USER MANUAL

Longo programmable controller
LPC-2.CX1
Communication module

Version 2

SMARTEH d.o.o. / Poljubinj 114 / 5220 Tolmin / Slovenia / Tel.: +386(0)5 388 44 00 / e-mail: info@smarteh.si / www.smarteh.si

Written by SMARTEH d.o.o.

User Manual

Document Version: 2
March, 2023

STANDARDS AND PROVISIONS: Standards, recommendations, regulations and provisions of the country in which the devices will operate, must be considered while planning and setting up electrical devices. Work on 100 .. 240 V AC network is allowed for authorized personnel only.

DANGER WARNINGS: Devices or modules must be protected from moisture, dirt and damage during transport, storing and operation.

WARRANTY CONDITIONS: For all modules LONGO LPC-2 – if no modifications are performed upon and are correctly connected by authorized personnel – in consideration of maximum allowed connecting power, warranty of 24 months is valid from the date of sale to the end buyer, but not more than 36 months after delivery from Smarteh. In case of claims within warranty time, which are based on material malfunctions the producer offers free replacement. The method of return of malfunctioned module, together with description, can be arranged with our authorized representative. Warranty does not include damage due to transport or because of unconsidered corresponding regulations of the country, where the module is installed.
This device must be connected properly by the provided connection scheme in this manual. Misconnections may result in device damage, fire or personal injury.
Hazardous voltage in the device can cause electric shock and may result in personal injury or death.
NEVER SERVICE THIS PRODUCT YOURSELF!
This device must not be installed in the systems critical for life (e.g. medical devices, aircrafts, etc.).

If the device is used in a manner not specified by the manufacturer, the degree of protection provided by the equipment may be impaired.

Waste electrical and electronic equipment (WEEE) must be collected separately!

LONGO LPC-2 complies to the following standards:

- EMC: EN 61000-6-3:2007 + A1:2011, EN 61000-6-1:2007, EN 61000-3-2:2006 + A1:2009 + A2: 2009, EN 61000-3-3:2013
- LVD: IEC 61010-1:2010 (3rd Ed.), IEC 61010-2-201:2013 (1st Ed.)

Smarteh d.o.o. operates a policy of continuous development. Therefore we reserve the right to make changes and improvements to any of the products described in this manual without any prior notice.

MANUFACTURER:
SMARTEH d.o.o.
Poljubinj 114
5220 Tolmin
Slovenia

Index

Longo programmable controller LPC-2.CX1

1 ABBREVIATIONS

DC	Direct Current
RX	Receive
TX	Transmit
UART	Universal Asynchronous Receiver-Transmitter

2 DESCRIPTION

LPC-2.CX1 is a communication module with a port for RS-485 communication. It's main feature is being a gateway for Modbus RTU or other UART protocols, exchanging data with the main module, where all of data is being interpreted. It also equipped with a MP3 player and an audio amplifier for an external speaker connection.

LPC-2.CX1 is controlled and powered from the main module (e.g. LPC-2.MU1, LPC-2.MC9, ...) via Right internal bus.

3 FEATURES

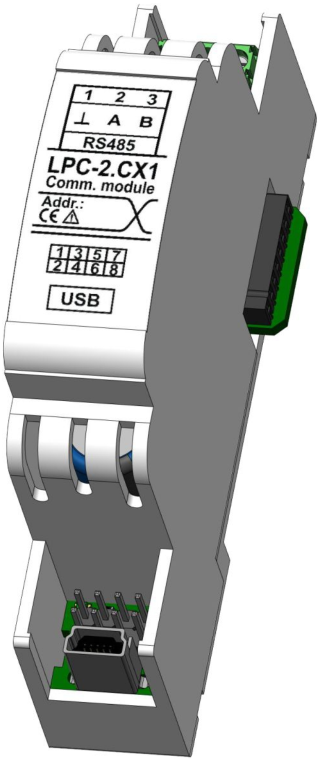


Figure 1: LPC-2.CX1 module

Table 1: Technical data
RS-485 master port for Modbus RTU or other UART protocols
MP3 player
10 W audio amplifier
Internal 4 MB Flash
USB for external memory or internal flash loading
Signal LEDs
Supplied from main module
Small dimensions and standard DIN EN50022-35 rail mounting

4 OPERATION

LPC-2.CX1 module can be controlled from main PLC module (e.g. LPC-2.MC9). Module parameters can be read or written via Smarteh IDE software. To use the LPC-2.CX1 module in your Smarteh IDE programming software configuration, you must select the universal module LPC-2.Cxx.
LPC-2.CX1 module can also be controlled by the Modbus RTU Slave main module (e.g. LPC-2.MU1).

4.1 SmartehIDE Parameters

Input

Receive data [i_Rx_Data_{1..298}]: Generic receive data.

