG)

These functions have to be latched

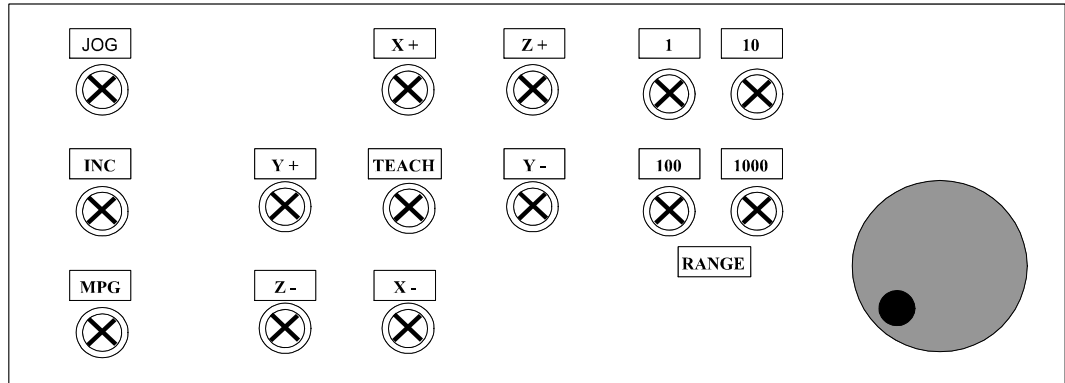
- Push button selection of 4 ranges:
 - Continuous Jog speed1, 2, 3, 4
 - Incremental Jog increment1, 2, 3, 4
 - Handwheel (MPG) factor1, 2, 3, 4

these functions have to be latched

Push button selection of axes (3) and direction (+, -):

- Continuous Jog movement as long as the push button is being pressed
- Incremental Jog1 increment / per push
- Handwheel (MPG) selection of the axis to be moved by MPG

Proposed layout of the additional machine control panel:



Appendix 3.2 Corresponding PLC program

The purpose of this chapter is to explain which tools of the CNC / PLC communication have to be used in order to realize the example.

Note:

This chapter will not explain how to edit the PLC program. For a brief information about how to use the Appendix 3.

For any information about the PLC language and the PLC development system IEC 1131 refer to the **PLC-programming with PA-1131-DS**.

To fulfil the requirements of this example the POU program (Customized_MCP) uses mainly the 'External mode selection' of the PLC CNC interface.

This chapter will not explain in details how the 'External mode selection' works. For any information about this function refer to the manual **PA-1131-PLC interfaces**.

The '**External mode selection**' function allows the selection of different operating modes of the CNC by setting the appropriate bits in the CNC interface DWORD **IN_EXTMODE**.

The PLC monitors appropriate the correct execution of the command by checking CNC output bits if they exist (e.g. ON_JOGMODE, ON_AUTO,)

DWORD ONEXTMODE which assumes, after successful execution the same value as requested **IN_EXTMODE**.

If the operator selects continuous jog (push button 'JOG') or incremental jog (push button 'INC'), a preparatory step has to be execute consisting of following commands selection:

- continuous jog IN_EXTMODE =70
- incremental jog IN_EXTMODE =102

As soon as this first step has been acknowledged (ONEXTMODE = IN_EXTMODE) the next PLC selection can be made according to the table below.

IN_EXTMODE

bit 8	bit 7	bit 6	Bit 5	bit 4	bit 3	bit 2	bit 1	
							1	Manual mode
					0	0		Hold movements
					0	1		Move plus
					1	0		Move minus
					1	1		Handwheel
			0	0				Range 1
			0	1				Range 2
			1	0				Range 3
			1	1				Range 4
0	0	0						Axis 1
0	0	1						Axis 2
0	1	0						Axis 3
0	1	1						

Example:

Move axis Y in JOG + at range 2

IN_EXTMODE	8	7	6	5	4	3	2	1
binary value	0	0	1	0	1	0	1	1

IN_EXTMODE = 43 (decimal)

Move axis Z in handwheel at range 4

IN_EXTMODE	8	7	6	5	4	3	2	1
binary value	0	1	0	1	1	1	1	1

IN_EXTMODE = 95 (decimal)

Appendix 3.3 Corresponding machine parameter

Appendix 3.3.1 Handwheel factors - ExtModeHandwheelFeed

The machine parameter **ExtModeHandwheelFeed** consists of 4 different indexes, and is located in the group '**PLCFunctions**'. Each index corresponds to the multiplying factor that will apply to the handwheel pulses input, according to the selected range 1 to 4.

