

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

STM 550B / EMSIB

EASYFIT Multisensor For IoT Applications (2.4 GHz BLE)



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

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

REVISION HISTORY

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

Version	Author	Reviewer	Date	Major Changes
1.0	MKA	RS, TM	14.01.2020	First public release

**Published by EnOcean GmbH, Kolpingring 18a, 82041 Oberhaching, Germany
www.enocean.com, info@enocean.com, phone +49 (89) 6734 6890**

© EnOcean GmbH, All Rights Reserved

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 EnOcean website: <http://www.enocean.com>. As far as patents or other rights of third parties are concerned, liability is only assumed for modules, not for the described applications, processes and circuits.

EnOcean does not assume responsibility for use of modules described and limits its liability to the replacement of modules determined to be defective due to workmanship. Devices or systems containing RF components must meet the essential requirements of the local legal authorities. The modules 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 modules 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.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

TABLE OF CONTENT

1	General description.....	7
1.1	Basic functionality	7
1.2	Product variants	8
1.3	Technical data.....	8
1.4	Environmental conditions	9
1.5	Packaging information.....	9
1.5.1	STM 550B.....	9
1.5.2	EMSIB.....	9
1.6	Ordering information	9
2	Functional overview	10
2.1	Product description.....	10
2.2	Product interface.....	10
2.2.1	LRN button and LED.....	11
2.2.2	Solar cell	12
2.2.3	Backup battery.....	12
2.2.4	Product label	12
2.3	Functional modes	13
2.3.1	Standard operation mode	13
2.3.2	Standby (Sleep) mode.....	14
2.3.3	Learn mode.....	14
2.3.4	Function test mode	14
2.3.5	Illumination test mode	15
2.3.6	Factory reset mode	15
2.4	Reporting interval.....	16
2.4.1	Energy considerations	16
2.4.2	Standard reporting interval	17
2.4.3	Illumination-controlled reporting interval	18
2.4.4	Temperature-controlled reporting interval.....	19
2.4.5	Humidity-controlled reporting interval	20
2.4.6	Acceleration-controlled reporting interval.....	21
2.4.7	Magnet contact sensor-controlled reporting interval	22
2.4.8	Arbitration between reporting intervals.....	22
3	Sensor functionality	23
3.1	Ambient light sensor.....	23
3.2	Solar cell.....	23
3.3	Temperature sensor.....	24
3.4	Humidity sensor	24
3.5	Acceleration sensor.....	25
3.6	Magnet contact sensor	26
4	Radio communication.....	27
4.1	Radio channels.....	27
4.2	Default radio transmission sequence.....	28
4.3	User-defined radio transmission sequences.....	28
4.3.1	User-defined radio transmission sequences	29

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

4.3.2	Three-channel sequence	29
4.3.3	Two-channel sequence	30
4.3.4	One channel sequence.....	30
4.4	Telegram format	31
4.4.1	Preamble	32
4.4.2	Access Address.....	32
4.4.3	Header	32
4.4.4	Source address.....	32
4.4.5	Static source address mode	33
4.4.6	Resolvable Private Address mode.....	34
4.4.7	Check Sum	35
4.5	Payload structure	36
4.5.1	Sensor status encoding	37
4.5.2	Sensor Data Descriptor.....	37
4.5.3	Data Size.....	38
4.5.4	Supported parameters.....	38
4.5.5	Acceleration vector encoding	39
5	Security	40
5.1	Security implementation	41
6	Commissioning.....	42
6.1	Radio-based commissioning	43
6.2	QR code-based commissioning	43
6.3	Commissioning via NFC interface.....	43
7	NFC interface	44
7.1	NFC interface parameters	44
7.2	NFC access protection	44
7.3	Using the NFC interface.....	45
7.4	NFC interface functions	46
7.4.1	NFC interface state machine.....	46
7.4.2	IDLE state.....	47
7.4.3	READY 1 state	47
7.4.4	READY 2 state	47
7.4.5	ACTIVE state.....	47
7.4.6	Read command	48
7.4.7	Write command	48
7.4.8	Password authentication (PWD_AUTH) command	49
8	NFC registers	50
8.1	NFC memory areas	50
8.2	NDEF	51
8.3	SW_VERSION	51
8.4	NFC HEADER.....	52
8.4.1	NFC HEADER area structure	52
8.5	CONFIGURATION.....	53
8.5.1	Using the NFC configuration functionality	53
8.5.2	CONFIGURATION area structure	53
8.5.3	NFC_PIN_CODE	54

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

8.5.4	SOURCE_ADDRESS	54
8.5.5	CH_REG1, CH_REG2, CH_REG3	54
8.5.6	TX_CFG	54
8.5.7	TX_POWER	56
8.5.8	ADV_INTERVAL	56
8.5.9	MANUFACTURER_ID	57
8.5.10	OPTIONAL_DATA	57
8.5.11	OPTIONAL_DATA_SIZE	57
8.5.12	SECURITY_KEY	58
8.5.13	SECURITY_KEY_ACCESS	58
8.5.14	SECURITY_CFG	59
8.5.15	REPORTING_CFG	60
8.5.16	LED_MODE	62
8.5.17	FUNCTIONAL_MODE	63
8.5.18	STANDARD_TX_INTERVAL	64
8.5.19	THRESHOLD_CFG1	65
8.5.20	THRESHOLD_CFG2	67
8.5.21	ACC_SENSOR_CFG	68
8.5.22	SOLAR_THRESHOLD	69
8.5.23	SOLAR_TX_INTERVAL	70
8.5.24	LIGHT_THRESHOLD	71
8.5.25	LIGHT_TX_INTERVAL	72
8.5.26	ACCELERATION_THRESHOLD	73
8.5.27	ACCELERATION_TX_INTERVAL	74
8.5.28	TEMPERATURE_THRESHOLD	75
8.5.29	TEMPERATURE_TX_INTERVAL	76
8.5.30	HUMIDITY_THRESHOLD	77
8.5.31	HUMIDITY_TX_INTERVAL	78
8.5.32	MAGNET_CONTACT_TX_INTERVAL	79
8.5.33	ILLUMINATION_TEST_RESULT	80
8.6	USER DATA	80
9	Installation recommendations	81
9.1	Setup instructions	81
9.2	Acceleration detection	82
9.3	Temperature and humidity measurement	83
9.4	Illumination measurement	84
9.4.1	Ambient light sensor	84
9.4.2	Solar cell	84
9.5	Magnet contact sensing	85
9.6	Energy harvesting	85
9.7	NFC configuration	85
10	Product history	86
A	Telegram parsing	87
A.1	Data telegram example	87
A.1.1	BLE advertising frame structure	87
A.1.2	Data telegram payload	87
A.1.3	Sensor data	88
A.2	Commissioning telegram example	89
A.2.1	BLE advertising frame	89

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

A.2.2	Commissioning telegram payload	89
B	Authentication	90
B.1	Input data	90
B.2	Constant algorithm parameters	91
B.3	Intermediate parameters.....	92
B.4	RFC3610 execution sequence.....	93
B.5	Execution example	94
C	Address resolution for resolvable private addresses (RPA)	95
C.1	RPA resolution flow.....	95
C.2	Address resolution example	96

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

1 General description

This user manual describes the functionality of the EnOcean multisensor module STM 550B and the Easyfit Multisensor for IoT applications with Bluetooth radio (EMSIB) which is based on this module. EMSIB integrates the STM 550x module with a wall mount and a cover frame into a finished product.

The term "STM 550B" is used throughout this user manual to describe both EMSIB and STM 550B unless otherwise noted.

1.1 Basic functionality

STM 550B is a flexible self-powered multi-sensor module capable of measuring temperature, humidity, illumination, magnet contact status and acceleration. It enables the realization of energy harvesting wireless sensors for light, building or industrial control systems communicating using the 2.4 GHz Bluetooth Low Energy (BLE) communication standard.

STM 550B uses the same mechanical form factor as the industry standard PTM 21x modules from EnOcean.

STM 550B integrates the following dedicated sensors:

- Temperature
- Humidity
- Ambient light level
- Acceleration
- Magnet contact

STM 550B will report regularly (by default approximately every 60 seconds, configurable via NFC) the latest measurements of these sensors. In addition, STM 550B can report its internal energy level, the amount of light available at the solar cell and the backup battery voltage (if a backup battery is present).

STM 550B will report immediately if the status (open / closed) of the magnet contact changes or if an acceleration change detected by the acceleration sensor exceeds a user-defined threshold for the first time.

Radio telegrams transmitted by STM 550B are authenticated AES-128 security based on a device-unique private key and a sequence counter. This ensures integrity and authenticity of the transmitted telegrams and prevents telegram replay (retransmission of previously transmitted telegrams).

STM 550B is self-supplied via an integrated solar cell which generates the energy required for its operation. For cases where ambient light is not sufficiently available, STM 550B provides the option to mount a CR1632 backup battery.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

1.2 Product variants

EMSIB is a finished product combining STM 550B with a wall mount and a design frame. STM 550B is the functional module within EMSIB using the PTM 21x form factor intended to be integrated into an OEM housing (e.g. a switch frame).

STM 550B and EMSIB are shown in Figure 1 below.



Figure 1 – STM 550B (left) and EMSIB (right)

1.3 Technical data

Antenna	Integrated whip antenna
Transmission power (at antenna base)	+4 dBm
Communication range (guidance only)	30 m line of sight / 10 m indoor environment
Communication standard	BLE Advertising
Radio frequency (min / max)	2402 / 2480 MHz
Radio channels (default)	BLE CH 37 / 38 / 39 (2402 / 2426 / 2480 MHz)
Data rate and modulation	1 Mbit/s GFSK (default setting) 2 Mbit/s GFSK (optional setting via NFC)
Temperature measurement range / accuracy ⁽¹⁾	-25 °C ... +65 °C / +- 0.3 K
Humidity measurement range / accuracy	0 ... 100 % r.h. / +- 3 % r.h.
Illumination measurement range / accuracy	0 ... 65000 Lux / +-10 %
Acceleration measurement range / resolution	+ - 4 g / 0.002 g
Acceleration threshold for immediate report	0.0625 g (default, configurable via NFC)
Update rate (excl. random offset)	Every 60 seconds (configurable via NFC)
Device configuration	LRN button and NFC interface
Device identification	Unique 48 Bit Device ID (factory programmed)
Security	AES128 (CBC mode) with sequence counter
Power supply	Integrated solar cell
Minimum light level (with default settings)	200 lux for 6 hours per day
Operating time in darkness (at 25°C)	4 days (after full charge with default settings)
Backup battery type (optional)	CR1632
Dimensions (STM 550B, Module only)	40 mm x 40 mm x 13 mm
Dimensions (EMSIB, Finished product)	49 mm x 49 mm x 13 mm

Note 1: STM 550B / EMSIB is designed for indoor use only and should only be used in the environmental conditions specified below

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

1.4 Environmental conditions

Operating Temperature	-5 °C ... +45 °C (indoor use in dry rooms only)
Humidity	0% to 90% r.h. (non-condensing)

1.5 Packaging information

1.5.1 STM 550B

Packaging Unit	100 units
Packaging Method	10 modules per tray, 10 trays per box

1.5.2 EMSIB

Packaging Unit	Individual unit packaging
Packaging Method	TBD packages within one box

1.6 Ordering information

Product	Type	Ordering Code	Frequency
STM 550B	Module (PTM form factor)	S6221-K516	2.4 GHz (BLE)
EMSIB	WLP (with frame and wall mount)	E6221-K516	2.4 GHz (BLE)

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2 Functional overview

2.1 Product description

The energy-harvesting multi-sensor module STM 550B from EnOcean provides wireless sensing functionality without batteries. Power is provided by an integrated solar cell. STM 550B transmits sensor data using the 2.4 GHz Bluetooth Low Energy standard.

STM 550B operates fully self-powered (no batteries required) when sufficient available ambient light (200 Lux for 6 hours per day) is available. In this configuration it is fully maintenance-free. For cases where sufficient ambient light is not available there is the option to mount a CR1632 backup battery.

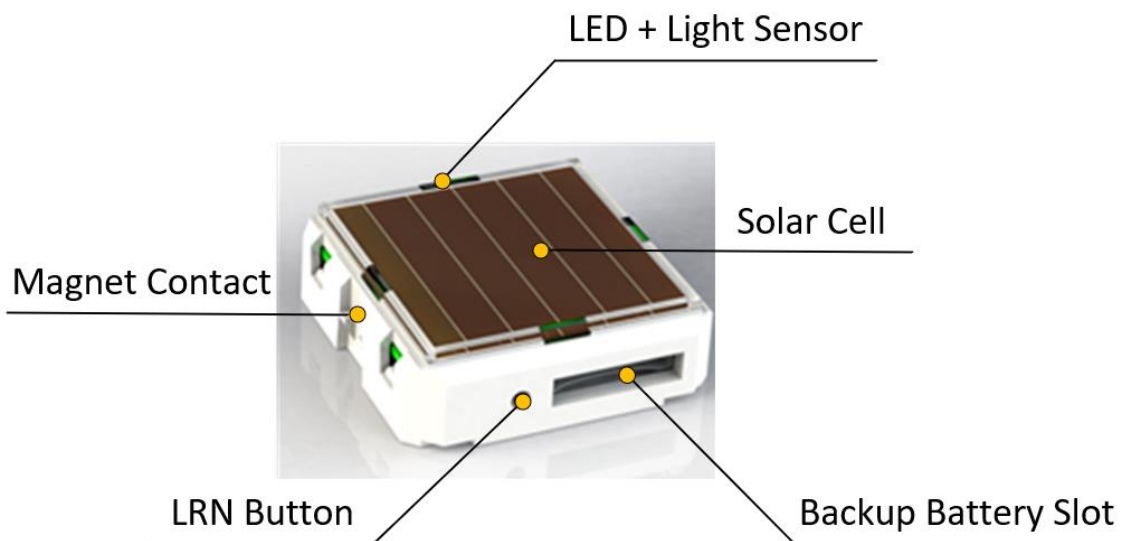
Radio telegrams transmitted by STM 550B are authenticated AES-128 security based on a device-unique private key and a sequence counter. This ensures integrity and authenticity of the transmitted telegrams and prevents telegram replay (retransmission of previously transmitted telegrams).

2.2 Product interface

The STM 550B product interface consists of the following elements:

- LRN button and LED as described in chapter 2.2.1
- Solar cell harvesting energy from ambient light as described in chapter 2.2.2
- Ambient light sensor (co-located with the LED) as described in chapter 3.
- Magnet contact sensor as described in chapter 3.6
- Backup battery slot for a CR1632 battery as described in chapter 2.2.3
- Backup battery ejector slot (on the back side) as described in chapter 2.2.3
- Product label (on the back side) as described in chapter 2.2.4

Figure 2 below shows the external interfaces on the front side of the STM 550B module.



STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

Figure 2 – STM 550B product interface (front side)

Figure 3 below shows the external interfaces on the back side of the STM 550B module within EMSIB. The orientation indicator points towards the side where the magnet contact is located.

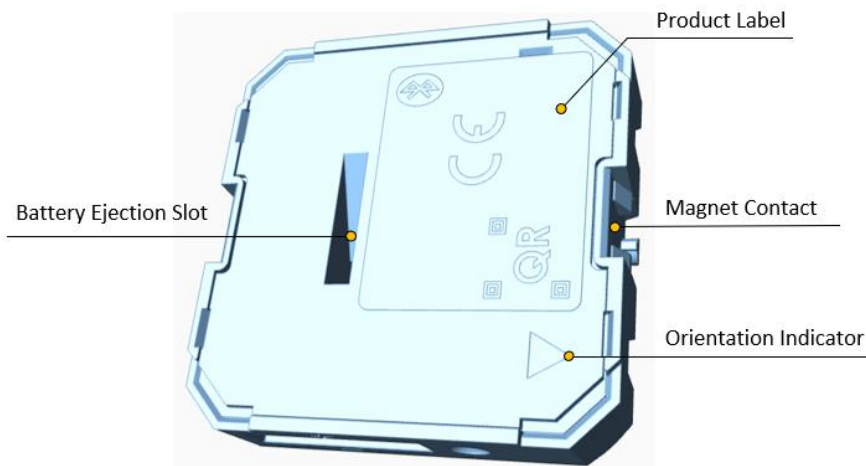


Figure 3 – STM 550B product interface (back side)

2.2.1 LRN button and LED

Most STM 550B device parameters can be configured using the NFC interface as described in chapter 7 and 8. Some of the most common parameters or states can additionally be configured using the LRN button.

Table 1 below lists those LRN button actions with the corresponding STM 550B response and LED feedback.

Type	Timing	STM 550B Response	LED Feedback	
Single Short	< 1s Press	Exit from Sleep Mode Send Learn Telegram	1 short blink	
Double Short	< 1s Press, < 1s Release, < 1s Press	Start Function Test (End after 2 min or upon any button press)	Transmit sensor status with highest possible update rate	
	< 1s Press, < 1s Release, < 1s Press, < 1s Release, < 1s Press			LED enabled: 2 short blinks
				LED disabled: No feedback
Long	3s < Press < 5s	Enter Sleep Mode (Disable LED and Radio)	Error: No feedback	
			Success: 3 short blinks	
Very Long	> 8s Press	Factory Reset	Error: No feedback	
			Success: 5 short blinks	

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

Table 1 – STM 550B user interface actions

2.2.2 Solar cell

The solar cell harvests energy from ambient light. The spectral response of the solar cell is optimized to harvest energy both from natural and from artificial light. The amount of available light influences the available power and thereby the possible update rate of STM 550B as discussed in chapter 2.4.1. The device should therefore be installed such that the amount of light available to the solar cell is maximized.

