**sbc** +
SAIA BURGESS CONTROLS

PCD1.B1010-A20

E-Line RIO 24DI, 10Rel
Starting with FW 1.08.xx

The L-Serie E-Line RIO modules are controlled via the RS-485 serial communication protocols S-Bus and Modbus for decentralised automation using industrial quality components. The data point mix is specifically designed for building automation applications.

The compact design according to DIN 43880 enables installation in electrical distribution boxes even in the most confined spaces. Installation and maintenance are facilitated by the local manual override for each output. Remote maintenance is also possible by accessing the manual override using the Saia PCD® controller's web interface. Programming is very efficient and fast using a complete FBox library with web templates for S-Bus. Individual programs may directly access the data points via Registers and Flags. Complete documentation is included in this data sheet.
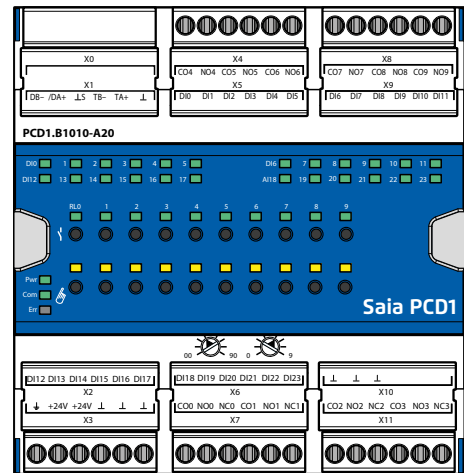
Features

▶ S-Bus protocol optimized for fast data exchange

▶ Modbus protocol for integration in multi-vendor installations*

▶ Local override operating level via web panel or buttons on the module

▶ Specific I/O mix suitable for HVAC systems

▶ Easy programming using the FBox library and web templates

▶ Industrial hardware in accordance with IEC EN 61131-2

▶ Pluggable terminal blocks protected by flaps

▶ Electrically isolated RS-485 interface

* By default the module is working in S-Bus Data Mode with Autobaud detection.
  To configure Modbus the Windows-based application "E-LineApp" is required

General technical data

Power supply

| Supply voltage	24 VDC, –15/+20% max. incl. 5% ripple (in accordance with EN/IEC 61131-2)
Electrically isolated	500 VDC between power supply and RS-485
Power consumption	1.2…3 W

Interfaces

Communications interface	RS-485 with electrical isolation Baud rate: 9,600, 19,200, 38,400, 57,600, 115,200 bps (Autobauding)
Address switch for S-Bus address	Two rotary switches 0…9 Address range 0…98
Terminating resistor	Integrated, can be activated via a wire jumper

General data

Ambient temperature	Operation: 0…+55 °C Storage: −40…+70 °C

Dimensions and installation
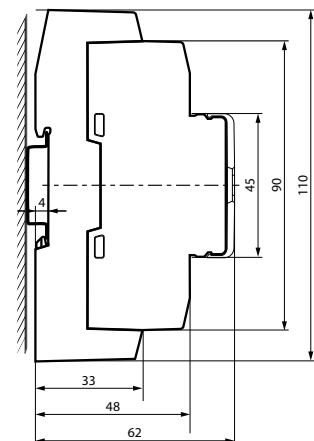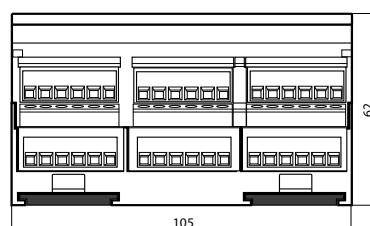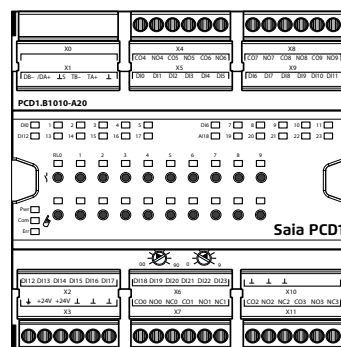
on a 35 mm top-hat rail (in accordance with DIN EN 60715 TH35)

Housing width 6 HP (105 mm)
Compatible with electrical control cabinet
(in accordance with DIN 43880, size 2 × 55 mm)

Input/output configuration

Digital inputs

Number	24
Input voltage	24 VDC, source operation (positive switching)
Switching level	Low: 0…5 V, High: 15…24 V
Input current	Typical 2 mA
Input filter time (DC)	Typical 8 ms

Relay outputs

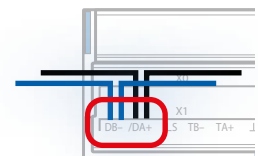|	
Number	10 (6 normally open/ 4 changeover)
Switching voltage max.	250 VAC / 30 VDC
Switching current max.	4 A (AC1, DC1)
Contact protection	None
Local manual override	Override operation by button

Terminal technology

Push-in spring terminals enable wiring with rigid or flexible wires with a diameter up to 1.5 mm². A max. of 1 mm² is permitted with cable end sleeves.
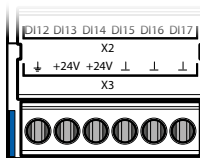
Connection concept

The device is supplied by a 24 VDC voltage supply.