Example:

In the given example, the handwheel factors should be 1, 10, 100, 1000

ExtModeHandwheelFeed

Index 1 = 1

Index 2 = 10

Index 3 = 100

Index 4 = 1000

Appendix 3.3.2 Increment values - ExtModeJogInkr

The machine parameter **ExtModeJogInkr** consists of 4 different indexes, and is located in the group '**PLCFunctions**'. Each index corresponds to the value of the increment that will be executed, according to the selected range 1 to 4.

Example:

In the given example, the increment values should be 1, 10, 100, 1000

ExtModeJogInkr

Index 1 = 1

Index 2 = 10

Index 3 = 100

Index 4 = 1000

Appendix 3.3.3 Continuous jog speeds - ExtModeManFeed

The machine parameter **ExtModeManFeed** consists of 4 different indexes, and is located in the group '**PLCFunctions**'. Each index corresponds to the percentage that will apply to the manual speed (**SAxisFeedAppl** from the **AxisControl** group) to determine the jog speed, according to the selected range 1 to 4.

Example:

In the given example, the continuous jog speed should be 1%, 10%, 50%, 100% of 4 m / min

ExtModeManFeed

Index 1 = 100
Index 2 = 1000
Index 3 = 5000
Index 4 = 10000

Appendix 3.4 PLC program for customized push buttons

PROGRAM Customized_MCP

VAR

O_Lamp_jog	AT %QX9.1 : BOOL;
O_Lamp_mpg	AT %QX9.2 : BOOL;
O_Lamp_inc	AT %QX9.3 : BOOL;
O_Lamp_Teach	AT %QX9.4 : BOOL;
O_Lamp_X_plus	AT %QX9.5 : BOOL;
O_Lamp_reset	AT %QX9.7 : BOOL;
O_Lamp_X_minus	AT %QX9.8 : BOOL;
O_Lamp_range_1	AT %QX10.1 : BOOL;
O_Lamp_range_2	AT %QX10.2 : BOOL;
O_Lamp_range_3	AT %QX10.3 : BOOL;
O_Lamp_range_4	AT %QX10.4 : BOOL;
O_Lamp_Y_plus	AT %QX10.5 : BOOL;
O_Lamp_Y_minus	AT %QX10.6 : BOOL;
O_Lamp_Z_plus	AT %QX10.7 : BOOL;
O_Lamp_Z_minus	AT %QX10.8 : BOOL;
I_X_plus	AT %IX11.1 : BOOL;
I_X_minus	AT %IX11.2 : BOOL;
I_Y_plus	AT %IX11.3 : BOOL;
I_Y_minus	AT %IX11.4 : BOOL;
I_Z_plus	AT %IX11.5 : BOOL;
I_Z_minus	AT %IX11.6 : BOOL;
I_reset	AT %IX11.7 : BOOL;
I_jog	AT %IX11.8 : BOOL;
I_mpg	AT %IX12.1 : BOOL;
I_inc	AT %IX12.2 : BOOL;
I_range_1	AT %IX12.3 : BOOL;
I_range_2	AT %IX12.4 : BOOL;
I_range_3	AT %IX12.5 : BOOL;
I_range_4	AT %IX12.6 : BOOL;

```

I_teach          AT %IX12.7 : BOOL;
Time_1:          TP;          (* Teach In Pulse *)
Time_2:          TP;          (* Reset pulse *)
Ext_manual:      INT;          (* Flag for external mode selection *)
Ext_direction:   INT;          (* External direction selection *)
Ext_range:       INT;          (* External range selection *)
Ext_axis:        INT;          (* External axis selection *)
Ext_reset_off:   F_TRIG;       (* External reset change *)
Change_mpg:      R_TRIG;       (* Rising edge of MPG selection *)

END_VAR

(*=====*)
(* Machine Control Panel *)
(*=====*)
(* Select Manual Mode *)
    IF NOT ON_CYCLEON
    AND I_jog
    THEN Ext_manual:= 1;
    END_IF;

    IF NOT ON_CYCLEON
    AND I_inc
    THEN Ext_manual:= 2;
    END_IF;

    IF NOT ON_CYCLEON
    AND I_mpg
    THEN Ext_manual:= 3;
    END_IF;

    Change_mpg(CLK:= I_mpg);

    IF I_reset = 1
    THEN Ext_manual:= 0;
    END_IF;

