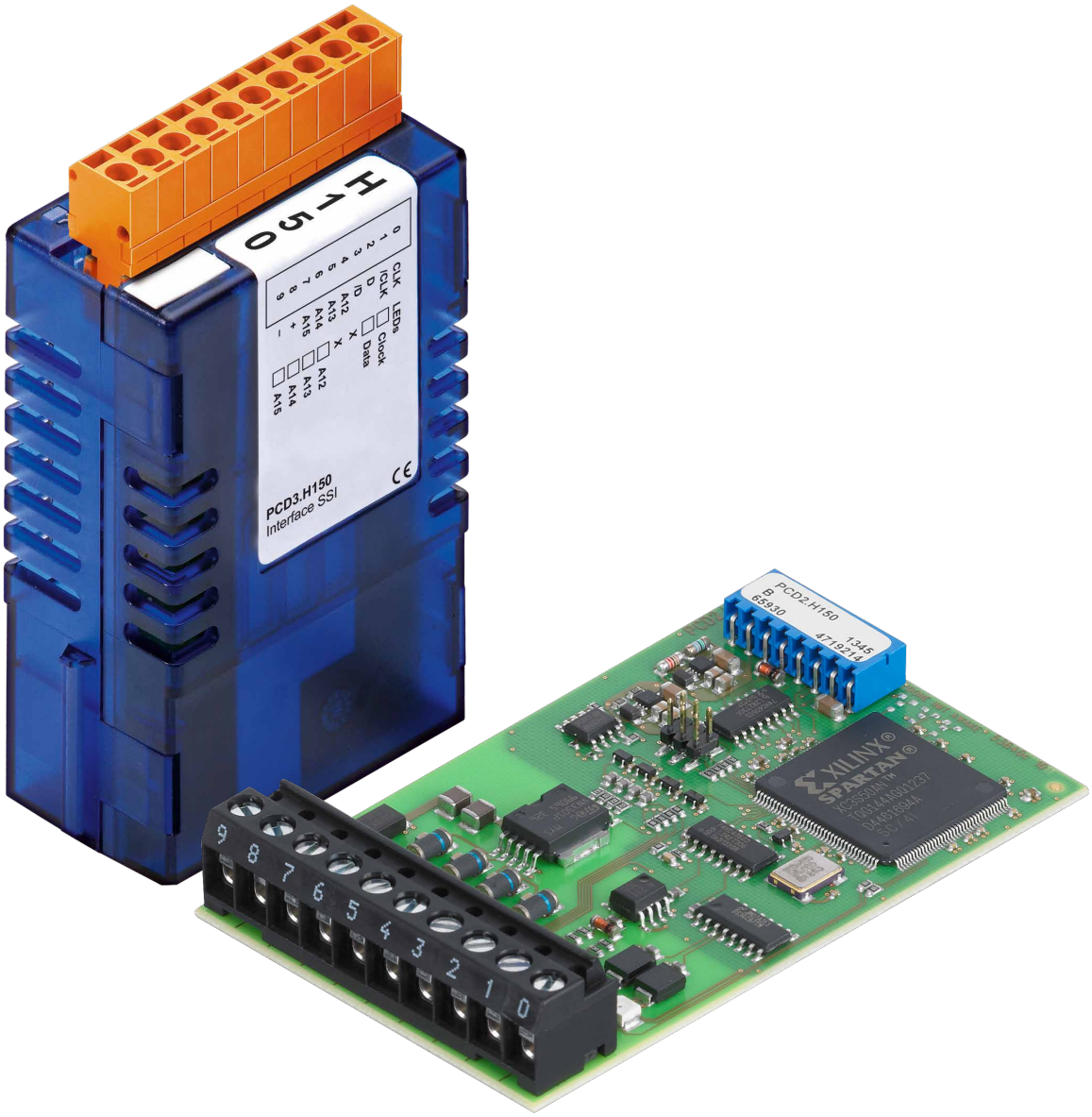


PCD2.H150 and PCD3.H150

SSI interface for absolute encoder

Document 26-761 | Edition ENG05 | 2019-08-02

0

0

0

0.1 Document History

Version	Published	Comments
EN01	-	Diverse internal versions
EN02	1999-05	First published edition
ENG05	2019-07-26 2019-08-02	- New logo and new company name - Deleted old Software - New Phone number - new chapter C “Hints”

0.2 About this manual

For some terms, abbreviations and source list used in this manual, see the appendix C.

0.3 Trademarks

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

Technical changes are subject to the state of technology.

Published in Switzerland

1 Introduction

1.1 General

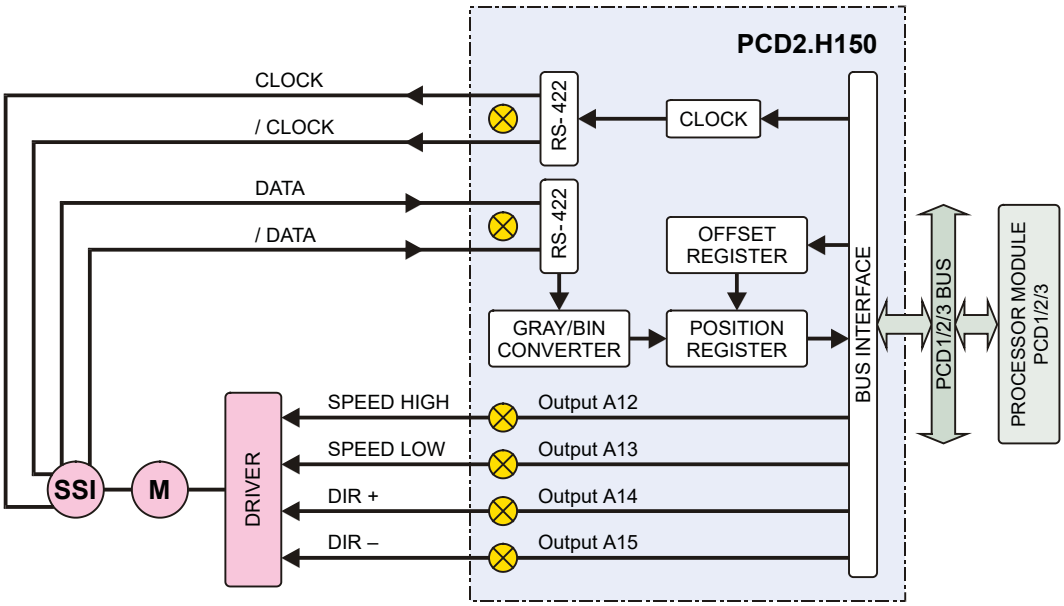
Function and benefits of SSI for absolute encoders

Incremental encoders do not provide absolute position information. In case of an emergency stop or power failure, a return to the home position is always necessary. Absolute transmitters using potentiometers provide sensitive analogue values and are subject to wear. Parallel absolute transmitters have the disadvantage of requiring multiwire connecting cable, which is liable to interference.

In 1984 the Max Stegmann GmbH company patented its synchronous serial interface (SSI) for absolute angle encoders, a method which made a good name for itself due to significant benefits in the field of positioning controllers:

- Only 4 × RS-422 lines required for clock and data transmission
- Secure data acquisition from the use of cyclic Gray code
- Galvanic isolation of the encoder from the controller by means of optocoupler
- The controller determines the timing for synchronization of data transmission from the encoder
- Possibility, in “ring-register mode”, of reading the SSI value twice and comparing
- Transmission speed up to 500 kbit/s

Block diagram of a standard motor drive



1.2 SSI-Interface PCD2.H150/PCD3.H150 for Saia PCD®

The modul PCD2.H150 may be plugged onto the I/O bus of any PCD1 or PCD2 (alone or as one of several). The PCD3.H150 module can be plugged into the I/O bus on any PCD3, even several times.

The modules PCD2.H150 and PCD3.H150 only differ in their design. The functions are identical. They are collectively named PCDx.H150 in this manual.

