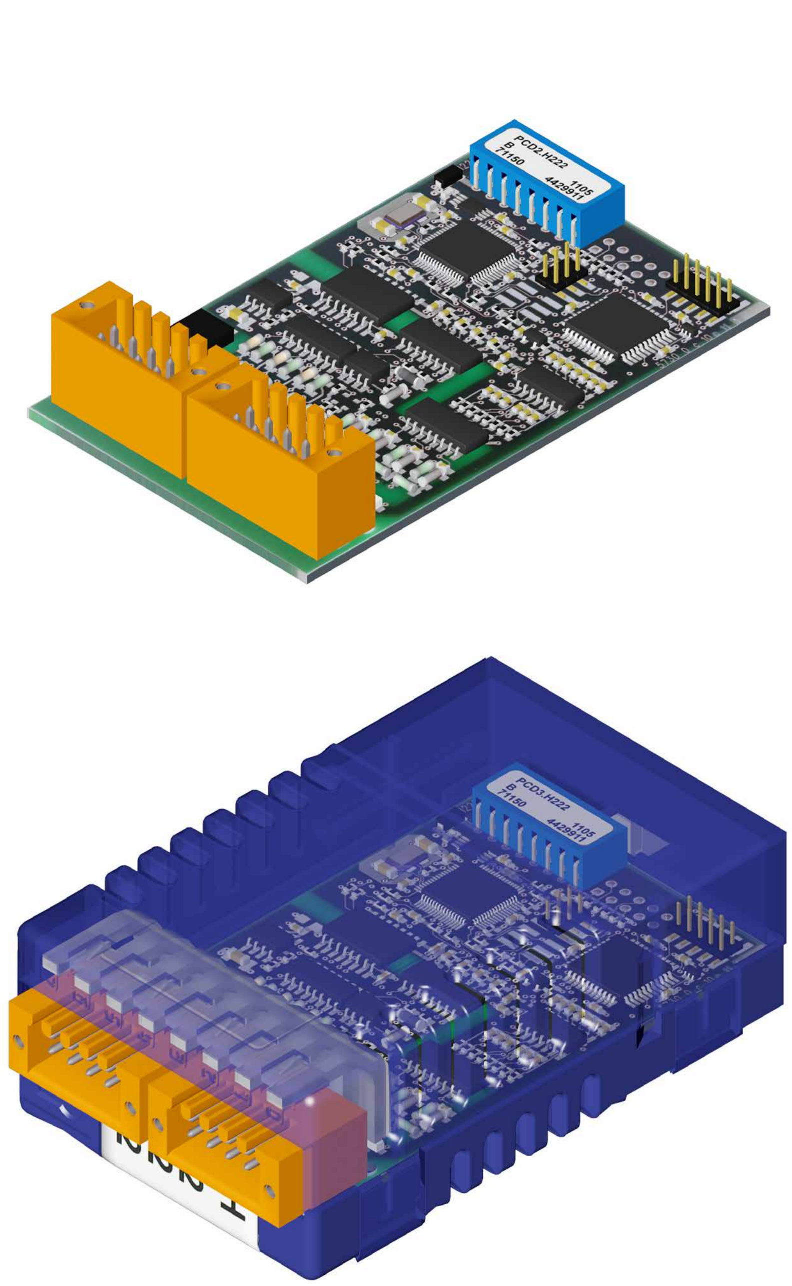
Manual

SAIA BURGESS CONTROLS



# Stepping motor module PCD2/3.H222

Document 26-878 | Version ENG05 | 2017-05-30

## 0.1 Document History

| Version | Edited | Published | Comments |
|---|---|---|---|
| EN01 | 2011-01-03 | 2011-04-15 | First version |
| EN02 | 2011-08-25 | 2011-08-26 | New phone number: +41 (0)26 / 672 72 72<br>Corrected the variable name: "IO.Slot0.IOAccess.X_AXIS_RD_ACTPOSABS" |
| EN03 | 2013-06-14<br>2014-01-09 | 2013-08-14<br>2014-01-09 | Logo and name changed<br>Deleted wrong chapter |
| EN04 | 2014-02-19 | 2014-02-19 | Various changes throughout the whole hand book |
| ENG05 | 2017-05-30 | 2017-05-30 | - New phone number: +41 (0)26 / 580 30 00<br>- Line "IO.Slot0.IOAccess.X_AXIS_RD_WARN" Deleted in 4.4.2 |

## 0.2 Trademarks

Saia PCD® and Saia PG5® are registered trademarks of Saia-Burgess Controls AG.

Modifications may be made in accordance with the latest technological standards.

Published in Switzerland

# 1 Introduction

This module can be connected to any I/O slot on a PCD2 or PCD3. It controls the power setting of a stepping motor axis up to a frequency of 20 kHz.

The maximum number of modules is defined by the limitations of the expansions PCD2 and PCD3. The base address 255 may not be occupied by a module.

## 1.1 Function and application

The module PCD2/3.H222 can control two stepping motor end phases. The module controls the required travel movements and transmits the three signals MOTEN (release), DIR (direction of rotation) and PUL (pulse) to the end phase. In addition to single steps and several steps at a constant speed, run profiles with trapezoidal or S shape can be performed with asymmetrical run-up and brake ramps. Positioning can be either absolute or relative.

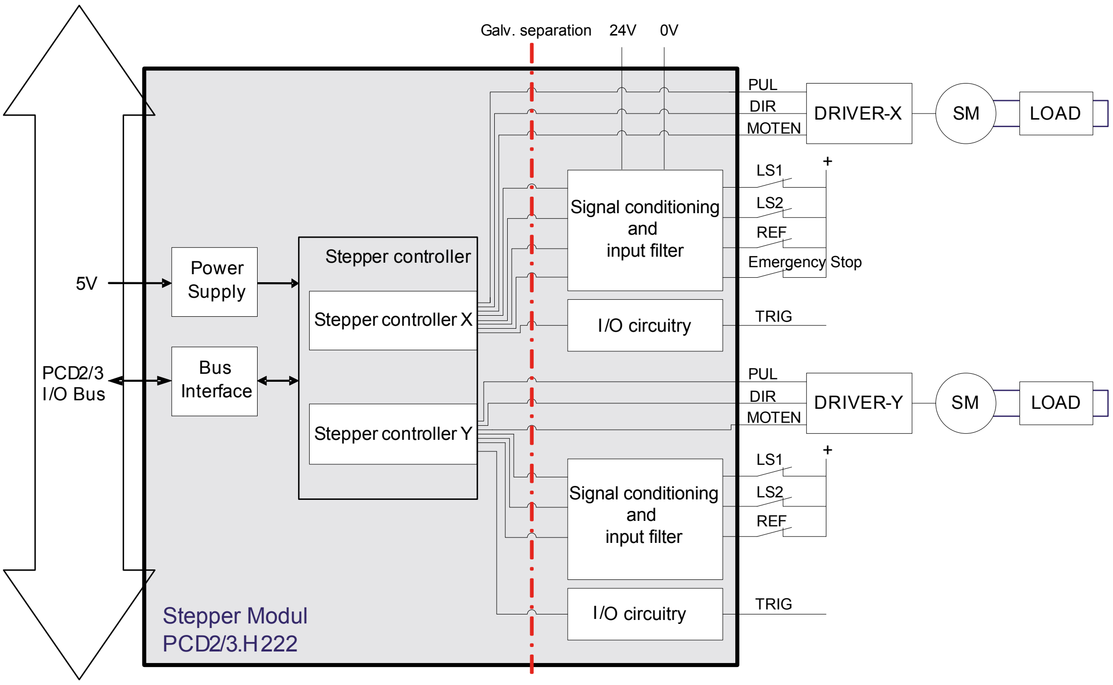
There are 3 inputs available per axis LS1 (limit switch 1), LS2 (limit switch 2) and REF (reference switch). This means that a single command can trigger positioning via the corresponding switch.

A configurable TRIG (trigger) connection per axis starts travel commands via a 24 V signal. If the TRIG connection is used as input for one axis and output for another, this allows run profiles to be started synchronously across more than one module.

A further input per module is available for detection of the emergency stop.

All inputs and outputs are electrically isolated from the I/O bus.

***Block diagram for a stepping motor drive***



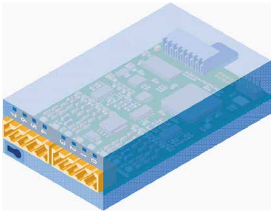
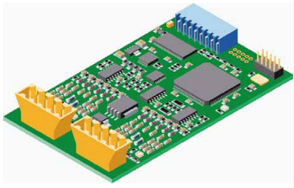
## 1.2 Main Characteristics

- A common emergency input
- 3 inputs (1 reference switch and 2 limit switches) per axis
- 3 outputs per axis (PUL, DIR, MOTEN)
- A configurable synchronising input/output per axis
- Possible positioning: single step, constant speed, run profiles automatic travel to reference or limit switch
- 16 profiles per axis for configuring S or trapezoidal shapes with asymmetric run-up and brake ramps
- absolute or relative position entry
- endless travel movements

## 1.3 Typical Applications

- Handling and assembly machines
- Pick and place functions
- Cost-effective palletting and assembly drives
- Automatic angle control, e.g. for cameras, floodlights, antennas, etc.
- Positioning of static axes (set up)

## 1.4 Mechanical Dimensions

| | |
|---|---|
| **PCD3 module**<br> | Dimensions: PCD3 I/O module<br>Casing colour: blue<br>Connection: 2x 10 pole plug-in spring terminal block |
| **PCD2 module**<br> | Dimensions: PCD2 I/O module<br>Connection: 2x 10 pole plug-in spring terminal block |

# 2 Technical Data

## 2.1 Technical data for hardware

### 2.1.1 24 VDC power supply

| Parameters | Unit |
|---|---|
| Voltage range | 18…32 VDC |
| Power consumption at 24 V | < 80 mA |
| Overvoltage protection | Yes (39 V ±10%) |
| Reverse voltage protection | no |

### 2.1.2 5 VDC power supply for PCD1/2/3 bus

| Parameters | Unit |
|---|---|
| Power consumption at 5 V | < 85 mA |

### 2.1.3 Inputs

| Parameters | Unit |
|---|---|
| Number | 7 |
| Voltage range | 0 … 32 V |
| Low range | 0 … 5 V |
| High range | 15 … 32 V |
| Threshold level Low-High | c. 10.9 V |
| Threshold level High-Low | c. 9.1 V |
| Hysteresis | c. 1.8 V |
| Input current (24 V) | approx. 5.2 mA |
| Turn-on delay after applying mains voltage | c. 200 µs |
| Turn-off delay | c. 200 µs |

### 2.1.4 Ouputs

| Parameters | Unit |
|---|---|
| Number | 6 |
| Step pulse (PUL) | active high |
| Pulse range for step signal | 16.5 µs |
| Directional signal (DIR) | low = forwards (cw)<br>high = backwards (ccw) |
| Movement signal (MOTEN) | high = motor enabled<br>low = motor not enabled |
| Load per output | < 20 mA |
| Operating mode | Economy operation (minus switched) |
| Turn-on delay MOTEN, DIR | < 15 µs |
| Turn-off delay MOTEN, DIR | < 25 µs |
| Turn-on delay PUL | < 1.5 µs |
| Turn-off delay PUL | < 500 ns |

### 2.1.5 Bidirectional connections

There are two bidirectional connections available:

Input/Output TRIG X axis.
Input/Output TRIG Y axis.

