ILLUMRA™

SELF-POWERED WIRELESS CONTROLS

Bluetooth BLE
Single and Dual Rocker Switch
USER MANUAL

Part Numbers: BTT-S1AWH & BTT-S2AWH (white)

Observe precautions! Electrostatic sensitive devices!

Patent protected:
WO98/36395, DE 100 25 561, DE 101 50 128,
WO 2004/051591, DE 103 01 678 A1, DE
10309334, WO 04/109236, WO 05/096482,
WO 02/095707,
US 6,747,573, US 7,019,241

REVISION HISTORY

The following major modifications and improvements have been made to this document:

Version	Author	Reviewer	Date	Major Changes
1.0	JFF		04.04.2017	Initial Release

Published by www.ILLUMRA.com, info@ILLUMRA.com, phone 801-349-1200

Important!
This information describes the type of component and shall not be considered as assured characteristics. No responsibility is assumed for possible omissions or inaccuracies. Circuitry and specifications are subject to change without notice. For the latest product specifications, refer to the ILLUMRA website: http://www.ILLUMRA.com.

As far as patents or other rights of third parties are concerned, liability is only assumed for switches, not for the described applications, processes and circuits. ILLUMRA does not assume responsibility for use of switches described and limits its liability to the replacement of switches determined to be defective due to workmanship. Devices or systems containing RF components must meet the essential requirements of the local legal authorities. The switches must not be used in any relation with equipment that supports, directly or indirectly, human health or life or with applications that can result in danger for people, animals or real value. Components of the switches are considered and should be disposed of as hazardous waste. Local government regulations are to be observed.
Packing: Please use the recycling operators known to you.

TABLE OF CONTENTS

1 GENERAL DESCRIPTION

1.1 Basic functionality

BTT-SxAyy enables the realization of energy harvesting wireless switches communicating based on the 2.4 GHz Bluetooth Low Energy (BLE) radio standard.

BTT-SxAyy pushbutton transmitters are self-powered (no batteries) and fully maintenance-free. They can therefore be used in all environments including locations that are difficult to reach or within hermetically sealed housings. The required energy is generated by an electrodynamic energy transducer actuated by pressing the switch.

When the switch is pushed down or released, electrical energy is created and a 2.4GHz radio telegram according to the Bluetooth BLE standard is transmitted. This radio telegram transmits the operating status of all two or four buttons depending on the model.

BTT-SxAyy telegram format has been defined to maximize compatibility with a wide range of devices including such supporting the Bluethooth standard. BTT-SxAyy radio telegrams are protected with AES-128 security based on a device-unique private key.

Figure 1 – BTT-SxAyy Product Photo

1.2 Technical data

Antenna	Integrated antenna
Output Power	+0 dBm
Radio Standard	BLE Advertising
Communication Range (Guidance Only)	75 m ideal line of sight / 10 m indoor environment
Radio Frequency (min / max)	2402 MHz / 2480 MHz
Default Radio Channel	BLE CH 37 / 38 / 39 (2402 MHz / 2426 MHz / 2480 MHz)
Advertising Events per press or release (min / max)	2 / 3
Data Rate and Modulation	1 Mbit/s GFSK
Configuration Interface	NFC Forum Type 2 Tag (ISO/IEC 14443 Part 2 and 3)
Device Identification	Unique 48 Bit Device ID (factory programmed)
Security	AES128 (CBC Mode) with Sequence Code
Power Supply	Integrated Kinetic Energy Harvester
Inputs	Single or Dual Rockers

1.3 Physical dimensions and mounting options

Dimensions of Single Rocker	4.5” x 2.75” x 0.62” (114 x 70 x 16 mm)
Dimensions of Double Rocker	4.5” x 2.75” x 0.62” (114 x 70 x 16 mm)
Weight of Single Rocker	3.9 oz (111g)
Weight of Dual Rocker	3.9 oz (111g)
Mounting	Screw or double sided tape onto flat surface

1.4 Environmental conditions

Operating Temperature	-25°C ... 65°C
Storage Temperature	-25°C ... 65°C
Humidity	0% to 95% r.h. (non-condensing)

1.5 Packaging information

Packaging Unit	96 units
Packaging Method	Each unit packed in a box, 96 units packed in a case

1.6 Ordering information

Par Number	Description	Frequency
BTT-S1AWH	Single Rocker Bluetooth Switch - White	2.4 GHz Bluetooth BLE
BTT-S2AWH	Dual Rocker Bluetooth Switch - White	2.4 GHz Bluetooth BLE

2 FUNCTIONAL INFORMATION

2.1 Product Overview

The Single and Dual rocker BlueTooth BLE Switches from ILLUMRA send the implementation of wireless remote controls without batteries. Power is provided by a built-in electrodynamic power generator. BTT-SxAyy device transmits data based on the 2.4GHz BlueTooth BLE standard.

2.2 Basic Functionality

BTT-SxAyy devices contain an electro-dynamic energy transducer which is actuated by an energy bow. This bow is pushed by an appropriate switch rocker mounted onto the device. An internal spring will release the energy bow as soon as it is not pushed down anymore.

When the energy bow is pushed down, electrical energy is created and a Bluetooth BLE radio telegram is transmitted which identifies the status (pressed or not pressed) of the four button. Releasing the energy bow similarly generates energy which is used to transmit a different radio telegram.

It is therefore possible to distinguish between radio telegrams sent when the energy bar was pushed and radio telegrams sent when the energy bar was released.

By identifying these different telegrams types and measuring the time between pushing and releasing of the energy bar, it is possible to distinguish between "Long" and "Short" button contact presses. This enables simple implementation of applications such as dimming control or blinds control including slat action.

2.3 User Interface

ZBT-SxAyy devices provide either 2 buttons (Single Rocker) or 4 buttons (Dual Rocker).

The state of the four button contacts (pressed or not pressed) is transmitted together with a unique device identification (32 Bit device ID) whenever the switch is pushed or released.

3 Telegram transmission

3.1 Radio channel parameters

BTT-SxAyy transmits advertising telegrams within the 2.4 GHz radio frequency band (2402MHz … 2480MHz) using the BLE advertising frame format.

By default, BTT-SxAyy will use the three BLE advertising channels (BLE Channel 37, 38 and 39) defined for transmission. The transmission of a radio telegram on these three advertising channels is called an Advertising Event.

Use of different radio channels within the frequency band from 2402 MHz to 2480 MHz is possible, see chapter 6.7.8.

The initialization value for data whitening is set as follows:

- For BLE channels is set according to specification (value = radio channel)
- For the custom radio channels the initialization value is equal to the offset from 2400 MHz (e.g. value = 3 for 2403 MHz)

Table 2 below summarizes radio channels supported by BTT-SxAyy.

Radio Channel	Frequency	Channel Type
BLE Radio Channels		
37	2402 MHz	BLE Advertising Channel
0	2404 MHz	BLE Data Channel
1	2406 MHz	BLE Data Channel
…		
10	2424 MHz	BLE Data Channel
38	2426 MHz	BLE Advertising Channel
11	2428 MHz	BLE Data Channel
12	2430 MHz	BLE Data Channel
…		
36	2478 MHz	BLE Data Channel
39	2480 MHz	BLE Advertising Channel
Custom Radio Channels		
40	2403 MHz	Custom Radio Channel
41	2405 MHz	Custom Radio Channel
…		
77	2477 MHz	Custom Radio Channel
78	2479 MHz	Custom Radio Channel

Table 1 – BTT-SxAyy supported radio channels

3.2 Default radio transmission sequence

BTT-SxAyy transmits telegrams in its standard configuration by using so-called Advertising Events.

An advertising event is defined as the transmission of the same radio telegram on all selected radio channels (by default this would be on BLE Channel 37, 38 and 39) one after another with minimum delay in between.

For reliability reasons, BTT-SxAyy will send several (minimum two, maximum three) advertising events for each button input. The resulting transmission sequence is shown in Figure 6 below.

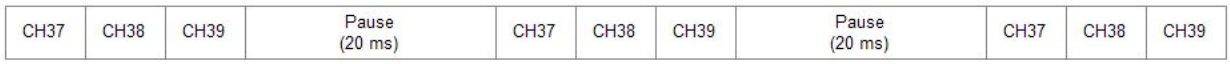


Figure 4 – Default radio transmission sequence

3.3 User-defined radio transmission sequences

In certain situations it might be desirable to transmit radio telegrams on channels other than the three advertising channels.

BTT-SxAyy therefore allows to select the radio channels to be used for the transmission of data telegrams and commissioning telegrams. The following transmission modes are supported:

- Both commissioning telegrams and data telegrams are transmitted on the advertising channels as three advertising events. This is the default configuration and described in chapter 3.2 above.
- Commissioning telegrams are transmitted on the advertising channels as three advertising events while data telegrams are transmitted in a user-defined sequence as described below.
- Both commissioning and data telegrams are transmitted in a user-defined sequence as described below.

The selection of the transmission mode is done using the CUSTOM CHANNEL MODE register of the NFC configuration interface as described in chapter 6.7.7.

BTT-SxAyy supports the following user-defined sequences:

- Three channel sequence
  This sequence is similar to the default Advertising Event with the difference that the user can select the radio channels to be used. The three channel sequence is described in chapter 3.3.1 below.

- Two channel sequence
  In this sequence the radio telegram is transmitted using four transmissions on two radio channels. It is described in chapter 3.3.2 below.

- One channel sequence
  In this sequence the radio telegram is transmitted using six transmissions on one radio channel. It is described in chapter 3.3.3 below.

3.3.1 Three channel sequence

The three channel radio transmission sequence is similar to the default transmission sequence. The difference is that the radio channels (BLE Channel 37, 38 and 39 in the default transmission sequence) can be selected using the Radio Channel Selection registers CH_REG1, CH_REG2 and CH_REG3.

The BTT-SxAyy telegram will in this mode be transmitted on the radio channel selected by CH_REG1 first, immediately followed by a transmission on the radio channel selected by CH_REG2 and a transmission on the radio channel selected by CH_REG3.

This transmission sequence will be sent three times in total with pauses of 20 ms in between as shown in Figure 7 below.

Figure 5 – Three channel radio transmission sequence

The format of CH_REG1, CH_REG2 and CH_REG3 is described in chapter 6.7.8.

3.3.2 Two channel sequence

The two channel radio transmission sequence removes transmission on the third radio channel (selected by CH_REG3) and instead repeats the transmission once more (four times in total).

The BTT-SxAyy telegram will in this mode be transmitted on the radio channel selected by CH_REG1 first, immediately followed by a transmission on the radio channel selected by CH_REG2.

This transmission sequence will be sent four times in total with pauses of 20 ms in between as shown in Figure 8 below.

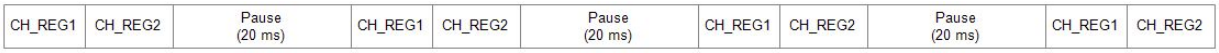


Figure 6 – Two channel radio transmission sequence

The format of CH_REG1 and CH_REG2 is described in chapter 6.7.8.

3.3.3 Single channel sequence

The single channel radio transmission sequence removes transmission on the second and third radio channel (selected by CH_REG2 and CH_REG3 respectively), i.e. all transmissions will be on the radio channel selected by CH_REG1.

The BTT-SxAyy telegram will be sent six times on this radio channel with pauses of 20 ms in between as shown in Figure 9 below.

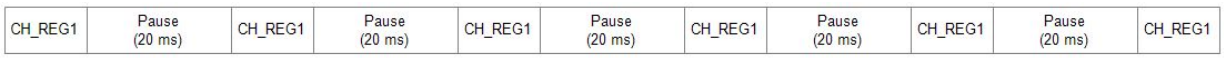


Figure 7 – Single channel radio transmission sequence

The format of CH_REG1 is described in chapter 6.7.8.

4 Telegram format

BTT-SxAyy transmits radio telegrams in the 2.4 GHz band according to BLE frame structure. For detailed information about the BLE standard, please refer to the applicable specifications.

Figure 10 below summarizes the BLE frame structure.

Preamble 0xAA	Access Address 0x8E89BED6	Header (2 Byte)	Source Address (6 Byte)	Payload (0 ... 31 Byte)	Check Sum (3 Byte)

Figure 8 – BLE frame structure

The content of these fields is described in more detail below.

4.1 Preamble

The BLE Preamble is 1 byte long and identifies the start of the BLE frame. The value of the BLE Preamble is always set to 0xAA.

4.2 Access Address

The 4 byte BLE Access Address identifies the radio telegram type. For advertising frames, the value of the Access Address is always set to 0x8E89BED6.

4.3 Header

The BLE Header identifies certain radio telegram parameters. Figure 11 below shows the structure of the BLE header.

TYPE (4 Bit)	UNUSED (2 Bit)	TX ADDR (1 Bit)	RX ADDR (1 Bit)	LENGTH (6 Bit)	UNUSED (2 Bit)
0010: TX-only Advertising (ADV_NONCONN_IND)	00	1: Random	0: Not used	Length of Address + Payload	00

Figure 9 – BLE header structure

4.4 Source address

The 6 byte BLE Source Address (MAC address) uniquely identifies each BTT-SxAyy product.