```

```
CASE Ext_manual OF
  1: IN_EXTMODE := 70;
    Ext_manual := 11;

  11: IF ONEXTMODE = IN_EXTMODE
    THEN Ext_manual:= 21;
    IN_EXTMODE:= 0;
    END_IF;

  2: IN_EXTMODE:= 102;
    Ext_manual:= 12;

  12: IF ONEXTMODE = IN_EXTMODE
    THEN Ext_manual:= 22;
    IN_EXTMODE:= 0;
    END_IF;
END_CASE;

(* External direction *)
  IF I_X_plus
  OR I_Y_plus
  OR I_Z_plus
  THEN Ext_direction:= 2;
  END_IF;

  IF I_X_minus
  OR I_Y_minus
  OR I_Z_minus
  THEN Ext_direction:= 4;
  END_IF;

  IF Ext_manual = 3
  THEN Ext_direction:= 6;
  END_IF;
```

```
IF I_X_plus = FALSE
AND I_Y_plus = FALSE
AND I_Z_plus = FALSE
AND I_X_minus = FALSE
AND I_Y_minus = FALSE
AND I_Z_minus = FALSE
AND Ext_manual <> 3
THEN Ext_direction:= 0;
END_IF;
```

(* External range *)

```
IF( I_range_1 OR Time_2.Q)
AND NOT I_range_2
AND NOT I_range_3
AND NOT I_range_4
THEN Ext_range:= 0;
END_IF;
```

```
IF I_range_2
AND NOT I_range_1
AND NOT I_range_3
AND NOT I_range_4
THEN Ext_range:= 8;
END_IF;
```

```
IF I_range_3
AND NOT I_range_1
AND NOT I_range_2
AND NOT I_range_4
THEN Ext_range:= 16;
END_IF;
```

```
IF I_range_4
AND NOT I_range_1
AND NOT I_range_2
AND NOT I_range_3
THEN Ext_range:= 24;
END_IF;
```

(* External axis *)

```
IF I_X_plus
OR I_X_minus
OR I_reset
OR Change_mpg.Q = TRUE
THEN Ext_axis:= 0;
END_IF;
```

```
IF I_Y_plus
OR I_Y_minus
THEN Ext_axis:= 32;
END_IF;
```

```
IF I_Z_plus
OR I_Z_minus
THEN Ext_axis:= 64;
END_IF;
```

(* External movement *)

```
IF( Ext_manual = 21 OR Ext_manual = 22 OR Ext_manual = 3 )
AND Ext_direction > 0
AND Ext_manual > 0
THEN IN_EXTMODE:= Ext_axis + Ext_range + Ext_direction + 1;
END_IF;
```

```
IF Ext_direction = 0
AND Ext_manual <> 0
AND Ext_manual <> 1
AND Ext_manual <> 11
AND Ext_manual <> 2
AND Ext_manual <> 12
THEN IN_EXTMODE:= 1;
END_IF;
```

(* External reset *)

```
Time_2(IN:= I_reset, PT :=T#0.2s);
Ext_reset_off(CLK:= I_reset);
```

```
IF Time_2.Q
THEN IN_EXTMODE:= 6;
END_IF;
```

```
IF Ext_reset_off.Q
THEN IN_EXTMODE:= 0;
END_IF;
```

(* Teach in lamp *)

```
Time_1(IN:= I_teach, PT :=T#0.2s);
O_Lamp_Teach := Time_1.Q;
```

(* Jog lamp *)

```
IF Ext_manual = 21
THEN O_Lamp_jog:= 1;
ELSE O_Lamp_jog:= 0;
END_IF;
```

(* Incr lamp *)

```
IF Ext_manual = 22
THEN O_Lamp_inc:= 1;
ELSE O_Lamp_inc:= 0;
END_IF;
```

(* Mpg lamp *)

```
IF Ext_manual = 3
THEN O_Lamp_mpg:= 1;
ELSE O_Lamp_mpg:= 0;
END_IF;
```

(* Range 1 lamp *)

```
IF Ext_range = 0
AND Ext_manual > 0
THEN O_Lamp_range_1:= 1;
ELSE O_Lamp_range_1:= 0;
END_IF;
```

(* Range 2 lamp *)

```
IF Ext_range = 8
AND Ext_manual > 0
THEN O_Lamp_range_2:= 1;
ELSE O_Lamp_range_2:= 0;
END_IF;
```

