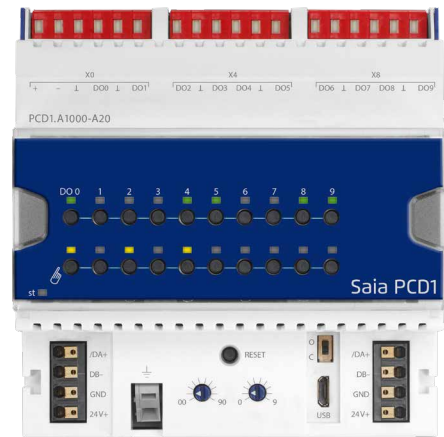


PCD1.E1000-A10

E-Line S-Serie RIO 12DI



The S-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 EN 60715 TH35 enables the use 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 using the access to the manual override by the web interface in the Saia PCD® controller. 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, a complete documentation is available from 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
- Easy programming using the FBox library and web templates
- Industrial hardware in accordance with IEC EN 61131-2
- Pluggable terminal blocks
- Bridge connectors for power supply and communication
- Bus termination on board
- Configurable Bi-Color LEDs and labelling for I/Os

* 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)
Power consumption	1.2 ... 3 W
Power supply bridge	24 VDC, 5 A max., up to 40 modules

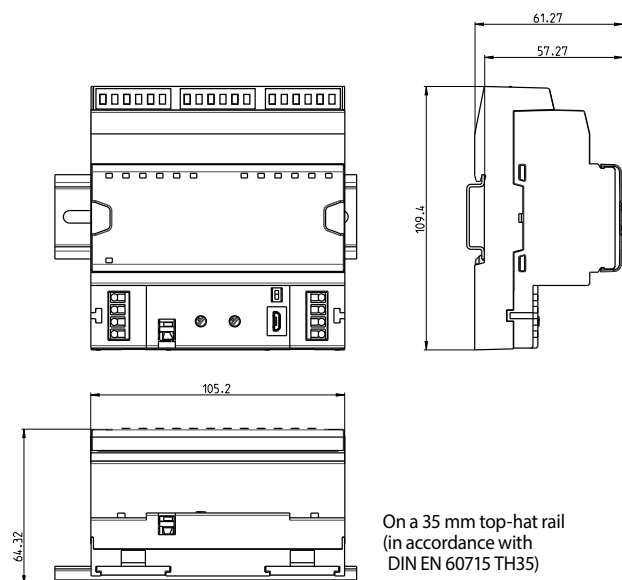
Interfaces

Communications interface	RS-485 Baud rate: 9,600, 19,200, 38,400, 57,600, 115,200 bps (Autobauding) Micro USB, Type B
Address switch	Two rotary switches 0 ... 9 Address range 0 ... 98
Bus termination	Integrated switch to activate and inactivate resistor termination

General data

Ambient temperature	Operation: 0 ... +55 °C Storage: -40 ... +70 °C
Protection class	IP 20
Package	Single carton package with 1 Module incl. terminal blocks, 1 bridge connector

Dimensions and installation

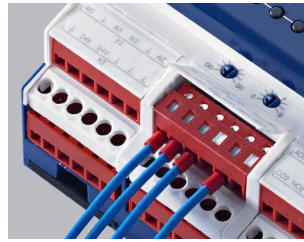


On a 35 mm top-hat rail
(in accordance with
DIN EN 60715 TH35)

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

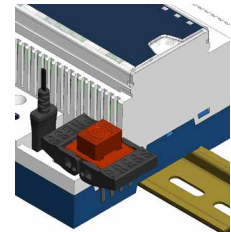
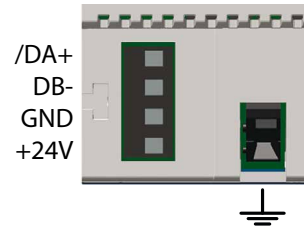
Terminal technology

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.



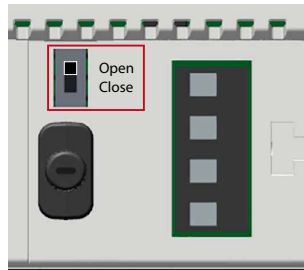
Connection concept

For easy installation the power supply and communication bus is available together at one connector. The push-in spring terminals enable wiring as well support the connector bridge.



Bus termination

The module provides an active bus termination. It is switched off by factory default. To enable the termination, the switch need to be in the "Close" position.