Bus wiring

DB– and /DA+ terminals must be used for exchanging data between modules. The bus is through-wired by using one terminal per bus line in order to not interrupt the bus connection when removing the connector on modules.
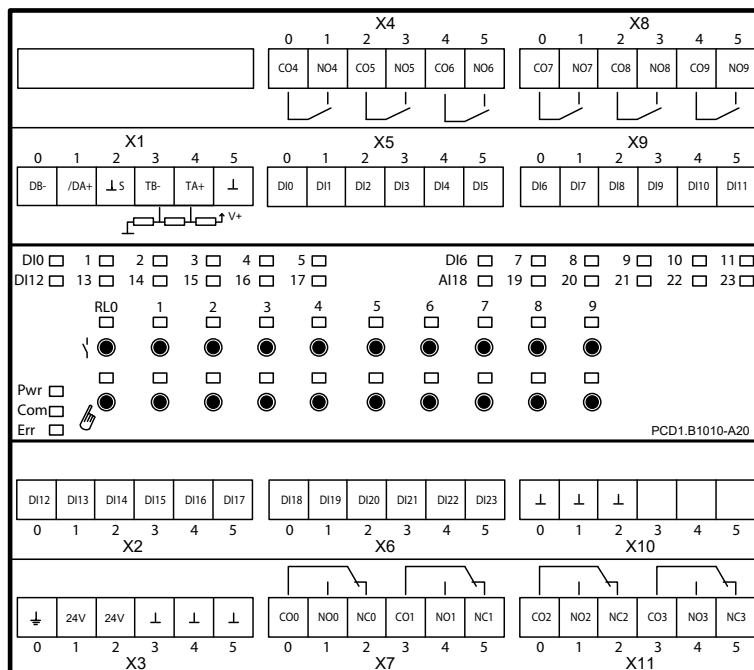
Flexible RS-485 cables with a cross-section of no more than 0.75 mm² must be used for bus wiring. A total cross-section of 1.5 mm² is allowed per terminal.

The communication bus can be terminated with internal terminating resistors using wire bridges.

Assignment overview

Connection diagrams

Digital input
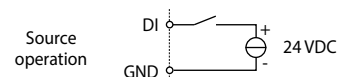
Relays (normally open)

Relay (changeover)
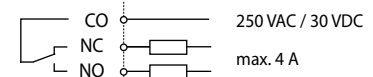
Terminating resistor
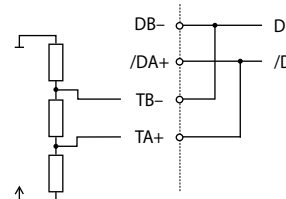
GND	⊥	ground
GND	⊥D	digital electrically isolated ground
GND	⊥A	analog electrically isolated ground
SGND	⊥S	signal ground
	⊥D#	# = alphanumeric index by different grounds

Programming

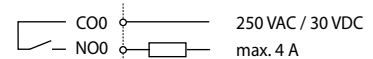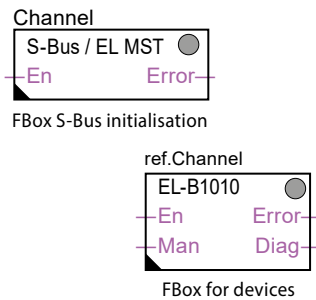**5** The modules are addressed and programmed with Saia PG5® Fupla FBoxes. Web templates are available for the operation and visualisation of the manual override function.

Fupla

Selector
All
Filter
☐ E-Line
⊞ Initialization channel
⊞ Mixed Rio

E-Line library

Channel

S-Bus / EL MST ⬤
—En   Error—

FBox S-Bus initialisation

ref.Channel

EL-B1010 ⬤
—En   Error—
—Man   Diag—

FBox for devices
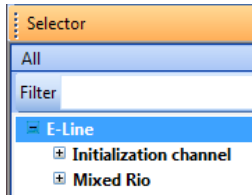
Communication FBox

▶ Data exchange for I/O via optimised S-Bus

▶ Configurable save state for bus interruption or timeout

▶ Direct generation of the symbols

▶ Reading and writing of the status of the manual override status
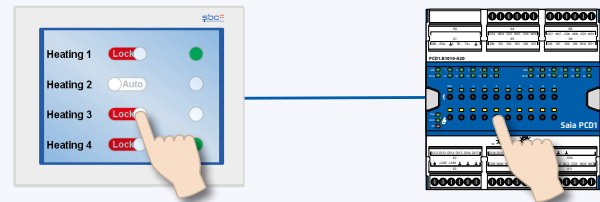
▶ Direct compatibility with web macros

ℹ Further information, including which FBoxes are supported, Getting Started, etc., can be found on our support page
www.sbc-support.com

Web templates

Web templates are available for the operation and visualisation of the manual override function.
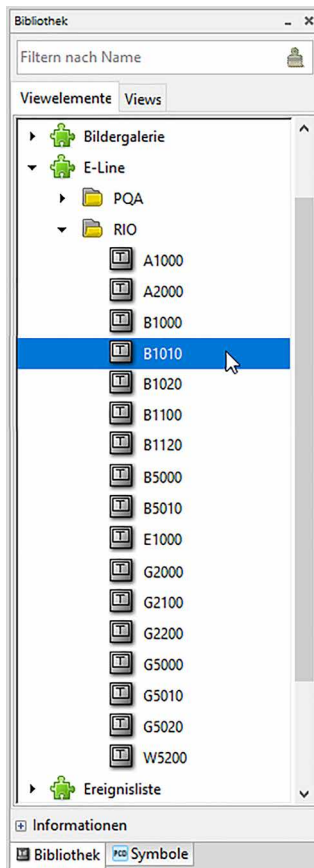
Manual operation

By using the local override function, commissioning can take place independently of the master station.