(* Range 3 lamp *)

```
IF Ext_range = 16
AND Ext_manual > 0
THEN O_Lamp_range_3:= 1;
ELSE O_Lamp_range_3:= 0;
END_IF;
```

(* Range 4 lamp *)

```
IF Ext_range = 24
AND Ext_manual > 0
THEN O_Lamp_range_4:= 1;
ELSE O_Lamp_range_4:= 0;
END_IF;
```

(* Axes selection lamps in Handwheel mode *)

```
IF(Ext_manual = 3 AND(I_X_plus = TRUE OR I_X_minus = TRUE))
    OR Change_mpg.Q = TRUE
THEN O_Lamp_X_plus:= TRUE;
    O_Lamp_X_minus:= TRUE;
    O_Lamp_Y_plus:= FALSE;
    O_Lamp_Y_minus:= FALSE;
    O_Lamp_Z_plus:= FALSE;
    O_Lamp_Z_minus := FALSE;
END_IF;
```

```
IF(Ext_manual = 3 AND(I_Y_plus = TRUE OR I_Y_minus = TRUE) )
THEN O_Lamp_Y_plus:= TRUE;
    O_Lamp_Y_minus:= TRUE;
    O_Lamp_X_plus:= FALSE;
    O_Lamp_X_minus:= FALSE;
    O_Lamp_Z_plus:= FALSE;
    O_Lamp_Z_minus:= FALSE;
END_IF;
```

```
IF(Ext_manual = 3 AND(I_Z_plus = TRUE OR I_Z_minus = TRUE) )
THEN O_Lamp_Z_plus:= TRUE;
    O_Lamp_Z_minus:= TRUE;
    O_Lamp_X_plus:= FALSE;
    O_Lamp_X_minus:= FALSE;
    O_Lamp_Y_plus:= FALSE;
    O_Lamp_Y_minus:= FALSE;
END_IF;
```

(* Axes selection lamps in Jog mode *)

```
IF Ext_manual = 21
AND I_X_plus = TRUE
THEN O_Lamp_X_plus:= TRUE;
END_IF;
```

```
IF Ext_manual = 21
AND I_X_plus = FALSE
THEN O_Lamp_X_plus:= FALSE;
END_IF;
```

```
IF Ext_manual = 21
AND I_X_minus = TRUE
THEN O_Lamp_X_minus:= TRUE;
END_IF;
```

```
IF Ext_manual = 21
AND I_X_minus = FALSE
THEN O_Lamp_X_minus:= FALSE;
END_IF;
```

```
IF Ext_manual = 21
AND I_Y_plus = TRUE
THEN O_Lamp_Y_plus:= TRUE;
END_IF;
```

```
IF Ext_manual = 21
AND I_Y_plus = FALSE
THEN O_Lamp_Y_plus:= FALSE;
END_IF;
```

```
IF Ext_manual = 21
AND I_Y_minus = TRUE
THEN O_Lamp_Y_minus:= TRUE;
END_IF;
```

```
IF Ext_manual = 21
AND I_Y_minus = FALSE
THEN O_Lamp_Y_minus:= FALSE;
END_IF;
```

```
IF Ext_manual = 21
AND I_Z_plus = TRUE
THEN O_Lamp_Z_plus:= TRUE;
END_IF;
```

```
IF Ext_manual = 21
AND I_Z_plus = FALSE
THEN O_Lamp_Z_plus:= FALSE;
END_IF;
```

```
IF Ext_manual = 21
AND I_Z_minus = TRUE
THEN O_Lamp_Z_minus:= TRUE;
END_IF;
```

```
IF Ext_manual = 21
AND I_Z_minus = FALSE
THEN O_Lamp_Z_minus:= FALSE;
END_IF;
```