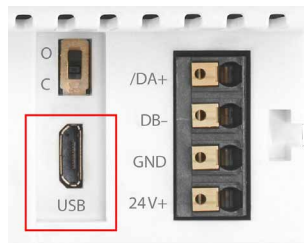
Status LED

OFF	No Power
Green	Communication OK
Green blink	Auto bauding in progress
Orange	No communication
Red	Error
Red/Green alternate	Booter mode (e.g. during Firmware download)
Red blink	Internal fatal error



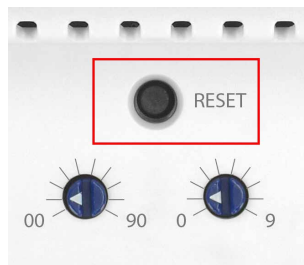
Service interface

The USB interface provides access to the Modbus configuration. Firmware updates can also be downloaded via Saia®PG5 Firmware Download tool.



Reset button

Pushed at power up: Module stays in Boot mode.
Pushed over 5 seconds: Reset to factory default.

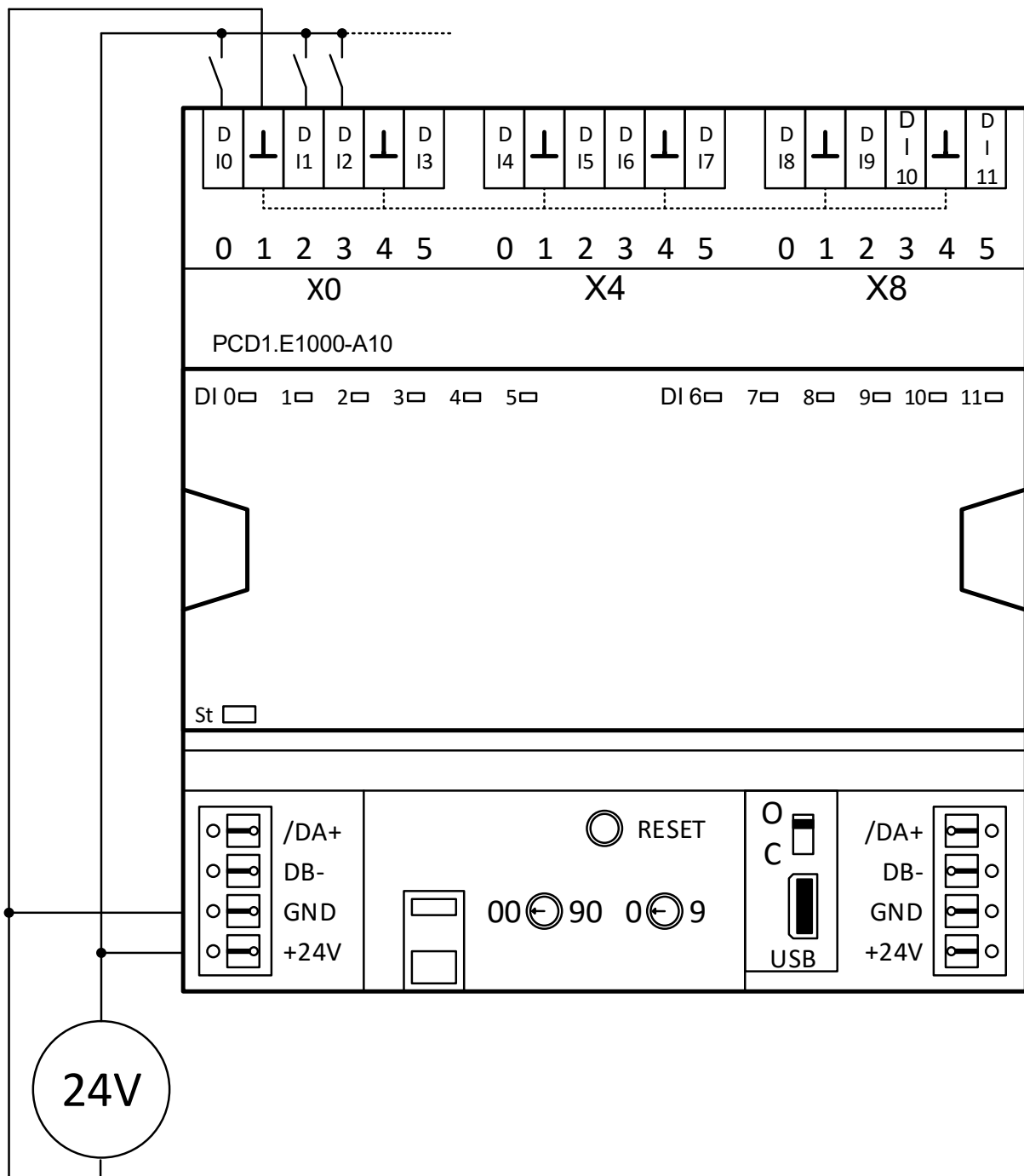


Input/output configuration

Digital inputs

Number	12
Input voltage	24 VDC, source operation (positive switching)
Switching level	Low: 0...5 V, High: 15...24 V
Input current	Typically: 2 mA
Input filter time (DC)	Typically: 8 ms

Assignment overview



LED Signalisation

Status LED

OFF	No Power
Green	Communication OK
Green blink	Auto bauding in progress
Orange	No communication
Red	Error
Red/Green alternate	Booter mode (e.g. during Firmware download)
