thermokon
Sensortechnik GmbH

Interface Description

EnOcean Gateway STC-RS485-EVC



enocean®

SmartACK

Content

1 Errata 2
2 General Notice 3
3 Data format RS485 interface 4
3.1 Baud rate 4
3.2 Parity 4
4 Definitions 5
4.1 Common telegram structure 5
4.1.1 Command (15 Bytes) 5
4.1.2 Response (14 Bytes) 5
4.2 Synchronization 5
4.3 Command 5
4.4 Filter mode and channels 5
4.5 EnOcean ID and Equipment Profiles (EEP) 5
4.6 Checksum 6
4.6.1 Example of checksum calculation 6
4.7 STC-EVC device address 6
5 Command summary 7
6 Commands for configuration 8
6.1 Write configuration (FFFF) 8
6.2 Teach in sensor via ID (FFF3) 10
6.3 Delete sensor/s from filter channel (FFFC) 14
6.4 Read sensor of filter channel (FFFA) 15
6.5 Read Chip- & Base-ID of the STC65 (FFF9) 17
6.6 Read Configuration of the STC65 (FFF8) 18
6.7 Read Firmware Version of the STC65 (FFF6) 19
6.8 Read filter state (FFF5) 20
6.9 Read all filter table channel (selective) (FFF4) 21
7 Smart Acknowledge Commands 22
7.1 Teach-in SMACK Device via learn button (FFFB) 22
7.2 Fill Mailbox for SMACK device / send data to SMACK device (6CD2) 25
8 Send EnOcean telegrams via EVC 27
8.1 Send F6, D5 and A5 EnOcean telegrams (6B) 27
8.2 Send VLD and MSC EnOcean telegrams (6B) 28
8.3 Response to command: send EnOcean telegram (6B) 29
9 Receiving EnOcean telegrams via EVC 30
9.1 Explanations 30
9.1.1 Bus Access 30
9.1.2 Repeater Function 30
9.1.3 VLD and MSC telegrams 30
9.1.4 Status 30
9.1.5 Telegram counter 30
9.1.6 Repeat counter 30
9.2 Telegram structure of RPS, 1BS and 4BS telegrams 31
9.3 Telegram structure of VLD and MSC telegrams 32
10 Configuration Software 33

1 Errata

Document	Hardware	Firmware	Date	Who	Description
1.0.0	Rev. A	1.0.0	14.04.2009	FF	First release
2.0.0	Rev. B	2.0.0	14.10.2013	FF	Appendices to new hardware (Rev B) and firmware 2.0.0 **Note:** Since hardware Rev B and firmware 2.0.0 only the RTU mode will be supported. The significantly slower and less efficient ASCII mode is no longer supported.
3.0.0	Rev. B	3.0.0	28.09.2015	JC	- Gateway supports VLD/MSC telegrams - optional data configurable (contains: RSSI, Destination address) - Compatibility mode for replacing old gateways - Up to 64 filter channels - Up to 15 Smart Acknowledge Devices (SMACK) - More commands - Boot loader implemented - Command “Read out all channel” added
3.0.1	Rev. B	3.0.0	01.10.2015	JC	- Command Code “Read Firmware” FF F7 corrected - Response “Device Teached-In” unified
3.0.2	Rev. B	3.0.0	20.10.2015	JC	- Release
3.0.3	Rev. B	3.0.0	04.11.2015	JC	- EnOcean Logos implemented
3.0.4	Rev. B	3.0.1	23.11.2015	JC	- Swap Byte 3 and 0 of RPS and 1BS devices in compatibility mode
3.0.5	Rev. B	3.0.1	01.04.2016	DF	- 8.1 & 8.2: Checksum Calculation corrected - Added ORG Bytes for MSC(D1) and VLD(D2)
3.0.6	Rev. B	3.0.2	01.04.2016	JC	FW Update: - Compatibility Mode: - Response “Send Telegram” corrected - Response of “Teach In” corrected - Exchanged DB3 and DB0 for Sending RPS, 1BS Telegrams via EnOcean - Mailbox for SMACK Devices with 4BS can be cleared (#DB = 0)

-

2 General Notice

The radio receiver module STC-RS485-EVC operates as a gateway between sensors or radio keys based on the EnOcean radio technology as well as controllers or control systems with RS485 interface. Flawlessly received telegrams are output unchanged to the RS485. Furthermore, EnOcean based telegrams can be sent.

Various STC-RS485-EVC can be operated by one RS485 bus segment. In order to avoid data loss by telegram collision, the RS485 wire is monitored by every STC. The STC is only allowed to send within a bus silent interval, whereas the telegram is sent event-controlled for two times to have an additional safety (parameter able).

The SRC-STC-RS485-EVC can be operated in two modes: gateway and with a filter. In the gateway mode, all telegrams received are forwarded to the RS485 bus. In the filter mode, up to 32 sensors can be seamlessly connected. Only those sensors taught-in are sent to RS485 bus.

Since Firmware Version 3.0.0 EnOcean Smart Acknowledge (SMACK) devices are supported (for example: Thermokon SR06 LCD). By means of SMACK, a bidirectional communication between sensor and central control unit is possible. Thanks to SMACK, set points and status can be overwritten externally and indicated via the integrated display.

Additionally, optional data of the sensors are transmitted via EVC. Depending on the configuration of the gateway, important information like the signal strength (RSSI) or the destination address can be obtained.

When changing-over dip switch 2.3, the compatibility mode is activated. Thanks to this mode, old STC devices can be easily replaced. Thus, the bus communication is adapted to the previous firmware version 2.0.0 so that no VLD/MSC telegrams, SMACK and various commands are not available. The responses to the commands correspond to the old software status, as well.

3 Data format RS485 interface

The data type depends on the hardware dip switch settings. For more details refer to datasheet.

3.1 Baud rate

The baud rate chosen mainly determines the number of sensors that can be evaluated by one RS485 bus segment. The higher the baud rate, the lower the bus load per radio telegram. In order to minimize the risk of a telegram collision and data loss, the highest possible baud rate should be basically selected.

Baud Rate	Max. number of sensors per bus segment
9,6 kbps	50
19,2 kbps	100
38,4 kbps	170
115,2 kbps	250

When arranging for the installation of the radio receivers and sensors within a building, it should be considered, that the sensors are only installed within the receiving range of one receiver, if possible. If a radio telegram is received by several receivers at the same time, this telegram is output to the RS485. This is inevitably leading to an increased bus load and an increased risk of telegram collisions.