BTT-SxAyy supports two source address modes:

- Static Source Address mode (default)
  In this mode, the source address is constant (but its lower 32 bit can be configured via NFC interface)

- Private Resolvable Address mode (NFC configurable)
  In this mode, the source address changes for each transmission

BTT-SxAyy uses by default Static Source Address mode.

Private Resolvable Address mode can be selected by setting the Private Source Address flag in the Configuration register (see chapter 6.7.6) to 0b0. These two address

modes are described in the following chapters.

4.4.1 Static source address mode

By default, BTT-SxAyy uses static source addresses meaning that the source address is constant during normal operation. The static source address can be read and configured (written) via NFC as described in chapter 6.

The structure of BTT-SxAyy static addresses is as follows:

- The upper 2 bytes of the source address are used to identify the device type and set to 0xE215 for all BTT-SxAyy devices (to designate the use of an Illumra BTT-SxAyy module). These two bytes cannot be changed.

- The lower 4 bytes are uniquely assigned to each device. They can be changed using the NFC configuration interface as described in chapter 6.7.2

Figure 12 below illustrates the static address structure used by BTT-SxAyy.

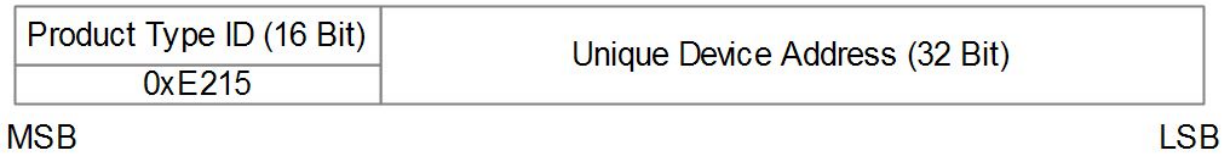


Figure 10 – BLE static source address structure

4.4.2 Private resolvable source address mode

For some applications it is desirable to modify (rotate) the source address used by BTT-SxAyy in order to prevent tracking of radio transmissions originating from a specific device. At the same time, each such device must remain uniquely identifiable by the receiver.

To achieve these goals, BTT-SxAyy can be configured via NFC to use random resolvable private addresses.

Using random resolvable private addresses requires that both BTT-SxAyy and the receiver both know a common key – the so-called Identity Resolution Key (IRK). BTT-SxAyy uses its device-unique random key as identity resolution key. This key can be configured via the NFC configuration interface as described in chapter 6.

For resolvable private addresses, the 48 bit address field is split into two sub-fields:

- prand
  This field contains a random number which always starts (two most significant bits) with 0b10. The prand value is changed for each telegram that is transmitted. Individual advertising events used to transmit one telegram (as described in chapter 3) use the same prand value.

- hash
  This field contains a verification value (hash) generated from prand using the

IRK The structure of a random resolvable private address is shown in Figure 13

below.

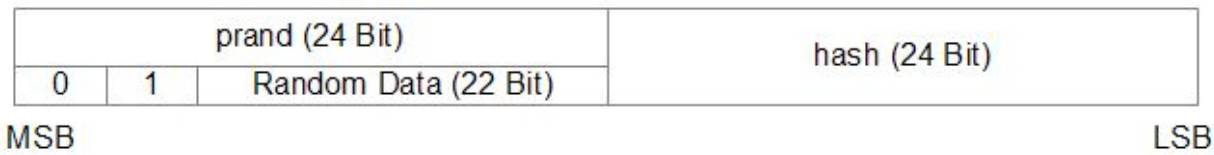


Figure 11 – BLE private resolvable source address structure

The prand value is encrypted using the IRK. The lowest 24 bit of the result (encrypted value) are then used as hash.

The concatenation of 24 bit prand and 24 bit hash will be transmitted as 48 bit private resolvable source address.

The receiving device maintains a list of IRK for all transmitters that have been commissioned to work with it.

Whenever the receiving device receives a radio telegram with private resolvable source address (identified by the most significant bits being set to 0b10), it will itself generate a 24 bit hash from the 24 bit prand sequentially using the IRK of each device that it has been learned into it.

If an IRK matches (i.e. when prand is encoded with this specific IRK then the result matches hash), then the receiver has established the identity of the transmitter.

So conceptually the IRK takes the role of the device source address while prand and hash provide a mechanism to select the correct IRK among a set

of IRK. This mechanism is illustrated in Figure 14 below.

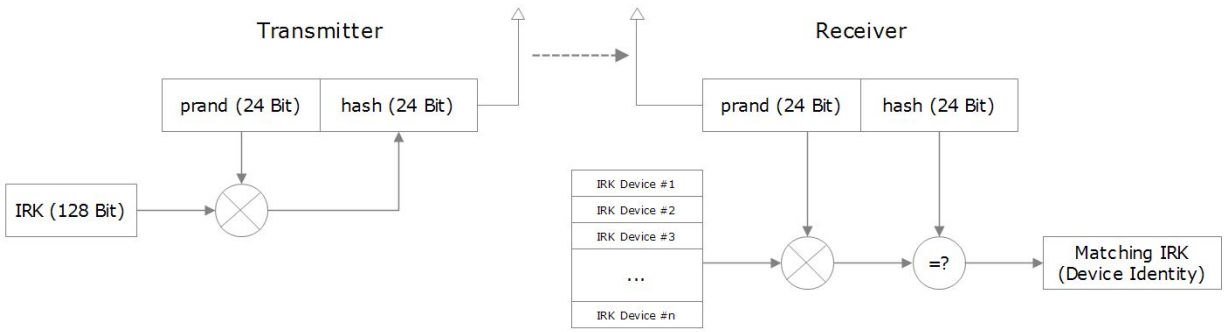


Figure 12 – Resolving private source addresses

4.5 Check Sum

The 3 byte BLE Check Sum is used to verify data integrity of received BLE radio telegrams. It is calculated as CRC (cyclic redundancy check) of the BLE Header, Source Address and Payload fields.

4.6 Payload

The payload of data telegrams is 13 … 17 bytes long (depending on the size of the Optional Data field) and consists of the following fields:

- Length (1 byte)
  The Length field specifies the combined length of the following fields. The content of the field depends on the size of the Optional Data field (which can be 0 / 1 / 2 or 4 byte).
  The resulting Length setting would be 12 / 13 / 14 or 16 byte (0x0C / 0x0D / 0x0E / 0x10) respectively

- Type (1 byte)
  The Type field identifies the data type used for this telegram. For BTT-SxAyy data telegrams, this field is always set to 0xFF to designate manufacturer-specific data field

- Manufacturer ID (2 byte)
  The Manufacturer ID field is used to identify the manufacturer of BLE devices based on assigned numbers. Illumra has been assigned 0x03DA as manufacturer ID code. The Manufacturer ID can be changed via the NFC configuration interface as described in chapter 6.7.4.

- Sequence Counter (4 byte)
  The Sequence Counter is a continuously incrementing counter used for security processing. It is initialized to 0 at the time of production and incremented for each telegram (data telegram or commissioning telegram) sent.

- Switch Status (1 byte)
  The Switch Status field reports the button action. The encoding of this field is described in chapter 4.7.

- Optional Data (0 / 1 / 2 or 4 byte)
  BTT-SxAyy provides the option to transmit additional user-defined data within each data telegram. This data can be used to identify user-specific properties.
  The length of the Optional Data field is defined in the Configuration register as described in chapter 6.7.6.

- Security Signature (4 byte)
  The Security Signature is used to authenticate BTT-SxAyy radio telegrams as described in chapter 4.8

Figure 15 below illustrates the data telegram payload.

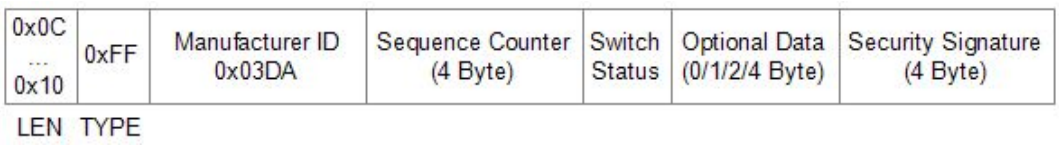


Figure 13 – Data telegram payload structure

4.7 Switch status encoding

The Switch Status field within the Payload data identifies the BTT-SxAyy action (rocker push or release). BTT-SxAyy uses the following sequence to identify and transmit the rocker status:

1. Determine direction of the rocker movement (Push Action or Release Action)
2. Read input status of all button contacts
3. Calculate data payload
4. Calculate security signature

In BTT-SxAyy, the type of action (Press Action or Release Action) is indicated by Bit 0 (Energy Bar). If a button contact has been actuated during Press Action or Release Action then this is indicated by the according status bit set to '1'.

Note that all contacts that were pressed during Press Action will be released during Release Action. The case of continuing to hold one (or several) button contacts during Release Action is mechanically not possible.

The switch status encoding used by BTT-SxAyy is shown Figure 16 in below.

Switch Status							
Reserved			B1	B0	A1	A0	ACTION TYPE
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Shall be 0b000			0 = No Action 1 = Action	0 = No Action 1 = Action	0 = No Action 1 = Action	0 = No Action 1 = Action	0 = Release Action 1 = Press Action

Figure 14 - BTT-SxAyy button action encoding

In the dual rocker variant BTT-S2Ayy, one rocker actuates B1 and B0 while the other rocker actuates A1 and A0.

In the single rocker variant BTT-S1Ayy, the rocker actuates B1 and B0. The buttons A1 and A0 are not used.

The direction of the actuation (press or release is indicated by the ACTION TYPE field.

4.8 BTT-SxAyy telegram authentication

BTT-SxAyy implements telegram authentication to ensure that only telegrams from senders using a previously exchanged security key will be accepted. Authentication relies on a 32 bit telegram signature which is calculated as shown in Figure 17 below and exchanged as part of the radio telegram.

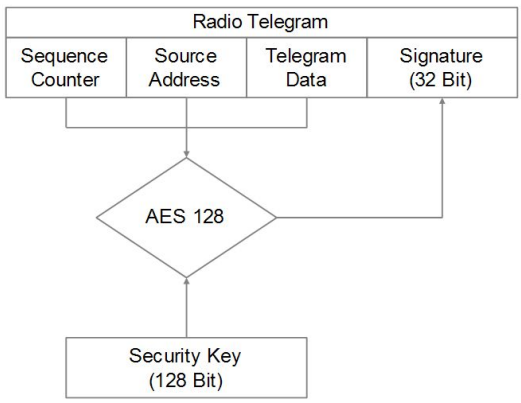


Figure 15 – Telegram authentication flow

Sequence counter, source address and the remaining telegram data together form the input data for the signature algorithm. This algorithm uses AES128 encryption based on the device-unique random security key to generate a 32 bit signature which will be transmitted as part of the radio telegram.

The signature is therefore dependent both on the current value of the sequence counter, the device source address and the telegram payload. Changing any of these three parameters will therefore result in a different signature.

The receiver performs the same signature calculation based on sequence counter, source address and the remaining telegram data of the received telegram using the security key it received from BTT-SxAyy during commissioning.

The receiver then compares the signature reported as part of the telegram with the signature it has calculated. If these two signatures match then the following statements are true:

- Sender (BTT-SxAyy) and receiver use the same security key
- The message content (address, sequence counter, data) has not been modified

At this point, the receiver has validated that the message originates from a trusted sender (as identified by its security key) and that its content is valid.

In order to avoid message replay (capture and retransmission of a valid message), it is required that the receiver tracks the value of the sequence counter used by BTT-SxAyy and only accepts messages with higher sequence counter values (i.e. not accepts equal or lower sequence counter values for subsequent telegrams).

4.8.1 Authentication implementation

BTT-SxAyy implements telegram authentication based on AES128 in CCM (Counter with CBC-MAC) mode as described in IETF RFC3610. At the time of writing, the RFC3610 standard could be found here: https://www.ietf.org/rfc/rfc3610.txt

The 13 Byte CCM Nonce (number used once – unique) initialization value is constructed as concatenation of 6 byte Source Address, 4 byte Sequence Counter and 3 bytes of value 0x00 (for padding).

Note that both Source Address and Sequence Counter use little endian format (least significant byte first).

Figure 18 below shows the structure of the AES128 Nonce.

AES128 Nonce (13 Byte)												
Source Address						Sequence Counter				Padding		
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 0	Byte 1	Byte 2	Byte 3	0x00	0x00	0x00

Figure 16 – AES128 Nonce structure

The AES128 Nonce and the 128 bit device-unique security key are then used to calculate a 32 bit signature of the authenticated telegram payload shown in Figure 19 below.

Authenticated Payload								
LEN	TYPE	MANUFACTURER	Sequence Counter				STATE	Optional Data
Byte 0	0xFF	0x03DA	Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	0 / 1 / 2 / 4 byte

Figure 17 – Authenticated payload

The calculated 32 bit signature is then appended to the data telegram payload as shown in Figure 15 in chapter 4.6.