This connection can be used as output or input. This means a master axis can be set for synchronising slave axes connected with it.

***Characteristics when used as input***

| Parameters | Unit |
|---|---|
| Voltage range | 0 … 32 V |
| Low range | 0 … 5 V |
| High range | 15 … 32 V |
| Threshold level Low-High | c. 9.3 V |
| Threshold level High-Low | c. 11 V |
| Hysteresis | c. 1.7 V |
| Input current (24 V) | 3.85 mA |
| Turn-on delay after applying mains voltage | < 170 µs |
| Turn-off delay | < 184 µs |

***Characteristics when used as output***

| Parameters | Unit |
|---|---|
| Voltage range | 0 … 32 V |
| Low range | 0 … 5 V |
| High range | 15 … 32 V |
| Load per output | > 200 mA |
| Turn-on delay after applying mains voltage | < 1.5 µs |
| Turn-off delay | < 28 µs |
| Operating mode | Source operation |

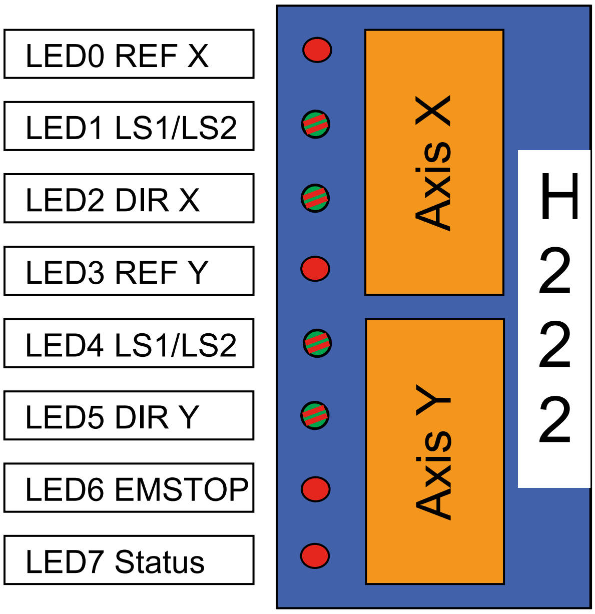
### 2.1.6 Operating conditions

| | |
|---|---|
| Ambient temperature | Operation: 0...+55 °C without forced ventilation, storage: -20...+85 °C |

### 2.1.7 Order information

| Model | Description | Weight |
|---|---|---|
| PCD2.H222 | Stepper module for 2 axes | 27 g |
| PCD3.H222 | Stepper module for 2 axes | 70 g |

## 2.2 Terminal connections



| X axis | | | |
|---|---|---|---|
| 0 | EMSTOP | LS1 | 1 |
| 2 | LS2 | REF | 3 |
| 4 | TRIG | MOTEN | 5 |
| 6 | DIR | PUL | 7 |
| 8 | 0V | 24V | 9 |

| Y axis | | | |
|---|---|---|---|
| 0 | | LS1 | 1 |
| 2 | LS2 | REF | 3 |
| 4 | TRIG | MOTEN | 5 |
| 6 | DIR | PUL | 7 |
| 8 | 0V | 24V | 9 |

An emergency stop switch can be connected to the input EMSTOP. Pin assignment is shown in the diagram. This input is used to detect an emergency stop and may not be used for safety functions. The connections LS1, LS2 und REF are inputs intended for positioning via one of the switches. In addition, LS1 and LS2 limit the useable range of a stepping motor axis. The configurability of these inputs allows both make and brake contacts to be used and switches not connected can be deactivated, hence improving troubleshooting.

If one of the inputs is deactivated it can be used as a digital input. The IO status register must be readout by the user program.

The TRIG connection can be configured as input or output and is used for event-triggered startup of different axes.

If the connection is activated and configured as input, a step movement started from the user program will only start when 24V are applied to the TRIG connection.

If TRIG is used as output it is switched on and off simultaneously with the MOTEN output signal.

MOTEN, DIR and PUL are outputs used in economy operation to control the stepping motor end phase.

The power supply to the module enters via the +24V and 0V connection. Both these signals are connected internally between the terminals of the X and Y axes.

With the exception of the EMSTOP signal, all +24V and 0V signals are available once for the X and once for the Y axis.

## 2.3 LED display

X axis

| | |
|---|---|
| LED 0 | Voltage in input REF |
| LED 1 (red) | Voltage in input LS1 |
| LED 1 (green) | Voltage in input LS2 |
| LED 2 (red) | Voltage in output MOT and output DIR |
| LED 2 (green) | Voltage in output MOT and 0V in output DIR |

Y axis

| | |
|---|---|
| LED 3 | Voltage in input REF |
| LED 4 (red) | Voltage in input LS1 |
| LED 4 (green) | Voltage in input LS2 |
| LED 5 (red) | Voltage in output MOT and output DIR |
| LED 5 (green) | Voltage in output MOT and 0V in output DIR |

For both axes

| | |
|---|---|
| LED 6: | Voltage in input EMSTOP |
| LED 7: | Visualisation of operating condition and error codes |

EMSTOP and REF are visualised via a single LED. LS1 and LS2 are indicated on a 2-colour LED (LS1 red and LS2 green). A 2-colour LED also indicates the status of DIR and MOTEN. If MOTEN = 1, the colour of the LED changes according to the rotational direction DIR (counterclockwise “ccw” red and clockwise “cw” green).

LED7 indicates a possible fault on one of the two axes.

# 3 Functional description

Module H222 is installed in a PCD2 or PCD3 and controlled from the user program via peripheral access. This starts communication between PCD and H222 via the standard IO bus. The module recognises new travel commands automatically and initiates the correct measures to execute them.

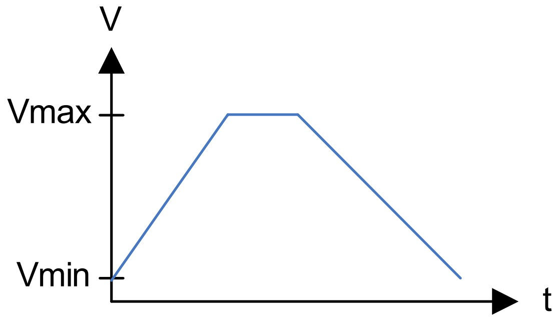
## 3.1 Configuring run profiles

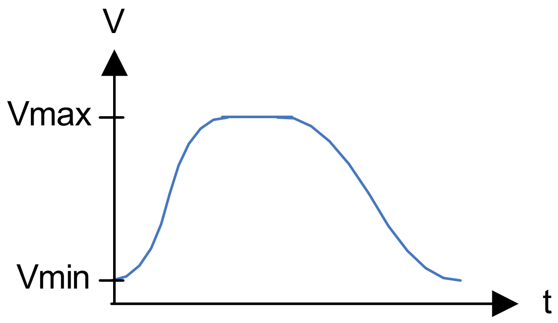
To avoid transmitting all parameters for a run profile every time from the PCD to the module, 16 run profiles can be defined and stored. These can then be activated by a single command.

A run profile comprises the following parameters:

| Parameters | Description | Range, unit | Resolution, comment |
|---|---|---|---|
| Vmin | Start-Stop speed | 10…10’000 Hz | 1 Hz (16 bit) |
| Vmax | Maximum speed | 20…20’000 Hz | 1 Hz (16 bit) |
| Acc | Average speed | 1...1000 kHz/s | 1 kHz/s (16 bit) |
| Dec | Average delay | 1...1000 kHz/s | 1 kHz/s (16 bit) |
| Jerk_acc | Jerk run-up ramp as a percentage | 0...50% | 1% (6 bit) |
| Jerk_dec | Jerk brake ramp as a percentage | 0...50% | 1% (6 bit) |

Both jerk values show at what percentage of time for the run-up or brake phase the linear acceleration or braking will have been reached. The linear phase will last 100%-2*jerk value. therefore a jerk value of 0% corresponds to a trapezoidal run profile, whereas 50 % indicates an S-curve.





The duration of a particular profile is determined by these parameters and the distance of travel.

### 3.1.1 Automatic profile correction

If values are set outside the permitted range, the module will correct these automatically to the corresponding minimum or maximum values. A warning will be sent. If one of the values Vmin, Vmax, Acc or Dec should be 0, the profile will not start and an error message will be sent.

There are dependencies between certain input parameters which must be respected If they are ignored it may happen that an S-curve is reduced to a trapezoidal run to avoid stepping errors.

## 3.2 Travel distance setting

The travel distance is set independent of the run profile either as a relative or absolute value or as an endless travel movement. The latter is a special case of a relative distance setting. Travel distance settings affect the positioning types “positioning with constant speed” and “positioning with run profile”.

### 3.2.1 Relative travel distance

A relative travel distance setting defines the number of steps to run. When the travel movement starts, the rotational direction must be indicated. The permitted number of steps lies between 0 and 16777215, whereby 16777215 indicates endless travel movement. All other values indicate the corresponding number of steps.

### 3.2.2 Absolute setting

Absolute positioning sets the end point when the current position is not known. The module automatically calculates the relative travel distance and rotational direction in relation to the current absolute position. When the travel movement starts, the parameter value rotational direction is ignored.

## 3.3 Response to emergency stop during operation

A ongoing movement of both axes of the module can be stopped via the emergency stop input with a predefined brake ramp. Since the effective position is no longer correct after an external fast shutdown, the machine must travel to the limit switch or reference switch after an emergency stop.

The module must initiate fast shutdowns for safety reasons independently.

### 3.4 Response to LimitSwitch during operation

When ‘LS1’ or ‘LS2’ are active, a brake ramp is triggered automatically. The current absolute position is not lost. However, an error message is displayed and must be acknowledged. The active range of a limit switch can only be abandoned by positioning via the corresponding limit switch or at least by a reference run.

### 3.5 Event-triggered axis start-up

If the TRIG connection is configured as an input and activated, travel commands can be implemented on one axis which will only start when 24V is applied to the TRIG connection. Any digital 24V signal can be used.

If two or more axes are started synchronously, the TRIG connection must be configured as input and activated for slave axes. The connection must be activated as output for the master axis. All connections must be connected to each other via an external cable. Firstly the travel commands for all slave aches are transmitted. These then wait for the start signal at the TRIG connection. This is given when the master axis starts at the corresponding point in time.