In addition, the manual operation can also be controlled remotely using a touch panel. If the bus line is cut off, the module keeps the manually set values. Traditional manual operation in the control cabinet door via potentiometers and switches can therefore be completely replaced by this solution.

Five operating modes can be selected for the manual operating function:

Operating modes	Description	Operation	
		at the module	via remote (com)
1	Manual operation deactivated	✘	✘
2	Operation permitted from the module only	✔	✘
3	Operation permitted from the module and limited operation from the panel. If manual operation is activated at the module, it cannot be reset from the panel.	✔	(conditional)
4	Unlimited operation from the panel and module	✔	✔
5	Panel operation (remote)	✘	✔

ℹ Depending on the application, reset of manually set values is allowed from a panel. To address this requirement, it is possible to deactivate or limit manual operation function.

ℹ The inputs of the E-Line RIO modules can be addressed via the standard S-Bus. However the FBox from the E-Line library is used for the configuration of these modules.

It is therefore recommended to use the optimised S-Bus protocol and the corresponding FBoxes from the E-Line library. Mixed mode operation is not recommended.

E-line App device setup

E-Line RIOs support the device setup by a windows application program connected via USB. The installer is available for download from the SBC support page: www.sbc-support.com ➔ E-Line RIO IO Modules.
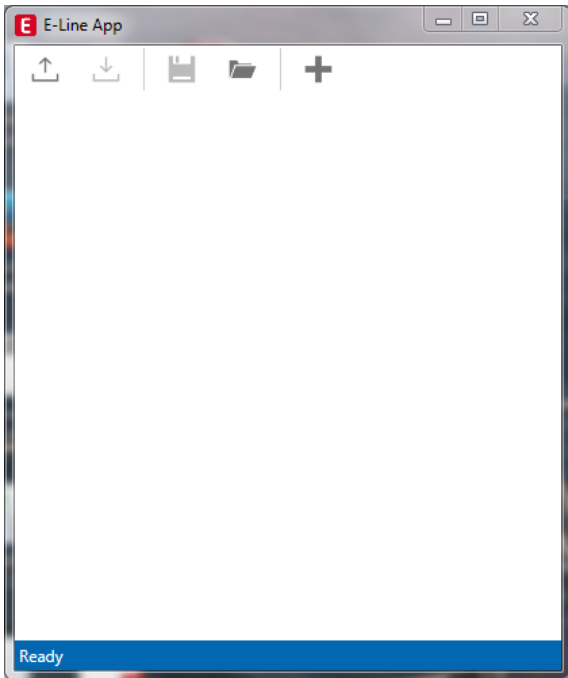
Create a new device configuration

Open an existing device configuration

Save the current settings as device configuration

Upload configuration from the device

Download settings to the device
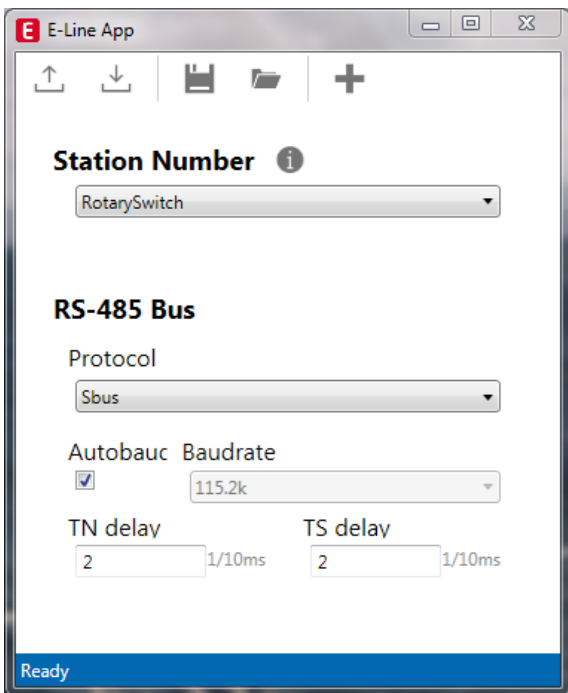
The station number can be set by the rotary switches at the device in the range of 0 … 98. If the rotary switches are set to position 99 the station number can be defined by the device configuration in a range of 0 … 253.

The serial communication protocol can be defined either as S-Bus or Modbus. By default the modules are delivered from factory with S-Bus.
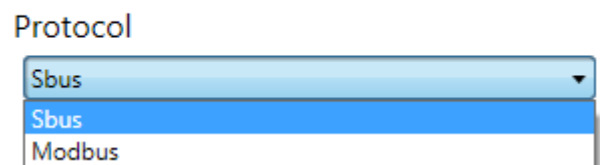
S-Bus settings
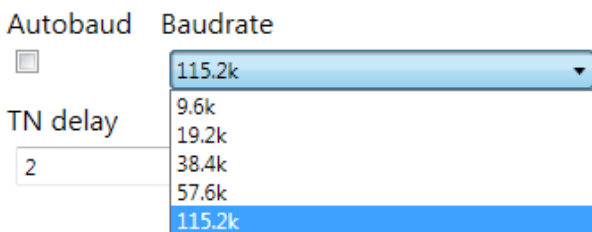
The Baudrate can be defined as automatic detection (default) or set to a specific value. The drop down choice will be available when the check box "Automatic" is unchecked. TN delay and TS delay shall be left at their default values of 2.

