Serial and I²C communication with eol-series

Contents Page

1	Communication over RS232	2
1.1.	Baud Rate – Settings – Handshake	2
1.2.	User commands	2
1.2.1.	Usual commands	2
1.2.2.	Command “Channel”	2
1.2.3.	Command “Group”	3
1.2.4.	I²C Address	3
1.3.	User questions	3
1.4.	Table of user commands (serial)	4
1.5.	Table of user questions (serial)	5
1.6.	Practical commands	6
1.6.1.	Normal switches	6
1.6.2.	Shutters	6
1.6.3.	Switches with more then one input-channel	
2.	Communication over I²C	8
3.	I²C master modus	9
3.1.	Introduction	9
3.2.	List of all the possible commands	9
3.3.	Table of I²C master error codes	10

1. Communication over RS232

1.1. Baud Rate – Settings - Handshake

The serial communication over RS232 is based on ASCII Commands and Questions. The handshake is by software.

Baud rate	The baud rate is pre-selected and can not be changed. Possible is from 2400 till 57600 (ask for 76800 and 115200)
Settings	The communication needs 8 Bit, 1 stop bit, no parity.
Handshake	- each command or question must be quit with CR (0x13) and LF (0x10) - each answer ended with CR, LF

1.2. User commands

1.2.1. Usual commands

The user can send the commands *Channel* and *Group* for select a new channel and *I2C Address* for select the new Address for the communication over the I²C bus. The command *Channel* is also used to set different options.

1.2.2. Command "Channel"

The command *Channel* can have one, two, three or four digits for the new channel number (for example **ch2**). Any number greater then the maximum channel will select the max. possible channel (for example: **ch33** will select channel 12 of a switch eol 1x12), more then four digits will be ignored (the command **ch00011** has no effect).
The command **ch0** switch to the "blind channel" (if this option is available, for example the type eol 1x8 b), or disconnects all the channels of a shutter (for example the type eol 8x1-1) and is ignored for other eol types. In case of a switch with blind channel (the eol name ended with b), a *Channel* command with the maximum or a greater number as the maximum channel will select the blind channel, too.

The options of the *Channel* command are:

Start options:	**chx**	-	Sets the actual status in the moment of power off to the start channel. (After restart the device switched to the status at last power off.)
	chs	-	Sets the actual selected channel to the start-channel. (After restart the device switched to this channel.)
	chd	-	Sets the default channel to the start-channel. (After restart shutters use channel 0 and switches channel 1.)
Blind options:	**chn**	-	Sets a switch with blind channel to his normal equivalent. (The Blind channel is hidden and can not be used.)
	chb	-	Activates the blind channel (default). (Get the status of the switch with the question **type?** and interpret the ending: "eol 1x8 b" = blind channel available, "eol 1x8 bn" = device with unused blind channel.)
Count:	**chp**		switch to the next higher channel
	chm		switch to the next lower channel

1.2.3. Command “Group”
The command *Group* is used special for shutters (like the eol 8x1-1) and must have two (eol 8x1-1) four (eol 10x1-1, eol 16x1-1) or eight (eol 32x1-1) hexadecimal digits. Eight hexadecimal digits must be terminated with **l** (for long). Each bit switched one channel (“0” = off, “1” = on). The lowest bit (0) switches the channel 1. For example: **gr38** will switch on the channels 4, 5 and 6; **gr00000020l** switch on channel 6 of an eol 32x1-1.

The command *Group* is also used special for eol types with more then one switch (for example eol 3 1x4 or eol 6 1x2). In the case of 1x2 switches (eol 6 1x2 and eol 12 1x2) each switch is controlled by one bit of the grxx(xx(xxxxl)) command. In the case of 1x4 switches each switch is controlled by a pair of bits of the grxx(xx(xxxxl)) command. The lowest bit(s) of the grxx(xx(xxxxl)) command corresponds to the first switch. For example: The command **gr21** used for the type eol 6 1x2 will set switch 1 and 6 to channel 2 and switch 2 … 5 to channel 1.
For this eol types (with more then on switch) the binary (hexadecimal) code can also be transferred as a decimal number with the *Channel* command. For example: **ch11** (= **gr0B**) used for the type eol 3 1x4 will set switch 1 to channel 4, switch 2 to channel 3 and switch 3 to channel 1.
Normal eol types with firmware 4.xx and higher are switched to the channel corresponding to the lowest bit was is set into the grxx(xx(xxxxl)) command. For example: **gr18** will switch to channel 4. The eol types with firmware 3.xx do ignore the grxx(xx(xxxxl)) command.