### 3.6 Positioning via limit switch

Two different starting positions must be differentiated for runs to a limit switch. Either the machine is between the two limit switches and moves towards the limit switch with the defined spped profile, brakes to a standstill as soon as the limit switch activates and then travels at Vmin in the opposite direction until the limit switch becomes inactive. Or else the machine is within the active range of the limit switch flag. The machine travels directly at Vmin to the valid range between the limit switches until the limit switch becomes inactive.

It is not possible to move directly to the nearest limit switch from an invalid range (outside the two limit switches). The machine must first move to within the valid range.

Positioning via one of the two limit switches is started via peripheral access and according to the required axis using the command X_AXIS_WR_LSREFMOVE or Y_AXIS_WR_LSREFMOVE. The parameter value determines which limit switch is used for positioning (1 = LS1, 2 = LS2).

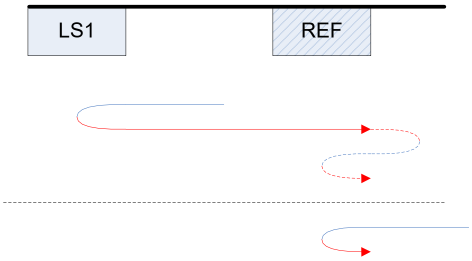
## 3.7 Reference run

Various modes are available for positioning via a connected reference switch. The reference switch must be activated independent of the mode, otherwise an error message will be sent.

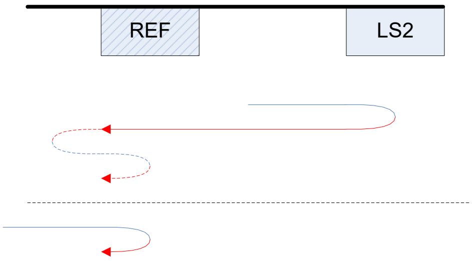
### 3.7.1 Searching for limit switch followed by travel to reference switch

If referencing is performed in a forwards direction (reference travel mode 3), the run starts with the active speed profile in a backwards direction until the reference or limit switch becomes active. This triggers the braking process. After coming to a standstill, the direction of travel is changed and the machine moves to the reference switch using the speed profile. When the reference switch becomes active, the delay response is triggered. If the other limit switch becomes active the reference run will stop and an error will be displayed.

If the reference switch is active after the delay ramp, the machine will continue to travel at Vmin without interruption until the reference switch becomes inactive. Otherwise the machine will first travel in the opposite direction at Vmin to the reference switch and after will travel at Vmin in a forwards direction out of the active range of the reference switch.



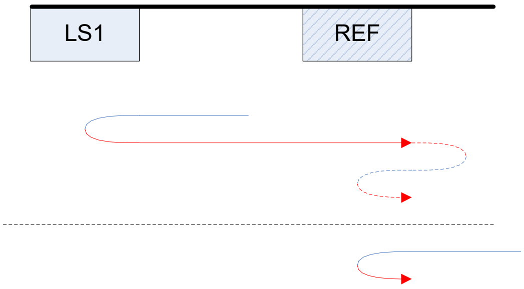
Reference travel mode 3



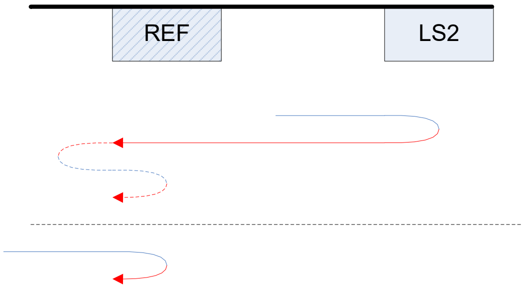
Reference mode 4

### 3.7.2 Searching for the reference switch outside the limit switches

In reference mode 5 it is possible to mount the reference switch outside the limit switches in order to continue moving to the reference switch despite the active limit switches.



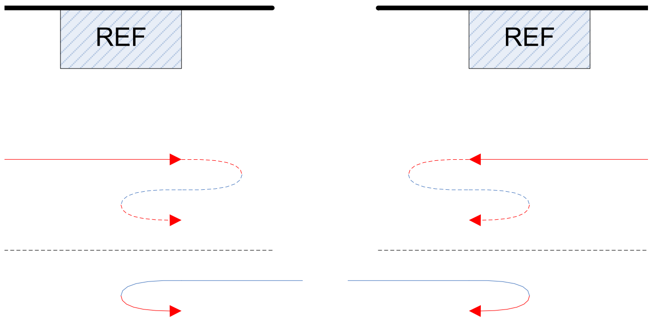
Reference travel mode 5



Reference mode 6

### 3.7.3 Direct travel to and across the reference switch

With this type of referencing the machine starts directly with the final rotational direction and crosses the reference switch. If LS1 becomes active in reference mode 7 (insofar as it is activated) the reference run is stopped and an error message is sent. The same applies to Mode 8 for LS2.



Reference mode 7 Reference mode 8

## 3.8 Single steps

Each single step entails a command from the PCD. This is issued with the defined value for Vmin. If the value has not been defined, the machine shifts at a frequency of 10 Hz. The maximum step sequence possible in single step operation is at 350 Hz.

## 3.9 Positioning with constant speed

When movement starts at Vmin the Vmin parameters and the number of steps to be performed are used. If the number of steps is set at 16,777,215 ($2^{24}$-1) movement will start with Vmin and will only stope when the corresponding limit switch is reached or when a stop command is sent from the user program.

## 3.10 Positioning using a run profile

This type of positioning is the most common and is the fastest method to reach a target position. The current run profile parameters are used and an S-curve, a trapezoid, a triangle or a constant speed can be implemented depending on the parameter settings and the positioning distance.

## 3.11 Stopping a run

Once started, runs can be aborted via a stop instruction. The predefined brake ramp is used and the axis brought to a standstill. This instruction triggers and error message which must be acknowledged before issuing the next travel command.

# 4 Programmable functions in PG5

Take note of the different activation options for the module via direct access (Media Mapping) or function blocks (FB)

## 4.1 Access using function blocks (FB)

The H222 module is used in a PCD2 or PCD3 and controlled using FBs in the PG5 programming tool. These trigger communication between the PCD and H222 via the standard IO bus. This sets the motion parameters and travel commands in the relevant register of the microcontroller on the expansion module.

The module firmware recognises new travel commands automatically and introduces the correct measures to run them. In order to prevent all parameters for one run profile from having to be sent from the SPS to the model each time, 16 run profiles can be defined which, after being saved, can be activated by a single command.

This model enables emergency stop, limit and reference switches to be connected. However, safety-related quick stoppages must be performed externally. By being able to configure the inputs, both break and make contacts can be used and switches that are not connected can be deactivated. This results in increased safety through better error detection.

The module is initialised once using the “Init” function block. The “InitMP” function block enables an entire run profile to be transmitted using a FB call. The “Exec” FB enables a “command” to be passed. Depending on the type of instruction, a databyte is sent or read as a return parameter. A list of all permissible “commands” and a description of them can be found in Section [...]

### 4.1.1 Diagnostic flag and register

The diagnostic register and diagnostic flag are updated for each initialised axis after each action.

**Structure of the rDiag diagnostic register:**

```
;          31.....24 23......16 15........8 7..........0
;          \ Error / \ FB No. / \ Par.No. / \Module No./
;
;          Error:      00h -> No special error
;                      01h -> H222 FB already in use
;          FB No.:     01h -> H222.Init
;                      02h -> H222.InitMP
;                      03h -> H222.Exec
;          Par.No.:    00h -> No parameter error
;                      01h -> Module number not correct
;                      02h -> Axis number not correct
;                      03h -> Command code not correct (<0 or >255)
;          Module No:  Number of the Module (FB parameter 1) which
;                      causes the error
```

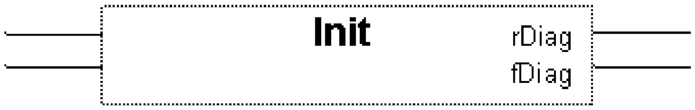
### 4.1.2 init FB

This FB initialises all H222 modules that have been defined in the corresponding .equ file. fDiag shows whether or not any errors have occurred during the initialisation. rDiag is used to identify these errors.

**The call for the "INIT" FB is as follows:**

```
CFB   H222.Init
R rDiag     ; Par. 1: Diagnostic register
F fDiag     ; Par. 2: Status flag (0: Init OK, 1: Error)
```

Or displayed as an FB call:



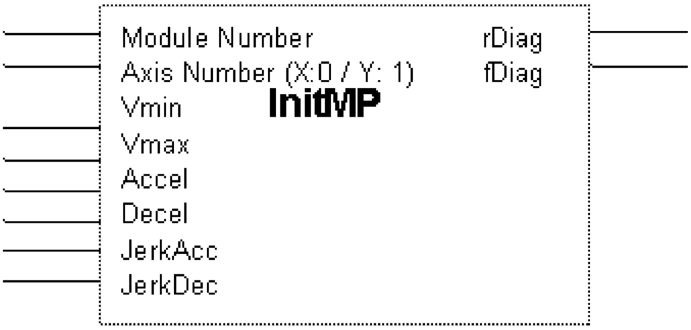
### 4.1.3 initMP FB

This FB initialises a motion profile (MP) and configures the limit, reference and emergency stop switches. rDiag delivers a response which helps to determine the cause of any errors. fDiag shows whether or not any errors have occurred during the initialisation.

**The call for the “INITMP” FB is as follows:**

```
CFB  H222.InitMP
R modNr    ; Par. 1: Module base address
R axisNr   ; Par. 2: Axis number (0 = X-axis, 1 = Y-axis)
R Vmin     ; Par. 3: Vmin in Hz
R Vmax     ; Par. 4: Vmax in Hz
R Accel    ; Par. 5: Acceleration in kHz/s
R Decel    ; Par. 6: Braking in kHz/s
R JerkAcc  ; Par. 7: Jerk value in percent for acceleration (0 to 50
%)
R JerkDec  ; Par. 8: Jerk value in percent for braking (0 to 50 %)
R rDiag    ; Par. 9: Diagnostic register
F fDiag    ; Par. 10: Status flag (0: Init OK, 1: Error)
```

Or displayed as an FB call:



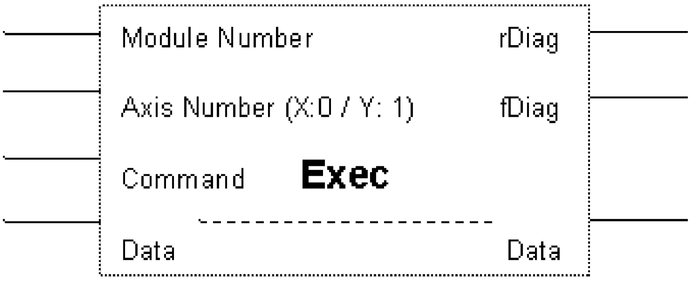
### 4.1.3 exec FB

The “exec” individual module allows the following commands to be run and, if required, further data such as the direction of rotation or similar to be sent. Other commands supply return values in data mode (see description of commands). fDiag signals that an error has occurred and rDiag helps to assign a cause to this error.

