
BLE RANGE
USER GUIDE

TABLE OF CONTENTS

1. GENERAL INFORMATION ABOUT BLUETOOTH LOW ENERGY

Bluetooth **Low Energy** technology is also called **LE** or **BLE Bluetooth**. This technology appeared in 2010 with the release of version 4.0 of the Bluetooth Core Specification.

Bluetooth Low Energy is an alternative to "classic Bluetooth". By "classic Bluetooth", we mean all versions of Bluetooth released before Core Specification 4.0.

Low Energy Bluetooth technology operates in the free band **ISM 2.4 GHz**. This technology relies on a **frequency hopping radio**. 40 physical channels are allocated and separated from each other by 2 MHz and used according to the FDMA. Three of them consist in **advertising channels** (they might be considered as signalization) and all the others are data channels. In contrast, conventional Bluetooth uses 80 channels separated from each other by 1 MHz.

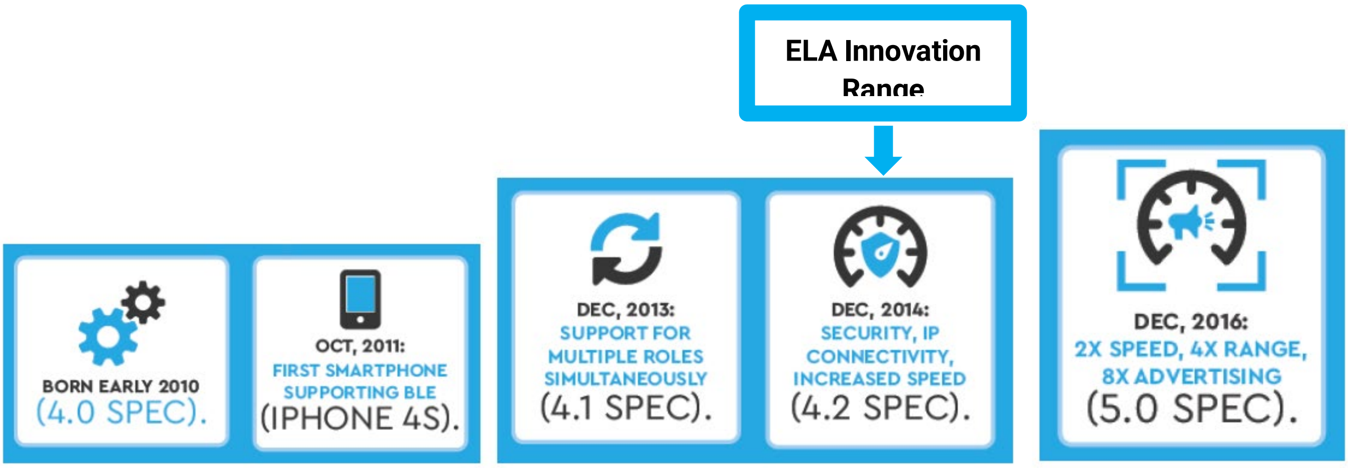


Figure 1: Evolution of Bluetooth Low Energy versions

Bluetooth SIG is the current standard in terms of information and specifications. The **Bluetooth Special Interest Group**, known as **SIG**, is the body that oversees the development of Bluetooth specifications, manages the various technology qualification processes and grants the needed licenses of the Bluetooth brand and technology to manufacturers.

Bluetooth SIG website	https://www.bluetooth.com/bluetooth-technology
BLE Specification	https://www.bluetooth.com/specifications
BLE Services and features	https://www.bluetooth.com/specifications/gatt

2. BLUETOOTH LOW ENERGY PRODUCTS BY ELA INNOVATION

DESIGNATION	*PRODUCT REFERENCE*	*DESCRIPTION*
Blue PUCK **ID**	IDF25240x	Tag Bluetooth PUCK Format with Identifier Option – iBeacon – Eddystone
Blue PUCK **BUZZ**	IDF25245x	Tag Bluetooth PUCK format with Identifier Option – Buzzer
Blue PUCK **T EN12830**	IDF30241x	Tag Bluetooth Format PUCK, integrated temperature sensor, EN12830 (2018) certified
Blue PUCK **T PROBE**	IDF25250x	Tag Bluetooth Format PUCK external temperature probe, EN12830 (2018) certified
Blue PUCK **RHT**	IDF25242x	Tag Bluetooth PUCK Format with humidity and temperature sensor option
Blue PUCK **MAG**	IDF25243x	Tag Bluetooth PUCK Format with magnetic sensor option
Blue PUCK **MOV**	IDF25244x	Tag Bluetooth PUCK Format with motion sensor option
Blue PUCK **PIR**	IDF25249x	Tag Bluetooth PUCK format with presence detection sensor
Blue PUCK **PROXIR**	IDF25252x	Tag Bluetooth PUCK format TOF infrared Ranging sensor
Blue PUCK **DI**	IDF24246x	Tag Bluetooth PUCK Format with digital input option
Blue COIN **ID**	IDF10240x	Tag Bluetooth Format COIN with Identifier option – iBeacon – Eddystone
Blue COIN **T**	IDF10241x	Tag Bluetooth COIN Format with temperature sensor option
Blue COIN **MAG**	IDF10243x	Tag Bluetooth COIN Format with magnetic sensor option
Blue COIN **MOV**	IDF10244x	Tag Bluetooth COIN Format with motion sensor option
Blue SLIM **ID**	IDF03240x	Tag Bluetooth SLIM Format with Identifier option– iBeacon – Eddystone
Blue LITE **ID**	IDF28240x	Tag Bluetooth LITE Format with Identifier option– iBeacon – Eddystone
Blue LITE **TOUCH**	IDF28242x	Tag Bluetooth LITE Format with a push-boutton
Aero ID	IDF10340X	Tag Bluetooth AERO Format with Identifier option– iBeacon – Eddystone

3. BLUE RANGE OPERATIONS BY ELA INNOVATION

3.1. REGULAR OPERATING MODE

- **Advertising Mode**

Frames are disseminated through "**Advertising**". Packets are sent periodically at a configurable recurrence comprised within the [0.1s; 10s] interval (firmware version < 4.0.0) and in the [0.1s; 86400 s] interval (firmware version > 4.0.0, (see section 2.5).

User data size is of 29 bytes. Data content are sensor information or fixed identifier, according to product (Identifier or Sensor). **For firmware version ≥4.0.0 advertising content and format can be fully customized on demand**

*See Frames Specifications document and section 5 of this document for more information on data sent and advertising frame format in "**Advertising**" mode.*

In some cases, a "**Scan Response**" frame may follow the "**Advertising**" frame:

- ✓ Battery level below 15%: battery level service available in the Scan Response section.
- ✓ A 15-character "Name" added in iBeacon or Eddystone UID format: "*Complete Local Name*" available in the "*Scan Response*" section.

- **Connected Mode**

The BLUE product range by ELA Innovation uses several functions in "Connected Mode". A link is set up between two devices and only these devices can communicate and exchange with each other.

You may establish a connection using a smartphone with a mobile application, or with a PC equipped with the ELA "*Device Manager*" application (provided you activated Bluetooth or connected a BLE dongle to the PC).

- ✓ Once you enter "*Connected Mode*", "***Advertising***" **is stopped by default**.
- ✓ It is possible to send commands to the tag to perform special actions or read data.
- ✓ It is possible to get a record of saved data (***Datalogger***) using *Connected Mode*. This datalogger will contain sensor data saved at a defined period with a timestamp for each data.

3.2. SPECIFIC OPERATION MODES

- **Fast advertising after NFC-field detection**

Starting from firmware version 3.0.0, the tag advertising period will be modified right after you approach an NFC-field to the tag.

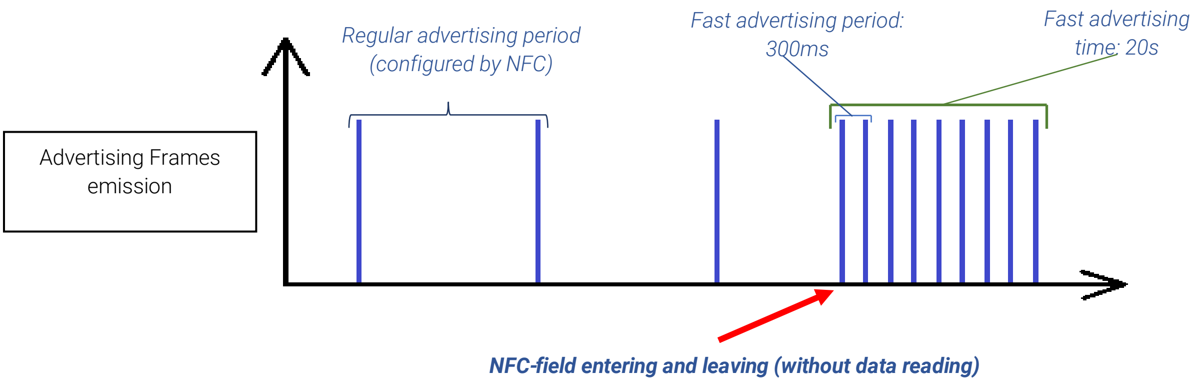


Figure 1 : Fast advertising after NFC Field detection chronograms

After 20 seconds, the advertising period will come back to its normal value if there is not any connection to the tag. There is no need to read the NFC memory to activate this function, any field leaving will trigger it.

This behaviour also happen when the tag reboot, after a reconfiguration for example.

This behaviour can allow easier connection to tags which advertising period is configured with a value greater than 3 seconds.

Note: *The advertising is stopped when the tag is on an NFC-field, thus the advertising will resume right after the tag leave the field.*

- **Long advertising period (FW vers. > 4.0.0)**

For tags with firmware vers. > 4.0.0, standard advertising period above 10 seconds are allowed. However, using such long advertising period makes cumbersome and, in some case, impossible establishing a connexion to the tag. To overcome this issue and for standard advertising period above 20 seconds, null payloads frames are sent during 10 sec and with a period of 1 sec.