It offers an economical solution for direct connection to SSI encoders.

As can be seen from the block diagram, four general-purpose digital outputs are provided. These could be used for the selection of fast/slow or the direction of rotation.

With additional I/O modules from the extensive PCD2 | PCD3 range, the same controller can also be used to control all other machine functions.

1.3 Main features

The PCDx.H150 offers a low-cost, direct connection of SSI encoders whose high reliability is due to:

- 1 synchronous serial interface (1 × RS-422 for the clock output and 1 × RS-422 for the data input).
- Configurable baud rate (up to 500 kHz), configurable resolution (up to max. 29 bit) and configurable data format (Gray or binary code).
- Selectable read mode: single or double value with comparison of values
- 4 additional 0.5 A 24 VDC digital outputs with short-circuit protection
- Simple programming due to clearly laid out FBs (function blocks for IL) and FBoxes (function boxes for FUPLA).

1.4 Typical areas of use

Control of high-stacking warehouses

Here large distances have to be travelled which, with frequent returns to the home position, would be very time consuming. SSI encoders remove this disadvantage.

Hydraulically operated axes on presses

They are often provided with potentiometers whose value is registered through analogue inputs. SSI linear transmitters are remarkable for their special advantages of accuracy, reliability and useful life.

Punch machines with cam drive

Due to the accuracy, reliability and reading speed of the PCDx.H150 - and to the computing power of the PCD's CPU - an electronic camshaft is realized.

SSI angle encoders have proved their worth especially under harsh operating conditions

such as arise with robot arms, radar scanners or the adjustable screw blades of ocean-going freighters.

1.5 Programming

Thanks to practical function blocks, the user can simply enter the parameters desired for the various functions. These FBs (IL) and FBoxes (FUPLA)[1)] are available with the PG4 and PG5 programming software under Windows. Individual functions are described in detail in the relevant manual and complemented with practical application examples.

Initialization command

INIT
- selection of module number
- selection of number of data bits
- selection of number of control bits
- selection of clock frequency
- selection of code format
- input of offset position or register
- selection of read mode

Execute command

EXEC
- to read absolute position
- to set the actual position to zero
- to read the module identifier

Diagnostics and error handling

- FB ‘Timeout’ to recognize a cable break
- Some flags to recognize a cable break or a faulty encoder
- Recognition of incorrect addressing
- Recognition of wrong FB parameters and programming errors

1) In preparation

2 Technical data

2.1 Technical data of the hardware

SSI interface

1 input for SSI data	RS-422, galvanically isolated
1 output for SSI clock	RS-422, not galvanically isolated, as the encoder input is normally isolated

Digital outputs

Number of outputs	4, with addresses A12 to A15 for use as required from I/O bus
Switching capacity	0.5A each in range 10 ... 32 VDC, residual ripple max. 10%
Short-circuit protection	yes, Imax = 1.5 A
Galvanic isolation	no
Potential drop	max. 0.3 V at 0.5 A
Logic	positive (positive switching)
Output delay	typically 50 µs, max. 100 µs, at ohmic load

Power supply

Internal supply from PCD1/2 bus	5 VDC, 20…45 mA
External by user for all outputs	24 VDC (10…32 VDC), max. 2 A, residual ripple max. 10%

Operating conditions

Ambient temperature	
Operation:	0…+50°C without forced ventilation
Storage:	–25…+85°C
Störimmunität	CE mark according to EN 50 081-1 and EN 50 082-2

Ordering details

PCD2.H150	SSI interface module
PCD3.H150	SSI interface module
PCD9.H15E*	Software library with function blocks can be downloaded free of charge from our technical support page https://www.sbc-support.com/en/product-index/pcd2/hxxx-counting-and-motion/h1xx/

2.2 Electrical specification

Internal power consumption

+5V	20…45 mA
Uext	0…10 mA (without load current)

External power supply

Terminal +/–	10…32 VDC smoothed, residual ripple max. 10% TVS diode 39 V ±10% max. 2 A for outputs, not protected against wrong polarity!

SSI interface

RS-422 input with galvanic separation for SSI data D, /D.
RS-422 output without galvanic separation for SSI clock CLK, /CLK
(the clock is usually separated in the encoder).

Digital outputs

4 digital outputs (A12…A15) for universal use.

Addressing	direct from the bus
Output current	5…500 mA, short circuit protected (I_{short}=1.5 A max.)
Voltage range	10…32 VDC smoothed
Voltage drop max.	max. 0.3 V at 0.5 A
Output delay	typical 50 µs max. 100 µs (for resistive loads))

No galvanic isolation on the outputs

2.3 SSI function

To handle all the known SSI interfaces, the PCDx.H150 is configurable with several parameters.

Resolution	configurable 8 to 29 data bit 0 to 2 control bit
Clock frequency	configurable 100 kHz, 200 kHz, 300 kHz, 500 kHz (Input filter designed for 500 kHz) Frequency has to be selected depending on cable length:

Cable length	Frequency
< 50 m	max. 500 kHz
< 100 m	max. 300 kHz
< 200 m	max. 200 kHz
< 400 m	max. 100 kHz

Data code	configurable Gray or binary
Read mode	normal mode (sing read) ring mode: double read and compare (not all encoders support this function)
Offset position	An offset position can be defined when initialising the PCDx.H150 The defined offset is always subtracted in the FBs. The command ‹SetZero› also uses the offset register.
Execution time	typ. 1.5 ms for reading the SSI value
Cable break detection	detected with the FB ‹timeout› (10 ms)
Flags	‹fTimeout› (if cable break present, encoder faulty, or addressing incorrect) ‹fPar_Err› (if a wrong FB parameter is sent) ‹fRing_err› (if compare error in double read)‘)

2.4 Addressing

All functions are accessible via the FBs, so no direct addressing is necessary.

For user accessible I/Os the following EQUates are defined in the file D2H150_B.EQU:

```
OutShort_x EQU I 11+BA_x; output diagnostic (H if shorted)
Output12_x EQU O 12+BA_x; output12
Output13_x EQU O 13+BA_x; output13
Output14_x EQU O 14+BA_x; output14
Output15_x EQU O 15+BA_x; output15
```

‹x› is the module number, 'BA_x' is the base address which must be defined in the file D2H150_B.MBA.

See also chapter 6: Programming

Bus addresses

The PCD2H150 uses 16 addresses:

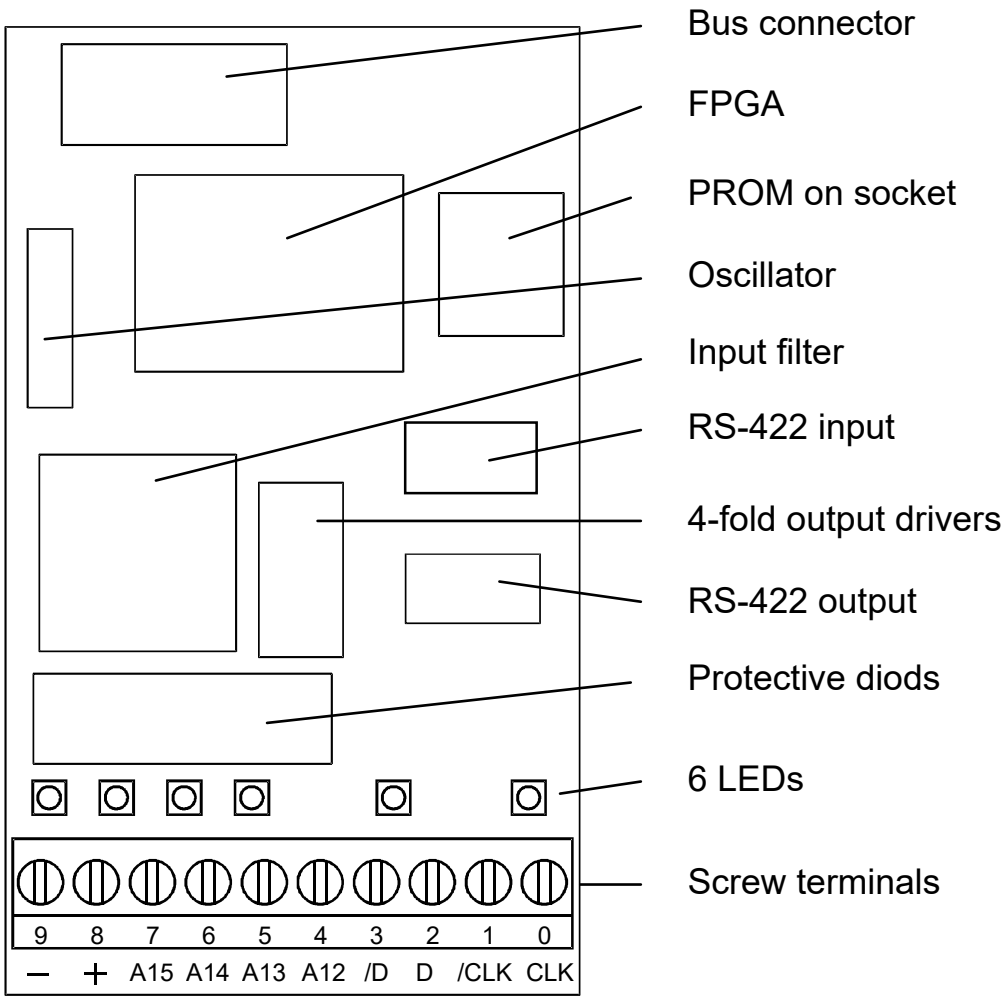
Outputs 0…11 and the Inputs 0…10 are only used by the FBs.

Outputs 12…15 can be accessed direct by the user.

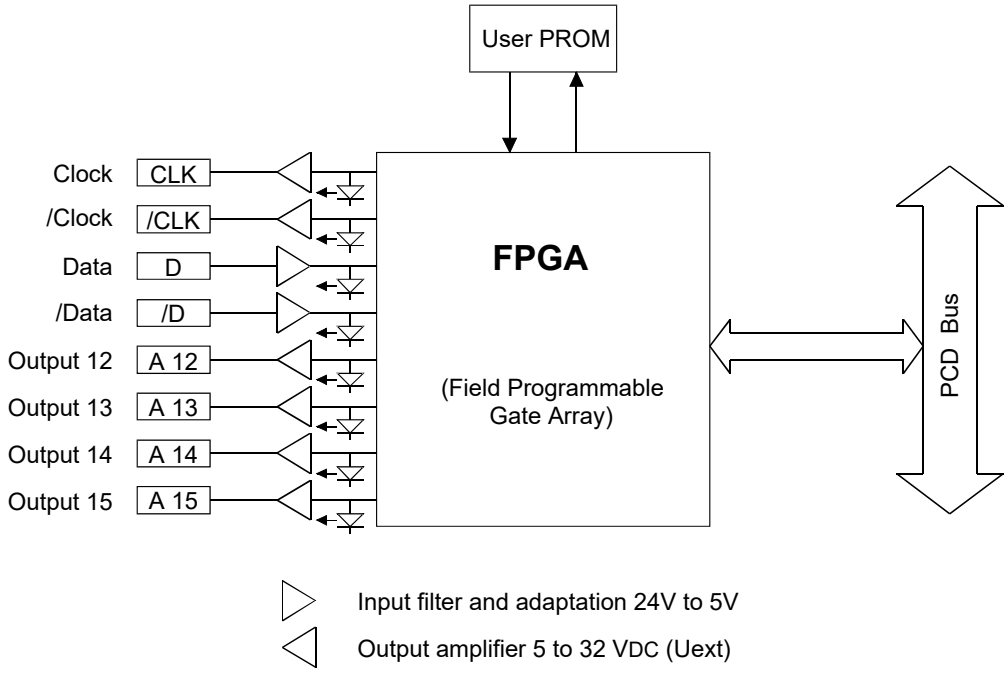
Address + BA	Data read (inputs)	Data write (outputs)
0	Data bit 0 (LSB)	Data bit 0 (LSB)
1	Data bit 1	Data bit 1
2	Data bit 2	Data bit 2
3	Data bit 3	Data bit 3
4	Data bit 4	Data bit 4
5	Data bit 5	Data bit 5
6	Data bit 6	Data bit 6
7	Data bit 7 (MSB)	Data bit 7 (MSB)
8	Data available (DA)	Write (WR)
9	SSI-Busy	Pointer address 0 (P0)
10	Ring-register error	Pointer address 1 (P1)
11	Output diagnostic	Pointer address 2 (P2)
12	Output 12 (actual state)	Output 12
13	Output 13 (actual state)	Output 13
14	Output 14 (actual state)	Output 14
15	Output 15 (actual state)	Output 15

3 Presentation

3.1 Assembled module



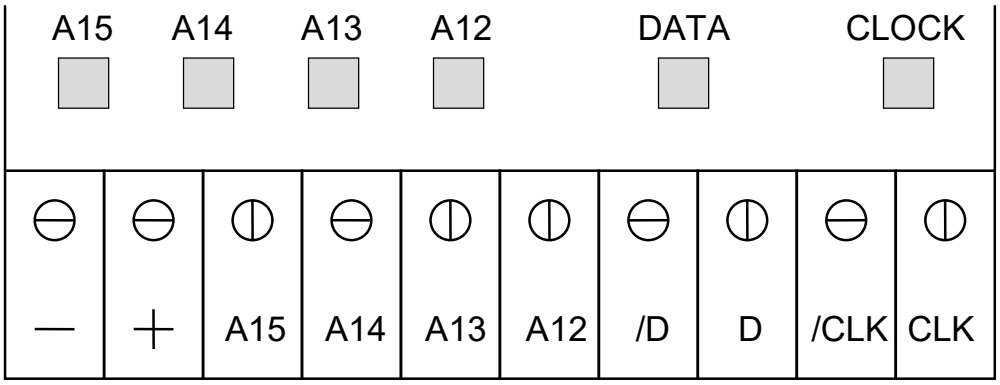
3.2 Simple logic diagram



4 Terminals, cable and meaning of the LEDs

4.1 Screw terminals

This picture shows the text on the printed circuit board. The I/O connector is standard from 0...9 (from right to left).



4.2 Inputs

Total	2	
Terminal 2 = D:	Data	position data (normal)
Terminal 3 = /D:	/Data	position data (inverted)

4.3 Outputs

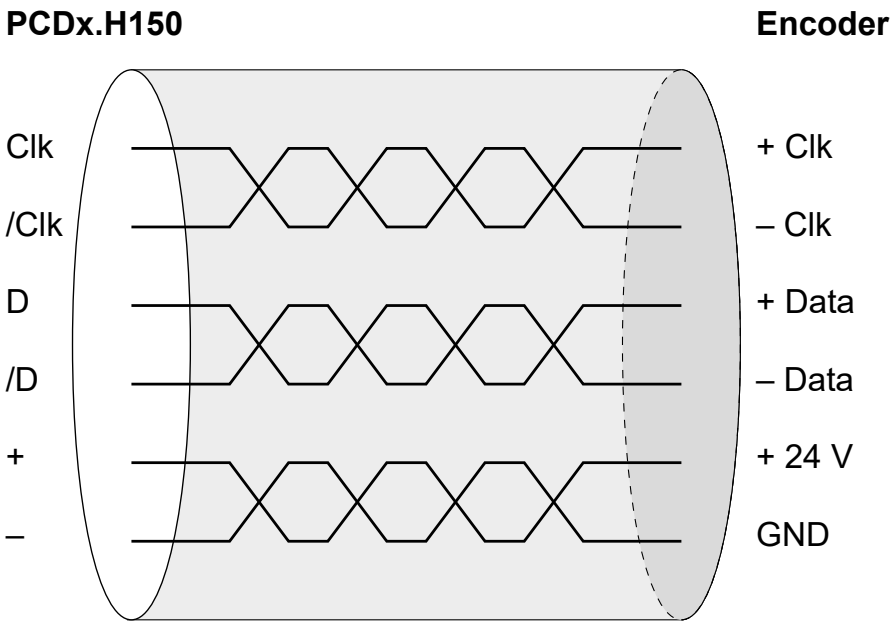
Total	6	
Terminal 0 = CLK:	Clock	normal
Terminal 1 = /CLK:	/Clock	inverted
Terminal 4 = A 12:	Speed high *)	
Terminal 5 = A 13:	Speed low *)	
Terminal 6 = A 14:	Dir +	positive direction *)
Terminal 7 = A 15:	Dir –	negative direction *)

4.4 Supply

Terminal 8 = +	+ 24 VDC
Terminal 9 = -	GND

*) or for general use

4.5 Connecting cables



To prevent interference problems, a shielded and twisted pair cable should be used and the shield should be connected to ground on both sides.