**The call for the “EXEC” FB is as follows:**

```
CFB    H222.Exec
R modNr    ; Par. 1: Module base address
R axisNr   ; Par. 2: Axis number (0 = X-axis, 1 = Y-axis)
R rCommand ; Par. 3: Command (see list of commands below)
R rData    ; Par. 4: Transfer or read parameter (currently optional)
R rDiag    ; Par. 5: Diagnostic register
F fDiag    ; Par. 6: Status flag (0: Exec OK, 1: Error)
```

Or displayed as an FB call:



**The exec FB module supports the following commands:**

**Commands:**

- Single step (SingleStep)
- Search for and leave reference run or LS or leave overrun area (LsRefMove)
- Positioning with constant velocity (MoveVconst)
- Positioning with velocity profile (StartMP)
- Stop motion command (Stop)
- Approve motion commands after “Stop” (Cont)
- Save run profile (SaveMP)
- Activate run profile (ActMP)
- Restart module firmware (ResMC)
- Clear error register (ClrError)

**Commands for which parameters can be set:**

- Load average acceleration (LdAcc)

- Load average braking delay (LdDec)
- Load maximum speed (LdVmax)
- Load minimum speed (LdVmin)
- Load jerk value for acceleration range (LdJPa)
- Load jerk value for braking range (LdJPd)
- Load destination as relative position (LdDestRel)
- Load destination as absolute position (LdDestAbs)
- Set absolute position for current axis position (LdActPosAbs)
- Load limit switch logic (LdIoConf)

**Read commands:**

- Read average acceleration (RdAcc)
- Read average braking delay (LdDec)
- Read maximum speed (RdVmax)
- Read minimum speed (RdVmin)
- Read jerk value for acceleration range (RdJPa)
- Read jerk value for braking range (RdJPd)
- Read destination as relative position (RdDestRel)
- Read destination as absolute position (RdDestAbs)
- Read actual position as relative position (RdActPos)
- Read actual positioning time (RdActRuntime)
- Read actual speed (RdActSpeed)
- Read actual absolute position (RdActPosAbs)
- Read error (RdError)
- Read status (RdStatus)
- Read I/O configuration (RdIoConf)
- Read firmware version (RdFwVersion)

### 4.1.5 Commands

| Par.3 (rCommand) | Par.4 (rData) |
|---|---|
| StartMP | initialize and start motion<br>Par.4 =<br>0: start actual motion profile cw<br>1: start actual motion profile ccw<br>(endless move if LdDestRel with value 0xFFFFFF (16777215) was set) |
| Stop | stops motion<br>(this instruction sets a HALT error in error register which needs to be cleared before next move instruction by Cont or ClrError instruction) |
| Cont | Clears HALT error in error register, which was set when Stop instruction was proceeded |
| SingleStep | single step with actual Vmin<br>Par.4 . =<br>0: direction cw<br>1: direction ccw |
| MoveVconst | Positioning with min. frequency<br>Par.4 . =<br>0: direction cw<br>1: direction ccw<br>(endless move if LdDestRel with value 0xFFFFFF (16777215) was set) |
| LsRefMove | positioning on limit switch or reference switch border<br>Par.4 . =<br>1: LS1<br>2: LS2<br>3: REF cw (search LS1 or REF in dir ccw)<br>4: REF ccw (search LS2 or REF in dir cw)<br>5: REF cw in invalid area under LS1(ccw first)<br>6: REF ccw in invalid area over LS2 (cw first)<br>7: REF cw (direct to REF)<br>8: REF ccw (direct to REF) |
| SaveMP | Save motion profile x (max. 16)<br>Par.4 . =<br>0: profile 0<br>1: profile 1<br>..<br>15: profile 15 |
| ActMP | activate motion profile x (max. 16)<br>Par.4 . =<br>0: profile 0<br>1: profile 1<br>..<br>15: profile 15 |
| ResMC | restart the module firmware, all profiles are deleted, errors are cleared<br>(Par.2 and Par.4 have no impact) |
| ClrError | Clears error register |

### 4.1.6 Write commands

| Par.3 (rCommand) | Par.4 (rData) |
|---|---|
| LdVmin | load min. frequency in Hz<br>Par.4 = value (2 Bytes) (range: 10…10’000) |
| LdVmax | load max. frequency in Hz<br>Par.4 = value (2 Bytes) (range: 20…20’000) |
| LdAcc | load acceleration in kHz/s<br>Par.4 = value (2 Bytes) (range: 1…1’000) |
| LdDec | load deceleration in kHz/s<br>Par.4 = value (2 Bytes) (range: 1…1’000) |
| LdJPa | load jerk value in percent for accel.<br>Par.4 = value (2 Bytes) (range: 0…50) |
| LdJPd | load jerk value in percent for decel.<br>Par.4 = value (2 Bytes) (range: 0…50) |
| LdIoConf | load input/output configuration<br>Bit 0: Input LS1 on/off (0 = on/1 = off)<br>Bit 1: Input LS2 on/off<br>Bit 2: Input REF on/off<br>Bit 3: Input Notstop on/off<br>Bit 4: Input LS1 Mode (0=opening/1=closing cont.)<br>Bit 5: Input LS2 Mode<br>Bit 6: Input REF Mode<br>Bit 7: Input Notstop<br>Bit 8: use TRIG as input (0)/output (1)<br>Bit 9: use TRIG (0 = TRIG not used/1 = TRIG used) |
| LdDestRel | load destination relative in steps<br>Par.4 = value (3 Bytes) (range: 0…16777216) |
| LdDestAbs | load destination as absolute position<br>Par.4 = value (4 Byte signed)<br>(range: -2’147’483’648…+2’147’483’647 but take care that relative destination is smaller than 16777216!)<br>LdDestAbs must be followed by MoveVconst or StartMP. Stop, SingleStep, LsRefMove will cancel LdDestAbs instruction. |
| LdActAbsPos | load absolute position for current motor position<br>Par.4 = value (4 Byte signed)<br>(range: -2’147’483’648…+2’147’483’647) |

### 4.1.7 Read commands

| Par.3 (rCommand) | Par.4 (rData) |
|---|---|
| RdVmin | read min. frequency in Hz<br>Par.4 = value (2 Bytes) |
| RdVmax | read max. frequency in Hz<br>Par.4 = value (2 Bytes) |
| RdAcc | read acceleration in kHz/s<br>Par.4 = value (2 Bytes) |
| RdDec | read deceleration in kHz/s<br>Par.4 = value (2 Bytes) |
| RdJPa | read jerk value in percent for accel.<br>Par.4 = value (2 Bytes) |
| RdJPd | read jerk value in percent for decel.<br>Par.4 = value (2 Bytes) |
| RdIoConf | read input/output configuration<br>Par.4 = value (2 Bytes)<br>Bit 0: Input LS1 on/off (0 = on/1 = off)<br>Bit 1: Input LS2 on/off<br>Bit 2: Input REF on/off<br>Bit 3: Input Notstop on/off<br>Bit 4: Input LS1 Mode (0=opening/1=closing cont.)<br>Bit 5: Input LS2 Mode<br>Bit 6: Input REF Mode<br>Bit 7: Input Notstop Mode<br>Bit 8: use TRIG as input (0)/output (1)<br>Bit 9: use TRIG (0 = TRIG not used/1 = TRIG used) |
| RdStatus | read module status<br>Bit 0: state of input LS1<br>Bit 1: state of input LS2<br>Bit 2: state of input REF<br>Bit 3: state of input Notstop<br>Bit 4: state of input/output TRIG<br>Bit 5: state of output DIR<br>Bit 6: state of output MOTEN<br>Bit 7: -<br>Bit 8: Wait for TRIG flag<br>Bit 9: on destination flag |
| RdDestRel | read destination relative in steps actual used for next motion commands<br>Par.4 = value (3 Bytes) |
| RdDestAbs | read destination absolute in steps actual used for next motion commands<br>Par.4 = value (3 Bytes) |
| RdActSpeed | read actual speed in Hz<br>Par.4 = value (2 Bytes) |
| RdActPos | read actual position in steps done since start of motion<br>Par.4 = value (3 Bytes) |
| RdActPosAbs | Read actual absolute position<br>Par.4 = value (4 Byte signed) |
| RdActRuntime | read actual runtime in ms<br>Par.4 = value (3 Bytes) |

| | |
|---|---|
| RdError | read error<br>Par.4 = value (2 Bytes)<br>Byte 0: Error code<br>0: no error<br>1: undefined error<br>2: communication error<br>3: prohibited parameter used<br>4: input disabled but used for motion<br>5: no REF found between LS1 and LS2<br>6: LS or REF move need to be done first when valid area was left<br>7: Stop instruction proceeded<br>8: EMSTOP is/was active<br>9: proceeded no. of steps did not match previewed<br>10: two or more parameter for motion profile do not fit together<br>Byte 1: Warning code<br>1: jerk values set to zero<br>2: Vmax never reached<br>3: step added at the end of motion profile to reach correct no. of steps<br>4: steps removed at the end of motion profile to reach correct no. of steps<br>5: Vstop to big (Vstop > 2*Vmin)<br>6: Vmin was bigger than Vmax -> Vmin reduced to Vmax<br>7: one or more parameter for motion profile out of range<br>8: no motion parameters loaded after module reset |
| RdFwVersion | read firmware version<br>Par.4 = value (4 Byte/Character)<br>format x.yy |

### Status display on module address

If the PCD does not communicate with the module (no H222 FB active), certain status bits are read as inputs directly on the module addresses. The LSB corresponds to the module base address.

Status bits: RO access

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x | x | x | x | FlagsValid | x | x | x | - | - | - | - | Error Y | Error X | OnDest Y | OnDest X |
| MSB | | | | | | | | | | | | | | | LSB |

OnDest X and Y: These bits are 0 from the moment when a travel command is invoked until this has been run. If a travel command is invoked and is waiting for a trigger signal, the “on destination” bit is also 0.