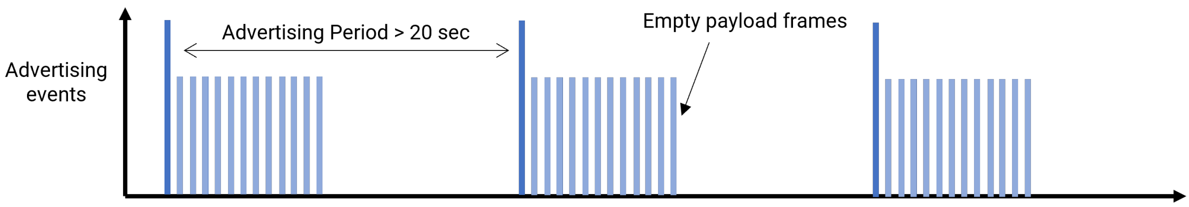


Figure 2 : Long advertising period chronograms

- **Dynamic Advertising periods**

ON EVENT (Legacy)

Tag with formats **MAG**, **MOV**, **PIR** and **DI** and firmware version < 4.0.0 version provide the **fast on-event frame functionality**.

- This frame sends data with **faster recurrence** (equal to one tenth of the advertising tag recurrence set in NFC). Data contained in this frame is the same as that contained in the simple advertising frame, but its recurrence varies.

- **Fast frames** appear during a period equal to the advertising period, and with a recurrence equal to one tenth of it. Thus, there are **10 frames**.

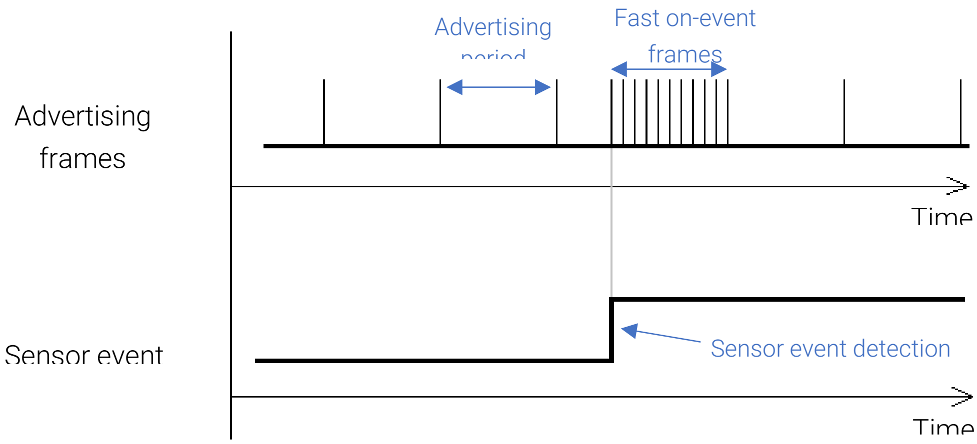


Figure 3 : Fast on-event advertising

These **fast frames** emission takes place at each sensor event:

- For **MAG format**: With each new magnet detection state (present and absent)
- For **MOV format**: At the beginning and end of each movement (depending on the submitted threshold
- For **DI format**: With each new digital input state (logical state 1 or 0)
- For **PIR format**: with each movement detected (sensor state stay at 1 during movement and falls to 0 few seconds after last movement)

In addition to these fast frames, the MAG, MOV and DI formats data also contains an event counter. This counter is incremented at each “rising edge” event detected by the sensor:

- For **MAG format**: With each new magnet detection (magnet present)
- For **MOV format**: At the beginning of each movement (depending on the submitted threshold)
- For **DI format**: With each new logical state 1 of digital input (input shorted)
- For **PIR format**: With each new infrared movement detected

The counter overflow value is 32767 (maximum counter value before reset to zero). The counter resets when a *Connected mode* command ***“RAZ_COUNT”*** is sent, or when the tag reboot.

EXTENDED DYNAMIC ADVERTISING (FW Vers. >4.0.0)

Starting from FW Vers. 4.0.0, *On-event fast advertising is extended to all ELA INNOVATION Tags with sensing capabilities:* ***T, T EN12830, T PROBE, RHT, MOV, ANG, PIR, PROXIR, DI, AI.***

In addition, this functionality is also augmented with the possibility to define more than one sensor threshold plus the possibility to operate either in **burst** or **lasting mode** (see Section 4.3).

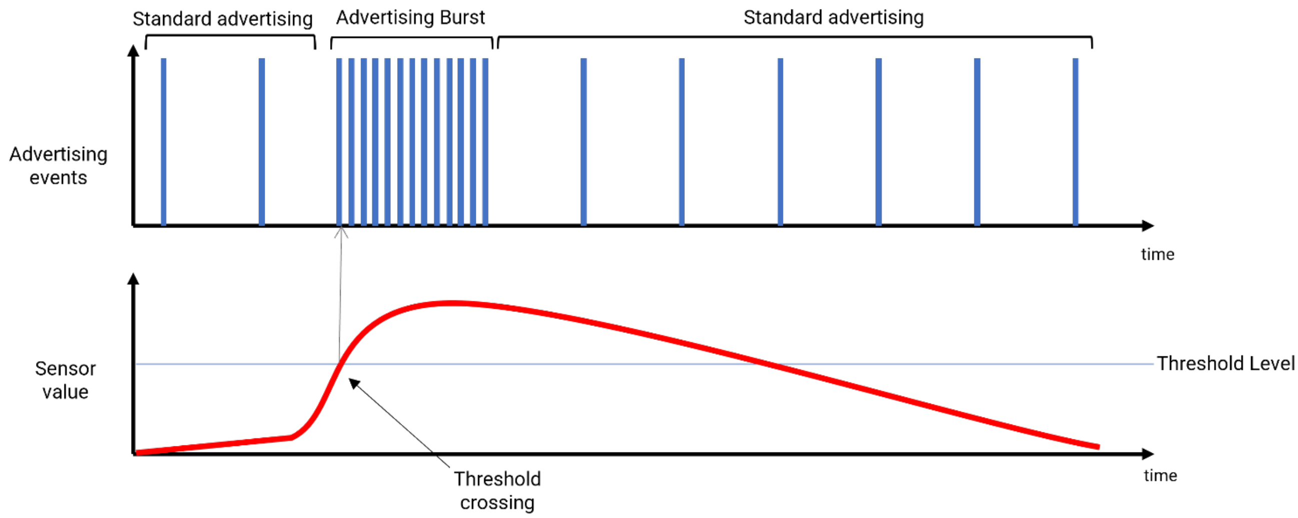


Figure 4 : Advertising chronograms for on-event burst mode

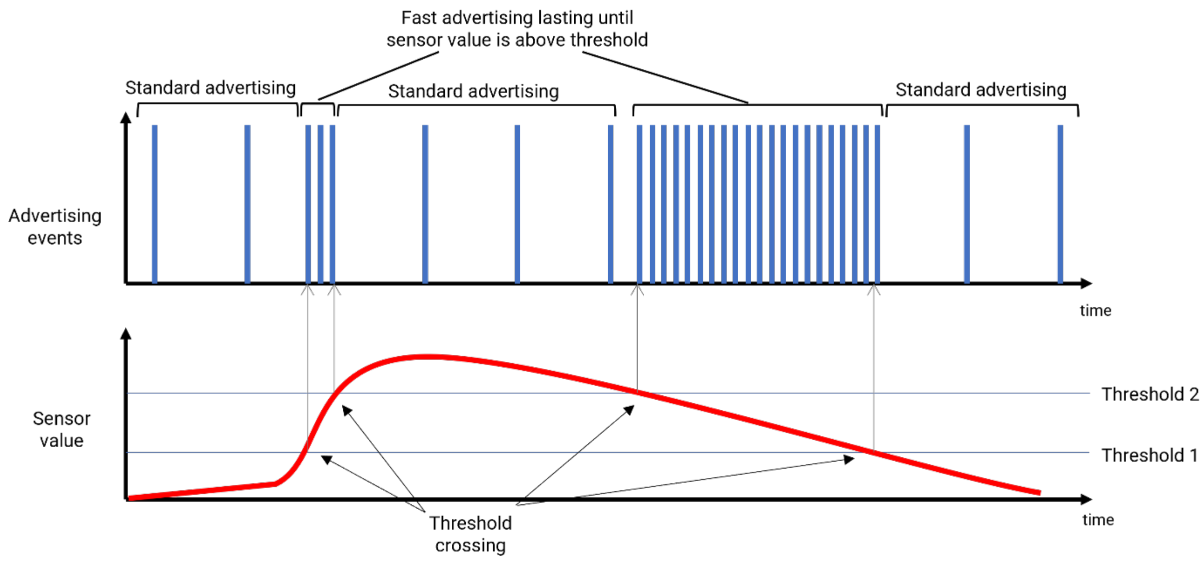


Figure 5 : Advertising chronograms for lasting mode with two threshold

4. ELA INNOVATION BLUE RANGE CONFIGURATION

4.1. GENERALITIES

ELA Innovation BLUE range products are equipped with a NFC chip used for tag configuration. This chip, used with Device Manager suite, allows to write operating parameters, among the following fields:

PARAMETER	POSSIBLE VALUES	ACTION	AVAILABILITY
Name	Maximum 15 characters [0-9 ; A-Z ; a-z ; SPACE, _, -]	Definition of the tag *Name*, transmitted by *Advertising*	Complete *Blue* range
Enable	True / False	*True*: Enable product operation. *False*: Turn OFF the product.	Complete *Blue* range
Power	[-40, -20, -16, -12, -8, -4, 0, +3, +4]	Definition of the product BLE power emission, unit is dBm	Complete *Blue* range
Format	[Id, T, RHT, MAG, MOV, ANG, iBeacon, Eddystone, Analog IN, Digi IN, Digi OUT, PIR, PROXIR]	Definition of Firmware operation	According to product
(Main) Advertising Period	[0.1 -> 10] in s or [100 - 86400000] in ms	Standard duration between two consecutive advertising events	Complete *Blue* range (extended value only available in firmware vers. ≥ 4.0.0
UUID (iBeacon)	32 characters [0-9 ; A-F]	Definition of iBeacon *UUID*, transmitted by *Advertising*	*Id* products with iBeacon
Major (iBeacon)	4 characters [0-9 ; A-F]	Definition of iBeacon *Major*, transmitted by *Advertising*	*Id* products with iBeacon
Minor (iBeacon)	4 characters [0-9 ; A-F]	Definition of iBeacon *Minor*, transmitted by *Advertising*	*Id* products with iBeacon
NID (Eddystone)	20 characters [0-9 ; A-F]	Definition of the Eddystone *NID*, transmitted by *Advertising*	*Id* products with Eddystone
BID (Eddystone)	12 characters [0-9 ; A-F]	Definition of the Eddystone B*ID*, transmitted by *Advertising*	*Id* products with Eddystone
Measurement period	[100 -> 86400]	Sensor Data acquisition period	*Blue* products with sensors and firmware vers. ≥ 4.0.0
Data Logger Period	[100 -> 86400]	Definition of the sensor data saving period for *datalogger* feature	*Blue* products with sensors
Data Logger Enable	True / False	*True*: Enable datalogger feature. *False*: Disable datalogger feature.	*Blue* products with sensors
Accerleration threshold	[32 ;8000]	Definition of the acceleration threshold for MOV format, unit is mg	*MOV* products Firmware vers < 4.0.0
PIR sensor sensitivity	[0,1,2,3]	*Sensitivity level for the PIR sensor: define the maximum detection distance 0: 50cm 1: 1m 2: 2m 3 :5m*	*PIR products with firmware vers. >3.0.1*

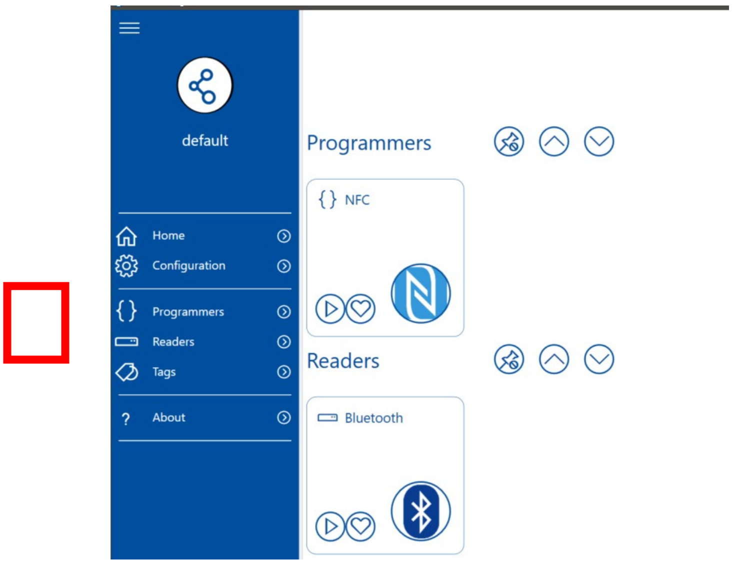
Mfr. Data Enable	True / False	*True*: Enable data transmission in ***Manufacturer Specific Data*** mode. *False*: Enable data transmission in ***Service Data*** mode.	Complete *Blue* range
MFR. ID	12 characters [0-9 ; A-F]	Definition of an hexadecimal identifier used in Id format when ***Manufacturer Specific Data*** are enabled.	Only used in *Id* products
Battery in Scan Response	True/false	*True* : Enable ***Battery voltage transmission into Scan Response*** frame. See related section of this document.	All products with *firmware vers.* ≥ 3.0.0
Advertising period 2	[100 - 86400000] in ms	Duration between two consecutives on sensor event advertising events	*Blue* products with sensors and firmware vers. ≥ 4.0.0
Sensor Threshold 01	[Min Max] of corresponding sensor value with resolution	Sensor threshold for event counter and on-event advertising. Value can be set between min and max sensor value and according to sensor resolution	*Blue* products with sensors and firmware vers. ≥ 4.0.0
Sensor Threshold 02	[Min Max] of corresponding sensor value with resolution	Secondary sensor threshold for on-event advertising. Value can be set between min and max sensor value and according to sensor resolution	*Blue* products with sensors and firmware vers. ≥ 4.0.0
Transmit Mode	[DT0 – DT1 - DT2]	**DT0**: static advertising period **DT1**: enable on-event advertising mode **DT2**: Enable lasting advertising mode	*Blue* products with sensors and firmware vers. ≥ 4.0.0
Burst Duration	[100, 86400000]	Duration of On-event advertising (Transmit Mode = DT1)	*Blue* products with sensors and firmware vers. ≥ 4.0.0
Edge Detect 01	[inf. Threshold, Supp. Threshold, Equal Threshold, Both Threshold]	Definition of threshold crossing rule for of on-event advertising mode	*Blue* products with sensors and firmware vers. ≥ 4.0.0
Edge Detect 02	[[inf. Threshold, Supp. Threshold, Between Threshold, Excluded Threshold	Definition of threshold crossing rule for lasting advertising mode	*Blue* products with sensors and firmware vers. ≥ 4.0.0
Frame type	[ELA Id, iBeacon, Eddystone, T, RHT, MAG, MOV, ANG, Analog IN, Digi IN, Digi OUT, PIR, PROXIR, Custom]	Definition of BLE data sent	All products with *firmware vers.* ≥ 4.0.0

4.2. BLUE RANGE TAG COFIGURATION WALK-THROUGH EXAMPLES

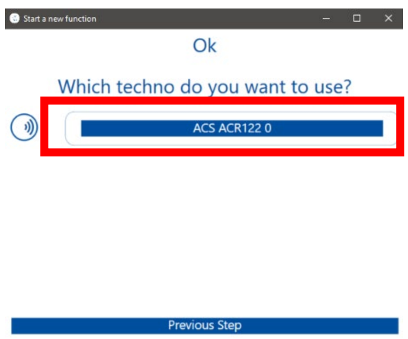
4.2.1. Tag configuration using device manager PC SOFTWARE

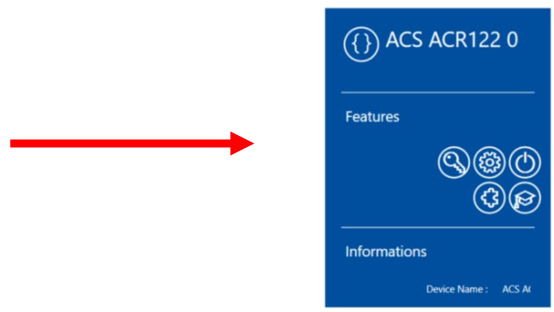
1. **Connect a NFC reader** to your desktop (example: NFC R/W 01 - ref. ACIOM177)
2. Start the "**Device Manager**" of your desktop

3. On the welcome main pannel click on the **"PLAY" icon** of the **widget « NFC »**



4. Choose the available **NFC reader** by **clicking** on the button 

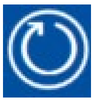


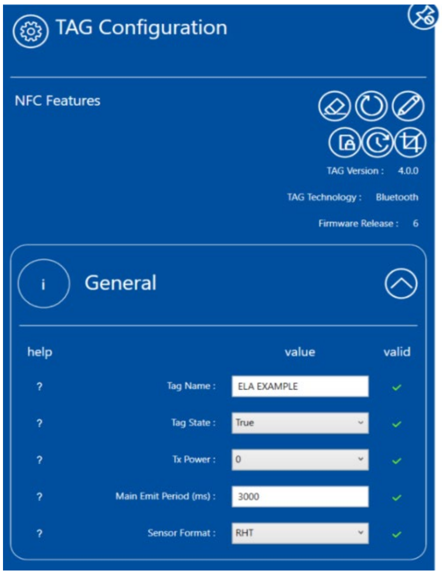


Once reader is selected, this window appears

5. Place the tag on the NFC reader

6. Click on the "***Configuration***" pictogram  to bring up the tag configuration window:

7. Click on « Refresh »  to bring up the current configuration read from the tag.



 Erase setting (default configuration)

 Read and refresh current tag configuration

 Write current configuration to the tag

 Display Tag security option (when

 Display Data Logger options

 Display calibration options

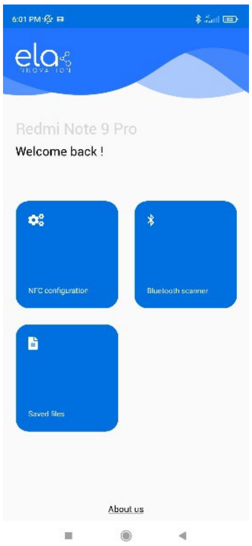
 Lock / Unlock tag configuration

 Enable / Desable Tag (batch operation)

4.2.2. Tag configuration using a smartphone

4.2.2.1. Turn ON / OFF a tag

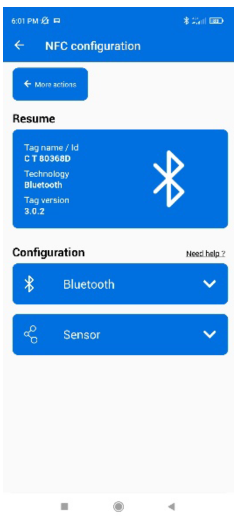
If you have a tag and you wish you turn it ON or OFF quickly, open **Device Manager Mobile**, then choose Configuration. Once in the configuration page, approach the top of the tag to the bottom of your smartphone to detect the NFC parameters. If you are using the iOS version, you will first need to choose “Scan” before reading the settings.

