

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## EnOcean Multi-Sensor For IoT Applications

### **STM 550 / EMSI**

19.02.2020



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 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**REVISION HISTORY**

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

Version	Author	Reviewer	Date	Major Changes
1.0	MKA	TM, EM, MH, MF	19.02.2020	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 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### TABLE OF CONTENT

1	General description.....	6
1.1	Basic functionality .....	6
1.2	Product variants .....	7
1.3	Technical data.....	7
1.4	Environmental conditions .....	8
1.5	Packaging information.....	8
1.5.1	STM 550 .....	8
1.5.2	EMSI .....	8
1.6	Ordering information .....	8
2	Functional overview .....	9
2.1	Product description .....	9
2.2	Product interface .....	9
2.3	Product label.....	10
2.4	Functional modes .....	11
2.4.1	Standard operation mode .....	11
2.4.2	Standby (Sleep) mode.....	12
2.4.3	Learn mode.....	12
2.4.4	Function test mode .....	12
2.4.5	Illumination test mode .....	14
2.4.6	Factory reset mode .....	14
2.5	Reporting interval.....	15
2.5.1	Energy considerations .....	15
2.5.2	Standard reporting interval .....	16
2.5.3	Illumination-controlled reporting interval .....	17
2.5.4	Temperature-controlled reporting interval .....	18
2.5.5	Humidity-controlled reporting interval .....	19
2.5.6	Acceleration-controlled reporting interval .....	20
2.5.7	Magnet contact sensor-controlled reporting interval .....	21
2.5.8	Arbitration between reporting intervals.....	21
3	Sensor functionality .....	22
3.1	Light level sensor .....	22
3.2	Solar cell .....	22
3.3	Temperature sensor.....	23
3.4	Humidity sensor .....	23
3.5	Acceleration sensor.....	24
3.5.1	Wake on acceleration .....	25
3.5.2	Acceleration sensor parameters .....	25
3.6	Magnet contact sensor .....	26
4	User interface .....	27
4.1	LRN button and LED .....	27
4.2	Backup battery .....	28
5	Radio communication.....	29
5.1	Radio Frame format.....	29

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

5.1.1	ERP1 frame format .....	29
5.1.2	ERP2 frame format .....	29
5.2	EnOcean Equipment Profiles (EEP) and SIGNAL telegrams .....	30
5.2.1	STM 550 supported EEP.....	30
5.2.2	EEP structure .....	31
5.2.3	1BS telegram .....	32
5.2.4	4BS telegram .....	32
5.2.5	VLD telegram .....	32
5.2.6	UTE (Universal Teach-in) telegram.....	32
5.2.7	Signal telegram .....	33
6	Security .....	34
6.1	Basic concepts .....	34
6.1.1	Telegram encryption .....	34
6.1.2	Telegram authentication .....	35
6.1.3	Dynamic security key modification .....	36
6.2	Security parameters .....	37
6.2.1	EURID .....	37
6.2.2	Security key .....	37
6.2.3	Rolling code .....	37
6.2.4	Security algorithm .....	37
6.3	STM 550 security implementation .....	38
7	Commissioning.....	39
7.1	Radio-based commissioning .....	40
7.2	QR code commissioning .....	40
7.3	Commissioning via NFC interface.....	40
8	NFC interface .....	41
8.1	NFC interface parameters .....	41
8.2	NFC access protection .....	41
8.3	Using the NFC interface.....	42
8.3.1	PC with dedicated NFC reader .....	42
8.3.2	Android or iOS smartphone with NFC.....	42
8.4	NFC interface functions .....	43
8.4.1	NFC interface state machine.....	43
8.4.2	IDLE state.....	44
8.4.3	READY 1 state .....	44
8.4.4	READY 2 state .....	44
8.4.5	ACTIVE state.....	44
8.4.6	Read command .....	45
8.4.7	Write command .....	45
8.4.8	Password authentication (PWD_AUTH) command .....	46
9	NFC registers .....	47
9.1	NFC memory areas .....	47
9.2	Device identification NDEF .....	48
9.3	User information NDEF.....	48
9.4	NFC HEADER.....	49
9.4.1	NFC HEADER area structure .....	49

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

9.5	CONFIGURATION.....	50
9.5.1	Using the NFC configuration functionality.....	50
9.5.2	CONFIGURATION area structure .....	50
9.5.3	NFC_PIN_CODE .....	51
9.5.4	PRODUCT_ID .....	51
9.5.5	USER_KEY .....	52
9.5.6	SECURITY_KEY_MODE.....	52
9.5.7	SECURITY_MODE.....	53
9.5.8	EEP .....	54
9.5.9	SIGNAL .....	55
9.5.10	LED_MODE .....	56
9.5.11	FUNCTIONAL_MODE.....	57
9.5.12	STANDARD_TX_INTERVAL .....	58
9.5.13	THRESHOLD_CFG1.....	59
9.5.14	THRESHOLD_CFG2.....	61
9.5.15	LIGHT_SENSOR_CFG .....	62
9.5.16	ACC_SENSOR_CFG .....	63
9.5.17	SOLAR_THRESHOLD.....	64
9.5.18	SOLAR_TX_INTERVAL .....	65
9.5.19	LIGHT_THRESHOLD .....	66
9.5.20	LIGHT_TX_INTERVAL .....	67
9.5.21	ACCELERATION_THRESHOLD .....	68
9.5.22	ACCELERATION_TX_INTERVAL .....	69
9.5.23	TEMPERATURE_THRESHOLD .....	70
9.5.24	TEMPERATURE_TX_INTERVAL .....	71
9.5.25	HUMIDITY_THRESHOLD.....	72
9.5.26	HUMIDITY_TX_INTERVAL.....	73
9.5.27	MAGNET_CONTACT_TX_INTERVAL.....	74
9.5.28	ILLUMINATION_TEST_RESULT.....	75
9.6	USER DATA .....	75
10	Installation recommendations .....	76
10.1	Setup instructions .....	76
10.2	Acceleration detection.....	77
10.3	Temperature and humidity measurement .....	78
10.4	Illumination measurement.....	79
10.4.1	Ambient light sensor .....	79
10.4.2	Solar cell .....	79
10.5	Magnet contact sensing.....	80
10.6	Energy harvesting .....	80
10.7	NFC configuration.....	81
11	Product history.....	82

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

### 1 General description

This user manual describes the functionality of the family of multisensor modules STM 550 and the family of Easyfit multisensor products EMSI. An EMSI Easyfit multisensor combines an STM 550 module with a wall mount, a cover frame, a magnet and an adhesive mounting tape into a ready to use product.

The STM 550 / EMSI multisensor family consists of the following members:

- STM 550 and EMSIA using 868.3 MHz radio (main market Europe)
- STM 550U and EMSIU using 902.875 MHz radio (main market US / Canada)
- STM 550J and EMSIJ using 928.35 MHz radio (main market Japan)

The term “STM 550” is used throughout this user manual to describe both EMSI Easyfit multisensors and STM 550 multisensor modules in all frequency variants unless otherwise noted.

#### 1.1 Basic functionality

STM 550 is a flexible self-powered multi-sensor module family 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 EnOcean radio standard.

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

STM 550 implements the following sensors:

- Temperature
- Humidity
- Illumination
- Acceleration
- Magnet contact

STM 550 will report periodically (by default approximately every 60 seconds, configurable via NFC) the latest measurements of these sensors. In addition, STM 550 can also report its internal energy level and the amount of light available at the solar cell.

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

Radio telegrams transmitted by STM 550 can be encrypted and authenticated using AES-128 security based on a device-unique private key and a sequence counter in accordance to the EnOcean Alliance Security Specification. This ensures integrity, confidentiality and authenticity of the transmitted telegrams and prevents telegram replay (retransmission of previously transmitted telegrams).

STM 550 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 550 provides the option to mount a CR1632 backup battery.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 1.2 Product variants

STM 550 is the functional module within EMSI using the PTM 21x form factor intended to be integrated into an OEM housing (e.g. a switch frame). EMSI is a finished product combining an STM 550 module with a wall mount, a design frame, a magnet (for magnet sensor functionality) and an adhesive mounting tape into a ready to use product.

STM 550 and EMSI are shown in Figure 1 below.



Figure 1 – STM 550 and EMSI

### 1.3 Technical data

<b>Antenna</b>	Integrated helix antenna
<b>Transmission frequency / power</b>	STM 550 / EMSIA: 868.300 MHz / +5 dBm STM 550U / EMSIU: 902.875 MHz / + 99 dBμV STM 550J / EMSIJ: 928.350 MHz / 0 dBm
<b>Transmission data rate</b>	125 kbit / s
<b>Communication range (for guidance only)</b>	200 m free field 30 m indoor environment
<b>Temperature measurement range / accuracy</b>	-20 °C ... +60 °C / +- 0.3 K <sup>(1)</sup>
<b>Humidity measurement range / accuracy</b>	0 ... 100 % r.h. / +- 3% r.h. <sup>(1)</sup>
<b>Illumination measurement range / accuracy</b>	0 ... 65000 lux / +-10 %
<b>Acceleration measurement range / accuracy</b>	+/- 2 g / +-0.03 g
<b>Acceleration threshold for immediate report</b>	0.03 g (default, configurable via NFC)
<b>Update rate (excl. random offset)</b>	Every 60 seconds (configurable via NFC)
<b>Device configuration</b>	LRN button and NFC interface
<b>Supported EEP (selectable via NFC)</b>	D2-14-41 (default) D2-14-40, A5-02-05, A5-04-01, A5-04-03 A5-06-02, A5-06-03, A5-14-05, D5-00-01
<b>Power supply</b>	Integrated solar cell
<b>Operating time in darkness</b>	4 days (after full charge)
<b>Backup power supply (optional)</b>	CR1632
<b>Dimensions (STM 550 Module)</b>	40 mm x 40 mm x 13 mm
<b>Dimensions (EMSI Finished product)</b>	49 mm x 49 mm x 13 mm

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

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

*specified below***1.4 Environmental conditions**

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

**1.5 Packaging information****1.5.1 STM 550**

<b>Packaging Unit</b>	100 units
<b>Packaging Method</b>	10 modules per tray, 10 trays per box

**1.5.2 EMSI**

<b>Packaging Unit</b>	Individual unit packaging
<b>Packaging Method</b>	TBD packages within one box

**1.6 Ordering information**

Type	Ordering Code	Frequency
<b>STM 550 (Module)</b> <b>EMSIA (Finished Product)</b>	S6201-K516 E6201-K516	868.300 MHz
<b>STM 550U (Module)</b> <b>EMSIU (Finished Product)</b>	S6251-K516 E6251-K516	902.875 MHz
<b>STM 550J (Module)</b> <b>EMSIJ (Finished Product)</b>	S6261-K516 E6261-K516	928.350 MHz



STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 2 Functional overview

### 2.1 Product description

The energy-harvesting multi-sensor module STM 550 provides wireless sensing functionality without batteries. Power is provided by a connected solar cell. STM 550 transmits sensor data using the EnOcean radio standard.

STM 550 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.

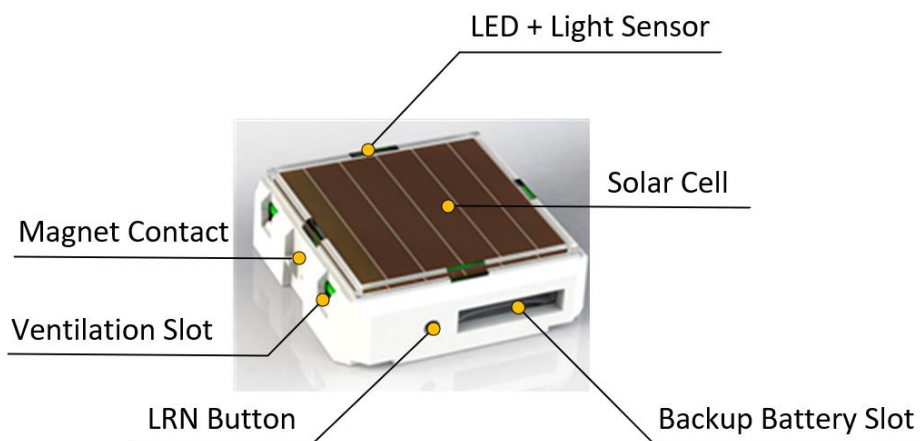
Radio telegrams transmitted by STM 550 can be authenticated and encrypted using AES-128 security based on a device-unique private key and a sequence counter in accordance to the EnOcean Alliance Security Specification. This ensures integrity, confidentiality and authenticity of the transmitted telegrams and prevents telegram replay (retransmission of previously transmitted telegrams).

