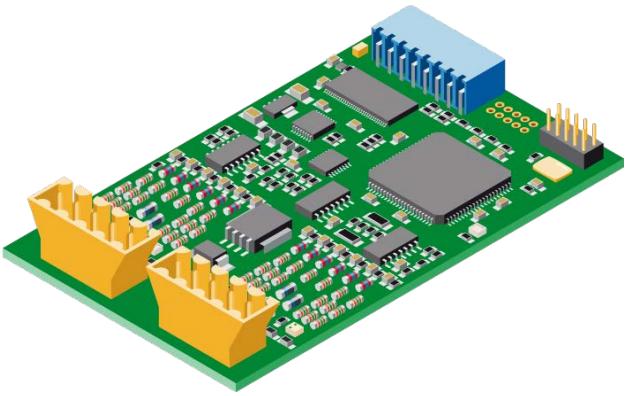
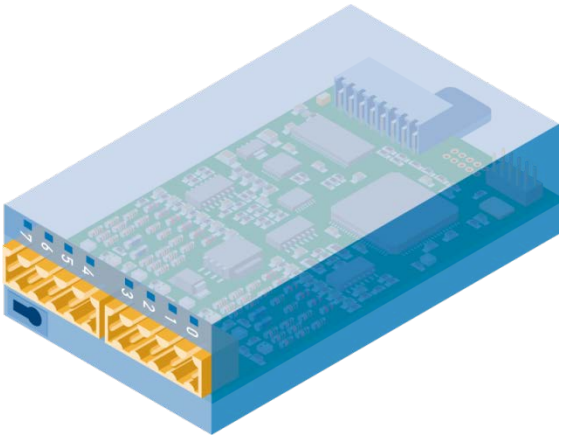
Manual:





PCD2/3.H112 Fast counting module with 2 channels
PCD2/3.H114 Fast counting module with 4 channels

Table of contents

Document versions

Version	Publication	Changes	Comments
EN01	2013-07-29		First version of manual
ENG02	2015-06-02		New phone number

Trademarks

Technical modifications and changes depending on state of the art.

1 Introduction

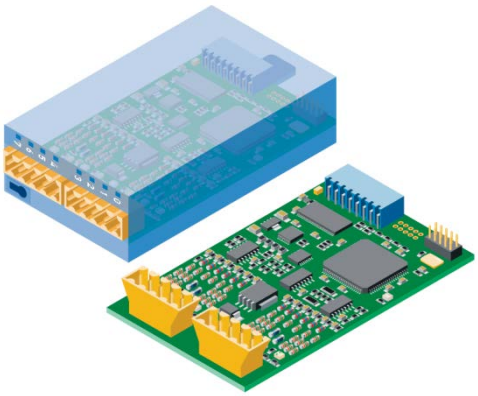
1.1 Purpose of this document

The purpose of this document is to provide useful information on, and describe how to configure and access, the fast counting modules PCD2/3.H112/4.

1.2 Functions and applications

This universal module enables counting functions up to 150 kHz for PCD2 and PCD3 controllers. The PCD and the fast counting module communicate via the I/O bus.

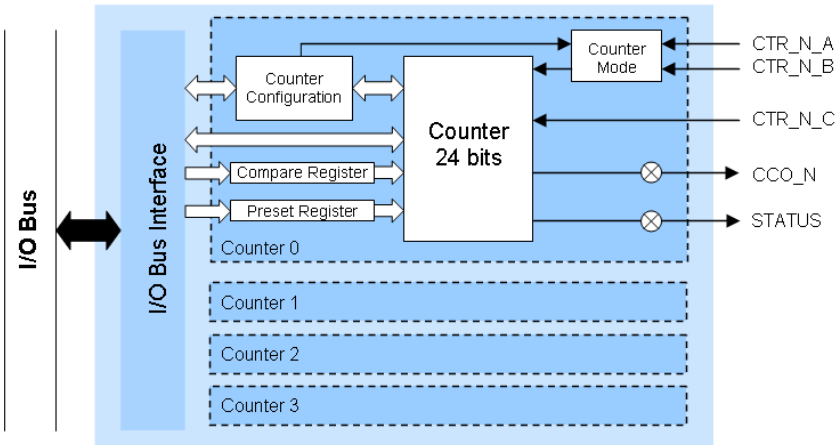
The module is suitable for counting revolutions, distances, volumes, etc. and for measuring by counting pulses. It has two inputs, A and B, and, in count modes ×1, ×2 or ×4, recognises the direction in which incremental shaft encoders turn. Input A and B are suitable for the connection of encoders used to count up/down automatically. The counter is loaded with a start value. The counter flag can be used to select a counting direction (up or down). A software enabler is used to start/stop the counter. The trigger signal can be used to record and read an intermediate value. With a preset signal, a count value can be provided and, in each case, loaded onto the counter.



At the end of a count, the counter's directly controlled output (CCO) can be used, for example, to trigger precise external switch operations or to release an interrupt. The CCO output is set/reset via the CCO flag.

1.3 System overview

Following schematic diagram provides a system overview.



1.4 System requirements

The fast counting module functions require the following system versions:

- **PG5 version 2.1.200** or higher
- **PCD firmware version 1.22.10** or higher

1.5 Main features

- 2 (H112) or 4 (H114) counter module
- 1 Counter Controlled Output (CCO) per counter
- 2 inputs – A and B – per counter
- 1 configurable input C per counter
- Counting range 0…16 777 215 (24 bit)
- Selectable digital filter for all inputs (10 kHz…150 kHz)

1.6 Typical application

- Handling and assembly machines
- Pick and place functions
- Economical palletising and assembly drives
- Automatic angle control, e.g. cameras, lights, antennas, etc.
- Static positioning axes (setup)

1.7 Supplementary manuals

- PG5 2.1 User guide | 26/732
- PCD1 series | 26/875
- PCD2 series | 26/737
- PCD3 series | 26/789

1.8 Reference to more information

Please visit our support site www.sbc-support.ch for more detailed information on our products.