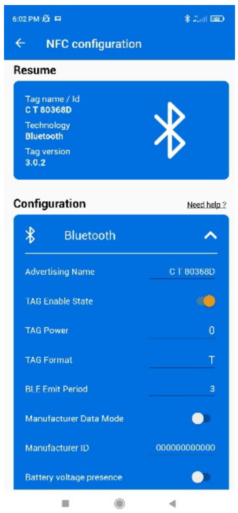


After you successfully read the tag state and parameters, you can choose “Quick actions” to quickly turn ON or OFF the tag. After clicking on one of these 2 actions, you will need to scan the tag again with the NFC chip to apply the modification.

4.2.2.2. Change tag settings

If you are willing to change some settings in the tag such as the emit power, the format or to turn ON/OFF the datalogger, you need to open **Device Manager Mobile** with the NFC activated on your smartphone, then choose configuration. Once in the configuration tab, bring the tag close to the bottom of your cell phone and remove it after it has been read by the NFC chip. The current state of the tag will be shown. Choose tag parameters to enter the settings menu.





To apply any modification, choose **Write**, then scan the tag with the NFC chip again.

4.2.3. Settings Restriction

4.2.3.1. Restriction applying to "Name" field

✓ Name must include **less than or up to 15 characters**

✓ Name **should not contain special characters** (but rather only letters, numbers, spaces, dash - and underscore _).

4.2.3.2. Datalogger restrictions

✓ When the **"Logger Enabled"** field of the NFC settings located under the device manager is **disabled**, the tag reboots and you will **lose all registered data** contained in the data logger.

✓ If you proceed to a **complete re-setting** of the tag by NFC, data **contained in the data logger is erased** from the tag memory.

4.2.3.3. Connected mode restrictions

✓ If the tag is connected to a device and is approached by a NFC field, the tag will disconnect and reboot.

4.2.3.4. Other restrictions

- **iBeacon format**

✓ You must fill in the complete UUID field of the iBeacon format: 32 characters ([0-9]; [A-F]).
✓ You must fill in the complete Major field of the iBeacon format: 4 characters ([0-9]; [A-F]).
✓ You must fill in the complete Minor field of the iBeacon format: 4 characters ([0-9]; [A-F]).

- **Eddystone format**

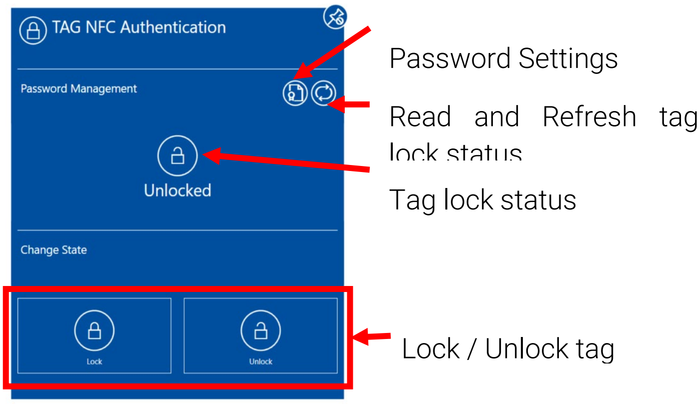
✓ You must fill in the complete NID field of the Eddystone format: 20 characters ([0-9]; [A-F]).
✓ You must fill in the complete BID field of the Eddystone format: 12 characters ([0-9]; [A-F])

4.3. SPECIFIC CONFIGURATION

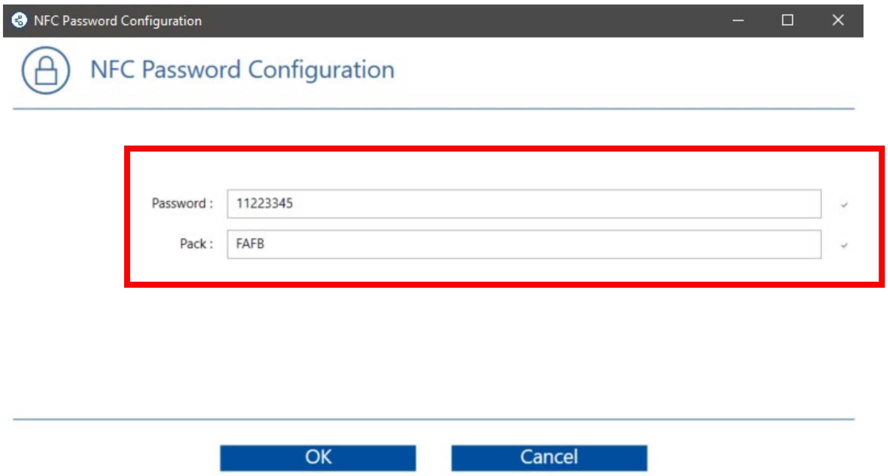
- **TAG NFC CHIP PASSWORD PROTECTION**

Starting from firmware version 2.1.0, it is possible to protect the tags NFC-chip writing by a password. The functionality is accessible on ***Device Manager***, on the ***Programmers*** section:

1. Display the tag authentication window by clicking on the Key pictogram: 



2. Click on **Define password** 

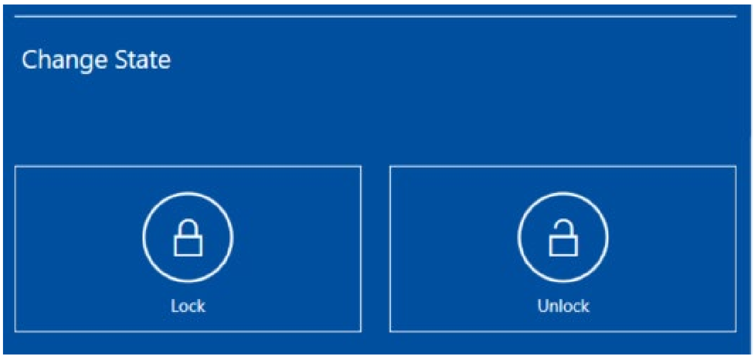


- NFC password must have exactly **8 hexadecimal characters.**
- Pack NFC validate the authentication of the tag but has no consequence on the password modification. It is recommended to leave it to its defaults value: 0x***FAFB***.

Note: It is not possible to recover a lost password. If you forgot your password, it will be necessary to return the product to ELA Innovation.

3. Click on **Update state** to read the current lock status of the tag (locked / unlocked)

Into the ***Change State*** area, the transition from a *Lock/Unlock* or *Unlock/Lock* state is done by clicking on the ***Lock*** or ***Unlock*** icons:



Note: If the password set in step 2. Is not correct, the Unlock command will have no effect on the tag

- **TEMPERATURE CALIBRATION NFC CONFIGURATION**

Starting from firmware version 3.0.0, it is possible to configure a 2^{nd}-polynomial calibration that can be used to correct temperature sensor value, to adjust measurement precision.

General Information

Calibration uses a 2^{nd} order polynomial correction formula **a**T^2+**b**T+**c**, where ***a***, ***b*** and ***c*** are configurable coefficients (T being the original temperature value measured by the sensor). These coefficients **can only be written via NFC and read in *connected mode***.

They are transmitted in the format ***X***e***Y***, where ***X*** is an integer between 32768 and 32767, followed by a exponent ***Y*** from -128 to 127. ***X***e***Y*** is equivalent to X.10^Y. Examples:

- 125e-5 = 0.00125
- 1e-2 = 0.01
- 12e-1 = 1.2

Examples of complete calibration procedure:

✓ Sensor reading before calibration = 25.00°C. Calibration polynomial [c, b, a]: [5e-1, 1e0, 0e0]. The corrected value is therefore: T_{cal} = 25.5°C

✓ Sensor reading before calibration = 25.00°C. Calibration polynomial [c, b, a]: [0e0, 101e-2, 0e0]. The corrected value is therefore: T_{cal} = 25.25°C

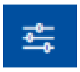
Enabling / disabling sensor value connection **can only be done via NFC**.

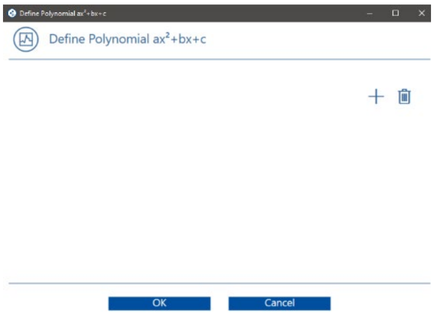
NFC Configuration

The fields for calibration and the calibration report are configured using the Device Manager application.

The window for configuring the fields is accessible via the “***Calibration***” icon. 

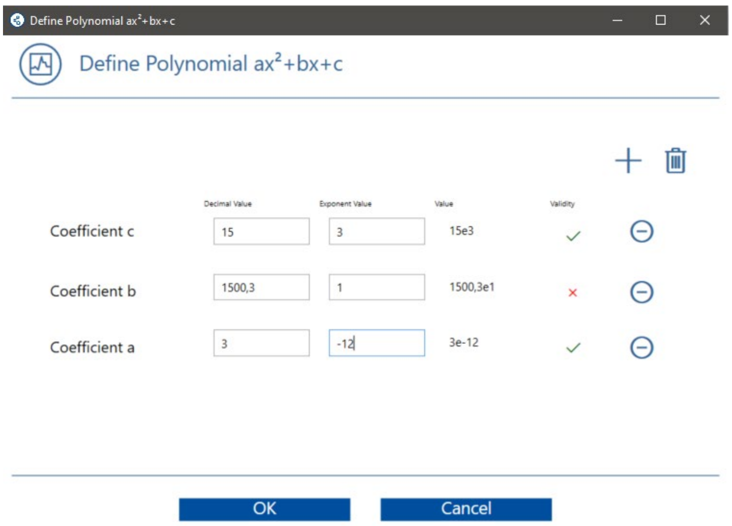
Here you can enable the calibration and also configure polynomial coefficients.