In addition to the RFC3610 standard itself, please consult also Appendix A for a step by step description of the authentication process.

5 BTT-SxAyy commissioning

Commissioning is the process by which a BTT-SxAyy is learned into a receiver (actuator, controller, gateway, etc.).

The following two tasks are required in this process:

- Device identification
  The receiver needs to know how to uniquely identify the specific BTT-SxAyy. This is achieved by using a unique 48 Bit ID (Source Address) for each BTT-SxAyy device as described in chapter 4.4. In addition, up to 4 byte of Optional Data can be configured as described in chapter 6.7.5

- Security parameter exchange
  The receiver needs to be able to authenticate radio telegrams from BTT-SxAyy in order to ensure that they originate from this specific device and have not been modified as described in chapter 4.8. This is achieved by exchanging a 128 Bit random security key used by BTT-SxAyy to authenticate its radio telegrams.

BTT-SxAyy provides the following options for these tasks:

- NFC-based commissioning
  The BTT-SxAyy parameters are read by a suitable commissioning tool (e.g. NFC smartphone with suitable software) which is already part of the network into which BTT-SxAyy will be commissioned. The commissioning tool then communicates these parameters to the intended receiver of BTT-SxAyy radio telegrams. NFC-based commissioning is described in chapter 6

- Camera-based commissioning
  Each BTT-SxAyy module contains an optically readable Data Matrix Code (DMC) which identifies its ID and its security key. This DMC can be read by a suitable commissioning tool (e.g. smartphone) which is already part of the network into which BTT-SxAyy will be commissioned. The commissioning tool then communicates these parameters to the intended receiver of BTT-SxAyy radio telegrams. The DMC structure is described in chapter 5.2.1

- Radio-based commissioning
  BTT-SxAyy can communicate its parameters via special radio telegrams (commissioning telegrams) to the intended receiver. To do so, BTT-SxAyy can be temporarily placed into radio-based commissioning mode as described in chapter 5.3

5.1 NFC-based commissioning

All required BTT-SxAyy parameters can be read via a suitable NFC reader and writer supporting the ISO/IEC 14443 Part 2 and 3 standards. The actual NFC implementation uses a Mifare Ultralight tag.

Commissioning via NFC should follow these steps:

1. Unlock the BTT-SxAyy device by using the default NFC PIN code 0x0000 E215

2. Read the BTT-SxAyy Source Address, Security Key and Sequence Counter and configure the receiver accordingly

3. **Important**: The pre-programmed random security key used by BTT-SxAyy can be obtained both from the product DMC code as described in chapter 5.2, from received commissioning telegrams as described in chapter 5.3 and via the NFC interface.
   For security-critical applications where unauthorized users could have physical access to the switch it is therefore strongly recommended to change the security key to a new security key as part of the NFC-based commissioning process. To do so, follow the procedure outlined in chapter 6.7.3.
   For additional security, NFC read-out of the new security key can be disabled by setting the Private Security Keyflag in the Configuration register before setting the new security key.
   This ensures that even persons knowing the correct PIN code to configure this specific switch cannot read out the programmed new security key. Please verify that you have properly documented the new security key as there is no possibility to retrieve this after it has been written.

4. **Important**: It is strongly recommended to disable radio-based commissioning after programming a new security key. This ensures that the new security key cannot be read out by triggering a commissioning telegram as described in chapter 5.3.
   To disable radio-based commissioning, set the Disable Radio Commissioning flag in the Configuration register to 0b1, see chapter 6.7.6.

5. **Important**: You should always change the NFC PIN code from its default setting to a new NFC PIN code and lock the NFC configuration interface. This step is mandatory to avoid access to the BTT-SxAyy configuration using the default PIN code. Should you lose the new NFC PIN code then BTT-SxAyy can be reset to factory mode (with the default NFC PIN code) by means of a factory reset as described in chapter
   5.4. For security reasons, this factory reset will always reset the security key to its pre-programmed value.

5.2 Camera-based commissioning

Each BTT-SxAyy device contains an optically readable Data Matrix Code (DMC) on the lower right hand side of the device label which can be used to automatically scan device parameters.

Figure 20 below highlights (green rectangle) the location of the commissioning DMC code.

Figure 18 – Location of the commissioning DMC code

The DMC uses the ECC200 standard and can be read by a by a suitable commissioning tool (e.g. smartphone) which is already part of the network into which BTT-SxAyy will be commissioned.

The commissioning tool then sends these parameters to the intended receiver of BTT-SxAyy radio telegrams.

5.2.1 DMC format

The commissioning DMC provided by the BTT-SxAyy module uses the following format:

<PRODUCT_NAME> ID<SOURCE_ ID>OOB<DEV ICE_KEY>

This identifies the following parameters:

- Product name of the module (always "PTM215B")
- 48 bit Static Source Address (unique for each device, starts with "E215" Prefix)
- 128 bit device-unique random security key (different for each device)

One possible DMC reading could for instance be:

PTM215B IDE21501500100 OOB0123456789 ABCDEF0123 456789 ABCDEF

For better readability, the same reading is shown below coloured red, green and blue to identify the different parts:

PTM215B IDE21501 500100 OOB0123456789 ABCDEF0123 456789 ABCDEF

This particular DMC reading would identify the following parameters:

- Product name of the module = PTM215B
- Static Source Address = E21501500100
- Device-unique random security key = 01234 56789ABCDEF01234567 89ABCDEF

5.3 Radio-based commissioning

For cases where both NFC and camera-based commissioning are not feasible it is possible to set BTT-SxAyy into a specific mode where it transmits commissioning telegrams.

This functionality can be disabled via the NFC configuration interface by setting the Disable Radio Commissioning flag in the Configuration register to 0b1 (see chapter 6.7.6).

5.3.1 Commissioning mode entry

Commissioning mode is entered using a special button contact sequence. This is illustrated in Figure 21 below.

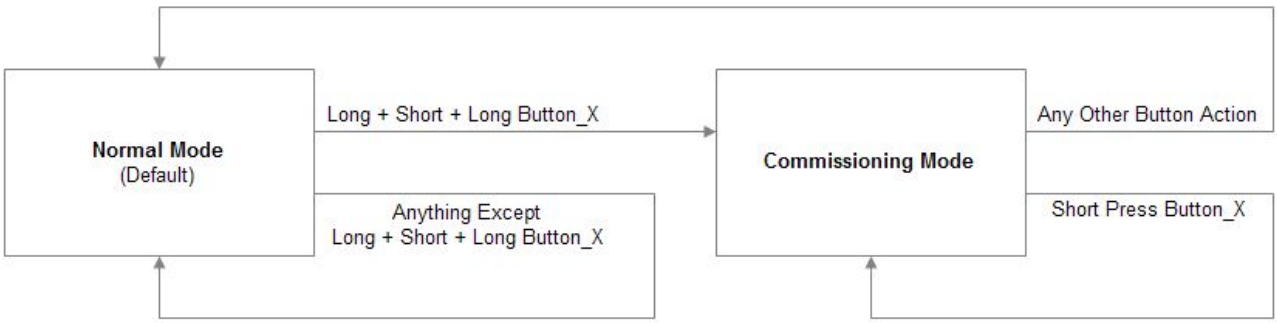


Figure 19 – Button sequence to enter radio-based commissioning mode

To enter commissioning mode, start by selecting one button (one side of one rocker) of BTT-SxAyy.

For the dual rocker (BTT-S2Ayy) case, this button can be either upper side of left rocker, lower side of left rocker, upper side of right rocker or lower side of right rocker. For the single rocker (BTT-S1Ayy) case, this can be either upper side of the rocker or lower side of the rocker. This selected button is referred to as Bu tt on_X in Figure 21 above.

Next, execute the following long-short-long sequence:

1. Press and hold the selected rocker on the selected side for more than 7 seconds before releasing it

2. Press the selected rocker on the selected side quickly (hold for less than 2 seconds)

3. Press and hold the selected rocker on the selected side again for more than 7 seconds before releasing it

Upon detection of this sequence, BTT-SxAyy will enter commissioning mode if the Disable Radio Commissioning flag in the Configuration register of the NFC interface is

set to 0b0 (default state).

5.3.2 Commissioning telegram transmission

BTT-SxAyy will transmit a commissioning telegram (on the radio channels selected as described in chapter 3.1) upon entering commissioning mode.

BTT-SxAyy will continue to transmit commissioning telegrams whenever the button used for entry into commissioning mode (Button_X) is pressed or released again.

The payload of commissioning telegrams is 30 bytes long and consists of the following fields:

- Length (1 byte)
  The Length field specifies the combined length of the following fields. For BTT-SxAyy commissioning telegrams, this field is always set to 0x1D to indicate 29 byte of manufacturer-specific data

- Type (1 byte)
  The Type field identifies the data type used for this telegram. This field is set to 0xFF to indicate a "Manufacturer-specific Data" field.

- Manufacturer ID (2 byte)
  The Manufacturer ID field is used to identify the manufacturer of BLE devices based on assigned numbers. By default, this field is set to 0x03DA (Illumra GmbH). This field can be changed via the NFC configuration interface as described in chapter 6.7.4.

- Sequence Counter (4 byte)
  The Sequence Counter is a continuously incrementing counter used for security processing. It is initialized to 0 at the time of production and incremented for each telegram (data telegram or commissioning telegram) sent.

- Security Key (16 byte)
  Each BTT-SxAyy module contains its own 16 byte device-unique random security key which is generated and programmed during manufacturing. It is transmitted during commissioning to enable the receiver to authenticate BTT-SxAyy data telegrams

- Static Source Address (6 byte)
  The Static Source Address is used to uniquely identify each BLE device. It is transmitted as part of the BLE frame as described in chapter 4.4.1.
  Some devices (most notable all iOS-based products) do not expose this address to their applications. This makes it impossible to use such applications to commission BTT-SxAyy. The Static Source Address is therefore again transmitted as part of the payload.

Figure 22 below illustrates the commissioning telegram payload.

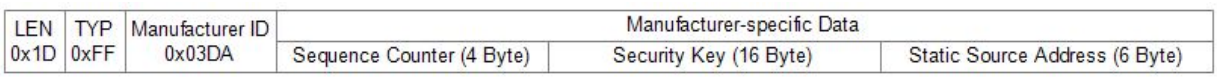


Figure 20 – Commissioning telegram payload structure

5.3.3 Exit from commissioning mode

Pressing any key except the button used for entry into commissioning mode (Button_X) will cause BTT-SxAyy to stop transmitting commissioning telegrams and return to normal data telegram transmission.

5.4 Factory Reset

BTT-SxAyy can be reset to its default settings by means of a factory reset.

This ensures that BTT-SxAyy can be reset to a known configuration in case the PIN for the NFC access has been lost or NFC access is not possible for other reasons

In order to execute such factory reset, the rocker(s) and the switch housing have to be removed from BTT-SxAyy so that all four BTT-SxAyy module contacts and the energy bar are accessible.

After that, all four button contacts (A0, A1, B0 and B1) have to be pressed at the same time while the energy bow of the BTT-SxAyy module is pressed down.

The energy bow must then be held at the down position for at least 10 seconds before being released. The button contacts A0, A1, B0 and B1 can be released at any time after pressing the energy bow down, i.e. it is no requirement to hold them as well for at least 10 seconds.

Upon detecting this input, BTT-SxAyy will restore the default settings of the following items:

- Static Source Address
- Security Key and Security Key Write register
  Both registers will be restored to the value of the factory-programmed security key
- Manufacturer ID
  The manufacturer ID will be reset to 0x03 DA (Illumra GmbH)
- NFC PIN Code
  The NFC PIN Code will be reset to 0x0000 E215

After such factory reset, Source Address and Security Key will again match the content of the DMC code on the unit label as described in chapter 7.

In addition, BTT-SxAyy will reset the following registers:

- Configuration register (to 0x00)
- Custom Channels Register (to 0x00)

6 NFC interface

BTT-SxAyy implements NFC Forum Type 2 Tag functionality as specified in the ISO/IEC 14443 Part 2 and 3 standards using an NXP NT3H2111 Mifare Ultralight tag.

This NFC functionality can be used to access (read and write) the BTT-SxAyy configuration memory and thereby configure the device as described in the following chapters.

Chapter 6.1 below gives an introduction to the NFC functionality and options to use the NFC interface.

For in-depth support for integrating the NXP NT3H2111 NFC functionality into PC or smartphone SW please contact NXP technical support.

6.1 Using the NFC interface

Using the NFC interface requires the following:

- NFC reader (either PC USB accessory or suitable smartphone / tablet)
- NFC SW with read, write, PIN lock, PIN unlock and PIN change functionality

Illumra recommends TWN4 from Elatec RFID Systems (https://www.elatec-rfid.com/en/) as USB NFC reader. This reader is shown in Figure 23 below.