2 Quick start

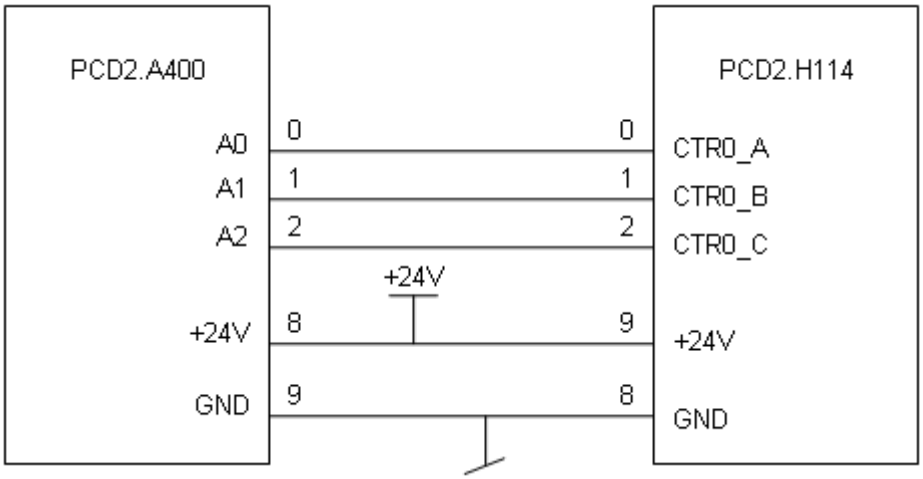
2.1 Example with PCD2

The following hardware has been used in the example project:

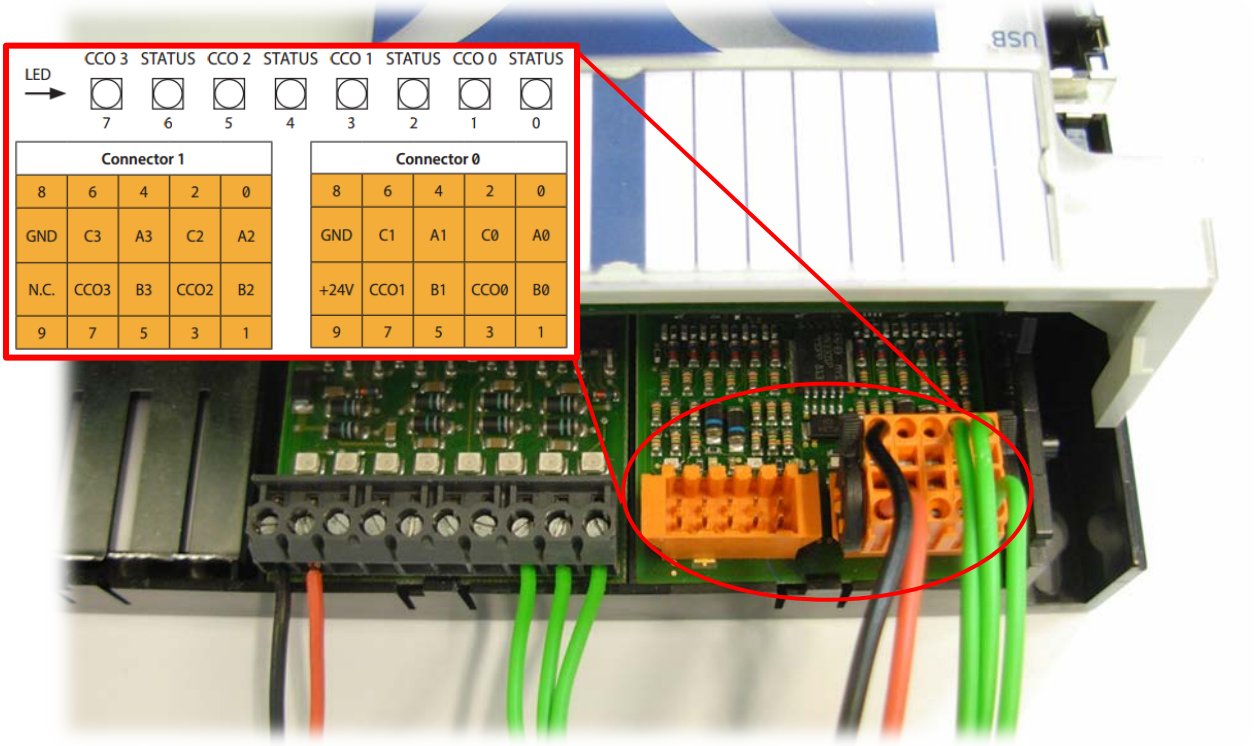
- **PCD2.M5540**
- **PCD2.A400**
- **PCD2.H114**

This is a simple example showing the basic operation of the PCD2.H114 module. We only use counter 0 with a PCD2.A400 to simulate signals on inputs A (CTR0_A), B (CTR0_B) and C (CTR0_C).