Modbus settings
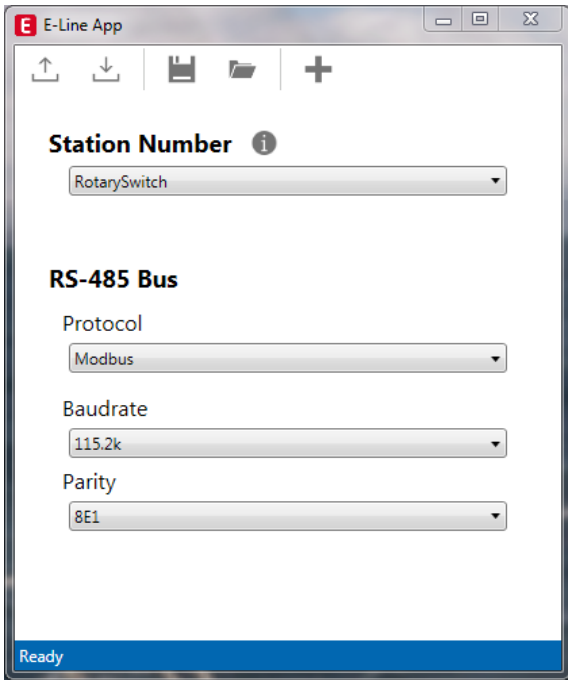
The Baudrate is set by default to 115k. It can be defined as choice of the list.
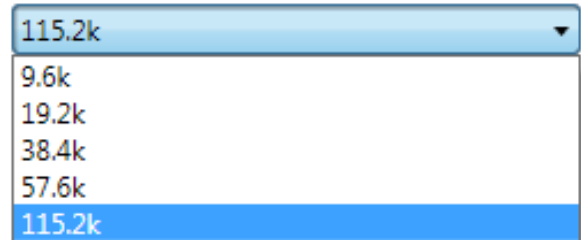
For best interoperability, the Parity Mode and number of Stop Bits can also be set.
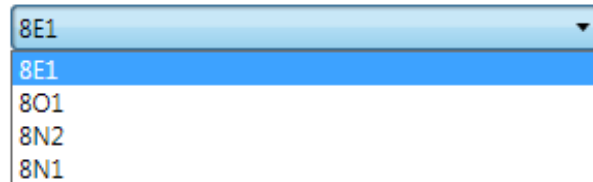
S-Bus communication

S-Bus communication is based on Saia PCD® S-Bus Data Mode. Only the set-up of a unique S-Bus address within the communication line is required to establish a communication between Saia PCD® controllers and E-Line RIO modules. The address can be set using the rotary switches at the front of the module. The baud rate will be learned from the network by factory default. In addition a Windows-based application is available for manual parameter setup. Configuration parameters as well as manual override state and value are saved non-volatile. A delay of about one second between a manual state change and non-volatile saving has to be taken into consideration.

Device address

▶ 0 … 98   Address is taken from the rotary switches

▶ 99   Address is taken from the device configuration. The address is settable with the E-Line configuration software.

Start-up procedure

▶ Reboot: All outputs are cleared (Off state)

▶ <1 sec.   Output in manual operation are set according to the state before power down.

▶ Outputs in automatic mode

If, after reboot, no telegram is received within the "safe state power-on timeout," the module enters into the safe state mode and sets the outputs according to their configured values.

On reception of a valid command telegram the outputs are controlled by the communication. When no communication update follows within the "safe state com. timeout" the module enters into safe state and sets the outputs according to their configured values.

Usage of the E-Line module-specific FBoxes

The usage of the E-Line module-specific FBoxes from the E-Line S-Bus Fupla library allows an easy and efficient commissioning of the E-Line RIO.

The FBox allows the definition and configuration of all possible functionalities of the E-Line RIO like manual override permission, usage of safe state mode, behaviour and colour of the LED's and so on.

In the background, the FBox uses the fast 'E-Line S-Bus' protocol for a high speed communication between the master and the RIO.

S-Bus communication

Direct access to the RIO media with standard S-Bus send and receive telegrams

The following chapter describes the media and parameter mapping to Registers and Flags for individual programming. For efficient PCD programming the E-Line RIO FBox family and templates are suitable for most applications. Only individual programming (e.g. Instruction List) requires standard S-Bus communication.