Red blink	Internal fatal error

Digital input

The Input indication LED can be configured in color and blink code separately for output state Low and High.

LED colour

- ▶ Off
- ▶ Red
- ▶ Green*
- ▶ Orange (red + green)

LED blink code

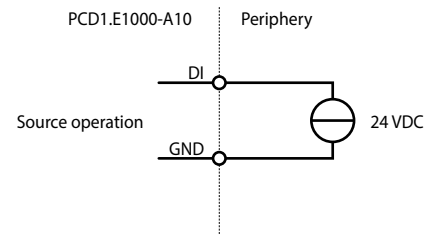
- ▶ No blink*
- ▶ Slow blinking (0.5 flashes per second)
- ▶ Fast blinking (2 flashes per second)

*Factory default

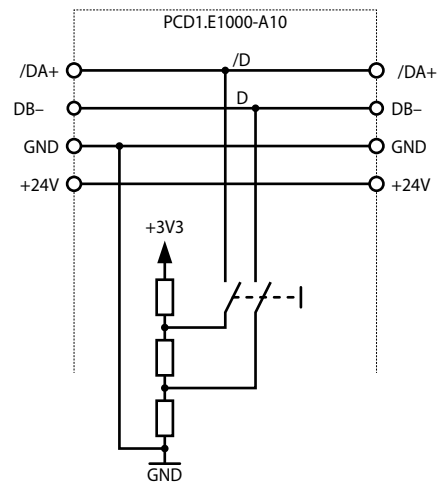
Remarks: In case of error on analogue I/O (overflow), the LED will blink at 1Hz.

Connection diagrams

Digital inputs



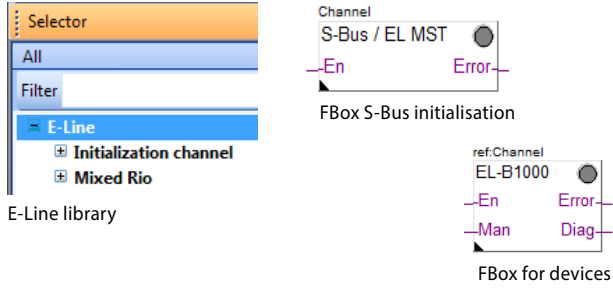
Power supply and bus termination



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



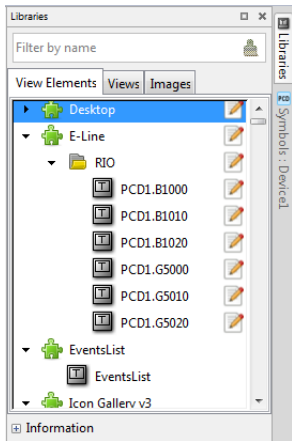
Communication FBox

- ▶ 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

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

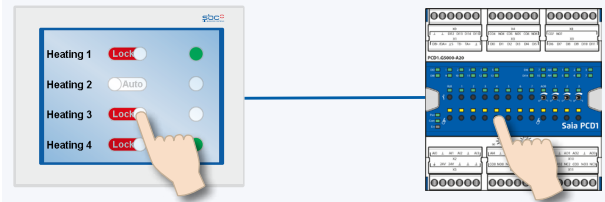
Web templates

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



The inputs / outputs 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.