2.1.1 Schematic diagram



2.1.2 Definition of the connectors

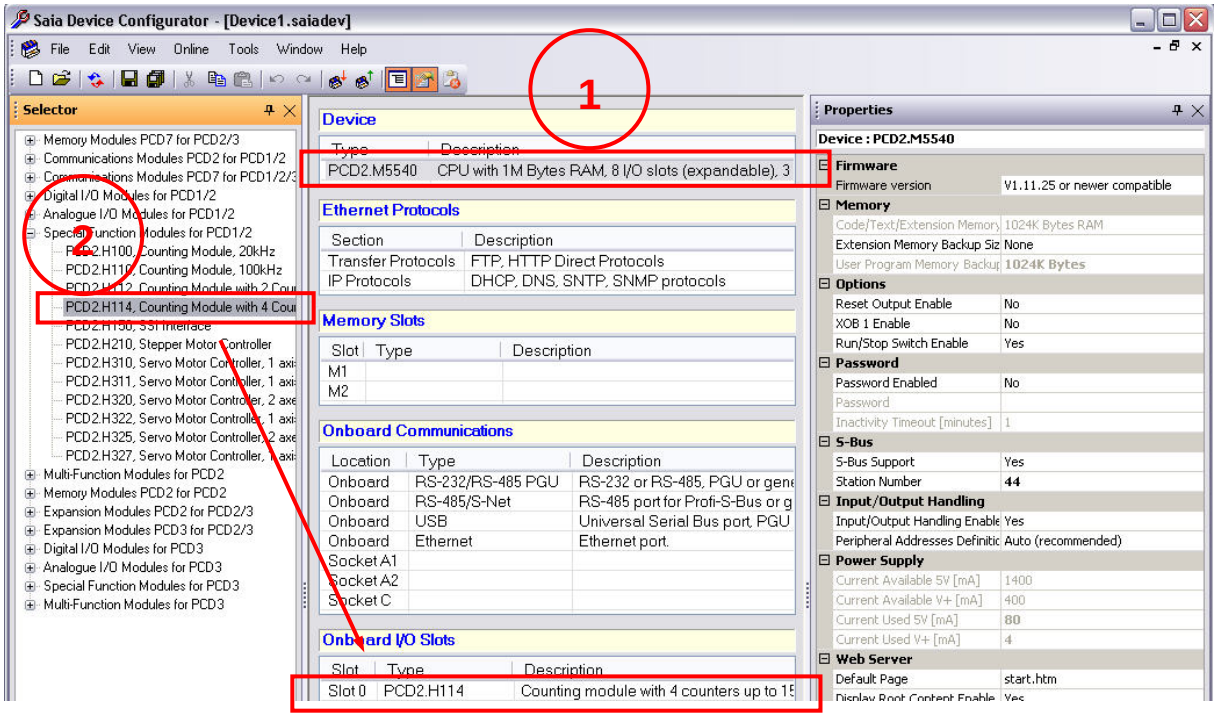


2.1.3 Description of signals

Signal name	Input / output	Description
Ax	I	Counter input
Bx	I	Direction input
Cx	I	Control input
CCOx	O	Command output

2.1.4 Hardware configuration

When you start a new project, the first thing to do is to select your CPU [1] and module [2] with the PG5 Device Configurator. You can find the module on the “Special Function Modules for PCD1/2” tab and then drag and drop onto the slot that you want. On the right-hand side, you will find the Properties window and it is possible to give an initial configuration for each counter. It is also possible to make changes to each property on a user program.



Device Configurator: Example with PCD2

Configurations in a user program have a higher priority than the configurations made with the Device Configurator.

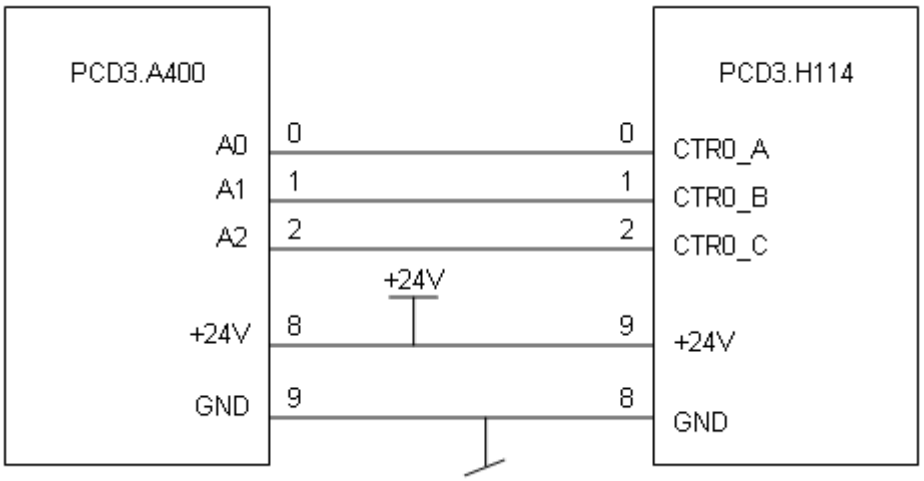
2.2 Example with PCD3

The following hardware has been used in the example project:

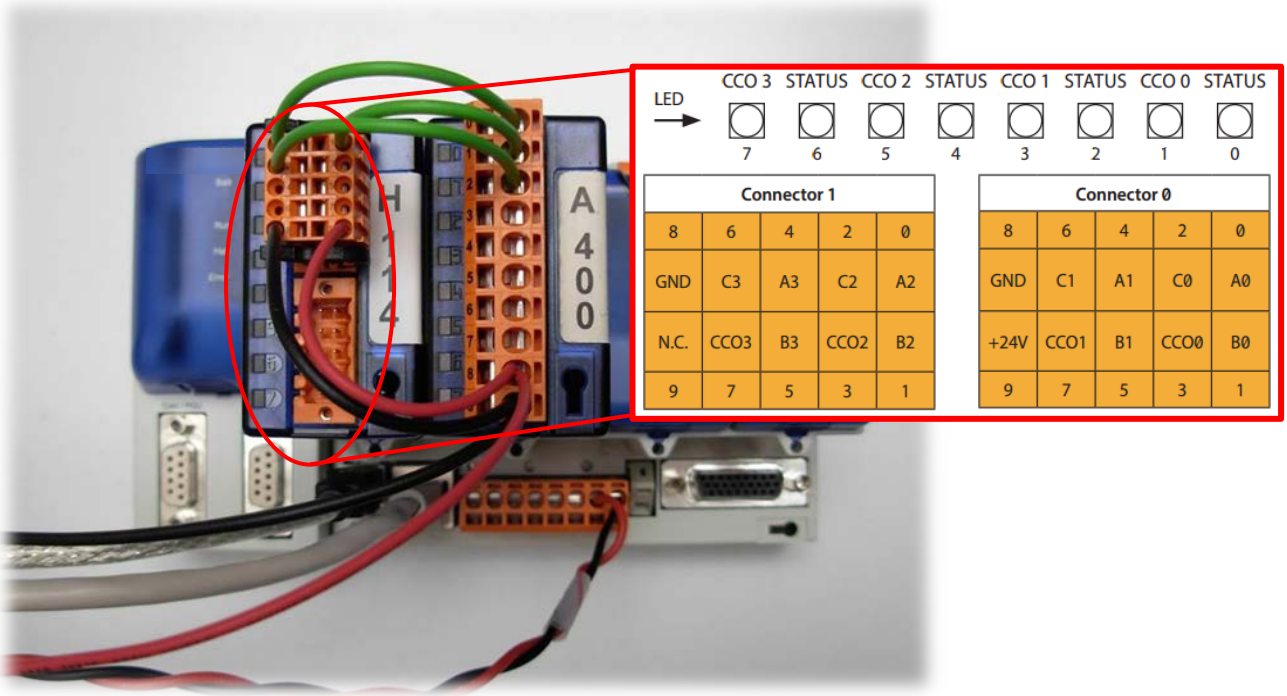
- **PCD3.M5540**
- **PCD3.A400**
- **PCD3.H114**