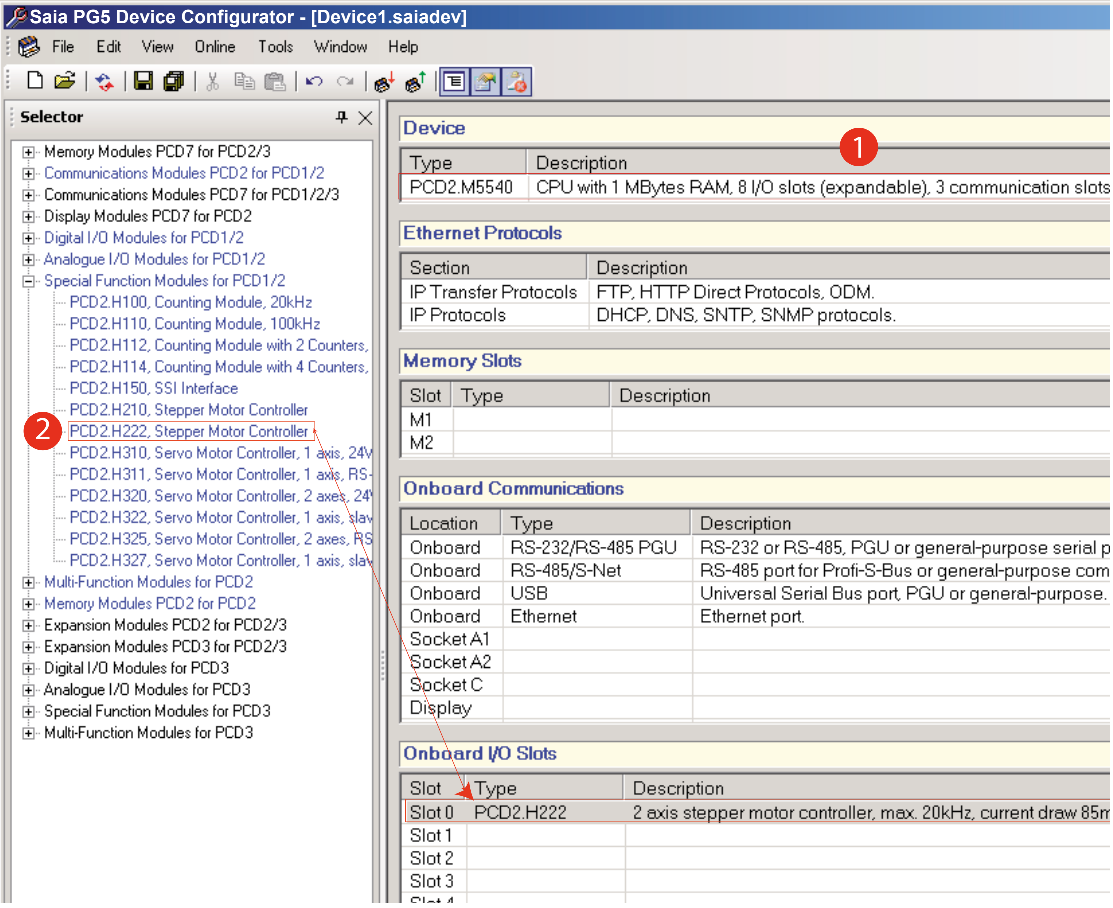
Error X and Y: If an error is detected for the relevant axis, the error bit is also set.

FlagsValid: This bit signals whether the flags on the input addresses 0…7 are valid (1) or not (0).

## 4.2 System configuration with the Device Configurator

At the start of a new project, first the central processing unit (CPU)❶ and the corresponding module must be selected using the Device Configurator. The module can be found under the “Special function modules for PCD 1/2” or “Special function modules for PCD 3” tab on the far left❷ and can be moved to the corresponding slot using drag and drop.

The properties window under the right-hand tab can be used to define access to the X and Y axes for inputs and outputs. The trigger function can be assigned to an input or output. This special trigger function can be used to configure the motion of the axes.


Saia PG5 Device Configurator - [Device1.saiadev]
File Edit View Online Tools Window Help
Selector
1
2

- Memory Modules PCD7 for PCD2/3
- Communications Modules PCD2 for PCD1/2
- Communications Modules PCD7 for PCD1/2/3
- Display Modules PCD7 for PCD2
- Digital I/O Modules for PCD1/2
- Analogue I/O Modules for PCD1/2
- Special Function Modules for PCD1/2
  - PCD2.H100, Counting Module, 20kHz
  - PCD2.H110, Counting Module, 100kHz
  - PCD2.H112, Counting Module with 2 Counters,
  - PCD2.H114, Counting Module with 4 Counters,
  - PCD2.H150, SSI Interface
  - PCD2.H210, Stepper Motor Controller
  - PCD2.H222, Stepper Motor Controller
  - PCD2.H310, Servo Motor Controller, 1 axis, 24V
  - PCD2.H311, Servo Motor Controller, 1 axis, RS-
  - PCD2.H320, Servo Motor Controller, 2 axes, 24V
  - PCD2.H322, Servo Motor Controller, 1 axis, slav
  - PCD2.H325, Servo Motor Controller, 2 axes, RS
  - PCD2.H327, Servo Motor Controller, 1 axis, slav
- Multi-Function Modules for PCD2
- Memory Modules PCD2 for PCD2
- Expansion Modules PCD2 for PCD2/3
- Expansion Modules PCD3 for PCD2/3
- Digital I/O Modules for PCD3
- Analogue I/O Modules for PCD3
- Special Function Modules for PCD3
- Multi-Function Modules for PCD3

**Device**

| Type | Description |
|---|---|
| PCD2.M5540 | CPU with 1 MBytes RAM, 8 I/O slots (expandable), 3 communication slots |

**Ethernet Protocols**

| Section | Description |
|---|---|
| IP Transfer Protocols | FTP, HTTP Direct Protocols, ODM. |
| IP Protocols | DHCP, DNS, SNTP, SNMP protocols. |

**Memory Slots**

| Slot | Type | Description |
|---|---|---|
| M1 | | |
| M2 | | |

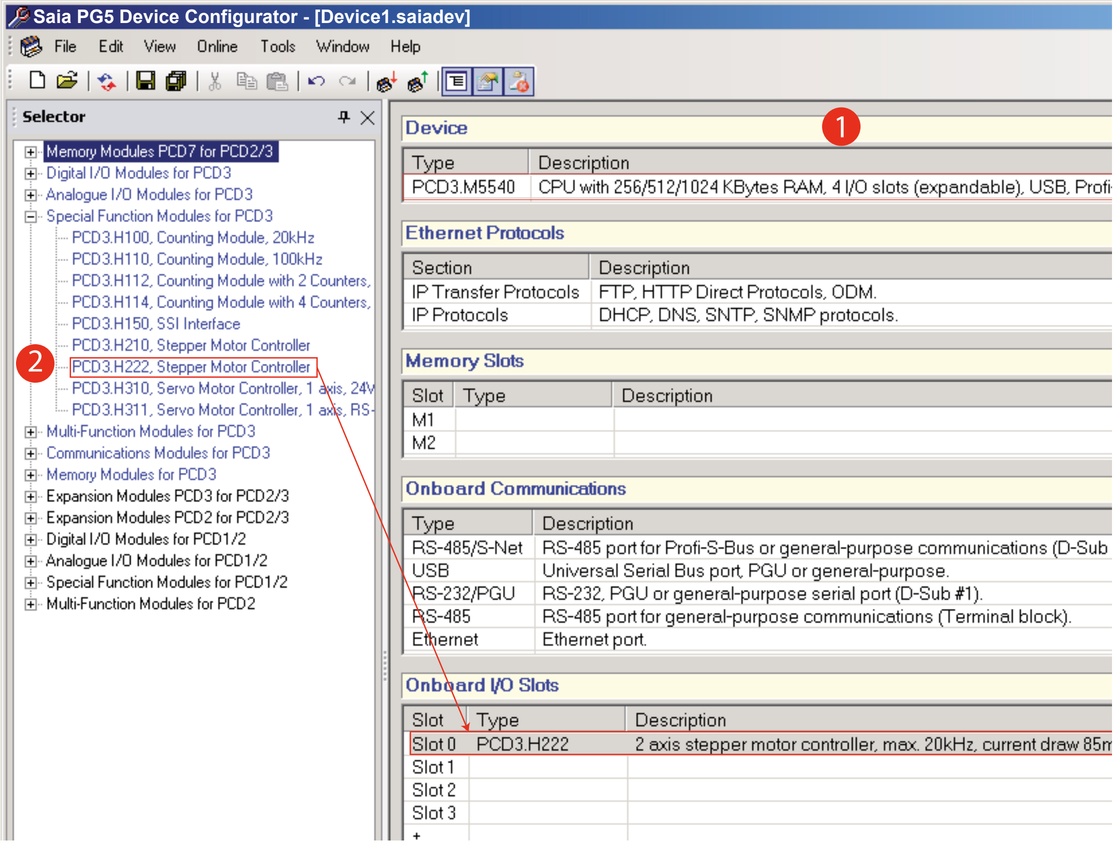
**Onboard Communications**

| Location | Type | Description |
|---|---|---|
| Onboard | RS-232/RS-485 PGU | RS-232 or RS-485, PGU or general-purpose serial p |
| Onboard | RS-485/S-Net | RS-485 port for Profi-S-Bus or general-purpose com |
| Onboard | USB | Universal Serial Bus port, PGU or general-purpose. |
| Onboard | Ethernet | Ethernet port. |
| Socket A1 | | |
| Socket A2 | | |
| Socket C | | |
| Display | | |

**Onboard I/O Slots**

| Slot | Type | Description |
|---|---|---|
| Slot 0 | PCD2.H222 | 2 axis stepper motor controller, max. 20kHz, current draw 85m |
| Slot 1 | | |
| Slot 2 | | |
| Slot 3 | | |

*Device Configurator: Example using PCD2.*


Saia PG5 Device Configurator - [Device1.saiadev]
File Edit View Online Tools Window Help
Selector
Memory Modules PCD7 for PCD2/3
Digital I/O Modules for PCD3
Analogue I/O Modules for PCD3
Special Function Modules for PCD3
PCD3.H100, Counting Module, 20kHz
PCD3.H110, Counting Module, 100kHz
PCD3.H112, Counting Module with 2 Counters,
PCD3.H114, Counting Module with 4 Counters,
PCD3.H150, SSI Interface
PCD3.H210, Stepper Motor Controller
PCD3.H222, Stepper Motor Controller
PCD3.H310, Servo Motor Controller, 1 axis, 24V
PCD3.H311, Servo Motor Controller, 1 axis, RS-
Multi-Function Modules for PCD3
Communications Modules for PCD3
Memory Modules for PCD3
Expansion Modules PCD3 for PCD2/3
Expansion Modules PCD2 for PCD2/3
Digital I/O Modules for PCD1/2
Analogue I/O Modules for PCD1/2
Special Function Modules for PCD1/2
Multi-Function Modules for PCD2
1
2

Device

| Type | Description |
|---|---|
| PCD3.M5540 | CPU with 256/512/1024 KBytes RAM, 4 I/O slots (expandable), USB, Profi |

Ethernet Protocols

| Section | Description |
|---|---|
| IP Transfer Protocols | FTP, HTTP Direct Protocols, ODM. |
| IP Protocols | DHCP, DNS, SNTP, SNMP protocols. |

Memory Slots

| Slot | Type | Description |
|---|---|---|
| M1 | | |
| M2 | | |

Onboard Communications

| Type | Description |
|---|---|
| RS-485/S-Net | RS-485 port for Profi-S-Bus or general-purpose communications (D-Sub |
| USB | Universal Serial Bus port, PGU or general-purpose. |
| RS-232/PGU | RS-232, PGU or general-purpose serial port (D-Sub #1). |
| RS-485 | RS-485 port for general-purpose communications (Terminal block). |
| Ethernet | Ethernet port. |