### 2.2 Product interface

The STM 550 product interface consists of the following elements:

- LRN button and LED
- Solar cell harvesting energy from ambient light
- Ambient light sensor (co-located with the LED)
- Magnet contact sensor
- Ventilation slots to ensure airflow to the temperature and humidity sensor
- Backup battery slot for a CR1632 battery
- Backup battery ejector slot (on the back side)
- Product label (on the back side)

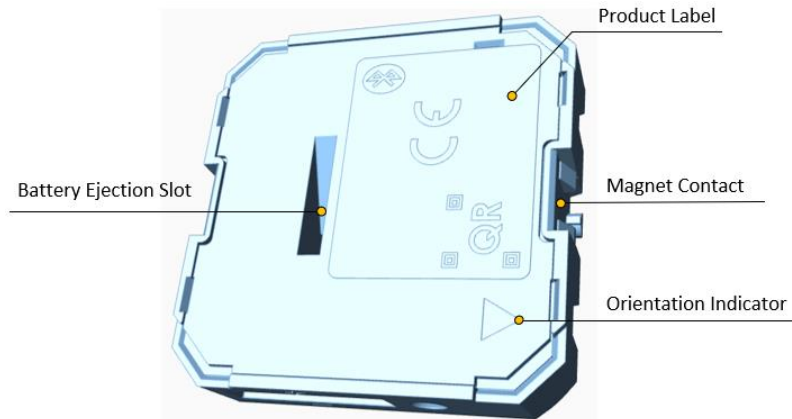
Figure 2 below shows the external interfaces on the front side of the STM 550 module.



**Figure 2 – STM 550 external interfaces (front side)**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

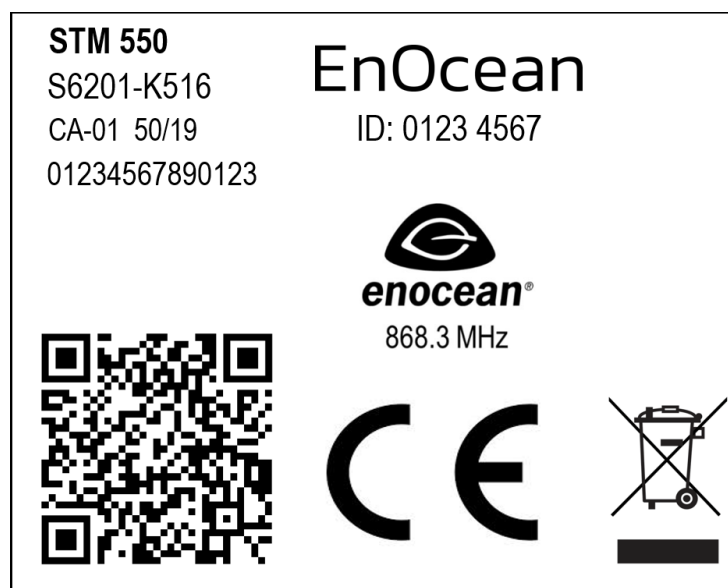
Figure 3 below shows the external interfaces on the back side of the STM 550. The orientation indicator points towards the side where the magnet contact is located.



**Figure 3 – STM 550 external interfaces (back side)**

## 2.3 Product label

Each STM 550 module contains a product label with a commissioning QR code. Figure 4 shows the STM 550 product label. STM 550U and STM 550J use the same label structure. Note the commissioning QR code (described in chapter 7.2) on the bottom left side.



**Figure 4 – STM 550 product label**

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 2.4 Functional modes

STM 550 supports six functional modes:

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



Out of the box, STM 550 will be in standby mode to conserve energy during transport and storage. Upon initial setup, STM 550 has to be set to standard operation mode by pressing the LRN button shortly as described in chapter 10.1 or via the NFC interface as described in chapter 9.5.11

### 2.4.1 Standard operation mode

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

The STM 550 wake-up timer is by default configured to wake-up STM 550 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 550 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 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**2.4.2 Standby (Sleep) mode**

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

In standby mode, STM 550 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 4.1 or using the MODE field of the FUNCTIONAL\_MODE NFC register as described in chapter 9.5.11.

**2.4.3 Learn mode**

In learn mode, STM 550 will transmit a Teach-in telegram to communicate the device source address and security key to a receiver. After that transmission, STM 550 will return back to standard operation mode.

Learn mode can be selected using the LRN button as described in chapter 4.1 or using the MODE field of the FUNCTIONAL\_MODE NFC register as described in chapter 9.5.11.

**2.4.4 Function test mode**

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

STM 550 can be set into function test mode via the LRN button as described in chapter 4.1 or via the MODE field of the FUNCTIONAL\_MODE NFC register as described in chapter 9.5.11.

Function Test Mode will be stopped once the LRN button is pressed or the functional mode is changed via the NFC interface.



Function test mode consumes significant energy due to the high transmission rate. It is therefore recommended to use this mode only for short periods of time. Make sure you exit function mode (by button press or via NFC) as soon as it is not required anymore. Consider mounting a backup battery for the duration of the testing if extensive testing is required.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**2.4.5 Illumination test mode**

During installation, STM 550 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 10.5.

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

After that, STM 550 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 9.5.28.

Illumination test mode can be selected using the MODE field of the FUNCTIONAL\_MODE NFC register as described in chapter 9.5.11.

**2.4.6 Factory reset mode**

STM 550 can be reset to its standard settings using factory reset mode. Upon entering this mode, STM 550 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 4.1 or using the MODE field of the FUNCTIONAL\_MODE NFC register as described in chapter 9.5.11.

**STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS****2.5 Reporting interval**

STM 550 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 550 will measure and report its status approximately once per minute.

STM 550 is designed to apply fluctuations up to  $\pm 10\%$  to any configured reporting interval to increase transmission reliability and meet regulatory requirements.

If STM 550 is configured to report acceleration, then it will always report the initial acceleration detection after a period without detected acceleration immediately. Likewise, if STM 550 is configured to report magnet contact status, then it will report any change in the status of the magnet contact sensor (open  $\rightarrow$  closed or closed  $\rightarrow$  open) immediately.

**2.5.1 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 550 will increase its power consumption since it will measure and transmit more often. Likewise, increasing the reporting interval of STM 550 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 550 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 2.4.5. 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 550 is installed.

STM 550 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 1 below.

	6 hrs / day	8 hrs / day	10 hrs / day	12 hrs / day
50 lux	Not supported	Not supported	180 s	120 s
100 lux	120 s	90 s	90 s	60 s
150 lux	90 s	60 s	60 s	45 s
200 lux	60 s	45 s	45 s	30 s
300 lux	45 s	30 s	30 s	20 s
400 lux	30 s	25 s	25 s	15 s
500 lux	25 s	20 s	20 s	15 s

**Table 1 – Minimum self-supplied reporting intervals**

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

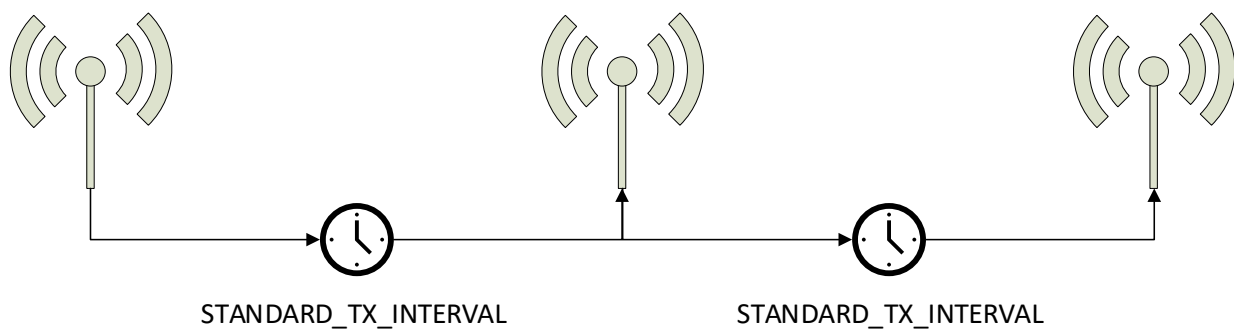
### 2.5.2 Standard reporting interval

The standard reporting interval determines the default interval between two status updates of STM 550.

The standard reporting interval can be adjusted using the `STANDARD_TX_INTERVAL` NFC register as described in chapter 9.5.12. Consider the available energy before lowering the reporting interval as discussed in chapter 2.5.1.

The default setting for the standard reporting interval is one status update once every 60 seconds (1 minute).

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



**Figure 5 – Standard reporting interval**

STM 550 can be configured to use a lower reporting interval, i.e. provide updates more often, based on certain conditions as described below.



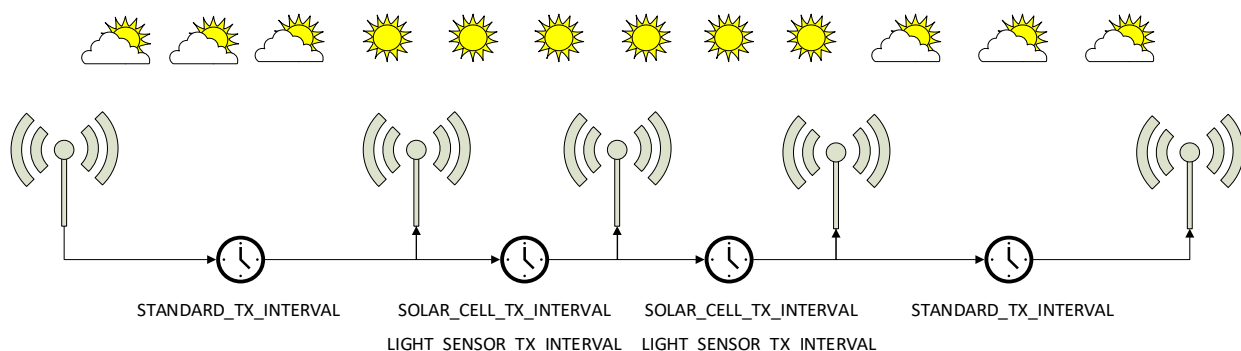
## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 2.5.3 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 550 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. 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 9.5.17 and 9.5.19 respectively
2. Configure the desired reporting interval using either the `SOLAR_TX_INTERVAL` or the `LIGHT_TX_INTERVAL` registers described in chapters 9.5.18 and 9.5.20 respectively
3. 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 9.5.13.