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:

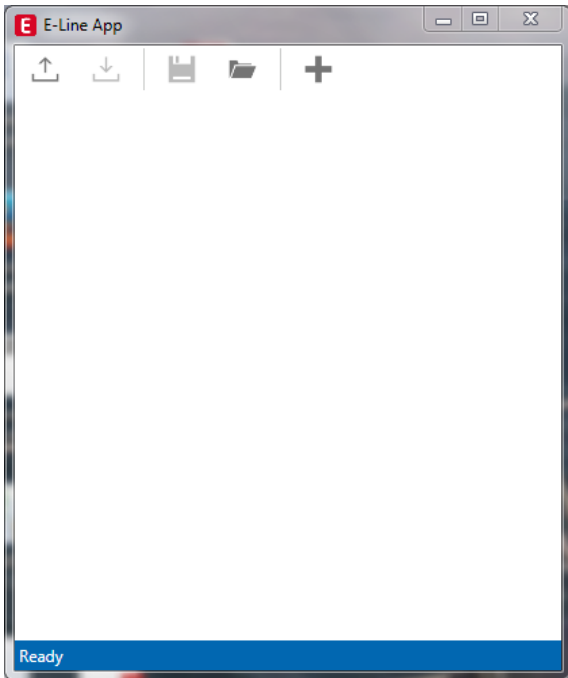
Operating modes	Description	Operation	
		at the module	via remote (S-Bus)
1	Manual operation deactivated	✗	✗
2	Operation permitted from the module only	✓	✗
3	Operation permitted from the module and limited operation from the panel. If manual operation is activated at the module, it cannot be reset from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








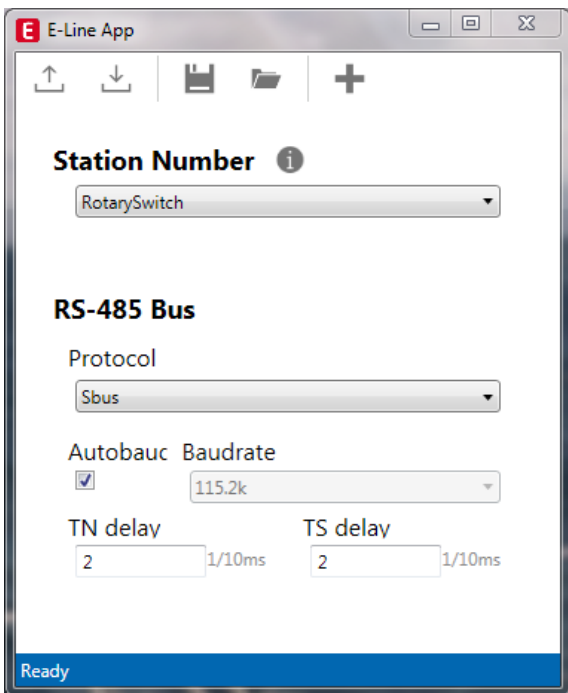
Depending on the application, reset of manually set values is allowed from a panel. To address this requirement, it is possible to deactivate or limit manual operation function.

E-line App device setup

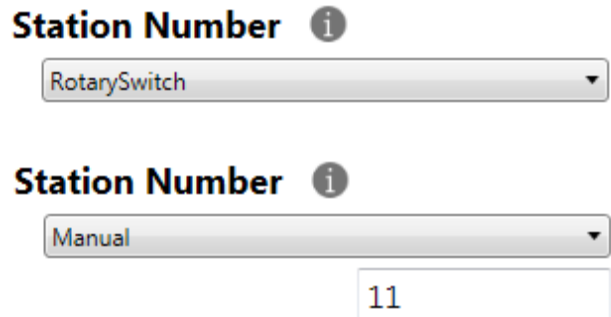
E-Line RIOs support the device setup by a windows application program connected via USB. The installer is available for download from the SBC support page: www.sbc-support.com → E-Line RIO IO Modules.



-  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

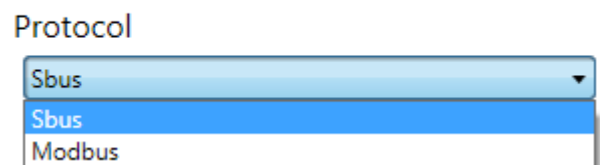


The station number can be set by the rotary switches at the device in the range of 0 ... 98. If the rotary switches are set to position 99 the station number can be defined by the device configuration in a range of 0 ... 253.