2.2.3 Backup battery

STM 550B provides a backup battery interface to mount a CR1632 battery for cases with insufficient ambient light. The backup battery has to be installed with the positive pole pointing upwards (i.e. towards the side of the solar cell).

EnOcean recommends Renata CR1632 due to its low self-discharge and high capacity. Gloves should be used when installing a backup battery to avoid contamination of the isolation area between the contacts that could lead to a higher self-discharge.

The backup battery can be removed (ejected) by using a small, non-conductive item (e.g. wooden toothpick) to push the battery out.



Do not insert any tools into the battery slot or the battery ejection slot! Doing so could create a short circuit or damage the PCB resulting in permanent damage!

2.2.4 Product label

Each STM 550B module contains a product label with a commissioning QR code. Figure 4 shows the STM 550B product label. Note the commissioning QR code (described in chapter is on the bottom left side.

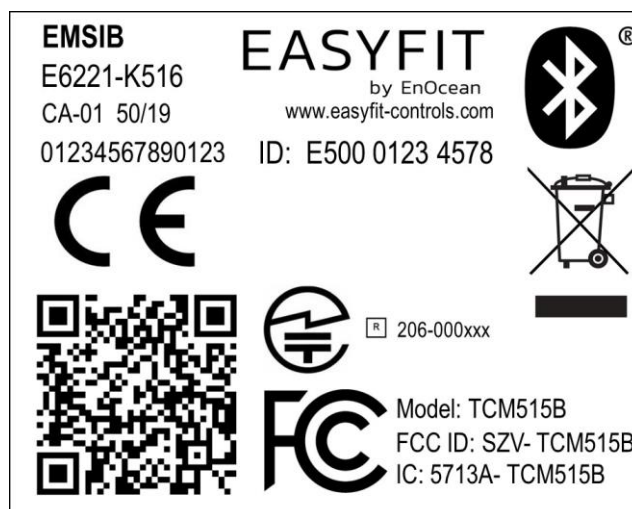


Figure 4 – STM 550B product label**2.3 Functional modes**

STM 550B supports six types of functional modes:

- Standard operation mode
- Standby (Sleep) mode
- Learn mode
- Function test mode
- Illumination test mode
- Factory reset mode



Out of the box, STM 550B will be in standby mode to conserve energy during transport and storage. Upon initial setup, STM 550B has to be set to standard operation mode by pressing the LRN button shortly as described in chapter 9.1 or via the NFC interface as described in chapter 8.5.17

2.3.1 Standard operation mode

During standard operation, STM 550B wakes up periodically and reports the current sensor status using data telegrams.

The STM 550B wake-up timer is configured to wake-up STM 550B approximately every 60 seconds. If acceleration exceeding the configured threshold is detected for the first time after a period without exceeding this threshold then STM 550B wakes up immediately (wake on acceleration event).

Likewise, if the status of the magnet contact changes (from open to closed or vice versa) then this is reported immediately as well (wake on magnet contact event).

The wake-up intervals are affected at random (i.e. a small random offset is added or subtracted) in order to increase the robustness of the radio transmission and to comply with regulatory requirements.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2.3.2 Standby (Sleep) mode

Standby (sleep) mode is the lowest power mode of STM 550B and is the out of the box state of STM 550B upon delivery. It is intended to be used during extended periods without operation such as device storage or transport.

In standby mode, STM 550B stops operation and conserves as much energy as possible. All functionality – except those needed to return to standard operation mode – are disabled in this mode.

Standby mode can be selected using the LRN button as described in chapter 2.2.1 or using the MODE field of the FUNCTIONAL_MODE NFC register as described in chapter 8.5.17.

2.3.3 Learn mode

In learn mode, STM 550B will transmit a commissioning telegram as described in chapter 6.1 to communicate the device source address and security key to a receiver. After that transmission, STM 550B will return back to standard operation mode.

Learn mode can be selected using the LRN button as described in chapter 2.2.1 or using the MODE field of the FUNCTIONAL_MODE NFC register as described in chapter 8.5.17.

2.3.4 Function test mode

In Function Test Mode, STM 550B will measure and report the status of the integrated sensors at the highest possible rate to verify the sensor functionality.

STM 550B can be set into function test mode via the LRN button as described in chapter 2.2.1 or via the MODE field of the FUNCTIONAL_MODE NFC register as described in chapter 8.5.17.

Function Test Mode will be active for 120 seconds unless the LRN button is pressed before or the functional mode is changed via the NFC interface.

2.3.5 Illumination test mode

During installation, STM 550B can measure and report the amount of ambient light available at its solar cell in order to determine a suitable installation location as discussed in chapter 9.5.

Upon activation of light test mode, STM 550B will first wait for 15 seconds so that the installer can leave the area to ensure a relevant measurement result.

After that, STM 550B will take measurements of the ambient light level using its solar cell every 5 seconds for a period of one minute and compute the average illumination based on those measurements.

The computed average illumination is then available in the NFC register ILLUMINATION_TEST_RESULT as described in chapter 8.5.33.

Illumination test mode can be selected using the MODE field of the FUNCTIONAL_MODE NFC register as described in chapter 8.5.17.

2.3.6 Factory reset mode

STM 550B can be reset to its standard settings using factory reset mode. Upon entering this mode, STM 550B will reset all configuration registers to their default settings and then restart operation in standard operation mode.

Factory reset mode can be selected using the LRN button as described in chapter 2.2.1 or using the MODE field of the FUNCTIONAL_MODE NFC register as described in chapter 8.5.17.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2.4 Reporting interval

STM 550B will transmit its status as data telegram at a regular interval, the so-called reporting interval. By default, the reporting interval is 60 seconds, i.e. STM 550B will measure and report its status approximately once per minute.

STM 550B is designed to apply fluctuations up to +/-10% to any configured reporting interval to increase transmission reliability and meet regulatory requirements.

STM 550B will always report the initial acceleration detection after a period without detected acceleration immediately. Likewise, STM 550B will report any change in the status of the magnet contact sensor (open -> closed or closed -> open) immediately.

2.4.1 Energy considerations

The default reporting interval of 60 seconds (one update per minute) is adjustable using the NFC interface as discussed in the subsequent chapters. The minimum possible reporting interval is 3 seconds and the maximum possible transmission interval is 65535 seconds.

Lowering the reporting interval of STM 550B will increase its power consumption since it will measure and transmit more often. Likewise, increasing the reporting interval of STM 550B will reduce its power consumption since it will measure and transmit less often.

To select the right reporting interval, it is essential to determine the amount of harvestable energy. STM 550B harvests energy from the available ambient light; therefore, the amount of available energy is determined mainly by the intensity of the available light and the amount of time during which the light is available.

The amount of available ambient light can be determined by executing an illumination test as described in chapter 2.3.5. The light availability period (the time during which the ambient light is available) has to be determined based on the lighting scheme used for the environment where STM 550B is installed.

STM 550B is designed to provide one update per minute based on 200 lux of ambient light available for 6 hours per day. The minimum supported update interval for self-supplied operation based on other conditions is summarized in Table 2 below.

	6 hrs / day	8 hrs / day	10 hrs / day	12 hrs / day
50 lux	Not supported	Not supported	600 s	300 s
100 lux	300 s	120 s	60 s	60 s
150 lux	120 s	45 s	30 s	30 s
200 lux	60 s	30 s	30 s	20 s
300 lux	30 s	20 s	15 s	15 s
400 lux	20 s	15 s	10 s	10 s
500 lux	15 s	10 s	10 s	10 s

Table 2 – Minimum self-supplied reporting interval for different lighting conditions

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2.4.2 Standard reporting interval

The standard reporting interval determines the longest interval between two status updates of STM 550B, i.e. the minimum update rate.

STM 550B can be configured to use a lower reporting interval, i.e. a higher update rate, based on acceleration and available light as described below. But under no circumstances will STM 550B report with a longer reporting interval, i.e. a lower update rate, than the standard reporting interval.

The default setting for the standard reporting interval is one status update once every 60 seconds (1 minute). This means that STM 550B will at least report its status once every 60 seconds while sufficient energy is available.

The standard reporting interval can be adjusted using the `STANDARD_TX_INTERVAL` NFC register as described in chapter 8.5.18. Consider the available energy before lowering the reporting interval as discussed in chapter 2.4.1.

Figure 5 below illustrates the use of the standard reporting interval.

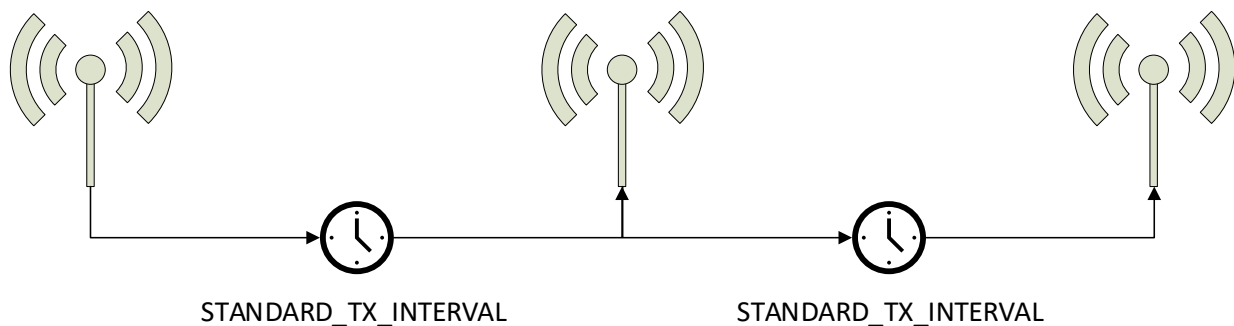


Figure 5 – Standard reporting interval

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2.4.3 Illumination-controlled reporting interval

If sufficient ambient light is available, then it might be desirable to receive status updates more often. For this, there are typically two main use cases:

- Adjust the update rate based on the ambient light available for harvesting
- Report more often during daytime (or when an office is lit) and less often during night-time (or when an office is dark) to adapt the reporting to the usage pattern

In both cases, the higher update rate would be used whenever the ambient light level is above a certain threshold. Figure 6 below illustrated the use of the illumination-controlled reporting interval.

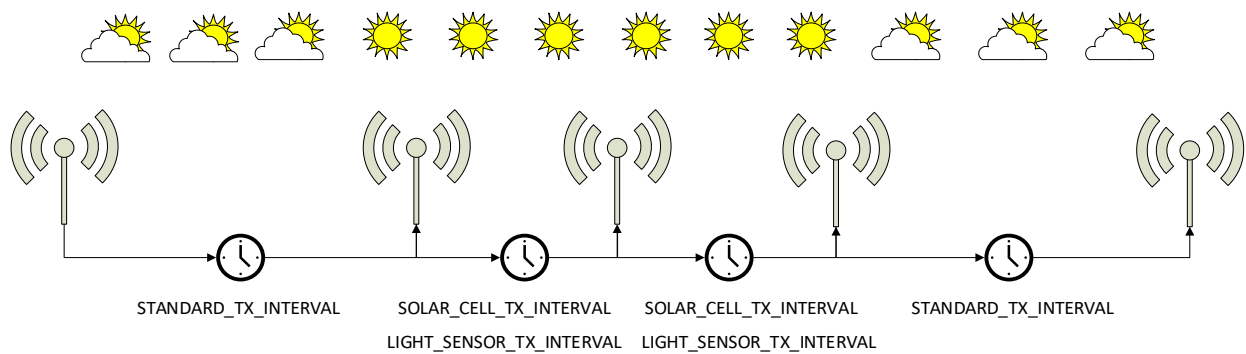


Figure 6 – Illumination-controlled reporting interval

STM 550B can use either the light level at the solar cell (harvested energy) or the light level at the ambient light sensor to trigger a higher update rate. To enable this feature, use the following steps:

1. Make sure that the light measurement (for the solar cell, for the light sensor or for both) is enabled in the REPORTING_CFG register as described in chapter 8.5.15
2. Configure the desired light level threshold either for the solar cell or the ambient light sensor using either the SOLAR_THRESHOLD or LIGHT_THRESHOLD registers described in chapters 8.5.22 and 8.5.24 respectively
3. Configure the desired reporting interval using either the SOLAR_TX_INTERVAL or the LIGHT_TX_INTERVAL registers described in chapters 8.5.23 and 8.5.25 respectively
4. Enable the use of the defined parameters using the SOLAR CELL and LIGHT SENSOR bit fields in the THRESHOLD_CFG1 register of the NFC interface as defined in chapter 8.5.19.

Consider the available energy before lowering the reporting interval as discussed in chapter 2.4.1.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2.4.4 Temperature-controlled reporting interval

In HVAC (heating, ventilation, air conditioning) applications it might be desirable to receive status updates more often if the measured temperature is significantly above or below the target value.

Figure 7 below illustrates the use of the temperature-controlled reporting interval.

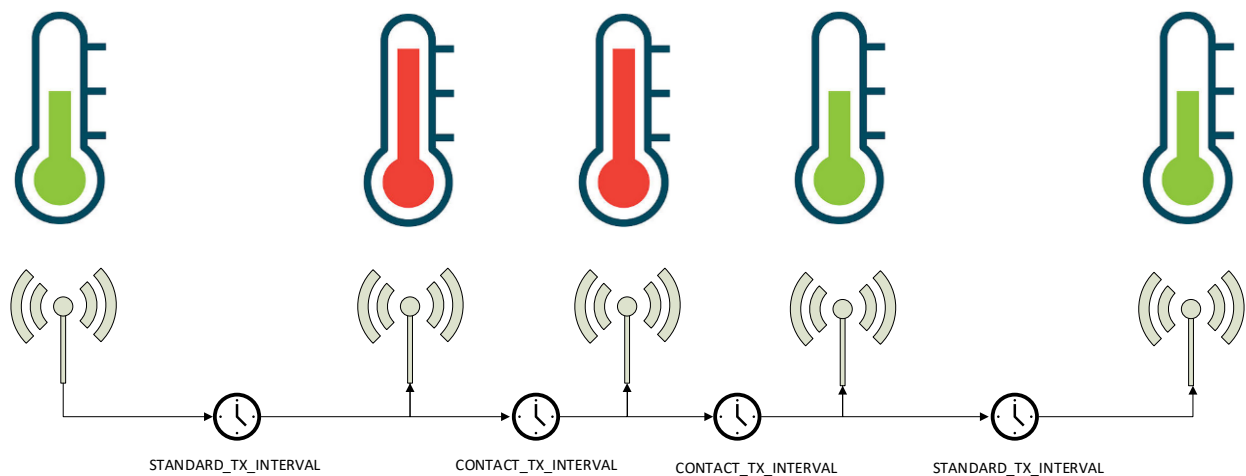


Figure 7 – Temperature-controlled reporting interval

STM 550B can use the temperature measured by the temperature and humidity sensor to trigger a higher update rate. To enable this feature, use the following steps:

1. Make sure that the temperature measurement is enabled in the REPORTING_CFG register as described in chapter 8.5.15
2. Configure the desired temperature threshold using the TEMPERATURE_THRESHOLD register described in chapter 8.5.28
3. Configure the desired reporting interval using the TEMPERATURE_TX_INTERVAL register described in chapter 8.5.29
4. Enable the use of the defined parameters using the TEMP SENSOR bit field in the THRESHOLD_CFG1 register of the NFC interface as defined in chapter 8.5.19.

Consider the available energy before lowering the reporting interval as discussed in chapter 2.4.1.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2.4.5 Humidity-controlled reporting interval

In HVAC (heating, ventilation, air conditioning) applications it might be desirable to receive status updates more often if the measured humidity is significantly above or below the target value.

Figure 8 below illustrates the use of the humidity-controlled reporting interval.

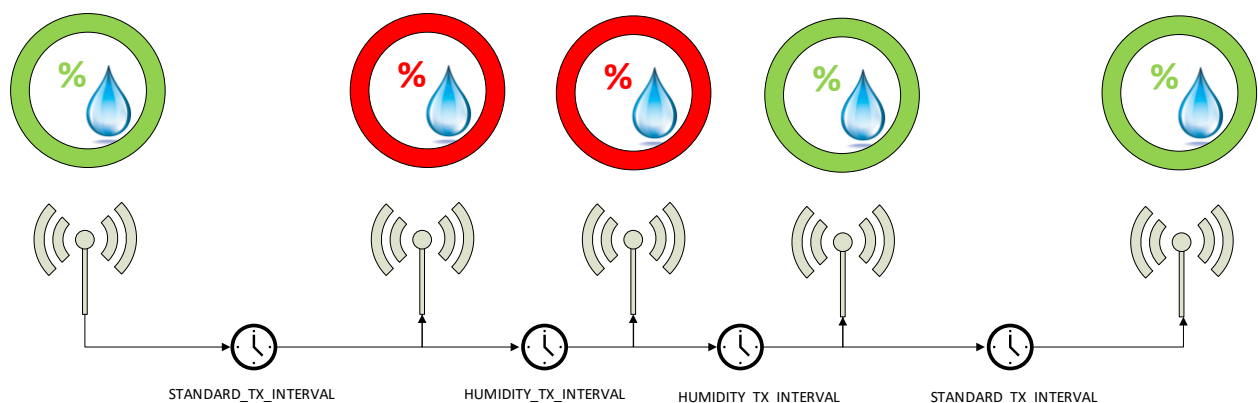


Figure 8 – Humidity-controlled reporting interval

STM 550B can use the humidity measured by the temperature and humidity sensor to trigger a higher update rate. To enable this feature, use the following steps:

1. Make sure that the humidity measurement is enabled in the REPORTING_CFG register as described in chapter 8.5.15
2. Configure the desired humidity threshold using the HUMIDITY_THRESHOLD register described in chapter 8.5.30
3. Configure the desired reporting interval using the HUMIDITY_TX_INTERVAL register described in chapter 8.5.31
4. Enable the use of the defined parameters using the HUMIDITY SENSOR bit field in the THRESHOLD_CFG1 register of the NFC interface as defined in chapter 8.5.19.