Click on the  button to pop the Polynomial value configuration window :

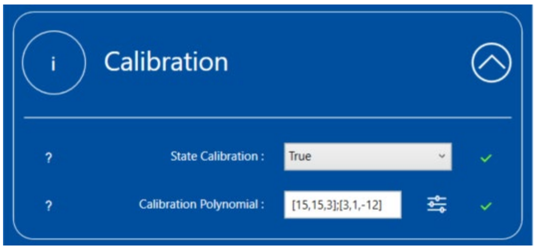


You can add a calibration coefficient by clicking on the button.

The window will check the coefficients and exponent values for integrity.



Click on ***OK*** when the values are set. They will appear on the previous window:



Do not forget to write the NFC configuration to the tag. 

- **INTELLIGENT MODE ADVERTISING SENSING and DATALOGGING (FW vers. > V4.1.0)**

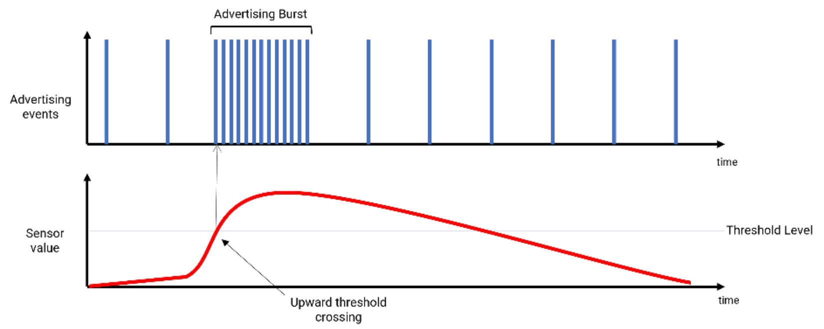
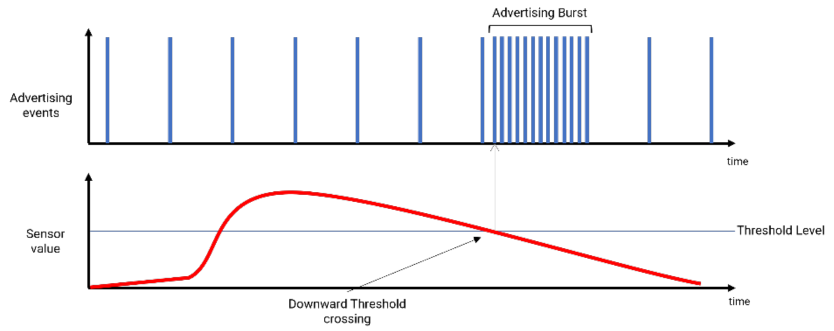
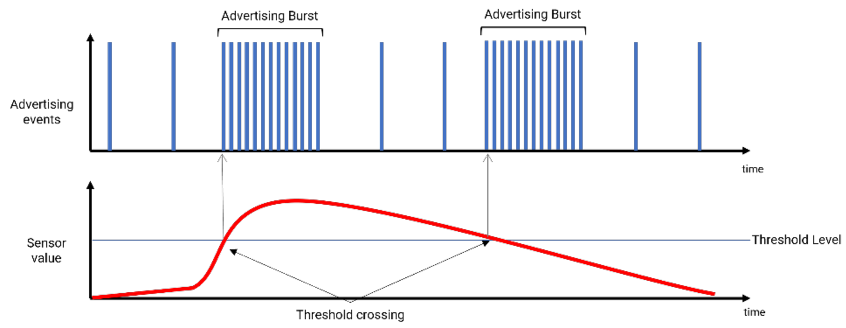
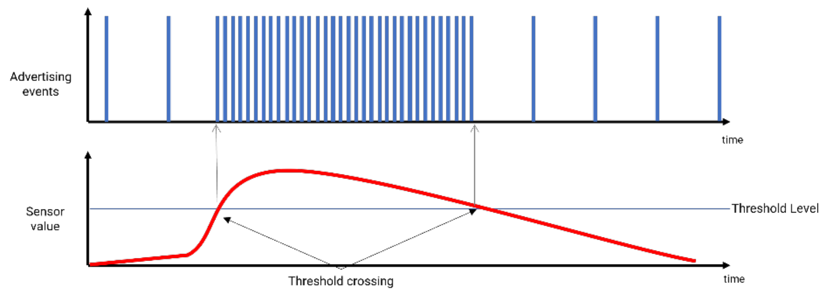
Starting from firmware version 4.0.0, it is possible to configure 2 periods for each of the three periodic processes in the tag (advertising, datalogging and sensor reading) and define transition rules according to sensor value. Using this feature is useful to extend device battery life (effectively active when it is needed, low power consumption when not) or extend the duration of recording with datalogger (4000 entries ring buffer)

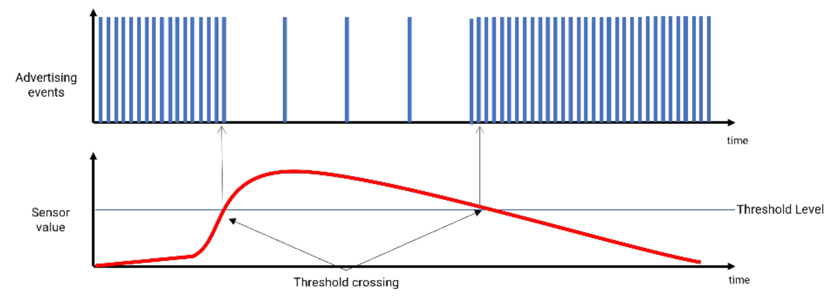
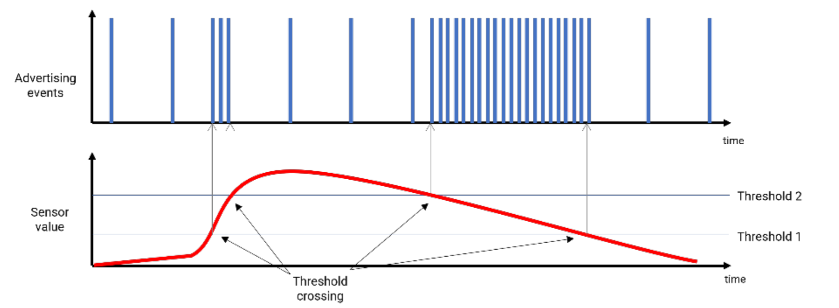
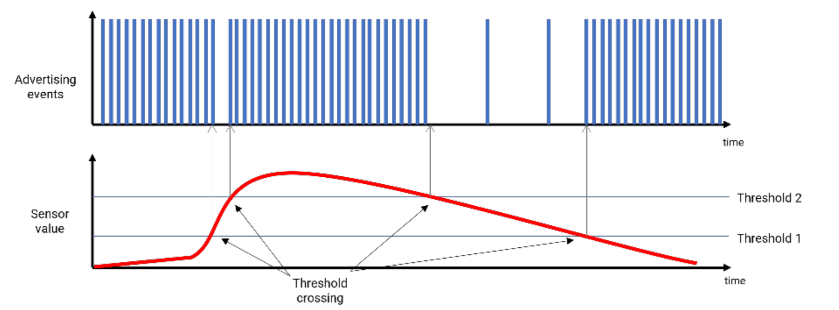
General Information

Two dynamic mode are defined: On-event mode (DT1) and lasting mode (DT2).

- For on-event mode, transition from base periods to periods 2 happens each time the sensor value crosses a threshold (configurable) and during a configurable duration.
- In lasting mode, the transition from base periods to periods 2 happen until the sensor value is above or below a threshold (configurable) or between or outside an interval define by two thresholds (configurable).

Following table gives advertising chronograms for the different configured transition rules for the advertising. Exaclty same process happens with the datalogging period and the sensor reading period.

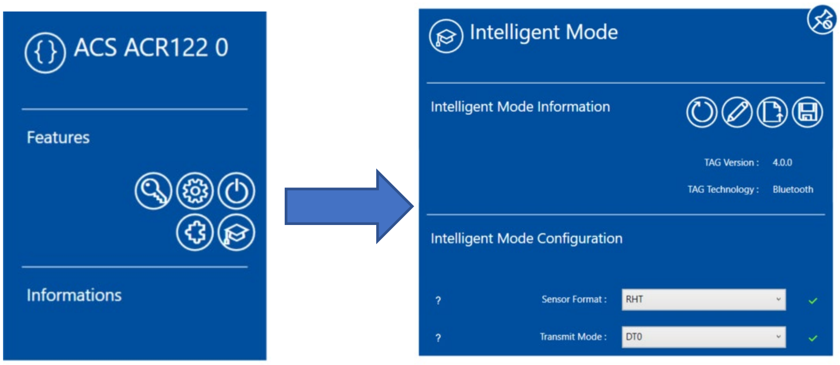
On Event Mode (DT1)	**Sensor value crossing direction (Edge detect 01)**	
	Upward (Inf. Threshold)	
	Downward (Supp. Threshold)	
	Upward & downward (both Threshold)	
Lasting Mode (DT2)	**Sensor value with respect to threshold (Edge detect 02)**	
	Above (Supp. Threshold)	

	Below (Inf. Threshold)	
	Between (Between threshold)	
	Outside (Exclude threshold)	

NFC Configuration

The second advertising period, dynamic advertising mode, threshold level and sensor signal to considered can be configured using Device Manager.

The window for configuring the fields is accessible via the "***Intelligent Mode***" icon. 