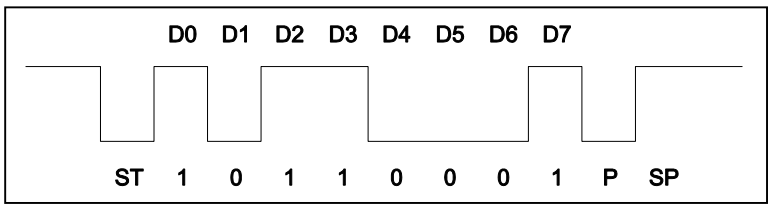
3.2 Parity

For transmission of data bytes always 11 bits are needed (Start bit, data bits, parity bit, stop bit).

Parity	Stop Bits
even	1
odd	1
no parity	2

Example:

Parity (P): even
Start bit (ST): 1
Data bits: 8
Stop bits: 1



4 Definitions

4.1 Common telegram structure

4.1.1 Command (15 Bytes)[1]

Description	Preamble	Command	Command specific data	Checksum	Device address
Value (HEX)	A5 5A	AA BB	00…FF	AA+BB+00…+FF	0…3F
#Bytes	2	2	9	1	1

4.1.2 Response (14 Bytes)

Description	Preamble	Device address	Command	Response specific data	Checksum
Value (HEX)	A5 5A	0…3F	AA BB	00…FF	A5+5A+…+00+…+FF
#Bytes	2	1	2	8	1

4.2 Synchronization

To synchronize the start of a telegram between sender and receiver a Preamble consisting of two sync bytes will be sent at the beginning of every telegram. The values are A5 and 5A. If configured, optional data can be transmitted additionally. This data will be added to the existing telegram shown above. To synchronize the beginning of optional data the Preamble B5 5B is sent.

4.3 Command

After synchronization two command bytes will be sent. With different commands the STC can be configured or operations will be done.

4.4 Filter mode and channels

STC-RS485 can be operated with a filter. Up to 64 (0x40) sensors or other EnOcean Devices can be teached in (channel 00…63 = 0x00…0x3F). The teach-in process can be done automatically with the learn button or manually by transmitting the sensor ID to the Gateway. The teach in mode depends on the command used.

4.5 EnOcean ID and Equipment Profiles (EEP)

Every EnOcean device has a unique 32-Bit CHIP-ID. For identification of sensor type, range etc. every device sends an equipment profile during the teaching-in. To find detailed information regarding EEPs, see www.enocean.com.

Beside the unique CHIP-ID every EnOcean module has a variable BASE-ID. This is used for sending telegrams with different IDs/channels. BASE-ID Byte3…1 is invariable but Byte 0 is variable in the range 0…127. Thus, 128 channels are available for sending telegrams. For Smart Acknowledge always the unique invariable CHIP-ID is used!

[1] To send VLD/MSC telegrams as well as for filling mailboxes the telegram length will be extended. Please refer to the corresponding chapter.

4.6 Checksum

The next to last byte of every telegram is the checksum. It can be calculated by adding all bytes in front of the checksum, except the preamble (A55A). Only the lowermost two characters, the least significant byte, are relevant. The separate checksum for optional data is calculated with the optional data preamble (B5 5B) (refer to examples b). Also for responses to commands, the preamble (A55A) is included to the checksum calculation (refer to example c).

4.6.1 Example of checksum calculation

4.6.1.1 Example telegram a (command):

A5 5A FF FE 02 07 01 90 56 DF 00 00 00 CC 3E

Calculation of cross sum:

Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	=	Cross sum
FF +	FE +	02 +	07 +	01 +	90 +	56 +	DF +	00 +	00 +	00	=	0x3CC

Only least significant byte/ lowest two characters (cross sum and logical AND with 0xFF):

Cross sum & 0xFF = 0x3CC & 0xFF = 0xCC = Checksum

4.6.1.2 Example telegram b (Command with optional data):

A5 5A 6B A5 00 00 00 00 00 00 00 00 00 10 3F B5 5B AB CD ED CB 00 40

Cross sum = 0x110 — *Checksum = 0x110 & 0xFF = 0x10*

Cross sum = 0x440 — *Checksum = 0x440 & 0xFF = 0x40*

Note: The separate checksum for optional data is calculated with the optional data preamble (B5 5B)!

4.6.1.3 Example telegram c (response to command):

A5 5A 3F FF F9 FF D3 D6 80 01 86 A7 AD 39

Cross sum = 0x839

Checksum = 0x839 & 0xFF = 0x39

Note: Checksum of responses is calculated with preamble (A5 5A)!

4.7 STC-EVC device address

Every STC-EVC can be configured with an address (dip switch 1: 1…6) in the range of 0…63 (0x00…0x3F). This address will be transmitted at the end of each command. Depending on this address, the bus access is operated. Regarding to this every STC device address should be unique on one bus line. In the following examples the device address is xx and should be replaced for test purposes.

5 Command summary

	Command			
#	**A**	**B**	**Description**	**Chapter**
1	FF	FF	Write configuration (Filter/Gateway, 1x/3x, OD On/Off)	6.1
2	FF	F3	Teach in via ID (No SMACK devices)	6.2
3	FF	FD	Teach in via learn button (No SMACK devices)	6.3
4	FF	FC	Delete sensor from filter channel / Delete all	6.4
5	FF	FB	Teach in SMACK device via learn button	7.1
6	FF	FA	Read sensor from filter channel	6.5
7	FF	F9	Read BASE- & CHIP-ID	6.6
8	FF	F8	Read configuration (Filter/Gateway, Repeat 1x/3x, etc.)	6.7
9	FF	F7	Read firmware version of the STC	6.8
10	FF	F5	Read filter state	6.9
11	FF	F4	Read all filter channel (selective)	6.10
10	6B	ORG[2]	Send EnOcean telegram	8
11	6C	D2	Fill SMACK mailbox / Write data for SMACK device	7.2

[2] ORG-Byte of the telegram that should be transmitted

6 Commands for configuration

The STC65-RS485 can be configured individually. In default mode, it works in gateway mode and transmits all incoming EnOcean telegrams three times. Configuration can be made with the following commands:

6.1 Write configuration (FFFF)

The basic configuration includes the filter/gateway mode, repeat mode and the optional data.