Consider the available energy before lowering the reporting interval as discussed in chapter 2.4.1.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2.4.6 Acceleration-controlled reporting interval

If an asset is in operation or it is being moved, then it might be desirable to receive status updates more often to determine its status or location.

Figure 9 below illustrates the use of the acceleration-controlled reporting interval.

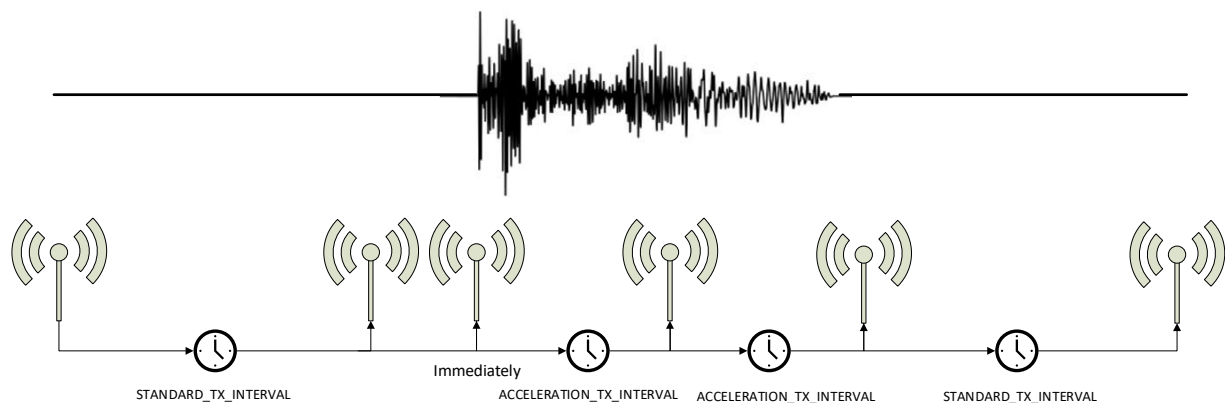


Figure 9 – Acceleration-controlled reporting interval

STM 550B can use acceleration events detected by the acceleration sensor to trigger a higher update rate. To enable this feature, use the following steps:

1. Make sure that the acceleration measurement is enabled in the `REPORTING_CFG` register as described in chapter 8.5.15
2. Configure the acceleration sensor parameters (full scale value and sampling frequency) and enable the wake-on-acceleration feature using the `ACC_SENSOR_CFG` register described in chapter 8.5.21
3. Configure the desired acceleration threshold using the `ACCELERATION_THRESHOLD` register described in chapter 8.5.26
4. Configure the desired reporting interval using the `ACCELERATION_TX_INTERVAL` register described in chapter 8.5.27
5. Enable the use of the defined parameters using the `ACCELERATION_SENSOR` bit field in the `THRESHOLD_CFG2` register of the NFC interface as defined in chapter 8.5.20.

Consider the available energy before lowering the reporting interval as discussed in chapter 2.4.1.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

2.4.7 Magnet contact sensor-controlled reporting interval

If a door or a window is opened when it normally should be closed (or vice versa), then it might be desirable to receive status updates more often to monitor its status. STM 550B can therefore be configured to use a lower reporting interval, i.e. a higher update rate, for one of the two magnet contact sensor status options (open or closed).

Figure 10 below illustrates the use of the magnet contact sensor-controlled reporting interval.

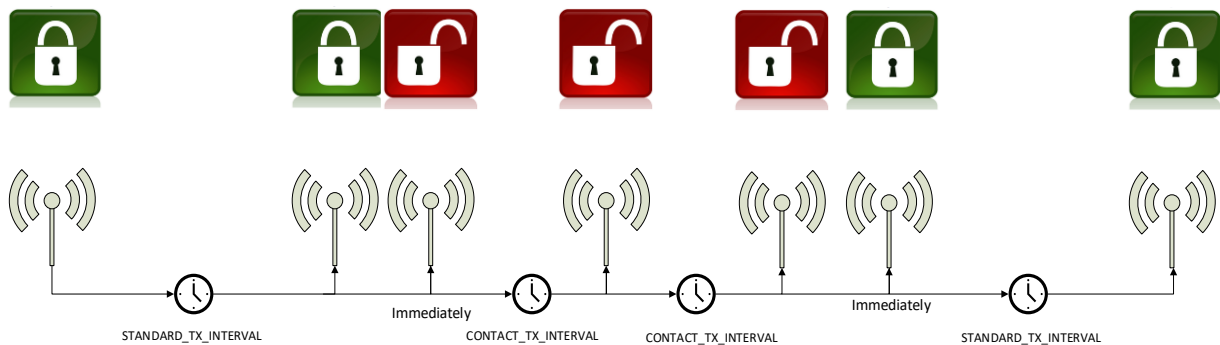


Figure 10 – Magnet contact sensor-controlled reporting interval

STM 550B can use the status of the magnet contact to trigger a higher update rate. To enable this feature, use the following steps:

1. Make sure that the magnet contact measurement is enabled in the REPORTING_CFG register as described in chapter 8.5.15
2. Configure the desired reporting interval using the MAGNET_CONTACT_TX_INTERVAL register described in chapter 8.5.27
3. Select the state of the magnet contact (open or closed) where the lower update interval should be active using the MC SENSOR bit field in the THRESHOLD_CFG2 register as defined in chapter 8.5.20.

Consider the available energy before lowering the reporting interval as discussed in chapter 2.4.1.

2.4.8 Arbitration between reporting intervals

If more than one condition for a lower reporting interval applies – e.g. both an acceleration exceeding the acceleration threshold is detected and the room is brightly lit in excess of the light level threshold – then the lowest of the corresponding reporting intervals will be selected.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

3 Sensor functionality

STM 550B integrates a set of sensors used to measure environmental parameters. The following chapters described their characteristics in detail.

3.1 Ambient light sensor

STM 550B integrates a dedicated ambient light sensor with narrow aperture and a spectral response optimized to mimic the human eye's perception of ambient light. This light sensor reports the light level directly underneath the sensor (spot measurement).

Figure 11 shows the spectrum response of the STM 550B illumination sensor compared to that of the human eye.

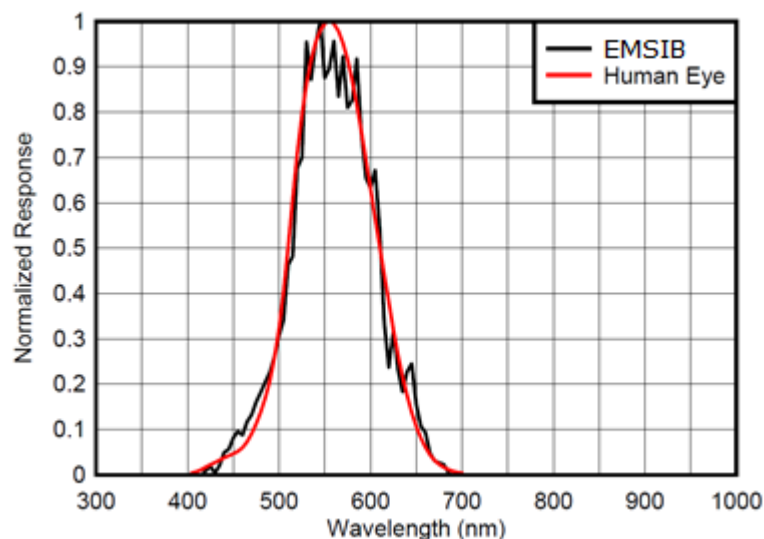


Figure 11 – Spectrum response of the ambient light sensor

3.2 Solar cell

In addition to the light sensor, STM 550B provides the option to use the calibrated solar cell response to report wide area illumination. This can be used both as input for lighting control systems (e.g. to report ambient light for daylighting applications) and to verify the available light level is sufficient for self-powered operation of STM 550B.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

3.3 Temperature sensor

STM 550B integrates a high-performance temperature sensor achieving accuracy of better than ± 0.3 °C throughout the entire operation temperature range and an accuracy of better than ± 0.2 °C for the typical indoor temperature range.

Figure 12 below shows the typical accuracy of the STM 550B temperature sensor as a function of the ambient temperature.

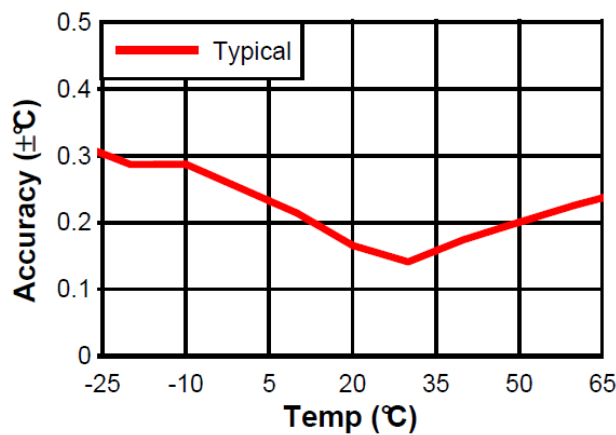


Figure 12 – Temperature sensor accuracy

3.4 Humidity sensor

STM 550B integrates a high-performance humidity sensor achieving accuracy of better than ± 3 % r.h. throughout the entire operation temperature range and an accuracy of better than ± 2 % r.h. for the typical indoor humidity range.

Figure 13 below shows the typical accuracy of the STM 550B humidity sensor as a function of the ambient humidity.

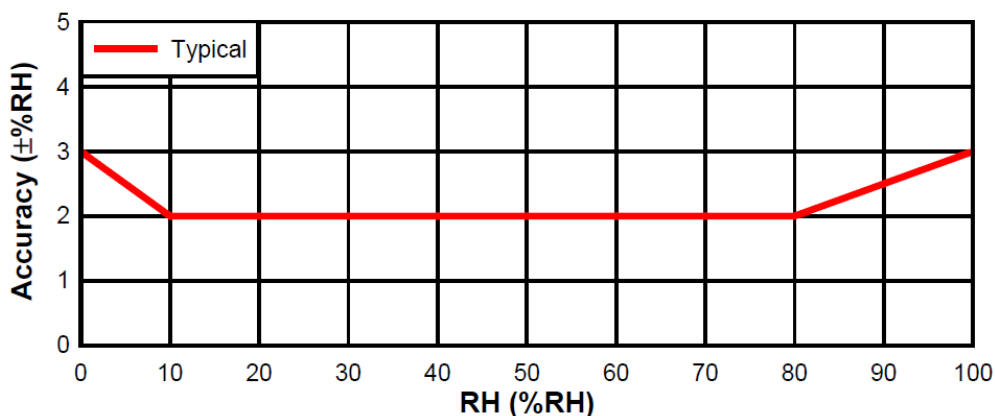


Figure 13 – Humidity sensor accuracy

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

3.5 Acceleration sensor

STM 550B integrates an acceleration sensor which enables two key use cases:

- Measuring device orientation relative to the earth gravity vector
This enables detecting cases where the device is tilted or rotated relative to gravity
- Measuring device acceleration by external force
This enables detecting cases where the device is moved or shaken

The second case uses the presence or absence of small vibrations (acceleration vector changes) to determine asset utilization (e.g. a motor to which STM 550B is attached is running) or asset movement (e.g. an asset to which STM 550B is attached changes its location).

If such change in acceleration is above a certain threshold then this will be reported as part of the regular data telegram. If the acceleration exceeds this threshold for the first time, then this will be reported immediately (wake-on-acceleration).

Figure 14 below shows the orientation of the reported acceleration vector relative to STM 550B.

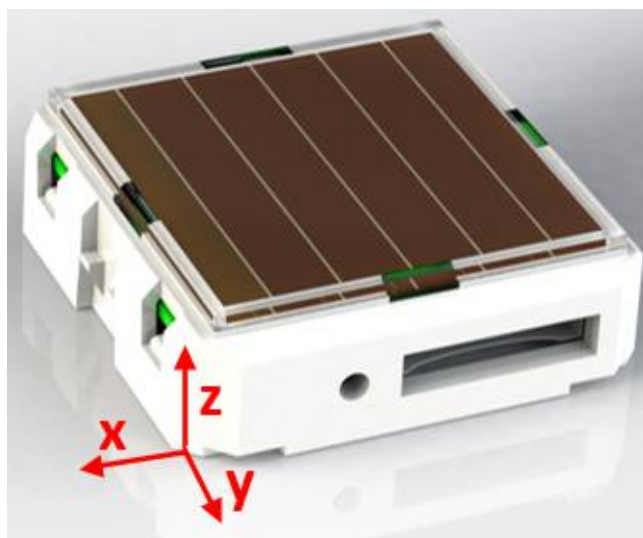


Figure 14 – Acceleration sensor orientation

Table 3 below shows the supported parameters of the acceleration sensor in STM 550B.

Parameter	Supported Options
Full scale	+2g, +4g, +8g, +16g
Sampling rate	1.6 Hz, 12.5 Hz, 25 Hz, 50 Hz
Wake threshold	1/64 ... 63/64 of full scale
Measurement resolution	10 bit for each (x, y, z) direction

Table 3 – Acceleration sensor parameters

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

3.6 Magnet contact sensor

STM 550B integrates a magnet contact sensor detecting presence or absence of a magnetic field in the proximity of the device. Refer to Figure 2 and Figure 3 for the location of the magnet contact sensor within STM 550B.

EMSIB product packaging includes a block magnet suitable for use with its magnet contact sensor. Figure 15 shows the outer appearance of this magnet.

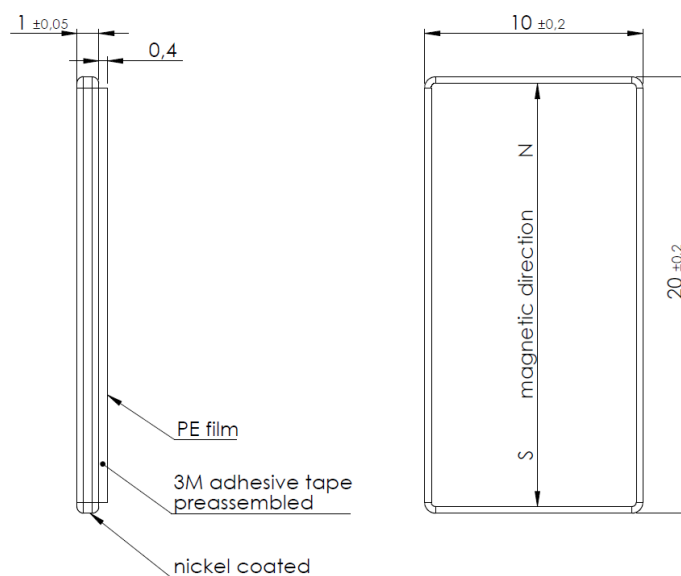


Figure 15 – Magnet outline

The magnet is made from N35 grade Neodymium material. Table 4 below lists the key parameters of this material.

	Br		Hcb		Hcj		(BH) _{max}		TW
	T	KGs	KA/m	KOe	KA/m	KOe	KJ/m ³	MGOe	°C
N35	1.17-1.21	11.7-12.1	876-899	11.0-11.3	≥955	≥12	263-279	33-35	≤80

Table 4 – N35 material parameters

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

4 Radio communication

STM 550B transmits BLE advertising telegrams within the 2.4 GHz radio frequency band (2400 MHz ... 2483.5 MHz) using the BLE advertising frame format.

4.1 Radio channels

STM 550B can use up to three radio channels for the redundant transmission (same telegram on different radio channels) of its telegrams.

By default, the three BLE advertising channels (Channel 37, 38 and 39) will be used for transmission. Use of different radio channels within the frequency band from 2402 MHz to 2480 MHz is possible using the NFC configuration interface, see chapter 8.5.5 and 8.5.6.

Table 5 below summarizes the radio channels supported by STM 550B.

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 5 – Supported radio channels

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

4.2 Default radio transmission sequence

STM 550B 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 on BLE Channel 37, 38 and 39) one after another with minimum delay in between.

For reliability reasons, STM 550B will send three advertising events for each telegram.

The default interval between consecutive advertising events is 20 ms. It is possible to reduce this interval to 10 ms using the ADV_INTERVAL NFC configuration register as described in chapter 8.5.8.

Figure 16 below shows the default radio transmission sequence.



Figure 16 – Default radio transmission sequence

4.3 User-defined radio transmission sequences

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

STM 550B therefore allows selecting 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 4.2 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 TX_CHANNEL_MODE field of the TX_CONFIG register of the NFC configuration interface as described in chapter 8.5.6.

4.3.1 User-defined radio transmission sequences

STM 550B supports the following user-defined radio transmission 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 4.3.2 below.
- Two-channel sequence
In this sequence the radio telegram is transmitted using six transmissions on two radio channels. It is described in chapter 4.3.3 below.
- One-channel sequence
In this sequence the radio telegram is transmitted using nine transmissions on one radio channel. It is described in chapter 4.3.4 below.