Digital inputs

Input	Input Value	Read/Write
Digital input 0	Flag 0	R
Digital input 1	Flag 1	R
…	Flag …	R
Digital input 23	Flag 23	R

Relay outputs

Output	Output Value	Read/Write	Manual override Communication	Read/Write*	Manual override Local	Read/Write**
Relay output 0	Flag 30	RW	Register 20	RW	Register 30	RW
Relay output 1	Flag 31	RW	Register 21	RW	Register 31	RW
Relay output 2	Flag 32	RW	Register 22	RW	Register 32	RW
Relay output 3	Flag 33	RW	Register 23	RW	Register 33	RW
Relay output 4	Flag 34	RW	Register 24	RW	Register 34	RW
Relay output 5	Flag 35	RW	Register 25	RW	Register 35	RW
Relay output 6	Flag 36	RW	Register 26	RW	Register 36	RW
Relay output 7	Flag 37	RW	Register 27	RW	Register 37	RW
Relay output 8	Flag 38	RW	Register 28	RW	Register 38	RW
Relay output 9	Flag 39	RW	Register 29	RW	Register 39	RW

* Writable only if S-Bus permission is set in the configuration, otherwise write has no effect
**Writing to these registers has no effect. Used only if hardware permission is set in the configuration

Normal operation: The outputs are set according the flag set by the communication.
Manual operation: The output are set according to the manual command, the communication flags are ignored.
Safe State: In case of a broken communication, a safe state value can be applied, see table Safe State Configuration.

Register format for manual override via S-Bus (Reg. 20 … 29):

Bit 0	Current output value
Bit 30	1: output is driven in manual override by S-Bus
Bit 31	1: output is driven in manual override by local push buttons

Register format for local manual override (Reg. 30 … 39):

Bit 0	Current output value
Bit 31	1: output is driven in manual override by local push buttons

S-Bus communication

Configuration for safe state and manual override

Output	Safe State Enable	Read/Write	Safe State Value	Read/Write
Relay output 0	Flag 320	RW	Flag 350	RW
Relay output 1	Flag 321	RW	Flag 351	RW
Relay output 2	Flag 322	RW	Flag 352	RW
Relay output 3	Flag 323	RW	Flag 353	RW
Relay output 4	Flag 324	RW	Flag 354	RW
Relay output 5	Flag 325	RW	Flag 355	RW
Relay output 6	Flag 326	RW	Flag 356	RW
Relay output 7	Flag 327	RW	Flag 357	RW
Relay output 8	Flag 328	RW	Flag 358	RW
Relay output 9	Flag 329	RW	Flag 359	RW
Communication safe state enable default 0 (disabled)			Flag 400	RW
Communication safe state timeout [ms] Valid values 1000 … 100,000,000, default 15,000			Register 590	RW
Manual operation mode Bit 0: Disabled Bit 1: Remote control limited*, default 1 Bit 2: Local operation enabled, default 1 Bit 3: Remote control unlimited*, default 0 Bits can be combined to enable remote and local operation			Register 592	RW

* If manual operation is locally activated at the module, the output value and manual state cannot be set/reset remotely

Manual operation mode:

▶ Disabled (0)

▶ Local operation only (4, Bit 2 set)

▶ Local operation enabled, remote limited (6, Bit 1 and 2 set), default

▶ Local and remote operation enabled (12, Bit 2 and 3 set)

▶ Remote operation only, local operation disabled (8, Bit 3 set)

The safe state enable flag and the safe state value are combined in the following way:

- Setting the enable flag to 0 keep the output value unchanged in case of safe state occurrence.
- Setting the enable flag to 1 writes the safe state value in case of safe state occurrence.

Device Information

Firmware version (Decimal xyyzz, 10802 ➜ 1.08.02)	Register 600	R
Number of supported registers	Register 601	R
Number of supported flags	Register 602	R
Product type (ASCII String)***	Register 605 … 608	R
Hardware version (Hex)	Register 609	R
Serial number (Hex)	Register 611 … 612	R
Communication protocol (1:S-Bus Slave, 3:Modbus)	Register 620	R
Communication baud rate	Register 621	R
Communication auto baud enable (0:disabled, 1:enabled)	Register 622	R
Communication TN delay *	Register 623	R
Communication TS delay **	Register 624	R
Communication module address	Register 626	R

* Time in 0.1 ms (e.g. 2 means 200 us) before setting activation of RS-485 line driver send mode (only used for S-Bus slave protocol)
** Time in 0.1 ms (e.g. 2 means 200 us) before sending the first character after line driver activation (only used for S-Bus slave protocol)
*** The four registers contain the ASCII characters of the product type.
E.g. for PCD1.A2000-A20:
0605: 50434431H   0606: 2E413230H   0607: 30302D41H   0608: 32300000H

Modbus communication

Modbus fulfils the requirements for standard communication protocols. It is based on Modbus RTU. The Windows-based configuration software is required to enable and set up the Modbus communication parameters. The device address can be set up with the rotary switches at the front of the module. Configuration parameters as well as manual override state and value are saved non-volatile. A delay of about one second between a manual state change and non-volatile saving has to be taken into consideration.

Device address
- ▶ 0 … 98   Address is taken from the rotary switches
- ▶ 99   Address is taken from the device configuration. The address is settable with the E-Line configuration software.

Start-up procedure
- ▶ Reboot: All outputs are cleared (Off state)
- ▶ <1 sec.   Output in manual operation are set according to the state before power down.
- ▶ Outputs in automatic mode

    If, after reboot, no telegram is received within the "safe state power-on timeout," the module enters into the safe state mode and sets the outputs according to their configured values.

    On reception of a valid command telegram the outputs are controlled by the communication. When no communication update follows within the "safe state com. timeout" the module enters into safe state and sets the outputs according to their configured values.

The following chapter describes the media and parameter mapping to Registers and Flags (=Coils).