Example f: activate filter mode, repeat telegram 3x, activate optional data (OD):
A55A FFFF 00 FF FF 000000000000 FC 3F

Example	Configuration	Preamble A	Preamble B	Command A	Command B	Filter/Gateway	Repeat 1x/3x	OD On/Off	-/-						Checksum	Device address (xx)
a	Filter, 1x, OD Off	A5	5A	FF	FF	00	00	00	00	00	00	00	00	00	FE	3F
b	Filter, 1x, OD On	A5	5A	FF	FF	00	00	FF	00	00	00	00	00	00	FD	3F
c	Filter, 3x, OD Off	A5	5A	FF	FF	00	FF	00	00	00	00	00	00	00	FD	3F
d	Filter, 3x, OD On	A5	5A	FF	FF	00	FF	FF	00	00	00	00	00	00	FC	3F
e	Gateway, 1x, OD Off	A5	5A	FF	FF	FF	00	00	00	00	00	00	00	00	FD	3F
f	Gateway, 1x, OD On	A5	5A	FF	FF	FF	00	FF	00	00	00	00	00	00	FC	3F
g	Gateway, 3x, OD Off	A5	5A	FF	FF	FF	FF	00	00	00	00	00	00	00	FC	3F
h	Gateway, 3x, OD On	A5	5A	FF	FF	FF	FF	FF	00	00	00	00	00	00	FB	3F

Example f:

Byte	Bit7 Bit0	Description	Example f (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xFF	FF
4	FILTER_MODE	0x00 = activate filter 0xFF =deactivate filter	00
5	REPEAT_MODE	0x00 = send EVC telegram 1x 0xFF = send EVC telegram 3x	FF
6	OPTIONAL_DATA	0x00 = activate optional data 0xFF = deactivate optional data	FF
7…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	FC
14	ADDRESS	address STC (Dip Switch 1-6)	3F

Response

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	COMMAND_A	0xFF	FF
4	COMMAND_B	0xFF	FF
5	FILTER_MODE	0x00 = filter activated 0xFF = filter deactivated	00
6	REPEAT_MODE	0x00 = send EVC telegram 1x 0xFF = send EVC telegram 3x	FF
7	OPTIONAL_DATA	0x00 = optional data deactivated 0xFF = optional data activated	FF
8	COMPATIBILTY_MODE	0x00 = Compatibility mode deactivated 0xFF = Compatibility mode activated	00
9…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 0 - 12	3A

6.2 Teach in sensor via ID (FFF3)

Example (HEX):

Preamble	Command	Channel	ORG	(FUNC)	(TYPE)	-/-	Sensor ID	Checksum	Device address (XX)
A5 5A	FF F3	00	A5	00	00	00	00 06 C3 21	EE	01

Description: Sensor with ID 0006C321 and ORG A5 should be teached-in to the STC with device address 1 on channel 00:
A55AFFF300A50000000006C32181 01

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xF3	F3
4	FILTER_TABLE_CHANNEL	Filter channel 0…63 = 0x00…0x3F	00
5	ORG	ORG Byte	A5
6	(FUNC)	Device function (optional)	00
7	(TYPE)	Device type (optional)	00
8	-/-	0x00	00
9	ID_BYTE3	32-Bit EnOcean ID	00
10	ID_BYTE2		06
11	ID BYTE1		C3
12	ID_BYTE0		21
13	CHECKSUM	Checksum of bytes 2 - 12	81
14	ADDRESS	address STC (Dip switch 1-6)	01

Response – Device Teached-In (0F 01)

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip switch 1-6)	01
3	RESPONSE_CODE_A	Response A = 0x0F	0F
4	RESPONSE_CODE_B	Response B = 0x01	01
5	FILTER_TABLE_CHANNEL	Filter channel 0…63 = 0x00…0x3F	00
6	ORG	ORG Byte	A5
7	(FUNC)	Device function (optional)	00
8	(TYPE)	Device type (optional)	00
9	ID_BYTE3	32-Bit EnOcean ID	00
10	ID_BYTE2		06
11	ID_BYTE1		C3
12	ID_BYTE0		21
13	CHECKSUM	Checksum of bytes 0 - 12	9F

Note: If there already is a teached-in device with the transmitted ID, the FILTER_TABLE CHANNEL will be 0xFF in response. The Sensor will not be teached-in for a second time. The following bytes will be 0.

6.3 Teach in sensor via learn button (FFFD)

Example (HEX)

Preamble	Command	channel	-/-	Checksum	Device address (XX)
A5 5A	FF FD	2B	00 00 00 00 00 00 00 00	27	1C

Description: A sensor should be taught-in via the learn button on STC device with address 28 on channel 43:

A55AFFFD2B0000000000000000271C

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xFD	FD
4	FILTER_TABLE_CHANNEL	Filter channel 0…63 = 0x00…0x3F	2B
5…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	27
14	ADDRESS	address STC (Dip Switch 1-6)	1C

Response to received Command

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	1C
3	COMMAND_A	Command A = 0xFF	FF
4	COMMAND_B	Command B = 0xFD	FD
5	FILTER_TABLE_CHANNEL	Filter channel 0…63 = 0x00…0x3F	2B
6	MAX_FILTER_TABLE_CHANNEL	Max. filter channel = 64 = 0x40	40
7…11	-/-	0x00	00
12	COMMAND_STATE	0x00 = OK 0xFF = Error default 0xFE=FILTER_TABLE_CHANNEL overload	00
13	CHECKSUM	Checksum of bytes 0 - 12	82

Note: FILTER_TABLE_CHANNEL must be smaller than MAX_FILTER_TABLE_CHANNEL. Otherwise the error 0xFE = FILTER_TABLE_CHANNEL overload will occur.

2. Response – Device Teached-In (0F 01)

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	1C
3	RESPONSE_CODE_A	Response A = 0x0F	0F
4	RESPONSE_CODE_B	Response B = 0x01	01
5	FILTER_TABLE_CHANNEL	Filter channel 0…63 = 0x00…0x3F	2B
6	ORG	telegram type	A5
7	FUNC	Device function	02
8	TYPE	Device type	05
9	ID_BYTE3	32-Bit EnOcean ID	00
10	ID_BYTE2		06
11	ID_BYTE1		C3
12	ID_BYTE0		21
13	CHECKSUM	Checksum of bytes 0 - 12	EC

Note: The profile containing ORG, FUNC and TYPE will be sent as a teach-in telegram by most of the EnOcean devices. The content of data bytes is depending on this profile. The so-called EEPs[3] specify how the data must be interpreted.