Type: USINT

Raw to engineering data: 0 .. 255 → 0 .. 255

Output

Transmit data [o_Tx_Data _{1..298}]: Generic transmit data.

Type: USINT

Raw to engineering data: 0 .. 255 → 0 .. 255

4.2 Modbus RTU and SmartehIDE Parameters - Modbus RTU mode

By specifying the first parameter *Protocol type*, Parameter number 1, Modbus RTU holding register 93 High byte (93.H), determine the type of communication on the RS-485 communication port. Based on this parameter, the content of further communication parameters is determined. The parameters are described in the tables below.

If protocol type Modbus RTU master driver is selected use the tables in this chapter numbered from 2 to 21.

In the first column named *Reg.* are Modbus RTU registers. These are used for controlling LPC-2.CX1 module from the Modbus RTU Slave main module (e.g. LPC-2.MU1).

In second column named *Par.* there are SmartehIDE parameter indexes. These are used for controlling LPC-2.CX1 module from the main module (e.g. LPC-2.MC9).

Table 2: Communication settings, Modbus RTU mode
4xxxx, Modbus RTU Holding registers
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
93.L[1]	1	Protocol_Type	Communication protocol type	0 → Universal UART 1 → Modbus RTU master driver
93.H[2]	2	Protocol_Option	Protocol option	Not used
94.L	3	UART_Speed	Selector of baudrate	0 → 300 1 → 600 2 → 1200 3 → 2400 4 → 4800 5 → 9600 6 → 19200 7 → 38400 8 → 57600 9 → 115200 other → 19200
94.H	4	UART_Parity	Selector of parity	0 → No parity bit 1 → Odd 2 → Even other → No parity bit
95.L	5	UART_Databits	Number of data bits	5 → 5 bits 6 → 6 bits 7 → 7 bits 8 → 8 bits other → 8 bits
95.H	6	UART_Stopbits	Number of stop bits	0 → 1 bit 1 → 1 bit 2 → 2 bits other → 2 bits

[1] Low byte
[2] High byte

Table 2: Communication settings, Modbus RTU mode 4xxxx, Modbus RTU Holding registers SmartehIDE output parameters

96	7,8	Rx_TLG_Timeout	Time delay between Modbus telegrams	0 → No timeout 1 .. 65535 → 1 .. 65535 ms
97	9,10	Rx_Data_Timeout	Timeout waiting for answer for Modbus telegrams	0 → No timeout 1 .. 65535 → 1 .. 65535 ms
98	11,12	Not used	Not used	
99	13,14	Tx_SoundCard_SendCommand	Transmit sound card data only when counter changes	1 .. 65535 → 1 .. 65535
100.L	15	Tx_SendCommand	Transmit data only when counter changes	0 → Continuous communication 255 → Communication stopped 1 .. 254 → 1 .. 254

Table 3: Communication feedback, Modbus RTU mode 3xxxx, Modbus RTU input registers SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
103.L	1	Protocol_Type_FBD	Communication protocol type feedback	0 → Universal UART 1 → Modbus RTU master driver
103.H	2	I2C_Session_Counter	It changes after each session, when data in module updates. It updates every cca.100 ms.	0 .. 255 → 0 .. 255
104.L	3	Tx_Counter	It increments when new Modbus or UART data were transmitted	0 .. 255 → 0 .. 255
104.H 105.L	4,5	Tx_Error_Counter	Number of failed communication telegrams	0 .. 65535 → 0 .. 65535
105.H	6	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Error 3 → Timeout
106.L	7	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed
106.H 107.L	8,9	SoundCardResponse_Counter	Increments when sound card response is received	0 .. 65535 → 0 .. 65535

Table 4: TxBuffer, MBS1 Head, Modbus RTU mode 4xxxx, Modbus RTU Holding registers SmartehIDE output parameters				
Reg.	**Par.**	**Name**	**Description**	**Raw → Engineering data**
101.L	17	Slave_Address	Modbus RTU port slave address	0 → Not active 1 .. 255 → SlaveID1 .. SlaveID 255
101.H	18	Modbus_Function	Modbus function codes	1 → Read Coils 2 → Read Discrete inputs 3 → Read Holding registers 4 → Read Input register 15 → Write Multiple Coils 16 → Write Holding registers
102	19,20	Register_Address	Modbus Slave register address	0 .. 65535 → 0 .. 65535
103.L	21	Data_Length	Number of words/bits. Depends on selected function.	0 .. 255 → 0 .. 255
103.H	22	Not used	Not used	
104	23,24	Buffer_Start	Data buffer start. Write commands are related to Tx buffer, Read commands to Rx buffer.	81 .. 286 → 81 .. 286 (Tx) 41 .. 286 → 41 .. 286 (Rx)