4.3.2 Three-channel sequence

The three-channel radio transmission sequence is similar to the default transmission sequence with the difference that the radio channels (BLE Channel 37, 38 and 39 in the default transmission sequence) can be selected using the registers CH_REG1, CH_REG2 and CH_REG3 as described in chapter 8.5.5.

In this mode, the telegram will 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.

The telegram will be transmitted using this sequence three times in total as shown in Figure 17 below.

This transmission uses a default INTERVAL setting of 20 ms; an alternative setting of 10 ms can be configured via NFC as described in chapter 8.5.8.

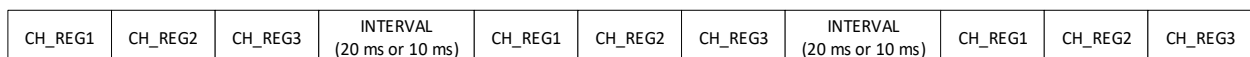


Figure 17 – Three channel radio transmission sequence

4.3.3 Two-channel sequence

The two-channel radio transmission sequence transmits radio telegrams on two user-defined radio channels (selected by CH_REG1 and CH_REG2 as described in chapter 8.5.5.) six times in total.

The 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 is shown in Figure 18 below.

This transmission sequence uses a default INTERVAL setting of 20 ms; an alternative setting of 10 ms can be configured via NFC as described in chapter 8.5.8.

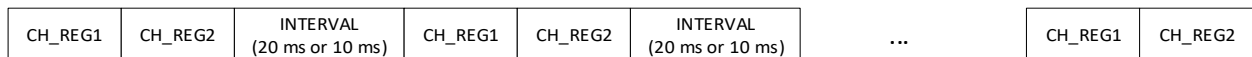


Figure 18 – Two channel radio transmission sequence

4.3.4 One channel sequence

The one-channel radio transmission sequence transmits radio telegrams on one user-defined radio channel (selected by CH_REG1 as described in chapter 8.5.5.) nine times in total. This is shown in Figure 19 below.

This transmission sequence uses a default INTERVAL setting of 20 ms; an alternative setting of 10 ms can be configured via NFC as described in chapter 8.5.8.

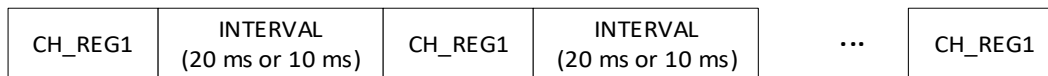


Figure 19 – Single channel radio transmission sequence

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

4.4 Telegram format

STM 550B transmits Bluetooth Low Energy (BLE) radio telegrams in the 2.4 GHz band. For detailed information about the Bluetooth Low Energy standard, please refer to the applicable specifications.

Figure 20 below summarizes the general BLE frame structure.

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

Figure 20 – BLE frame structure

Figure 21 below shows specific properties used by STM 550B within the general BLE frame structure.

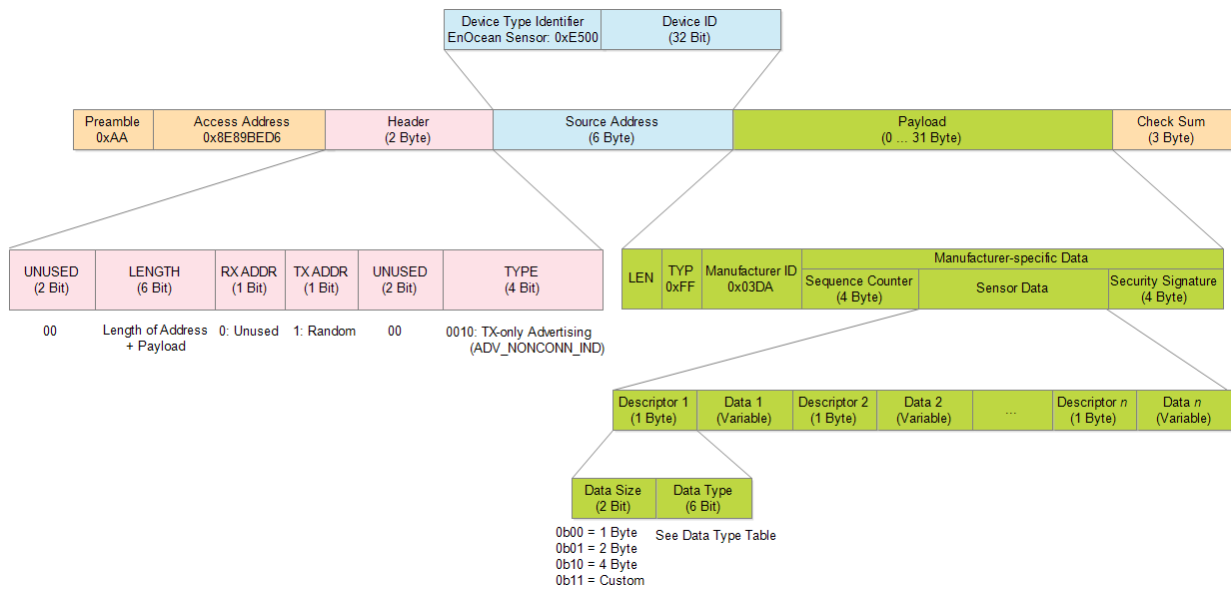


Figure 21 – BLE frame structure

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

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

4.4.1 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.

4.4.2 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.

4.4.3 Header

The BLE Header identifies certain radio telegram parameters. Figure 22 below shows the structure of the BLE header.

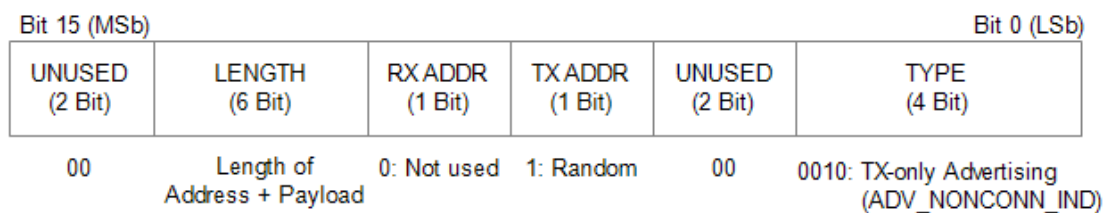


Figure 22 – BLE header structure

4.4.4 Source address

The 6 byte BLE Source Address (MAC address) uniquely identifies each STM 550B product.

STM 550B 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)
- Resolvable Private Address mode (NFC configurable)
In this mode, the source address changes for each transmission

STM 550B uses by default Static Source Address mode. Resolvable Private Address mode can be selected by setting the RPA field in the SECURITY_CONFIG register to 0x01 as described in chapter 8.5.14.

4.4.5 Static source address mode

By default, STM 550B uses static source addresses meaning that the source address is constant during normal operation. The lower 4 byte of the static source address can be modified via NFC as described in chapter 8.5.4.

The structure of STM 550B static addresses is therefore as follows:

- The upper 2 bytes of the source address are for EnOcean Bluetooth sensors always set to 0xE500 to enable filtering according to product type
- The lower 4 bytes are uniquely assigned to each device. They can be read and changed using the NFC configuration interface as described in chapter.

Figure 23 below illustrates the static address structure used by STM 550B.



Figure 23 – BLE static source address structure

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

4.4.6 Resolvable Private Address mode

For some applications it is desirable to modify (rotate) the source address used by STM 550B in order to prevent tracking of its radio transmissions. At the same time, each STM 550B device must remain uniquely identifiable by the receiver. To achieve these goals, STM 550B can be configured via NFC to use resolvable private addresses (RPA).

Using resolvable private addresses requires that both STM 550B and the receiver both know a common key – the so-called Identity Resolution Key (IRK).

STM 550B uses its device-unique random key as identity resolution key. This key can be modified via the NFC configuration interface as described in chapter 8.5.12.

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 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 24 below.

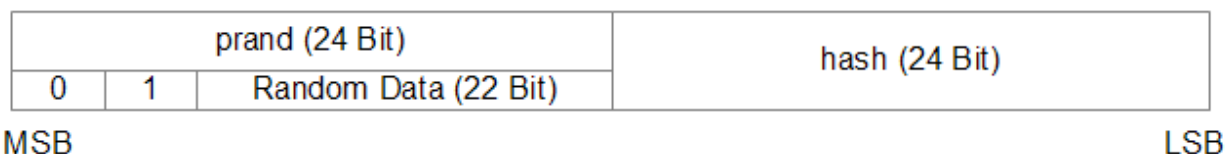


Figure 24 – BLE resolvable private address structure

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 resolvable private address.

The receiver maintains a list of IRK for all transmitters that are known to it (have been commissioned to work with it). Whenever it receives a radio telegram with resolvable private 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.

Figure 25 below illustrates the address resolving scheme for random private addresses. For an example of resolving an address, please refer to Appendix C.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

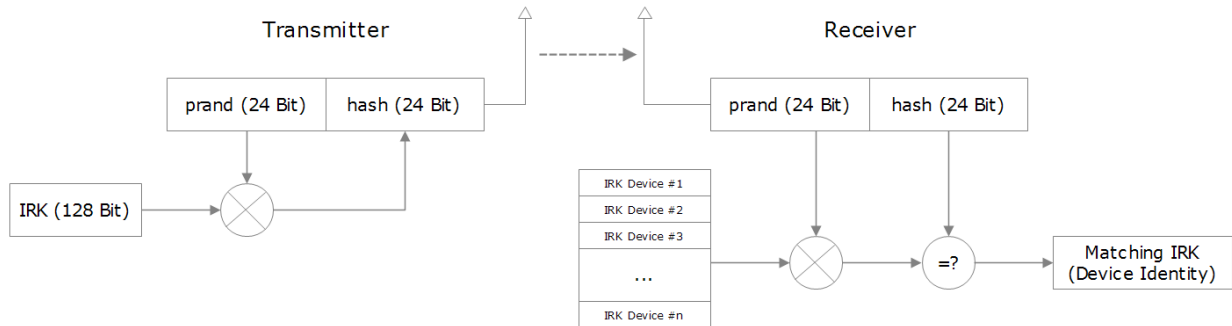


Figure 25 – Resolving resolvable private addresses

4.4.7 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.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

4.5 Payload structure

The payload of STM 550B data telegrams can in general be up to 31 bytes long (depending on the size of the sensor data) and consists of the following fields:

- Length (1 byte)
The Length field specifies the combined length of the following fields and depends on the size of the Sensor Status field. The minimum length is 13 byte and the maximum length is 31 byte
- Type (1 byte)
The Type field identifies the data type used for this telegram. For STM 550B 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. EnOcean has been assigned 0x03DA as manufacturer ID code.
- 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.
- Sensor Data (variable size)
The Sensor Data field reports the measured values of the sensors. The encoding of this field is described in chapter 4.5.1.
- Security Signature (4 byte)
The Security Signature is used to authenticate EnOcean BLE sensor data telegrams, see chapter 5.

Figure 26 below illustrates the general telegram payload structure.

LEN	TYP 0xFF	Manufacturer ID 0x03DA	Manufacturer-specific Data		
			Sequence Counter (4 Byte)	Sensor Data (2 ... 19 Byte)	Security Signature (4 Byte)

Figure 26 – Telegram payload structure

4.5.1 Sensor status encoding

The Sensor Status field within the Payload data identifies the status of the connected sensors. The Sensor Status field is composed of sub-fields (one per sensor attribute).

Each sub-field consists of two items:

- **Sensor Data Descriptor**
The descriptor identifies the type and the size of the following sensor data
- **Sensor Data**
The sensor data field encodes the measured value

Figure 27 below shows the structure of the sensor status field.

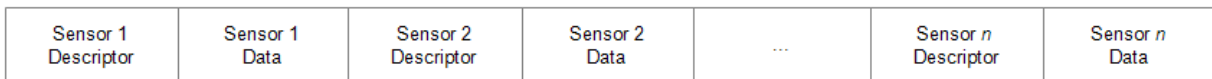


Figure 27 – Sensor Status field structure

4.5.2 Sensor Data Descriptor

The Sensor Data Descriptor describes type and size of the following sensor data field. It explicitly specifies the size to ensure forward compatibility, i.e. to enable future receivers to parse sensor telegrams containing unknown data types.

The Sensor Data Descriptor structure is shown in Figure 28 below.



Figure 28 – Sensor Data Descriptor field structure

The sensor data descriptor explicitly specifies the data size to ensure forward compatibility for the case where an existing sensor does not “understand” subsequently introduced measurement parameters and therefore can’t determine the size of their data field.

In this case, the sensor can use the length information provided by this field to determine the start of the next sensor descriptor field (which might contain usable data).

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

4.5.3 Data Size

The following values are possible for the Data Size field:

- 0b00 = 8 bit size (or implicit definition, e.g. commissioning telegram = 22 byte)
- 0b01 = 16 bit size
- 0b10 = 32 bit size
- 0b11 = Extended size, the size is specified in the first byte of the Sensor Data field

4.5.4 Supported parameters

STM 550B can report a variety of parameters depending on the product configuration. Parameter reporting can be configured via the REPORTING_CFG register of the NFC configuration interface as described in chapter 8.5.15 .

Keep in mind that the combined parameter length of all items cannot exceed 19 byte due to the BLE Frame structure as shown in Figure 26. Each parameter requires 1 byte Type ID in addition to the Data Length shown in Table 6 below. It is therefore not possible to report all supported parameters.

Table 6 below summarizes the parameters that can be reported by STM 550B and notes the ones that are enabled by default. Please refer to Appendix A for examples of how to interpret an STM 550B data and an STM 550B commissioning telegram.

TYPE ID	Size	Content	Min Value	Max Value	Res	Encoding	Default Reporting
Data Telegrams							
0x00	2	Temperature	-327.67°C	327.66°C	0.01 °C	16 bit signed int	Enabled
0x01	2	Voltage	-16383.5mV	16383 mV	0.5 mV	16 bit signed int	Adaptive
0x02	1	Energy Level	0%	100%	0.5 %	8 bit unsigned int (0 ... 200)	Adaptive
0x04	2	Illumination (Solar cell)	0 lx	65 533 lx	1 lx	16 bit unsigned int	Disabled
0x05	2	Illumination (Sensor)	0 lx	65 533 lx	1 lx	16 bit unsigned int	Enabled
0x06	1	Relative Humidity	0 % r.h.	100 % r.h.	0.5 % rh	8 bit unsigned int (0 ... 200)	Enabled
0x0A	4	Acceleration Vector	See Table 3 and Table 4				Enabled
0x23	1	Magnet Contact	0x01 = Open, 0x02 = Closed			Enumeration	Enabled
0x3C	1, 2, 4	Optional Data	User-defined data				Disabled
Commissioning Telegrams							
0x3E	22	Commissioning Info	16 byte AES key followed by 6 byte advertising address				LRN button

Table 6 – Supported parameters

4.5.5 Acceleration vector encoding

The 32 bit acceleration vector is encoded as shown in Figure 29 below.

ACCELERATION_VECTOR										
Bit 31	Bit 30	Bit 29	...	Bit 20	Bit 19	...	Bit 10	Bit 9	...	Bit 0
STATUS		Z_VECTOR			Y_VECTOR			X_VECTOR		

Figure 29 – ACCELARATION_VECTOR structure

The encoding of the STATUS bit field of the acceleration vector is shown in Table 7 below.

STATUS	Interpretation
0b00	Acceleration value out of bound
0b01	Periodic update
0b10	Acceleration wake
0b11	Sensor disabled

Table 7 – Status field encoding for acceleration vector

The X_VECTOR, Y_VECTOR and Z_VECTOR fields encode the acceleration vector as 10 bit value ranging from 0 ... 1023 according to the following transformation equation:

$$Acceleration = \frac{VECTOR - 512}{100}$$

Table 8 below illustrates this encoding.

VECTOR	Decimal value	Interpretation
0b00:0000:0000	0	- 5.12 g
0b10:0000:0000	512	0 g
0b11:1111:1111	1023	+ 5.11 g

Table 8 – VECTOR field encoding for acceleration vector

Please see Appendix A.1.3 for a step by step data parsing example including the acceleration vector.

5 Security

STM 550B 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 30 below and exchanged as part of the radio telegram.

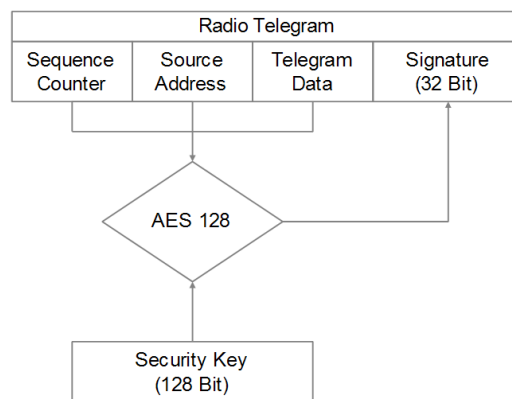


Figure 30 – Telegram authentication flow

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 STM 550B 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 (STM 550B) 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 STM 550B and only accepts messages with higher sequence counter values (i.e. not accepts equal or lower sequence counter values for subsequent telegrams).

5.1 Security implementation

STM 550B 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 31 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 31 – 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 32 below.

Authenticated Sensor Telegram Data									
LEN	TYPE	MANUFACTURER	SENSOR DATA						
Length	0xFF	0x03DA	DESC1	DATA1	DESC2	DATA2	...	DESC _n	DATA _n

Figure 32 – Authenticated payload

The calculated 32 bit signature is then appended to the data telegram payload as shown in in chapter 4.5.

Appendix B provides an example how to authenticate a received data telegram.