In exceptional situations, this arrangement may not be the best possible one. In such cases, the earthing plan must be discussed with a specialist (preferably the switchboard engineer).

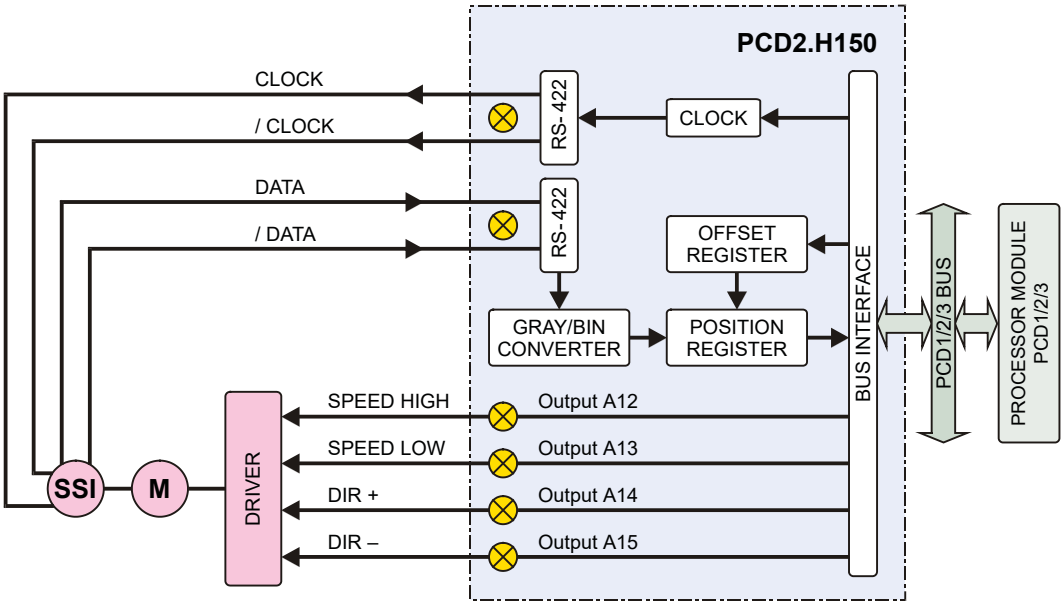
Meaning of the LEDs (see picture on previous page)

The module is equipped with 6 LEDs:

- 1 LED for the SSI output CLOCK, which flashes if a clock burst is sent to the absolute encoder
- 1 LED for the SSI input DATA, which flashes if the absolute encoder is sending position data. If the SSI interface works correctly, these two LEDs flash when the FB ‘Exec’ command: ‘RdPosition’ is executed.
  The LED pulse for CLOCK and DATA is extended to 5 ms so that it is visible. This means that both LEDs are constantly on if the read period in the user program is shorter that 5 ms.
- 4 LEDs show the state of the 4 digital outputs A12…A15.
  If one output (or more) is in a short circuit condition, the LEDs of outputs which are H blink with a frequency of 6 Hz.

5 Functional description

5.1 Block diagram of the module



As the block diagram shows, the SSI encoder is capable of direct connection.

In addition, 4 digital outputs are at the user's disposal. They can be used to select any desired functions, such as direction of rotation or fast/slow speed.

5.2 Module description

One or more PCD2.H150 modules can be plugged onto the I/O bus of any PCD1 or PCD2 device.
One or more PCD3.H150 modules can be plugged onto the I/O bus of any PCD3.

The module PCDx.H150 offers an economical solution for the direct connection of SSI encoders.

Moreover, by using further I/O modules from the extensive PCD2 and PCD3 range, all other machine functions can be controlled by the same program.

5.2.1 Offset equalization

In the INIT FB or Fbox, a constant offset can be entered to adjust the absolute zero position. This offset equalizes the transmitting device's absolute deviation which resulted from assembly. Every time the controller is powered up, this value is taken into account so that the transmitting device's position corresponds to an absolute value. The offset is always subtracted from the absolute position. If an offset is to be added, a negative value should be entered.

5.2.2 'SetZero' FB or FBox

In contrast to offset equalization, the 'SetZero' function can be used to declare a current position as the zero position. In this case, parameter 6 of the 'Init' FB must be written with register 'roffset_x'. This register must not be overwritten by the user.

5.2.3 Compare on absolute position

With SSI encoders it is advisable to compare the absolute position for 'higher than the reference' or 'lower than the reference'. (The positive or negative flag should be used after a compare function).

If you want to compare for 'position = reference' (zero flag) the read interval (depending on the user program but min. 1.5 ms) should be higher than the max. position change interval (depending on resolution and motion velocity) to guarantee that the reference position is detected.

5.3.4 Short-circuit protection

The 4 outputs are protected against short circuits.

If one output (or more) is shorted, the current is internally limited to max 1.5 A.

If the chip temperature rises above 150 °C (which is reached in about 1 to 2 sec), the output driver detects the over-temperature, switches all 4 outputs off and sets the diagnostic flag.

If the temperature drops below 150 °C the driver tries to switch the outputs on again.

The short circuit condition is displayed by blinking of the output LEDs

6 Programming

The programming of the PCD for the use of the counting and positioning modules PCD2.H150 and PCD3.H150 is carried out via the PCD user program by means of the standard programming tools «PG4» from version 2.0. (There are separate FBs for using the older programming tool «PG3»).

Programming with PG5 is identical, but other directories must be used.

Programming is either in IL (instruction list) with FBs (function blocks) or in FUPLA with FBoxes (in preparation). The FBs can be obtained using reference PCD9.H15E which are downloable from our web-site www.sbc-support.com.

Since motion control tasks always concern sequential processes, it is preferable if user programs are written in GRAFTEC, while individual steps and transitions can be edited either in IL with FBs or in FUPLA with FBoxes. User programs, however, can also be written purely in BLOCTEC or FUPLA.

7 Error handling and diagnostics

7.1 Definition errors checked by the assembler

The following definition errors in file D2H150_b.MBA are checked during assembly:

```
$IF NbrModules < 1; if NbrModules = 0 no EQUates are defined,
```

error message is:

```
$REPORT Remark: No H150 used (NbrModules = 0 in D2H150_B.MBA)
```

```
$IF NbrModules > 16; if NbrModules > 16 no EQUates are defined,
```

the error message is:

```
$REPORT Error: more than 16 Modules H150 defined (NbrModules =
0...16)
```

If an invalid command code is used for the FB ‹Exec› (e.g. ‹RdIdenti› instead of ‹RdIdent›) the assembler shows an error message:

```
«Symbol not defined H150.RdIdenti»
```

If the ‹$group H150› is missing, the assembler shows error messages:

```
«Symbol not defined for each command»
```

7.2 Error handling in run

7.2.1 Wrong parameters

In the FB ‘Init’, parameters are checked to confirm that they are in the valid parameter range.