Figure 21 – Elatec TWN4 MultiTech Desktop NFC Reader

TWN4 can be configured as CDC / Virtual COM port and can then be accessed like any serial interface. It provides all necessary commands for the NFC interface, specifically to:

- Read data from configuration memory and write data to configuration memory
- Authenticate the user (to allow read / write of protected memory) via 32 bit PIN

NFC functionality is also available in certain Android smartphones and tablets. NXP provides a SW framework that can be used with Android devices and can advise regarding suitable tablets and smartphones.

NFC communication distance is for security reasons set to require direct contact between reader and switches based on BTT-SxAyy.

6.2 NFC interface functions

For a detailed description about the NFC functionality, please refer to the ISO/IEC 14443 standard.

For specific implementation aspects related to the NXP implementation in NT3H2111, please refer to the NXP documentation which at the time of writing was available at this link:

http://cache.nxp.com/documents/data_sheet/NT3H2111_2211.pdf

The following chapters summarize the different functions for reference purposes.

6.2.1 NFC interface state machine

Figure 24 below shows the overall state machine of the NFC interface.

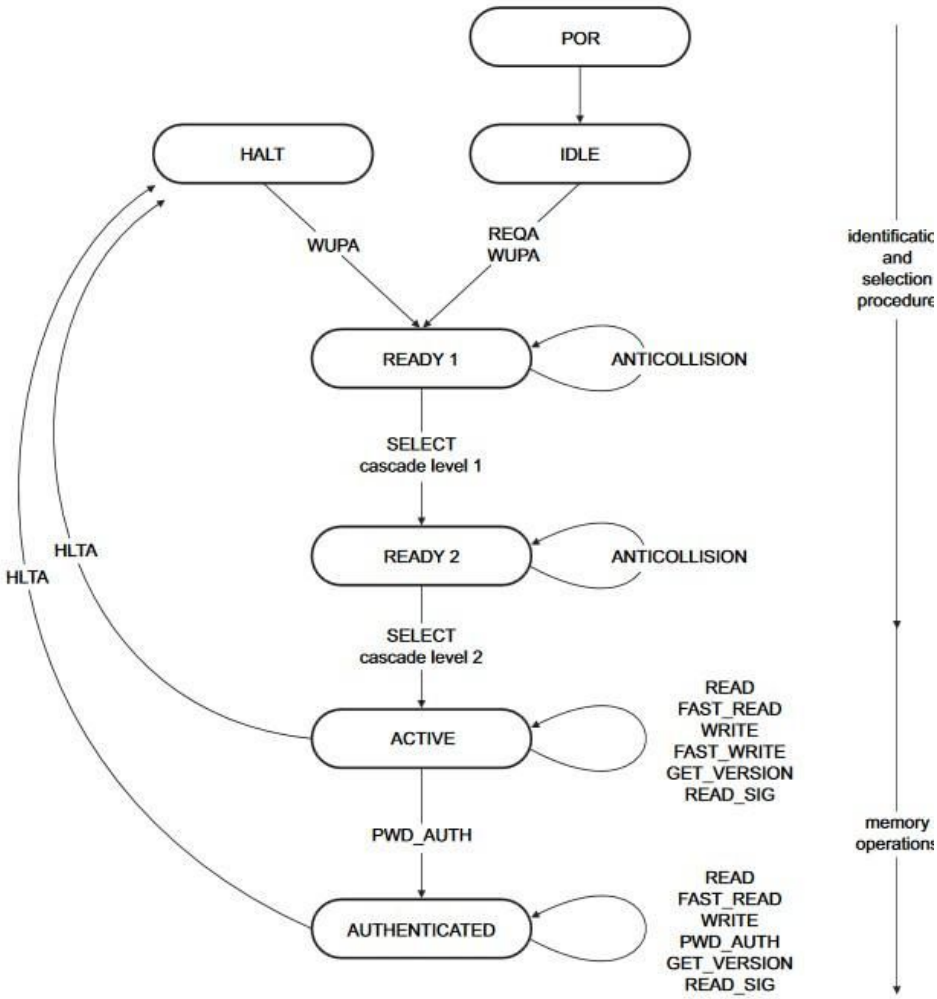


Figure 22 – NFC interface state machine

6.2.2 IDLE state

IDLE is the waiting state after a Power-On Reset (POR), i.e. after the NFC tag has been introduced into the magnetic field of the NFC reader.

The NFC tag exits the IDLE state towards the READY 1 state when either a REQA or a WUPA command is received from the NFC reader. REQA and WUPA commands are transmitted by the NFC reader to determine whether any cards are present within its working range.

Any other data received by the NFC tag while in IDLE state is discarded and the NFC tag will remain in IDLE state.

6.2.3 READY 1 state

READY 1 is the first UID resolving state where the NFC tag resolves the first 3 bytes of the 7 byte UID using the ANTICOLLISION or SELECT commands for cascade level 1.

READY 1 state is exited after the SELECT command from cascade level 1 after the matching complete first part of the UID has been executed. The NFC tag then proceeds into READY 2 state where the second part of the UID is resolved.

6.2.4 READY 2 state

READY 2 is the second UID resolving state where the NFC tag resolves the remaining 4 bytes of the 7 byte UID using the ANTICOLLISION or SELECT commands for cascade level 2.

READY 2 state is exited after the SELECT command from cascade level 2 with the matching complete part of the UID has been executed. The NFC tag then proceeds into ACTIVE state where the application-related commands can be executed.

6.2.5 ACTIVE state

ACTIVE state enables read and write accesses to unprotected memory.

If access to protected memory is required then the tag can transition from the ACTIVE state to AUTHENTICATED state by executing the PWD_AUTH command in conjunction with the correct 32 bit password.

6.2.6 Read command

The READ command requires a start page address, and returns the 16 bytes of four NFC tag pages (where each page is 4 byte in size).

For example, if the specified address is 03h then pages 03h, 04h, 05h, 06h are returned. Special conditions apply if the READ command address is near the end of the accessible memory area.

Figure 25 below shows the read command sequence.

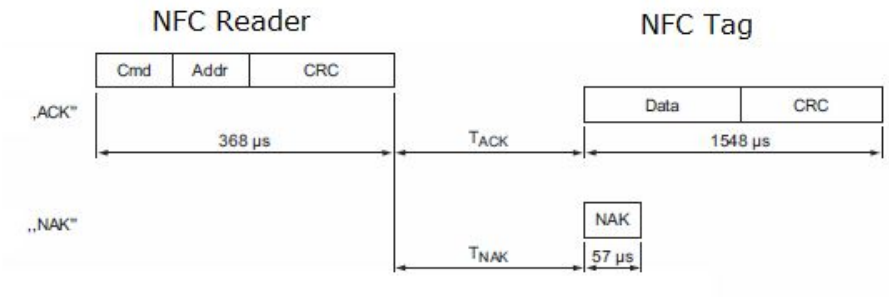


Figure 23 – NFC read command sequence

6.2.7 Write command

The WRITE command requires a start page address and returns writes 4 bytes of data into that page.

Figure 26 below shows the read command sequence.

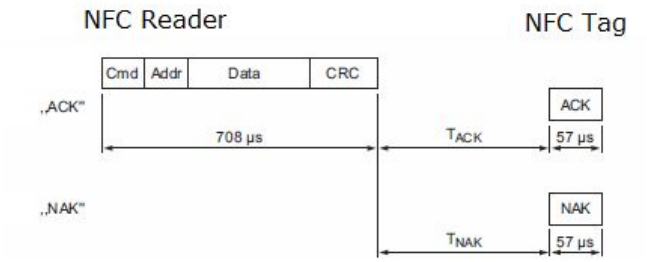


Figure 24 – NFC write command sequence

6.2.8 Password authentication (PWD_AUTH) command

The protected memory area can be accessed only after successful password verification via the PWD_AUTH command.

The PWD_AUTH command takes the password as parameter and, if successful, returns the password authentication acknowledge, PACK.

Figure 27 below shows the password authentication sequence.

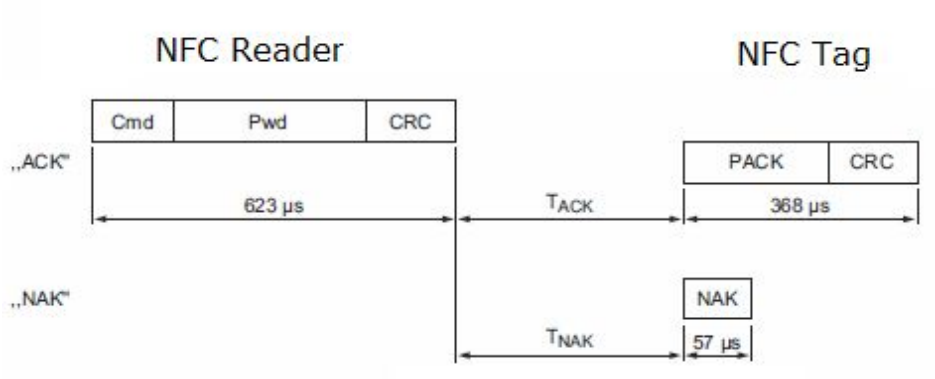


Figure 25 – Password authentication sequence

After successful authentication, the password can be changed by writing the new password to memory page 0xE5.

Note that a read access to page 0xE5 always return 0x0000 0000 , i.e. it is not possible to read out the current PIN code.

6.3 Using TWN4 as USB NFC reader

Elatec RFID Systems provides a PC software called "Director" as part of their software support package. At the time of writing, this was available at this address: https://www.elatec-rfid.com/en/download-center/contact-form-twn4-devpack-sdk/

Figure 28 below shows the user interface of this software.

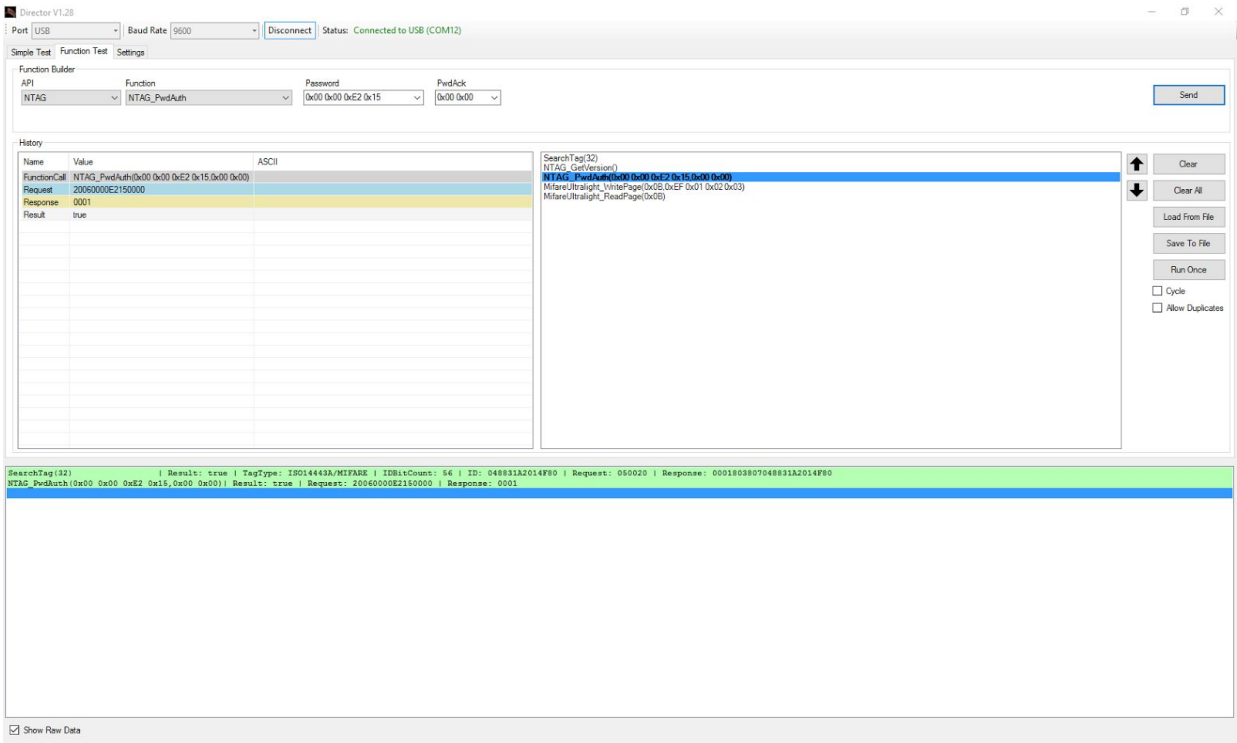


Figure 26 – User interface of TWN4 Director

By using this software, it is easily possible to generate the required serial commands that have to be sent via CDC / Virtual COM port to TWN4 and understand the structure of the response that will be received back.