6 Commissioning

Commissioning is the process by which STM 550B 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 this specific STM 550B module. This is achieved by using a unique 48 Bit ID (Source Address) for each STM 550B module.
- **Security parameter exchange**
The receiver needs to be able to authenticate radio telegrams from STM 550B in order to ensure that they originate from this specific device and have not been modified. This is achieved by exchanging a 128 Bit random security key used by STM 550B to authenticate its radio telegrams.

STM 550B provides the following options for these tasks:

- **Radio-based commissioning**
STM 550B can communicate its parameters via special radio telegrams (commissioning telegrams) to the intended receiver. Transmission of such telegrams can be triggered by using the LRN button.
- **QR code commissioning**
Each STM 550B module contains an optically readable Quick Response (QR) Code which identifies its ID and its security key. This QR code can be read by a suitable commissioning tool (e.g. smartphone) which is already part of the network into which STM 550B will be commissioned. The commissioning tool then communicates these parameters to the intended receiver of STM 550B radio telegrams.
- **NFC commissioning**
Each STM 550B module contains an NFC interface allowing to read device parameters and to configure the device.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

6.1 Radio-based commissioning

Radio-based commissioning is used to associate STM 550B with other devices by sending a dedicated radio telegram (a so-called commissioning telegram).

To do so, STM 550B can transmit a dedicated commissioning telegram identifying its relevant parameters as described in chapter 4.5.4. Appendix A.2 gives an example how to parse such commissioning telegram.

The commissioning telegram will by default be transmitted on the BLE advertising channels (CH 37, 38 and 39). Use of custom radio channels is possible as described in chapter 4.3.

The transmission of the commissioning telegram is triggered by pressing the LRN button or by setting the MODE field of the FUNCTIONAL_MODE register to 0b0010 as described in chapter 8.5.17.

Radio-based commissioning mode is intended for applications where NFC commissioning cannot be used. Radio-based commissioning can be disabled via NFC.

6.2 QR code-based commissioning

QR code-based commissioning reads the required parameters from a dedicated QR code in the product label (described in chapter 2.2.4). The parameters are encoded according to the ANSI/MH10.8.2-2013 industry standard. The QR code shown in Figure 4 of chapter 2.2.4 encodes the following string:

```
30SE50012345678+Z0123456789ABCDEF0123456789ABCDEF+30PE6221-K516+2PCA01+S01234567890123
```

Identifier	Length of data (excluding identifier)	Value
30S	12 characters	Static Source Address (hex)
Z	32 characters	Security Key (hex)
30P	Up to 10 characters	Ordering Code (E6221-K516)
2P	4 characters	Step Code - Revision (CA-01)
S	14 characters	Serial Number

Table 9 – QR code format

From this content, it is possible to extract the device address (E50077010000) and the security key (4DB34DB3070EFC6713FE39E13CF3C939) which can then be used to commission STM 550B into a receiver and to authenticate STM 550B data telegrams as described in chapter 5.

6.3 Commissioning via NFC interface

STM 550B implements NFC Forum Type 2 Tag functionality as specified in the ISO/IEC 14443 Part 2 and 3 standards.

This NFC functionality can be used to read the device address and the security key of STM 550B as described in chapters 7 and 8.

7 NFC interface

STM 550B implements an NFC configuration interface that can be used to access (read and write) the STM 550B configuration memory and thereby configure the device as described in the following chapters.

NFC communication distance is for security reasons set to require direct contact between the NFC reader and the STM 550B module.

Note that STM 550B temporarily stops operation while the NFC reader is actively connected to the NFC interface of STM 550B. STM 550B operation will automatically resume operation once the NFC reader has been disconnected.

7.1 NFC interface parameters

The NFC interface of STM 550B uses NFC Forum Type 2 Tag functionality as specified in the ISO/IEC 14443 Part 2 and 3 standards. It is implemented using an NXP NT3H2111 Mifare Ultralight tag.

7.2 NFC access protection

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 `0x0000E500`.

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 `0x4B` of the CONFIGURATION area as described in chapter 8.5.3.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

7.3 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

For PC applications, EnOcean recommends the TWN4 Multitech 2 HF NFC Reader with simple CDC interface (order code T4BT-FB2BEL2-SIMPL) from Elatec RFID Systems (sales-rfid@elatec.com). This reader is shown in Figure 33 below.



Figure 33 – Elatec TWN4 MultiTech Desktop NFC Reader

NFC functionality is also available in certain Android and iOS smartphones and tablets. Check with the device manufacturer for the NFC capability of such devices.

7.4 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 under this link:

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

The following chapters summarize the different functions for reference purposes.

7.4.1 NFC interface state machine

Figure 34 below shows the overall state machine of the NFC interface.

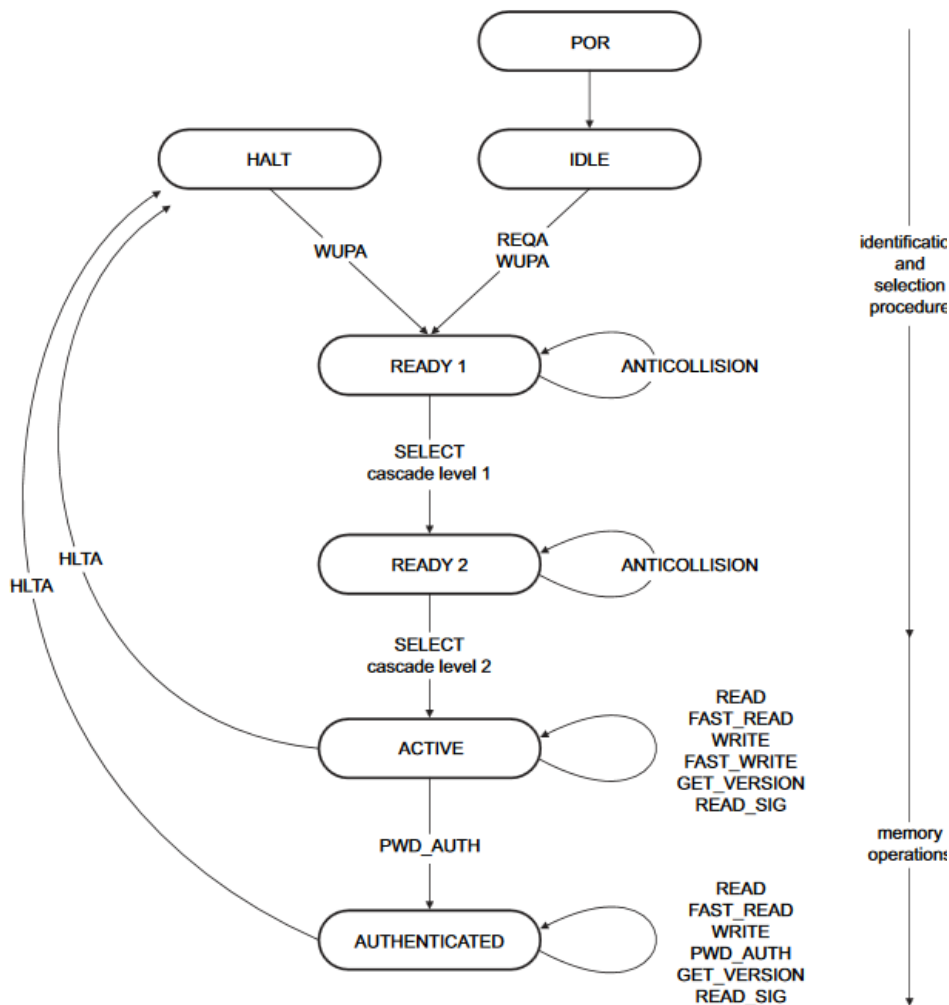


Figure 34 – NFC interface state machine

7.4.2 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.

7.4.3 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 with 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.

7.4.4 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.

7.4.5 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.

7.4.6 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 35 below shows the read command sequence.

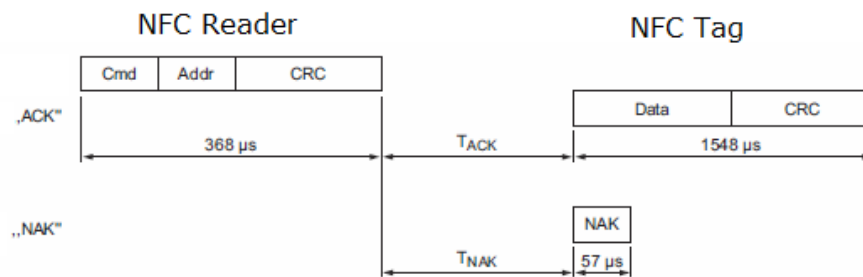


Figure 35 – NFC read command sequence

7.4.7 Write command

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

Figure 36 below shows the read command sequence.

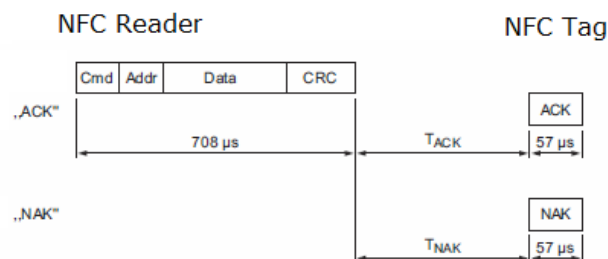


Figure 36 – NFC write command sequence

7.4.8 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 37 below shows the password authentication sequence.

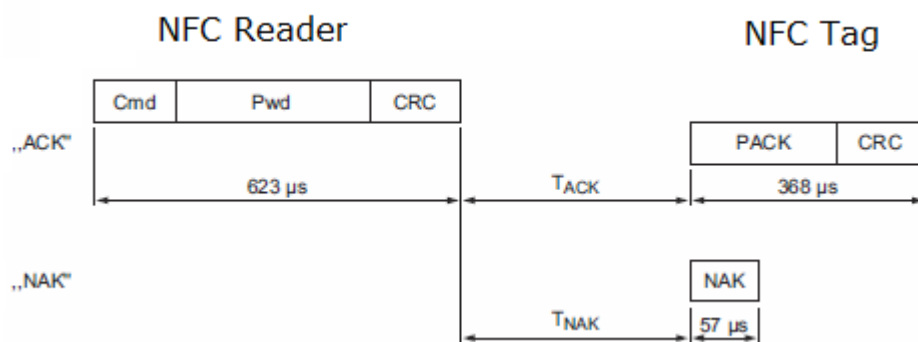


Figure 37 – Password authentication sequence

After successful authentication, the password can be changed by writing the new password to memory page 0x4B.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

8 NFC registers

The NFC memory is organized in pages (smallest addressable unit) where each page contains 4 byte of data. Several pages with similar functionality form an NFC memory area.

8.1 NFC memory areas

These NFC pages are allocated into the following areas:

- NDEF (Public read-only access; no PIN required)
This area contains an NDEF string identifying key device parameters
- PUBLIC INFO (Public read-only access; no PIN required)
This area contains key device parameters in binary format
- NFC HEADER (Public read-only access; no PIN required)
This area contains information about the NFC revision
- INTERNAL DATA (Non-accessible)
This area contains calibration values and internal parameters and cannot be used
- CONFIGURATION (Read and Write access, PIN required)
This area contains device configuration registers
- USER DATA (Read and Write access, PIN required)
This area allows the user to store up to 64 byte of data.
STM 550B does not use this area and does not interpret its content in any way.

The organization of the STM 550B NFC memory map is shown in Table 10 below.

NFC Address	Memory Area	Content
0x00 ... 0x1A	NDEF	NDEF Device identification string (read-only)
0x1B	SW VERSION	SW version in 4 number notation (e.g. 1.0.0.0)
0x1C ... 0x1F	INTERNAL DATA	Internal data (Do not use)
0x20 ... 0x22	NFC HEADER	NFC memory revision (read-only)
0x23 ... 0x3F	INTERNAL DATA	Internal data (Do not use)
0x40 ... 0x52	CONFIGURATION	Configuration registers (Read / Write, PIN protected)
0x53 ... 0xCF	INTERNAL DATA	Internal data (Do not use)
0xD0 ... 0xDF	USER DATA	User data (64 byte read / write access, PIN protected)
0xE0 ... 0xEB	INTERNAL DATA	Internal data (Do not use)

Table 10 – STM 550B NFC memory areas

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

8.2 NDEF

The NDEF area contains a device identification string using the NDEF (NFC Data Exchange Format) standard that is readable by most NFC-capable reader devices (including smartphones).

An example device identification string from the NDEF area of STM 550B would be:

```
30SE50000006DF+Z4BBE99C695FB5AB91DB8499E9206EE6F+30PS6221-K516+2PCA03+3C20
```

This NDEF string encodes the parameters shown in Table 11 below.

Identifier	Length of data (excl. identifier)	Value
30S	12 characters	Static Source Address (6 byte, variable)
Z	32 characters	Security Key (16 byte, variable)
30P	10 characters	Ordering Code (S6221-K516)
2P	4 characters	Step Code and Revision (CA03)
3C	2 characters	Header Start Address (0x20)

Table 11 – NDEF Parameters

Note that the security key is only available if the Security Key Access in the Security Configuration register is set to 0b00 (Public Access) as described in chapter 8.5.13. Otherwise the security key will be listed as all zeros.

8.3 SW_VERSION

The SW_VERSION register contains information about the version of the STM 550B module firmware.

The SW_VERSION register is organized as a sequence of 4 byte in the following order:

- Major revision (Byte 0)
- Minor revision (Byte 1)
- Major sub-revision (Byte 2)
- Minor sub-revision (Byte 3)

If the value in SW_VERSION would be 0x01020304 then the corresponding SW version would be 1.2.3.4.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

8.4 NFC HEADER

The NFC HEADER area contains information about the NFC memory structure and can therefore be used to distinguish between different NFC memory layouts.

8.4.1 NFC HEADER area structure

The structure of the NFC HEADER area is shown in Figure 38 below.

NFC Address	Content			
	Byte 0	Byte 1	Byte 2	Byte 3
0x20	START (0xE0)	LENGTH (0x0A)	VERSION (0x01)	OEM MSB (0x00)
0x21	OEM LSB (0x0B)	DEVICE_IDENTIFIER (0x0B0003)		
0x22	REVISION (0x02)	END (0xFE)	UNUSED (0x0000)	

Figure 38 – NFC HEADER area structure

The NFC HEADER contains the following fields:

- **START**
This field identifies the start of the NFC header and is always set to 0xE0
- **LENGTH**
This field identifies the length of the NFC header.
For STM 550B, this field is set to 0x0A since the header structure is 10 bytes long
- **VERSION**
This field identifies the major revision and is set to 0x01 currently
- **OEM**
The 16 bit OEM field identifies the manufacturer of the device so that manufacturer-specific layout implementations can be determined.
For EnOcean GmbH this field is set to 0x000B
- **DEVICE_IDENTIFIER**
The 24 bit DEVICE_IDENTIFIER field identifies an individual device from the range of devices manufactured by the manufacturer specified in the OEM field.
For STM 550B, the DEVICE_IDENTIFIER is set to 0x0B0003
- **REVISION**
The REVISION field identifies the exact revision of the NFC layout. This REVISION will be incremented whenever a change to the NFC layout is made.
- **END**
The END field identifies the end of the NFC header and is always set to 0xFE. The number of bytes from START to END must equal LENGTH, otherwise the NFC header is invalid.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

8.5 CONFIGURATION

The CONFIGURATION area allows configuring the device parameters and is therefore the most important part of the NFC memory.

Read or write access to the CONFIGURATION area is only possible after issuing a PWD_AUTH command as described in chapter 7.4.8 using the correct 32 bit PIN code.

8.5.1 Using the NFC configuration functionality

Before making any changes to the default configuration, be sure to familiarize yourself with the functionality of the device and the effect of the intended changes.

STM 550B will not accept the setting of non-valid values for its parameters. If any parameter is non-valid then all changes made will be rejected and the previous configuration will be restored.

8.5.2 CONFIGURATION area structure

The structure of the CONFIGURATION area is shown in Figure 38 below.

NFC Address	Content			
	Byte 0	Byte 1	Byte 2	Byte 3
0x40	SOURCE ADDRESS			
0x41	CH_REG1	CH_REG2	CH_REG3	TX_CFG
0x42	TX_POWER	ADV_INTERVAL	MANUFACTURER_ID	
0x43	OPTIONAL_DATA			
0x44	SECURITY_KEY (128 Bit) All zero if security key access is disabled			
...				
0x47				
0x48	SECURITY_KEY_ACCESS	SECURITY_CFG	RFU	
0x49	REPORTING_CFG	OPTIONAL_DATA_SIZE	LED_MODE	FUNCTIONAL_MODE
0x4A	STANDARD_TX_INTERVAL		RFU	
0x4B	NFC_PIN_CODE			
0x4C	THRESHOLD_CFG1	THRESHOLD_CFG2	RFU	ACC_SENSOR_CFG
0x4D	SOLAR_THRESHOLD		SOLAR_TX_INTERVAL	
0x4E	LIGHT_THRESHOLD		LIGHT_TX_INTERVAL	
0x4F	ACCELERATION_THRESHOLD		ACCELERATION_TX_INTERVAL	
0x50	TEMPERATURE_THRESHOLD		TEMPERATURE_TX_INTERVAL	
0x51	HUMIDITY_THRESHOLD		HUMIDITY_TX_INTERVAL	
0x52	RFU		MAGNET_CONTACT_TX_INTERVAL	
0x53	RFU			
0x54	ILLUMINATION_TEST_RESULT		RFU	

Figure 39 – CONFIGURATION area structure

8.5.3 NFC_PIN_CODE

The PIN code used to protect access to the NFC CONFIGURATION memory area should be changed from the default value to a user-specific value to avoid unauthorized access to the device configuration.