In the FB ‘Exec’, only the command code is checked. Parameter 1 (module no.) and parameter 3 (register) are not checked because this would slow down the execution time.

If a parameter is outside the valid range it is set to the minimum value, the error flag **‘fPar_Err’** is set and the diagnostic register **‘rDiag’** is loaded with an error code.

The flag ‘fPar_Err’ and ‘rDiag’ are not reset by the FBs. This has to be done by the user program (e.g. in XOB 16).

The error is coded as follows:

```
rDiag 31......24 23......16 15......8 7......0
      \reserve / \ FB No. / \Par. No./\Modul No./
                           (Init   = FB 1)
                           (Exec = FB 2)

 FB Exec Ringregister error     = Errorcode 5 (in 2nd byte)
                 Timeout error  =     ”     6      ”
```

Example: If the clock frequency (parameter 4) in FB ‘Init’ of module 2 is wrong (> 3), the ‘rDiag’ register is loaded with the code 00 01 04 02 hex

The diagnostic register always contains the last error. The effective address of ‘rDiag’ is listed in the file ‘project.map’.

If a parameter is outside the valid range, it is set to the minimum value before it is sent to the **PCDx.H150**.

7.2.2 Short circuit protection

The 4 outputs are protected against short circuits.

If one output (or more) is shorted, the current is internally limited to max 1.5 A.

If the chip temperature rises over 150 °C (which is reached in about 1 to 2 sec), the output driver detects the over-temperature, switches all 4 outputs off and sets the diagnostic flag.

If the temperature drops below 150 °C the driver tries to switch the outputs on again.

The short circuit condition is indicated by blinking of the output LEDs.

7.3 Timeout

If the PCDx.H150 cannot be accessed (Busy always H or DA always L), the FB ‘Exec’ with the commands ‘RdPosition’ or ‘SetZero’ is quit after a timeout of 10 ms and the flag ‘fTimeout’ is set H. This flag should be checked in the user program after each read to guarantee that the read position is correct and the encoder is working.

The following errors can be detected with the timeout :

- SSI module not fitted (or wrong base address)
- SSI module faulty
- Absolute encoder faulty
- Absolute encoder not connected or not powered
- Cable break on SSI data line or encoder supply

If the SSI clock line alone is broken and all the other lines are O.K, the PCDx.H150 cannot detect this error. In this case the maximum position value is read (depending on the resolution).

For generating the FB timeout, the 1 ms system counter is used. For a PCD2.M1xx you have to use the firmware version <$ 2C> (July 1995) or later. The PCD1, PCD2.M480, PCD2.M5xx0 and PCD3 supported this feature from the beginning.

7.4 Interference problems

In very noisy environments the SSI encoders can be desynchronised (by mistaking an interference spike for a clock edge) since the input filters of the encoders are very small (designed for clock frequencies up to 1 MHz).

Proper installation is very important:

- use shielded twisted-pair cables for the SSI (RS-422) interface
- connect both sides of the shield to ground (if no potential differences are present)
- separate the SSI cable from power cables
- use ring-register mode (if supported by the encoder)

If the encoder supports ring-register mode, this method allows interference immunity to be greatly increased.

In **ring-register** mode the PCDx.H150 outputs the specified number of clocks twice and the encoder sends the SSI value twice. The two positions are compared in the PCDx.H150.

If they are equal, the position is assumed to be correct and is stored in the register defined in par. = 3.

If the two values are not equal, the FB repeats the read and compare. If after 3 double reads the values are still not equal, the FB sets the flag ‘fRing err’ and loads the diagnostic register with an error code (see chapter 7.2: Error handling). In this case either the transmission is subject to serious interference or the PCDx.H150 has been initialized with incorrect encoder parameters (e.g. wrong number of data bits).

In ring-register mode, the flag ‘fRing_err’ should be checked after each ‘RdPosition’ to guarantee that the position is correct.

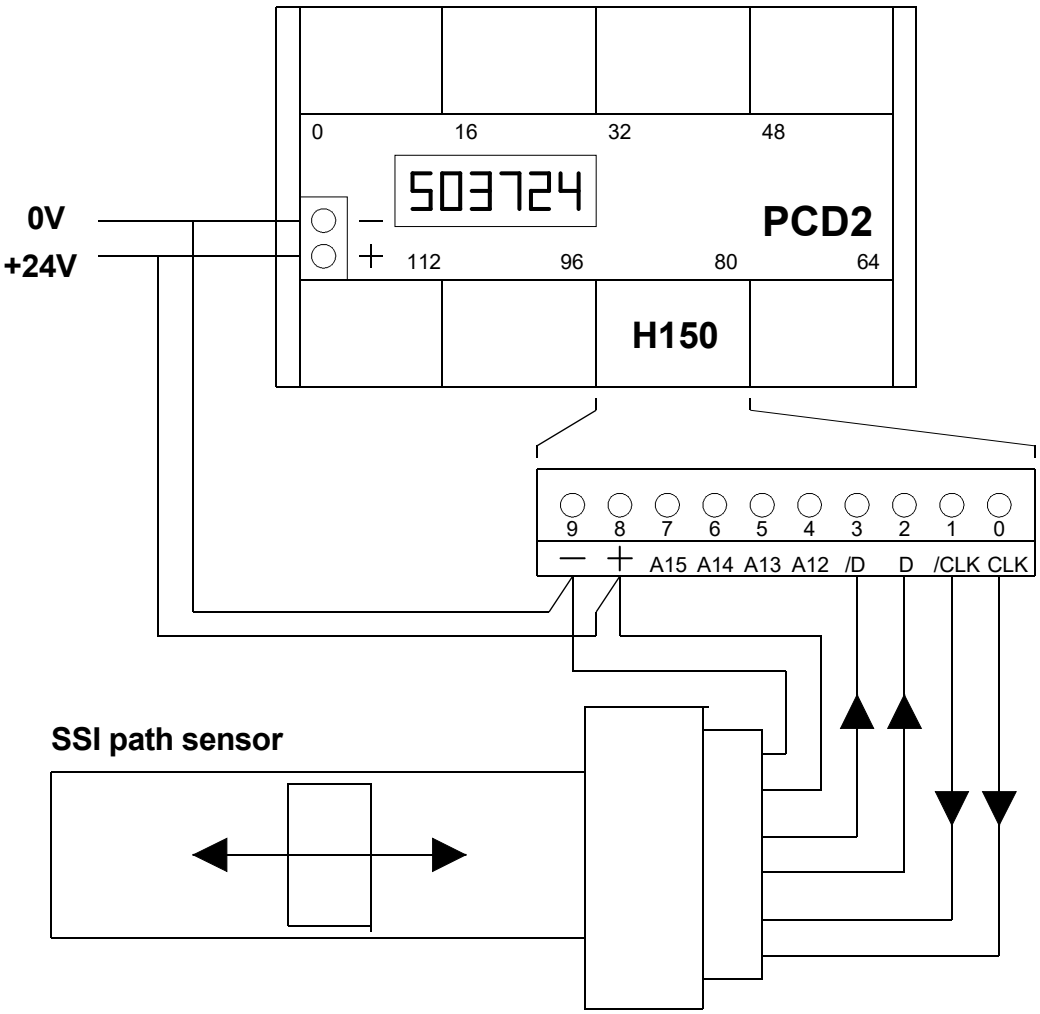
7.5 Error overview

possible error	diagnostic mechanism	flag or register
cable break: SSI data or supply	Time out	‹fTimeout› = H
cable break: SSI clock	No	‘fTimeout’ = H
wrong parameter input	Check the command code in FB EXEC	‘fParErr’ and ‘rDiag’
wrong SSI values	no, ring mode to be used	
SSI module or encoder faulty	Timeout	‘fTimeout’ = H

8 User example

8.1 User example in IL with FBs

Minimum arrangement to operate absolute positioning using an SSI absolute encoder.