6.3.1 Useful commands

The following commands are especially useful:

- Search Tag (maximum ID bytes)
  Used to search for a connected tag and identify type and ID of such tag. This should always be used as first operation ahead of any read / write / authenticate actions.
  Example: Search Tag (32)

- NTAG_PwdAuth (32 bit password as hex bytes, 16 bit password_ack as hex bytes)
  Used to authenticate access to the protected memory area Example:
  NTAG_PwdAuth (0x00 0x00 0xE2 0x15 , 0x00 0x00)

- NTAG_Read (page)
  Used to read one page
  of data Example:
  NTAG_Rea d (0x04)

- NTAG_Write (page , data)
  Used to write one page of data
  Example: NTAG_Write (0x40 , 0x12 0x34 0x56 0x78)

- NTAG_Write (0xE5 , PIN Code)
  Used to set a new pin code by writing to page 0xE5
  Example: NTAG_Write (0xE5 , 0x12 0x34 0x56 0x78)

6.3.2 Translation into binary data

In order to use these commands within a user application, they have to be translated into raw data. This can be done by enabling the "Show Raw Data" feature in the command log of the Director software as shown in Figure 29 below.

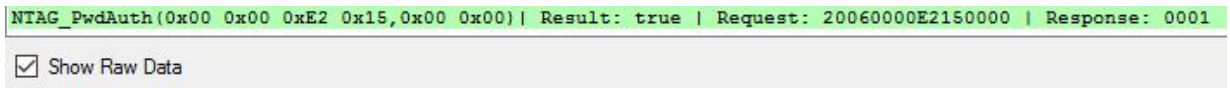


Figure 27 – Enabling raw data display

This raw data can then be transmitted to TWN4 via a virtual COM port. TWN4 will respond to the request with the corresponding response as shown in Figure 30 below.

Figure 28 – Binary data exchange

6.4 Configuration memory organization

The BTT-SxAyy configuration memory is divided into the following areas:

- Public data
- Protected data

In addition to that, BTT-SxAyy maintains a private configuration memory region used to store default parameters and confidential information which is not accessible to the user.

Figure 31 below illustrates the configuration memory organization used by BTT-SxAyy.

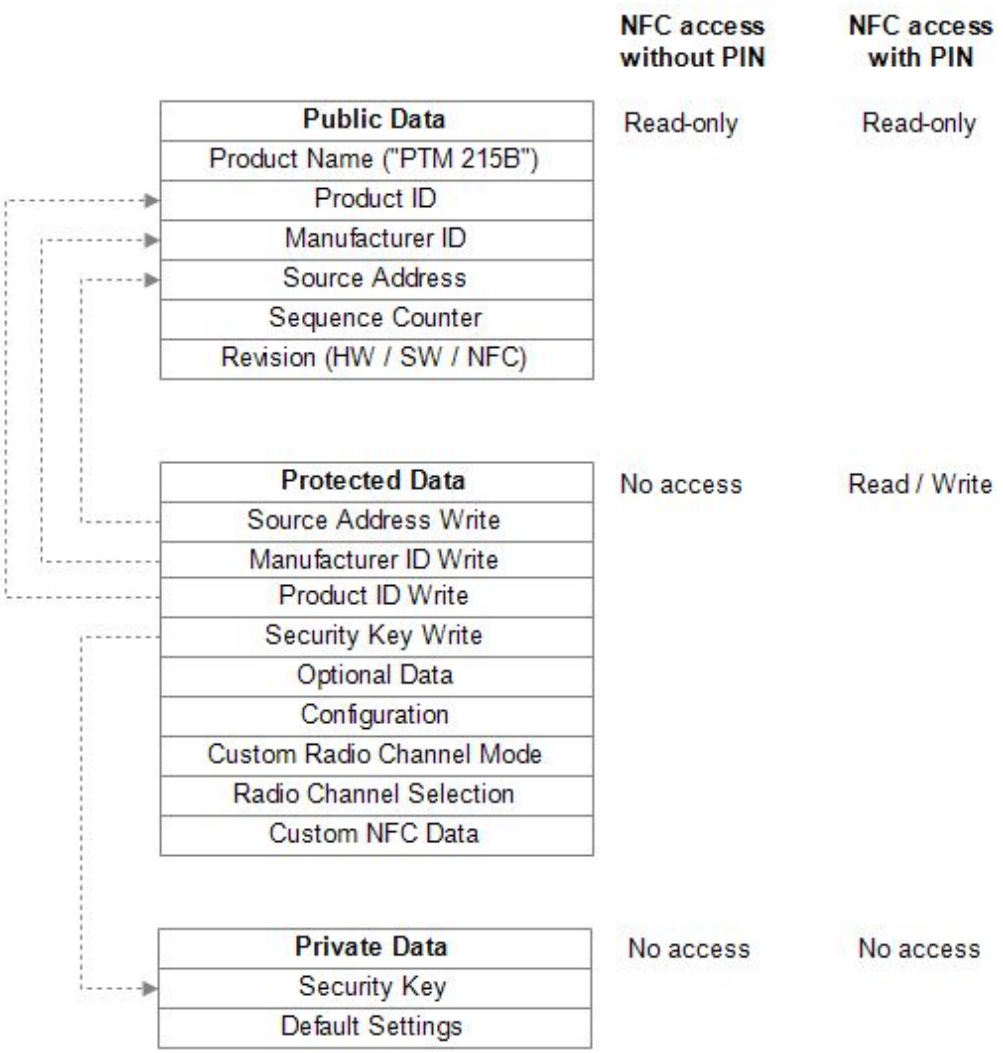


Figure 29 – Configuration memory organization

6.5 Memory Address Map

The NFC-accessible configuration memory is organized in memory pages where each memory page is 4 bytes wide. An NFC access reads 16 bytes (4 pages) or writes 4 bytes (one page). The addresses map of the configuration memory is shown in Table 3 below. The byte order is little endian, i.e. byte 0 will be read first and byte 3 last.

Area	NFC Page	Byte Offset	Byte 0 (LSB)	Byte 1	Byte 2	Byte 3 (MSB)
Public Memory Area						
Public	0 (0x00)	0	Reserved			
Public	…	…				
Public	3 (0x03)	12				
Public	4 (0x04)	16	Product Name "BTT-SxAyy"			
Public	5 (0x05)	20				
Public	6 (0x06)	24	Product ID			
Public	7 (0x07)	28				
Public	8 (0x08)	32	NFC Revision		Manufacturer ID	
Public	9 (0x09)	36	Reserved			
Public	10 (0x0A)	40	Hardware Revision			
Public	11 (0x0B)	44	Software Revision			
Public	12 (0x0C)	48	Static Source Address			
Public	13 (0x0D)	52	Sequence Counter			
Protected Memory Area						
Protected	14 (0x0E)	56	Configuration	Custom CH Mode	Reserved	
Protected	15 (0x0F)	60	Opt Data 0	Opt Data 1	Opt Data 2	Opt Data 3
Protected	16 (0x10)	64	Product ID Write			
Protected	17 (0x11)	68				
Protected	18 (0x12)	72	Source ID Write			
Protected	19 (0x13)	76	Manufacturer ID Write		Reserved	
Protected	20 (0x14)	80	Security Key Write			
Protected	…	…				
Protected	23 (0x17)	92				
Protected	24 (0x18)	96	CH_REG1	CH_REG2	CH_REG3	Reserved
Protected	25 (0x19)	100	Reserved			
Protected	…	…				
Protected	31 (0x1F)	124				
Protected	32 (0x20)	128	Customer NFC Data			
Protected	…	…				
Protected	95 (0x5F)	380				

Protected	96 (0x60)	384	Reserved
Protected	…	…	
Protected	225 (0x10)	900	
Protected	229 (0xE5)	916	PIN Code (Write Only)

Table 2 – Configuration memory address map

6.6 Public data

Public data can be read by any NFC-capable device supporting the ISO/IEC 14443 Part 2 and 3 standards. No specific security measures are used to restrict read access to this data.

The following items are located in the public data area:

- BTT-SxAyy Product Name
  This is always "PTM 215B" to designate the module used within the rocker switch

- BTT-SxAyy Product ID
  This is an 8 byte field which is by default set to 0x0000000000000 000.
  Product ID and Manufacturer ID can be configured by the customer as required to uniquely identify his PTM 215B based products, see chapter 6.7.4

- BTT-SxAyy Manufacturer ID
  This is an 2 byte field used to identify the manufacturer of a BLE product, see chapter 4.6. This field is by default set to 0x03DA (EnOcean GmbH).
  Product ID and Manufacturer ID can be configured by the customer as required to identify his BTT-SxAyy based products, see chapter 6.7.4

- BTT-SxAyy Static Source Address
  This is a 4 byte field used to identify the static source address used by BTT-SxAyy, see chapter 4.4.1. Each BTT-SxAyy is pre-programmed with an individual static source address.
  The Static Source Address can be configured by the customer as required to identify his BTT-SxAyy based product, see chapter 6.7.2

- Hardware Revision, Software Revision and NFC
  Revision These fields identify the device revision

- Telegram sequence counter
  This is a 4 byte field which is initialized to 0 during manufacturing and incremented for each transmitted telegram. Receivers shall never accept telegrams containing sequence counter values equal or less than previously received values to avoid replay attacks.

Changing the Static Source Address, Manufacturer ID and Product ID fields is only possible via protected data access as described below to prevent unauthorized modification.

For security reasons, the telegram sequence counter cannot be written or reset by any mechanism.

6.7 Protected Data

The following items are located in the protected data area:

- Source Address Write register
  This 4 byte register is used to update the lower 4 byte of the Static Source Address, see chapter 6.7.2

- Product ID Write register
  This 8 byte register is used to update the Product ID, see chapter 6.7.4

- Manufacturer ID Write register
  This 4 byte register is used to update the Manufacturer ID, see chapter 6.7.4

- Security Key Write register
  This 16 byte register is used to update the security key used by BTT-SxAyy, see chapter 6.7.3

- Optional Data register
  This 4 byte register contains optional data that can be transmitted as part of all data telegrams, see chapter 4.6. Optional Data 0 is sent first, Optional Data 3 last.

- Configuration register
  This 1 byte register is used to configure the functional behavior of BTT-SxAyy, see chapter 6.7.6

- Custom Channel Mode register
  This 1 byte register is used to configure the number of different radio channels used for data and commissioning telegram transmission, see chapter 6.7.7

- Radio Channel Selection registers (CH_REG1, CH_REG2 and CH_REG3)
  These 1 byte registers are used to configure the actual radio channels used whenever the Custom Channel Mode register is set to a user-defined value, see chapter 6.7.8

- Custom NFC Data
  BTT-SxAyy reserves 64 byte for customer-specific NFC data, see chapter 6.7.9

6.7.1 PIN Code

Protected data access is only possible after unlocking the configuration memory with the correct 32 bit PIN code. By default, the protected area is locked and the default pin code for unlocking access is 0x0000 E215.

The default pin code shall be changed to a user-defined value as part of the installation process. This can be done by unlocking the NFC interface with the old PIN code and then writing the new PIN code to page 0xE5 as described in chapter 6.3.1.

6.7.1 Configuration of product parameters

BTT-SxAyy allows no direct modification of the following parameters:

- Static Source Address
- Product ID
- Manufacturer ID
- Security Key

In order to modify these parameters, the user has to write the new value into specific registers (Source Address Write, Product ID Write, Manufacturer ID Write and Security Key Write) in the protected data area and set the according Update flag in the Configuration register.

After that, the user has to push and release one rocker of BTT-SxAyy.

6.7.2 Source Address Write register

The Source Address Write register is 4 bytes wide and can be used to modify the lower 32 bit of the BTT-SxAyy Static Source Address. The upper 16 bit of the BTT-SxAyy Static Source Address are always fixed to 0xE215 to identify the module type (BTT-SxAyy).

In order to do change the lower 32 bit of the Static Source Address, follow these steps:

1. Write new source address into the Source Address Write register
2. Set the Update Source Address flag in the Configuration register to 0b1
3. Actuate (press and release) one rocker of BTT-SxAyy

BTT-SxAyy will determine that it should modify the Static Source Address based on the setting of the Update Source Address flag and copy the value of the Source Address Write register to the lower 32 bit of the Source Address register.

After successful execution, BTT-SxAyy will clear the Update Source Address flag to 0b0.

6.7.3 Security Key Write register

The Security Key Write register is 16 byte wide and contains the device-unique random security key.

The factory programmed key can be replaced with a user defined key by following these steps:

1. Write new security key into the Security Key Write register
   Note that for security reasons, setting the Security Key to the following values is not possible:
   - 0x0000000000000 00000000000 00000000
   - 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

   If the Security Key Write register is set to one of these values then no update of the Security Key will occur.

2. Set the Update Security Key flag in the Configuration register to 0b1

3. If the key should be write-only (not readable after the key update) then set the Private Security Key flag in the Configuration register to 0b1

4. Actuate (press and release) one rocker of BTT-SxAyy

BTT-SxAyy will determine that it should modify the security key based on the setting of the Update Security Key flag and copy the value of the Security Key Write register to the Security Key register in private memory.

After successful execution, BTT-SxAyy will clear the Update Security Key flag to 0b0.

If the Private Key flag in the Configuration register is set to 0b0, then the content of the Security Key Write register will be maintained at its current value. This addresses use cases where the security key shall be readable for users having the correct PIN code.