To do so, first authenticate with the current PIN code and then write the new PIN code (32 bit value) to the NFC_PIN_CODE register.

8.5.4 SOURCE_ADDRESS

Each STM 550B module uses a unique 6 byte address (Static Source Address) to identify its messages. This address is assigned during manufacturing.

The two most significant byte of this address are always 0xE500; i.e. the address always starts with 0xE500. The four least significant byte of this address can be read from the SOURCE_ADDRESS register.

The 6 byte address can then be calculated as $(0xE500 \ll 32) + \text{SOURCE_ADDRESS}$

8.5.5 CH_REG1, CH_REG2, CH_REG3

The channel selection registers CH_REG1, CH_REG2 and CH_REG3 define the radio channels used for custom radio transmission sequences as described in chapter 4.3.

The encoding of the radio channels follows the definition listed in Table 5; their default settings are listed in Table 12 below.

Default Values	
CH_REG1	0x25 (CH 37)
CH_REG2	0x26 (CH 38)
CH_REG3	0x27 (CH 39)

Table 12 – Default settings for CH_REG1, CH_REG2 and CH_REG3

8.5.6 TX_CFG

The transmission configuration register TX_CFG identifies the custom radio transmission sequence type (encoded by the bit field CHANNEL_MODE) and the data rate (encoded by the bit field DATA_RATE).

TX_CFG (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU		DATA_RATE		CHANNEL_MODE			

Figure 40 – TX_CFG register

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

The encoding for CHANNEL_MODE is shown Table 13 below.

CHANNEL_MODE	Transmission Sequence
0b0000 (Default)	Commissioning and data telegrams in standard Advertising Mode
0b0001	Commissioning telegrams in standard Advertising Mode Data telegrams on 3 user-defined radio channels
0b0010	Commissioning telegrams in standard Advertising Mode Data telegrams on 2 user-defined radio channels
0b0011	Commissioning telegrams in standard Advertising Mode Data telegrams on 1 user-defined radio channel
0b0100	Commissioning and Data telegrams on 3 user-defined radio channels
0b0101	Commissioning and Data telegrams on 2 user-defined radio channels
0b0110	Commissioning and Data telegrams on 1 user-defined radio channel
0b0111 ... 0b1111	Unused, will be treated as 0b0000

Table 13 – Transmission channel settings

The encoding for the DATA_RATE bit field is shown in Table 14 below.

DATA_RATE	Data Rate
0b00 (Default)	1 Mbit/s data rate
0b01	2 Mbit/s data rate
0b10, 0b11	Reserved, do not use

Table 14 – DATA_RATE bit field encoding

8.5.7 TX_POWER

The register TX_POWER identifies the transmission power used by STM 550B. Figure 41 below shows the structure of the TX_POWER register.

TX_POWER (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU						POWER	

Figure 41 – TX_POWER register

The encoding for the POWER bit field is shown in Table 15 below.

POWER	Transmission Power
0b00 (Default)	+4 dBm
0b01	0 dBm
0b10, 0b11	Reserved, do not use

Table 15 – POWER bit field encoding

8.5.8 ADV_INTERVAL

The register ADV_INTERVAL identifies the interval between consecutive advertising events. Figure 42 below shows the structure of the ADV_INTERVAL register.

ADV_INTERVAL (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU						INTERVAL	

Figure 42 – ADV_INTERVAL register

The encoding for the INTERVAL bit field is shown in Table 16 below.

INTERVAL	Advertising Interval
0b00 (Default)	20 ms
0b01	10 ms
0b10, 0b11	Reserved, do not use

Table 16 – INTERVAL bit field encoding

8.5.9 MANUFACTURER_ID

The register MANUFACTURER_ID identifies the manufacturer of the device using the 16 bit company identifier assigned by Bluetooth SIG. The default setting of 0x03DA identifies EnOcean GmbH as the manufacturer of the device.

At the time of writing, the list of assigned company identifiers could be found here: <https://www.bluetooth.com/specifications/assigned-numbers/company-identifiers/>

8.5.10 OPTIONAL_DATA

The register OPTIONAL_DATA identifies additional (up to 4 byte) of user-defined data that can be transmitted as part of each data telegram. This data can be used to convey additional information about the use of STM 550B and is user-defined.

The amount of OPTIONAL_DATA to be transmitted (0, 1, 2 or 4 byte) is determined by the OPTIONAL_DATA_SIZE register described below.



Before making changes to the OPTIONAL_DATA_SIZE and the REPORTING_CFG registers, you need to ensure the resulting payload length will not exceed 19 byte. See chapter 4.5.4.

8.5.11 OPTIONAL_DATA_SIZE

The OPTIONAL_DATA_SIZE register determines how many user-defined data bytes from the OPTIONAL_DATA register will be transmitted as part of data telegrams. Figure 46 below shows the structure of the OPTIONAL_DATA_SIZE register.

OPTIONAL_DATA_SIZE (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU						SIZE	

Figure 43 – OPTIONAL_DATA_SIZE register

The encoding used by the SIZE bit field is shown in Table 17 below.

SIZE	Optional Data size
0b00 (Default)	No optional data reported
0b01	1 byte optional data reported (Byte 0 of OPTIONAL_DATA)
0b10	2 byte optional data reported (Byte 0 and Byte 1 of OPTIONAL_DATA)
0b10	4 byte optional data reported (Byte 0 ... Byte 3 of OPTIONAL_DATA)

Table 17 – SIZE bit field encoding

8.5.12 SECURITY_KEY

The register SECURITY_KEY identifies the 16 byte security key used to authenticate the data telegrams of STM 550B.

Access to the SECURITY_KEY via this register or via its copy in the NDEF ore PUBLIC INFO area can be disabled by the user via the SECURITY_KEY_ACCESS register described below.

8.5.13 SECURITY_KEY_ACCESS

The register SECURITY_KEY_ACCESS allows protecting access to the device-unique security key via the SECURITY_KEY field of the NFC interface or via the transmission of a LRN telegram as described in chapter 6.1.

Figure 44 below shows the structure of the SECURITY_KEY_ACCESS register.

SECURITY_KEY_ACCESS (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU				LRN_TEL_ACCESS		NFC_ACCESS	

Figure 44 – SECURITY_KEY_ACCESS register

The encoding for the NFC_ACCESS bit field is shown in Table 18 below.

NFC_ACCESS	Security key access via NFC
0b00 (Default)	Public NFC Access: Security key visible in NFC, No NFC PIN required
0b01	Private NFC Access: Security key visible in NFC, NFC PIN required
0b10	No NFC Access: Security key not visible in NFC
0b11	Reserved, do not use

Table 18 – NFC_ACCESS bit field encoding

The encoding for the LRN_TEL_ACCESS bit field is shown in Table 19 below.

LRN_TEL_ACCESS	Security key access via LRN telegram
0b0 (Default)	LRN Telegram (containing security key) enabled
0b1	LRN Telegram (containing security key) disabled

Table 19 – LRN_TEL_ACCESS bit field encoding

8.5.14 SECURITY_CFG

The SECURITY_CFG register allows the configuration of the security parameters used by STM 550B. Figure 45 below shows the structure of the SECURITY_CFG register.

SECURITY_CFG (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU			RPA	SECURITY_MODE			

Figure 45 – SECURITY_CFG register

The encoding for the SECURITY_MODE bit field is shown in Table 20 below.

STM 550B currently always uses authentication with a 32 bit sequence counter generating a 32 bit message integrity code (MIC) based on a 128 bit random device-unique security key. Other security modes might be added in the future.

SECURITY_MODE	Security Mode
0b0000 (Default)	Authentication with 32 bit sequence counter and 32 bit MIC
Others	Reserved (do not use)

Table 20 – SECURITY_MODE bit field encoding

STM 550B uses by default a device-unique static source address (constant throughout the lifetime of the device) starting with 0xE500 as described in chapter 4.4.5.

STM 550B can alternatively use a resolvable private address (RPA) to obfuscate the origin of its data telegrams as described in chapter 4.4.6 and in Appendix C.

The selection between these two modes is done using the RPA bit field as shown in Table 21 below.

RPA	Address Mode
0b0 (Default)	Use Static Source Address
0b1	Use Resolvable Private Address (RPA)

Table 21 – RPA bit field encoding

8.5.15 REPORTING_CFG

The REPORTING_CFG register determines which items are transmitted within data telegrams. Figure 46 below shows the structure of the REPORTING_CFG register.

REPORTING_CFG (Default Setting: 0xDA)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MC SENSOR	ACC SENSOR	RFU	TRH SENSOR	LIGHT SENSOR	SOLAR CELL	ENERGY LEVEL	

Figure 46 – REPORTING_CFG register



Before making changes to the REPORTING_CFG and the OPTIONAL_DATA_SIZE registers, you need to ensure the resulting payload length will not exceed 19 byte. See chapter 4.5.4.

The ENERGY bit field allows selecting if the voltage of the backup battery and / or the energy level of the internal energy store containing the harvested energy is reported. The encoding for the ENERGY bit field is shown in Table 22 below.

ENERGY	Energy storage level / Backup battery voltage reporting	
0b00	None	No reporting
0b01	Adaptive	If backup battery present: Report backup battery voltage If backup battery not present: Report energy storage level
0b10 (Default)	Energy storage	Report energy storage level
0b11	Both	Report backup battery voltage and energy storage level

Table 22 – ENERGY bit field encoding

The SOLAR_CELL bit field determines if the illumination of the solar cell is reported within data telegrams. The encoding for the SOLAR_CELL bit field is shown in Table 23 below.

SOLAR_CELL	Solar cell illumination reporting
0b0 (Default)	Disabled: Do not report solar cell illumination
0b1	Enabled: Report solar cell illumination

Table 23 – SOLAR_CELL bit field encoding

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

The LIGHT SENSOR bit field determines if the illumination measured by the light sensor is reported within data telegrams. The encoding for the LIGHT SENSOR bit field is shown in Table 24 below.

LIGHT SENSOR	Illumination reporting
0b0	Disabled: Do not report illumination
0b1 (Default)	Enabled: Report illumination

Table 24 – LIGHT SENSOR bit field encoding

The TRH SENSOR bit field determines if the temperature and humidity measured by the temperature and humidity sensor are reported within data telegrams. The encoding for the TRH SENSOR bit field is shown in Table 25 below.

TRH SENSOR	Temperature and humidity reporting
0b0	Disabled: Do not report temperature and humidity
0b1 (Default)	Enabled: Report temperature and humidity

Table 25 – TRH SENSOR bit field encoding

The ACC SENSOR bit field determines if the acceleration measured by the temperature and humidity sensor are reported within data telegrams. The encoding for the ACC SENSOR bit field is shown in Table 26 below.

ACC SENSOR	Acceleration reporting
0b0	Disabled: Do not report acceleration
0b1 (Default)	Enabled: Report temperature and humidity

Table 26 – ACC SENSOR bit field encoding

The MC SENSOR bit field determines if the magnet contact status measured by the magnet contact sensor is reported within data telegrams. The encoding for the MC SENSOR bit field is shown in Table 27 below.

MC SENSOR	Magnet contact reporting
0b0	Disabled: Do not report acceleration
0b1 (Default)	Enabled: Report temperature and humidity

Table 27 – MC SENSOR bit field encoding

8.5.16 LED_MODE

The LED_MODE register determines the brightness of the LED. Figure 47 below shows the structure of the LED_MODE register.

LED_MODE (Default Setting: 0x01)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU						LED	

Figure 47 – LED_MODE register

The encoding used by the LED bit field is shown in Table 28 below.

LED	LED Intensity
0b00	LED is disabled
0b01 (Default)	LED is enabled
0b10, 0b11	Reserved (Do not use)

Table 28 – LED bit field encoding

8.5.17 FUNCTIONAL_MODE

The FUNCTIONAL_MODE register can be used to switch between standard operation mode and Standby (Sleep) mode as described in chapter 2.2.1.

Figure 48 below shows the structure of the FUNCTIONAL_MODE register.

FUNCTIONAL_MODE (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU				MODE			

Figure 48 – FUNCTIONAL_MODE register

The encoding used by the MODE bit field is shown in Table 29 below.

MODE	Functional Mode
0b0000 (Default)	Standard Operation Mode
0b0001 (00B)	Standby (Sleep) Mode
0b0010	Learn Mode
0b0011	Function Test Mode
0b0100	Illumination Test Mode
0b0111	Factory Reset
Others	Reserved (Do not use)

Table 29 – MODE bit field encoding



Out of the box, STM 550B is delivered in standby (sleep) mode to disable radio transmission during transport in line with air transport regulation. STM 550B will start operation upon pressing the LRN button or changing the mode using the MODE field.

After STM 550B started operation the first time, subsequent factory resets will result in STM 550B being in standard operation mode.

8.5.18 STANDARD_TX_INTERVAL

STM 550B uses a standard reporting interval which can be automatically adjusted based on sensor readings as described in chapter 2.4. The standard reporting interval is set by the register STANDARD_TX_INTERVAL shown in Figure 49 below.

STANDARD_TX_INTERVAL (Default Setting: 0x003C)				
Bit 15	Bit 14	...	Bit 1	Bit 0
STANDARD INTERVAL				

Figure 49 – STANDARD_TX_INTERVAL register

The encoding used by the STANDARD INTERVAL bit field is shown in Table 30 below.

STANDARD INTERVAL	Standard Reporting Interval
0x0000, 0x0001, 0x0002	Not supported (Do not use)
0x0003	3 seconds (minimum setting)
...	...
0x003C (Default)	60 seconds (default setting)
...	...
0xFFFF	65535 seconds (maximum setting)

Table 30 – STANDARD INTERVAL bit field encoding

8.5.19 THRESHOLD_CFG1

STM 550B can reduce the reporting interval based on the illumination of the solar cell, the light illumination of the ambient level sensor, the temperature or the humidity as described in chapter 2.4.

The use of these reduced reporting intervals is enabled by the THRESHOLD_CFG1 register shown in Figure 50 below.

THRESHOLD_CFG1 (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HUMIDITY SENSOR		TEMP SENSOR		LIGHT SENSOR		SOLAR CELL	

Figure 50 – THRESHOLD_CFG1 register

The encoding used by the SOLAR CELL bit field is shown in Table 31 below.

SOLAR CELL	Reporting interval reduction based on solar cell illumination
0b00 (Default)	Disabled (No reporting interval reduction)
0b01	Reserved (Do not use)
0b10	Enabled, Reporting interval reduction if light above threshold
0b11	Reserved (Do not use)

Table 31 – SOLAR CELL bit field encoding

The encoding used by the LIGHT SENSOR bit field is shown in Table 32 below.

LIGHT SENSOR	Reporting interval reduction based on light sensor illumination
0b00 (Default)	Disabled (No reporting interval reduction)
0b01	Reserved (Do not use)
0b10	Enabled, Reporting interval reduction if light above threshold
0b11	Reserved (Do not use)

Table 32 – LIGHT SENSOR bit field encoding

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

The encoding used by the TEMP SENSOR bit field is shown in Table 33 below.

TEMP SENSOR	Reporting interval reduction based on temperature
0b00 (Default)	Disabled (No reporting interval reduction)
0b01	Enabled, Reporting interval reduction if temperature below threshold
0b10	Enabled, Reporting interval reduction if temperature above threshold
0b11	Reserved (Do not use)

Table 33 – TEMP SENSOR bit field encoding

The encoding used by the HUMIDITY SENSOR bit field is shown in Table 34 below.

HUMIDITY SENSOR	Reporting interval reduction based on temperature
0b00 (Default)	Disabled (No reporting interval reduction)
0b01	Enabled, Reporting interval reduction if humidity below threshold
0b10	Enabled, Reporting interval reduction if humidity above threshold
0b11	Reserved (Do not use)

Table 34 – HUMIDITY SENSOR bit field encoding

8.5.20 THRESHOLD_CFG2

STM 550B can reduce the reporting interval based on the acceleration sensor status and the magnet contact status as described in chapter 2.4.

The use of these reduced reporting intervals is enabled by the THRESHOLD_CFG2 register shown in Figure 51 below.

THRESHOLD_CFG2 (Default Setting: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU				MC SENSOR		ACC SENSOR	

Figure 51 – THRESHOLD_CFG2 register

The encoding used by the ACC SENSOR bit field is shown in Table 35 below.

ACC SENSOR	Reporting interval reduction based on acceleration
0b00 (Default)	Disabled (No reporting interval reduction)
0b01	Reserved (Do not use)
0b10	Enabled, Reporting interval reduction if acceleration above threshold
0b11	Reserved (Do not use)

Table 35 – ACC SENSOR bit field encoding

The encoding used by the MC SENSOR bit field is shown in Table 36 below.

MC SENSOR	Reporting interval reduction based on magnet contact status
0b00 (Default)	Disabled (No reporting interval reduction)
0b01	Enabled, Reporting interval reduction if contact is open
0b10	Enabled, Reporting interval reduction if contact is closed
0b11	Reserved (Do not use)

Table 36 – MC SENSOR bit field encoding

8.5.21 ACC_SENSOR_CFG

The operation parameters of the acceleration sensor in STM 550B can be configured using the ACC_SENSOR_CFG registers shown in Figure 52 below. Refer to chapter 3.5 for a description of the acceleration sensor function.

ACC_SENSOR_CFG (Default Setting: 0x19)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU			WAKE	SAMPLING RATE		FULL SCALE	

Figure 52 – ACC_SENSOR_CFG register

The encoding used by the FULL SCALE bit field is shown in Table 35 below.