The individual elements are:

- PCD2 (or PCD1) equipped with at least 1 PCD2.H150
  (1 PCD2.F510/530)
- SSI absolute encoder, e.g. Temposonics LP, type PA-S-1000M-D70-1-S2G6-1
- Power supply 24 VDC smoothed

The SSI sensor should have the following data:

- Number of data bit: 24
- Number of control bit: 0
- Clock frequency: 100 kHz
- Code: Gray
- Offset position: 0
- Operating mode: normal

The PCD2.H150 module is plugged into socket 6 (base address 80).

Task of the program

When the PCD is powered up, the absolute position of the SSI path transmitter should be displayed on the display module or by the debugger.

The program is called “intro.src” and should be located in project “h150e”.

In addition to the main task of the program (to read absolute position) the correct operation of the module is monitored. Moreover, the 4 outputs are involved and have the following functions:

- Output 12 = H, if position is < Pos. 1 (1000) ist,
- Output 13 = H, if position lies between Pos 1 and Pos. 2,
- Output 14 = H, if position is > Pos. 2 (5000) ist.
- If an error is detected in module function, all 4 outputs are = H.

With PCD input 0 = H, any required position can be declared as the zero position. However, it is doubtful whether this function has any practical significance as, in this case, even a “normal” (cheaper) incremental shaft encoder could be used, whereby the zero point would have to be redefined by homing each time the controller is restarted. However, this is exactly what the absolute encoder was intended to avoid.

This zero position should not be confused with a fixed offset position, such as can be used for the absolute adjustment of individual transmission devices. This correctional value is defined as the “rOffset” parameter in the initialization FB and applies for the entire lifetime of that transmitter.

For this demonstration program, the “wait100ms” FB can be left as it is. In practice, i.e. when other parts of user programs are running, this FB would slow down the program in an unreliable way and ought not to be structured in this way. It would be preferable to pack the whole H150 example into one FB and then call that FB every 100 ms.

The example is contained under the file name “H150_Ex2.src” in the PCD9.H15E software package and can be put into operation after the base address has been adapted in the “d2h150_b.mba” file.

```
;*********************************
;** Example program for PCD2.H150 **
;*********************************
; Author :  L.Riedo  21.04.99
; Filename: H150_Ex2.src
;
; This program reads the absolute position with the H150.
; The actual position is compared with two reference positions
; Pos1 and Pos2 (defined as constants).
; The 4 outputs on the module H150 are set depending on the
; compare result:
;       output 12 = H if position < Pos1
;       output 13 = H if Pos1 < position < Pos2
;       output 14 = H if position > Pos2
;
; All outputs (12 to 15) are set H in case of an error condition
; (ex. encoder not connected)
;
; System requirements:
; 1 H150 on the base address defined in the file D2H150_B.MBA
; 1 Input module on base address 0
; optional : 1 display module

$include D2H150_b.EQU     ; user and FB specific definitions
$group H150

;---- user resources (dynamic resource allocation) ----
InputZero   EQU    I 0    ; base address of input module

rPosition   EQU    R       ; register for SSI value
rOffset     EQU    R       ; register for offsetposition
rSignature  EQU    R       ; register for Identification
fDyn        EQU    F       ; flag for dynamisation
WAIT100ms   EQU    FB      ; wait FB
SSI_ERR     EQU    FB      ; error handling FB

;---- constants ----
Pos1        EQU    1000    ; compare position 1
Pos2        EQU    5000    ; compare position 2

;*************************************************************

XOB    16                 ; startup block

LD     rOffset
       0                  ; No offset value.
                          ;  If the old offset position is to be used,
                          ;   the register rOffset_x should be defined
                          ;   as offset position (par. 6) in the FB INIT
```

8

```
;---- initialisation according to the encoder specification ----
CFB    INIT              ; initialize module
     K 1                 ;  module nr. (K 1...16)
       24                ;  no. of databit (8...29)
       0                 ;  control bit (0...2)
       2                 ;  frequency 0,1,2,3 (100k, 200k, 300k, 500kHz)
       0                 ;  code (0=gray, 1=bin)
       rOffset           ;  offset position
       0                 ;  mode (0=normal, 1=Ringmode)

CFB    Exec              ; FB Exec
     K 1                 ;  module nr
       RdIdent           ;  command: read module identification
       rSignature        ;  target register

DSP    rSignature        ; Display the signature of the module
                         ;  H150 : 24xx (xx is FPGA program version)

CFB    WAIT100ms         ; wait time of 100 ms

EXOB                     ; end of startup block


;******************************
;*** M A I N  P R O G R A M  ***
;******************************

COB    0                 ; cyclic block
       0                 ;  0 = no monitoring

;---- set position to 0 ----
STH    InputZero         ;
dyn    fDyn              ; flag for dynamisation
CFB  H Exec              ; call FB if InputZero is H
     K 1                 ;  modul nr
       SetZero           ;  command: set position to 0
       rNotused          ;  no parameter used

;---- read position ----
CFB    Exec              ; Execution FB
     K 1                 ;  module nr
       RdPosition        ;  command: read position
       rPosition         ;  target register

*) DSP rPosition         ; display the position
                         ;  (if a display is equipped)

STH    fTimeout          ; timeout ?
ORH    fRing_err         ; Ringregister error ?
CFB  H SSI_ERR           ; call SSI error FB if Errorflag = H
JR   H endcomp           ; no position compare if error
```

```
;----- position < Pos1 ? --------
CMP    rPosition          ; absolute position
       Pos1               ;  compare position 1
ACC  N                    ; set accu depending on the compare result
OUT    Output12_1         ; set output 12 on H150 if Pos1 not reached
RES    Output13_1         ; and reset output 13
RES    Output14_1         ; and reset output 14
RES    Output15_1         ; and reset output 15

;----- position > Pos2 ? --------
CMP    rPosition          ; absolute position
       Pos2               ;  compare position 2
ACC  P                    ; set accu depending on the compare result
OUT    Output14_1         ; set output 14 on H150 if Pos2 reached
RES    Output12_1         ; and reset output 12
RES    Output13_1         ; and reset output 13
RES    Output15_1         ; and reset output 15
JR   H endcomp

;--- Pos1 < position < Pos2 ? ---
CMP    rPosition          ; absolute position
       Pos1               ;  compare position 1
ACC  P                    ; set accu depending on the compare result
OUT    Output13_1         ; set output 13 on H150 if Pos1 reached
RES    Output12_1         ; and reset output 12
RES    Output14_1         ; and reset output 14
RES    Output15_1         ; and reset output 15

endcomp:
CFB    WAIT100ms          ; wait time of 100ms

ECOB                      ; end of cyclic block

;************************************************************

FB     SSI_ERR            ; user error handling

SET    Output12_1
SET    Output13_1
SET    Output14_1
SET    Output15_1

EFB
```