[3] Please refer to www.enocean.com

6.4 Delete sensor/s from filter channel (FFFC)

Example (HEX):

Preamble	Command	channel	-/-	Checksum	Device address (XX)
A5 5A	FF FC	0A	00 00 00 00 00 00 00 00	05	3F

Description: Sensor in channel 10 should be deleted in STC with address 63:
A55AFFFC0A0000000000000000053F

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xFC	FC
4	FILTER_TABLE_CHANNEL	0…63 = 0x00…0x3F; 0xFE = Delete ALL	0A
5…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	05
14	ADDRESS	address STC (Dip Switch 1-6)	3F

Response

Description: STC with address 63 deleted sensor with ORG-Byte A5 and ID 0185B8C4 in channel 10.

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	COMMAND_A	Command A =0xFF	FF
4	COMMAND_B	Command B = 0xFC	FC
5	FILTER_TABLE_CHANNEL	Filter channel = 0…63 = 0x00…0x3F	0A
6	ORG	ORG-Byte of the deleted device	A5
7	ID_BYTE3	32 Bit EnOcean ID of the deleted device	01
8	ID_BYTE2		85
9	ID_BYTE1		B8
10	ID_BYTE0		C4
11…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 0 - 12	EA

6.5 Read sensor of filter channel (FFFA)

Example (HEX):

Preamble	Command	channel	-/-	Checksum	Device address (XX)
A5 5A	FF FA	0E	00 00 00 00 00 00 00 00	07	3F

Description: Get information about taught-in sensors in channel 14 of Device 63:
A55AFFFA0E0000000000000000073F

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xFA	FA
4	FILTER_TABLE_CHANNEL	Filter channel 0…63 = 0…3F	0E
5…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	07
14	ADDRESS	address STC (Dip Switch 1-6)	3F

Response

Description: In channel 0E of STC with address 63 a SMACK device with Profile D2-00-01 (ORG-FUNC-TYPE) is taught-in.

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	CODE_A	Command A = 0xFF	FF
4	CODE_B	Command B = 0xFA	FA
5	FILTER_TABLE_CHANNEL	Filter channel 0…63 = 0…3F	0E
6	ORG	telegram type	D2
7	FUNC	Device function	00
8	TYPE	Device type	01
9	ID_BYTE3	32 Bit EnOcean ID	01
10	ID_BYTE2		8A
11	ID_BYTE1		FB
12	ID_BYTE0		91
13	CHECKSUM	Checksum of bytes 0 - 12	F5

Note: If no EnOcean device is taught-in to the requested channel, the response contains the default ID = 0xFF FF FF FF.

The profile (ORG, FUNC and TYPE) is transmitted by most of the EnOcean devices during the taught-in process. The content of data bytes is depending on the profile. The so-called EEPs[4] specify the data interpretation.

If the value FILTER_TABLE_CHANNEL exceeds the maximal number of channels (64), the response will return 0xFF in FILTER_TABLE_CHANNEL and 0x00 in all other bytes.

[4] Please refer to www.enocean.com

6.6 Read Chip- & Base-ID of the STC65 (FFF9)

Example (HEX):

Preamble	Command	-/-	Checksum	Device address (XX)
A5 5A	FF F9	00 00 00 00 00 00 00 00	F8	3F

Description: Get EnOcean BASE- and CHIP-ID of the Gateway with address 62:
A55AFFF9000000000000000000F83F

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	0xFF	FF
3	COMMAND_B	0xF9	F9
4…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 0 - 12	F8
14	ADDRESS	address STC (Dip Switch 1-6)	3F

Response

Description: The CHIP_ID (here: 01 86 A7 AD) is used for SMACK telegrams. The BASE_ID + Offset (0…127) (here: FF D3 D6 80) can be used for sending common telegrams (command 0x6B).

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	COMMAND_A	0xFF	FF
4	COMMAND_B	0xF9	F9
5	BASE_ID_BYTE3	32-Bit EnOcean BASE ID	FF
6	BASE_ID_BYTE2		D3
7	BASE_ID_BYTE1		D6
8	BASE_ID_BYTE0		80
9	CHIP_ID_BYTE3	32-Bit EnOcean CHIP ID	01
10	CHIP_ID_BYTE2		86
11	CHIP_ID_BYTE1		A7
12	CHIP_ID_BYTE0		AD
13	CHECKSUM	Checksum of bytes 0 - 12	39

6.7 Read Configuration of the STC65 (FFF8)

Example (HEX)

Preamble	Command	-/-	Checksum	Device address (XX)
A5 5A	FF F8	00 00 00 00 00 00 00 00 00	F7	3F

Description: Read Configuration of the STC with address 63:
A55AFFF8000000000000000000F73F

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xF8	F8
4…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	F7
14	ADDRESS	address STC (Dip Switch 1-6)	3F

Response to received command

Description: Device with address 63 is configured as a gateway, telegrams will not be repeated and optional data is activated. Compatibility mode is deactivated.

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	COMMAND_A	Command A = 0xFF	FF
4	COMMAND_B	Command B = 0xF8	F8
5	GATEWAY_FILTER	Gateway = 0xFF Filter = 0x00	FF
6	REPEAT_MODE	telegram Repeat 1x = 0x00 3x = 0xFF	00
7	OPTIONAL_DATA	optional data activated = 0xFF optional data deactivated = 0x00	FF
8	COMPATIBILTY_MODE	Compatibility mode activated = 0xFF Compatibility mode deactivated = 0x00	00
7…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 0 - 12	33

6.8 Read Firmware Version of the STC65 (FFF7)

Example (HEX):

Preamble	Command	-/-	Checksum	Device address (XX)
A5 5A	FF F7	00 00 00 00 00 00 00 00	F6	3E

Description: Read Firmware Version of the STC-Gateway with address 62:
A55AFFF7000000000000000000F63E

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xF7	F7
4…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	F6
14	ADDRESS	address STC (Dip Switch 1-6)	3E

Response

Description: The Firmware Version of the STC-Gateway with address 62 is 3.0.0.

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3E
3	COMMAND_A	Command A = 0xFF	FF
4	COMMAND_B	Command B = 0xF7	F7
5	FIRMWARE_VERSION_MAIN	STC-EVC Firmware Version	03
6	FIRMWARE_VERSION_SUB		00
7	FIRMWARE_VERSION_REVISION		00
8…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 0 - 12	36

6.9 Read filter state (FFF5)

Example (HEX):

Preamble	Command	-/-	Checksum	Device address (XX)
A5 5A	FF F5	00 00 00 00 00 00 00 00 00	F4	3F

Beschreibung: Get next free filter channel, maximal number of filter channel, number of teached-in SMACK devices and maximal number of SMACK devices:
A55AFFF5000000000000000000F43F

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xF5	F5
4…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	F4
14	ADDRESS	Address STC (Dip Switch 1-6)	3F

Response

Description: The next free filter table channel of the STC-Gateway with Address 63 is 4. A maximum number of 0x0F = 16 SMACK devices of overall maximum 0x40 = 64 devices can be teached in. There are 6 teached in SMACK devices.