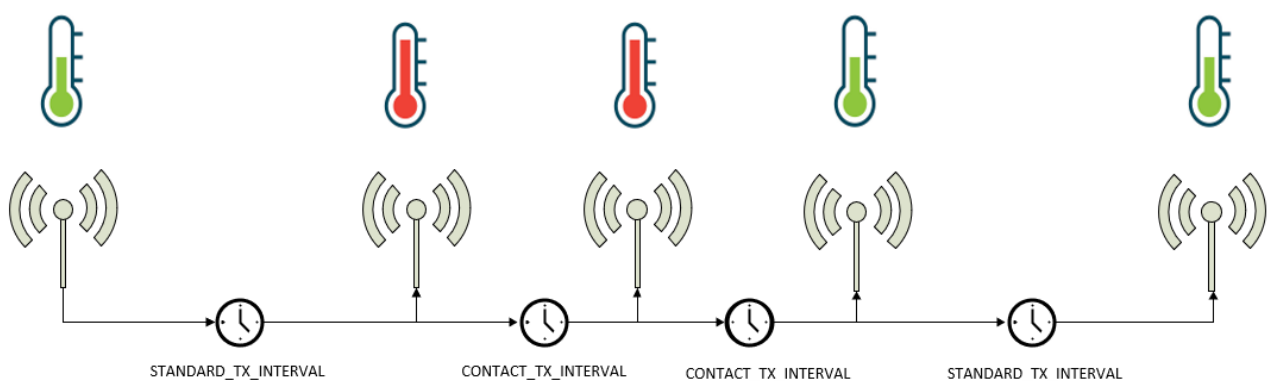
Consider the available energy before lowering the reporting interval as discussed in chapter 2.5.1.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

**2.5.4 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 550 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 EEP selected using the EEP register as described in chapter 9.5.8 reports the measured temperature
2. Configure the desired temperature threshold using the `TEMPERATURE_THRESHOLD` register described in chapter 9.5.23
3. Configure the desired reporting interval using the `TEMPERATURE_TX_INTERVAL` register described in chapter 9.5.24
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 9.5.13.

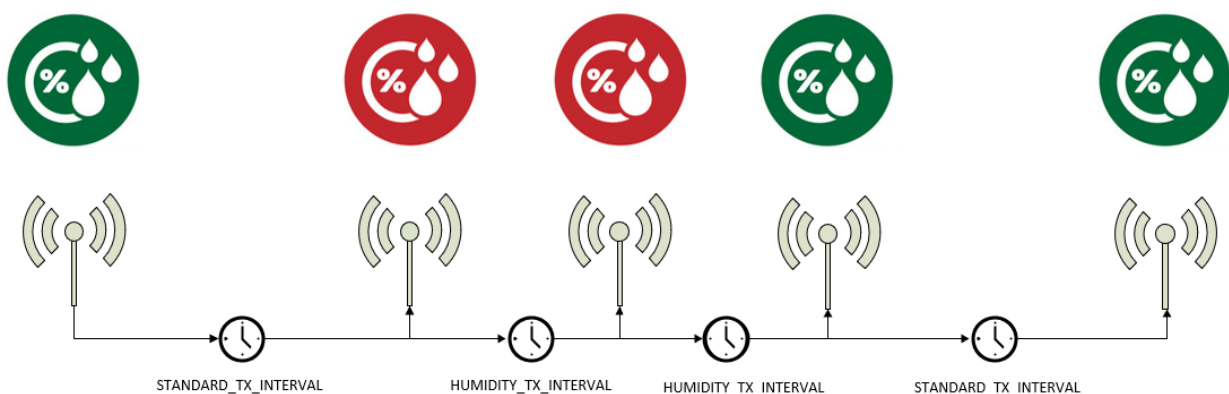
Consider the available energy before lowering the reporting interval as discussed in chapter 2.5.1.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

**2.5.5 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 550 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 EEP selected using the EEP register as described in chapter 9.5.8 reports humidity
2. Configure the desired humidity threshold using the HUMIDITY\_THRESHOLD register described in chapter 9.5.25
3. Configure the desired reporting interval using the HUMIDITY\_TX\_INTERVAL register described in chapter 9.5.26
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 9.5.13.

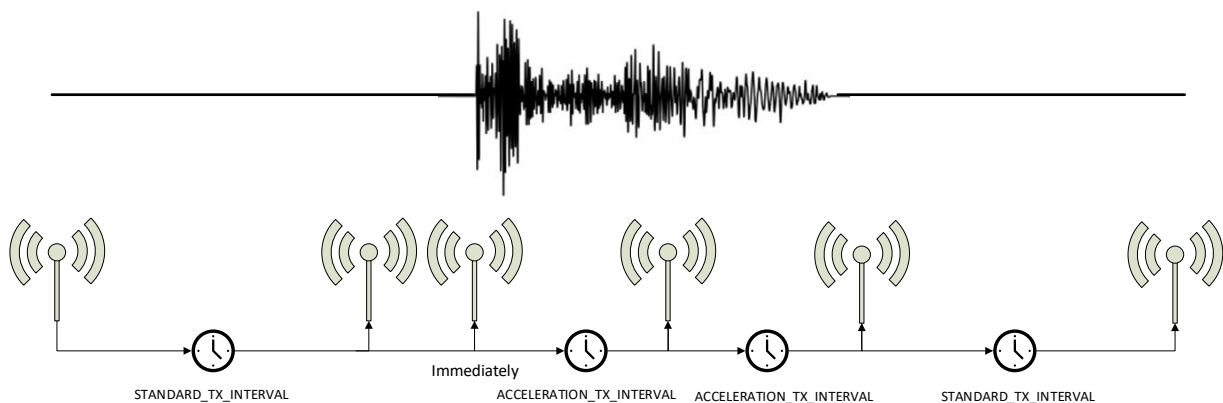
Consider the available energy before lowering the reporting interval as discussed in chapter 2.5.1.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

**2.5.6 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 550 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 EEP selected using the EEP register as described in chapter 9.5.8 reports acceleration
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 9.5.16
3. Configure the desired acceleration threshold using the ACCELERATION\_THRESHOLD register described in chapter 9.5.21
4. Configure the desired reporting interval using the ACCELERATION\_TX\_INTERVAL register described in chapter 9.5.22
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 9.5.14.

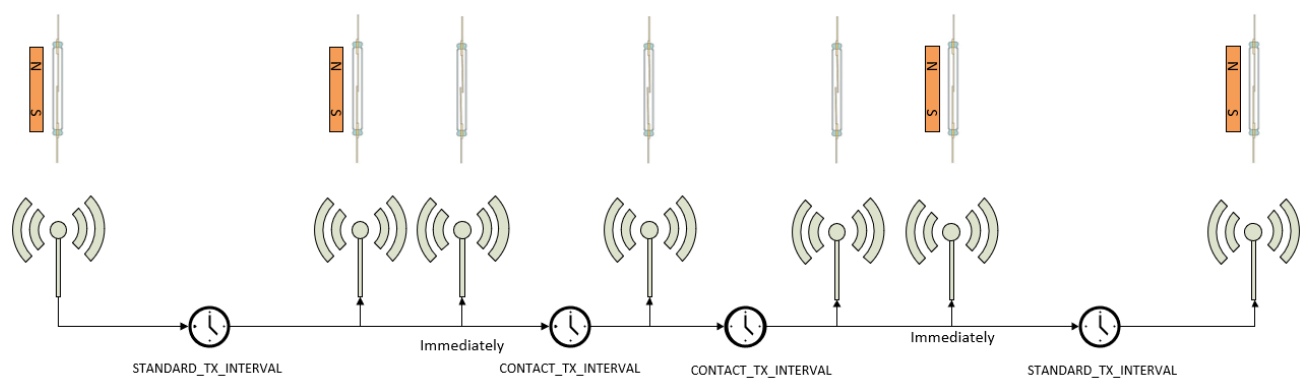
Consider the available energy before lowering the reporting interval as discussed in chapter 2.5.1.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 2.5.7 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 550 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 550 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 EEP selected using the EEP register as described in chapter 9.5.8 reports the magnet contact status
2. Configure the desired reporting interval using the `MAGNET_CONTACT_TX_INTERVAL` register described in chapter 9.5.22
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 9.5.14.

Consider the available energy before lowering the reporting interval as discussed in chapter 2.5.1.

### 2.5.8 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 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

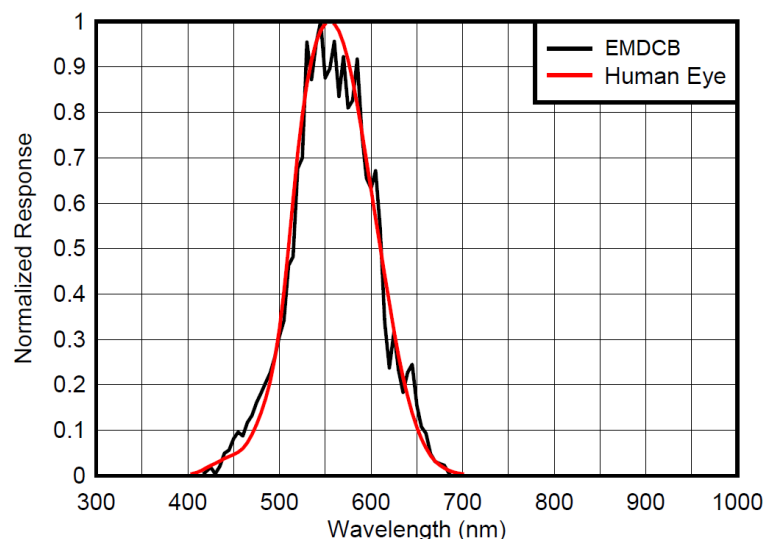
### 3 Sensor functionality

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

#### 3.1 Light level sensor

STM 550 integrates a dedicated light level 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 550 illumination sensor compared to that of the human eye.



**Figure 11 – Spectrum response of the light level sensor**

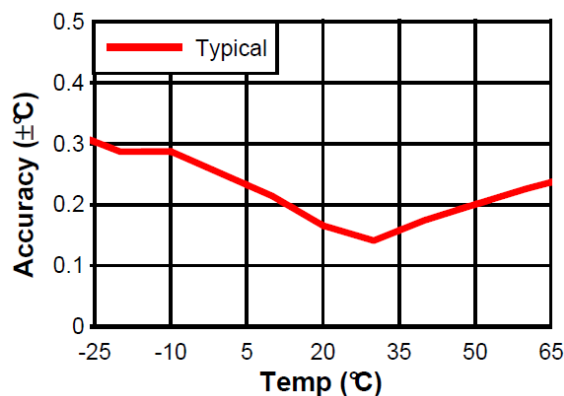
#### 3.2 Solar cell

In addition to the light sensor, STM 550 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 550.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 3.3 Temperature sensor

STM 550 integrates a high-performance temperature sensor achieving accuracy of better than  $\pm 0.3$  °C throughout the entire operation temperature range and an accuracy of better than  $\pm 0.2$  °C for the typical indoor temperature range. Figure 12 below shows the typical accuracy of the STM 550 temperature sensor as a function of the ambient temperature.

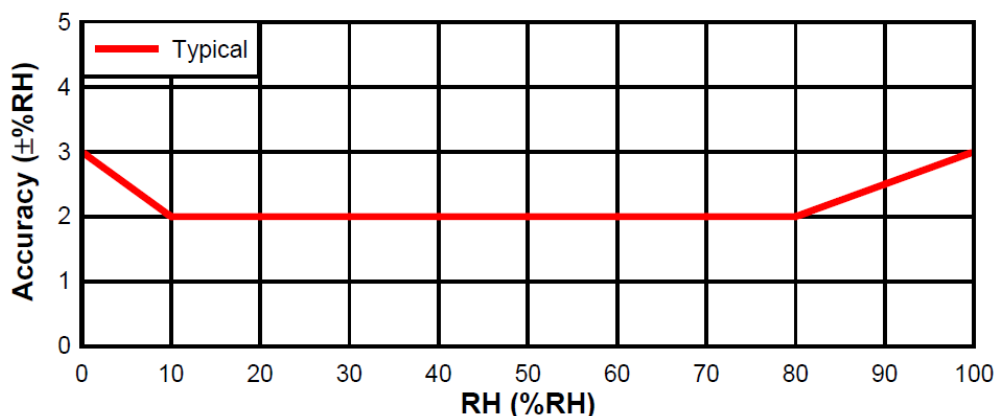


**Figure 12 – Temperature sensor accuracy**

To determine the overall system accuracy, the quantization error (reporting step size) determined by the selected EnOcean Equipment Profile (EEP) has to be added to this value.

### 3.4 Humidity sensor

STM 550 integrates a high-performance humidity sensor achieving accuracy of better than  $\pm 3$  % r.h. throughout the entire operation temperature range and an accuracy of better than  $\pm 2$  % r.h. for the typical indoor humidity range. Figure 13 below shows the typical accuracy of the STM 550 humidity sensor as a function of the ambient humidity.