If the Private Key flag in the Configuration register is set to 0b1, then the content of the Security Key Write register will be cleared to 0x000000000000000 00000000000 000000 after successful execution. This addresses use cases where the security key shall never be readable (even for users having the correct PIN code).

The Security Key Write register will maintain this value of 0x000000000 0000000000 00000000000 00 even if the Private Key flag in the Configuration register is subsequently cleared to 0b0. This ensures that it is not possible to read a security key which was written with the Private Key flag in the Configuration register being set.

Note that it is not possible to read the current security key via NFC if the Security Key Write register has been accidentally overwritten or cleared via NFC write. In this case it is necessary to write a new security key (as described above) or to reset the device to its default security key by means of a factory reset.

The protected memory is designed to support 1,000 modifications of the security key.

6.7.4 Product ID and Manufacturer ID Write register

The Product ID register is 8 bytes wide and can be used to specify a publicly-accessible parameter (e.g. 777 a user-specific ID or name) that can be read by an NFC commissioning tool in order to determine the specific product type.

The Manufacturer ID is 2 bytes wide and specifies the manufacturer of a BLE product and is transmitted as part of each BLE telegram. By default, the manufacturer ID is set to 0x03 DA (EnOcean) and it can be changed to a different OEM identifier.

Product ID and Manufacturer ID can be changed by following these steps:

1. Write the desired Product ID (8 bytes using HEX or ASCII encoding according to user choice) into the Product ID Write register. Setting the Product ID register to 0x0000000000000 000 will cause BTT-SxAyy not to update the Product ID.

2. Write the desired Manufacturer ID (2 bytes) into the Manufacturer ID Write register. Setting the Manufacturer ID Write register to 0x0000 will cause BTT-SxAyy not to update the Manufacturer ID.

3. Set the Update Product and Manufacturer ID flag in the Configuration register to 0b1.

4. Actuate (press and release) one rocker of BTT-SxAyy.

BTT-SxAyy will determine that it should update the Product ID and Manufacturer ID based on the setting of the Update Product and Manufacturer ID flag and copy any non-zero value of the Product ID Write register to the Product ID register and any non-zero value of the Manufacturer ID Write Register to the Manufacturer ID register.

After that, BTT-SxAyy will clear the Update Product and Manufacturer ID flag to 0b0.

6.7.5 Optional Data register

The Optional Data register can be used to specify up to 4 bytes of custom data that will be transmitted as part of each data telegram. This optional data can store user-specific or application-specific information.

The size of the Optional Data field is specified in the Configuration register and can be 0 byte (not present, default), 1 byte, 2 bytes or 4 bytes.

If the size of the Optional Data field is set to a non-zero value in the Configuration register then BTT-SxAyy will read the corresponding amount of data from the Optional Data register beginning with the least significant byte (Byte 0 – Optional Data 0).

Note that using the optional data feature requires additional energy for the radio telegram transmission and might therefore reduce the total number of redundant telegrams which are transmitted.

6.7.6 Configuration register

The Configuration register is 1 byte wide and contains configuration flags. Figure 32 below shows the structure of the Configuration register.

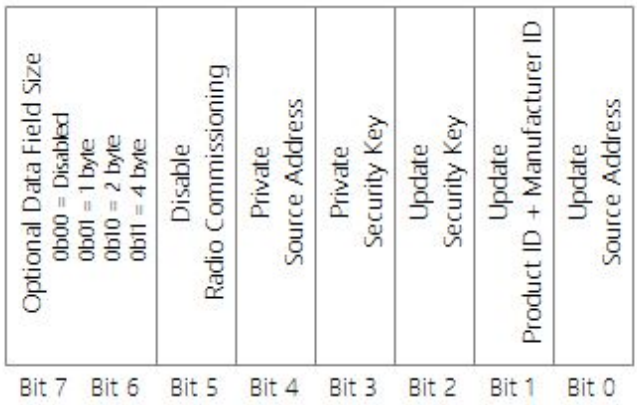


Figure 30 – Configuration register structure

6.7.7 Custom Channel Mode register

The Custom Channel Mode register is 1 byte wide and allows selection of the custom radio transmission modes as described in chapter 3.3.

Table 4 below shows the supported custom radio transmission settings.

Setting	**Meaning**
0x00 (Default)	Commissioning and data telegrams in standard Advertising Mode (Using BLE Advertising Channels CH37, CH38 and CH39) Note: This is equivalent to setting Custom Channel Mode = 0x04 in conjunction with CH_REG1 = 0x25 , CH_REG2 = 0x26 and CH_REG3 = 0x27
0x01	Commissioning telegrams in standard Advertising Mode Data telegrams on 3 user-defined radio channels
0x02	Commissioning telegrams in standard Advertising Mode Data telegrams on 2 user-defined radio channels
0x03	Commissioning telegrams in standard Advertising Mode Data telegrams on 1 user-defined radio channel
0x04	Commissioning and Data telegrams on 3 user-defined radio channels
0x05	Commissioning and Data telegrams on 2 user-defined radio channels
0x06	Commissioning and Data telegrams on 1 user-defined radio channel
0x07 … 0xFF	Unused, will be treated as 0x00

Table 3 – Custom Channel Mode register settings

6.7.8 Radio Channel Selection registers

If the Custom Channel Mode register is set to a value other than 0x00, then the radio channels for transmission are selected using the CH_REG1, CH_REG2 and CH_REG3 registers as described in chapter 3.3. Each of these registers is 1 byte wide and uses the encoding shown in Table 5 below.

Note that two channel types can be used:

- Standard BLE radio channels (BLE Channel 0 … BLE Channel 39 using the even frequencies from 2402 MHz to 2480 MHz as described in chapter 3)
- Custom radio channels in between the standard BLE channels (odd frequencies from 2403 MHz to 2479 MHz)

CH_REGn Value	Frequency	Channel Type
BLE Radio Channels		
37	2402 MHz	BLE Advertising Channel
0	2404 MHz	BLE Data Channel
1	2406 MHz	BLE Data Channel
…		
10	2424 MHz	BLE Data Channel
38	2426 MHz	BLE Advertising Channel
11	2428 MHz	BLE Data Channel
12	2430 MHz	BLE Data Channel
…		
36	2478 MHz	BLE Data Channel
39	2480 MHz	BLE Advertising Channel
Custom Radio Channels		
40	2403 MHz	Custom Radio Channel
41	2405 MHz	Custom Radio Channel
…		
77	2477 MHz	Custom Radio Channel
78	2479 MHz	Custom Radio Channel

Table 4 – Radio Channel Selection register settings

6.7.9 Customer Data

BTT-SxAyy allocates 64 pages (256 bytes) for customer data that can be read and written via the NFC interface in protected mode.

The main intention is to enable storing OEM-specific information such as product type, revision, date code or similar. There is however no restriction (other than the maximum size of 256 byte) on the type of content that can be stored in this memory region.

BTT-SxAyy will not access or modify this memory region.

Users should keep in mind that the content of this memory region will not be affected by a factory reset. This means that after a factory reset, the content of this memory region can be read using the default PIN code. This region should therefore not be used to store sensitive data.

6.8 Private Data

The private data area stores the following items:

- Security Key
- Default settings

The content of the private data area is not externally accessible.

6.8.1 Security Key

The Security Key field contains the 128 bit private key used for authenticating BTT-SxAyy telegrams and for resolving private source addresses.

This register is programmed with a random value during manufacturing. It can be changed using the Security Key Write feature described in chapter 6.7.3.

6.8.2 Default Settings

The Default Settings field contains a backup of the following BTT-SxAyy factory settings:

- Static Source Address
- Security Key
- Manufacturer ID
- NFC PIN Code

These default settings can be restored by means of a factory reset as described in chapter 5.4.

7 Device Label

Each BTT-SxAyy rocker pad contains a device label as shown in Figure 33 below.

Figure 31 – BTT-SxAyy module label

This device label identifies the following parameters in writing:

- Manufacturing date (week 26, 2014 in above example)
- Static Source Address (E2 15 01 50 01 00 in above example)

Note that the device label also contains a DMC code in the lower right corner as described in chapter 5.2.1.

8 APPLICATION INFORMATION

8.1 Transmission range

The main factors that influence the system transmission range are:
- Type and location of the antennas of receiver and transmitter
- Type of terrain and degree of obstruction of the link path
- Sources of interference affecting the receiver
- "Dead spots" caused by signal reflections from nearby conductive objects.

Since the expected transmission range strongly depends on the system conditions, range tests should always be performed to determine the reliably achievable range under the given conditions.

The following figures should be treated as a rough guide only:

- Line-of-sight connections
  Typically 10 m range in corridors, up to 30 m in halls

- Plasterboard walls / dry wood
  Typically 10 m range, through max. 2 walls

- Ferro concrete walls / ceilings
  Typically 5 m range, through max. 1 ceiling (depending on thickness)

- Fire-safety walls, elevator shafts, staircases and similar areas should be considered as shielded

The angle at which the transmitted signal hits the wall is very important. The effective wall thickness – and with it the signal attenuation – varies according to this angle. Signals should be transmitted as directly as possible through the wall. Wall niches should be avoided.

Other factors restricting transmission range include:
- Switch mounting on metal surfaces (up to 30% loss of transmission range)
- Hollow lightweight walls filled with insulating wool on metal foil
- False ceilings with panels of metal or carbon fibre
- Lead glass or glass with metal coating, steel furniture

The distance between the receiver and other transmitting devices such as computers, audio and video equipment that also emit high-frequency signals should be at least 0.5 m.

9 REGULATORY INFORMATION

The BTT-SxAyy device has been certified according to FCC, IC and CE regulations. Changes or modifications not expressly approved by Illumra could void the user's authority to operate the equipment.

9.1 CE / R&TTE for Europe Union

According to lows of the member states of the European Union OEM manufacturer or distributor are responsible for the conformity of the product. In order to support our customers we have done a summary for download at the product web site (Attestation of Conformity).

Note the following requirements for CE certification:

The existing R&TTE directive will be replaced by RE-D (radio equipment directive) valid from 13th of June 2016.

Within the one year transition period (until 12th of June 2017), RE-D and R&TTE are valid for CE declarations. The definition of final test standards for RE-D are expected to be released beginning 2017.

OEM manufacturers or distributors which sell this component as a product to his (final) customers have to fulfill all requirements of the radio equipment directive (RE-D).

RE-D contains at least following requirements for OEM manufacturers or distributors:

- OEM is in charge for product branding / labeling with his name and full postal address
- OEM needs to label product frequency band and max. transmitting power
- OEM has to provide a user manual and security information in the local language
- OEM has to provide its own CE declaration (based on this and additional documents)
- OEM has to provide type, charge or serial number
- OEM has to fulfill all additional requirements according to RE-D such as market surveillance or 10 years record retention.

For details and national translations, please see:
http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32014L0053

9.2 FCC (United States) Certificate

TCB

GRANT OF EQUIPMENT AUTHORIZATION

Certification
Issued Under the Authority of the
Federal Communications Commission
By:

TCB

EMCCert Dr. Rasek GmbH
Stoernhofer Berg 15
91364 Unterleinleiter,
Germany

Date of Grant: 09/26/2016

Application Dated: 09/26/2016

EnOcean GmbH
Kolpingring 18a
Oberhaching, 82041
Germany

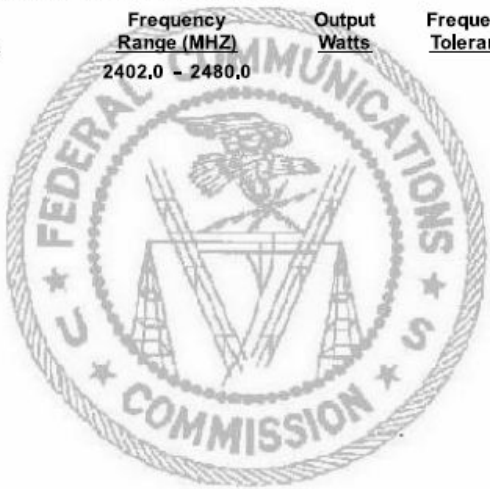
Attention: Armin Anders , Director Product Marketing

NOT TRANSFERABLE

EQUIPMENT AUTHORIZATION is hereby issued to the named GRANTEE, and is VALID ONLY for the equipment identified hereon for use under the Commission's Rules and Regulations listed below.

FCC IDENTIFIER: SZV-PTM215B
Name of Grantee: EnOcean GmbH
Equipment Class: Part 15 Low Power Communication Device Transmitter
Notes: 2402 MHz - 2480 MHz transmitter

Grant Notes	FCC Rule Parts	Frequency Range (MHZ)	Output Watts	Frequency Tolerance	Emission Designator
	15C	2402.0 - 2480.0			



9.2.1 FCC (United States) Regulatory Statement

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) this device may not cause harmful interference, and

(2) this device must accept any interference received, including interference that may cause undesired operation.