Onboard I/O Slots

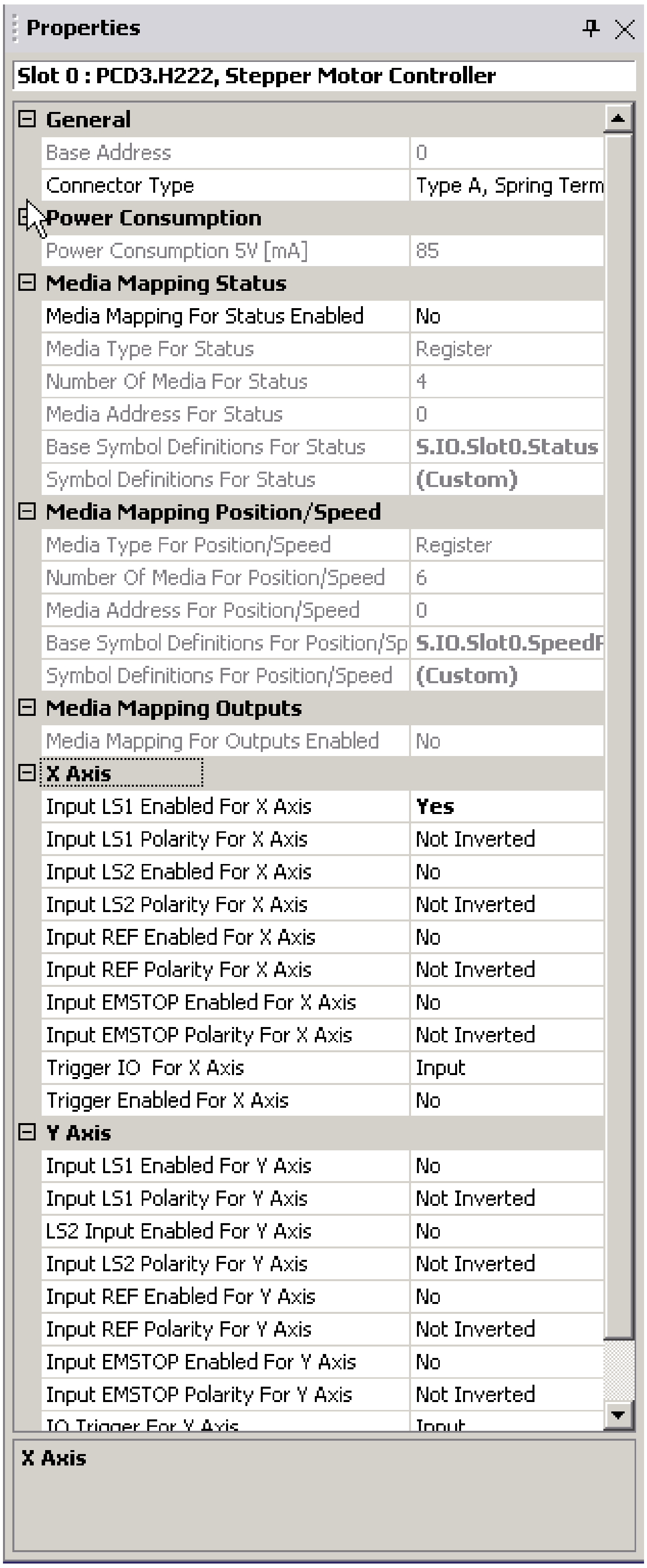
| Slot | Type | Description |
|---|---|---|
| Slot 0 | PCD3.H222 | 2 axis stepper motor controller, max. 20kHz, current draw 85m |
| Slot 1 | | |
| Slot 2 | | |
| Slot 3 | | |
| + | | |

*Device Configurator: Example using PCD3.*

## 4.3 Properties

The specified configuration is displayed here under the right-hand tab:

The operational sequence of both axes and the trigger function can be defined in this configuration. If a limit switch or a reference switch is required, it can be activated using these properties.



Properties

Slot 0 : PCD3.H222, Stepper Motor Controller

| | |
|---|---|
| ⊟ **General** | |
| Base Address | 0 |
| Connector Type | Type A, Spring Term |
| ⊟ **Power Consumption** | |
| Power Consumption 5V [mA] | 85 |
| ⊟ **Media Mapping Status** | |
| Media Mapping For Status Enabled | No |
| Media Type For Status | Register |
| Number Of Media For Status | 4 |
| Media Address For Status | 0 |
| Base Symbol Definitions For Status | **S.IO.Slot0.Status** |
| Symbol Definitions For Status | **(Custom)** |
| ⊟ **Media Mapping Position/Speed** | |
| Media Type For Position/Speed | Register |
| Number Of Media For Position/Speed | 6 |
| Media Address For Position/Speed | 0 |
| Base Symbol Definitions For Position/Sp | **S.IO.Slot0.SpeedP** |
| Symbol Definitions For Position/Speed | **(Custom)** |
| ⊟ **Media Mapping Outputs** | |
| Media Mapping For Outputs Enabled | No |
| ⊟ **X Axis** | |
| Input LS1 Enabled For X Axis | **Yes** |
| Input LS1 Polarity For X Axis | Not Inverted |
| Input LS2 Enabled For X Axis | No |
| Input LS2 Polarity For X Axis | Not Inverted |
| Input REF Enabled For X Axis | No |
| Input REF Polarity For X Axis | Not Inverted |
| Input EMSTOP Enabled For X Axis | No |
| Input EMSTOP Polarity For X Axis | Not Inverted |
| Trigger IO For X Axis | Input |
| Trigger Enabled For X Axis | No |
| ⊟ **Y Axis** | |
| Input LS1 Enabled For Y Axis | No |
| Input LS1 Polarity For Y Axis | Not Inverted |
| LS2 Input Enabled For Y Axis | No |
| Input LS2 Polarity For Y Axis | Not Inverted |
| Input REF Enabled For Y Axis | No |
| Input REF Polarity For Y Axis | Not Inverted |
| Input EMSTOP Enabled For Y Axis | No |
| Input EMSTOP Polarity For Y Axis | Not Inverted |
| IO Trigger For Y Axis | Input |

**X Axis**

## 4.4 Direct access instruction set

### 4.4.1 Write commands

| Value | Constant/variable: | Description: |
|---|---|---|
| B | IO.Slot0.IOAccess.X_AXIS_WR_STARTMP | Initialise and start the travel run:<br>0: Start the active run profile cw<br>1: Start the active run profile ccw |
| B | IO.Slot0.IOAccess.X_AXIS_WR_SINGLESTEP | Single step at Vmin:<br>0: Direction cw<br>1: Direction ccw |
| B | IO.Slot0.IOAccess.X_AXIS_WR_MOVEVCONST | Steps with constant velocity<br>0: Direction cw<br>1: Direction ccw |
| B | IO.Slot0.IOAccess.X_AXIS_WR_LSREFMOVE | Positioning on the limit switch or reference switch:<br>1: LS1<br>2: LS2<br>3: REF cw (search for LS1 or REF in direction ccw)<br>4: REF cw (search for LS2 or REF in direction ccw)<br>5:REF cw in invalid range in LS1 (ccw first)<br>6: REF cw in invalid range in LS2 (cw first)<br>7: REF cw (immediately in direction REF)<br>8: REF ccw (immediately in direction REF) |
| B | IO.Slot0.IOAccess.X_AXIS_WR_STOP | Stops the travel motion<br>(This instruction defines a STOP error in the error register which must be removed via CONT before the next motion instruction). |
| B | IO.Slot0.IOAccess.X_AXIS_WR_CONT | Clears a STOP error in the error register, which was caused by the STOP instruction. |
| W | IO.Slot0.IOAccess.X_AXIS_WR_VMIN | Load minimum frequency in Hz:<br>Range: 10…10,000 |
| W | IO.Slot0.IOAccess.X_AXIS_WR_VMAX | Load maximum frequency in Hz:<br>Range: 20…20,000 |
| W | IO.Slot0.IOAccess.X_AXIS_WR_ACC | Load acceleration in kHz/s:<br>Range: 1…1,000 |
| W | IO.Slot0.IOAccess.X_AXIS_WR_DEC | Load delay in kHz/s:<br>Range: 1…1,000 |
| W | IO.Slot0.IOAccess.X_AXIS_WR_JERKACC | Load jerk value in percent for acceleration:<br>Range: 0…50 |
| W | IO.Slot0.IOAccess.X_AXIS_WR_JERKDEC | Load jerk value in percent for delay:<br>Range: 0…50 |
| B | IO.Slot0.IOAccess.X_AXIS_WR_SAVEMP | Save run profile x (max. 16):<br>0: Profile 0<br>1: Profile 1<br>…<br>15: Profile 15 |
| DW | IO.Slot0.IOAccess.X_AXIS_WR_DESTREL | Load relative target in steps:<br>Range: 0…16777215 (3 bytes)<br>(Endless motion if the value 0xFFFFFF (16777215) is set) |

| | | |
|---|---|---|
| DW | IO.Slot0.IOAccess.X_AXIS_WR_DESTABS | Load target as absolute position:<br>Range: -2,147,483,648…<br>+2,147,483,647 (4 bytes)<br>(The relative target must be less than 16,777,215 steps).<br>WR_DESTABS must be followed by WR_MOVEVCONST or WR_STARTMP. WR_STOP, WR_SINGLESTEP and WR_LSREFMOVE will delete this instruction. |
| DW | IO.Slot0.IOAccess.X_AXIS_WR_ACTPOSABS | Define absolute position for current axis position:<br>Range: -2,147,483,648…<br>+2,147,483,647 (4 bytes) |
| B | IO.Slot0.IOAccess.X_AXIS_WR_IOCONF | Load input configuration:<br>Bit 0: Input LS1 on/off (0 = on/1 = off)<br>Bit 1: Input LS2 on/off<br>Bit 2: Input REF on/off<br>Bit 3: Input emergency stop on/off<br>Bit 4: Input LS1 mode<br>Input LS1 mode<br>(0 = open/1 = close contact)<br>Bit 5: Input LS2 mode<br>Bit 6: Input REF mode<br>Bit 7: Input emergency stop |
| B | IO.Slot0.IOAccess.X_AXIS_WR_TRIGCONF | Trigger Load input/output configuration:<br>Bit 0: Use TRIG as input (0)/ output (1)<br>Bit 1: Use TRIG<br>(0 = TRIG not used/1 = TRIG used) |
| B | IO.Slot0.IOAccess.X_AXIS_WR_ACTMP | Load run profile x (max. 16):<br>0: Profile 0<br>1: Profile 1<br>…<br>15: Profile 15 |