**Figure 13 – Humidity sensor accuracy**

To determine the overall system accuracy, the quantization error (reporting step size) determined by the selected EnOcean Equipment Profile (EEP) has to be added to this value.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

### 3.5 Acceleration sensor

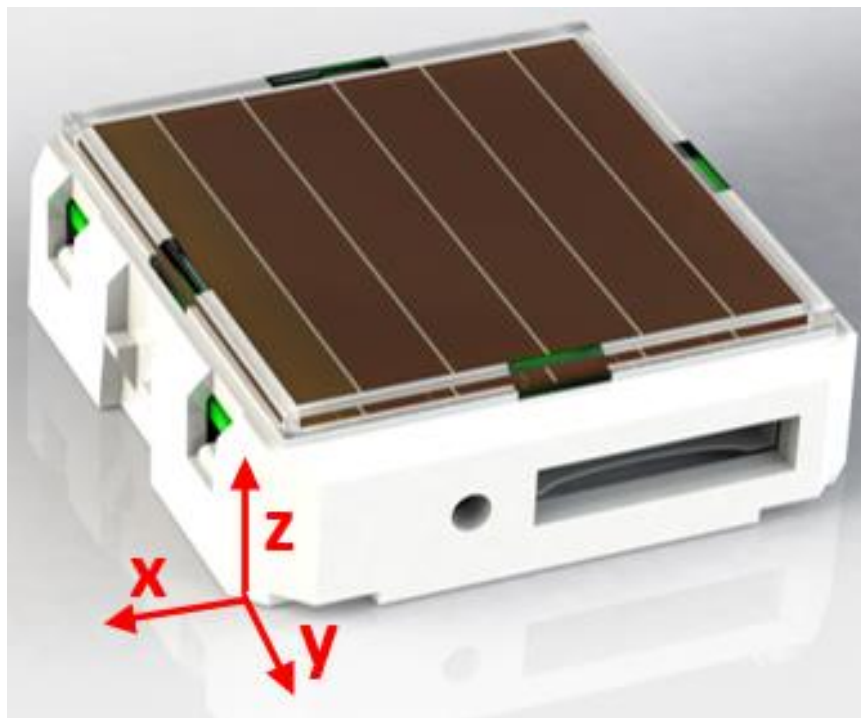
STM 550 integrates an acceleration sensor which provides two types of data:

- Acceleration vector orientation  
This allows determining the direction into which STM 550 is either accelerated (moved) or permanently oriented (positioned relative to the earth gravity vector). See chapter 10.2 for an example.
- Acceleration vector change  
This allows determining if STM 550 is moved or shaken

The second case (acceleration vector change) can be used to determine the presence or absence of small vibrations (acceleration vector changes). Examples use cases causing such small vibrations include asset utilization (e.g. a motor to which STM 550 is attached is running) or asset movement (e.g. an asset to which STM 550 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 acceleration vector relative to STM 550.



**Figure 14 – Acceleration sensor orientation**



STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

### 3.5.1 Wake on acceleration

If acceleration exceeding the configured threshold is detected for the first time after a period without exceeding this threshold then STM 550 wakes up immediately (wake on acceleration event).

Wake-on acceleration can be used to immediately trigger the transmission of a message when an item is moved or starts to vibrate. The threshold for the wake on acceleration feature is adjustable as described below.

### 3.5.2 Acceleration sensor parameters

The acceleration sensor integrated in STM 550 allows configuring the following parameters:

- **Full-scale magnitude**  
The full-scale magnitude determines the maximum acceleration magnitude that will be reported. Higher settings allow reporting higher magnitudes but will result in less resolution and thereby less sensitivity. The default measurement of  $\pm 2g$  should be sufficient for most use cases.
- **Acceleration threshold**  
The acceleration threshold determines the threshold of acceleration vector change required to trigger a wake-on vibration event as described above or to reduce the transmission interval as described in chapter 2.5.6.  
Setting a lower acceleration threshold results in a higher sensitivity to acceleration vector changes. STM 550 uses by default the minimum possible threshold corresponding to  $1/64$  of its full-scale magnitude.
- **Sampling rate**  
The sampling rate determines how often the acceleration vector will be measured. Higher sampling rates allow detecting shorter vibrations but require more energy to do so. Higher sampling rates should only be used if minor vibrations are not detected when using the minimum acceleration threshold.

Table 2 below shows the supported configuration options for each of these parameters and their default settings in STM 550.

Parameter	Supported Options (bold = default)
Full-scale magnitude	<b><math>\pm 2g</math></b> , $\pm 4g$ , $\pm 8g$ , $\pm 16g$
Acceleration threshold	<b><math>1/64</math></b> ... $63/64$ of full scale
Sampling rate	$1.6\text{ Hz}$ , <b><math>12.5\text{ Hz}</math></b> , $25\text{ Hz}$ , $50\text{ Hz}$
Measurement resolution	10 bit for each (x, y, z) direction

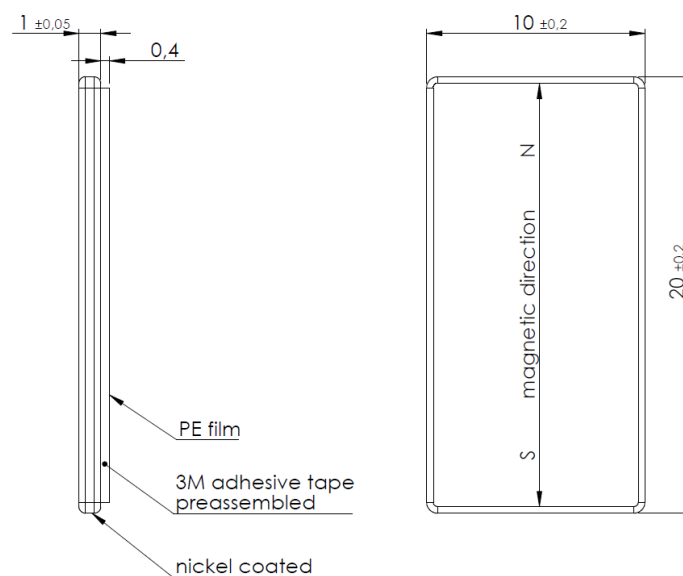
**Table 2 – Acceleration sensor parameters**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 3.6 Magnet contact sensor

STM 550 integrates a magnet contact sensor detecting presence or absence of a magnetic field in the proximity of the it. Refer to Figure 2 and Figure 3 for the location of the magnet contact sensor within STM 550 and to chapter 10.5 for mounting instructions.

EMSI 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 3 below lists the key parameters of this material for reference if use of alternative magnets is planned.

	Br		Hcb		Hcj		(BH) <sub>max</sub>		TW
	T	KGs	KA/m	KOe	KA/m	KOe	KJ/m <sup>3</sup>	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 3 – N35 material parameters**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 4 User interface

The user interface of STM 550 consist of the following items:

- LRN button and LED
- Backup battery interface

Please refer to chapterso identify the location of these items. They are described in more detail below.

#### 4.1 LRN button and LED

Most STM 550 device parameters can be configured using the NFC interface as described in chapter 9. Some of the most common parameters or states can additionally be configured using the LRN button.

Table 4 below lists those LRN button actions with the corresponding STM 550 response and LED feedback.

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

**Table 4 – STM 550 user interface actions**

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**4.2 Backup battery**

STM 550 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 via the battery ejector slot shown in Figure 3.



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!

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 5 Radio communication

STM 550 devices communicate using radio telegrams encoded according to the EnOcean Equipment Profile (EEP) specification and the EnOcean Alliance Signal Telegram specification on a radio link according to the EnOcean Alliance Radio Protocol (ERP).

STM 550 uses the ERP1 standard (ISO 14543-3-10) while STM 550U and STM 550J use the ERP2 (ISO 14543-3-11) standard.

### 5.1 Radio Frame format

As described above, STM 550 uses ERP1 as radio link while STM 550U and STM 550J use ERP2.

Note that EnOcean radio transceivers such as TCM 310 or TCM 515 will convert both ERP1 and ERP2 into the same EnOcean Serial Protocol (ESP3) format so that this difference is normally not noticeable.

#### 5.1.1 ERP1 frame format

The ERP1 radio frame format is shown in Figure 16 below.

RORG	DATA	SENDER EURID	STATUS	CRC
------	------	--------------	--------	-----

**Figure 16 – ERP1 Frame Format**

The most relevant fields of the ERP1 frame are the following:

- RORG (containing the EEP or SIGNAL RORG)
- SENDER EURID (Device address of the sender)
- DATA (Telegram payload containing the EEP)

#### 5.1.2 ERP2 frame format

The ERP2 radio frame format is shown in below.

LENGTH	HEADER	EXT_HEADER	SENDER EURID	DATA	CRC
--------	--------	------------	--------------	------	-----

**Figure 17 – ERP2 Frame Format**

The most relevant fields of the ERP2 frame are the following:

- HEADER (including the EEP or SIGNAL RORG)
- SENDER EURID (Device address of the sender)
- DATA (Telegram payload containing the EEP)

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 5.2 EnOcean Equipment Profiles (EEP) and SIGNAL telegrams

The data section within EnOcean radio telegrams uses one of the EnOcean Equipment Profiles (EEP) or one of the SIGNAL telegram types defined by EnOcean Alliance to encode sensor information. The EEP used is selected by the sender and must be supported by the receiver.

#### 5.2.1 STM 550 supported EEP

STM 550 supports a wide range of EEP suitable for different use cases. Table 5 below lists the supported EEP.

Profile	Type	Reported Parameters	Parameter Range	Size / Resolution
D5-00-01	1BS	Magnet Contact	Open / Closed	1 Bit
A5-02-05	4BS	Temperature	0°C ... 40°C	8 bit
A5-04-01	4BS	Temperature	0°C ... 40°C	8 Bit
		Humidity	0% ... 100% r.h.	8 Bit
A5-04-03	4BS	Temperature	-20°C ... 60°C	10 Bit
		Humidity	0% ... 100% r.h.	8 Bit
A5-06-02	4BS	Light Sensor	0 lx ... 1020 lx	8 Bit
A5-06-03	4BS	Light Sensor	0 lx ... 1000 lx	10 Bit
A5-14-05	4BS	Vibration Detector	Above / Below threshold	1 Bit
D2-14-40	VLD (9 Byte)	Temperature	-40°C ... 60°C	10 Bit
		Humidity	0% ... 100% r.h.	8 Bit
		Illumination	0 ... 100000 lx	17 Bit
		Acceleration x-axis	+/- 2.5g	10 Bit
		Acceleration y-axis	+/- 2.5g	10 Bit
		Acceleration z-axis	+/- 2.5g	10 Bit
		Acceleration Status	Above / Below threshold	2 Bit
D2-14-41 (Default)	VLD (9 Byte)	Temperature	-40°C ... 60°C	10 Bit
		Humidity	0% ... 100% r.h.	8 Bit
		Illumination	0 ... 100000 lx	17 Bit
		Acceleration x-axis	+/- 2.5g	10 Bit
		Acceleration y-axis	+/- 2.5g	10 Bit
		Acceleration z-axis	+/- 2.5g	10 Bit
		Acceleration Status	Above / Below threshold	2 Bit
		Magnet Contact	Open / Closed	1 Bit

**Table 5 – Supported EEP**

The default EEP used by STM 550 is D2-14-41 (VLD with 9 byte payload). It is possible to select another one of the supported EEP via the EEP register of the NFC configuration interface described in chapter 9.5.8.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**5.2.2 EEP structure**

Each EEP is identified using three fields:

- **RORG**  
RORG identifies the high-level telegram type, e.g. rocker switch telegram (RPS), one-byte sensor telegram (1BS), four-byte sensor telegram (4BS), variable length telegram (VLD), Universal Teach-in with EEP (UTE), etc.
- **FUNC**  
FUNC identifies the function group to which this telegram belongs, e.g. the function group of temperature sensors within the four-byte sensor telegram type
- **VARIANT (or TYPE)**  
VARIANT (which is confusingly also called TYPE) identifies the exact sensor variant within the function group, e.g. a 0 °C – 40 °C temperature sensor that is defined within the function group of temperature sensors

Figure 18 below shows the structure of the EEP identifier.