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

FB     WAIT100ms

LD   T 0
       1
wait: STH  T 0
      JR   H wait

      EFB

$endgroup                ;(H150)
```

*) If an attempt is made to display figures larger than 999’999 (or -99’999) on the screen, the display switches off, the error flag is set and, unless XOB 13 has been programmed, the CPU’s error LED is also activated. To avoid unnecessary complication of the demonstration program, it omits any programming routine to prevent the display of excessively large values.

A Appendix

A.1 Symbols

This symbol refers the reader to further information in this or another manual or in technical information brochures.
There is generally no direct link to these documents.

This symbol appears next to instructions that require strict compliance.

This symbol warns the reader of the risk of electric discharge upon contact.
Recommendation: You should at least touch the negative terminal (PGU port casing) on the system before coming into contact with the electronic components.
It is preferable to be permanently connected to an earthing lug on the wrist with the negative terminal.

The explanations next to this symbol are valid for the Saia PCD® Classic series only.

The explanations next to this symbol are valid for the Saia PCD® xx7 series only.

A

A.2 Abbreviations

SSI — Synchronous Serial Interface (SSI) is a widely used serial interface standard for industrial applications between a master (e.g. controller) and a slave (e.g. sensor). SSI is based on RS-422 standards and has a high protocol efficiency in addition to its implementation over various hardware platforms, making it very popular among sensor manufacturers. SSI was originally developed by Max Stegmann GmbH in 1984 for transmitting the position data of absolute encoders – for this reason, some servo/ drive equipment manufacturers refer to their SSI port as a "Stegmann Interface". It was formerly covered by the German patent DE 34 45 617 which expired in 1990. It is very suitable for applications demanding reliability and robustness in measurements under varying industrial environments.

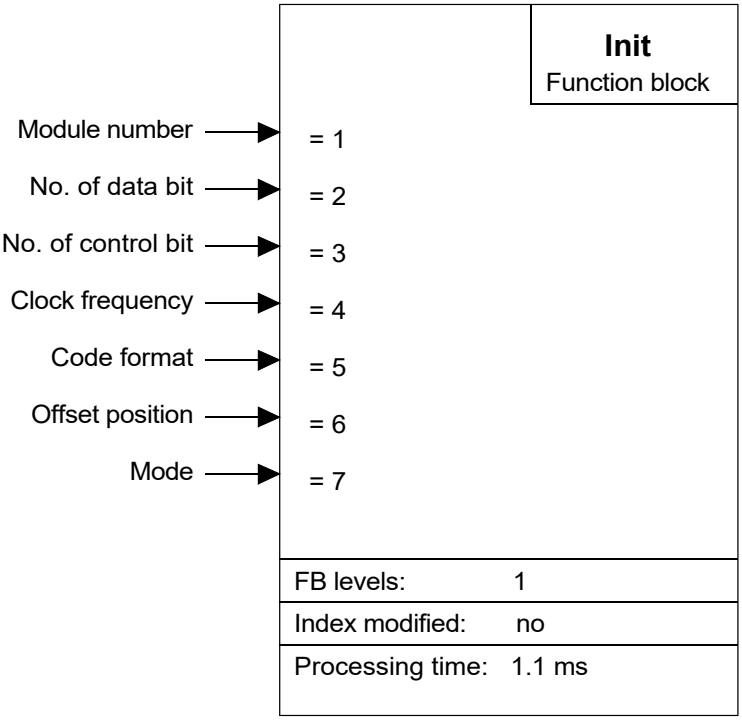
Soucre: https://en.wikipedia.org/wiki/Synchronous_Serial_Interface

B Summary of all software elements for programming in IL

B.1 Function block 'Init'

Init

FB: Initialization of PCDx.H150 module



B

Function description:

With this FB the settings of the PCDx.H150 module are defined and the base address is read from the D2H150_B.MBA file.

Description of I/Os:

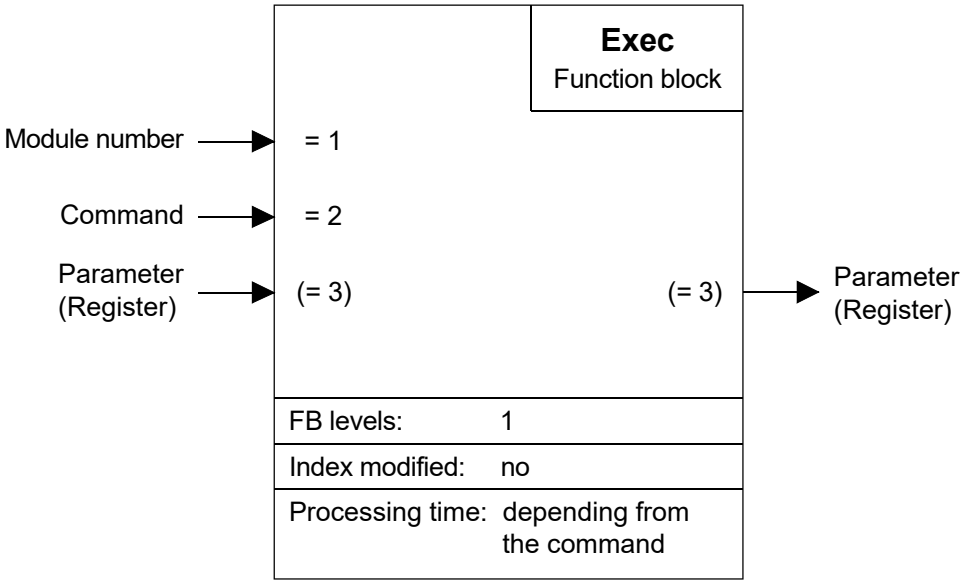
Par.	Designation	Type	Format	Value	Comment
= 1	Module number	K	integer	0 - 16	0 – 16
= 2	Number of data bit		integer	8 - 29	8 – 29
= 3	Number of control bit		integer	0 - 2	0 – 2
= 4	Clock frequency		integer	0 - 3	0 = 100 kHz, 1 = 200 kHz, 2 = 300 kHz, 3 = 500 kHz
= 5	Code from encoder		integer	0 - 1	0 = gray → bin, 1 = bin
= 6	Offset position	R	integer	0 - 231 - 1	R0 – R4095 / rOffH15_x
= 7	Ring mode *)		integer	0 - 1	0 = normal, 1 = Ring mode

*) In 'ring mode' the number of data bits and the number of control bits or empty bits (if present) must be set correctly according to the encoder's data sheet. If not, the 'fRing_err' flag will always be set, because reading the second (comparative) position value will have shifted.

B.2 Function Block 'Exec' (Execute command)

Exec

FB: Execute command of PCDx.H150 module



Function description:

This FB is used to execute commands for the PCDx.H150 module.

The module number (Parameter 1) must be a constant (k 1 ... k 16).

The base addresses and the number of modules are defined in the file D2H150_B.MBA.

The FBs support max. 16 x PCD2.H150 modules per PCD system.

The commands (parameter 2) are described on the following pages.

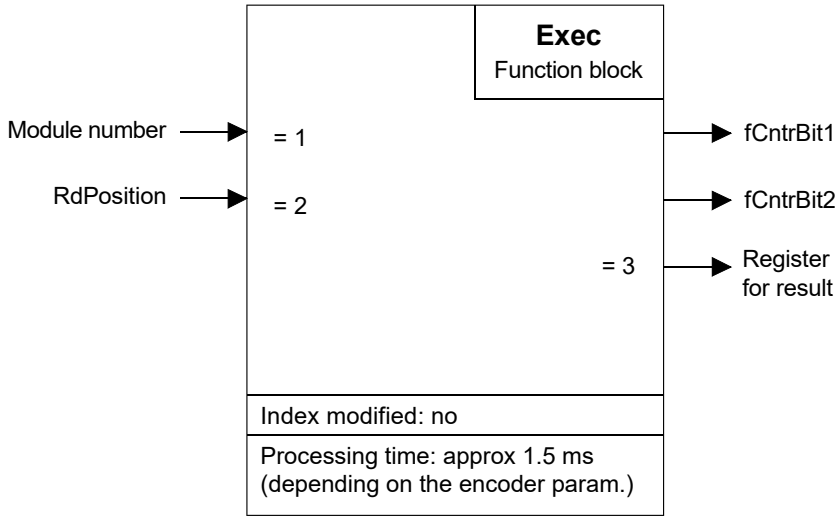
Parameter 3 defines the register where the position (command 'RdPosition') or the signature (command 'RdIdent') are stored.
If no register is used (command 'SetZero'), any register or the defined register 'rNotUsed' can be used as parameter 3.

Par.	Description	Type	Format	Value
= 1	Module no.	K		1 – 16
= 2	Command code		Symbol	
= 3	Parameter	R		0 - 4096

B.3 Commands for FB 'Exec' (FB parameters)

RdPosition

Command: Read absolute position



Function description:

The command 'RdPosition' is used to send the SSI clock and store the read position in the register defined in parameter 3.

If control bits are defined (max. 2 bits after the position value) they are removed from the position data and copied to the specific flags 'fCntrBit1_x' and 'fCntrBit2_x' ('x' is the module number).

Note: If the data is sent in gray code, the control bits are also converted (gray → bin) and are falsified if they were already in binary code.

B

Ring register

In ring mode (configured in FB Init) the PCDx.H150 reads the same SSI position twice and compares both values. If they are equal, the position is assumed to be correct and is stored in the register defined in par. =3.