Table 5: TxBuffer, MBS2 Head, Modbus RTU mode 4xxxx, Modbus RTU Holding registers SmartehIDE output parameters				
Reg.	**Par.**	**Name**	**Description**	**Raw → Engineering data**
105.L	25	Slave_Address	Modbus RTU port slave address	0 → Not active 1 .. 255 → SlaveID1 .. SlaveID 255
105.H	26	Modbus_Function	Modbus function codes	1 → Read Coils 2 → Read Discrete inputs 3 → Read Holding registers 4 → Read Input register 15 → Write Multiple Coils 16 → Write Holding registers
106	27,28	Register_Address	Modbus Slave register address	0 .. 65535 → 0 .. 65535
107.L	29	Data_Length	Number of words/bits. Depends on selected function.	0 .. 255 → 0 .. 255
107.H	30	Not used	Not used	
108	31,32	Buffer_Start	Data buffer start. Write commands are related to Tx buffer, Read commands to Rx buffer.	81 .. 286 → 81 .. 286 (Tx) 41 .. 286 → 41 .. 286 (Rx)

Table 6: TxBuffer, MBS3 Head, Modbus RTU mode
4xxxx, Modbus RTU Holding registers
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
109.L	33	Slave_Address	Modbus RTU port slave address	0 → Not active 1 .. 255 → SlaveID1 .. SlaveID 255
109.H	34	Modbus_Function	Modbus function codes	1 → Read Coils 2 → Read Discrete inputs 3 → Read Holding registers 4 → Read Input register 15 → Write Multiple Coils 16 → Write Holding registers
110	35,36	Register_Address	Modbus Slave register address	0 .. 65535 → 0 .. 65535
111.L	37	Data_Length	Number of words/bits. Depends on selected function.	0 .. 255 → 0 .. 255
111.H	38	Not used	Not used	
112	39,40	Buffer_Start	Data buffer start. Write commands are related to Tx buffer, Read commands to Rx buffer.	81 .. 286 → 81 .. 286 (Tx) 41 .. 286 → 41 .. 286 (Rx)

Table 7: TxBuffer, MBS4 Head, Modbus RTU mode
4xxxx, Modbus RTU Holding registers
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
113.L	41	Slave_Address	Modbus RTU port slave address	0 → Not active 1 .. 255 → SlaveID1 .. SlaveID 255
113.H	42	Modbus_Function	Modbus function codes	1 → Read Coils 2 → Read Discrete inputs 3 → Read Holding registers 4 → Read Input register 15 → Write Multiple Coils 16 → Write Holding registers
114	43,44	Register_Address	Modbus Slave register address	0 .. 65535 → 0 .. 65535
115.L	45	Data_Length	Number of words/bits. Depends on selected function.	0 .. 255 → 0 .. 255
115.H	46	Not used	Not used	
116	47,48	Buffer_Start	Data buffer start. Write commands are related to Tx buffer, Read commands to Rx buffer.	81 .. 286 → 81 .. 286 (Tx) 41 .. 286 → 41 .. 286 (Rx)

Table 8: TxBuffer, MBS5 Head, Modbus RTU mode
4xxxx, Modbus RTU Holding registers
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
117.L	49	Slave_Address	Modbus RTU port slave address	0 → Not active 1 .. 255 → SlaveID1 .. SlaveID 255
117.H	50	Modbus_Function	Modbus function codes	1 → Read Coils 2 → Read Discrete inputs 3 → Read Holding registers 4 → Read Input register 15 → Write Multiple Coils 16 → Write Holding registers
118	51,52	Register_Address	Modbus Slave register address	0 .. 65535 → 0 .. 65535
119.L	53	Data_Length	Number of words/bits. Depends on selected function.	0 .. 255 → 0 .. 255
119.H	54	Not used	Not used	
120	55,56	Buffer_Start	Data buffer start. Write commands are related to Tx buffer, Read commands to Rx buffer.	81 .. 286 → 81 .. 286 (Tx) 41 .. 286 → 41 .. 286 (Rx)

Table 9: TxBuffer, MBS6 Head, Modbus RTU mode
4xxxx, Modbus RTU Holding registers
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
121.L	57	Slave_Address	Modbus RTU port slave address	0 → Not active 1 .. 255 → SlaveID1 .. SlaveID 255
121.H	58	Modbus_Function	Modbus function codes	1 → Read Coils 2 → Read Discrete inputs 3 → Read Holding registers 4 → Read Input register 15 → Write Multiple Coils 16 → Write Holding registers
122	59,60	Register_Address	Modbus Slave register address	0 .. 65535 → 0 .. 65535
123.L	61	Data_Length	Number of words/bits. Depends on selected function.	0 .. 255 → 0 .. 255
123.H	62	Not used	Not used	
124	63,64	Buffer_Start	Data buffer start. Write commands are related to Tx buffer, Read commands to Rx buffer.	81 .. 286 → 81 .. 286 (Tx) 41 .. 286 → 41 .. 286 (Rx)