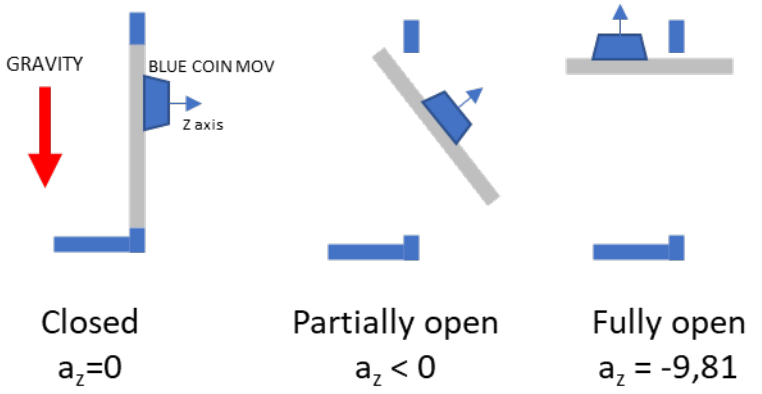
1. Click on the incon to read and refresh intelligent mode configuration
2. Select the desired sensor format and transmit mode (DT0: static advertising, DT1: On-event mode, DT2: Lasting mode
3. Select the desired sensor value signal to consider and set the second advertising period / data logging period / sensor period
4. Select the sensor value crossing direction (Edge detect 01) or the sensor value condition with respect to threshold
5. Set thresholds values (and burst duration for on-event mode only)
6. Click the icon to write the current configuration to the tag

Example of uses

Use case 1: Tilting doors openings monitoring

Tilting doors are very common in trailers and several applications require monitoring their opening position:

- Detection of unauthorized opening for which alerts must be sent as soon as the doors is no more in its closed position
- Driving security for which an alerts must be sent until the doors is not properly closed (and prevent the driver to start driving)



One way to detect its opening position is to measure the angle between the acceleration of gravity and the door plan. This can be done using a Blue COIN MOV with ANG format and by looking at the z axis acceleration value.

In order to save power and reduce the number of advertising message sent, the following parameters of intelligent mode can be used:

Parameter	Value (detection of unauthorized opening)	Value (Driving security)
Tag Main Configuration Parameters		
Sensor Format	ANG	ANG
Main emit period	3600 sec	3600 sec
Measurement period	1 sec	1 sec
Intelligent Mode Parameters		
Transmit Mode	DT1	DT2
Detection sensor info	ANG Z	
Emit period 02	1 sec	1 sec
Measurement period 02	1 sec	1 sec
Event Tx Time	10 sec	NA
Edge detect	Both Threshold	Inf threshold
Sensor threshold	0	0

For the detection of unauthorized opening case, the doors is first in the closed position and the tag measured a z axis acceleration of 0 value it advertise a keep alive message every 1 hour. When someone tries to open the door the z-axis acceleration goes below 0 and the tag start sending message every 1 sec for 10 seconds. When the doors is closed again the sensor sends another burst of 10 advertising frames.

For the Driving security case, when the door is closed the sensor sense a z axis acceleration of 0 value and advertise a keep alive message every 1 hours also. However, when someone tries to open the door and until the doors is closed again, the tag sends advertising message every 1 seconds.

In slow mode, the life span of the Blue COIN MOV (ANG mode) is 5 years while in the fast-advertising mode the life span is only 1 years. Depending on the duration spend in fast advertising mode the lifespan of the device can be extended up to a factor of 5.

Use case 2: Cold chain alerts

Temperature sensitive goods, such as medicine, must be transported in temperature regulated environment between 2°C to 6°C. Outside the temperature range active substances may be deteriorated: fully destroyed if the temperature goes below 2°C and with a reduce expiry date if the temperature goes above 6°C (the reduction being proportional to the temperature excursion and the time spent outside the recommended range).

It is thus necessary to keep track of the temperature during transportation with a system that:

- Increase advertising rate when temperature goes outside the recommended temperature range => ALERT
- Log with sec resolution the time spend outside the recommended temperature range to re-evaluate expiry date

Using a Blue COIN T with the following configuration meets these needs:

Parameter	Value
Tag Main Configuration Parameters	
Main emit Period	60 sec
Measurement Period	10 sec
Data Logger period	3600 sec
Intelligent Mode Parameters	
Transmit mode	DT2
Emit Period 02	1 sec
Measurement Period 02	1 sec
Data Logger Period 02	1 sec
Edge Detect	Exclude Threshold
Sensor Threshold 01	2°C
Sensor Threshold 02	6°C

When the temperature is in the 2°C-4°C range the tag performs a temperature measurement every 10 second, transmit one advertising frame every minute and saves a temperature value every one hour. When the temperature goes outside the 2°C-4°C range, the tag read a temperature measurement, save it and send an advertising frame every second.

Similary to the previous example, the life span of a tag in the slow operation mode is 5 years (Coin format) compared to 1 year in the fast operation mode. Besides the datalogging duration is drastically increased.

5. FRAME FORMAT AND CONTENT

5.1. GENERALITIES

BLE protocol fixes the length of BLE packets to 47 Bytes maximum among which a maximum of 37 are define by the user and are generally referred as the **payload**.

In this payload, the 6 first bytes are reserved for the advertiser address (mac address), the following 3 are used to flag the type of advertising frame and the 29 remaining contains actual data of interest. Identification of the data types can follow "**Service Data"** identification standardized by BLE specification or can be customs using "**Manufacturer Specific Data**" types.

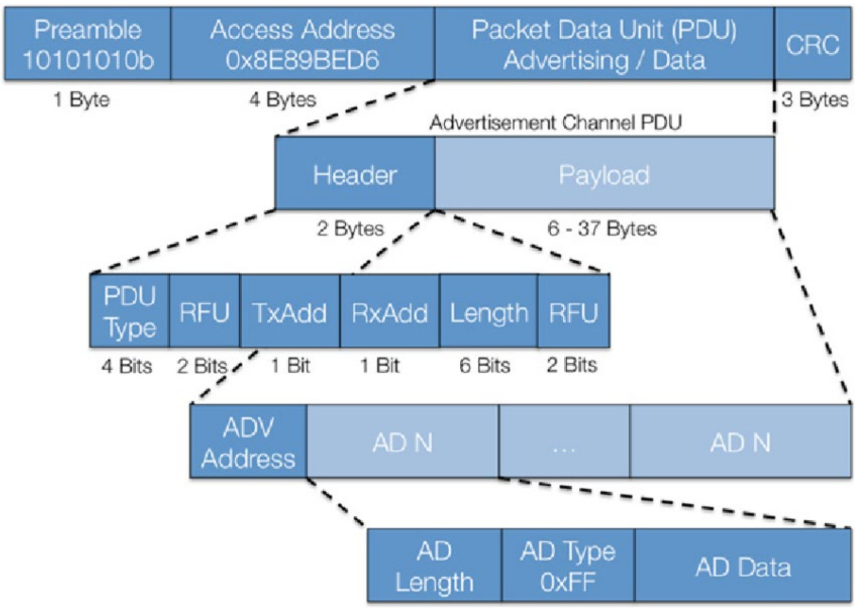


Figure 6 : Advertising frame format as per BLE SIG specifications

In Firmware version $< 4.0.0$, several pre-defined frame formats are available to the user (legacy frame format) depending on the configured tag format. Such frame are described in detail in the BLE frame specification document available on ELA website. "**Service Data**" type identification is used as a standard for interoperability with generic BLE scanner but **"Manufacturer Specific data"** type identification frame format are also available.

In Firmware version $\geq 4.0.0$, fully customized frame format are also available on demand, legacy frame format being available by default. In these custom frame formats the value / data of each byte can be specified by the user and configured at ELA Innovation factory.

5.2. SENSOR DATA IN "SERVICE DATA" FRAME (Legacy)

- « T », « T EN » and « T Probe » formats example:

Raw data:
0x0201060516 6E2A AB 0A 1009425055434B5354 3830304131324E41

0x6E2A : *Temperature* service

T° data:
- 0xAB : LSB
- 0x0A : MSB

T° = 0AAB = 2731 * 0.01 = 27.31°C

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x6E2AAB0A
10	0x09	0x425055434B53543830304131324E41

Name (ASCII)

Note: For a negative temperature, data is sent in 2-complement: for example, -27.31°C is 6E2A55F5

- « MAG » format example:

Raw data:
0x0201060516 062A FB 0A 1009425055434B535438 30304131324E41

0x062A : *Alert Status* service

MAG data:
- 0xFB : LSB
- 0x0A : MSB

Hexa.	0	A	F	B
Binary	0000	1010	1111	1011

⇨ 1: instantaneous sensor state (magnet present)

⇨ 1010 1111 101: event counter value on 15 bits, 1405 in this example

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x062AFB0A
11	0x09	0x425055434B53543830304131324E41

Name (ASCII)

- MOV » format example:

Raw data:
0x0201060516 062A FB 0A 1009425055434B53543 830304131324E41

0x062A : *Alert Status* service

MOV data:
- 0xFB : LSB
- 0x0A : MSB

Hexa.	0	A	F	B
Binary	0000	1010	1111	1011

⇨ 1 : instantaneous sensor state (movement detection)

⇨ 1010 1111 101 : event counter value on 15 bits, 1405 in this example

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x062AFB0A
11	0x09	0x425055434B53543830304131324E41

Name (ASCII)

- « ANG » format example:

Raw data:
0x0201060516A12A05FF0AFBC90755434B53543830304131324E41

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0xA12A05FF0AFBC907
11	0x09	0x55434B53543830304131324E41

Name (ASCII)

0xA12A : *Magnetic 3D* service
ANG data:

- 0x05 : X-axis LSB
- 0xFF : X-axis MSB
- 0x0A : Y-axis LSB
- 0xFB : Y-axis MSB
- 0xC9 : Z-axis LSB
- 0x07 : Z-axis MSB

X-axis : 0xFF05 => -251 mg acceleration on X-axis
Y-axis : 0xFB0A => -1270 mg acceleration on Y-axis
Z-axis : 0xFF05 => +1993 mg acceleration on Z-axis

Note: Values are coded on 16-bits with 12 significant bits and 4 sign bits. The values are expressed in mg (+2g/-2g). Negative data are sent in 2-complement.