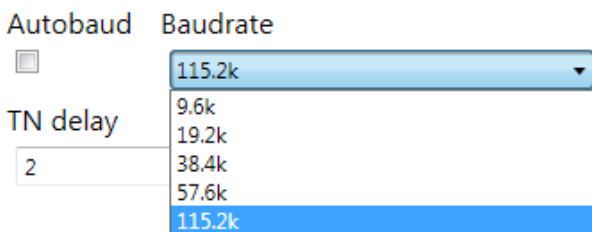


The serial communication protocol can be defined either as SBus or Modbus. By default the modules are delivered from factory with SBus.

RS-485 Bus

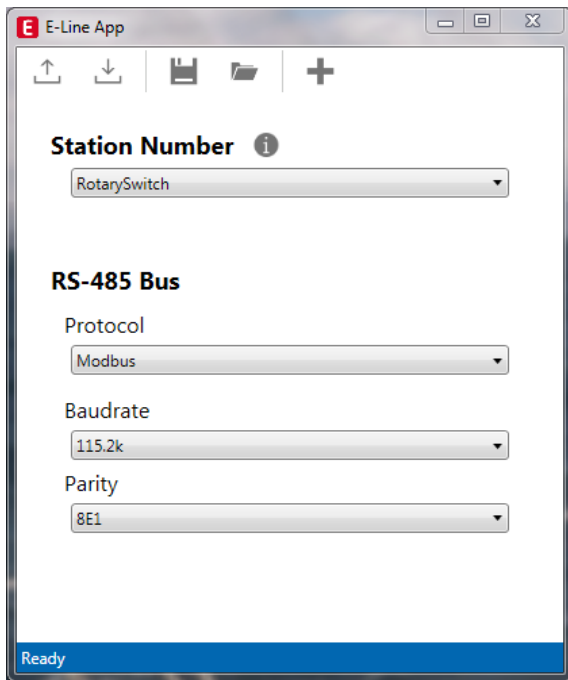


SBus settings



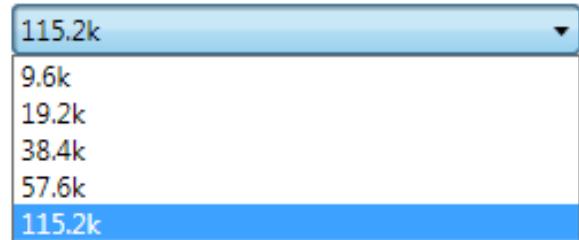
The Baudrate can be defined as automatic detection (default) or set to a specific value. The drop down choice will be available when the check box "Automatic" is unchecked. TN delay and TS delay shall be left at their default values of 2.

Modbus settings



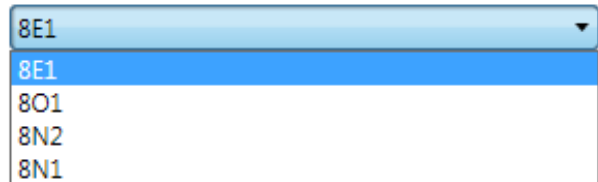
The Baudrate is set by default to 115k. It can be defined as choice of the list.

Baudrate



For best interoperability the Parity Mode and number of Stop Bits can also be set.

Parity



S-Bus communication

SBus communication is based on Saia PCD® SBus 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 by the rotary switches at the front of 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 none 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
 - Is no telegram received after reboot within the “safe state power-on timeout” the module enters as will 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 followed 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 allow to define and configure 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 does use the fast 'E-Line S-Bus' protocol for a high speed communication between the master and the RIO.

The image displays three parts of the software interface:

- Left Panel:** A tree view of 'E-Line S-Bus' devices. Under 'Mixed Rio', the device 'EL-PCD1.E1000' is selected.
- Middle Panel:** A wiring diagram showing connections for 'Channel S-Bus / EL MST', 'EL_Diagnostic.ref.Channel', and 'RIO_9.ref.Channel'. Each connection has a 'start' button and 'En' and 'Error' indicators.
- Right Panel:** The 'Properties' window for 'FBox : EL-PCD1.E1000'. It shows configuration parameters:

General	
(Name)	RIO_9
Reference	Channel
Comment	
Adjust Variables	
S-Bus address	9
Comm interval inputs/outputs	On each cycle
Diagnostic:	
Up/download configurations:	
Led configurations:	
Led frequency & color DI 0	20000
Led frequency & color DI 1	20000
Led frequency & color DI 2	20000
Led frequency & color DI 3	20000
Led frequency & color DI 4	20000
Led frequency & color DI 5	20000
Led frequency & color DI 6	20000
Led frequency & color DI 7	20000
Led frequency & color DI 8	20000
Led frequency & color DI 9	20000
Led frequency & color DI 10	20000
Led frequency & color DI 11	20000
Comment	

S-Bus communication

Direct access to the RIO medias 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) require standard SBus communication.