Table 10: TxBuffer, MBS7 Head, Modbus RTU mode
4xxxx, Modbus RTU Holding registers,
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
125.L	65	Slave_Address	Modbus RTU port slave address	0 → Not active 1 .. 255 → SlaveID1 .. SlaveID 255
125.H	66	Modbus_Function	Modbus function codes	1 → Read Coils 2 → Read Discrete inputs 3 → Read Holding registers 4 → Read Input register 15 → Write Multiple Coils 16 → Write Holding registers
126	67,68	Register_Address	Modbus Slave register address	0 .. 65535 → 0 .. 65535
127.L	69	Data_Length	Number of words/bits. Depends on selected function.	0 .. 255 → 0 .. 255
127.H	70	Not used	Not used	
128	71,72	Buffer_Start	Data buffer start. Write commands are related to Tx buffer, Read commands to Rx buffer.	81 .. 286 → 81 .. 286 (Tx) 41 .. 286 → 41 .. 286 (Rx)

Table 11: TxBuffer, MBS8 Head, Modbus RTU mode
4xxxx, Modbus RTU Holding registers
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
129.L	73	Slave_Address	Modbus RTU port slave address	0 → Not active 1 .. 255 → SlaveID1 .. SlaveID 255
129.H	74	Modbus_Function	Modbus function codes	1 → Read Coils 2 → Read Discrete inputs 3 → Read Holding registers 4 → Read Input register 15 → Write Multiple Coils 16 → Write Holding registers
130	75,76	Register_Address	Modbus Slave register address	0 .. 65535 → 0 .. 65535
131.L	77	Data_Length	Number of words/bits. Depends on selected function.	0 .. 255 → 0 .. 255
131.H	78	Not used	Not used	
132	79,80	Buffer_Start	Data buffer start. Write commands are related to Tx buffer, Read commands to Rx buffer.	81 .. 286 → 81 .. 286 (Tx) 41 .. 286 → 41 .. 286 (Rx)

Table 12: TxBuffer, Data, Modbus RTU mode 4xxxx, Holding registers SmartehIDE output parameters				
Reg.	**Par.**	**Name**	**Description**	**Raw → Engineering data**
133.L	81	Tx_Modbus_Buffer_Start	Start of transmit Modbus buffer	0 .. 65535 → 0 .. 65535
...	...			
235.H	286	Tx_Modbus_Buffer_End	End of transmit Modbus buffer	0 .. 65535 → 0 .. 65535
236.L	287	Tx_Sound_Buffer_Start	Start of transmit sound card Modbus buffer	0 .. 65535 → 0 .. 65535
...	...			
241.H	298	Tx_Sound_Buffer_End	End of transmit sound card Modbus buffer	0 .. 65535 → 0 .. 65535

Table 13: RxBuffer, MBS1 Head, Modbus RTU mode 3xxxx, Modbus RTU Input registers SmartehIDE input parameters				
Reg.	**Par.**	**Name**	**Description**	**Raw → Engineering data**
109.L	13	Rx_Counter	Number of successful transactions	0 .. 255 → 0 .. 255
109.H	14	Rx_Error_Counter	Number of failed transactions	0 .. 255 → 0 .. 255
110.L	15	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Modbus error 3 → Timeout 4 → Wrong parameters
110.H	16	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed

Table 14: RxBuffer, MBS2 Head, Modbus RTU mode
3xxxx, Modbus RTU Input registers
SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
111.L	17	Rx_Counter	Number of successful transactions	0 .. 255 → 0 .. 255
111.H	18	Rx_Error_Counter	Number of failed transactions	0 .. 255 → 0 .. 255
112.L	19	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Modbus error 3 → Timeout 4 → Wrong parameters
112.H	20	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed

Table 15: RxBuffer, MBS3 Head, Modbus RTU mode
3xxxx, Modbus RTU Input registers
SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
113.L	21	Rx_Counter	Number of successful transactions	0 .. 255 → 0 .. 255
113.H	22	Rx_Error_Counter	Number of failed transactions	0 .. 255 → 0 .. 255
114.L	23	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Modbus error 3 → Timeout 4 → Wrong parameters
114.H	24	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed

Table 16: RxBuffer, MBS4 Head, Modbus RTU mode
4xxxx, Modbus RTU Input registers
SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
115.L	25	Rx_Counter	Number of successful transactions	0 .. 255 → 0 .. 255
115.H	26	Rx_Error_Counter	Number of failed transactions	0 .. 255 → 0 .. 255
116.L	27	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Modbus error 3 → Timeout 4 → Wrong parameters
116.H	28	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed

Table 17: RxBuffer, MBS5 Head, Modbus RTU mode
3xxxx, Modbus RTU Input registers
SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
117.L	29	Rx_Counter	Number of successful transactions	0 .. 255 → 0 .. 255
117.H	30	Rx_Error_Counter	Number of failed transactions	0 .. 255 → 0 .. 255
118.L	31	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Modbus error 3 → Timeout 4 → Wrong parameters
118.H	32	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed

Table 18: RxBuffer, MBS6 Head, Modbus RTU mode
3xxxx, Modbus RTU Input registers,
SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
119.L	33	Rx_Counter	Number of successful transactions	0 .. 255 → 0 .. 255
119.H	34	Rx_Error_Counter	Number of failed transactions	0 .. 255 → 0 .. 255
120.L	35	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Modbus error 3 → Timeout 4 → Wrong parameters
120.H	36	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed

Table 19: RxBuffer, MBS7 Head, Modbus RTU mode
3xxxx, Modbus RTU Input registers
SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
121.L	37	Rx_Counter	Number of successful transactions	0 .. 255 → 0 .. 255
121.H	38	Rx_Error_Counter	Number of failed transactions	0 .. 255 → 0 .. 255
122.L	39	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Modbus error 3 → Timeout 4 → Wrong parameters
122.H	40	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed

Table 20: RxBuffer, MBS8 Head, Modbus RTU mode 3xxxx, Modbus RTU Input registers SmartehIDE input parameters				
Reg.	**Par.**	**Name**	**Description**	**Raw → Engineering data**
123.L	41	Rx_Counter	Number of successful transactions	0 .. 255 → 0 .. 255
123.H	42	Rx_Error_Counter	Number of failed transactions	0 .. 255 → 0 .. 255
124.L	43	Modbus_Status	Status of last Modbus transaction	0 → OK 1 → Modbus error 3 → Timeout 4 → Wrong parameters
124.H	44	Modbus_Error_Code	Last Modbus transaction error code	0 → None 1 → Illegal function 2 → Illegal data address 3 → Illegal data value 4 → Slave device failure 5 → Acknowledge 6 → Slave busy 7 → Not acknowledge 8 → Memory parity error 10 → Gateway path unavailable 11 → Gateway target failed

Table 21: RxBuffer, Data, Modbus RTU mode 3xxxx, Modbus RTU Input registers SmartehIDE input parameters				
Reg.	**Par.**	**Name**	**Description**	**Raw → Engineering data**
125.L	45	Rx Modbus_Buffer_Start	Start of receive Modbus buffer	0 .. 65535 → 0 .. 65535
…	…			
245.H	286	Rx_Modbus_Buffer_End	End of receive Modbus buffer	0 .. 65535 → 0 .. 65535
246.L	287	Rx_Sound_Buffer_Start	Start of receive sound card Modbus buffer	0 .. 65535 → 0 .. 65535
…	…			
251.H	298	Rx_Sound_Buffer_End	End of receive sound card Modbus buffer	0 .. 65535 → 0 .. 65535

4.3 Modbus RTU and SmartehIDE Parameters - Universal mode

By specifying the first parameter Protocol type (Parameter number 1, Modbus RTU holding register 93.H), determine the type of communication on the RS-485 communication port. Based on this parameter, the content of further communication parameters is determined. The parameters are described in the tables below.

If protocol type Universal *UART* is selected use the tables in this chapter numbered from 22 to 26.

In first column named *Reg.* are Modbus RTU registers. These are used for controlling LPC-2.CX1 module from the Modbus RTU Slave main module (e.g. LPC-2.MU1).

In second column named *Par.* there are SmartehIDE parameter indexes. These are used for controlling LPC-2.CX1 module from the main module (e.g. LPC-2.MC9).

Table 23: Communication settings, Universal mode
4xxxx, Modbus RTU Holding registers
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
93.L	1	Protocol_Type	Communication protocol type	0 → Universal UART 1 → Modbus RTU master driver
93.H	2	Protocol_Option	Protocol option	Not used
94.L	3	UART_Speed	Selector of baudrate	0 → 300 1 → 600 2 → 1200 3 → 2400 4 → 4800 5 → 9600 6 → 19200 7 → 38400 8 → 57600 9 → 115200 other → 19200
94.H	4	UART_Parity	Selector of parity	0 → No parity bit 1 → Odd 2 → Even other → No parity bit
95.L	5	UART_Databits	Number of data bits	5 → 5 bits 6 → 6 bits 7 → 7 bits 8 → 8 bits other → 8 bits
95.H	6	UART_Stopbits	Number of stop bits	0 → 1 bit 1 → 1 bit 2 → 2 bits other → 2 bits
96	7,8	Rx_TLG_Timeout	Telegram timeout for receiving data	0 → No timeout 1 .. 65535 → 1 .. 65535 ms
97	9,10	Rx_Data_Timeout	Data timeout for receiving data	0 → No timeout 1 .. 65535 → 1 .. 65535 ms