Supported Modbus services:
- ▶ Function code 1 (read coils)
- ▶ Function code 3 (read registers)
- ▶ Function code 15 (write multiple coils)
- ▶ Function code 16 (write multiple registers)

Modbus communication

Read coils

Request						
Address	**Function**	**Start Address**		**Number of coils to read**		**CRC**	
0…254	1	High-Byte	Low-Byte	High-Byte	Low-Byte	High-Byte	Low-Byte

Reply						
Address	**Function**	**No. of Byte**	**Data**			**CRC**	
0…254	1	0…256	Coil 0…7	Coil 8…15	…	High-Byte	Low-Byte

Write coils

Request									
Address	**Function**	**Start Address**		**Number of Coils to write**		**Coil data**			**CRC**	
0…254	15	High-Byte	Low-Byte	High-Byte	Low-Byte	No. of Bytes	Coil 0…7	…	High-Byte	Low-Byte

Reply						
Address	**Function**	**Start Address**		**Number of written Coils**		**CRC**	
0…254	15	High-Byte	Low-Byte	High-Byte	Low-Byte	High-Byte	Low-Byte

Read register

Request						
Address	**Function**	**Start Address**		**No. of Register to read**		**CRC**	
0…254	3	High-Byte	Low-Byte	High-Byte	Low-Byte	High-Byte	Low-Byte

Reply						
Address	**Function**	**No. of Byte**	**Register Start Addr + 0**		**Addr + n**	**CRC**	
0…254	3	0…256	High-Byte	Low-Byte	…	High-Byte	Low-Byte

Write register

Request									
Address	**Function**	**Start Address**		**No. of Registers**		**No. of Bytes**	**Data Word: Start Addr + 0**		**Addr + n**	**CRC**	
0…254	16	High-Byte	Low-Byte	High-Byte	Low-Byte	2…256	Low-Byte	High-Byte	…	High-Byte	Low-Byte

Reply						
Address	**Function**	**Start Address**		**No of written Registers**		**CRC**	
0…254	16	High-Byte	Low-Byte	High-Byte	Low-Byte	High-Byte	Low-Byte

The CRC has to be calculated over all telegram bytes starting with address field up to the last data byte. The CRC has to be attached to the data. Please find an example at the appendix of this document. For more details, please refer the publicly available Modbus documentation www.modbus.org.

Modbus communication

Digital inputs

Input	Input Value	Read/Write
Digital input 0	Coil 0	R
Digital input 1	Coil 1	R
…	Coil …	R
Digital input 23	Coil 23	R

Relay outputs Digital outputs

Output	Output Value	Read/Write	Manual override Communication	Read/Write*	Manual override Local	Read/Write**
Relay output 0	Coil 30	RW	Value Reg. 40 Enable Reg. 41	RW	Value Reg. 60 Enable Reg. 61	RW
Relay output 1	Coil 31	RW	Value Reg. 42 Enable Reg. 43	RW	Value Reg. 62 Enable Reg. 63	RW
Relay output 2	Coil 32	RW	Value Reg. 44 Enable Reg. 45	RW	Value Reg. 64 Enable Reg. 65	RW
Relay output 3	Coil 33	RW	Value Reg. 46 Enable Reg. 47	RW	Value Reg. 66 Enable Reg. 67	RW
Relay output 4	Coil 34	RW	Value Reg. 48 Enable Reg. 49	RW	Value Reg. 68 Enable Reg. 69	RW
Relay output 5	Coil 35	RW	Value Reg. 50 Enable Reg. 51	RW	Value Reg. 70 Enable Reg. 71	RW
Relay output 6	Coil 36	RW	Value Reg. 52 Enable Reg. 53	RW	Value Reg. 72 Enable Reg. 73	RW
Relay output 7	Coil 37	RW	Value Reg. 54 Enable Reg. 55	RW	Value Reg. 74 Enable Reg. 75	RW
Relay output 8	Coil 38	RW	Value Reg. 56 Enable Reg. 57	RW	Value Reg. 76 Enable Reg. 77	RW
Relay output 9	Coil 39	RW	Value Reg. 58 Enable Reg. 59	RW	Value Reg. 78 Enable Reg. 79	RW

* Writable only if Modbus permission is set in the configuration, otherwise write has no effect

**Writing to these registers has no effect. Used only if hardware permission is set in the configuration

Normal operation: The outputs are set according the flag set by the communication.

Manual operation: The output are set according to the manual command, the communication flags are ignored.

Safe State: In case of a broken communication, a safe state value can be applied, see table Safe State Configuration.