RORG	FUNC	VARIANT
0x00 ... 0xFF	0x00 ... 0x3F	0x00 ... 0x7F
8 bit	6 bit	7 bit

**Figure 18 – EEP identifier structure**

The EEP identifier is typically only transmitted during the initial teach-in (paring) between devices. For special cases (e.g. devices using more than one EEP), data telegrams might specify the EEP that is used.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 5.2.3 1BS telegram

1 Byte Sensor (1BS) telegrams are identified by the RORG field being set to 0xD5 which is followed by one byte of payload (Bit0 ... Bit7). 1BS telegrams are used exclusively to encode the status (open / closed) of a binary contact (typically a magnet contact)

The payload of 1BS telegrams encodes either the contact status (1BS Data Telegram) during normal operation or identifies a teach-in telegram (1BS Teach-in Telegram).

The distinction between data and teach-in telegrams is made based on the status of Bit4. If this bit is set to 0 then the telegram is a 1BS Teach-in Telegram; if this bit is set to 1 then the telegram is a 1BS Data Telegram.

### 5.2.4 4BS telegram

4 Byte Sensor (4BS) telegrams are identified by the RORG field being set to 0xA5 which is followed by four bytes of payload (Bit0 ... Bit31).

The payload of 4BS telegrams encodes either the sensor status (4BS Data Telegram) during normal operation or identifies EEPROM and manufacturer of the device during teach-in (4BS Teach-in Telegram).

The distinction between data and teach-in telegrams is made based on the status of Bit28. If this bit is set to 0 then the telegram is a 4BS Teach-in Telegram; if this bit is set to 1 then the telegram is a 4BS Data Telegram.

### 5.2.5 VLD telegram

Variable Length Data (VLD) telegrams are identified by the RORG field being set to 0xD2. They carry a variable length payload which can be between 1 and 14 byte long.

### 5.2.6 UTE (Universal Teach-in) telegram

Variable Length Data (VLD) telegrams carry a variable length payload, therefore it is not possible to use one bit at a pre-defined location to distinguish between data and teach-in telegrams.

Devices communicating using VLD data telegrams therefore use the generic Universal Teach-in with EEPROM (UTE) format when transmitting a teach-in telegram. The format of such UTE telegram is shown in Figure 19 below.

UTE RORG	UTE DATA						
BYTE0	BYTE0	BYTE1	BYTE2	BYTE3	BYTE4	BYTE5	BYTE5
0xD4	CTRL	CHANNEL	MANUFACTURER_ID		FUNC	TYPE	RORG

**Figure 19 – UTE Telegram Structure**



## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 5.2.7 Signal telegram

SIGNAL telegrams are used to encode generic system conditions independent of specific sensor functionality of the device. Examples of such system conditions are internal energy level, available ambient energy and backup battery status.

SIGNAL telegrams are identified by having the RORG field of the data telegram set to 0xD0. After that, the SIGNAL type (what is reported) is identified by the 1 byte long MID field which is followed by the data corresponding to this SIGNAL type. Figure 20 below shows the structure of a SIGNAL telegram.

SIGNAL RORG	SIGNAL Type (MID)	SIGNAL Data
0xD0	0x00 ... 0xFF	Depending on SIGNAL Type

**Figure 20 – SIGNAL Telegram Structure**

Table 6 below lists the SIGNAL types supported by STM 550 together with their reported data.

MID	Content	Data
0x06	Energy status (remaining energy)	1 byte integer value (expressing %) Valid values: 0 ... 100
0x0D	Energy delivery of the harvester	1 byte Enumeration Valid values: 0x00 (best) ... 0x04 (worst)
0x10	Backup battery status	1 byte integer value (expressing %) Valid values: 0 ... 100

**Table 6 – Supported SIGNAL Types**

The transmission of each supported SIGNAL telegram can be individually enabled and disabled using the via the SIGNAL configuration register of the NFC interface as described in chapter 9.5.9.

STM 550 will transmit each of the enabled SIGNAL telegram types once for every  $n$  EEP (data) telegrams with  $n$  being a configurable parameter that can be set via the SIGNAL configuration register of the NFC interface as described in chapter 9.5.9.

The default setting is that each enabled SIGNAL telegram type will be transmitted once every 32 EEP (data) telegrams, i.e. STM 550 will provide an update about the enabled internal parameters approximately once every 32 minutes.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 6 Security

STM 550 implements the security handling functions as specified in the EnOcean security specification: <https://www.enocean-alliance.org/sec/>

### 6.1 Basic concepts

Security for radio transmission addresses two main issues:

- Unauthorized interception (reception and correct interpretation) of transmitted data  
In doing so, a third (unauthorized) party is able to understand the content of a received content.
- Unauthorized transmission of radio telegrams  
In doing so, a third (unauthorized) party is able to transmit a radio telegram that is treated by a receiver as valid request.

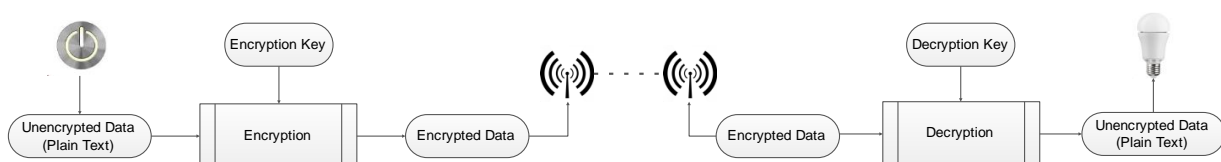
Somewhat loosely speaking, the goal of security has to be preventing an unauthorized person (often referred to as an *Attacker*) both from learning about the current state of a system and from actively changing it.

These goals can be achieved via techniques such as telegram encryption, telegram authorization and dynamic security key modification. All three techniques will be reviewed in the subsequent chapters for reference.

#### 6.1.1 Telegram encryption

The goal of telegram encryption is to prevent unauthorized receivers from correctly interpreting the content of a telegram.

In order to do so, the original (plain text) data is *encrypted* with a *key* thus transforming it into encrypted, unreadable data. Only when the correct key is known it is possible to transform – *decrypt* – the encrypted data into readable data again. Figure 21 below shows the concept.



**Figure 21 – Telegram encryption**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 6.1.2 Telegram authentication

The goal of telegram authentication is to prevent unauthorized senders to transmit apparently valid commands causing the receiver to perform unauthorized actions.

Telegram authentication works by creating a *message signature* (often called *Cipher-based Message Authentication Code* or *CMAC* in short) based on the content of the telegram and the secret key.

Essentially, the telegram data is transformed via a defined algorithm using the secret key into a unique, fixed size signature. STM 550 by default uses a signature length of 32 bit to authenticate each message.

For an optimal signature algorithm, the likelihood of two different messages creating the same message signature is inversely proportional to the signature size, so for a 32 bit signature the likelihood would be approximately one in 4 billion.

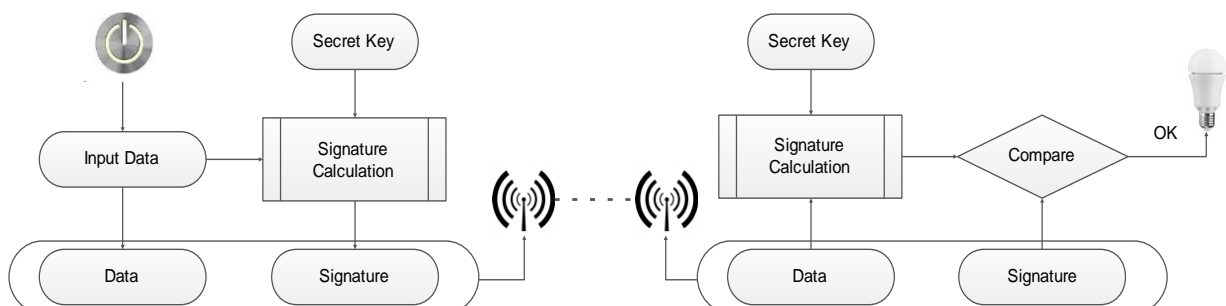
For message authentication purposes, the message signature (CMAC) is typically appended to the message itself and transmitted together with it.

When the receiver receives such a message, it will itself calculate the CMAC based on the secret key and the content of the received message. The receiver then compares the CMAC it calculated with the CMAC it received as part of the message.

If both CMAC are the same, then the receiver can establish two important facts:

1. The message originates from an owner of the secret key
2. The content of the message has not been modified

Figure 22 below illustrates the content authorization via a CMAC signature.



**Figure 22 – Telegram authentication**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

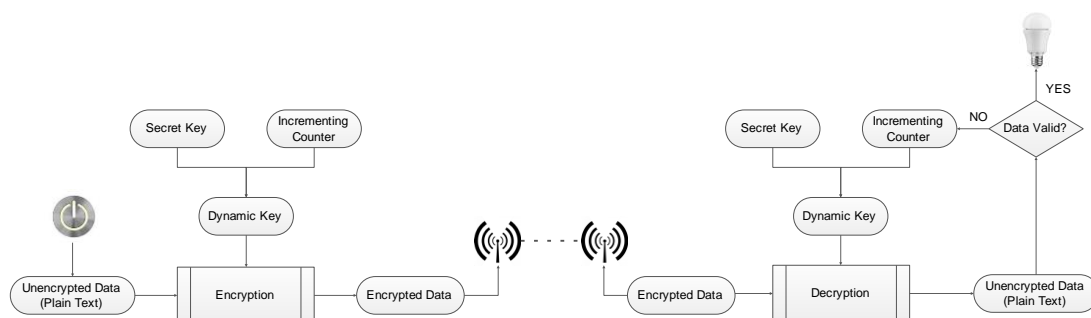
### 6.1.3 Dynamic security key modification

One fundamental problem with both content protection and content authorization is that using the same input data (plain text) with the same key always yields the same encrypted data and the same signature.

This enables attacks based on monitoring previous system behaviour. If an attacker has observed that a certain data telegram results in a certain light being turned on then he could use this information to identify - or even actively send - similar telegrams in the future. This type of attack is often called *Replay Attack* since it works by reusing (replaying) previously used data telegrams.

In order to prevent this type of attack, either the telegram data or the security key must change to ensure that identical input data does not create identical encrypted radio telegrams. The mechanism used by the transmitter to change the telegram data or the security key has to be known to the receiver in order to correctly decrypt and authenticate received data telegrams.

The change of telegram data or security key is typically ensured by means of monotonously incrementing counters. The value of such counter is then used to either modify the telegram payload or the security key. EnOcean systems use the latter approach.



**Figure 23 – Dynamic security key modification**

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 6.2 Security parameters

The following security parameters are used to define secure communication based on EnOcean Alliance security specification between a sender and a receiver:

- EURID (Device ID of the sender)
- Rolling code size and current value
- Signature (MAC) size
- Security algorithm

Those parameters are communicated from STM 550 to the receiver during teach-in either via a secure teach-in telegram, via NFC configuration or via the QR code of STM 550.

### 6.2.1 EURID

The EURID identifies the sender of each radio telegram using a unique 6 byte value. The EURID of an EnOcean device is assigned at manufacturing and cannot be changed.

### 6.2.2 Security key

The security key is a random 128 Bit (16 byte) value that is known only to the sender and the receiver(s). It is used to encrypt, decrypt and authenticate telegrams.

### 6.2.3 Rolling code

The rolling code is a monotonously incrementing counter used to modify the encryption key of secure telegrams as described in chapter 6.1.3. The rolling code is generated by the sender and monitored by the receiver.

STM 550 uses by default a 32 bit rolling code counter which will be initialized to 0 at the time of production and increment whenever a telegram is transmitted.

The receiver will store the most recently received rolling code value and only accept telegrams with higher rolling code values to avoid retransmission of previously transmitted messages.