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	Adress STC (Dip Switch 1-6)	3F
3	COMMAND_A	Command A = 0xFF	FF
4	COMMAND_B	Command B = 0xF5	F5
5	NEXT_FREE_FILTER_CHANNEL	Next free filter table channel	04
6	MAX_FILTER_CHANNEL	Maximal number of filter table channel	40
7	-/-	0x00	00
8	SMACK_LEARNED_CLIENTS	Teached in SMACK devices	06
9	MAX_SMARTACK_DEVICES	Maximal number of teached-in SMACK devices	0F
10…12	-/-	0x00	00
13	CHECKSUM	Checksum of Bytes 0 - 12	8B

6.10 Read all filter table channel (selective) (FFF4)

Description: This command will get all filter table channels. Depending on KIND_OF_CHANNEL(Byte 4) you will get all, all free, every teached-in or every SMACK teached-in filter table channel. Every response telegram will contain ORG, FUNC, TYPE and CHIP-ID of the associated channel.

Note: Please consider that this command may cause high bus traffic.

Example (HEX):

Preamble	Command	Parameter	-/-	Checksum	Device address (XX)
A5 5A	FF F4	02	00 00 00 00 00 00 00 00	F5	05

Description: Get all teached in devices on STC with address 05:
A55AFFF402000000000000000F505

Byte	Bit7 Bit0	Desription	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xF5	F4
4	KIND_OF_CHANNEL	Respond selected channels: 0x00 = All 0x01 = All free 0x02 = Teached-in devices 0x03 = = Teached-in SMACK devices	02
5…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	F5
14	ADDRESS	Address STC (Dip Switch 1-6)	05

Response

Description: The response exists of 4 telegrams, by means 4 devices are learned on the STC with address 05. Data (ORG, FUNC, TYPE, ID) of the associated sensors, switch, etc. are shown below.

Antworttelegramme

Byte	*0…1*	*2*	*3…4*	*5*	*6*	*7*	*8*	*9…12*	*13*
#	Präambel	Geräteadresse (XX)	Befehl	Kanal	ORG	FUNC	TYPE	32 Bit CHIP_ID	Prüfsumme
1	A5 5A	05	FF F4	00	A5	10	10	01 85 B8 C4	BE
2	A5 5A	05	FF F4	03	A5	10	06	00 05 CB 9F	24
3	A5 5A	05	FF F4	0A	D2	00	01	01 8B 0C 32	9E
4	A5 5A	05	FF F4	38	F6	02	01	00 2B 2E DE	5F

7 Smart Acknowledge Commands

Since Firmware Version 3.0.0 the STC65-RS485 EVC Gateway supports EnOceans Smart Acknowledge procedure for bidirectional communication with self-powered sensors.

Please consider the following requirements for the use of SMACK:

- While teaching-in a SMACK device, only one gateway should be in the teach-in mode!
- Only 1 filter channel per SMACK device. Do not teach-in in more than one filter channel!
- Do no teach in a smack device in more than one gateway!
- Channels 0…15 = 0x00…0x0E of the STC are provided for SMACK devices.
- The more SMACK devices are taught-in, the longer it will take until every single mailbox is updated (up to 30 seconds).
- After receiving a “Fill Mailbox” (6C D2) command the transmitted values will be continuously updated in the mailbox. To clear the mailbox content send the same command again with #DB = 0.
- SMACK is only available in simple learn mode. Simple learn mode does not support repeater between smack sensor and the STC.

7.1 Teach-in SMACK Device via learn button (FFFB)

Example (HEX):

Preamble	Command	channel	-/-	Checksum	Device address (XX)
A5 5A	FF FB	09	00 00 00 00 00 00 00 00	03	3F

Description: A SMACK device should be learned-in via the teach-in button on channel 09 on STC with address 63:
A55AFFFB0900000000000000000033F

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0xFF	FF
3	COMMAND_B	Command B = 0xFB	FB
4	FILTER_TABLE_CHANNEL	Filter channel 0…14 = 0x00…0x0E	09
5…12	-/-	0x00	00
13	CHECKSUM	Checksum of bytes 2 - 12	03
14	ADDRESS	address STC (Dip Switch 1-6)	3F

1. Response to received command

Description: Command FF FB was received. There are already taught-in 02 (Byte 6) SMACK devices.

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	COMMAND_A	Command A = 0xFF	FF
4	COMMAND_B	Command B = 0xFB	FB
5	FILTER_TABLE_CHANNEL	Filter channel 0…14 = 0x00…0x0E	09
6	NUMBER_OF_TEACHED_IN_SMACK_DEVICES	Number of SMACK devices 0…14 = 0x00…0x0E	02
7	MAX_SMACK_DEVICES	Max number of SMACK devices 0x0F	0F
8…11	-/-	0x00	00
12	SMACK_TEACH_IN_STATE	0x00 = OK 0xFF = Error default 0xFE=FILTER_TABLE_CHANNEL overload 0xFD = TCM communication failed	00
13	CHECKSUM	Checksum of bytes 0 - 12	52

Note: FILTER_TABLE_CHANNEL must be lower than MAX_SMACK_DEVICES.

2. Response – Device Teached-In (0F 01)

Description: A SMACK Device (Profile D2-00-01) with ID 01 8B 0C 32 was taught-in via learn button in channel 09 on STC with address 63

A55A3F0F0109D20001018B0C32F4

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	COMMAND_A	Command A = 0x0F	0F
4	COMMAND_B	Command B = 0x01	01
5	FILTER_TABLE_CHANNEL	Filter channel = 0…14 = 0x00…0x0E	09
6	ORG	telegram type	D2
7	FUNC	Device function	00
8	TYPE	Device type	01
9	ID_BYTE3	32-Bit EnOcean ID	01
10	ID_BYTE2		8B
11	ID_BYTE1		0C
12	ID_BYTE0		32
13	CHECKSUM	Checksum of bytes 0 - 12	F4

Note:
For the SMACK-Teach-In process it is important that only one gateway is in the teach-in mode. A SMACK device may only be taught-in at one filter channel. Do not teach-in a SMACK device in more than one gateway.