FULL SCALE	Full scale value of the acceleration sensor
0b00	+2 g
0b01 (Default)	+4 g
0b10	+8 g
0b11	+16 g

Table 37 – FULL SCALE bit field encoding

The encoding used by the SAMPLING RATE bit field is shown in Table 38 below.

SAMPLING RATE	Sampling rate of the acceleration sensor
0b00	1.6 Hz
0b01	12.5 Hz
0b10 (Default)	25 Hz
0b11	50 Hz

Table 38 – SAMPLING RATE bit field encoding

The wake-on-acceleration function can be enabled and disabled using the WAKE bit field as shown in Table 39 below.

WAKE	Wake-on-acceleration
0b0	Disabled
0b1 (Default)	Enabled

Table 39 – WAKE bit field encoding

8.5.22 SOLAR_THRESHOLD

If reduction of the reporting interval based on the solar cell light level has been enabled then the light level threshold is defined by SOLAR_THRESHOLD register as shown in Figure 53 below.

SOLAR_THRESHOLD (Default Setting: 0x00C8)				
Bit 15	Bit 14	...	Bit 1	Bit 0
SOLAR CELL THRESHOLD				

Figure 53 – SOLAR_THRESHOLD register

The encoding used by the SOLAR CELL THRESHOLD bit field is shown in Table 40 below.

SOLAR CELL THRESHOLD	Threshold
0x0000	0 lux (minimum setting)
...	...
0x00C8 (Default)	200 lux seconds (default setting)
...	...
0xFFFF	65535 lux (maximum setting)

Table 40 – SOLAR CELL THRESHOLD bit field encoding

The default setting for the solar cell threshold is 200 lux which corresponds to good availability of ambient light for harvesting.

8.5.23 SOLAR_TX_INTERVAL

If solar cell illumination-controlled reporting has been enabled and the solar cell illumination is above the defined threshold then the resulting reporting interval will be determined by the register SOLAR_TX_INTERVAL shown in Figure 54 below.

SOLAR_TX_INTERVAL (Default Setting: 0x003C)				
Bit 15	Bit 14	...	Bit 1	Bit 0
SOLAR CELL INTERVAL				

Figure 54 – SOLAR_TX_INTERVAL register

The encoding used by the SOLAR CELL INTERVAL bit field is shown in Table 41 below.

SOLAR CELL INTERVAL	Solar cell illumination-based reporting interval
0x0000, 0x0001, 0x0002	Not supported (Do not use)
0x0003	3 seconds (minimum setting)
...	...
0x003C (Default)	60 seconds (default setting)
...	...
0xFFFF	65535 seconds (maximum setting)

Table 41 – SOLAR CELL INTERVAL bit field encoding

The default reporting interval while solar cell illumination is above the solar cell illumination threshold is 60 seconds. This can be reduced according to user requirements keeping in mind the energy balance.

8.5.24 LIGHT_THRESHOLD

If reduction of the reporting interval based on the light level measured by the light sensor has been enabled then the light level threshold is defined by LIGHT_THRESHOLD register as shown in Figure 55 below.

LIGHT_THRESHOLD (Default Setting: 0x00C8)				
Bit 15	Bit 14	...	Bit 1	Bit 0
LIGHT SENSOR THRESHOLD				

Figure 55 – LIGHT_SENSOR_THRESHOLD register

The encoding used by the LIGHT SENSOR THRESHOLD bit field is shown in Table 42 below.

LIGHT SENSOR THRESHOLD	Threshold
0x0000	0 lux (minimum setting)
...	...
0x00C8 (Default)	200 Lux seconds (default setting)
...	...
0xFFFF	65535 lux (maximum setting)

Table 42 – LIGHT_SENSOR_THRESHOLD bit field encoding

The default setting for the light sensor threshold is 200 lux.

8.5.25 LIGHT_TX_INTERVAL

If solar cell illumination-controlled reporting has been enabled and the solar cell illumination is above the defined threshold then the resulting reporting interval will be determined by the register LIGHT_TX_INTERVAL shown in Figure 56 below.

LIGHT_TX_INTERVAL (Default Setting: 0x003C)				
Bit 15	Bit 14	...	Bit 1	Bit 0
LIGHT SENSOR INTERVAL				

Figure 56 – LIGHT_TX_INTERVAL register

The encoding used by the LIGHT SENSOR INTERVAL bit field is shown in Table 43 below.

LIGHT SENSOR INTERVAL	Light sensor illumination-based reporting interval
0x0000, 0x0001, 0x0002	Not supported (Do not use)
0x0003	3 seconds (minimum setting)
...	...
0x003C (Default)	60 seconds (default setting)
...	...
0xFFFF	65535 seconds (maximum setting)

Table 43 – LIGHT SENSOR INTERVAL bit field encoding

The default reporting interval while the light level measured by the light sensor is above the light sensor illumination threshold is 60 seconds. This can be reduced according to user requirements keeping in mind the energy balance.

8.5.26 ACCELERATION_THRESHOLD

If reduction of the reporting interval based on acceleration has been enabled then the acceleration threshold is defined by ACCELERATION_THRESHOLD register as shown in Figure 61 below.

Note that this threshold is also used for the wake-on-acceleration function as described in chapter 3.5.

ACCELERATION_THRESHOLD (Default Setting: 0x0001)				
Bit 15	Bit 14	...	Bit 1	Bit 0
ACCELERATION_THRESHOLD				

Figure 57 – ACCELERATION_THRESHOLD register

The encoding used by the ACCELERATION_THRESHOLD bit field is shown in Table 48 below.

ACCELERATION_THRESHOLD	Threshold
0b000000	Reserved (Do not use)
0b000001 (Default)	1/64 * Full scale
0b000010	2/64 * Full scale
...	
0b111110	62/64 * Full scale
0b111111	63/64 * Full scale

Table 44 – ACCELERATION_THRESHOLD bit field encoding

8.5.27 ACCELERATION_TX_INTERVAL

If acceleration-controlled reporting has been enabled and the acceleration exceeds the defined acceleration threshold then the resulting reporting interval will be determined by the register ACCELERATION_TX_INTERVAL shown in Figure 62 below.

ACCELERATION_TX_INTERVAL (Default Setting: 0x003C)				
Bit 15	Bit 14	...	Bit 1	Bit 0
ACCELERATION INTERVAL				

Figure 58 – ACCELERATION_TX_INTERVAL register

The encoding used by the ACCELERATION INTERVAL bit field is shown in Table 49 below.

ACCELERATION INTERVAL	Acceleration-based reporting interval
0x0000, 0x0001, 0x0002	Not supported (Do not use)
0x0003	3 seconds (minimum setting)
...	...
0x003C (Default)	60 seconds (default setting)
...	...
0xFFFF	65535 seconds (maximum setting)

Table 45 – ACCELERATION INTERVAL bit field encoding

The default reporting interval when the acceleration meets the defined threshold criteria is 60 seconds. This can be reduced according to user requirements keeping in mind the energy balance.

8.5.28 TEMPERATURE_THRESHOLD

If reduction of the reporting interval based on the temperature has been enabled then the temperature threshold is defined by TEMPERATURE_THRESHOLD register as shown in Figure 59 below.

TEMPERATURE_THRESHOLD (Default Setting: 0x0000)				
Bit 15	Bit 14	...	Bit 1	Bit 0
TEMPERATURE_THRESHOLD				

Figure 59 – TEMPERATURE_THRESHOLD register

The encoding used by the TEMPERATURE_THRESHOLD bit field is shown in Table 46 below.

TEMPERATURE_THRESHOLD	Threshold
0x0000 (Default)	0.0°C
0x0001	0.1°C
...	
0x028A	65.0°C
Reserved	
0xFF38	-20.0°C
...	
0xFFFF	-0.1°C

Table 46 – TEMPERATURE_THRESHOLD bit field encoding

8.5.29 TEMPERATURE_TX_INTERVAL

If temperature-controlled reporting has been enabled and the temperature meets the defined threshold criteria then the resulting reporting interval will be determined by the register TEMPERATURE_TX_INTERVAL shown in Figure 60 below.

TEMPERATURE_TX_INTERVAL (Default Setting: 0x003C)				
Bit 15	Bit 14	...	Bit 1	Bit 0
TEMPERATURE INTERVAL				

Figure 60 – TEMPERATURE_TX_INTERVAL register

The encoding used by the TEMPERATURE INTERVAL bit field is shown in Table 47 below.

TEMPERATURE INTERVAL	Temperature-based reporting interval
0x0000, 0x0001, 0x0002	Not supported (Do not use)
0x0003	3 seconds (minimum setting)
...	...
0x003C (Default)	60 seconds (default setting)
...	...
0xFFFF	65535 seconds (maximum setting)

Table 47 – TEMPERATURE INTERVAL bit field encoding

The default reporting interval when the temperature meets the defined threshold criteria is 60 seconds. This can be reduced according to user requirements keeping in mind the energy balance.

8.5.30 HUMIDITY_THRESHOLD

If reduction of the reporting interval based on the humidity has been enabled then the humidity threshold is defined by HUMIDITY_THRESHOLD register as shown in Figure 61 below.

HUMIDITY_THRESHOLD (Default Setting: 0x0000)				
Bit 15	Bit 14	...	Bit 1	Bit 0
HUMIDITY_THRESHOLD				

Figure 61 – HUMIDITY_THRESHOLD register

The encoding used by the HUMIDITY_THRESHOLD bit field is shown in Table 48 below.

HUMIDITY_THRESHOLD	Threshold
0x0000 (Default)	0.0 % r.h.
0x0001	0.5 % r.h.
...	
0x00C7	99.5 %r.h.
0x00C8	100.0 %r.h.
Others	Reserved (Do not use)

Table 48 – HUMIDITY_THRESHOLD bit field encoding

8.5.31 HUMIDITY_TX_INTERVAL

If humidity-controlled reporting has been enabled and the humidity meets the defined threshold criteria then the resulting reporting interval will be determined by the register HUMIDITY_TX_INTERVAL shown in Figure 62 below.

HUMIDITY_TX_INTERVAL (Default Setting: 0x003C)				
Bit 15	Bit 14	...	Bit 1	Bit 0
HUMIDITY INTERVAL				

Figure 62 – HUMIDITY_TX_INTERVAL register

The encoding used by the HUMIDITY INTERVAL bit field is shown in Table 49 below.

HUMIDITY INTERVAL	Humidity-based reporting interval
0x0000, 0x0001, 0x0002	Not supported (Do not use)
0x0003	3 seconds (minimum setting)
...	...
0x003C (Default)	60 seconds (default setting)
...	...
0xFFFF	65535 seconds (maximum setting)

Table 49 – HUMIDITY INTERVAL bit field encoding

The default reporting interval when the humidity meets the defined threshold criteria is 60 seconds. This can be reduced according to user requirements keeping in mind the energy balance.

8.5.32 MAGNET_CONTACT_TX_INTERVAL

If magnet contact-controlled reporting has been enabled and the magnet contact state matches the state configured in the THRESHOLD_CFG2 register then the resulting reporting interval will be determined by the register MAGNET_CONTACT_TX_INTERVAL shown in Figure 63 below.

MAGNET_CONTACT_TX_INTERVAL (Default Setting: 0x003C)				
Bit 15	Bit 14	...	Bit 1	Bit 0
MAGNET CONTACT INTERVAL				

Figure 63 – MAGNET_CONTACT_TX_INTERVAL register

The encoding used by the MAGNET CONTACT INTERVAL bit field is shown in Table 49 below.

MAGNET CONTACT INTERVAL	Magnet contact-based reporting interval
0x0000, 0x0001, 0x0002	Not supported (Do not use)
0x0003	3 seconds (minimum setting)
...	...
0x003C (Default)	60 seconds (default setting)
...	...
0xFFFF	65535 seconds (maximum setting)

Table 50 – MAGNET CONTACT INTERVAL bit field encoding

The default reporting interval when the magnet contact state meets the defined state is 60 seconds. This can be reduced according to user requirements keeping in mind the energy balance.

8.5.33 ILLUMINATION_TEST_RESULT

If STM 550B has executed an illumination test as described in chapter 2.3.5 then the result (the measured average light level at the solar cell) will be stored in the ILLUMINATION_TEST_RESULT register shown in Figure 64 below.

ILLUMINATION_TEST_RESULT (Default Setting: 0x0000)				
Bit 15	Bit 14	...	Bit 1	Bit 0
ILLUMINATION TEST RESULT				

Figure 64 – ILLUMINATION_TEST_RESULT register

8.6 USER DATA

The USER DATA area allows the user to read and write up to 64 byte of data after entering the correct PIN code. Typical use cases include storing information about the configuration or the installation of the device (by whom, when, what). STM 550B does not use this area and does not interpret its content in any way.

9 Installation recommendations

9.1 Setup instructions

Before installing STM 550B into its intended location, a sufficient initial charge should be provided to STM 550B and its correct operation should be verified.

To do so, follow these steps:

1. Place STM 550B under bright light (daylight or bright light source) for 5 minutes to provide an initial charge
2. Press the LRN button once so that STM 550B will start operation
3. Check that STM 550B transmits radio telegrams at the configured update interval (by default once every 60 seconds). The LED will blink every time a telegram is transmitted (unless this has been disabled via NFC).
4. Use a suitable receiver to capture the STM 550B data telegrams and verify that all required parameters are reported. Consider disabling the measurement and reporting of non-required parameters to conserve energy.
5. Check the light level reported by STM 550B at the intended installation location to verify that sufficient light is available for the energy harvesting functionality. Maximize the amount of light available for energy harvesting as much as possible.
6. Make sure that the installation location is chosen according to the guidelines in the subsequent chapters to maximize the measurement accuracy.

After those steps, STM 550B is ready for installation into its intended location.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

9.2 Acceleration detection

Acceleration detection enables to track the utilization of an asset (is a machine running, is a chair occupied, ...) based on the characteristic vibrations associated with this utilization.

Acceleration sensors – unlike gyroscopic or compass sensors – work based on changes of the acceleration vector. When an acceleration sensor (i.e. STM 550B) is at rest then the acceleration vector is the gravity vector of the earth. Changes in asset orientation (e.g. measuring the tilt of a window to distinguish between open and closed state) can therefore only be detected if the orientation of STM 550B relative to the gravity vector changes.

Figure 65 below illustrates two cases:

- The left case (window tilt) could be detected if STM 550B is attached to the window part that is tilted since the orientation of STM 550B relative to the gravity vector would change
- The right case (window open / rotation) could not be detected if STM 550B is attached to the window part that is rotated since the orientation of STM 550B relative to the gravity vector would remain the same.

The surface towards which STM 550B is attached should therefore be chosen to maximize orientation changes relative to the gravity vector.

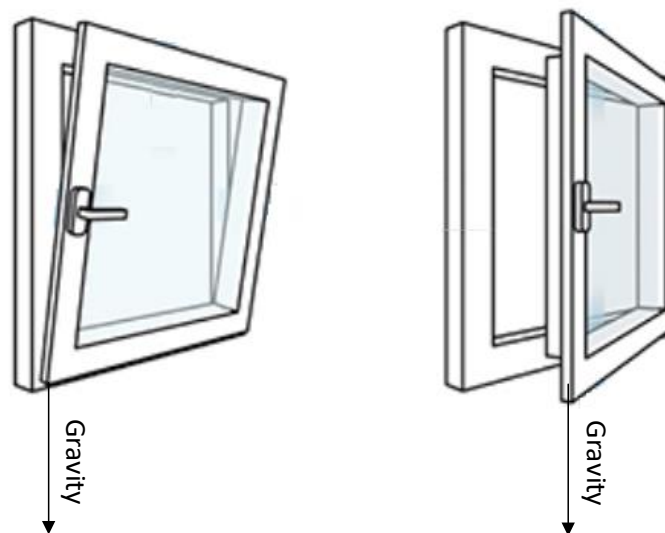


Figure 65 – Tilt versus gravity vector

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

In addition to that, the following points should be considered to maximize the reliability of utilization tracking:

- STM 550B should be firmly attached to the asset without any damping to ensure that any vibration of the asset will be properly propagated to STM 550B
- STM 550B should be attached to the asset at the location where the vibration is maximized. For instance, when tracking the utilization of office chairs, the highest acceleration is typically observed at the back rest.
- The acceleration threshold for wake-on-acceleration should be selected such that utilization / motion is reliably detected without false triggers due to spurious vibration (e.g. people walking by)
- Should the default sensitivity be insufficient even at the lowest threshold then the sampling rate should be increased

9.3 Temperature and humidity measurement

The dedicated temperature and humidity sensor integrated into STM 550B accurately measures and reports temperature and humidity present at its surface. To achieve the best possible accuracy, it is important to consider the following points:

- Installation height
The sensor should be installed at a height that is representative for the use case. For the case of an office, the sensor should be mounted at desk level.
- Sun light or heat exposure
The sensor should be mounted such that it is not directly exposed to sunlight or heat (e.g. close to a radiator)
- Disturbances
The sensor should be mounted such that the influence from disturbances such as the air stream from air condition units is minimized. Consider also the possible temperature gradient between wall and room when mounting the sensor directly onto a wall.
- Air flow
The sensor should be mounted such that the airflow from the target measurement area towards the air inlets is maximized. This will ensure the lowest possible response time of the sensor. Avoid mounting the sensor in niches or slots with little air flow.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

9.4 Illumination measurement

STM 550B offers the option to measure the ambient light level either via the ambient light sensor or via the solar cell. This can be configured using the REPORTING_CFG register of the NFC interface as described in chapter 8.5.15. By default, the ambient light sensor is used.