If the two values are not equal, the FB repeats the read and compare. If after 3 double reads the values are still not equal, the FB sets the flag 'fRing_err' and loads the diagnostic register with an error code (see chapter 8.2: Error handling). In this case either the transmission is subject to serious interference or wrong encoder parameters have been initialized.
(See also FB Init).

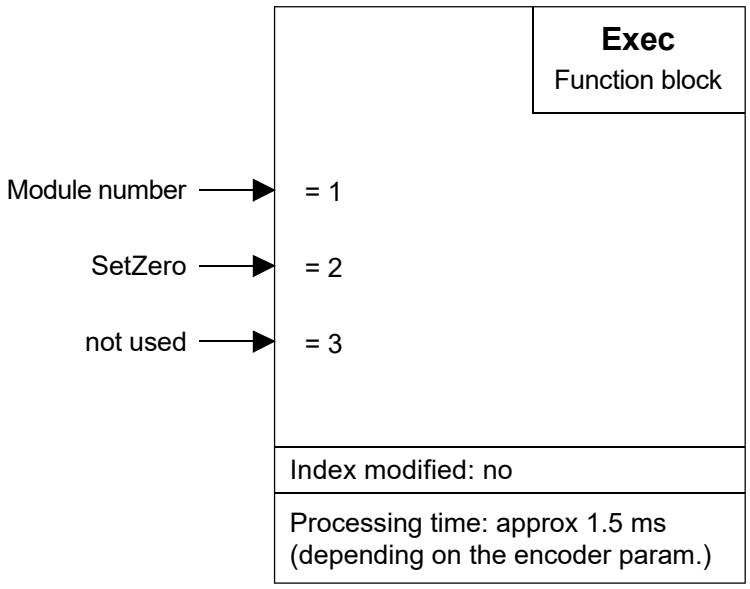
Note: Not all encoders support this mode.

Description of I/Os:

Par.	Description	Type	Format	Value
= 1	Module number			1 - 16
= 2	Command: RdPosition			
= 3	Destination register	R	Integer 8 – 29 Bit	0 - 4095

SetZero

Command: Set actual position to zero



Function description:

With the command 'SetZero' the actual position is defined as 0. This takes two steps:

1. The actual position is read without considering the offset position.
2. The read position is stored in the specific offset register of the addressed module.

The next time the command 'RdPosition' is executed, it returns 0 as position.

The function 'SetZero' uses the same offset register than the FB 'Init' (Parameter 6).

If the axis is moved in the negative direction after a 'SetZero' command, the next 'RdPostion' will return a negative position value, which normally does not exist with absolute encoders.

If this offset value is to be used always, i.e. even after starting up again, register 'roffset_x' must be given as parameter 6 to FB 'Init'. In this case, the register must not be overwritten.

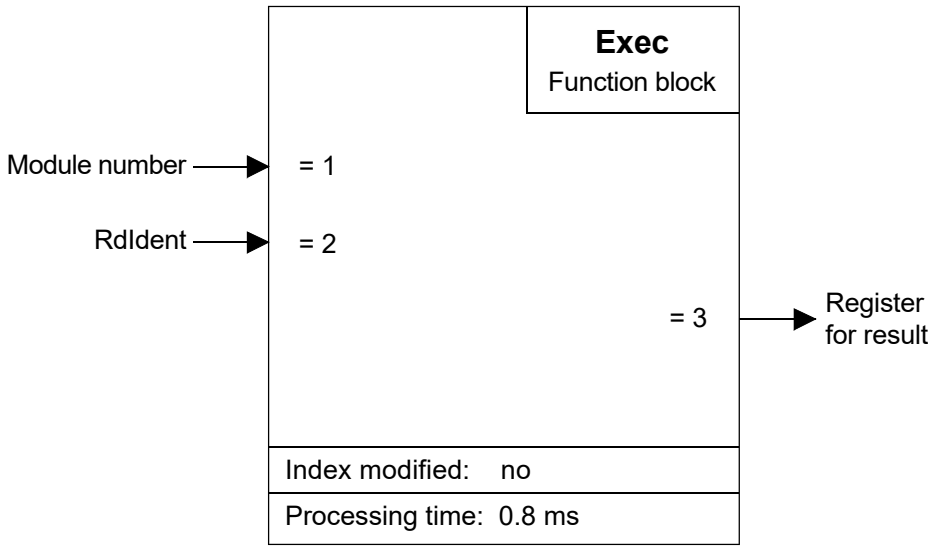
To cancel this position offset, the register 'roffset_x' must be loaded with zero. Another possibility is to provide parameter 6 of the 'Init' FB with a register which has been loaded with zero, and then process it.

Description of I/Os:

Par.	Description	Type	Format	Value
= 1	Module number	K		1 - 16
= 2	Command: SetZero			
= 3	Empty PCD register	R		0 - 4095

RdIdent

Command: Read the module identification



Function description:

This command can be used to check the correct function of the PCDx.H150 module and verify the FPGA version. If the module works correctly, the value 24xx will be read (see table below). If the module is faulty (or the addressing is wrong), the value 0 is read.

Description of I/Os:

Par.	Description	Type	Format	Value
= 1	Module number			1 - 16
= 2	Command: RdIdent			
= 3	Module identification	R	Integer 12 Bit	0 - 4095

Table of identification codes for PCDx.H150:

Value	FPGA version
2400	Version HE0
2401	Version HE1
...	...
2415	Version HEF

C Informations

C.1 General informations

Attention

These devices must only be installed by a professional electrician, otherwise there is the risk of fire or the risk of an electric shock.

Warning

Product is not intended to be used in safety critical applications, using it in safety critical applications is unsafe.

Warning - Safety

The unit is not suitable for the explosion-proof areas and the areas of use excluded in EN61010 Part 1.

Warning - Safety

Check compliance with nominal voltage before commissioning the device (see type label). Check that connection cables are free from damage and that, when wiring up the device, they are not connected to voltage.

Note

In order to avoid moisture in the device due to condensate build-up, acclimatise the device at room temperature for about half an hour before connecting.

Cleaning

The device can be cleaned in dead state with a dry cloth or cloth soaked in soap solution. Do not use caustic or solvent-containing substances for cleaning.

Maintenance

These devices are maintenance-free. If damaged during transportation or storage, no repairs should be undertaken by the user.

Guarantee

Opening the module invalidates the guarantee.

C

C2. Standards and certificates

WEEE Directive 2012/19/EC Waste Electrical and Electronic Equipment directive
The product should not be disposed of with other household waste. Check for the nearest authorized collection centers or authorized recyclers. The correct disposal of end-of-life equipment will help prevent potential negative consequences for the environment and human health.

EAC Mark of Conformity for Machinery Exports to Russia, Kazakhstan or Belarus.

C

D Contact, support and repair addresses

D.1 Contact

Saia-Burgess Controls AG
Bahnhofstrasse 18
3280 Murten, Switzerland

Telephone switchboard ..... +41 26 580 30 00
Telephone SBC Support ..... +41 26 580 31 00
Fax ..... +41 26 580 34 99

D.2 Support

E-mail Support: ..... support@saia-pcd.com
Support site: ..... www.sbc-support.com
SBC site: ..... www.saia-pcd.com

International representations &
SBC sales companies: ..... www.saia-pcd.com/contact

D.3 Repair

Postal address for customers to return products in Switzerland:

Saia-Burgess Controls AG
After sales service
Bahnhofstrasse 18
3280 Murten, Switzerland

D