- « RHT » format example:

Raw data:
0x0201060516 6E2A5E0A004166F2A301009425055434B53543830304131324E41

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x6E2A5E0A004166F2A30
11	0x09	0x55434B53543830304131324E41

Name (ASCII)

0x6E2A : *Temperature* service
0x6F2A : *Humidity* service

RH data:

- 0x30 : RH data i.e. 48% relative humidity

T° data:

- 0x5E : LSB
- 0x0A : MSB

T° = 0A5E = 2654 * 0.01 = 26.54°C

Note: For a negative temperature, data is sent in 2-complement

- « DI » format example:

Raw data:

0x0201060516062A0A0004163F2A020E09424 55F544553545F544F52494E

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x062A0A00
11	0x09	0x42455F544553545F544F52494E

Name (ASCII)

0x062A : *Alert Status* service

DI data:

- 0x0A : LSB
- 0x00 : MSB

Hexa.	0	0	0	A
Binary	0000	0000	0000	1010

⇨ 0 : instantaneous input state (input state OFF)

⇨ 0000 0000 0000 101 : event counter value on 15 bits, 1405 in this example

Note: In this example, the *DI* data is 0x000A, i.e. counter is at 4 increments (transition from state 0 to state 1 on digital input), and the instantaneous input state is 0 (input in OFF state).

- « Analog IN » format example:

Raw data:
0x0201060516582AAC0B1009425055434B5354 3830304131324E41

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x582AAC0B
11	0x09	425055434B53543830304131324E41

Name (ASCII)

0x582A : *Analog Output* service

Analog IN data:

- 0xAC : LSB
- 0x0B : MSB

Analog voltage measure: 0x0BAC = 2988mV

Note: Analog input voltage measure are in mV unit.

- « PIR » format example:

Raw data:
0x0201060516782A1B001109425055434B5354 3830304131324E41

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x782A1B00
11	0x09	425055434B53543830304131324E41

Name (ASCII)

0x282A : *Rainfall service*

PIR data:

- 0x1B : LSB
- 0x00 : MSB

Hexa.	0	0	1	B
Binary	0000	0000	0001	1011

⇨ 1: instantaneous sensor state (infrared movement detected)

⇨ 0000 0001 101: event counter value on 15 bits, 13 in this example

- « TOUCH » format example:

Raw data:
0x0201060516832A1B001109425055434B5354 38303041313234E41

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x832A1B00
11	0x09	425055434B535438303041313234E41

Name (ASCII)

0x832A : *Altitude service*

TOUCH data:
- 0x1B : LSB
- 0x00 : MSB

Hexa.	0	0	1	B
Binary	0000	0000	0001	1011

⇨ 1: instantaneous sensor state (press button pressed)

⇨ 0000 0001 101: event counter value on 15 bits, 13 in this example

- « PROXIR » format example:

Raw data:
0x02010605168E2A1B001109425055434B5354 38303041313234E41

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x8E2A1B00
11	0x09	425055434B535438303041313234E41

Name (ASCII)

0x282A : *Rainfall service*

PROXIR data:
- 0x1B : LSB
- 0x00 : MSB

Hexa.	0	0	1	B
Binary	0000	0000	0001	1011

⇨ 1: instantaneous sensor validity (not valid)

⇨ 0000 0001 101: distance to target in mm on 15 bits, 13 mm in this example

5.3. SENSOR DATA IN "MANUFACTURER SPECIFIC DATA" FRAME (Legacy)

1) ELA Innovation *Company Identifier* (*CIN*) is 0x0757.

2) In *ELA_ID* and *Digi OUT* formats, it is possible to configure a hexadecimal number (max. 0xFFFFFFFFFFFF) which will be sent in advertising frame. This field is named "***Manufacturer Data ID***" in the NFC configuration. This number is called "MFR_ID" in this document frame formats.

In « *Manufacturer Specific Data* », sensor data are encoded the same way as in "Service Data" mode, only the "data type" in hexadecimal is modified:

- 0x16: for « ***Service data*** »
- 0xFF: for « ***Manufacturer Specific Data*** »

Manufacturer Specific Data are transmitted as follows:

ELA CIN	DataInfo	Data

- Here is an example with a « T » format frame:

Raw data:
0x02010606FF5707124D0A0B09425055434B5
3543830304131324E41

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0xFF	0x5707124D0A
11	0x09	0x425055434B53543830304131324E41

Name (ASCII)

0x5707 => **ELA CIN** : ELA Innovation *Company Identifier number*
- 0x07 : LSB
- 0x57 : MSB

Soit 0x**0757**.

0x12 : DataInfo : Indicate the following data : Temperature data here

Data : T° data:
- 0x4D : LSB
- 0x0A : MSB

ll sensor data are listed in the table below:

Field		Length	Description
Temperature data (T)	DataInfo	1 byte	0x12 (bit7-4=1 et bit3-0=2)
	Data	2 bytes	Temperature on 16 signed bits / 0,01°C step
Humidity data (RH)	DataInfo	1 byte	0x21 (bit7-4=2 and bit3-0=1): Relative Humidity
	Data	1 byte	Humidity on 8 unsigned bits / 1 % step RH de 0 à 100 %
Magnetic data (MAG)	DataInfo	1 byte	0x32 (bit7-4=3 and bit3-0=2)
	Data	2 bytes	Event (state change) counter on the 15 (unsigned) MSB Instantaneous state on LSB
Movement data (MOV)	DataInfo	1 byte	0x42 (bit7-4=4 and bit3-0=2)
	Data	2 bytes	Event counter (threshold overflow) on the 15 (unsigned) MSB Instantaneous state on LSB
Infrared movement data (PIR)	DataInfo	1 byte	0x92 (bit7-4=9 and bit3-0=2)
	Data	2 bytes	Event (infrared movement detected) counter on the 15 (unsigned) MSB Instantaneous state on LSB
	DataInfo	1 byte	0x56 (bit7-4=5 and bit3-0=6)

Accelerometer data (ANG)	Data	6 bytes	X-axis acceleration on 16 signed bits (range +/-2G) Y-axis acceleration on 16 signed bits (range +/-2G) Z-axis acceleration on 16 signed bits (range +/-2G)
Digital Input data (DI)	DataInfo	1 byte	0x62 (bit7-4=6 and bit3-0=2)
	Data	2 bytes	Event (input state change) counter on the 15 (unsigned) MSB Instantaneous input state on LSB
Analog Input data (AI)	DataInfo	1 byte	0x72 (bit7-4=7 and bit3-0=2)
	Data	2 bytes	Voltage measured in mV on 16 unsigned bits
Press Button data (TOUCH)	DataInfo	1 byte	0xB2 (bit7-4=4 and bit3-0=2)
	Data	2 bytes	Event counter (state change) on the 15 (unsigned) MSB Instantaneous state on LSB
Distance (PROXIR)	DataInfo	1 byte	0xA2 (bit7-4=4 and bit3-0=2)
	Data	2 bytes	Distance to the target in mm on the 15 (unsigned) MSB Measurement integrity on LSB

Identifiers format data (*Id* and *DO*) offer to transmit an identifier configured by NFC:

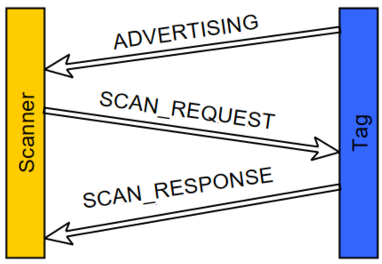
***Id* format data**	DataInfo	1 byte	0x06 (bit7-4=0 and bit3-0=6)
	Data	6 bytes	*MFR_ID* configured by NFC
Digital output data (*DO*)	DataInfo	1 byte	0x86 (bit7-4=8 and bit3-0=6)
	Data	6 bytes	*MFR_ID* configured by NFC

5.4. *SCAN RESPONSE FRAME (legacy)*

In some formats and versions, the tag can send a frame called « Scan Response frame ».

Once an advertising packet has been received by a scanner, further information can be requested. Then the tag responds with the "scan response" frame.

This frame is located right after the advertising frame and contains different data depending on the version and format.
The data sent in "Scan response" frame is also formatted either in Service mode or in Manufacturer Specific mode.



5.5. BATTERY INFORMATION (legacy)

Battery capacity

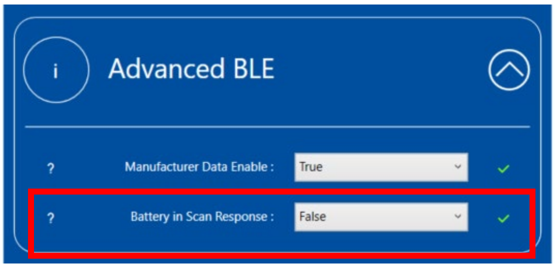
ELA Innovation's tags are based on the transmission of battery information in the Scan Response when the capacity of the battery falls below 15%. The formatting of the information is as follows:

Frame type		Service Data	Service Data	Mfr. Spec. Data
Version		1.0.0, 2.0.0, 2.1.x	≥2.2.0	≥2.0.0
Transmission		Batt. capacity < 15%	Batt. capacity < 15%	Batt. capacity < 15%
Frame bytes	1	Length : 0x04	Length: 0x04	Length : 0x05
	2	Type : 0x16	Type : 0x16	Type : 0xFF
	3	Battery Serv. LSB : 0x0F	Battery Serv. LSB : 0x19	ELA_CIN_LSB : 0x57
	4	Battery Serv. MSB : 0x18	Battery Serv. MSB : 0x2A	ELA_CIN_MSB: 0x07
	5	Batt. data (%)	Batt. data (%)	BATT_DATA_ID: 0xF1
	6	Not used	Not used	Batt. data (%)
	7	Not used	Not used	Not used

Battery voltage

From version 3.0.0 onwards, it is possible to transmit battery voltage information for all formats. For this purpose, the "**Battery voltage presence**" option must be configured in the NFC memory.

When the option is activated, the tag no longer transmits battery capacity information below 15%.