(* Axes selection lamps in Incremental mode *)

```
IF Ext_manual = 22
AND I_X_plus = TRUE
THEN O_Lamp_X_plus:= TRUE;
END_IF;
```

```
IF Ext_manual = 22
AND I_X_minus = TRUE
THEN O_Lamp_X_minus := TRUE;
END_IF;
```

```
IF Ext_manual = 22
AND I_Y_plus = TRUE
THEN O_Lamp_Y_plus:= TRUE;
END_IF;
```

```
IF Ext_manual = 22
AND I_Y_minus = TRUE
THEN O_Lamp_Y_minus:= TRUE;
END_IF;
```

```
IF Ext_manual = 22
AND I_Z_plus = TRUE
THEN O_Lamp_Z_plus:= TRUE;
END_IF;
```

```
IF Ext_manual = 22
AND I_Z_minus = TRUE
THEN O_Lamp_Z_minus:= TRUE;
END_IF;
```

```
IF Ext_manual = 22
AND ON_CYCLEON = FALSE
THEN O_Lamp_X_plus:= FALSE;
      O_Lamp_X_minus:= FALSE;
      O_Lamp_Y_plus:= FALSE;
      O_Lamp_Y_minus:= FALSE;
      O_Lamp_Z_plus:= FALSE;
      O_Lamp_Z_minus:= FALSE;
END_IF;
```

(* Axes selection lamps Off *)

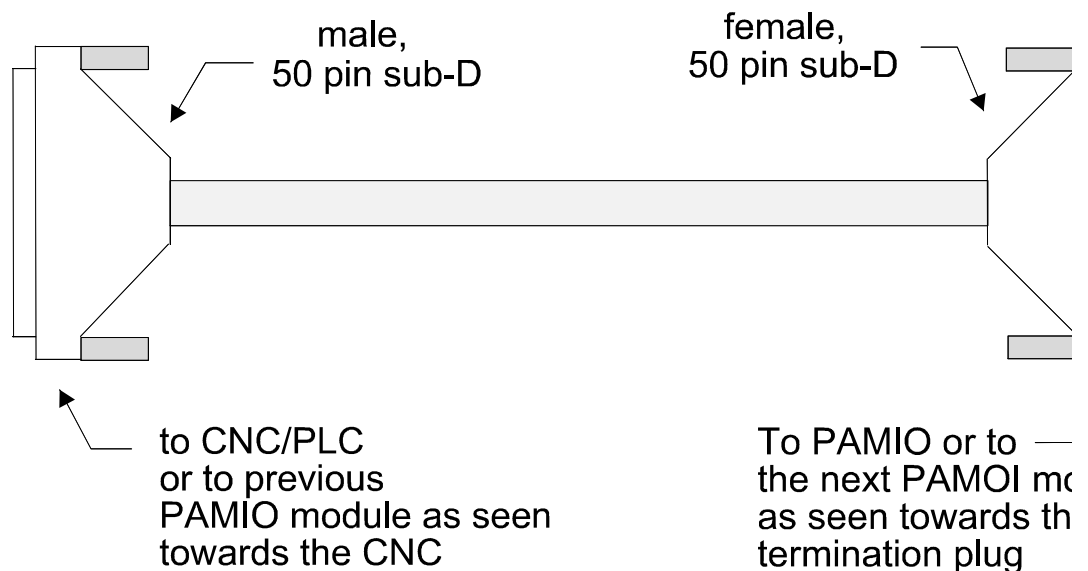
```
IF Time_2.Q = TRUE
THEN O_Lamp_X_plus:= FALSE;
      O_Lamp_X_minus:= FALSE;
      O_Lamp_Y_plus:= FALSE;
      O_Lamp_Y_minus:= FALSE;
      O_Lamp_Z_plus:= FALSE;
      O_Lamp_Z_minus:= FALSE;
END_IF;
```

Appendix 4 PA cables

Note:

- All the cables delivered by Power Automation are only designed for stationary use. The bending radius cannot be smaller than 30 mm.

cable type A			
connectors			
length	Name	CNC side	PAMIO side
1 m	10 10080200	50 pin sub-D, male	50 pin sub-D, female
2 m	10 10080201	50 pin sub-D, male	50 pin sub-D, female
5 m	10 10080300	50 pin sub-D, male	50 pin sub-D, female
10 m	10 10080400	50 pin sub-D, male	50 pin sub-D, female
20 m	10 10080500	50 pin sub-D, male	50 pin sub-D, female
35 m	10 10080600	50 pin sub-D, male	50 pin sub-D, female



cable type B			
connectors			
length	Name	CNC side	PAMIO side
2 m	10 01070100	50 pin sub-D, male	50 pin flat, multi point

cable type C			
connectors			
length	Name	CNC side	PAMIO side
2 m	Delivered with I/O box	50 pin sub-D, female	50 pin sub-D, male

cable type D			
connectors			
length	Name	CNC side	PAMIO side
2 m	Delivered with I/O box	15 pin sub-D, female	9 pin sub-D, male