9.3 IC (Industry Canada) Certificate

EMCC DR. RAŠEK

FCB under the Canada-EC MRA
TCB under the USA-EC MRA
RFCAB under the Japan-EC MRA
Notified Body R&TTE Directive 99/5/EC
Notified Body RED Directive 2014/53/EU
Notified Body EMC Directive 2014/30/EU
No. CA001711G

TECHNICAL ACCEPTANCE CERTIFICATE CANADA

CERTIFICAT D'ACCEPTABLITÉ TECHNIQUE CANADA

CERTIFICATION No. / No. DE CERTIFICATION ▶ **5713A-PTM215B**

ISSUED TO / DELIVRE A ▶ **EnOcean GmbH**

Street Address / Numéro et rue: Kolpingring 18 a
City / Ville: Oberhaching
Province or State / Province ou Etat: Germany
Postal Code / Code postal: 82041

TYPE OF EQUIPMENT / GENRE DE MATERIEL ▶ Low Power Device (2400–2483.5 MHz)
PMN ▶ PTM 215B
HVIN ▶ PTM 215B
FVIN ▶

ANTENNA / ANTENNE ▶ Integrated / Incorporé
ANTENNA GAIN / GAIN D'ANTENNE ▶

FREQUENCY RANGE / BANDE DE FRÉQUENCES	EMISSION TYPE / GENRE D'ÉMISSION	RF POWER / PUISSANCE H.F.	SPECIFICATION / ISSUE / DATE / SPÉCIFICATION / ÉDITION / DATE
2402 - 2480 MHz	947KG1D	96.6 dBµV/m	RSS-210 / 9 / August 2016

TEST LABORATORY / LABORATOIRE D'ESSAY ▶ EMCCons DR. RASEK GmbH & Co. KG
CN 3464C OATS 3464C-1

Street Address / Numéro et rue: Stoernhofer Berg 15
City / Ville: Unterleinleiter
Province or State / Province ou Etat: Germany
Postal Code / Code Postal: 91364
Name / Nom: Ludwig Kraft
Tel: +49 9194 7263-301
E-mail: l.kraft@emcc.de
Fax: +49 9194 7263-309

Certification of equipment means only that the equipment has met the requirements of the above-noted specification. Licence applications, where applicable to use certified equipment, are acted on accordingly by the ISED issuing office and will depend on the existing radio environment, service and location of operation. This certificate is issued on condition that the holder complies and will continue to comply with the requirements and procedures issued by ISED. The equipment for which this certificate is issued shall not be manufactured, imported, distributed, leased, offered for sale or sold unless the equipment complies with the applicable technical specifications and procedures issued by ISED.

I hereby attest that the subject equipment was tested and found in compliance with the above-noted specification.

La certification du matériel signifie seulement que le matériel a satisfait aux exigences de la norme indiquée ci-dessus. Les demandes de licences nécessaires pour l'utilisation du matériel certifié sont traitées en conséquence par le bureau de délivrance d'ISDE et dépendent des conditions radio ambiantes, du service et de l'emplacement d'exploitation. Le présent certificat est délivré à la condition que le titulaire satisfasse et continue de satisfaire aux exigences et aux procédures d'ISDE. Le matériel à l'égard duquel le présent certificat est délivré ne doit pas être fabriqué, importé, distribué, loué, mis en vente ou vendu à moins d'être conforme aux procédures et aux spécifications techniques applicables publiées par ISDE.

J'atteste par la présente que le matériel a fait l'objet d'essai et jugé conforme à laspécification ci-dessus.

DATE 26 September 2016

Certification Officer

9.3.1 IC (Industry Canada) Regulatory Statement

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions:

(1) this device may not cause interference, and

(2) this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence.
L'exploitation est autorisée aux deux conditions suivantes :

(1) l'appareil ne doit pas produire de brouillage, et

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.”

10 Product History

Table 6 below lists the product history of the BTT-SxAyy module used within BTT-SxAyy.

Revision	Release date	Key changes versus previous revision
DA-3	October 2016	First release for lead customers
DB-4	February 2017	Broad market release incorporating lead customer feedback o Added option to change Manufacturer ID o Added option to use custom radio channels o Changed NFC memory organization to include additional configuration registers o Changed format of commissioning telegram: o Addition of Static Source Address o Removal of Product Name (‘PTM215B')

Table 5 – Product History

A Authentication of BTT-SxAyy data telegrams

BTT-SxAyy provides the option to authenticate its data telegrams as described in chapter 4.8. The authentication mechanism used by BTT-SxAyy is standardized as RFC3610. The full RFC3610 specification could be found here at the time of writing and should be used as a primary source of information: https://www.ietf.org/rfc/rfc3610.txt

The following description aims to summarize the security processing steps for users not deeply familiar with cryptography in general or RFC3610 in particular.

A.1 Algorithm input parameters

The purpose of the security processing in BTT-SxAyy is to calculate a unique signature that can be used to verify authenticity (telegram has not been modified) and originality (telegram comes from the assumed sender) of a telegram.

To do so, two types of algorithm parameters are required:

- Constant algorithm input parameters
  These parameters identify high level algorithm and telegram properties and are the same for any BTT-SxAyy telegram

- Variable algorithm input parameters
  These parameters identify telegram-specific parameters and therefore depend on the specifics of the transmitted telegram

A.1.1 Constant input parameters

The RFC3610 implementation in BTT-SxAyy requires two constant input parameters:

- Length field size
  This is the size (in bytes) of the field used to encode the length of the input data (which is the payload to be authenticated).
  The maximum size of BTT-SxAyy payload to be authenticated is 13 bytes; therefore one byte would be easily sufficient to encode the payload size. The minimum value permitted by the standard is however 2 bytes which is therefore chosen.

- Signature size
  This is the desired size of the generated signature which is 4 bytes for

BTT-SxAyy Table 7 below summarizes these constant algorithm parameters.

Parameter	Comment / Description	Example
Length Field Size	Size (in bytes) of the field used to encode the input length	2 (always, minimum permissible size)
Signature Size	Desired size (in bytes) of the signature generated by the algorithm	4 (always)

Table 6 – Constant algorithm input parameters

A.1.2 Variable input parameters

The RFC3610 implementation in BTT-SxAyy requires four variable input parameters:

- Source address
  The 6 byte source address used to identify the sender of an authenticated message. The source address is required in little endian (least significant byte first) format.

- Input data (Payload to be authenticated)
  The authenticated payload contains source address, sequence counter, switch status and optional data (if present). See chapter 4.8 for a description of the authenticated payload.

- Input length (Size of the payload to be authenticated)
  The length of the payload to be authenticated depends on the amount of optional data used in the telegram. This is configured via the Configuration register, see chapter 6.7.6.
  By default, no optional data is present and the length of the authenticated payload is 9 bytes.

- Sequence counter
  Each BTT-SxAyy contains a sequence counter which is initialized to zero during production and increased for each telegram that is sent.
  The sequence counter is transmitted as part of the input data.
  The receiver of BTT-SxAyy telegrams keeps track of this counter and will accept only telegrams with counter values higher than the highest previously used value. This eliminates the possibility of reusing previously transmitted telegrams.
  Note that the individual (identical) advertising telegrams used to encode the same data telegram use the same sequence counter value.

- Security key
  Each BTT-SxAyy is programmed with a random 16 byte security key during manufacturing. This key can be modified using the NFC interface, see chapter 6.7.3.

Table 8 below summarizes these parameters.

Parameter	Comment / Description	Example
Source Address	Unique source address of the BTT-SxAyy module (little endian)	B819000015 E2 (little endian representation of E21500001 9B8)
Input Data	Telegram data to be authenticated	0CFFDA03D00A000003
Input Length	Length of input data (in bytes, encoded using 2 bytes)	0x0009 (if optional data size = 0, default) 0x000 A (if optional data size = 1) 0x000 B (if optional data size = 2) 0x000 D (if optional data size = 4)
Sequence Counter	Incrementing counter to avoid replay Part of the input data (byte 4 … 7)	D00A0000 (little endian representation of the counter value 00000 AD0)
Security Key	128 bit random key that is known both to sender and receiver	3DDA31AD44767 AE3CE56DCE2B3CE2ABB

Table 7 – Variable input parameters

A.1.3 Obtaining the security key

All required parameters except the security key can be directly extracted from the received message that shall be authenticated.

The security key – the common secret shared between sender and receiver – has to be obtained via specific mechanisms. As described in chapter 5, there are three different ways to obtain the security key of a given BTT-SxAyy module:

- Obtaining the key via the NFC configuration interface
- Obtaining the key via the product DMC code
- Obtaining the key via a dedicated commissioning telegram. Each option is described now in detail.

A.1.3.1 Obtaining the security key via NFC interface

Using the Elatec TWN4 reader (as described in chapter 6.3), the security key can be read using the following command sequence:

Search Tag (32)
NTAG_PwdAuth (0x00 0x00 0xE2 0x15 ,
0x00 0x00) NTAG_Read (0x14)

This is equivalent to the following binary command sequence:

Request: 050020
Response : 000180380704883 1A2014 F8020060000 E2150000

Request: 20060000 E215000 0
Response : 0001

Request: 20001 4
Response : 00013 DDA31AD44767AE3CE56DCE2B3CE2ABB

The tag response to the last command - NTAG_Read (0x14) - contains the password:

NTAG_Read (0x14) Result: true Page: 3DDA31AD44767 AE3CE56DCE2B3CE2ABB

The password of this device is therefore: 3DDA31AD44767 AE3CE56DCE2B3CE2ABB

A.1.3.2 Obtaining the security key via the product DMC code

Each BTT-SxAyy module contains a DMC code on its product label which identifies source address and security key of the module, see chapter 5.2.

The DMC code of the device used for this tutorial is shown in Figure 34 below.

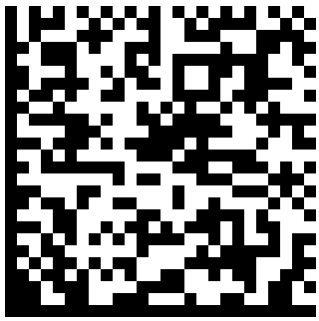


Figure 32 – Example DMC code

This DMC code can be read using a suitable DMC code reader (e.g. QRbot smartphones). The content of this example DMC code is:

PTM215B I DE21500 0019 B8OOB3DDA31AD4476 7AE3CE56DCE2B3CE2ABB

The structure of the DMC code is described in chapter 5.2.1. The location of the security key in above DMC string is marked red for reference. This means that the security key of this device is:

3DDA31AD44767 AE3CE56DCE2B3CE2ABB

A.1.3.3 Obtaining the security key via a commissioning telegram

BTT-SxAyy modules can send dedicated commissioning telegrams that identify their security key. Transmission of such commissioning telegrams can be triggered by means of a specific button sequence as described in chapter 5.3.

Note that this feature can be disabled via the NFC commissioning interface by setting the Disable Radio Commissioning flag in the Configuration register to 0b1 (see chapter 6.7.6). The resulting commissioning telegram has the following payload:

1D FF DA 03 56 04 0000 3D DA 31 AD 44 76 7A E3 CE 56 DC E2 B3 CE 2A B8BB 19 00 00
15 E2

Please see Figure 22 in chapter 5.3.2 for a description of the commission telegram structure. The location of the security key is for reference highlighted red above. This means that the security key of this device is:

3DDA31AD44767 AE3CE56DCE2B3CE2ABB

A.2 Internal parameters

The RFC3610 implementation in BTT-SxAyy derives a set of internal parameters for further processing from the provided input parameters.

Again, there are two types of internal parameters:

- Constant internal parameters
  These parameters are based on the high level algorithm and telegram properties and are the same for any BTT-SxAyy telegram

- Variable input parameters
  These parameters are based on the telegram-specific parameters and therefore depend on the specifics of the transmitted telegram

A.3 Constant internal parameters

The RFC3610 implementation in BTT-SxAyy derives two internal parameters – M' and L' – based on the input data and uses them to construct A0_Flag and B0_Flag which – together with the iteration counter i – are required for subsequent processing.

The value of these internal parameters - listed in Table 9 below - is the same for all BTT-SxAyy telegrams.

Parameter	Comment / Description	Example
M'	Binary encoded output length M ' = (Out put length / 2) - 1	0b001 (always)
L'	Binary encoded length field size L' = length field size - 1	0b001 (always)
A0_Flag	L'	0x01 (always)
B0_Flag	(0b01 <<6) + (M '<<3) + L'	0x49 (always)
i	Iteration counter	0x0000 (always)

Table 8 – Constant internal parameters

A.4 Variable internal parameters

The RFC3610 implementation in BTT-SxAyy derives four internal parameters – Nonce, A0, B0 and B1 – based on the telegram specific input data and the constant internal parameters.

These variable internal parameters - listed in Table 10 below - are then used together with the security key to calculate the actual signature.