This is a simple example to show the basic operation of the PCD3.H114 module. We only use counter 0 with a PCD3.A400 to simulate signals on inputs A (CTR0_A), B (CTR0_B) and C (CTR0_C).

2.2.1 Schematic diagram



2.2.2 Definition of the connectors

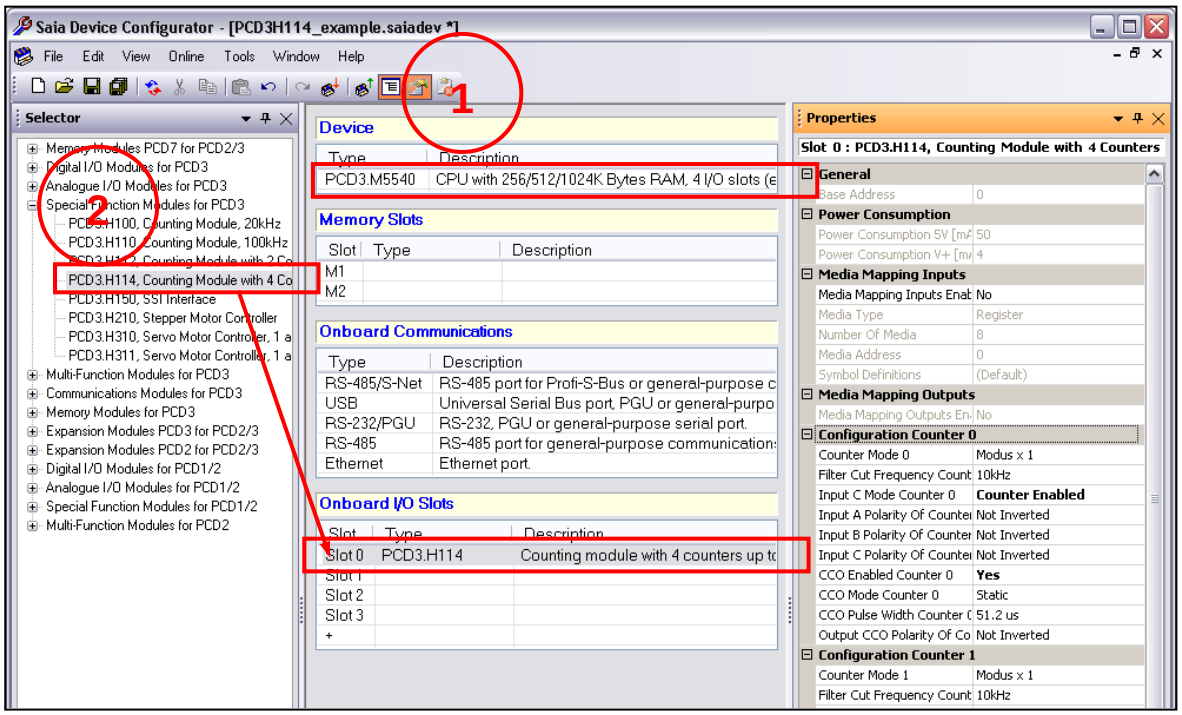


2.2.3 Description of signals

Signal name	Input / output	Description
Ax	I	Counter input
Bx	I	Direction input
Cx	I	Control input
CCOx	O	Command output

2.2.4 Hardware configuration

When you start a new project, the first thing to do is to select your CPU [1] and module [2] with the Saia Device Configurator. You can find the module on the “Special Function Modules for PCD3” tab and then drag and drop onto the slot that you want. On the right-hand side, you will find the Properties window and it is possible to provide an initial configuration for each counter. It is also possible to make changes to each property on a user program.



Device Configurator: Example with PCD

Configurations in a user program have a higher priority than the configurations made with the Device Configurator.

2.2.5 Software

The initial configuration on the Device Configurator is shown below:

Configuration Counter 0	
Counter Mode 0	Modus x 1
Filter Cut Frequency Count	10kHz
Input C Mode Counter 0	**Counter Enabled**
Input A Polarity Of Counte	Not Inverted
Input B Polarity Of Counte	Not Inverted
Input C Polarity Of Counte	Not Inverted
CCO Enabled Counter 0	**Yes**
CCO Mode Counter 0	Static
CCO Pulse Width Counter (	51.2 us
Output CCO Polarity Of Co	Not Inverted

The example program sets the state of input B to high and makes pulses on input A. In this configuration, the counter counts up at each rising edge of input A. Input C is configured in “Counter Enable” mode, the program sets the state of this input to high.

The output CCO will be active when the value of the compare register is reached. CCO mode is static, the “CCO Pulse Width Counter 0” parameter has no effect here.

With the Device Configurator, you can enter some parameters, but not all. It is just a basic configuration. To make a full configuration, you have to write to the “I/O Configuration Register” and “Mode Register”. Please refer to Chapter 5.