### 4.4.2 Read commands

| Value | Constant/variable: | Description: |
|---|---|---|
| B | IO.Slot0.IOAccess.X_AXIS_RD_IOSR | Bit 7: Reserved<br>Bit 6: Status of output MOTEN<br>0: Axis stopped<br>1: Axis in motion<br>Bit 5: Status of output DIR<br>0: Axis moving CW<br>1: Axis moving CCW<br>Bit 4: Status of input/output TRIG<br>Bit 3: Status of input EMSTOP<br>Bit 2: Status of input REF<br>Bit 1: Status of input LS2<br>Bit 0: Status of input LS1 |
| B | IO.Slot0.IOAccess.X_AXIS_RD_SR | Bit 7:2 Reserved<br>Bit 1: OnDest target marking<br>0: not at end position<br>1: End position reached<br>Bit 0: Wait for TRIG<br>0: Positioning underway, not started<br>1: Positioning underway if TRIG=1 |
| W | IO.Slot0.IOAccess.X_AXIS_RD_ERR_ WARN | Read access returns the last error and warning code from the module. After reading the code, an acknowledgment is sent. |
| DW | IO.Slot0.IOAccess.X_AXIS_RD_SPEED | Read access returns the current speed (Hz). |
| DW | IO.Slot0.IOAccess.X_AXIS_RD_RUN-TIME | Returns the current positioning time (from the start of the travel command) in ms (3 bytes). |
| DW | IO.Slot0.IOAccess.X_AXIS_RD_ACT-POSREL | Read access returns the number of steps made since the start of the travel command. (3 bytes) |
| W | IO.Slot0.IOAccess.X_AXIS_RD_ACT-POSABS | Read access returns the current position. (4 bytes) |
| W | IO.Slot0.IOAccess.X_AXIS_RD_VMIN | Returns the minimum frequency in Hz. |
| W | IO.Slot0.IOAccess.X_AXIS_RD_VMAX | Returns the maximum frequency in Hz. |
| W | IO.Slot0.IOAccess.X_AXIS_RD_ACC | Returns acceleration in kHz/s. |
| W | IO.Slot0.IOAccess.X_AXIS_RD_DEC | Returns delay in kHz/s. |
| W | IO.Slot0.IOAccess.X_AXIS_RD_JER-KACC | Returns acceleration in percent. |
| W | IO.Slot0.IOAccess.X_AXIS_RD_JERK-DEC | Returns delay in percent. |
| DW | IO.Slot0.IOAccess.X_AXIS_RD_DEST-REL | Returns the number of steps used for the travel commands (3 bytes). |
| DW | IO.Slot0.IOAccess.X_AXIS_RD_DE-STABS | Returns the absolute target position to be used for the next travel command (4 bytes). (4 bytes) |
| B | IO.Slot0.IOAccess.X_AXIS_RD_IOCONF | Configuration of inputs. (For description, see WR_IOCONF) |
| B | IO.Slot0.IOAccess.X_AXIS_RD_TRIG-CONF | Configuration of the TRIG connection. (For description, see WR_ TRIGCONFIG) |
| DW | IO.Slot0.IOAccesss. RD_MODULEFW-VERSION | Returns the firmware version in the format x.yy in ASCII. |

Additional information can be found in Section 4.3.2 “Read commands”.

! Depending on the wiring and the status of LS1 and LS2, LED 1 and LED 4 may light up orange.

# 5 Program adjustments when replacing H210 with H222

The most important differences in the program when switching from the H210 module to H222 are listed below:

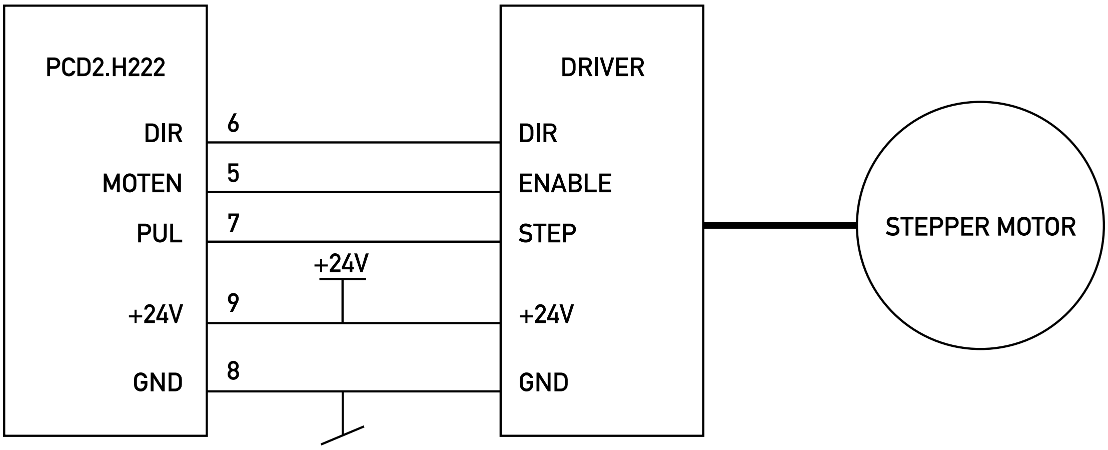
- H222.Init only initialises the module and not the run profile.
  → H222.Init and H222.InitMP are required to place the axes in motion.
- Selection of frequency range no longer required à Simplifying Init
- Vmin, Vmax and Acceleration no longer depend on the frequency range but are to be interpreted directly as frequency values
  → A conversion is required
- Limit, reference and emergency off switches must be activated by new commands
- Reference travel parameter is started with H222.Exec and the relevant reference parameter instead of H210.Home. H222.Exec commands and the number of parameters are not identical to commands for H210.Exec
  → all commands must be checked

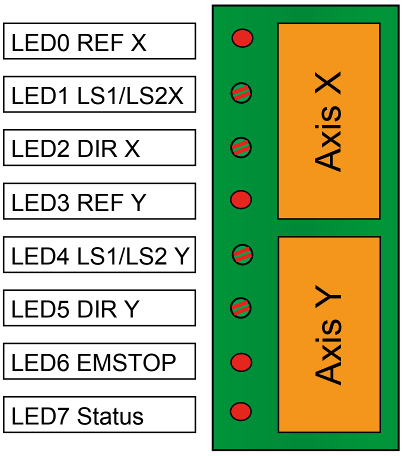
# 6 Examples

## 6.1 Hardware PCD2

- PCD2.H222
- PCD2.M5540
- Stepping motor end phase (driver)

This example demonstrates a simple application of the PCD2.H222 module:





**Axis X**

| | | |
|---|---|---|
| 0 | EMSTOP | LS1 |
| 2 | LS2 | REF |
| 4 | TRIG | MOTEN |
| 6 | DIR | PUL |
| 8 | OV | 24V |

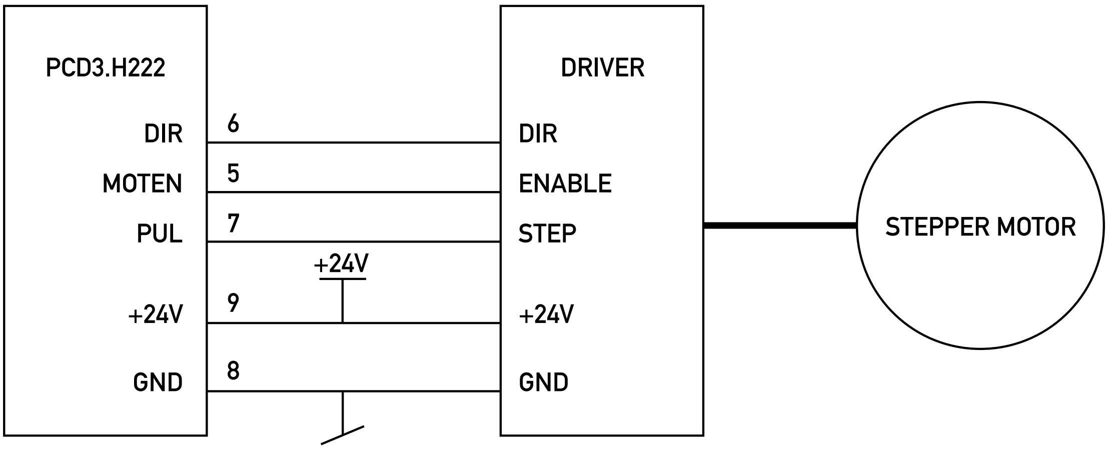
**Axis Y**

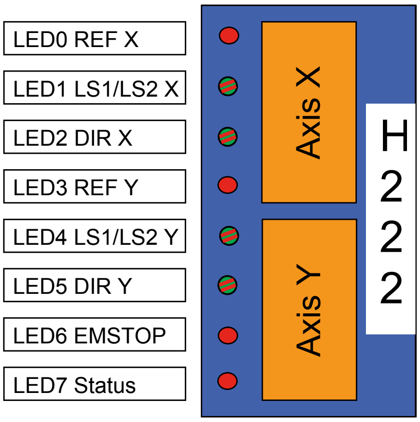
| | | |
|---|---|---|
| 0 | | LS1 |
| 2 | LS2 | REF |
| 4 | TRIG | MOTEN |
| 6 | DIR | PUL |
| 8 | OV | 24V |

## 6.2 Hardware PCD3

- PCD3.H222
- PCD3.M5340
- Stepping motor end phase (driver)

This example demonstrates a simple application of the PCD3.H222 module:





**Axis X**

| | | | |
|---|---|---|---|
| 0 | EMSTOP | LS1 | 1 |
| 2 | LS2 | REF | 3 |
| 4 | TRIG | MOTEN | 5 |
| 6 | DIR | PUL | 7 |
| 8 | OV | 24V | 9 |

**Axis Y**

| | | | |
|---|---|---|---|
| 0 | | LS1 | 1 |
| 2 | LS2 | REF | 3 |
| 4 | TRIG | MOTEN | 5 |
| 6 | DIR | PUL | 7 |
| 8 | OV | 24V | 9 |

## 6.3 Program description

In the first part of the IL program various registers are defined which are used in the program run.

The XOB 16 defines the values for a run profile and writes them to the H222 module in the IO slot 0.