7.2 Fill Mailbox for SMACK device / send data to SMACK device (6CD2)

This command is for filling a mailbox of a SMACK device. The STC saves the data and will send it after receiving a request from the SMACK device. The data accepted by a SMACK device is shown in the EEPs. If the mailbox is filled once, the STC saves the data and will response every request with the saved data. For clearing the mailbox it is necessary to send a "Fill Mailbox" command with NUMBER_OF_DATA_BYTES = #DB = 0. The response to a SMACK device request will be: Mailbox empty.

Actually the maximal number of data bytes for filling the mailbox is 18. The value NUMBER_OF_DATA_BYTES = #DB must be set correctly. This value determines how many data bytes of the transmitted telegram is interpreted and sent via EnOcean. Unused data bytes should be filled with 0.

Example:
Four data bytes for a SR06 LCD should be sent to the mailbox at the STC. The SR06 LCD is taught-in at channel 14 and is configured as Smart Acknowledge device.

- ➔ #DB = 4 -> DB3…0 -> 4 data Bytes
- ➔ Fill unused data bytes with zero: DB17…4 = 0
- ➔ Transmitted data (4 Bytes) depends on the EEP of the SMACK device (here: D2-11-03)

				DB17…4	DB3…0	Byte	
Sync	Command	channel	#DB	14 Bytes	4 Bytes	Checksum	XX
A5 5A	6C D2	0E	04	14* 0x00	A1 10 15 31	47	3F

A5 5A 6C D2 0E 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A1 10 15 31 47 3F

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	COMMAND_A	Command A = 0x6C	6C
3	COMMAND_B	Command B = 0xD2	D2
4	FILTER_TABLE_CHANNEL	Filter channel 0…15 =0x00…0x0E	0E
5	NUMBER_OF_DATA_BYTES	#DB = 0…18 = 0x00…0x12	04
6…19	DATA_BYTE17…4	data byte 17…4	00
20	DATA_BYTE3	data byte 3	A1
21	DATA_BYTE2	data byte 2	10
22	DATA_BYTE1	data byte 1	15
23	DATA_BYTE0	data byte 0	31
24	CHECKSUM	Checksum of bytes 2 - 22	47
25	ADDRESS	address STC (Dip Switch 1-6)	3F

Note: The more SMACK devices/mailboxes are used, the longer it will take until every single mailbox is updated (up to 30 seconds)

Response

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	COMMAND_A	Command A = 0x6C	6C
4	COMMAND_B	Command B = 0xD2	D2
5	FILTER_TABLE_CHANNEL	Filter channel = 0…15 = 0x00…0x0E	0E
6	ID_BYTE3	32 Bit EnOcean ID	01
7	ID_BYTE2		8D
8	ID_BYTE1		FE
9	ID_BYTE0		56
10…11	-/-	0x01	01
12	SMACK_STATE	SMACK device 0x00 = Yes; 0x01 = No	00
13	CHECKSUM	Checksum of bytes 0 – 12	6E

Note: SMACK_STATE = this error may occur if the selected channel is not a SMACK device or the device is not taught-teached in correctly.

8 Send EnOcean telegrams via EVC

The STC-RS485-EVC can be used to send EnOcean telegrams. Therefore, an address range of 127 addresses is reserved. The data bytes and the ORG-Byte can be set completely free. Since software version 3.0.0 optional data and addressed destination telegrams (ADT) are supported.

8.1 Send F6, D5 and A5 EnOcean telegrams (6B)

Byte	Bit7 Bit0			Description
0	SYNC_BYTE1 (A5 Hex)			Preamble
1	SYNC_BYTE0 (5A Hex)			
2	COMMAND_A (6B Hex)			Command: Send telegram = 6B
3	ORG			ORG-Byte = F6, D5 or A5
4	DATA_BYTE3			data bytes 0...3
5	DATA_BYTE2			
6	DATA_BYTE1			
7	DATA_BYTE0			
8	ID_BYTE3			32-Bit TCM -ID
9	ID_BYTE2			
10	ID_BYTE1			
11	ID_BYTE0 (+ 0…127)			
12	STATUS (4Bit)	T-C (2Bit)	RP-C (2Bit)	Status and repeat counter
13	CHECKSUM			Checksum of bytes 2 - 12
14	ADDRESS			address STC (Dip switch 1-6)
15	SYNC_BYTE1 (B5 Hex)			Preamble of optional data
16	SYNC_BYTE0 (5B Hex)			
17	DESTINATION_ID_BYTE3			Destination address (ADT) 0xFFFFFFFF = Broadcast
18	DESTINATION_ID_BYTE 2			
19	DESTINATION_ID_BYTE 1			
20	DESTINATION_ID_BYTE 0			
21	reserved			0x00
22	CHECKSUM_OPTIONAL_DATA			Checksum of bytes 15…21

8.2 Send VLD (D2) and MSC (D1) EnOcean telegrams (6B)

Byte	Bit7 Bit0			Description
0	SYNC_BYTE1 (A5 Hex)			Preamble
1	SYNC_BYTE0 (5A Hex)			
2	COMMAND_A (6B Hex)			Command: Send telegram = 6B
3	ORG			ORG-Byte = D1 or D2
4	NUMBER_OF_DATA_BYTES			Number of transmitted data bytes (1…14)
5	DATA_BYTE13			data byte 13
6…17	DATA_BYTE12…1			data bytes 12…1
18	DATA_BYTE0			data byte 0
19	ID_BYTE3			32-Bit EnOcean ID
20	ID_BYTE2			
21	ID_BYTE1			
22	ID_BYTE0 (+ 0…127)			
23	STATUS (4Bit)	T-C (2 Bit)	RP-C (2 Bit)	Status and repeat counter
24	CHECKSUM			Checksum of bytes 2 - 23
25	ADDRESS			address STC (Dip Switch 1-6)
26	SYNC_BYTE1 (B5 Hex)			Preamble of optional data
27	SYNC_BYTE0 (5B Hex)			
28	DESTINATION_ID_BYTE3			Destination address (ADT) 0xFFFFFFFF = Broadcast
29	DESTINATION_ID_BYTE 2			
30	DESTINATION_ID_BYTE 1			
31	DESTINATION_ID_BYTE 0			
32	reserved			0x00
33	CHECKSUM_OPTIONAL_DATA			Checksum of bytes 25…31

Note:
NUMBER_OF_DATA_BYTES determines how many data bytes are interpreted and sent via EnOcean.