Register format for manual override via Modbus (Reg. 40 … 59):

Bit 0	Current output value
Enable Reg. Bit 14	1: output is driven in manual override by Modbus
Enable Reg. Bit 15	1: output is driven in manual override by local push buttons

Register format for local manual override (Reg. 60 … 79):

Value Reg. Bit 0	Current output value
Enable Reg. Bit 15	1: output is driven in manual override by local push buttons

Modbus communication

Configuration for safe state and manual override

Output	Safe State Enable	Read/Write	Safe State Value	Read/Write
Relay output 0	Coil 320	RW	Coil 350	RW
Relay output 1	Coil 321	RW	Coil 351	RW
Relay output 2	Coil 322	RW	Coil 352	RW
Relay output 3	Coil 323	RW	Coil 353	RW
Relay output 4	Coil 324	RW	Coil 354	RW
Relay output 5	Coil 325	RW	Coil 355	RW
Relay output 6	Coil 326	RW	Coil 356	RW
Relay output 7	Coil 327	RW	Coil 357	RW
Relay output 8	Coil 328	RW	Coil 358	RW
Relay output 9	Coil 329	RW	Coil 359	RW
Communication safe state enable default 0 (disabled)			Coil 400	RW
Communication safe state timeout [ms] Valid values 1000 … 100,000,000, default 15,000			Reg. 1180, 1181	RW
Manual operation mode Bit 0: Disabled Bit 1: Remote control limited*, default 1 Bit 2: Local operation enabled, default 1 Bit 3: Remote control unlimited*, default 0 Bits can be combined to enable remote and local operation			Register 1184	RW

* If manual operation is locally activated at the module, the output value and manual state cannot be set/reset remotely

Manual operation mode:

▶ Disabled (0)

▶ Local operation only (4, Bit 2 set)

▶ Local operation enabled, remote limited (6, Bit 1 and 2 set), default

▶ Local and remote operation enabled (12, Bit 2 and 3 set)

▶ Remote operation only, local operation disabled (8, Bit 3 set)

The safe state enable flag and the safe state value are combined in the following way:

- Setting the enable coil to 0 keep the output value unchanged in case of safe state occurrence.
- Setting the enable coil to 1 writes the safe state value in case of safe state occurrence.

Device Information

Firmware version (Decimal xyyzz, 10802 ➔ 1.08.02)	Register 1200	R
Number of supported registers	Register 1202	R
Number of supported flags	Register 1204	R
Product type (ASCII String)*	Register 1210 … 1217	R
Hardware version (Hex)	Register 1218	R
Serial number (Hex)	Register 1222 … 1224	R
Communication protocol (1: S-Bus Slave, 3: Modbus)	Register 1240	R
Communication baud rate	Register 1242	R
Communication auto baud enable (0:disabled, 1:enabled)	Register 1244	R
Communication Mode 0: 8,E,1; 1: 8,O,1; 2: 8,N,2; 3: 8,N,1	Register 1250	R
Communication module address	Register 1252	R

* The eight registers contain the ASCII characters of the product type.
E.g. for PCD1.A2000-A20:
1210…1217: 5043H | 4431H | 2E41H | 3230H | 3030H | 2D41H | 3230H | 0000H

Modbus communication

CRC Generation Example

(Source: http://modbus.org/docs/PI_MBUS_300.pdf, the following content of this page is copied from the referenced document. In case of any questions, please check out the original source)

The function takes two arguments: unsigned char *puchMsg; A pointer to the message buffer containing binary data to be used for generating the CRC unsigned short usDataLen; The quantity of bytes in the message buffer. The function returns the CRC as a type unsigned short.

CRC Generation Function

```
unsigned short CRC16(puchMsg, usDataLen) ;
unsigned char *puchMsg ;                        /* message to calculate CRC upon */
unsigned short usDataLen ;                       /* quantity of bytes in message */
{
        unsigned char uchCRCHi = 0xFF ;          /* high byte of CRC initialized */
        unsigned char uchCRCLo = 0xFF ;          /* low byte of CRC initialized */
        unsigned uIndex ;                         /* will index into CRC lookup table */
        while (usDataLen--)                       /* pass through message buffer */
        {
                uIndex = uchCRCHi ^ *puchMsgg++;   /* calculate the CRC */
                uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex];
                uchCRCLo = auchCRCLo[uIndex];
        }
        return (uchCRCHi << 8 | uchCRCLo);
}
```

High-Order Byte Table

```
/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40 };
```

Low-Order Byte Table

```
/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04,
0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,
0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,
0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,
0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0,
0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C,
0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,
0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,
0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40 };
```

NOTE

Extra low voltages (ELV) or secure low voltages (SELV) are voltages up to 50 Volts.

NOTE

Low voltages are voltages between 50 … 250 Volts.

INSTALLATION DIRECTION FOR SWITCHING LOWER VOLTAGES

For reasons of safety it is not allowed that extra low voltages and low voltages are connected to two adjacent relay contacts. Neither different phases may be connected to two adjacent relay contacts. But a relay contact between them can be left empty.

If a Saia PCD® system module is connected to low voltage, then all components which are electrically connected to this system must be approved for low voltage.

When using low voltage, all connections to the relay contacts, which are connected to the same circuit, must be protected by a common fuse.
The individual load circuits, on the other hand, may be protected individually by a fuse.