1.2.4. I²C Address
The command *I²CAddress* can have one, two or three digits. The I²C address must be between 1 and 127, if not, the command will be ignored. (For example **i2c112** sets the I²C address to 112).

1.3. User questions

The user can ask for the actual channel number with *Channel Number?* (**ch?**) The answer of a normal switch is the actual number of the connected channel as a number of one, two, three or four digits without left-hand zeros or any letter.
If you ask a shutter with **ch?**, the answer is the lowest closed channel.

Equally the user can ask for the actual channel with *Group?* (**gr?)** The answer of **gr?** is a two or four byte hexadecimal ASCII number. This command is special made for shutters and devices with more then one switch.
Into the answer of **gr?** a shutter will set all bits corresponded to closed channels to “1”.
A device with more then one switch gives a binary coded answer corresponded to the chxx(xx(xxxx)) command. (See “Practical commands” for more details.)
(The **gr?** question at devices with firmware 3.xx will be ignored.)
A normal switch marked the bit with his actual channel number with ”1”. Since the answer has max. four hexadecimal digits, the max. returned channel can be 16.
For example: the answer “08” of a normal switch means the channel 4 is switched on, the same answer of a shutter means channel 4 is closed, all the other channels are cut off.

With *I²C-Address?* (**i2c?**) the actual address of the I²C bus is given back.
With *Delay?* (**delay?**) the user can ask for the delay time between two switches (this time is for protection of the piezos and mechanics).
Type? (**type?**) and *Firmware?* (**firmware?**) are asking for the used eol-type and the used software into the eol.

1.4. Table of user commands (serial)

Byte	6	5	4	3	2	1	handshake
I2C Address (3)	i	2	c	[Adr3]	[Adr2]	[Adr1]	CR, LF
I2C Address (2)		i	2	c	[Adr2]	[Adr1]	CR, LF
I2C Address (1)			i	2	c	[Adr1]	CR, LF
Channel (4)	c	h	[Ch4]	[Ch3]	[Ch2]	[Ch1]	CR, LF
Channel (3)		c	h	[Ch3]	[Ch2]	[Ch1]	CR, LF
Channel (2)			c	h	[Ch2]	[Ch1]	CR, LF
Channel (1)				c	h	[Ch1]	CR, LF
Group (8)* g r [Gr8][Gr7][Gr6]	[Gr5]	[Gr4]	[Gr3]	[Gr2]	[Gr1]	l	CR, LF
Group (4)*	g	r	[Gr4]	[Gr3]	[Gr2]	[Gr1]	CR, LF
Group (2)*			g	r	[Gr2]	[Gr1]	CR, LF
Blind channel enable				c	h	b	CR, LF
Blind channel disable				c	h	n	CR, LF
Set start channel				c	h	s	CR, LF
Set start channel at power off**				c	h	x	CR, LF
Set default start channel				c	h	d	CR, LF

Adr1,2,3 - 1, 2 or 3 Bytes of I²C address [1 … 127]

Ch1,2,3 - 1, 2 or 3 Bytes (0..9) of the number of channel to switch on [(0),1...999]
(Channel 0 can used only with shutters for switch off all channels or for select the blind channel of switches with this option.)

Gr1,2,3,4 - 2 or 4 Bytes (hexadecimal, 0…F) for channel selection
(each bit controls one channel, for example:
use “gr1c” to switch on channels 3, 4 and 5)

* Specially for shutters like eol 8x1-1 and eol types with more then one switch like the eol 3 1x4.
(All other types of eol with firmware 4.xx are switched to the lowest channel of the grxx(xx(xxxxl)) command, for example: “gr1c” switched to channel 3 of a switch; firmware 3.xx will ignore this command.)
The Group(8) command sends a 32 bit word (long int) and is terminated with “l”; for example: “gr789abcdel” sends the long integer 2023406814, the question “gr?” returns 8 byte “789ABCDE”.

** The last status of the channel(s) before the power is switched off will be saved for set at the next start. (Only for firmware 4.xx and higher, the power must switch off normally, not with a short cut, the capacity into the device gives enough time to safe the values.)