2.2.6 Program description

This program shows a simple use of the PCD3.H114 module. Basic parameters are configured on the Device Configurator. Only counter 0 is used.

If the program is running without changes, counter 0 counts up at each rising edge of input A. Some operations, e.g. Reset, Preset, etc., are possible using the different flags.

Set a high flag to show the effect. If you want to change any parameters, first change the register value then set the corresponding flag. Impulses on input A are made on the Fupla sheet. In this example, it is not possible to be in Mode X2 or Mode X4 because input B is always high; only input A changes its state.

2.2.7 IL program code

Definitions:

```
;-------------------------------------------------------------;
Init        EQU         FB      ; General Init
;-------------------------------------------------------------;
Enable      EQU         R :=1   ; Register with Enable value
Disable     EQU         R :=0   ; Register with Disable value
;---------- Counter0 ----------------------------------       ;
rCounter0   EQU         R 100   ; Register with current Counter value
rPreset0    EQU         R 1     ; Register with Preset value (to load)
rTrigger0   EQU         R 2     ; Register with Trigger value
rCompare0   EQU         R 3     ; Register with Compare value (to load)
rValue0     EQU         R 4     ; Register with direct value to write counter
rIO_Conf0   EQU         R 5     ; Register with I/O configuration
rMode_Conf0 EQU         R 6     ; Register with mode configuration
fReset0     EQU         F 0     ; Flag to reset counter
fPreset0    EQU         F 1     ; Flag to preset counter
fTrigger0   EQU         F 2     ; Flag to make trigger on counter
fIO_Conf0   EQU         F 3     ; Flag to apply the I/O configuration
fMode_Conf0 EQU         F 4     ; Flag to apply the mode configuration
fEnable0    EQU         F 5     ; Flag to enable counter
fDisable0   EQU         F 6     ; Flag to disable counter
fWrCounter0 EQU         F 7     ; Flag to write directly to the counter register
Cmd_CTR     EQU          F 10   ; Flag to begin pulses on input A (Fupla sheet)
;=============================================================================================
; XOB
;=============================================================================================
    XOB     16
    CFB     init                ; Call initialisation block
    EXOB
```

Main program:

```
;=============================================================================================
; See Initialisation on next page
    COB     0                   ; Start of main program
            0
    RDP     IO.Slot0.IOAccess.COUNTER0_VALUE_READ ; Read counter value ; This command reads…
            rCounter0           ; … the counter0 value and puts it on register rCounter0
;----------Write counter 0 ----------          ; Set flag fWrCounter0 (F 7) if you want to direct write counter 0 value
    STH     fWrCounter0
    JPD     L Pres
    WRP     IO.Slot0.IOAccess.COUNTER0_VALUE_WRITE ; This command writes directly to counter…
                                rValue0         ; … with the value of register rValue0 (R 4)
    RES     fWrCounter0
;----------                    Preset counter 0 ----------; Set flag fPreset0 (F 1) if you want to make a preset on coun-
ter0
Pres:       STH                 fPreset0
```

```
                JPD             L Rst
                WRP             IO.Slot0.IOAccess.COUNTER0_PRESET_VALUE ; This command writes a preset value to…
                                rPreset0            ; … “Counter Preset Register” with the value of register rPreset0
                WRPB            IO.Slot0.IOAccess.COUNTER0_PRESET ; This command makes a preset on counter0
                Enable
                RES             fPreset0
;----------------               Reset counter 0 ----------------    ; Set flag fReset0 (F 0) if you want to make a reset on coun-
ter0
Rst:            STH             fReset0
                JPD             L Trig
                WRPB            IO.Slot0.IOAccess.COUNTER0_RESET ; This command makes a reset on counter0
                                Enable
                RES             fReset0
;----------------               Trigger counter 0 ----------------    ; Set flag fTrigger0 (F 2) if you want to make a trigger on
counter0
Trig:           STH             fTrigger0
                JPD             L IOConf
                WRPB            IO.Slot0.IOAccess.COUNTER0_CNTR_STATE_TO_TRIGGER ; This command makes a…
                                Enable              ; … trigger on counter0 and puts the result on “Counter Trigger Regis-
ter”…
                RES             fTrigger0
                RDP             IO.Slot0.IOAccess.COUNTER0_TRIGGER_VALUE ; This command reads the “Counter Trig-
ger…
                                rTrigger0           ; … Register” and puts the value on register rTrigger0 (R 2)
;----------------IO Conf. counter 0 ----------------          ; Set flag fIO_Conf0 (F 3) if you want to change IO config of counter 0
IOConf:         STH             fIO_Conf0
                JPD             L ModConf
                WRPB            IO.Slot0.IOAccess.COUNTER0_IO_CONFIG ; This command writes new config to “Counter0
I/O…
                                rIO_Conf0           ; Configuration Register” with value on rIO_Conf0 (R 5). If you want to
                RES             fIO_Conf0           ; change config, first write “rIO_Conf0” and then load it on counter
;----------------               Mode Conf. counter 0 -------------   ; Set flag fMode_Conf0 (F 4) if you want to change Mode con-
fig of
ModConf:        STH             fMode_Conf0                 ; counter 0
                JPD             L Ena
                WRPB            IO.Slot0.IOAccess.COUNTER0_MODE_CONFIG ; This command writes new config to
                rMode_Conf0       ; “Counter0 Mode Register” with value on rMode_Conf0 (R 6). If you
                RES             fMode_Conf0                 ; want to change config, first write “rMode_Conf0” and then
load it.
;----------------               Enable counter 0 ----------------; Set flag fEnable0 (F 5) if you want to enable counter 0.
Ena:            STH             fEnable0
                JPD             L Dis
                WRPB            IO.Slot0.IOAccess.COUNTER0_ENABLED ; This command enables counter 0
                                Enable
                RES             fEnable0
;----------------               Disable counter 0 ----------------; Set flag fDisable0 (F 6) if you want to disable counter 0.
Dis:            STH             fDisable0
                JPD             L End
                WRPB            IO.Slot0.IOAccess.COUNTER0_ENABLED ; This command disables counter 0
                                Disable
                RES             fDisable0
```

```
;-------------------------------------------------------------;
End:        ECOB                    ; End
```