Table 23: Communication settings, Universal mode
4xxxx, Modbus RTU Holding registers
SmartehIDE output parameters

98	11,12	Rx_Data_Length	Length of received data	0 .. 1000 → 0..1000 bytes other → AUTO (receive data until timeout or buffer is full)
99	13,14	Tx_Data_Length	Number of bytes to transmit	0 .. 270 → 0 .. 270
100.L	15	Tx_SendCommand	Transmit data only when counter changes	0 .. 255 → 0 .. 255

Table 24: Communication feedback, Universal mode
3xxxx, Modbus RTU input registers
SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
103.L	1	Protocol_Type_FBD	Communication protocol type feedback	0 → Universal UART 1 → Modbus RTU master driver
103.H	2	I2C_Session_Counter	It changes after each session, when data in module updates. It updates every cca.100 ms.	0 .. 255 → 0 .. 255
104.L	3	Rx_Counter	Receive counter changes if new data in Rx_Buffer is available.	0 .. 255 → 0 .. 255
104.H 105.L	4,5	Rx_Data_Length	Number of data in Rx Buffer.	0 .. 65535 → 0 .. 65535
105.H	6	Rx_TLG_Status	Received telegram status	Bit0 → TLG timeout Bit1 → Data (byte) timeout Bit2 → Parity error Bit3 → Framing error Bit4 → Data overflow
106.L	7	Rx_Reception_Status	Receive reception status	0 → No reception 1 → Reception in progress 2 → Reception ended (timeout) 3 → Reception ended (data received) 11 → Error in reception
106.H 107.L	8,9	Rx_Timeout_Max	Maximal timeout between received data bytes in received telegram.	0 .. 65535 → 0 .. 65535 us

Table 25: TxBuffer, Data, Universal mode
4xxxx, Modbus RTU holding registers
SmartehIDE output parameters

Reg.	Par.	Name	Description	Raw → Engineering data
100.H	16	Tx_Buffer_Start	Start of transmit UART buffer	0 .. 65535 → 0 .. 65535
...	...			
227.H	270	Tx_Buffer_End	End of transmit UART buffer	0 .. 65535 → 0 .. 65535

Table 26: RxBuffer, Data, Universal mode
3xxxx, Modbus RTU input registers
SmartehIDE input parameters

Reg.	Par.	Name	Description	Raw → Engineering data
109.L	13	Rx_Buffer_Start	Start of receive UART buffer	0 .. 65535 → 0 .. 65535
...	...			
237.H	270	Rx_Buffer_End	End of receive UART buffer	0 .. 65535 → 0 .. 65535

4.4 Sound module command format

Table 27: Sound module command format

Length	Command	Parameters	Checksum
1 byte	1 byte	n bytes	1 byte

Length is determined by the sum of all bytes in command.
Length = Number of bytes: Length + Command + Parameters + Checksum

Checksum is determined by summing the values of all bytes in the command. For the checksum value, only the lower byte is used.
Checksum = Lower byte of the following sum: Length + Command + Parameters

Table 28: Sound module command example (Volume control)

Length	Command	Parameters	Checksum
0x04	0xAE	0x1E	0xD0

Length: 1 byte (Length) + 1 byte (Command) + 1 byte (Parameter) + 1 byte (Checksum) = 4
Checksum: 0x04 + 0xAE + 0x1E = 0xD0

Table 29: Sound module return code format

Command	Parameters	Description
xx	0x00	OK
	0x01	Error
	0x02	Does not have this file

4.5 Sound module commands list

Write commands

Table 30: Play flash file by index

Length	Command	File index high byte	File index low byte	Checksum
05	A0	00	01	xx

Table 31: Play flash file by name

Length	Command	File name ASCII code				Checksum
07	A1	54 (T)	30 (0)	30 (0)	31 (1)	xx

Table 32: Play USB file by index

Length	Command	File index high byte	File index low byte	Checksum
05	A6	00	01	xx

Table 33: Play USB file by name

Length	Command	File name ASCII codes				Checksum
07	A7	54 (T)	30 (0)	30 (0)	31 (1)	xx

Table 34: Play USB file by folder name and index

Length	Command	Folder name ASCII codes			File index high byte	File index low byte	Checksum
08	A8	54 (F)	30 (0)	31 (1)	00	01	xx

Table 35: Pause

Length	Command	Checksum
03	AA	AD

Table 36: Stop

Length	Command	Checksum
03	AB	AE

Table 37: Next

Length	Command	Checksum
03	AC	AF

Table 38: Previous

Length	Command	Checksum
03	AD	B0

Table 39: Volume

Length	Command	Volume	Checksum
04	AE	00 .. 1F	xx

Table 40: Playback mode

Length	Command	Volume	Checksum
04	AF	00 → Single (default) 01 → Singe track loop 02 → All tracks loop 03 → Random	xx