Digital inputs

Input	Input Value	Read/Write
Digital input 0	Flag 0	R
Digital input 1	Flag 1	R
Digital input 2	Flag 2	R
Digital input 3	Flag 3	R
Digital input 4	Flag 4	R
Digital input 5	Flag 5	R
Digital input 6	Flag 6	R
Digital input 7	Flag 7	R
Digital input 8	Flag 8	R
Digital input 9	Flag 9	R
Digital input 10	Flag 10	R
Digital input 11	Flag 11	R

LED Configuration

Relay output 0	Flag 330	RW
Relay output 1	Flag 331	RW
Relay output 2	Flag 332	RW
Relay output 3	Flag 333	RW
Relay output 4	Flag 334	RW
Relay output 5	Flag 335	RW
Relay output 6	Flag 336	RW
Relay output 7	Flag 337	RW
Relay output 8	Flag 338	RW
Relay output 9	Flag 339	RW
Relay output 10	Flag 340	RW
Relay output 11	Flag 341	RW

Register format:

Bit 0 ... 7	I/O state Low	LED color
Bit 8 ... 15	I/O state Low	LED blink code
Bit 16 ... 23	I/O state High	LED color
Bit 24 ... 32	I/O state High	LED blink code

LED color 0: Off

- 1: Red
- 2: Green
- 3: Orange (red + green)

LED blink code

- 0: No blink
- 1: Slow blinking (0.5 flashes per second)
- 2: Fast blinking (2 flashes per second)

Factory default: Low: off, High: LED color 2 (green), no blink
Remarks: In case of error on analogue I/O (overflow),
the LED will blink at 1 Hz

The LEDs can be configured individually depending on the I/O state in colour and blink code.

S-Bus communication

Device Information

Firmware version (Decimal xyzzyz, 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:SBus 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 SBus 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 SBus 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 modules. 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
 - Is no telegram received after reboot within the "safe state power-on timeout" the module enters as will 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 followed 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)

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		Coil 0 ... 7	Coil 8 ... 15	CRC	
0 ... 254	1	No. of Byte		Coil 0 ... 7	Coil 8 ... 15	High-Byte	Low-Byte

Write coils

Request									
Address	Function	Start Address		Number of Coils to write		Coil		CRC	
0 ... 254	15	High-Byte	Low-Byte	High-Byte	Low-Byte	0 ... 7	Coil 0 ... 7	High-Byte	Low-Byte

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

Read register

Request							
Address	Function	Start Address		Number 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		CRC	
0 ... 254	3	0 ... 256		High-Byte	Low-Byte	High-Byte	Low-Byte

Write register

Request								
Address	Function	Start Address		No. of Bytes	Data Words		CRC	
0 ... 254	3	High-Byte	Low-Byte	0 ... 256	Low-Byte	High-Byte	High-Byte	Low-Byte

Reply							
Address	Function	Start Address		No of Registers written		CRC	
0 ... 254	3	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 230V

Input	Input Value	Read/Write
Digital input 0	Flag 0	R
Digital input 1	Flag 1	R
Digital input 2	Flag 2	R
Digital input 3	Flag 3	R
Digital input 4	Flag 4	R
Digital input 5	Flag 5	R
Digital input 6	Flag 6	R
Digital input 7	Flag 7	R
Digital input 8	Flag 8	R
Digital input 9	Flag 9	R
Digital input 10	Flag 10	R
Digital input 11	Flag 11	R