Parameter	Comment / Description	Example
Nonce	13 byte initialization vector based on concatenation of source address, sequence counter and padding, see 4.8.1	FE19000015 E2D00A0000000000
A0	A0_Flag followed by Nonce followed by 2 bytes 0x00	01FE19000015 E2D00A0000000000000
B0	B0_Flag followed by Nonce followed by 2 bytes 0x00 (no message to encode)	49FE19000015 E2D00A0000000000000
B1	Input Length followed by Input Data followed by 5 / 4 / 3 / 1 byte of 0x00 padding (for optional data size = 0 / 1 / 2 / 4 byte)	00090 CFFDA03D00A0000030000000000

Table 9 – Variable internal parameters

A.5 Algorithm execution sequence

The algorithm uses the variable internal parameters A_0, B_0, B_1 together with the private key to generate the authentication vector T_0 using three AES-128 and two XOR operations. The algorithm execution sequence is shown in Figure 35 below.

The first four bytes of T_0 are then used to authenticate BTT-SxAyy telegrams.

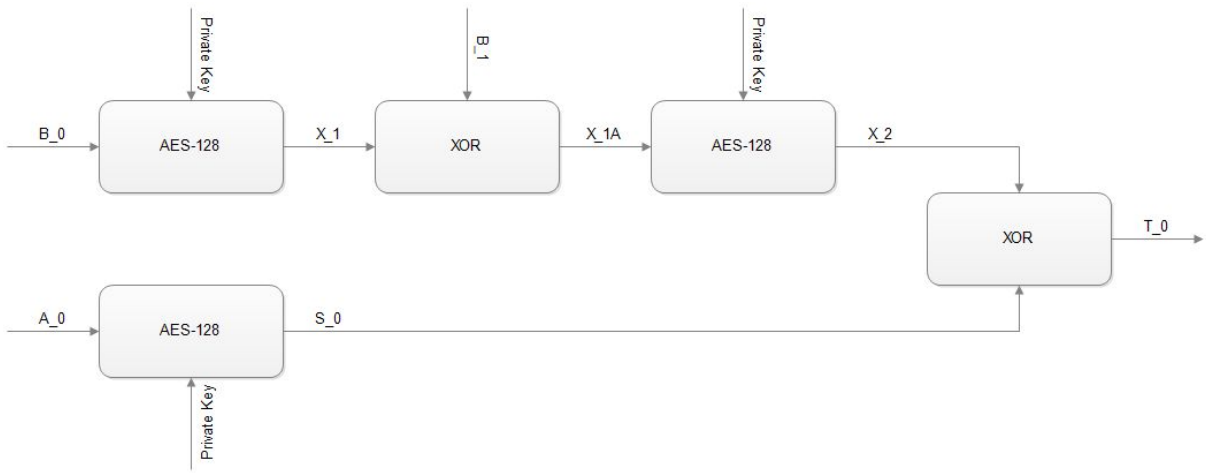


Figure 33 – Authentication algorithm sequence

A.6 Examples

The following four chapters give step by step examples based on one actual device and 0 / 1 / 2 or 4 bytes of optional data.

A.6.1 Data telegram without optional data

For this example, we consider the following telegram payload received from a BTT-SxAyy with the source address E21500 0019 B8 and security key 3DDA31AD44767 AE3CE56DCE2B3CE2ABB:

0C FF DA 03 5D 04 00 00 11 B2 FA 88 FF

The last four bytes of this payload (B2 FA 88 FF) are the sender provided signature which has to be authenticated (compared against the signature, the receiver calculates based on its own security key).

The variable input parameters are therefore the following:

Parameter	**In this example**
Source Address	B819000015 E2 (little endian representation of E215000019 B8)
Input Data	0CFFDA035D04000 011
Input Length	0x0009
Sequence Counter	5D040000
Security Key	3DDA31AD44767 AE3CE56DCE2B3CE2ABB

The constant internal parameters are always the same:

Parameter	**In this example**
A0_Flag	0x01 (always)
B0_Flag	0x49 (always)
i	0x0000 (always)

Based on variable input data and constant internal algorithm parameters, we can now derive the following variable internal parameters:

Parameter	**In this example**
Nonce	B819000015 E25D040000 000000
A0	01B819000015 E25D040000 0000000000
B0	49B819000015 E25D040000 0000000000
B1	00090 CFFDA035D04000011 0000000000

We can now calculate the signature using AES128 and XOR operations.

At the time of writing, a suitable online AES calculator could be found here: http://testprotect.com/appendix/AEScalc

Likewise, a suitable XOR calculator could be found here: http://xor.pw/?

X_1 = AES128 (B0 , Key)
X_1 = AES128 (49B819000015 E25D040000 000000000 0 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
X_1 = 41 e f 09792 ae 152 ae 52 c671435 c1 f 247d

X_1A = XOR (X_1 , B_1)
X_1A = XOR (41 e f 09792 ae 152 ae52 c671435 c1 f 247d , 00090 CFFDA035D04000011 0000000000)
X_1A = 41 e60586 f 0e20 f aa 52 c660435 c1 f 247d

X_2 = AES128 (X1A , Key)
X_2 = AES128 (41e60586 f 0e20 f aa 52 c660435 c1 f 247d , 3DDA31AD447 67AE3CE56DCE2B3CE2ABB)
X_2 = 8d89 e733d a516 ae 3e08 f 9e30184909 f c

S_0 = AES128 (A0 , Key)
S_0 = AES128 (01B819000015 E25D040000 000000000 0 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
S_0 = 3 f 736 f cc 8bca f 2d4 aa bca0260 f ab7976

T_0 = XOR (X_2 , S_0)
T_0 = XOR (8d89 e733d a516 ae 3e08 f 9e30184909 f c , 3 f 736 f cc 8bca f 2d4aa bca 0260 f ab7976) T_0 = b2 f a88 ff 519b98374 a333e1617 e2708a

The calculated signature is formed by the first four bytes of T_0, i.e. it is B2 FA 88 FF. The calculated signature matches the signature that was transmitted as part of the payload. This proves that the telegram originates from a sender that possesses the same security key and the telegram content has not been modified.

A.6.2 Data telegram without 1 byte optional data

For this example, we consider the following telegram payload received from a BTT-SxAyy with the source address E21500 0019 B8 and security key 3DDA31AD44767 AE3CE56DCE2B3CE2ABB:

0D FF DA 03 62 04 00 00 10 12 B9 FE AC C1

The last four bytes of this payload (B9 FE AC C1) are the sender provided signature which has to be authenticated. The variable input parameters are therefore the following:

Parameter	**In this example**
Source Address	B819000015 E2 (little endian representation of E215000019 B8)
Input Data	0DFFDA036204000 01012
Input Length	0x000 A
Sequence Counter	62040000
Security Key	3DDA31AD44767 AE3CE56DCE2B3CE2ABB

Based on variable input data and constant internal algorithm parameters, we can now derive the following variable internal parameters:

Parameter	**In this example**
Nonce	B819000015 E262040000 000000
A0	01B819000015 E262040000 0000000000
B0	49B819000015 E262040000 0000000000
B1	000A0DFFDA03620 400001012 00000000

We can now calculate the signature as follows:

X_1 = AES128 (B0 , Key)
X_1 = AES128 (49B819000015 E262040000 000000000 0 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
X_1 = dc8d685 f 968 e795b23 f 4370b3091 f 33 f

X_1A = XOR (X_1 , B_1)
X_1A = XOR (dc8d685 f 968 e795b23 f 4370b3 091 f 33 f , 000A0DFFDA0362 0400001012 00000000) X_1A = dc8765 a04c8d1b5 f 23 f 427193091 f 33 f

X_2 = AES128 (X1A , Key)
X_2 = AES128 (dc8765 a04 c8d1b5 f 23 f 427193091 f 33 f , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
X_2 = 231b e2 ff 54ca 62 f b38d3 2eaaa f 1b447d

S_0 = AES128 (A0 , Key)
S_0 = AES128 (01B819000015 E262040000 000000000 0 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
S_0 = 9ae 54 e3e95de9 f 91 a0c279537b c25b00

T_0 = XOR (X_2 , S_0)
T_0 = XOR (231b e2 ff 54 ca 62 f b38d32 eaaa f 1b447d , 9ae 54 e3e95d e9 f 91a0c279537b c25b00) T_0 = b9 f eacc 1c114 f d6 a981157 f 9d4d91 f

7d

The calculated signature is formed by the first four bytes of T_0, i.e. it is B9 FE AC C1.

A.6.3 Data telegram without 2 byte optional data

For this example, we consider the following telegram payload received from a BTT-SxAyy with the source address E21500 0019 B8 and security key 3DDA31AD44767 AE3CE56DCE2B3CE2ABB:

0E FF DA 03 63 04 00 00 11 12 34 52 E0 51 16

The last four bytes of this payload (52 E0 51 16) are the sender-provided signature which has to be authenticated. The variable input parameters are therefore the following:

Parameter	**In this example**
Source Address	B819000015 E2 (little endian representation of E215000019 B8)
Input Data	0EFFDA036304000 0111234
Input Length	0x000 B
Sequence Counter	62040000
Security Key	3DDA31AD44767 AE3CE56DCE2B3CE2ABB

Based on variable input data and constant internal algorithm parameters, we can now derive the following variable internal parameters:

Parameter	**In this example**
Nonce	B819000015 E263040000 000000
A0	01B819000015 E263040000 0000000000
B0	49B819000015 E263040000 0000000000
B1	000B0EFFDA03630 40000111234 000000

We can now calculate the signature as follows:

X_1 = AES128 (B0 , Key)
X_1 = AES128 (49B819000015 E263040000 000000000 0 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
X_1 = ab5 ec 24b eabc9dd ee b73751 c7734 cc 64

X_1A = XOR (X_1 , B_1)
X_1A = XOR (ab5 ec24b ea bc9ddee b73751 c7734 cc 64 , 000B0EFFDA0363 040000 11123 4000000)
X_1A = ab55 cc b430b ff edae b73640 e4334 cc64

X_2 = AES128 (X1A , Key)
X_2 = AES128 (ab55 cc b430b ff edae b73640 e4334 cc 64 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
X_2 = d33 e96d7 a105 c4e85432 07 f 9e75 e6c f e

S_0 = AES128 (A0 , Key)
S_0 = AES128 (01B819000015 E263040000 000000000 0 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)

S_0 = 81d ec 7c16915 c6647d92 b0668 f 65 e9c9

T_0 = XOR (X_2 , S_0)
T_0 = XOR (d33 e96d7 a105 c4e8543207 f 9e75e6c f e , 81d ec 7c16915 c6647d92b0668 f 65 e9c9) T_0 = 52 e05116 c810028 c29 a0b79 f 683b85 37

The calculated signature is formed by the first four bytes of T_0, i.e. it is 52 E5 11 16.

A.6.4 Data telegram without 4 byte optional data

For this example, we consider the following telegram payload received from a BTT-SxAyy with the source address E21500 0019B8 and security key 3DDA31AD44767 AE3CE56DCE2B3CE2ABB:

10 FF DA 03 6A 04 00 00 10 12 34 56
78 2C 9E 10 95

The last four bytes of this payload (2C 9E 10 95) are the sender provided signature which has to be authenticated. The variable input parameters are therefore the following:

Parameter	**In this example**
Source Address	B819000015 E2 (little endian representation of E215000019 B8)
Input Data	10FFDA036A04000 01012345678
Input Length	0x000 D
Sequence Counter	6A040000
Security Key	3DDA31AD44767 AE3CE56DCE2B3CE2ABB

Based on variable input data and constant internal algorithm parameters, we can now derive the following variable internal parameters:

Parameter	**In this example**
Nonce	B819000015 E26A040000 000000
A0	01B819000015 E26A040000 0000000000
B0	49B819000015 E26A040000 0000000000
B1	000D10FFDA036A040000101234 567800

We can now calculate the signature as follows:

X_1 = AES128 (B0 , Key)
X_1 = AES128 (49B819000015 E26A040000 000000000 0 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
X_1 = 434 f a5855 b8 a8a8ae 99b f 1cb114 a51b7

X_1A = XOR (X_1 , B_1)
X_1A = XOR (434 f a5855b8 a8a8ae 99b f 1cb114 a51b7 , 000D10FFDA036A04000010123 45678 00) X_1A = 4342b57 a8189 e08 ee 99be1d9251 c29b7

X_2 = AES128 (X1A , Key)
X_2 = AES128 (4344b57 a8189 e08 ee 99b e1d9251 c29b7 , 3DDA31AD447 67AE3CE56DCE2B3CE2ABB)
X_2 = 12 c78b85 a4ec b6 f 34d a f f 7651db8 e386

S_0 = AES128 (A0 , Key)
S_0 = AES128 (01B819000015 E263040000 000000000 0 , 3DDA31AD4476 7AE3CE56DCE2B3CE2ABB)
S_0 = 3e599b103 f 33447 e6b46 eec 4a042d0 bc

T_0 = XOR (X_2 , S_0)

T_0 = XOR (12 c78b85 a4ec b6 f 34da f f 7651d b8 e386 , 3e599b103 f 33447 e6b46 eec 4a042d0b c) T_0 = 2c9e10959 bd ff 28d26 e919 a1bd f a333a

The calculated signature is formed by the first four bytes of T_0, i.e. it is 2C 9E 10 95.