9.4.1 Ambient light sensor

The ambient light sensor measures and reports the light level with a spectral response close to the human eye's perception of ambient light. The following points should be considered when using the ambient light sensor:

- **Aperture**
The sensor measures the light level within a small radius around its centre axis. If the lighting conditions within that area are not representative for the overall conditions, then the result might be different from expectation.
- **Surface**
The most common application for a ceiling-mounted illumination sensor is to measure the light level at a working desk surface underneath. In this application, the measured light level depends on the reflectivity of the surface. Simply put, a dark desk surface will give a totally different result compared to a white desk surface even when the same luminous flow is directed towards it.
- **Obstruction**
Any obstruction between the sensor and the intended measurement area (desk surface, window) will significantly impact the measurement result. Maintaining a clear line of sight between measurement area and illuminations sensor is therefore essential.
- **Interference**
To ensure accurate measurement results, it is essential to minimize interference from other light sources not contributing to the illumination at the target measurement area. For instance, when measuring the light level at a desk surface, interference might occur due to direct light from the window or from or upwards emission of indirect light sources (floor lamps etc)

9.4.2 Solar cell

The solar cell has a much larger area and aperture compared to the ambient light sensor. Therefore, the light level measured by the solar cell is typically more representative of the average illumination within a wider area.

Note that the solar cell does not apply a spectral response curve close to the human eye's perception of ambient light to the received illumination. The illumination reported by the solar cell will therefore typically be larger than that reported by the ambient light sensor depending on the spectral properties of the ambient light.

Calibration at the receiver is suggested to obtain best results for the given lighting situation.

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

9.5 Magnet contact sensing

If STM 550B is used to detect the presence of a magnet using its magnet contact sensor (e.g. for door or window monitoring), then the magnet has to be in close proximity to the magnet contact sensor for the case where a "Magnet Present" (or "Closed") condition shall be detected. Refer to Figure 2 and Figure 3 for the location of the magnet contact sensor.

Attach the magnet to the intended surface (e.g. door or window) such that the centre of the magnet is in line with the location of the magnet contact sensor and that the distance between magnet and sensor housing is less than 1 cm. Verify that the state (e.g. door open or closed) is reported as expected.

9.6 Energy harvesting

STM 550B is powered by ambient light using its integrated solar cell. For best performance it is therefore essential to maximize the amount of light available for harvesting.

Harvestable light will typically be either natural light (daylight coming in through windows etc) or artificial light (direct or reflected light from indoor luminaires). If natural light is available (e.g. from a window) then the solar cell of STM 550B should be oriented as much as possible towards that.

STM 550B is designed to operate self-supplied with its standard parameters based on 200 lux of illumination at its solar cell for at least 6 hours per day. The exact amount of available light can be determined by executing an illumination test as described in chapter 2.3.5.

Lower levels of available light can be addressed by configuring a lower reporting rate via NFC as discussed in chapter 2.4.1. If the available light is insufficient, then STM 550B offers the option for a CR1632 backup battery as described in chapter 2.2.

9.7 NFC configuration

STM 550B can be flexibly configured for a wide range of application scenarios using the NFC configuration interface as described in chapters 7 and 8.

Before making any configuration changes, be sure to familiarize yourself with the device functionality and determine the energy constraints based on the available ambient light as discussed in chapter 2.4.1. Be especially careful not to configure higher update rates (low reporting intervals) before ensuring that sufficient light is available.

Should you be unsure about the current NFC configuration, then execute a factory reset as described in chapter 2.3.5 to reset all configuration registers to their default setting.

After completing the NFC configuration and ensuring that all functionality works as required, it is recommended to lock the NFC configuration interface by changing the NFC PIN code from its default value to a different (secret) value. Make sure the new PIN code is properly noted down.

10 Product history

Table 51 below lists the product history of EMSIB.

Revision	Release	Key changes versus previous revision
CA-02	Jun 2019	First prototypes for lead customer evaluation
CA-03	Jan 2020	Addition of adaptive reporting Addition of LRN, Function test, Illumination and Factory Reset mode New NFC configuration layout

Table 51 – Product History

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

A Telegram parsing

This chapter provides examples of STM 550B telegrams and shows how to interpret them.

A.1 Data telegram example

We consider the following raw data (excluding CRC) captured from an actual STM 550B module:

```
D6 BE 89 8E 42 23 00 00 01 77 00 E5 1C FF DA 03 FB 27 00 00 02 68 45 8D 01 40 42 09
06 57 8A F7 91 E6 5E 23 01 0F 01 57 D3
```

A.1.1 BLE advertising frame structure

The message above can be parsed according to the Bluetooth standard for advertising messages as shown below.

Field	Length	Data	Interpretation
BLE Access Address	4 byte	0x8E89BED6	Constant (always used)
BLE Frame Control	2 byte	0x2342	Length = 35 byte
BLE Source Address	6 byte	0xE50077010000	Device-unique address
Length of payload	1 byte	0x1C	28 byte of payload follow
Type of payload	1 byte	0xFF	Manufacturer-specific data
Manufacturer ID	2 byte	0x03DA	EnOcean GmbH
Payload	25 byte	FB 27 00 00 02 68 45 8D 01 40 42 09 06 57 8A F7 91 E6 5E 23 01 0F 01 57 D3	

Table 52 – Advertising message parsing

A.1.2 Data telegram payload

The EnOcean payload can be parsed as shown below.

Field	Length	Data	Interpretation
Sequence Counter	4 byte	0x000027FB	Incrementing message counter
Sensor Data	17 byte	02 68 45 8D 01 40 42 09 06 57 8A F7 91 E6 5E 23 01	
Telegram Signature	4 byte	0x0F0157D3	Authentication signature

Table 53 – EnOcean payload parsing

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

A.1.3 Sensor data

The sensor data can be parsed as shown below.

Descr	Length	Type	Data	Value
0x02	0b00 (8 bit)	0b000010 (Energy Level)	0x68	52 %
0x45	0b01 (16 bit)	0b000101 (Sensor Illuminance)	0x018D	397 lx
0x40	0b01 (16 bit)	0b000000 (Temperature)	0x0942	23.70°C
0x06	0b00 (8 bit)	0b000110 (Relative humidity)	0x57	43.5 % r.h.
0x8A	0b10 (32 bit)	0b001010 (Acceleration vector)	0x5EE691F7 0b01 0b0111101110 0b0110100100 0b011110111	Periodic Update X = -0.18g Y = -0.92g Z = -0.09g
0x23	0b00 (8 bit)	0b100011 (Contact Sensor)	0x01	Open

Table 54 – Sensor data parsing

The acceleration data of 0x5EE691F7 can be parsed as follows:

Binary representation: 0b01011110111001101001000111110111

```

Status   Data && 0xC0000000 >> 30 = 0b01           -> Periodic Update
X        Data && 0x3FF00000 >> 20 = 0b0111101110 = (494 - 512) / 100 = -0.18g
Y        Data && 0x000FFC00 >> 10 = 0b0110100100 = (420 - 512) / 100 = -0.92g
Z        Data && 0x000003FF           = 0b011110111 = (503 - 512) / 100 = -0.09g
    
```


STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

A.2 Commissioning telegram example

We consider the following raw data (excluding CRC) captured from the same STM 550B module as in the previous chapter:

```
D6 BE 89 8E 42 25 00 00 01 77 00 E5 1E FF DA 03 E4 2D 00 00 3E 4D B3 4D B3 07 0E FC
67 13 FE 39 E1 3C F3 C9 39 00 00 01 77 00 E5
```

A.2.1 BLE advertising frame

The message above can be parsed according to the Bluetooth standard for advertising messages as shown below.

Field	Length	Data	Interpretation
BLE Access Address	4 byte	0x8E89BED6	Constant (always used)
BLE Frame Control	2 byte	0x2542	Length = 37 byte
BLE Source Address	6 byte	0xE50077010000	Device-unique address
Length of payload	1 byte	0x1E	30 byte of payload follow
Type of payload	1 byte	0xFF	Manufacturer-specific data
Manufacturer ID	2 byte	0x03DA	EnOcean GmbH
Payload	27 byte	E4 2D 00 00 3E 4D B3 4D B3 07 0E FC 67 13 FE 39 E1 3C F3 C9 39 00 00 01 77 00 E5	

Table 55 – Advertising message parsing

A.2.2 Commissioning telegram payload

The EnOcean payload can be parsed as shown below.

Field	Length	Data	Interpretation
Sequence Counter	4 byte	00002DE4	Incrementing message counter
Field Identifier	1 byte	0x3E	Commissioning Telegram (22 byte)
Device Key	16 byte	4DB34DB3070EFC6713FE39E13CF3C939	
Source Address	6 byte	0xE50077010000	

Table 56 – EnOcean commissioning telegram payload parsing

B Authentication

B.1 Input data

For this example, we consider the following input data:

Parameter	Example
Input data (Message to be authenticated)	1CFFDA03FB2700000268458D0140420906578AF791E65E2301
Source address (Message sender)	E50077010000
Security key	4DB34DB3070EFC6713FE39E13CF3C939
Authentication signature to be verified	0F0157D3

Table 57 – Input data

From the data telegram and the commissioning telegram, we can derive the following parameters required for the authentication algorithm:

Parameter	Example
Source Address (little endian format)	0000017700E5
Length of input data	0x0019 (25 byte)
Sequence Counter (little endian format)	FB270000

Table 58 – Derived input data

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

B.2 Constant algorithm parameters

The RFC3610 implementation requires two constant algorithm parameters:

- Length field size (in byte)
This is the size of the field used to encode the length of the input data (which is the payload to be authenticated).
The maximum size of sensor payload to be authenticated is 31 byte; 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 (in byte)
The desired signature size is 4 byte for sensor data telegrams

Additionally, the RFC3610 implementation derives two algorithm parameters – M' and L' – based on the constant algorithm parameters and uses them to construct $A0_Flag$ and B_0_Flag which – together with the iteration counter i – are required for subsequent processing.

The value of all these internal parameters - listed in Table 59 below - is the same for all EnOcean BLE telegrams.

Parameter	Comment / Description	Value
Length Field Size	Size (in bytes) of the field used to encode the input length	2
Output Length	Desired signature size (in byte)	4
M'	Binary encoded output length: $M' = (\text{Output Length} / 2) - 1$	0b001
L'	Binary encoded length field size: $L' = \text{Length Field Size} - 1$	0b001
$A0_Flag$	L'	0x01
$B0_Flag$	$(0b01 \ll 6) + (M' \ll 3) + L'$	0x49
I	Iteration counter	0x0000

Table 59 – Constant algorithm parameters

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

B.3 Intermediate parameters

The RFC3610 implementation used in EnOcean BLE products derives four internal parameters – Nonce, A0, B0, B1 and B2 – based on the telegram specific input data and the constant internal parameters.

These variable internal parameters are described in Table 60 below. The values of these parameters are calculated based on the input data given in chapter B.1.

Parameter	Comment / Description	Value in the example
Nonce	13 byte initialization vector based on concatenation of 6 byte little endian source address, 4 byte little endian sequence counter and 3 byte 0x00 padding	0000017700E5FB270000000000
A0	A0_Flag followed by Nonce followed by 2 byte 0x00	010000017700E5FB2700000000000000
B0	B0_Flag followed by Nonce followed by 2 byte 0x00 (no message to encode)	490000017700E5FB2700000000000000
B1	Input Length followed by first 14 byte of Input Data	00191CFFDA03FB2700000268458D0140
B2	Remaining Input Data (up to 16 byte) with 0x00 padding to reach 16 byte in total	420906578AF791E65E2301000000000

Table 60 – Intermediate parameters

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

B.4 RFC3610 execution sequence

The RFC3610 algorithm uses the variable internal parameters A_0 , B_0 , B_1 and B_2 together with the private key to generate the authentication vector T_0 using four AES-128 and three XOR operations. The algorithm execution sequence is shown in Figure 66 below.

The first four bytes of T_0 are then used to authenticate EnOcean BLE multi-sensor data telegrams.

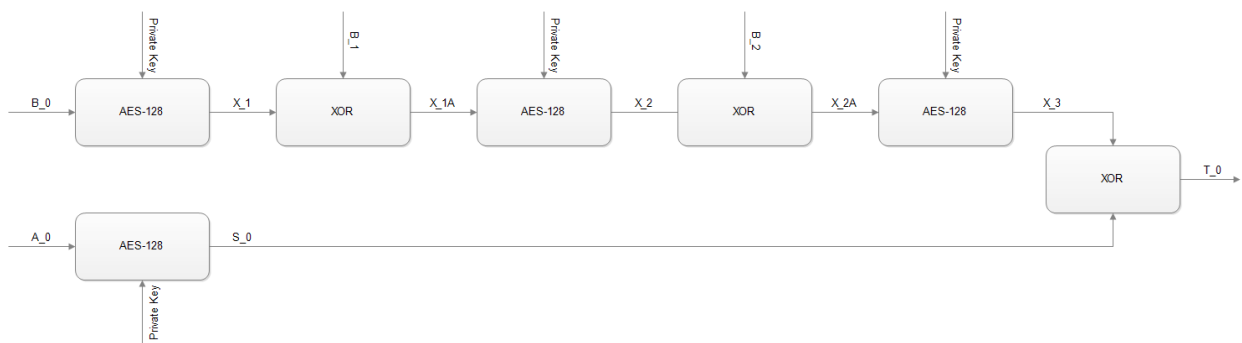


Figure 66 – RFC3610 execution sequence

STM 550B / EMSIB - EASYFIT MULTISENSOR FOR IOT APPLICATIONS (2.4 GHZ BLE)

B.5 Execution example

At the time of writing, a suitable online AES calculator could be found here: <http://testpro-tect.com/appendix/AEScalc>

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

We can now calculate the signature using a sequence of AES128 and XOR operations as shown in Figure 66 as follows:

```

X_1 = AES128(B0, Key)
X_1 = AES128(490000017700E5FB270000000000000, 4DB34DB3070EFC6713FE39E13CF3C939)
X_1 = F900DE65FE42BA0949F4D53B63DF8CA1

X_1A = XOR(X_1, B_1)
X_1A = XOR(F900DE65FE42BA0949F4D53B63DF8CA1, 00191CFFDA03FB2700000268458D0140)
X_1A = F919C29A2441412E49F4D75326528DE1

X_2 = AES128(X_1A, Key)
X_2 = AES128(F919C29A2441412E49F4D75326528DE1, 4DB34DB3070EFC6713FE39E13CF3C939 )
X_2 = 779D55AA355662EF2512B818FF01E9E1

X_2A = XOR(X_2, B_2)
X_2A = XOR(779D55AA355662EF2512B818FF01E9E1, 420906578AF791E65E2301000000000)
X_2A = 359453FDBFA1F3097B31B918FF01E9E1

X_3 = AES128(X_2A, Key)
X_3 = AES128(359453FDBFA1F3097B31B918FF01E9E1, 4DB34DB3070EFC6713FE39E13CF3C939)
X_3 = 88C57785525A2DF4EEFDA39DAEAA08EA

S_0 = AES128(A0, Key)
S_0 = AES128(010000017700E5FB270000000000000, 4DB34DB3070EFC6713FE39E13CF3C939)
S_0 = 87C4205662899442400842515B704DED

T_0 = XOR(X_3, S_0)
T_0 = XOR(88C57785525A2DF4EEFDA39DAEAA08EA, 87C4205662899442400842515B704DED)
T_0 = 0F0157D330D3B9B6AEF5E1CCF5DA4507

```

The calculated signature is formed by the first four bytes of T₀, i.e. it is 0F0157D3.

The calculated signature matches the signature that was transmitted as part of the data telegram payload (see chapter 4.5).

This proves that the telegram originates from a sender that possesses the same security key and the telegram content has not been modified.

C Address resolution for resolvable private addresses (RPA)

STM 550B provides the option to obfuscate its identity by means of using resolvable private addresses (RPA) as described in chapter 4.4.6. The following chapters describe how to resolve such addresses.

C.1 RPA resolution flow

The execution flow for resolving private addresses (RPA) is shown in Figure 67 below.

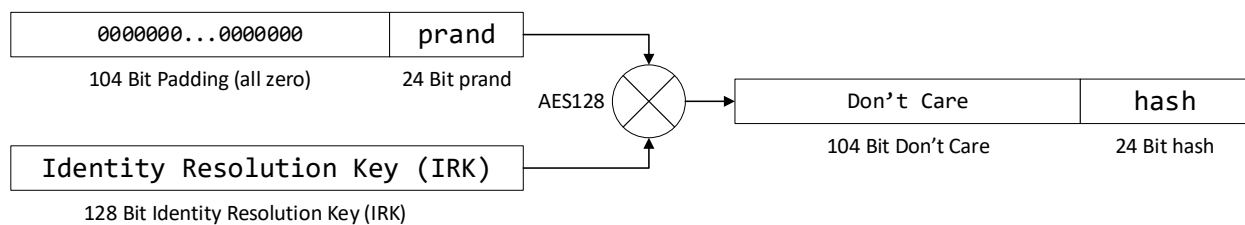


Figure 67 – Execution flow for resolving private addresses (RPA resolution)

The input to the RPA resolution flow is the prand part of the resolvable private address field of the received telegram together with one (or several) locally stored IRK.

The receiver will then try for each locally stored IRK if the hash generated using the execution flow above matches the hash part of the resolvable private address field of the received telegram. If it does match then the IRK identifies the device from which this telegram originated.