Table 41: Audio output

Length	Command	Volume	Checksum
04	B6	01 → Speaker output	BB

Table 42: Audio source

Length	Command	Volume	Checksum
04	D2	00 → Internal flash 02 → USB	xx

Read commands

Table 43: Volume status

Length	Command	Checksum
03	C1	C4
Answer		
C1	00 .. 1F → 0 .. 31 volume level	

Table 44: Working status

Length	Command	Checksum
03	C2	C5
Answer		
C2	01 → playing 02 → stopped 03 → paused	

Table 45: Number of files in flash

Length	Command	Checksum
03	C3	C6
Answer		
C3	00 00 .. FF FF → 0 .. 65535 number of files	

Table 46: Number of files in USB

Length	Command	Checksum
03	C7	CA
Answer		
C7	00 00 .. FF FF → 0 .. 65535 number of files	

Table 47: Number of files in USB in specific folder

Length	Command	Folder name[3] in ASCII	Checksum
03	C8	xx xx xx xx xx	xx
Answer			
C8	00 00 .. FF FF → 0 .. 65535 number of files		

Table 48: Current playing file index

Length	Command	Checksum
03	C9	CC
Answer		
C9	00 00 .. FF FF → 0 .. 65535 file index	

Table 49: Current connection status

Length	Command	Checksum
03	CA	CD
Answer		
CA	00 → No connection 01 → Flash 04 → USB 05 → USB and Flash 08 → PC 09 → PC and Flash	

Table 50: Current playing file name

Length	Command	Checksum
03	CB	CE
Answer		
CB	Song name by ASCII code (8 bytes)	

[3] The folder name length is fixed to 5 characters

5 INSTALLATION

5.1 Connection scheme

Figure 2: Connection scheme

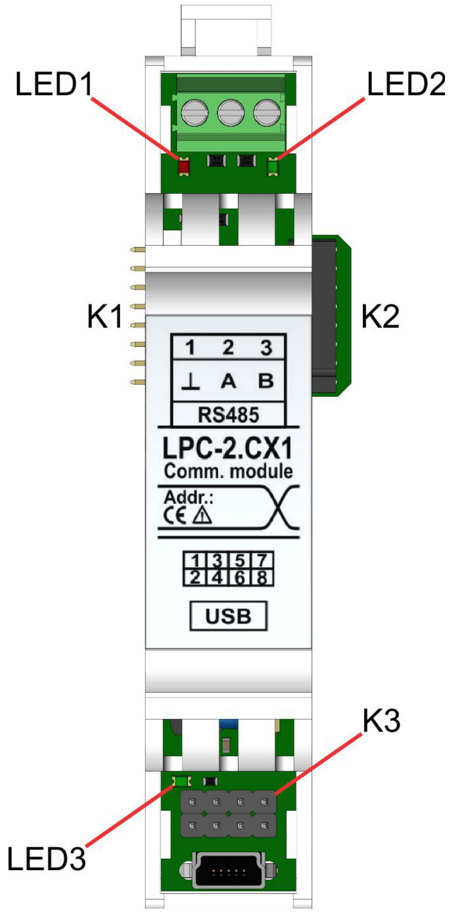


Table 51: RS485[1]

RS485.1	⊥	GND
RS485.2	RS-485 (A)	RS-485 standard voltage levels
RS485.3	RS-485 (B)	

Table 52: K1

K1.1	/	Not used
K1.2	/	Not used
K1.3	/	Not used
K1.4	/	Not used

Table 52: K1

K1.5	+ 24V DC	External power supply for audio amplifier
K1.6	⊥	GND
K1.7	Speaker Output	Speaker Output
K1.8	⊥	GND

Table 53: USB

USB	USB	Mini USB Type B receptacle connector

Table 54: K2

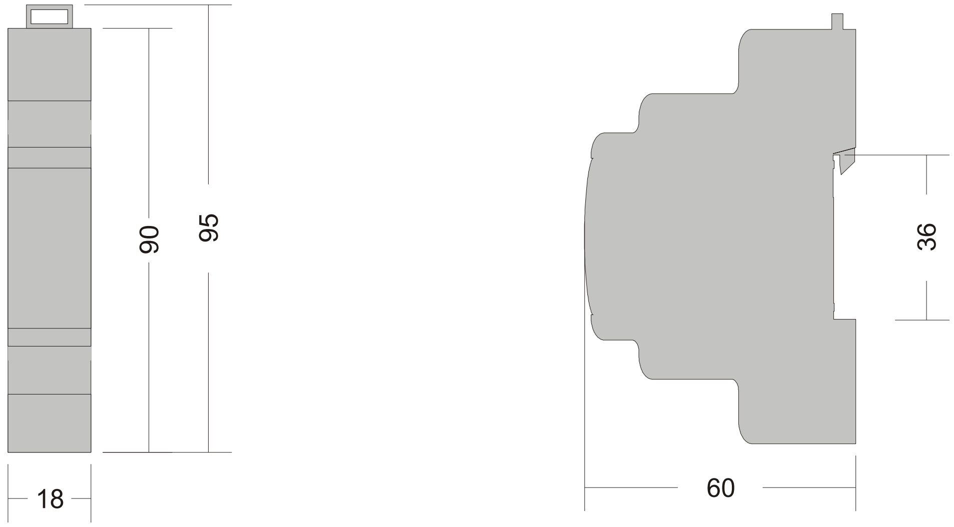
Internal BUS	Data & DC power supply	Connection to I/O module