### 6.2.4 Security algorithm

EMDC uses variable AES based on AES-128 to encrypt and authenticate its telegrams.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**6.3 STM 550 security implementation**

STM 550 supports both standard and high security modes as defined by EnOcean Alliance. The security mode can be selected both via the LRN button and via the NFC interface.

For high security mode, the default security level format (SLF) is set to use a 4 byte sequence counter to generate a 4 byte signature.

For backwards compatibility with legacy systems, it is possible to select via NFC a legacy mode using a 3 byte sequence counter to generate a 3 byte signature.

STM 550 will use secure chained telegrams (SEC\_CDM) if high security mode is used and the telegram payload (including rolling code and CMAC) exceeds 14 byte.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 7 Commissioning

Commissioning is the process by which STM 550 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 550 device. This is achieved by using a unique 48 Bit ID (Source Address) for each STM 550 device.
- **Security parameter exchange**  
The receiver needs to be able to authenticate radio telegrams from STM 550 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 550 to authenticate its radio telegrams.

STM 550 provides the following options for these tasks:

- **Radio-based commissioning**  
STM 550 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 550 device 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 550 will be commissioned. The commissioning tool then communicates these parameters to the intended receiver of STM 550 radio telegrams.
- **NFC commissioning**  
Each STM 550 device contains an NFC interface allowing to read device parameters and to configure a user-defined security key.

**STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS**

---

**7.1 Radio-based commissioning**

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

To do so, STM 550 can transmit a dedicated commissioning telegram identifying its relevant parameters as discussed in chapter 5.2. Transmission of the commissioning telegram is triggered by pressing the LRN button.

Radio-based commissioning mode is intended for applications where NFC commissioning cannot be used. Radio-based commissioning can be disabled via NFC if the device is operating in high security mode to prevent transmission of the security key over the air.

**7.2 QR code commissioning**

QR code-based commissioning reads the required parameters from a dedicated QR code in the product label (described in chapter 2.3). The parameters are encoded according to the ANSI/MH10.8.2-2013 industry standard.

The QR code on the product label in chapter 2.3 encodes the following string:

30S000012345678+Z0123456789ABCDEF0123456789ABCDEF+30PE6201-K516+2PCA01+S01234567890123

Identifier	Length of data (excluding identifier)	Value
30S	12 characters	EURID (hex)
Z	32 characters	Security Key (hex)
30P	Up to 10 characters	Ordering Code (E6201-K516)
2P	4 characters	Step Code - Revision (CA-01)
S	14 characters	Serial Number

**Table 7 – QR code format**

From this content, it is possible to extract the device address (000012345678) and the security key (0123456789ABCDEF0123456789ABCDEF) which can then be used to commission STM 550 into a receiver and to decrypt and authenticate STM 550 data telegrams as described in chapter 6.

**7.3 Commissioning via NFC interface**

STM 550 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 to configure the security key of STM 550 as described in chapter 9.



## **8 NFC interface**

STM 550 implements an NFC configuration interface that can be used to access (read and write) the STM 550 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 550 device.

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

Once the NFC reader has been disconnected, STM 550 will implement basic parameter validation (parameters within permitted boundaries) before accepting updated NFC parameters.

Should an updated parameter be out of bounds then all updated parameters will be rejected and the previous configuration will be retained. Otherwise the updated NFC parameters will be processed and the product parameters will be adapted accordingly.

### **8.1 NFC interface parameters**

The NFC interface of STM 550 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.

### **8.2 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.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

### 8.3 Using the NFC interface

Using the NFC interface requires the following:

- NFC reader  
This can be either a USB NFC reader connected to a PC or a suitable smartphone with NFC functionality
- NFC SW with read, write, PIN lock, PIN unlock and PIN change functionality  
This can be either a PC application or an Android / iOS app

These options are described in more detail below.

#### 8.3.1 PC with dedicated NFC reader

For PC-based applications, EnOcean recommends the TWN4 Multitech 2 HF NFC Reader (order code T4BT-FB2BEL2-SIMPL) from Elatec RFID Systems ([sales-rfid@elatec.com](mailto:sales-rfid@elatec.com)).

This reader is shown in Figure 24 below.



**Figure 24 – Elatec TWN4 MultiTech Desktop NFC Reader**

#### 8.3.2 Android or iOS smartphone with NFC

NFC functionality is available in certain Android (e.g. Samsung Galaxy S7 / S8 / S9 / S10) and iOS (iPhone7 or newer, firmware version 13 or newer) smartphones.

EnOcean provides the configuration app “EnOcean Tool” for these devices which can be downloaded directly from the respective app store.

At the time of writing, the tool was available from the Google Play Store using this link:  
<https://play.google.com/store/apps/details?id=de.enocean.easytool&hl=en>

Likewise, the tool was available from the Apple Store using this link:  
<https://apps.apple.com/de/app/enocean-tool/id1497283202>

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 8.4 NFC interface functions

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

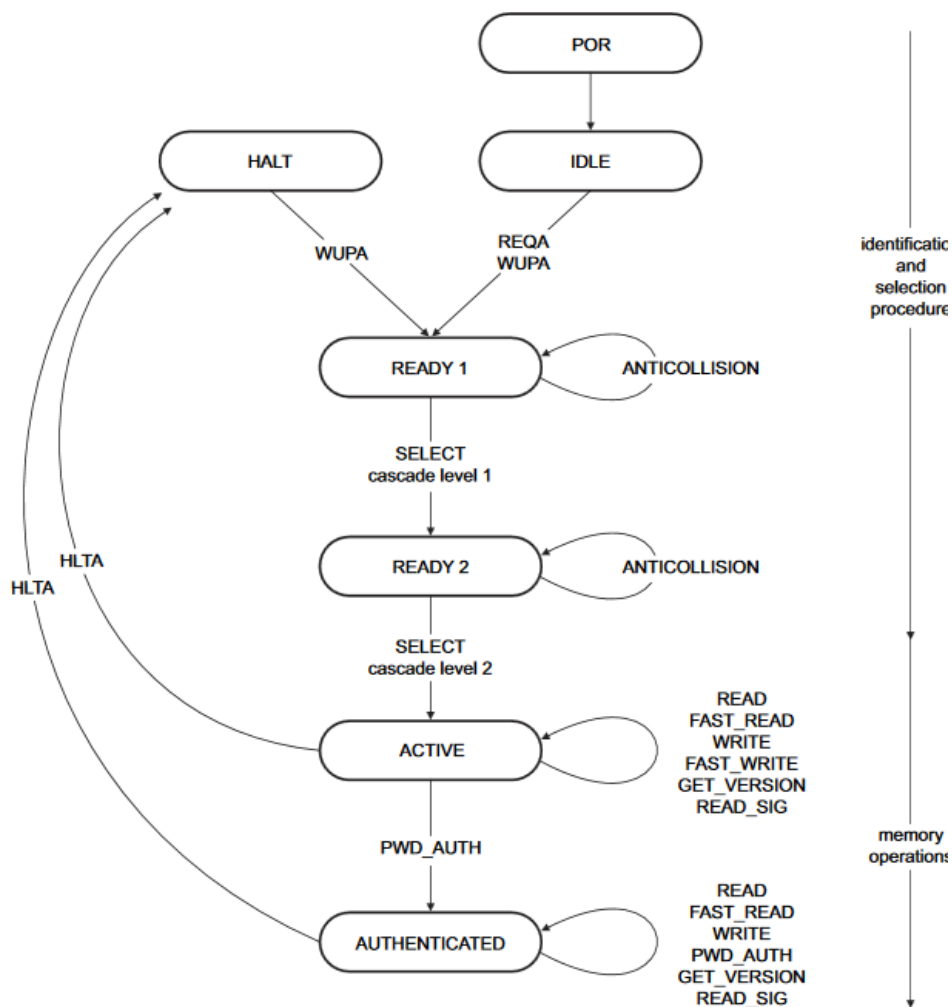
For specific implementation aspects related to the NXP implementation in NT3H2111, please refer to the NXP documentation which at the time of writing was available under this link:

[http://cache.nxp.com/documents/data\\_sheet/NT3H2111\\_2211.pdf](http://cache.nxp.com/documents/data_sheet/NT3H2111_2211.pdf)

The following chapters summarize the different functions for reference purposes.

#### 8.4.1 NFC interface state machine

Figure 25 below shows the overall state machine of the NFC interface.



**Figure 25 – NFC interface state machine**

### **8.4.2 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.

### **8.4.3 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.

### **8.4.4 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.

### **8.4.5 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.

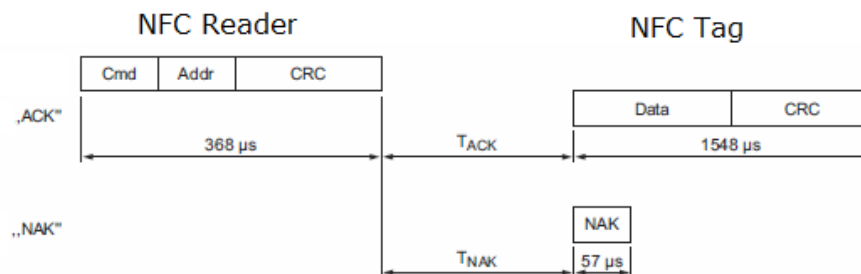
## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 8.4.6 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 26 below shows the read command sequence.

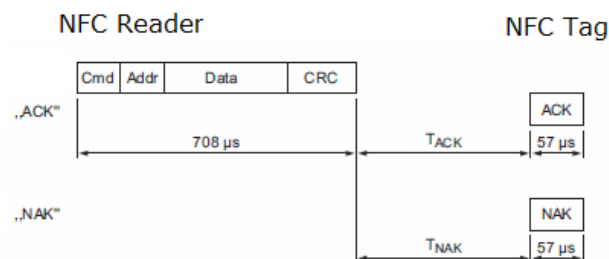


**Figure 26 – NFC read command sequence**

### 8.4.7 Write command

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

Figure 27 below shows the read command sequence.



**Figure 27 – NFC write command sequence**

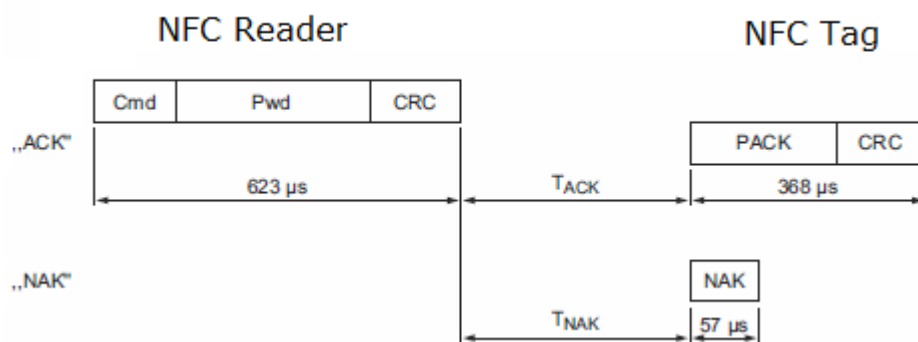
## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

#### 8.4.8 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 28 below shows the password authentication sequence.



**Figure 28 – Password authentication sequence**

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

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

## 9 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.

### 9.1 NFC memory areas

These NFC pages are allocated into the following areas:

- Device Identification NDEF string (Public read-only access; no PIN required)  
This area contains an NDEF string identifying key device parameters
- User Information NDEF string (Public read / write access; no PIN required)  
This area allows any user to read or write information about the device such as the intended installation location or additional instructions
- 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 such as information about the installation or any other data that the user deems relevant.  
STM 550 does not use this area and does not interpret its content in any way.

The organization of the STM 550 NFC memory map is shown in Table 8 below.

NFC Address	Memory Area	Content
0x00 ... 0x1D	PRODUCT NDEF	Device identification NDEF string (read-only)
0x1E ... 0x30	USER NDEF	User information NDEF string (read / write access)
0x31 ... 0x33	NFC HEADER	NFC memory revision (read-only)
0x34 ... 0x3F	INTERNAL DATA	Internal data (Do not use)
0x40 ... 0x54	CONFIGURATION	Configuration registers (Read / Write, PIN protected)
0x55 ... 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 8 – STM 550 NFC memory areas**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.2 Device identification 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 550 could be:

```
6PEN0+30S0000012345678+1P000B00000004C+30PS6221-K516+2PDA04+2Z01234567891234
+3C31+01000000
```

This NDEF string encodes the parameters shown in Table 9 below.