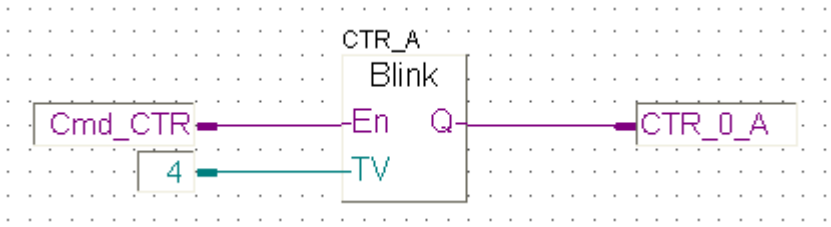
Initialisation:

```
;=============================================================================================
            FB                      init        ; Start of initialisation
             ACC                    H                   ; Set accumulator
            RES                     Cmd_CTR             ; Reset the command for the CTR_A (Fupla sheet)
;----------Init counter 0 ----------;
            LD                      rValue0             ; Value to load the initial counter value or to write counter directly
                                    0           ;
            LD                      rPreset0            ; Value to load the Preset Register
                                    60                  ;
            LD                      rCompare0           ; Value to load the Compare Register
                                    5           ;
            LD                      rIO_Conf0           ; Value to load the I/O Configuration Register
                                    00010001Y           ; See manual for bits description
            LD                      rMode_Conf0         ; Value to load the Mode Register
                                    00001000Y           ; See manual for bits description
            WRPB                    IO.Slot0.IOAccess.COUNTER0_ENABLED ; Enable counter. This must be done even if input C
is on
                                    Enable              ; “Counter Enable” mode
            WRP   IO.Slot0.IOAccess.COUNTER0_COMPARE_VALUE ; Load Compare Register with value on…
                                    rCompare0  ; … rCompare0
            WRP   IO.Slot0.IOAccess.COUNTER0_PRESET_VALUE ; Load Preset Register with value on…
                                    rPreset0     ; … rPreset0
            WRP                     IO.Slot0.IOAccess.COUNTER0_VALUE_WRITE ; Load counter with value on rValue0
                                    rValue0
            SET                     O 17                ; Set input CTR_B_0 (input B)
            SET                     O 18                ; Set input CTR_C_0 (input C)
            SET                     Cmd_CTR             ; Set the command to make pulses on CTR_A (Fupla sheet)


            EFB                     ; End of initialisation
;=============================================================================================
```

2.2.8 Fupla code

At the end of the initialisation, the “Cmd_CTR” flag is set. Cmd_CTR activates the blinker and input A blinks with a 0.4s time value



FBox description:

Inputs / outputs

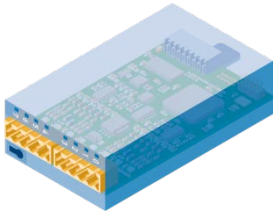
En	Enable	Enable signal starting the blink function.
TV	Time Value	Time value in 1/10 of a second.
Q	Blinker state	Blinking output signal.

Blinks output Q with periods of time value while enable input is high. The Time Value is given in 1/10 of a second. When En is low, Q is set low.

3 Technical data

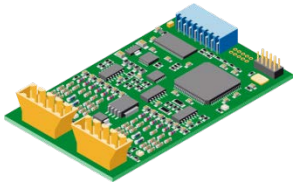
3.1 Mechanical dimensions

PCD3 module



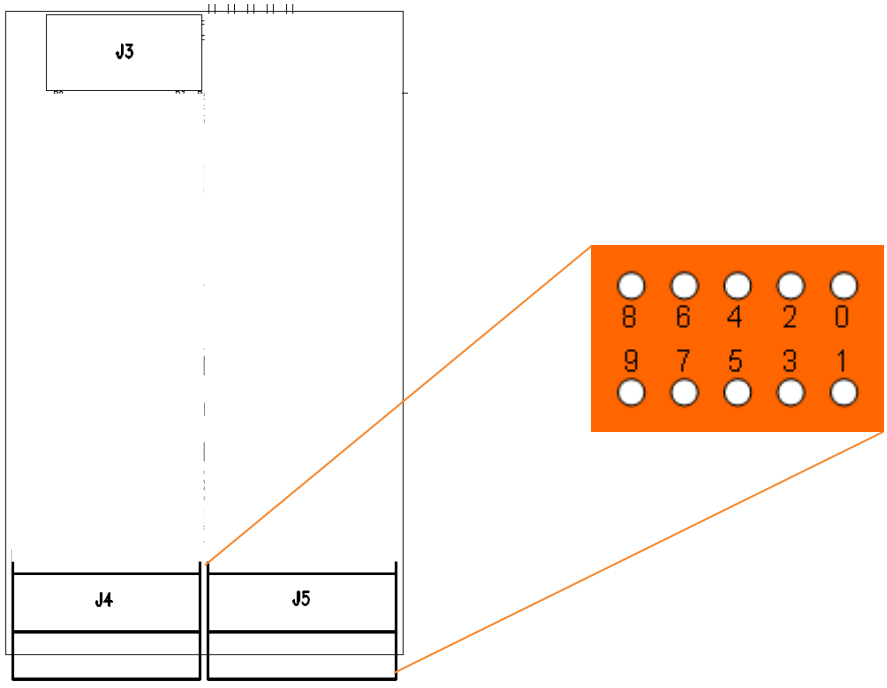
- **Housing colour: Blue**
- **Connection: 2 x pluggable spring terminal block**

PCD2 module



- **Without housing**
- **Connection: Pluggable screw terminal block**

3.2 Connector information



J5.0	J5.1	J5.2	J5.3	J5.4	J5.5	J5.6	J5.7	J5.8	J5.9
A0	B0	C0	CCO0	A1	B1	C1	CCO1	GND	+24V
J4.0	J4.1	J4.2	J4.3	J4.4	J4.5	J4.6	J4.7	J4.8	J4.9
A2	B2	C2	CCO2	A3	B3	C3	CCO3	GND	n. c.

