Sample project PCD2-3 H150 Example

Content

Project history

Date	Author	Modification
23.04.2007	C. Durrer	Creation of documentation (version 1) and adaptation of project for PG5 1.4.130 and PCD3.M5540

1. Summary

.

1.1 Functional description

This sample project is intended to show it is possible to

- Use a PCD2.H150 or PCD3.H150 for reading the position of an SSI absolute encoder (cyclical reading).
- Use the FBs for the H150 modules in XOBs for event-driven reading of the position.

All those functionalities are programmed in Instruction List (IL) with use of the FBs for the H150 modules. These FB's are included in the PG5 controls suite and only need to be included in the files.

1.2 Hardware and software used

Hardware:

PCD:	PCD3.M5540	Firmware 037
Input module:	PCD3.E110	Mounted in slot 0
H150[1]:	PCD3.H150	Mounted in slot 1
SSI Endoder	Adapt the parameters in the initialisation part of the program to the specification of your SSI encoder	

Minimal software versions:
SAIA PG5 SP1.4.130

[1] The PCD2.H150 and the PCD3.H150 can be addressed identically with the same FBs. The hardware is also identical except of the terminals and the housing.

2. Procedure before programming

2.1 Preparing hardware

The connections are to be installed according to the manual 27/761. It might be that the description of the different wires is not on all manufacturers' description the same. However, the following colour code should fit most cases:

SSI signal	Wire colour	PCD2/3.H150 terminal
+T or /CLK	Green	1
-T or CLK	Yellow	0
+D or /D	Gray	3
-D or D	Pink	2
0 V	White	9
+U	Brown	(8) in case of 24V supply of encoder

2.2 Configuring the PCD

The configuration of the PCD is not really relevant in this example. You can easily change the hardware (by e.g. using a PCD2). The only thing you have to do is adapting the hardware- and software settings in PG5.

2.3 Configuring the base address of the H-modules

In case you prefer not to place the input card on slot 0 and the H150 on slot 1, please adapt the following:

- The symbol InputZero to the base address of your input module
- Change the following line of the file D2H150_B.MBA:

```
BA_1        EQU    16       ;Base address of module 1
```

by adapting the address 16 to the base address of the H150.

3. Functional description of the example files

3.1 H150_ReadCyclically.src

This program reads the absolute position using the PCD2/3.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 high in case of an error condition (ex. encoder not connected). The position can be set to 0 by pressing input 0.

This zero position is not written to the SSI encoder but is rather an offset that is added (on the PCD) each time the value is read.

Program structure

This file consists of 3 program blocks.

- Initialisation block (XOB 16)
- Main Program (COB 0)
- Error FB (called each time an error occurrs)

XOB 16
In the XOB 16 the FB H150.INIT is called. This FB configures the H150 according to the parameters passed. In this FB e.g the frequency of the SSI clock can be selected.
Once the H150 is configured (initialized), its signature (firmware version) is read and written to the register H150.rSignature.

COB 0
In the Cyclic Organisation Block 0 (COB 0), the position is written to 0 (on a rising edge of InputZero). This is established by calling the FB H150.EXEC with the Command SetZero:

```
CFB     H Exec              ; call FB if InputZero is H
        K 1                 ; module nr (according to *.mba)
        SetZero             ; command: set position to 0
        rNotused            ; no parameter used
```

The following call of the FB EXEC contains the command “RdPosition” reads the position from the SSI encoder. The read value (stored in register “rPosition”) is later on compared against the limit positions Pos1 and Pos2.

FB SSI_ERR
By checking the state of the flag H150.fTimeout and H150.fRing_rr after the call of the command “RdPosition”, an eventual error would be detected (in COB 0). If this is the case, the FB SSI_ERR is called. In this FB actions can be taken (e.g. a stop of the machine or similar). In this example code, all the outputs of the H150 are set high.

3.2 H150_ReadOnEvent.src

Also this program reads the absolute position using the PCD2/3.H150. The difference is that the position is not read cyclically but on event (if an external interrupt is triggered). Instead of executing the FB's for reading the position in a COB, they are placed in XOB 20 and XOB 21 (triggered by the external interrupts Int0 and Int1).

The functionality of the program remains mainly the same, except that it is only possible setting the position to 0 and reading the values if an external interrupt is triggered.
This behaviour could be required in case the position of a specific event (indicated by an interrupt) is to measured as precisely as possible.

Note that it is not allowed to execute the FBs for the motion modules in the same program from the COBs as well as from XOB's. The reason is that in this case it would be possible that an executed FB from the COB could be interrupted by its call from an XOB!

3.3 Sources

This document is an extract from chapter 9 in manual 26/761 (Manual for PC2.H150).