Identifier	Length of data (excl. identifier)	Value
6P	3 characters	Standard: "ENO"
30S	12 characters	EURID (6 byte, variable)
1P	12 characters	EnOcean Alliance Product ID STM 550: „000B00000004C“ STM 550U: „000B00000004D“ STM 550J: „000B00000004E“
30P	10 characters	Ordering Code STM 550: "S6201-K516" STM 550U: "S6251-K516" STM 550J: "S6261-K516"
2P	4 characters	Step Code and Revision ("DA04")
2Z	14 characters	NFC UID (14 byte, globally unique)
3C	2 characters	Header Start Address ("31" = 0x31)
16S	8 characters	SW Version Example: 01000000 = 01.00.00.00

**Table 9 – NDEF Parameters**

### 9.3 User information NDEF

The NDEF area allows the user to store a string of up to 64 characters starting at page 0x20 and ending at page 0x2F. The remaining pages in this area (0x1E, 0x1F, 0x30) provide the required NDEF formatting information and cannot be changed by the user.



STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 9.4 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.

### 9.4.1 NFC HEADER area structure

The structure of the NFC HEADER area is shown in Figure 29 below.

NFC Address	Content			
	Byte 0	Byte 1	Byte 2	Byte 3
0x31	START (0xE0)	LENGTH (0x0A)	VERSION (0x01)	OEM MSB (0x00)
0x32	OEM LSB (0x0B)	DEVICE_IDENTIFIER (0x000003)		
0x33	REVISION (0x03)	END (0xFE)	UNUSED (0x0000)	

**Figure 29 – 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 550, 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 550, the DEVICE\_IDENTIFIER is set to 0x000003
- **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 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5 CONFIGURATION

The CONFIGURATION area allows configuring the device parameters and is therefore the most important part of the NFC memory. Configuration registers larger than 8 bit use big endian format, i.e. the most significant byte comes first.

Read or write access to the CONFIGURATION area is only possible after issuing a PWD\_AUTH command as described in chapter 8.4.8 using the correct 32 bit PIN code.

#### 9.5.1 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 550 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.

#### 9.5.2 CONFIGURATION area structure

The structure of the CONFIGURATION area is shown in Figure 30 below.

NFC Address	Content			
	Byte 0	Byte 1	Byte 2	Byte 3
0x40	PRODUCT_ID (as characters in ASCII format) Will be copied to NDEF Header			
0x41				
0x42				
0x43	RFU			
0x44	USER_KEY (128 Bit) (Write Only - Will be reset to zero after it has been copied to internal memory) Can be used as alternative security key instead of FACTORY_KEY			
...				
0x47				
0x48	SECURITY_KEY_MODE	SECURITY_CFG	RFU	
0x49	ECP	SIGNAL	LED_MODE	FUNCTIONAL_MODE
0x4A	STANDARD_TX_INTERVAL		RFU	
0x4B	NFC_PIN_CODE			
0x4C	THRESHOLD_CFG1	THRESHOLD_CFG2	LIGHT_SEN- SOR_CFG	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		RFU	
0x54	LIGHT TEST RESULT		RFU	

**Figure 30 – CONFIGURATION area structure**

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

### 9.5.3 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.

### 9.5.4 PRODUCT\_ID

The EnOcean Alliance Product ID uniquely identifies each product within the EnOcean Alliance ecosystem. The Product ID consists of a 2 byte manufacturer identification code (assigned by EnOcean Alliance) and a 4 byte product identification code (assigned by the manufacturer).

EnOcean has been assigned the manufacturer identification code 0x000B. EnOcean has assigned the following product identification codes to STM 550:

STM 550: 0000004C  
STM 550U: 0000004D  
STM 550J: 0000004E

The PRODUCT\_ID register contains the Product ID in ASCII format (12 characters) and allows changing both manufacturer and product identification. Changing the PRODUCT\_ID will also cause the PRODUCT ID field in the NDEF string (described in chapter 9.2) to be updated.

Figure 31 below shows the structure of the PRODUCT\_ID register. This register contains the sequence of 12 ASCII characters (1 byte each) starting with CH0 and ending with CH11.

PRODUCT_ID											
CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10	CH11
Manufacturer ("000B")				Product ID ("0000004C", "0000004D" or "0000004E")							

**Figure 31 – PRODUCT\_ID**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.5 USER\_KEY

Each STM 550 module is pre-programmed at the factory with a randomly generated 128 bit security key (FACTORY\_KEY). This key will by default be used to encrypt and authenticate STM 550 radio telegrams when operating in high security mode.

In certain applications it might be desirable to assign a different (user-defined) security key (USER\_KEY) during commissioning to STM 550. This can be done by writing the user-defined security key to the USER\_KEY register and setting KEY SELECTION field of the SECURITY\_KEY\_MODE to 0b01 as described below.

Note that the USER\_SECURITY\_KEY register is a write-only register meaning that it is not possible to read back a user-defined security key.

### 9.5.6 SECURITY\_KEY\_MODE

The register SECURITY\_KEY\_MODE allows selecting if FACTORY\_KEY or USER\_KEY should be used to encrypt and authenticate STM 550 radio telegrams in high security mode. In addition, it allows disabling the transmission of Secure Teach-in telegrams in order to protecting the security key.

Note that if the transmission of a secure teach-in telegram has been disabled and is subsequently re-enabled then USER\_KEY will be reset to FACTORY\_KEY.

Figure 32 below shows the structure of the SECURITY\_KEY\_MODE register.

SECURITY_KEY_MODE (Default: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU				SECURE LRN TELEGRAM		KEY SELECTION	

**Figure 32 – SECURITY\_KEY\_MODE register**

The encoding for the KEY SELECTION bit field is shown in Table 10 below.

KEY SELECTION	Security key used
0b00 (Default)	FACTORY_KEY is used
0b01	USER_KEY is used
0b10, 0b11	Reserved, do not use

**Table 10 – KEY SELECTION bit field encoding**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

The encoding for the SECURE LRN TELEGRAM bit field is shown in Table 11 below.

<b>SECURE LRN TELEGRAM</b>	<b>Secure LRN telegram</b>
0b00 (Default)	Secure LRN Telegram (containing security key) enabled
0b01	Secure LRN Telegram (containing security key) disabled
0b10, 0b11	Reserved, do not use

**Table 11 – SECURE LRN TELEGRAM bit field encoding**

### 9.5.7 SECURITY\_MODE

The register SECURITY\_MODE identifies the security settings used by STM 550. Figure 33 below shows the structure of the SECURITY\_MODE register.

SECURITY_MODE (Default: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU				SECURITY FORMAT		SECURITY MODE	

**Figure 33 – SECURITY\_MODE register**

The encoding for the SECURITY MODE bit field is shown in Table 12 below.

<b>SECURITY MODE</b>	<b>Security Mode</b>
0b00 (Default)	Standard, can be changed by the user with the LRN button
0b01	High Security, can be changed by the user with the LRN button
0b10	Standard, cannot be changed by the user with the LRN button
0b11	High Security, cannot be changed by the user with the LRN button

**Table 12 – SECURITY MODE bit field encoding**

The encoding for the SECURITY FORMAT bit field is shown in Table 13 below.

<b>SECURITY FORMAT</b>	<b>Advertising Interval</b>
0b00 (Default)	32 bit RLC and 32 bit CMAC
0b01	24 bit RLC and 24 bit CMAC
0b10, 0b11	Reserved, do not use

**Table 13 – SECURITY FORMAT bit field encoding**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.8 EEP

The EEP register determines the EnOcean Equipment Profile (EEP) used by STM 550 for the transmission of data telegrams. Figure 34 below shows the structure of the EEP register.

EEP (Default: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU				EEP			

**Figure 34 – EEP register**

The encoding used by the SIZE bit field is shown in Table 14 below.

EEP	EnOcean Equipment Profile
0b0000 (Default)	D2-14-41
0b0001	D2-14-40
0b0010	A5-02-05
0b0011	A5-04-03
0b0100	A5-06-03
0b0101	A5-06-02
0b0110	D5-00-01
0b0111	A5-14-05
0b1000	A5-04-01
Others	Reserved, do not use

**Table 14 – EEP bit field encoding**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.9 SIGNAL

STM 550 supports reporting its energy status, the current energy delivery of the harvester and the backup battery status via SIGNAL telegrams as described in chapter 5.2.7.

The type and the transmission rate of SIGNAL telegrams is controlled by the SIGNAL register shown in Figure 35 below.

SIGNAL (Default: 0x31)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRANSMISSION RATE				RFU	BACKUP BATTERY	ENERGY DELIVERY	ENERGY STATUS

**Figure 35 – SIGNAL register**

STM 550 supports the following SIGNAL types:

- ENERGY STATUS (MID 0x06)
- ENERGY DELIVERY (MID 0x0D)
- BACKUP BATTERY STATUS (MID 0x10)

Transmission of these supported SIGNAL types can be enabled by setting the associated configuration bit in the SIGNAL register to 0b1 and disabled by setting the associated configuration bit to 0b0.

By default, the reporting of the energy status (MID 0x06) is enabled while the reporting of energy delivery (MID 0x0D) and backup battery status (MID 0x10) is disabled.

STM 550 allows additionally to configure at what rate each enabled SIGNAL telegram will be sent using the TRANSMISSION RATE bit field as shown in Table 15 below. If more than one SIGNAL telegram is enabled, then this rate will apply to all enabled telegrams. The SIGNAL telegrams will be transmitted at different times (i.e. not all at the same time one after another) in that case.

TRANSMISSION RATE	Transmission rate for SIGNAL telegrams
0b0000	One SIGNAL telegram every 8 data telegrams
0b0001	One SIGNAL telegram every 16 data telegrams
0b0010	One SIGNAL telegram every 24 data telegrams
0b0011 (Default)	One SIGNAL telegram every 32 data telegrams
...	
0b1111	One SIGNAL telegram every 128 data telegrams

**Table 15 – TRANSMISSION RATE bit field encoding**

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**9.5.10 LED\_MODE**

The LED\_MODE register determines the brightness of the LED. Figure 36 below shows the structure of the LED\_MODE register.

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

**Figure 36 – LED\_MODE register**

The encoding used by the LED bit field is shown in Table 16 below.

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

**Table 16 – LED bit field encoding**



## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.11 FUNCTIONAL\_MODE

The FUNCTIONAL\_MODE register can be used to switch between the different functional modes of STM 550 as described in chapter 2.4.

Figure 37 below shows the structure of the FUNCTIONAL\_MODE register.

FUNCTIONAL_MODE (Default: 0x00, OOB: 0x01)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU				MODE			

**Figure 37 – FUNCTIONAL\_MODE register**

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

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

**Table 17 – MODE bit field encoding**

Note that the default configuration (after factory reset) is standard operation mode while the out of box configuration (for new devices) is standby (sleep) mode.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.12 STANDARD\_TX\_INTERVAL

STM 550 uses a standard reporting interval which can be automatically adjusted based on sensor readings as described in chapter 2.5. The standard reporting interval is set by the register STANDARD\_TX\_INTERVAL shown in Figure 38 below.

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

**Figure 38 – STANDARD\_TX\_INTERVAL register**

The encoding used by the STANDARD INTERVAL bit field is shown in Table 18 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 18 – STANDARD INTERVAL bit field encoding**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

**9.5.13 THRESHOLD\_CFG1**

STM 550 can reduce the reporting interval based on the illumination of the solar cell, the light illumination of the light level sensor, the temperature or the humidity as described in chapter 2.5.

The use of these reduced reporting intervals is enabled by the THRESHOLD\_CFG1 register shown in Figure 39 below.