Wiring example with fuses and 3 phases
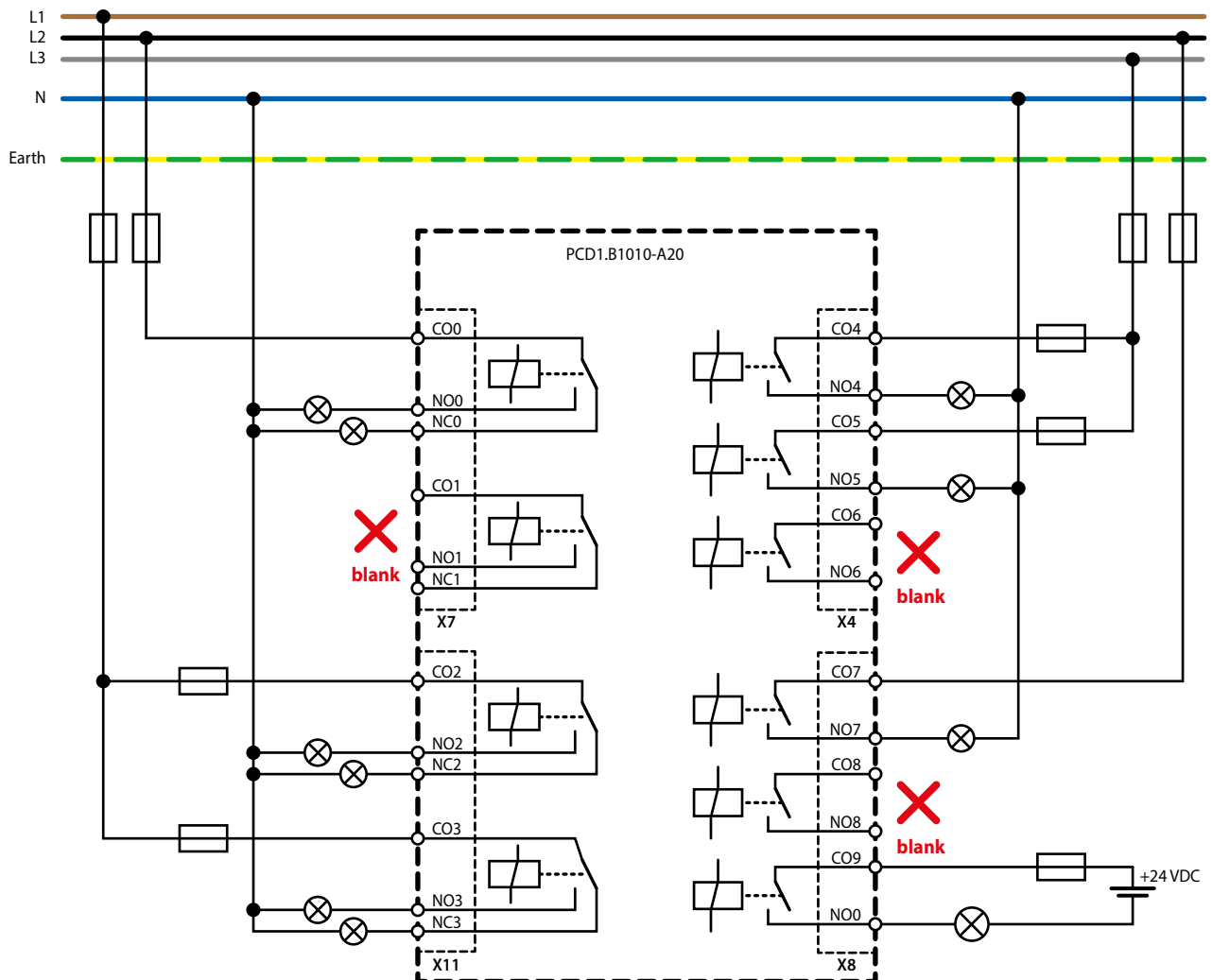
ATTENTION

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

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.**

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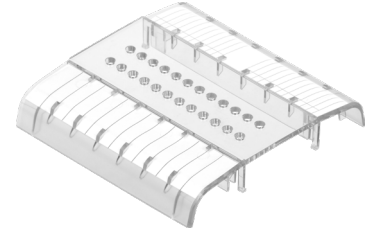The product should not be disposed of with other household waste. Check for the nearest authorized collection centers or authorized recyclers. The correct disposal of end-of-life equipment will help prevent potential negative consequences for the environment and human health.

PCD1.B1010-A20
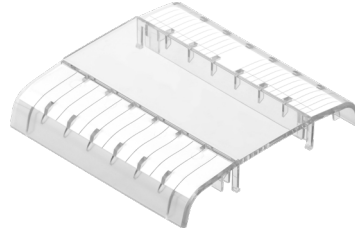
PCD1.K0206-005

PCD1.K0206-025

Terminal set
32304321-003-S

Order details

Type	Short description	Description	Weight
PCD1.B1010-A20	E-Line RIO 24DI, 10Rel	E-Line digital input/output module Manual priority operating level for all outputs Status LED for inputs and outputs Supply 24 VDC 24 digital inputs; 24 VDC (source operation) 6 relay normally open 250 VAC / 30 VDC, 4 A (DC1) 4 relay changeover 250 VAC / 30 VDC, 4 A (DC1) 1 interface RS-485 (S-Bus)	385 g
PCD1.K0206-005	E-Line labelling set 5 × 6 HP*	E-Line cover and labelling set consisting of 5 × covers (6 HP = 105 mm) and labelling sheet for mounting in the automation control cabinet	365 g
PCD1.K0206-025	E-Line labelling set 5 × 6 HP* with holes	E-Line cover and labelling set with holes consisting of 5 × covers (6 HP = 105 mm) with holes for manual override operating level and labelling sheet for mounting in the automation control cabinet	365 g
32304321-003-S	Terminal set	6-pin terminal. Set of 6 terminal blocks	40 g

* Horizontal pitch: 1 HP corresponds to 17.5 mm