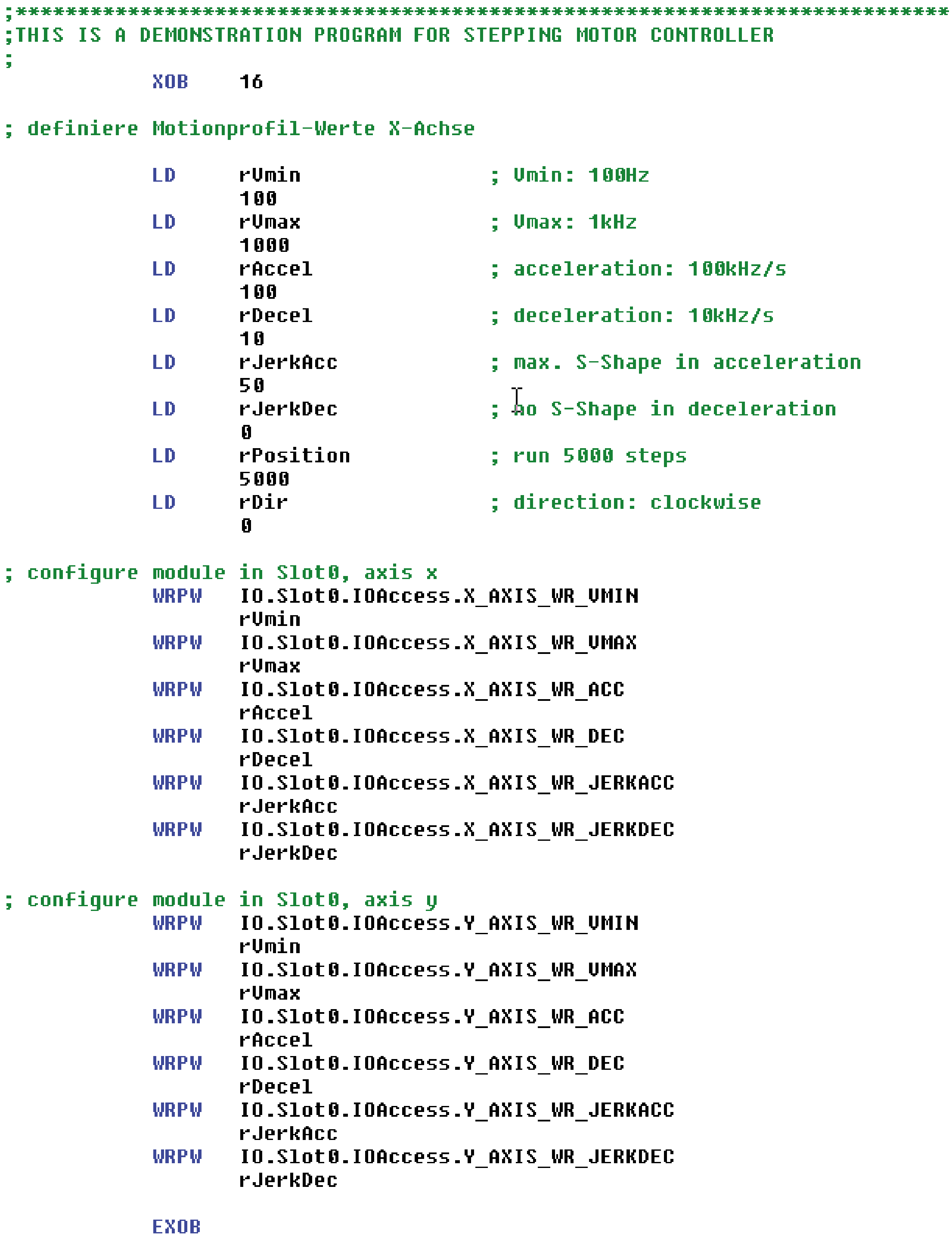
The main program, the COB 0, verifies whether the axis is at the end position. When this is the case, a new travel command is started with the predefined run profile, the number of steps in rPosition and the direction of rotation in rDIR.

## 6.4 IL program code

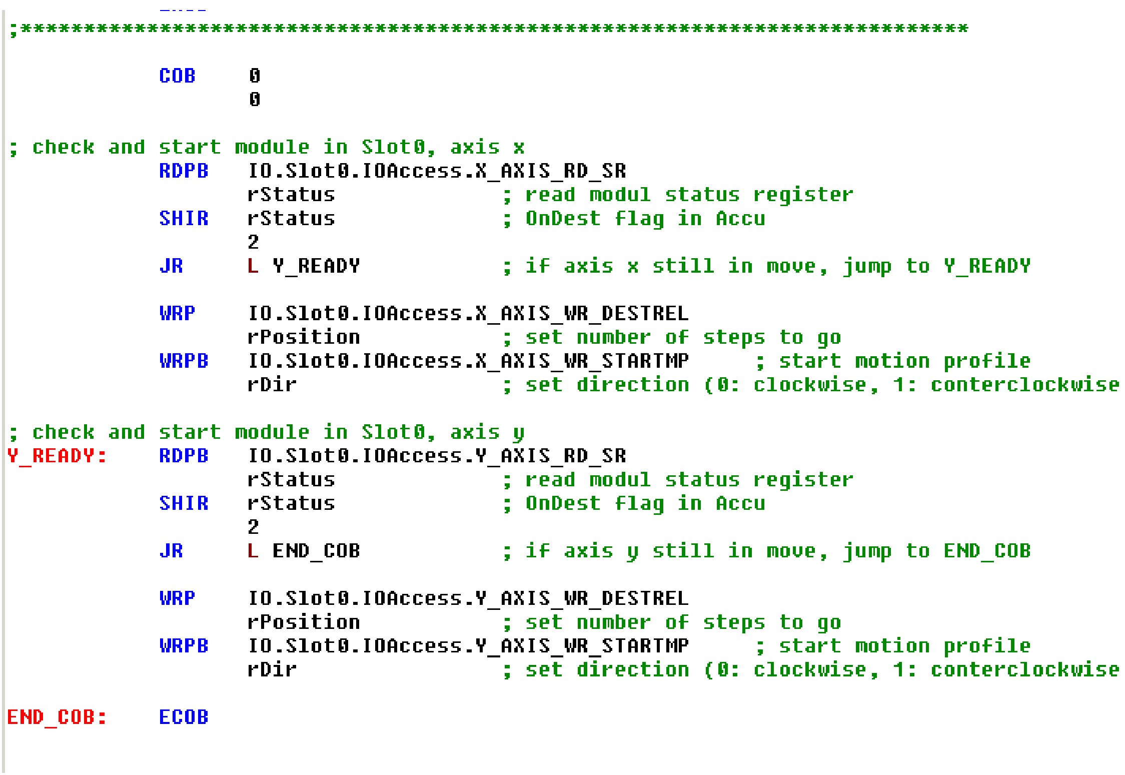
### *Definitions*

```
rVmin       EQU     R                       ; minimal speed
rVmax       EQU     R                       ; maximal speed
rAccel      EQU     R                       ; acceleration value
rDecel      EQU     R                       ; deceleration value
rJerkAcc    EQU     R                       ; jerk vaule for acceleration
rJerkDec    EQU     R                       ; jerk vaule for deceleration
rPosition   EQU     R                       ; number of steps to move
rDir        EQU     R                       ; 0=Vorwärts, 1=Rückwärts
rStatus     EQU     R
```

### *Initialisation*



```
;******************************************************************************
;THIS IS A DEMONSTRATION PROGRAM FOR STEPPING MOTOR CONTROLLER
;
            XOB     16

; definiere Motionprofil-Werte X-Achse

            LD      rVmin                   ; Vmin: 100Hz
                    100
            LD      rVmax                   ; Vmax: 1kHz
                    1000
            LD      rAccel                  ; acceleration: 100kHz/s
                    100
            LD      rDecel                  ; deceleration: 10kHz/s
                    10
            LD      rJerkAcc                ; max. S-Shape in acceleration
                    50
            LD      rJerkDec                ; no S-Shape in deceleration
                    0
            LD      rPosition               ; run 5000 steps
                    5000
            LD      rDir                    ; direction: clockwise
                    0

; configure module in Slot0, axis x
            WRPW    IO.Slot0.IOAccess.X_AXIS_WR_VMIN
                    rVmin
            WRPW    IO.Slot0.IOAccess.X_AXIS_WR_VMAX
                    rVmax
            WRPW    IO.Slot0.IOAccess.X_AXIS_WR_ACC
                    rAccel
            WRPW    IO.Slot0.IOAccess.X_AXIS_WR_DEC
                    rDecel
            WRPW    IO.Slot0.IOAccess.X_AXIS_WR_JERKACC
                    rJerkAcc
            WRPW    IO.Slot0.IOAccess.X_AXIS_WR_JERKDEC
                    rJerkDec

; configure module in Slot0, axis y
            WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_VMIN
                    rVmin
            WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_VMAX
                    rVmax
            WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_ACC
                    rAccel
            WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_DEC
                    rDecel
            WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_JERKACC
                    rJerkAcc
            WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_JERKDEC
                    rJerkDec

            EXOB
```



```
;***************************************************************************

            COB    0
                   0

; check and start module in Slot0, axis x
            RDPB   IO.Slot0.IOAccess.X_AXIS_RD_SR
                   rStatus             ; read modul status register
            SHIR   rStatus             ; OnDest flag in Accu
                   2
            JR     L Y_READY           ; if axis x still in move, jump to Y_READY

            WRP    IO.Slot0.IOAccess.X_AXIS_WR_DESTREL
                   rPosition           ; set number of steps to go
            WRPB   IO.Slot0.IOAccess.X_AXIS_WR_STARTMP     ; start motion profile
                   rDir                ; set direction (0: clockwise, 1: conterclockwise

; check and start module in Slot0, axis y
Y_READY:    RDPB   IO.Slot0.IOAccess.Y_AXIS_RD_SR
                   rStatus             ; read modul status register
            SHIR   rStatus             ; OnDest flag in Accu
                   2
            JR     L END_COB           ; if axis y still in move, jump to END_COB

            WRP    IO.Slot0.IOAccess.Y_AXIS_WR_DESTREL
                   rPosition           ; set number of steps to go
            WRPB   IO.Slot0.IOAccess.Y_AXIS_WR_STARTMP     ; start motion profile
                   rDir                ; set direction (0: clockwise, 1: conterclockwise

END_COB:    ECOB
```

# A Appendix

## A.1 Symbols

| | |
|---|---|
| i | This symbol refers the reader to more detailed information contained in this or another manual or in technical brochures.<br>Generally there is no direct link to these documents. |
| ⚡ | This symbol warns the reader of the risk of an electric discharge risk if the device is touched.<br>**Recommendation:** As a minimum precaution, always touch the negative terminal of the system (the housing of the PGU socket) before touching electronic components. It is preferable to connect yourself permanently to the negative terminal through an earthing strip around your wrist. |
| ! | This symbol appears next to mandatory instructions. |
| Classic | Explanations next to this symbol apply only to the Saia PCD® Classic series. |
| xx7 | Explanations next to this symbol apply only to the Saia PCD® xx7 series. |

## A.2 Address for Saia-Burgess Controls AG

**Saia-Burgess Controls AG**
Bahnhofstrasse 18
3280 Murten / Schwitzerland

| | |
|---|---|
| Telephone | +41 26 580 30 00 |
| Telefax | +41 26 580 34 99 |

| | |
|---|---|
| E-mail :........................................ | support@saia-pcd.com |
| Homepage : ................................. | www.saia-pcd.com |
| Support : ...................................... | www.sbc-support.ch |
| International branche offices &<br>SBC sales companies :.................. | www.saia-pcd.com/contact |

**Postal address for returns from customers of the Swiss sales office:**
only for products with Saia-Burgess Controls AG order numbers!

**Saia-Burgess Controls AG**
Service Après-Vente,
Bahnhofstrasse 18,
3280 Murten, Switzerland