THRESHOLD_CFG1 (Default: 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 39 – THRESHOLD\_CFG1 register**

The encoding used by the SOLAR CELL bit field is shown in Table 19 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 19 – SOLAR CELL bit field encoding**

The encoding used by the LIGHT SENSOR bit field is shown in Table 20 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 20 – LIGHT SENSOR bit field encoding**

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

The encoding used by the TEMP SENSOR bit field is shown in Table 21 below.

<b>TEMP SENSOR</b>	<b>Reporting interval reduction based on temperature</b>
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 21 – TEMP SENSOR bit field encoding**

The encoding used by the HUMIDITY SENSOR bit field is shown in Table 22 below.

<b>HUMIDITY SENSOR</b>	<b>Reporting interval reduction based on temperature</b>
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 22 – HUMIDITY SENSOR bit field encoding**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.14 THRESHOLD\_CFG2

STM 550 can reduce the reporting interval based on the acceleration sensor status and the magnet contact status as described in chapter 2.5.

The use of these reduced reporting intervals is enabled by the THRESHOLD\_CFG2 register shown in Figure 40 below.

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

**Figure 40 – THRESHOLD\_CFG2 register**

The encoding used by the ACC SENSOR bit field is shown in Table 23 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 23 – ACC SENSOR bit field encoding**

The encoding used by the MC SENSOR bit field is shown in Table 24 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 24 – MC SENSOR bit field encoding**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

**9.5.15 LIGHT\_SENSOR\_CFG**

STM 550 allows the user to select if the reported light level is the one measured by the ambient light sensor or the one measured by the solar cell. Refer to chapter 3.1 for a description of the ambient light sensor and to chapter 3.2 for a description of the solar cell functionality.

The selection between these two options is made using the LIGHT\_SENSOR\_CFG register shown in Figure 41 below.

LIGHT_SENSOR_CFG (Default: 0x00)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU							INPUT SELECTION

**Figure 41 – LIGHT\_SENSOR\_CFG register**

The encoding used by the INPUT SELECTION bit field is shown in Table 25 below.

INPUT SELECTION	Input value used for illumination reporting
0b0 (Default)	Ambient light sensor
0b1	Solar cell

**Table 25 – INPUT SELECTION bit field encoding**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.16 ACC\_SENSOR\_CFG

The operation parameters of the acceleration sensor in STM 550 can be configured using the ACC\_SENSOR\_CFG registers shown in Figure 42 below. Refer to chapter 3.5 for a description of the acceleration sensor function.

ACC_SENSOR_CFG (Default: 0x14)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU			WAKE	SAMPLING RATE		FULL SCALE	

**Figure 42 – ACC\_SENSOR\_CFG register**

The encoding used by the FULL SCALE bit field is shown in Table 23 below.

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

**Table 26 – FULL SCALE bit field encoding**

The encoding used by the SAMPLING RATE bit field is shown in Table 27 below.

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

**Table 27 – SAMPLING RATE bit field encoding**

The wake-on-acceleration function can be enabled and disabled using the WAKE bit field as shown in Table 28 below.

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

**Table 28 – WAKE bit field encoding**

### 9.5.17 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 43 below.

SOLAR_THRESHOLD					
Bit 15	Bit 14	...		Bit 1	Bit 0
SOLAR_CELL_THRESHOLD					

**Figure 43 – SOLAR\_THRESHOLD register**

The encoding used by the SOLAR CELL THRESHOLD bit field is shown in Table 29 below.

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

**Table 29 – 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.



### 9.5.18 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 44 below.

SOLAR_TX_INTERVAL					
Bit 15	Bit 14	...		Bit 1	Bit 0
SOLAR_CELL_INTERVAL					

**Figure 44 – SOLAR\_TX\_INTERVAL register**

The encoding used by the SOLAR CELL INTERVAL bit field is shown in Table 30 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 30 – 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.

### 9.5.19 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 45 below.

LIGHT_THRESHOLD				
Bit 15	Bit 14	...	Bit 1	Bit 0
LIGHT SENSOR THRESHOLD				

**Figure 45 – LIGHT\_SENSOR\_THRESHOLD register**

The encoding used by the LIGHT SENSOR THRESHOLD bit field is shown in Table 31 below.

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

**Table 31 – LIGHT SENSOR THRESHOLD bit field encoding**

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

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

**9.5.20 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 46 below.

LIGHT_TX_INTERVAL					
Bit 15	Bit 14	...		Bit 1	Bit 0
LIGHT_SENSOR_INTERVAL					

**Figure 46 – LIGHT\_TX\_INTERVAL register**

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

<b>LIGHT SENSOR INTERVAL</b>	<b>Light sensor illumination-based reporting interval</b>
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 32 – 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.

### 9.5.21 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 51 below.

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

ACCELERATION_THRESHOLD					
Bit 15	Bit 14	...		Bit 1	Bit 0
ACCELERATION_THRESHOLD					

**Figure 47 – ACCELERATION\_THRESHOLD register**

The encoding used by the ACCELERATION\_THRESHOLD bit field is shown in Table 37 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 33 – ACCELERATION\_THRESHOLD bit field encoding**

### 9.5.22 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 52 below.

ACCELERATION_TX_INTERVAL				
Bit 15	Bit 14	...	Bit 1	Bit 0
ACCELERATION INTERVAL				

**Figure 48 – ACCELERATION\_TX\_INTERVAL register**

The encoding used by the ACCELERATION INTERVAL bit field is shown in Table 38 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 34 – 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.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.23 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 49 below.

TEMPERATURE_THRESHOLD					
Bit 15	Bit 14	...		Bit 1	Bit 0
TEMPERATURE THRESHOLD					

**Figure 49 – TEMPERATURE\_THRESHOLD register**

The encoding used by the TEMPERATURE\_THRESHOLD bit field is shown in Table 35 below.

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

**Table 35 – TEMPERATURE\_THRESHOLD bit field encoding**

### 9.5.24 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 50 below.

TEMPERATURE_TX_INTERVAL				
Bit 15	Bit 14	...	Bit 1	Bit 0
TEMPERATURE INTERVAL				

**Figure 50 – TEMPERATURE\_TX\_INTERVAL register**

The encoding used by the TEMPERATURE INTERVAL bit field is shown in Table 36 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 36 – 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.

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

### 9.5.25 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 51 below.

HUMIDITY_THRESHOLD					
Bit 15	Bit 14	...		Bit 1	Bit 0
HUMIDITY THRESHOLD					

**Figure 51 – HUMIDITY\_THRESHOLD register**

The encoding used by the HUMIDITY\_THRESHOLD bit field is shown in Table 37 below.

HUMIDITY_THRESHOLD	Threshold
0x0000	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 37 – HUMIDITY\_THRESHOLD bit field encoding**



### 9.5.26 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 52 below.

HUMIDITY_TX_INTERVAL				
Bit 15	Bit 14	...	Bit 1	Bit 0
HUMIDITY INTERVAL				

**Figure 52 – HUMIDITY\_TX\_INTERVAL register**

The encoding used by the HUMIDITY INTERVAL bit field is shown in Table 38 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 38 – 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.

### 9.5.27 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 52 below.

MAGNET_CONTACT_TX_INTERVAL				
Bit 15	Bit 14	...	Bit 1	Bit 0
MAGNET CONTACT INTERVAL				

**Figure 53 – MAGNET\_CONTACT\_TX\_INTERVAL register**

The encoding used by the MAGNET CONTACT INTERVAL bit field is shown in Table 38 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 39 – 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.

### 9.5.28 ILLUMINATION\_TEST\_RESULT

If STM 550 has executed an illumination test as described in chapter 2.4.5 then the result (the measured average light level at the solar cell) will be stored in the ILLUMINATION\_TEST\_RESULT register shown in Figure 54 below.

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

**Figure 54 – ILLUMINATION\_TEST\_RESULT register**

## 9.6 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 550 does not use this area and does not interpret its content in any way.

## **10 Installation recommendations**

### **10.1 Setup instructions**

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

To do so, follow these steps:

1. Place STM 550 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 550 will start operation
3. Check that STM 550 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 550 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 550 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 550 is ready for installation into its intended location.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 10.2 Acceleration detection

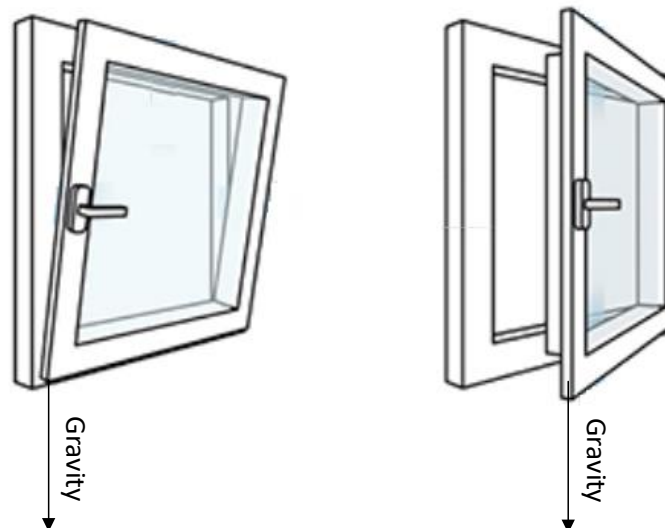
Acceleration detection enables to track the utilization of an asset (is a machine running, is a chair occupied, ...) based on the characteristic vibrations associated with this utilization.

Acceleration sensors – unlike gyroscopic or compass sensors – work based on changes of the acceleration vector. When an acceleration sensor (i.e. STM 550) 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 550 relative to the gravity vector changes.

Figure 55 below illustrates two cases:

- The position of the window in the left case (window tilt) could be detected if STM 550 is attached to the window part that is tilted since the orientation of STM 550 relative to the gravity vector would change
- The position of the window in the right case (window open / rotation) could not be detected if STM 550 is attached to the window part that is rotated since the orientation of STM 550 relative to the gravity vector would remain the same. STM 550 would however report that the window was moved due to the resulting acceleration and deceleration.

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



**Figure 55 – Tilt versus gravity vector**

## STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

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

- STM 550 should be firmly attached to the asset without any damping to ensure that any vibration of the asset will be properly propagated to STM 550
- STM 550 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

### 10.3 Temperature and humidity measurement

The dedicated temperature and humidity sensor integrated into STM 550 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. When designing your own housing around an STM 550 module, consider the location of the ventilation slots (there are four of them in total) as shown in Figure 2 and maximize the airflow towards them.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

## 10.4 Illumination measurement

STM 550 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 LIGHT\_SENSOR\_CFG register of the NFC interface as described in chapter 9.5.15. By default, the ambient light sensor is used.

### 10.4.1 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)

### 10.4.2 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 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**10.5 Magnet contact sensing**

If STM 550 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 STM 550 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 sensor.

Attach the magnet to the intended surface (e.g. door or window) such that the centre of the large side of the magnet is oriented towards the location of the magnet contact sensor in STM 550 and that the distance between magnet and STM 550 housing is less than 1 cm for the “Magnet Present” condition. Verify that the state (e.g. door open or closed) is reported as expected.

**10.6 Energy harvesting**

STM 550 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 550 should be oriented as much as possible towards that.

STM 550 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. STM 550 can operate for 4 days without available energy after being exposed to 200 lux for 2 days.

Lower levels of available light can be addressed by configuring a lower reporting rate via NFC as discussed in chapter 2.5.1. If the available light is insufficient, then STM 550 offers the option for a CR1632 backup battery as described in chapter 2.2.

The exact amount of available light can be determined by executing an illumination test as described in chapter 2.4.5.



## **10.7      NFC configuration**

STM 550 can be flexibly configured for a wide range of application scenarios using the NFC configuration interface as described in chapters 8 and 9.

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 2.5.1. 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 2.4.5 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.

STM 550 / EMSI – ENOCEAN MULTI-SENSOR FOR IOT APPLICATIONS

---

**11 Product history**

Table 40 below lists the product history of STM 550.

Revision	Release	Key changes versus previous revision
DA-04	Mar 2020	Initial release

**Table 40 – Product History**