If the VLD/MSC telegram is an addressed destination telegram (Destination address ≠ FF FF FF FF) only nine data bytes (DATA_BYTE8…DATA_BYTE0) can be transmitted. All other data bytes are ignored.

If ID_BYTE0 (0…127) exceeds the maximal value of 127 the ID offset will begin at 0 again
(0 = 0; 1 = 1; … ; 127 = 127; 128 = 0; 129 = 1; … 255 = 127)

Example: ID_BYTE0 = 0x81 = 129 -> 0x01 = 1 = EnOcean Base ID + 1 = Sender ID

8.3 Response to command: send EnOcean telegram (6B)

The STC-RS485-EVC will return an acknowledge via EVC after receiving the sent EnOcean telegram command:

Byte	Bit7 Bit0	Description
0	SYNC_BYTE1 (A5 Hex)	Preamble
1	SYNC_BYTE0 (5A Hex)	
2	ADDRESS	address STC (Dip Switch 1-6)
3	COMMAND	0x6B
4	RETURN_CODE	0x58 = EnOcean telegram sent 0x59 = telegram sent as broadcast (no optional data available) 0xFF = ERROR
5	ID_BYTE3	32-Bit EnOcean ID
6	ID_BYTE2	
7	ID_BYTE1	
8	ID_BYTE0 (+ 0…127)	
9	OPTIONAL_DATA_STATE	State of optional data
10	ORG_BYTE	ORG_BYTE of the telegram sent; 0xFF: Error: unknown ORG Byte
11…12	-/-	
13	CHECKSUM	Checksum of Bytes 0 - 12

Note: If optional data is activated (via command FFFF) but not transmitted, the destination address will be 0xFF FF FF FF (Broadcast).

OPTIONAL_DATA_STATE:

Bit0: 0 = optional data not required 1 = optional data required
Bit1: 0 = optional data not available 1 = optional data available

Bit1	Bit0	DEZ	Description Bit1	Description Bit0	Result
0	0	0	optional data not available	optional data not required	OK
0	1	1	optional data not available	optional data required	Broadcast: Return Code = 0x59
1	1	3	optional data available	optional data required	OK

9 Receiving EnOcean telegrams via EVC

This chapter describes the telegram structure of received EnOcean telegrams via RS485-EVC. Generally every faultlessly received EnOcean based telegram is transmitted via RS485-EVC. The interpretation of the data bytes depends on the device type. For further information, refer to the associated data sheet as well as the EEPs.

9.1 Explanations

9.1.1 Bus Access

Before a telegram is sent, the STC examines the RS485 bus for a bus silent interval. If two receivers are trying to send to one bus at the same time, the telegrams get lost. Thus, the telegram is repeatedly sent two times, whereas the bus access is additionally event-controlled. As the address adjusted affects the calculation of the random time, the same is only allowed to be allocated once per bus segment.

9.1.2 Repeater Function

If radio repeaters are used for an extension of the radio path, theoretically the possibility is given that a receiver gets the original telegram and a few milliseconds later the repeater telegram. If this happens, the telegram of the repeater is rejected and not output to the RS485.

9.1.3 VLD and MSC telegrams

9.1.3.1 Telegram length

VLD and MSC telegrams will be transmitted in 35 Bytes. This length is independent of the EnOcean telegram received. How many data bytes may be interpreted is shown in Byte 4: DATA_LENGTH.

9.1.3.2 Addressed destination telegrams

If a VLD/MSC telegram is a destination addressed telegram only 9 data bytes can be transmitted.

9.1.4 Status

Like the data bytes, the meaning of the 4 bit field „STATUS“ is also depending on the respective device type.

9.1.5 Telegram counter

A radio telegram is output to the RS485 for three times. The 2bit field „T-C“ shows which of the three RS485 telegrams is concerned. (Values: 0, 1, 2)

9.1.6 Repeat counter

This field shows whether the radio telegram received is the original telegram of the sensor or whether it was passed on to the receiver via the radio repeater. (Values: 0, 1)

9.2 Telegram structure of RPS, 1BS and 4BS telegrams

Example: data Bytes (3…0): 00 72 94 09
Sender ID: 01 85 B8 C4
RSSI: 0x2E = 46 -> -46 dBm

A5 5A 3F 07 00 72 94 09 01 85 B8 C4 08 5E B5 5B 00 FF FF FF FF 2E 02 3C

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3F
3	ORG	ORG Byte = RPS/1BS/4BS = F6/D5/A5	07 = 4BS
4	DATA_BYTE3	data Bytes 0...3	00
5	DATA_BYTE2		72
6	DATA_BYTE1		94
7	DATA_BYTE0		09
8	ID_BYTE3	32-Bit sensor ID	01
9	ID_BYTE2		85
10	ID_BYTE1		B8
11	ID_BYTE0		C4
12	STATUS (4Bit) / T-C (2 Bit) / RP-C (2 Bit)	Status and repeat counter	08
13	CHECKSUM	Checksum of bytes 0 - 12	5E
14	SYNC_BYTE1 (B5 Hex)	Preamble of optional data	B5
15	SYNC_BYTE0 (5B Hex)		5B
16	Reserved	0x00	00
17	DESTINATION_ID_BYTE 3	Destination address (ADT) 0xFFFFFFFF = Broadcast	FF
18	DESTINATION_ID_BYTE 2		FF
19	DESTINATION_ID_BYTE 1		FF
20	DESTINATION_ID_BYTE 0		FF
21	**R**eceived **S**ignal **S**trength **I**ndication (RSSI)	**RSSI** [dBm] (Wert decimal, without minus) (-90…-45 = bad…good)	2E
22	FILTER_TABLE_CHANNEL	Filter channel 0x00…0x3F (only filter mode); else 0xFF	02
23	CHECKSUM_OPTIONAL_DATA	Checksum of bytes 14…21	3C

9.3 Telegram structure of VLD and MSC telegrams

Example: Data length: 0x0B = 11 data Bytes -> DATA_BYTE0…DATA_BYTE10
Data Bytes (10…0): 5E 4D 3C 2B 1A FF EE DD CC BB AA
Data Bytes (13…11): 00 00 00
Sender ID: 01 86 A7 C6
RSSI: 0x30 = 48 -> -48 dBm