1.5. Table of user questions (serial)

Byte	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**handshake**
Firmware?	f	i	r	m	w	a	r	e	?	CR, LF
Answer*			v	e	r	3	.	0	1	CR, LF
Delay?				d	e	l	a	y	?	CR, LF
Answer*					1	4		m	s	CR, LF
Type ?					t	y	p	e	?	CR, LF
Answer**	e	o	l		1	x	8		m	CR, LF
I²C Address?						i	2	c	?	CR, LF
Answer*	[I 2C-addr		e	s	s	:		3	4	CR, LF
Channel Number?							c	h	?	CR, LF
Answer*									3	CR, LF
Group Number?							g	r	?	CR, LF
Answer***								9	C	CR, LF

* for example

** the ending m means the switch has a multi mode fiber
the ending b means the switch has a blind channel
the ending bn means the blind channel is hidden

*** for example the answer 9c means that channel 3, 4, 5 and 8 of a shutter are switched on

1.6. Practical commands

1.6.1. Normal switches

Do use the commands **chx … chxxxx** and **ch?** to control normal switches.

1.6.2. Shutters

Do use the commands **grxx**, **grxxxx, grxxxxxxxxl** and **gr?** to control shutters.
Each bit controls one channel.

eol 8x1-1: **grxx**
eol 10x1-1 and eol 16x1-1: **grxxxx**
eol 32x1-1: **grxxxxxxxxl**

1.6.3. Switches with more then one input-channel

Do use the commands **grxx**, **grxxxx, grxxxxxxxxl** and **gr?**.
Also you can use **chxxxx** and **ch?**

eol 6 1x2

Each bit controls one switch, if the Bit =0, then the switch is in position 1, if the Bit=1, then the switch is in position 2.

switch		Bit of the **grxx** command	switch		Bit of the **grxx** command
A	-	0	D	-	3
B	-	1	E	-	4
C	-	2	F	-	5

eol 3 1x4

Each switch is controlled by a pair of Bits:

channel	pair of Bits
1	00
2	01
3	10
4	11

switch		Bit of the **grxx** command
A	-	0, 1
B	-	2, 3
C	-	4, 5

eol 12 1x2

Each bit controls one switch, if the Bit =0, then the switch is in position 1, if the Bit=1, then the switch is in position 2.

switch		Bit of the **grxxxx** command	switch		Bit of the **grxxxx** command
A	-	0	G	-	6
B	-	1	H	-	7
C	-	2	I	-	8
D	-	3	J	-	9
E	-	4	K	-	10
F	-	5	L	-	11

eol 6 1x4

Each switch is controlled by a pair of Bits:

channel	pair of Bits
1	00
2	01
3	10
4	11

switch		Bit of the **grxxxx** command
A	-	0, 1
B	-	2, 3
C	-	4, 5
D	-	6, 7
E	-	8, 9
F	-	10, 11

2. Communication over I²C

A read command at the I²C-bus with the devices address gives back the actual channel of the device (only for devices with less than 256 channels).

A write command at the I²C-bus with the devices address will set the channel of the device to the number of the sent byte, if this is lower then 128.

Special functions (commands) of the sent byte and her serial command equivalent:

I²C	-	RS232	-	comment
255	-	chx	-	safe the channel at power off for next restart
254	-	chs	-	sets the actual channel as start-channel
253	-	chd	-	sets the default channel as start-channel
252	-	chb	-	enable blind channel
251	-	chn	-	disable blind channel
250	-		-	pre selection the next ch or gr command for general call
249	-		-	execute the pre selected command (by general call)
248	-		-	set the start channel
224	-		-	no operation
223	-	chxxx	-	set channel (+ 8 Bit data), select channels from 0 … 255
222	-	grxx	-	set group (+ 8 Bit data)
209	-	chxxxx	-	set channel (+ 2 x 8 Bit data), select channels from 0 … 65535
208	-	grxxxx	-	set group (+ 2 x 8 Bit data)
207	-	ch?	-	question: channel?, answer 1 Byte
206	-	gr?	-	question: group?, answer 1 Byte
205	-	gr?	-	question: group?, answer 2 Byte
204	-	type?	-	question type?, answer 1 Byte
202	-	ch?	-	question: channel?, answer 2 Byte (only for switches with more than 255 channels)
	-	ch?	-	question: channel?, answer 1 Byte (response without request), (only for switches with less than 128 channels)
0..127	-	chxxx		select channel, 1 Byte, channel 0…127

Note: The commands 222 and 223 needs one additional Byte to send and the commands 208 and 209 needs two additional Bytes to send to complete the command (first low-Byte, then high-Byte). After send the commands 204, 206 and 207 the master must read one Byte and after send the commands 202 and 205 the master must read two Bytes (first low-Byte, then high-Byte). The commands 205, 206, 208 and 222 are used for shutters and devices with more then one switch. (The 8 Bit commands 206 and 222 are used for devices with less then 9 possibilities.) The commands 207 and 223 are used for switches with less then 256 positions, for more positions are used the commands 202 and 209.