Table 55: K3

Internal BUS	Data & DC power supply	Connection to I/O module

5.2 Mounting instructions

Figure 3: Housing dimensions



- Dimensions in millimeters.

⚠ All connections, module attachments and assembling must be done while module is not connected to the main power supply.

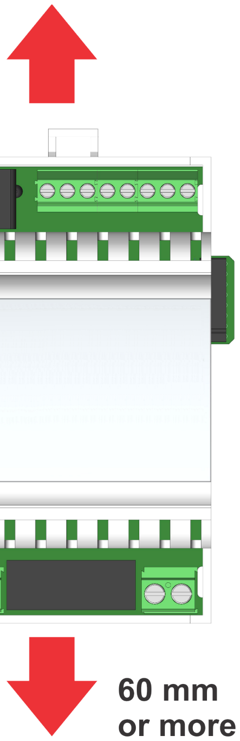
Mounting instructions:

1. Switch OFF main power supply.
2. Mount LPC-2.CX1 module to the provided place inside an electrical panel (DIN EN50022-35 rail mounting).
3. Mount other LPC-2 modules (if required). Mount each module to the DIN rail first, then attach modules together through K1 and K2 connectors.
4. Connect input and output wires according to the connection scheme in Figure 2.
5. Switch ON main power supply.

Dismount in reverse order. For mounting/dismounting modules to/from DIN rail a free space of at least one module must be left on the DIN rail.

NOTE: LPC-2 main module should be powered separately from other electrical appliance connected to LPC-2 system. Signal wires must be installed separately from power and high voltage wires in accordance with general industry electrical installation standard.

Figure 4: Minimum clearances



The clearances above must be considered before module mounting.

6 TECHNICAL SPECIFICATIONS

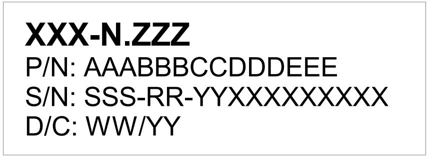
Table 56: Technical specifications

Power supply	From the main module via internal bus
Power supply for audio amplifier	From the main module via internal bus or external 24 V DC
Max. power consumption	10 W
Speaker	4 ohm, 10 W
Internal Flash	4 MB
USB	Mini B type, device mode or host mode, USB OTG
Connection type for RS485	Screw type connector for stranded wire 0.75 to 1.5 mm^2
Connection type for K1	2x4 pin header, R 2.54 mm
Dimensions (L x W x H)	90 x 18 x 60 mm
Weight	40 g
Ambient temperature	0 to 50 °C
Ambient humidity	max. 95 %, no condensation
Maximum altitude	2000 m
Mounting position	vertical
Transport and storage temperature	-20 to 60 °C
Pollution degree	2
Overvoltage category	II
Electrical equipment	Class II (double insulation)
Protection class	IP 30

7 MODULE LABELING

Figure 5: Label

Label (sample):



XXX-N.ZZZ
P/N: AAABBBCCDDDEEE
S/N: SSS-RR-YYXXXXXXXXX
D/C: WW/YY

Label description:

1. **XXX-N.ZZZ** - full product name.
    - **XXX-N** - Product family
    - **ZZZ** - product
2. **P/N: AAABBBCCDDDEEE -** part number.
    - **AAA** - general code for product family,
    - **BBB** - short product name,
    - **CCDDD** - sequence code,
        - **CC** - year of code opening,
        - **DDD** - derivation code,
    - **EEE** - version code (reserved for future HW and/or SW firmware upgrades).
3. **S/N: SSS-RR-YYXXXXXXXXX -** serial number.
    - **SSS** - short product name,
    - **RR** - user code (test procedure, e.g. Smarteh person xxx),
    - **YY** - year,
    - **XXXXXXXXX** - current stack number.
4. **D/C: WW/YY -** date code.
    - **WW** - week and
    - **YY** - year of production.

Optional

1. ***MAC***
2. ***Symbols***
3. ***WAMP***
4. ***Other***

8 CHANGES

The following table describes all the changes to the document.

Date	V.	Description
17.03.23	2	Update of tables 25, 44 and 47.
28.02.23	1	The initial version, issued as *LPC-2.CX1 module UserManual*.

9 NOTES