A5 5A 3E D2 0B 00 00 00 5E 4D 3C 2B 1A FF EE DD CC BB AA 01 86 A7 C6 C8 FD B5 5B 01 FF FF FF FF 30 00 3D

Byte	Bit7 Bit0	Description	Example (HEX)
0	SYNC_BYTE1 (A5 Hex)	Preamble	A5
1	SYNC_BYTE0 (5A Hex)		5A
2	ADDRESS	address STC (Dip Switch 1-6)	3E
3	ORG	ORG Byte, 0xD1 = MSC or 0xD2 = VLD	D2
4	DATA_LENGTH = x = 1…14	Number of following data bytes 1…14	0B
5-18	DATA_BYTE_13…DATA_BYTE_0	data byte 13…0	00 00 00 5E 4D 3C 2B 1A FF EE DD CC BB AA
19	ID_BYTE3	32-Bit sensor ID	01
20	ID_BYTE2		86
21	ID_BYTE1		A7
22	ID_BYTE0		C6
23	STATUS (4Bit) | T-C (2Bit) | RP-C (2Bit)	Status and repeat counter	C8
24	CHECKSUM	Checksum of bytes 0…9+x	FD
25	SYNC_BYTE1 (B5 Hex)	Preamble of optional data	B5
26	SYNC_BYTE0 (5B Hex)		5B
27	reserved	0x00	01
28	DESTINATION_ID_BYTE3	Destination address (ADT) 0xFFFFFFFF = Broadcast	FF
29	DESTINATION_ID_BYTE 2		FF
30	DESTINATION_ID_BYTE 1		FF
31	DESTINATION_ID_BYTE 0		FF
32	**R**eceived **S**ignal **S**trength **I**ndication (RSSI)	**RSSI** [dBm] (Wert decimal, without minus) (-90…-45 = bad…good)	30
33	FILTER_TABLE_CHANNEL	Filter channel 0x00…0x3F (only filter mode); else 0xFF	00
34	CHECKSUM_OPTIONAL_DATA	Checksum of bytes 11+x….19+x	3D

10 Configuration Software[5]

By means of the configuration software, the device can be parameterized and the sensor data can be read out.

Com-Parameter

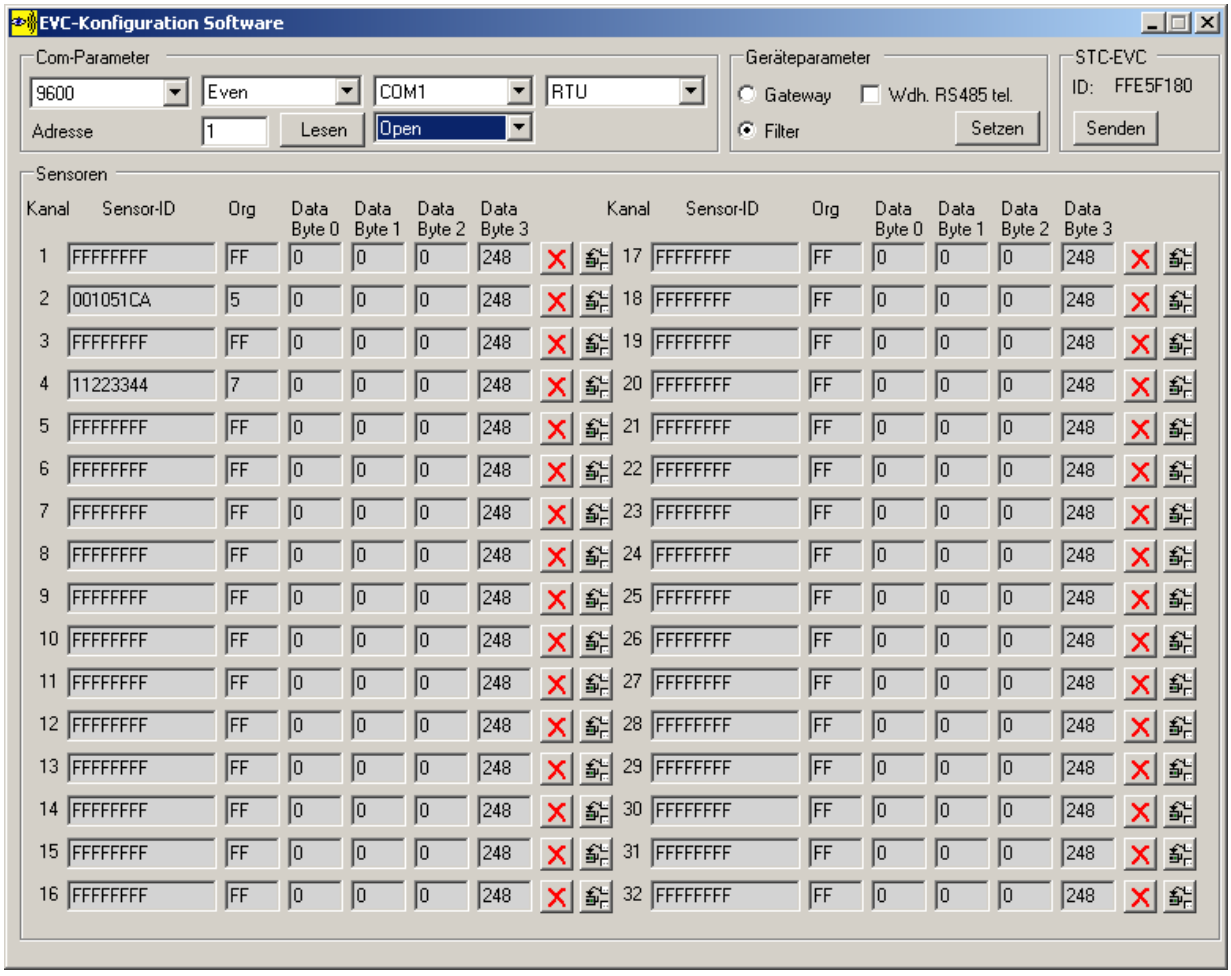
- Setting of communication properties
- “Read“ the device is read out again
- address -> setting of the device that shall be configured

Device property

- Setting of gateway or filter model
- By means of a hook at “Rea. RS485 tel.“ the telegrams are sent to the bus for 3 times otherwise only once
- “Set“ the device takes over all settings and is read out again

STC-EVC

- “ID” is the wireless address of the STC-EVC. If the field is empty, the read out procedure must be started again or the device cannot send any telegrams.
- “Send“ a new window is opened. In this window telegrams can be sent.



[5] A new Release of the configuration software will be released soon.