3.3 Technical data for the hardware

3.3.1 24VDC supply

The external 24VDC supply is used for the CCO outputs.

Parameter	Value
Voltage range	24VDC +25% / -20%
Polarity protection	No

3.3.2 Inputs Ax, Bx, Cx

Parameter	Value
Voltage range	0…32V
Low range	0…5V
High range	15…32V
Trigger level low-high	approx. 10V
Trigger level high-low	approx. 9V
Hysteresis	> 0.5V
Input current	5…6mA
Time delay	< 2µs

3.3.3 Outputs CCOx

Parameter	Value
Voltage range	10…32V
Output current	> 0.5A
Voltage drop (0.5A)	< 2V
Switch-on time	Typ. < 100µs
Switch-off time	Typ. < 30µs
Short circuit protection	Yes
Temperature protection	Yes

4 External interface

4.1 Description of signals

Signal name	I/O	Description
Ax	I	Counter input
Bx	I	Direction input
Cx	I	Control input
CCOx	O	Command output

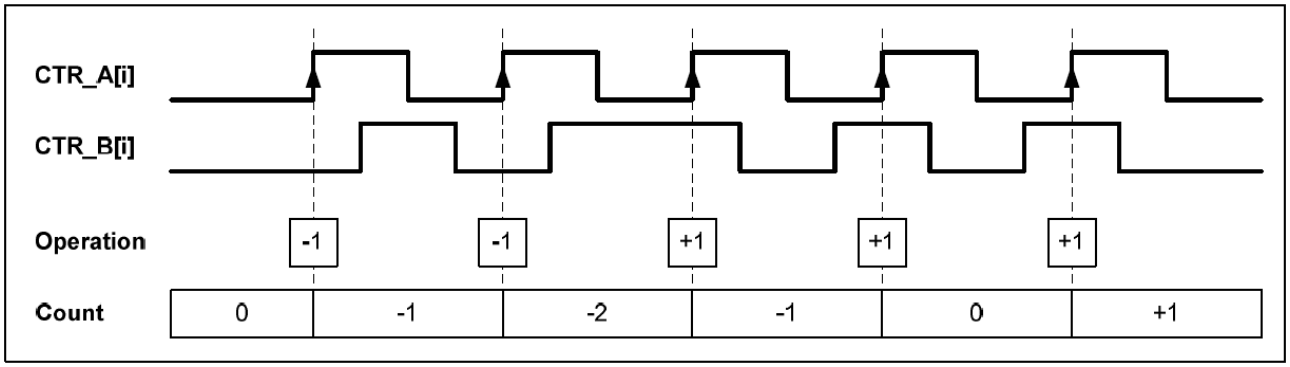
The inputs Ax and Bx are the inputs used to count. Input Cx define different functions such as:

- Trigger (edge sensitive)
- Counter Enable (state sensitive)
- Counter Preset (edge sensitive)
- Counter Reset (edge sensitive)

The interpretation of signals Ax and Bx is described in the chapter below.

4.2 Description of protocols

4.2.1 Mode x1 protocol

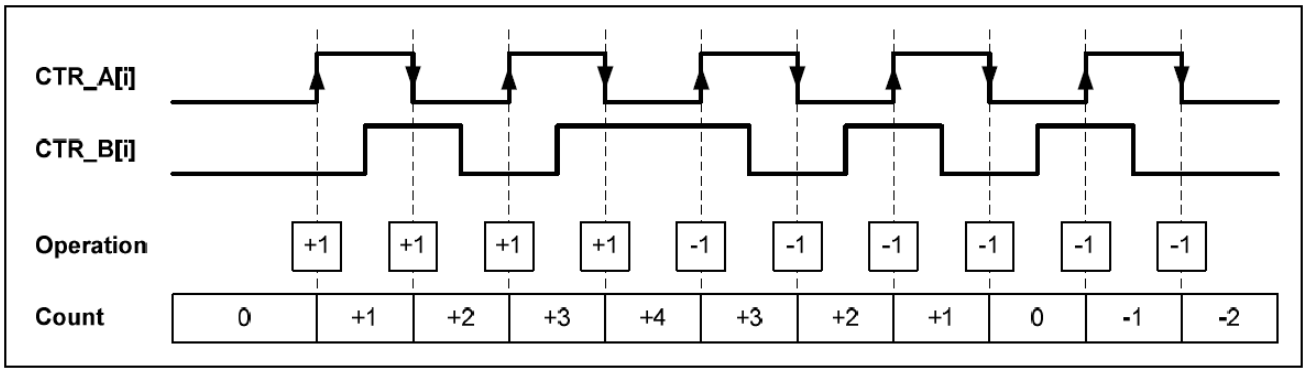


In this mode, the counter is set on every positive edge of Ax and counts up or down depending on the Bx state.

Truth table in Mode x1

Ax	Bx	Operation
↑	0	Counter down -1
↑	1	Counter up +1

4.2.2 Mode x2 protocol

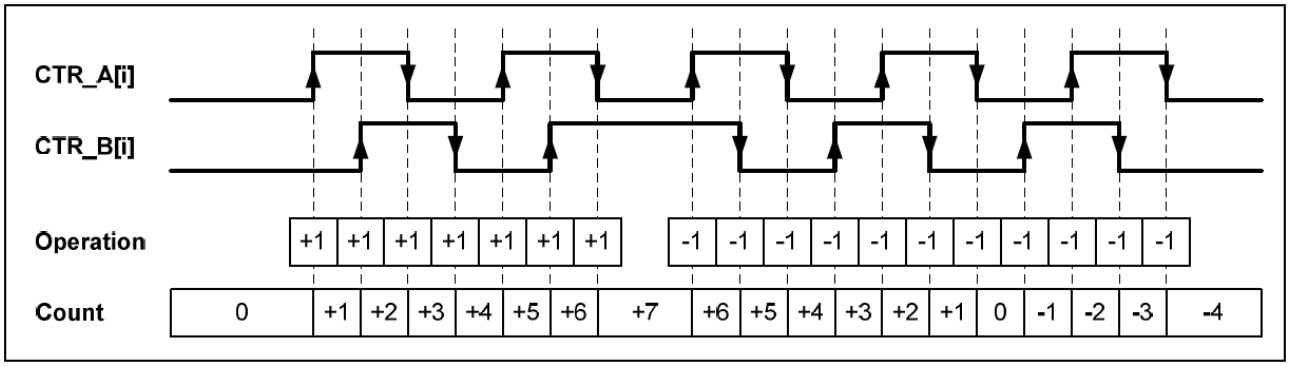


In this mode, the counter is set on every edge of Ax (positive and negative) and counts up or down depending on the Bx state.