LED Configuration

LED Digital input	Output L, Reg. / Output H, Reg.	Read/Write
LED Digital input 0	Output L, Reg. 660 Output H, Reg. 661	RW
LED Digital input 1	Output L, Reg. 662 Output H, Reg. 663	RW
LED Digital input 2	Output L, Reg. 664 Output H, Reg. 665	RW
LED Digital input 3	Output L, Reg. 666 Output H, Reg. 667	RW
LED Digital input 4	Output L, Reg. 668 Output H, Reg. 669	RW
LED Digital input 5	Output L, Reg. 670 Output H, Reg. 671	RW
LED Digital input 6	Output L, Reg. 672 Output H, Reg. 673	RW
LED Digital input 7	Output L, Reg. 674 Output H, Reg. 675	RW
LED Digital input 8	Output L, Reg. 676 Output H, Reg. 677	RW
LED Digital input 9	Output L, Reg. 678 Output H, Reg. 679	RW
LED Digital input 10	Output L, Reg. 680 Output H, Reg. 681	RW
LED Digital input 11	Output L, Reg. 682 Output H, Reg. 683	RW

Register format:

Output L, Bit 0 ... 7	I/O state Low	LED color
Output L, Bit 8 ... 15	I/O state Low	LED blink code
Output H, Bit 16 ... 23	I/O state High	LED color
Output H, Bit 24 ... 32	I/O state High	LED blink code

LED color

- 0: Off
- 1: Red
- 2: Green
- 3: Orange (red + green)

LED blink code

- 0: No blink
- 1: Slow blinking (0.5 flashes per second)
- 2: Fast blinking (2 flashes per second)

Factory default: Low: off, High: LED colour 2 (green), no blink

Remarks: In case of error on analogue I/O (overflow), the led will blink at 1 Hz

The LEDs can be configured individually depending on the I/O state in colour and blink code.

Device Information

Firmware version (Decimal xyzzyz, 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: SBus 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.
Eg. for PCD1.A2000-A20:
1210...1217: 5043H | 4431H | 2E41H | 3230H | 3030H | 2D41H | 3230H | 0000H

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--)>0                             /* pass through message buffer */
    {
        uIndex = uchCRCHi ^ *puchMsg++;                /* calculate the CRC */
        uchCRCHi = uchCRCLo ^ uchCRCHi[uIndex];
        uchCRCLo = uchCRCHi[uIndex];
    }
    return (uchCRCHi << 8 | uchCRCLo);
}
```

High-Order Byte Table

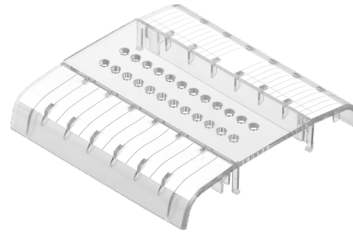
```
/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41,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

Low-Order Byte Table

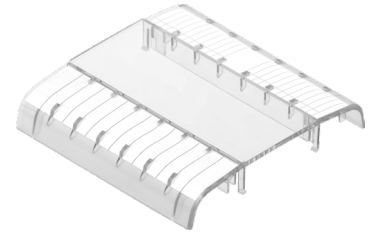
```
/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04,
0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,
0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
0x14, 0x04, 0x05, 0x15, 0x07, 0x17, 0x16, 0x06, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0x10, 0xD0,
0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0xF4, 0x34,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0x38, 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, 0x83, 0x41, 0x81, 0x80, 0x40 };
```



PCD1.E1000-A10



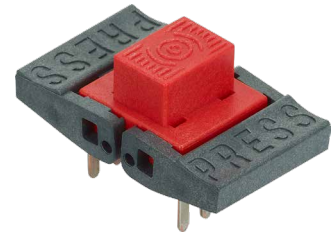
PCD1.K2026-005



PCD1.K2026-025



Terminal set



Connector bridge

Order details

Type	Short description	Description	Weight
PCD1.E1000-A10	E-Line S-Serie RIO 12DI	E-Line S-Serie Digital input module status LED for inputs supply 24 VDC 12 Digital inputs 24 VDC (source operation) 1 interface RS-485 (S-Bus and Modbus) 1 USB Service interface	180 g
PCD1.K2026-005	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	50 g
PCD1.K2026-025	E-Line labelling set 5 × 6 HP w. h.	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	50 g
In preparation	Terminal set	6-pin terminal. Set of 6 terminal blocks	40 g
In preparation	Connector bridge	Set of 10 connector bridges for interconnection of power supply and communication bus.	100 g

Saia-Burgess Controls AG

Bahnhofstrasse 18 | 3280 Murten, Switzerland
T +41 26 580 30 00 | F +41 26 580 34 99
www.saia-pcd.com

support@saia-pcd.com | www.sbc-support.com