For example: To set a device with 12 switches (eol 12 1x2) you can use the command 208 (and 209, too). Each switch is controlled by one bit. To set the switches 1…3 to channel 2 and the switches 4…12 to channel 1 (this is binary = 0000.0000.0111) you can send following Bytes to the slave: slave-address, 208, 7, 0. To get back the channel information, you can send: slave-address, 208 and then receive two Bytes. The Bytes should have the values: 7 and 0.
Example for use of general call to set all switches/shutters connected at the I²C-bus exactly at same time: Send 250 with general call, then preset the new channels of all connected devices, then send 249 with general call to set all pre selected channels at same time.

3. I²C master modus

3.1. Introduction

Each switch with firmware 4.xx can be used as I²C master. So you can control up to 128 switches with only one serial interface. To use this function please give each switch another I²C address (with the command **i2c…**into the serial mode). Then connect all devices over the I²C bus and connect two resistors from the bus to +5V. Connect one switch to the serial bus, this is the I²C master. If you send a command like **i[254]i[ADR]t** over the serial bus to the master, he will send the command 254 to the slave with the address ADR. If the transmission was correct, the master will send the return status 254 over the serial bus back to the PC in form of **i[254]c**. In case of error, the error code in form of **i[error code]c**.
With the command **[channel]i[223]i[ADR]t** you can set the channel of a slave. With the question **i[207]i[ADR]t** you ask for the actual channel and with **i[1]i[ADR]r** you can get the 1 Byte answer of the question. Note: all the values into [] are binary and only one Byte.
With all the commands it is possible to use the address of the master itself, too.

3.2. List of all the possible commands

RS232 command	return status	comment
i [255] i [ADR] t	255	= chx
i [254] i [ADR] t	254	= chs
i [253] i [ADR] t	253	= chd
i [252] i [ADR] t	252	= chb
i [251] i [ADR] t	251	= chn
i [250] i [ADR] t	250	pre selection the next ch or gr command for general call
i [249] i [ADR] t	249	execute the pre selected general call
i [248] i [ADR] t	248	select start channel
i [224] i [ADR] t	224	NOOP
i [ch_l] i [ADR] t	[ch_l]	select channel, one Byte (0..127) (only for channels les then 128)
[channel] i [223] i [ADR] t	223	select channel, one Byte (0..255)*

[gr1,2] i [222]i [ADR] t	222	select group, hexadecimal, one Byte (0-7)*
[channel1] i [channel2] i [ADR] h	209	select channel, two Byte (0...65535)*
[gr1,2] i [gr3,4] i [ADR] g	208	select group, hexadecimal, two Byte (0-15)*
i [207] i [ADR] t	207	question: channel?, answer 1 Byte
i [206] i [ADR] t	206	question: group?, answer 1 Byte
i [205] i [ADR] t	205	question: group?, answer 2 Byte
i [204] i [ADR] t	204	question type?, answer 1 Byte (type-number)
i [202] i [ADR] t	202	question: channel?, answer 2 Byte
[b1] i [b2] i [b3] i [ADR] 3	[b1]	sends three Byte to address ADR (used for other devices)*
i [1] i [ADR] r		receive of 1 Byte as answer of a previous question answer: i [data] i [status] c (status=master error code) (without question you will receive the channel number)
i [2] i [ADR] r		receive of 2 Byte as answer of a previous question answer: [data2] i [data1] i [status] c

* in case of first byte = [10] you must send an additional character in front: i[10]i[223]i[ADR]t

3.3. Table of I²C master error codes

I²C - master error codes - general

192 - no error
191 - no data requested
190 - to many data requested
189 - bad I²C address
188 - unknown question

I²C - master error codes – receiver mode

147 - MT_DATA_NACK instead of MT_DATA_ACK received
146 - neither MT_DATA_ACK nor MT_DATA_NACK received
145 - time out on transmit data
144 - MT_SLA_NACK instead of MT_SLA_ACK received
143 - neither MT_SLA_ACK nor MT_SLA_NACK received
142 - time out on transmit address
141 - no START condition
140 - time out in START condition

I²C - master error codes – transmitter mode

137 - MT_DATA_NACK instead of MT_DATA_ACK received
136 - neither MT_DATA_ACK nor MT_DATA_NACK received
135 - neither MT_DATA_ACK nor MT_DATA_NACK received, last Byte
134 - time out on transmit data
133 - MT_SLA_NACK instead of MT_SLA_ACK received
132 - neither MT_SLA_ACK nor MT_SLA_NACK received
131 - time out on transmit address
130 - no START condition
129 - time out in START condition
128 - other error