4.2.3 Truth table in Mode x2

Ax	Bx	Operation
↑	0	Counter up +1
↑	1	Counter down -1
↓	0	Counter down -1
↓	1	Counter up +1

4.2.4 Mode x4 protocol



In this mode, the four edges of Ax and Bx act with the counter and count up or down depending on inputs states. See table below.

4.2.5 Truth table in Mode x4

Ax	Bx	Operation
↑	0	Counter up +1
↑	1	Counter down -1
↓	0	Counter down -1
↓	1	Counter up +1
0	↑	Counter down -1
1	↑	Counter up +1
0	↓	Counter up +1
1	↓	Counter down -1

5 Register bank

Command	Read / Write	Description
IO.Slot0.IOAccess.CNT0SR	R	Counter Status Register Bit 7:6: Reserved Bit 5: Current state of CMF (Compare Match Flag) Bit 4: Reserved Bit 3: Current state of CCO (Counter Control Output) Bit 2: Current state of input C Bit 1: Current state of input B Bit 0: Current state of input A
IO.Slot0.IOAccess.COUNTER0_CNTR_STATE_TO_TRIGGER	W	Counter Trigger 0: Nothing 1: Put the counter value on the Counter Trigger Register
IO.Slot0.IOAccess.COUNTER0_PRESET	W	Counter Preset 0: Nothing 1: Load the counter with the value of the Counter Preset Register
IO.Slot0.IOAccess.COUNTER0_RESET	W	Counter Reset 0: Nothing 1: Resets the counter
IO.Slot0.IOAccess.COUNTER0_ENABLED	W	0: Counter Disable 1: Counter Enable
IO.Slot0.IOAccess.COUNTER0_CCO_ENABLED	R / W	Enable/Disable/Clear CCO output 0: CCO disabled/clear 1: CCO enabled (CCO is defined by the CMF flag and by the Counter Mode Register)
IO.Slot0.IOAccess.COUNTER0_IO_CONFIG	R / W	Counter IO Configuration Register Bit 7:5: Inputs A, B, C digital filter 000: 10kHz cut frequency 001: 20kHz cut frequency 010: 50kHz cut frequency 011: 100kHz cut frequency 100: 150kHz cut frequency 101: Reserved Bit 4: Enable/Disable/Clear Output CCO 0: CCO disabled/clear

		1: CCO enabled (CCO is defined by the CMF flag and by the Counter Mode Register) Bit 3: Output CCO Polarity 0: Not inverted 1: Inverted Bit 2: Input C Polarity 0: Not inverted 1: Inverted Bit 1: Input B Polarity 0: Not inverted 1: Inverted Bit 0: Input A Polarity 0: Not inverted 1: Inverted
IO.Slot0.IOAccess.COUNTER0_MODE_CONFIG	R / W	Counter Mode Register Bit 7: Reserved Bit 6:5: CCO pulse width 00: 51.2 µs 01: 102.4 µs 10: 204.8 µs 11: 409.6 µs Bit 4:3: Input C Mode 00: Trigger 01: Counter Enable 10: Counter Preset 11: Counter Reset Bit 2: CCO Mode 0: Static 1: Dynamic Bit 1:0: Counter Mode 00: Mode x1 01: Mode x2 10: Mode x4 11: Reserved
IO.Slot0.IOAccess.COUNTER0_VALUE_READ	R	Read Counter Value and put it on a Register
IO.Slot0.IOAccess.COUNTER0_VALUE_WRITE	W	Write Counter Value with a value to a defined register
IO.Slot0.IOAccess.COUNTER3_COMPARE_VALUE	W	Counter Compare Register, contains the value to be compared
IO.Slot0.IOAccess.COUNTER0_PRESET_VALUE	W	Counter Preset Register, contains the preset value

IO.Slot0.IOAccess.COUNTER0_TRIGGER_VALUE	R	Counter Trigger Register, contains the last triggered value
IO.Slot0.IOAccess.FPGA_VERSION	R	FPGA version (version number)
IO.Slot0.IOAccess.IOCR0	R / W	Same as COUNTER0_IO_CONFIG
IO.Slot0.IOAccess.MCR0	R / W	Same as COUNTER0_MODE_CONFIG

6 Ordering information

Type	Description	Weight
PCD2.H112	Intelligent fast counting module, 150 kHz, 2 counting channels with incremental encoders	24 g
PCD2.H114	Intelligent fast counting module, 150 kHz, 4 counting channels with incremental encoders	27 g
PCD3.H112	Intelligent fast counting module, 150 kHz, 2 counting channels with incremental encoders	66 g
PCD3.H114	Intelligent fast counting module, 150 kHz, 4 counting channels with incremental encoders	70 g

Appendix

Symbols

This symbol refers the reader to other information in this manual, in another manual, or in technical documentation on this topic. Direct links to other documentation are not provided.

Instructions with this symbol must always be observed.

Mailing address for Saia-Burgess Controls

Saia-Burgess Controls AG
Bahnhofstrasse 18
CH-3280 Murten
Switzerland

T +41 26 580 30 00
F +41 26 580 34 99

www.saia-pcd.com

www.sbc-support.com