Once the option is enabled, the battery voltage information is transmitted in the "Scan Response" frame with the following formatting:

Frame type		All
Version		≥3.0.0
Transmission		Battery voltage presence = 1
Frame Bytes	1	Length : 0x06
	2	Type : 0xFF
	3	ELA_CIN_LSB : 0x57
	4	ELA_CIN_MSB: 0x07
	5	BATT_DATA_ID: 0xF2
	6	Batt. voltage (mV) LSB
	7	Batt. voltage (mV) MSB

Frame examples showing battery information:

Received frame: ELA ID, Service Data, v3.0.0 Battery voltage presence = 0		**Received frame: ELA T, MFR Spec. Data, v3.0.0 Battery voltage presence = 0**	
Name	*BE_BATTERY*	**Name**	*BE_BATTERY*
		Measured temp.	*27.12°C (0x0A98)*
Battery cap.	*13% (0x0D)*	**Battery cap**	*13% (0x0D)*

Received frame: ELA ID, Service Data, v3.0.0

Raw data:

0x0201060B0942455F424154544552590
416192A0D

Details:

LEN.	TYPE	VALUE
2	0x01	0x06
11	0x09	0x42455F42415454455259
4	0x16	0x192A0D

Received frame: ELA T, MFR Spec. Data, v3.0.0

Raw data:

0x02010606FF570712980A0B0942455F4
241545445525905FF5707F10D

Details:

LEN.	TYPE	VALUE
2	0x01	0x06
6	0xFF	0x570712980A
11	0x09	0x42455F42415454455259
5	0xFF	0x5707F10D

T° Data
Name
Battery cap. (SR Frame)

In Eddystone and iBeacon formats, the battery information is located before the Tag Name :

Received frame : iBeacon, v2.1.0		**Received frame: Eddystone, v3.0.0 Battery voltage presence = 0**	
Name	*BE_BATTERY*	**Name**	*BE_BATTERY*
Battery cap.	*13% (0x0D)*	**Battery cap.**	*13% (0x0D)*

Received frame : iBeacon, v2.1.0

Raw data:

0x0201061AFF4C000215010203040506 0
708090A0B0C0D0E0F10020B010AC4041
60F180D0B0942455F42415454455259

Details:

LEN.	TYPE	VALUE
2	0x01	0x06
26	0xFF	0x4C000215010203040506070809 0 A0B0C0D0E0F10020B010AC4
4	0x16	0x0F180D
11	0x09	0x42455F42415454455259

Received frame: Eddystone, v3.0.0

Raw data:

0x0201060303AAFE1716AAFE00ED01020
30405060708090A010203040A0B000004
16192A0D0B0942455F42415454455259

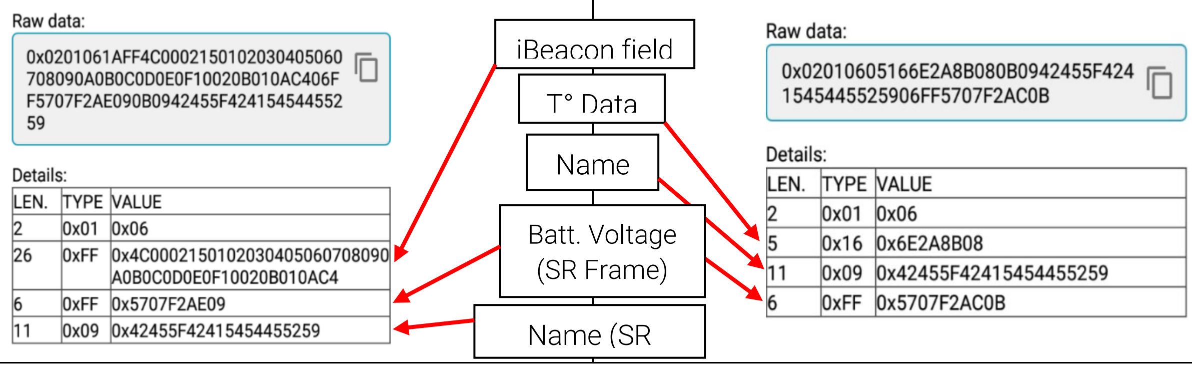
Details:

LEN.	TYPE	VALUE
2	0x01	0x06
3	0x03	0xAAFE
23	0x16	0xAAFE00ED0102030405060708 09 0A010203040A0B0000
4	0x16	0x192A0D
11	0x09	0x42455F42415454455259

iBeacon field
Eddystone field
Battery cap. (SR Frame)
Name (SR Frame)

Received frame: iBeacon, v3.0.0 ***Battery voltage presence = 1***	
Name	*BE_BATTERY*
Batt. voltage	*2.478V (0x09AE)*

Received frame: ELA T, Service Data, v3.0.0 ***Battery voltage presence = 1***	
Name	*BE_BATTERY*
Measured temp	*21.87°C (0x088B)*
Batt. voltage	*2.988 V (0x0BAC)*



5.6. INFORMATION ABOUT IBEACON, EDDYSTONE

Tags settings available in iBeacon format : iBeacon

- Compliance with Apple specific data such as:

  Flags – Length – Type - Company ID - Beacon Type - Proximity UUID - Major - Minor

- You can add an additional "Name", which is send it in the "Scan Response" BLE frame and configure it in the "Name" field from the Device Manager
- Specification: https://developer.apple.com/ibeacon/

Tags settings available in Eddystone UID format : Eddystone

- Compliance with specific Google Data Eddystone UID format such as:

  A unique, static ID with a 10-byte Namespace component and a 6-byte Instance component

- You may add an additional "Name", which is send in the "Scan Response" BLE frame and configure it in the "Name" field
- Specification: https://developers.google.com/beacons/overview

5.7. CUSTOM FRAME FORMAT

- **Specification Rules**

Firmware version $\geq 4.0.0$ offers the possibility to customize entirely the frame format of the advertising and scan response payloads. The custom frame format are defined during the tag production and will be remain identical all over the operation of the tag.

Custom frame format specification follows the following rules:

- Frames are split in blocks that in turns, define the value of several bytes. A maximum of 5 blocks can be defined for each of the Advertising and Scan Response frame.

- Blocks are ordered which means that the bytes defined by the first block will be on the most significant side, the bytes defined by the second block will follow and so on.

- Block can be conditional which means that a block can be included in the payload only if a condition on sensor data is verified. A maximum of 2 conditional blocks per frame can be defined. For each condition, the block can be defined when the condition is valid or invalid.

- If the sum of the number of bytes over all blocks must not exceed 28, extra bytes are discarded

- Bloc type must be defined among the list of table 1

- Data included in a block can be of following types (table 1)

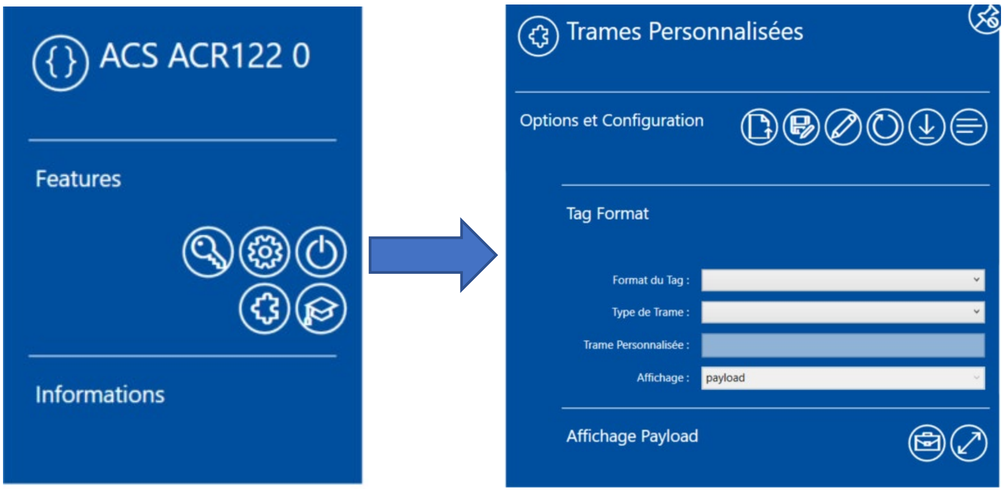
    - A static hexadecimal value (table 1)
    - Sensor data (table 1)

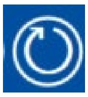
- Sensor value can be specifically rescaled and formatted (table 1)

Bloc type	Conditions	Data type			Scaling	Data format
		STATIC	Local naming	Sensor		
Incomplete List of UUID16	Always present	USER DATA	NAME	T	X1000	uint8
Complete List of UUID16	Strictly inferior		MFR_ID	RH	X100	int8
Incomplete List of UUID32	Strictly superior		Eddystone NID	MAG (cpt + état)	X10	uint16 LSB
Complete List of UUID32	Equal to		Eddystone BID	MOV (cpt + état)	X1	uint16 MSB
Incomplete List of UUID128	In between		iBeacon UUID	ACC_X	/10	int16 LSB
Complete List of UUID128	Outside		iBeacon MAJOR	ACC_Y	/100	int16 MSB
Shortened Local Name			iBeacon MINOR	ACC_Z	/1000	uint32 [0-7]
Complete Local Name				DI (cpt + état)		uint32 [8-15]
TX Power level				AI		uint32 [16-23]
Solicited list of UUID16				PIR (cpt + état)		uint32 [24-31]
Solicited list of UUID128				CHOC		int32 [0-7]
Service Data 16				Tension pile		int32 [8-15]
Service Data 32				Capacité pile (%)		int32 [16-23]
Service Data 128				Puissance TX à 0m		int32 [24-31]
Manufacturer Specific Data				Puissance TX à 1m		FP32 [0-7]
				Compteur MAG		FP32 [8-15]
				Etat MAG		FP32 [16-23]
				Compteur MOV		FP32 [24-31]
				Etat MOV		FP24 [0-7]
				Compteur DI		FP24 [8-15]
				Etat DI		FP24 [16-23]
				Compteur PIR		
				Etat PIR		
				Puissance TX brute		
				Touch (cpt + état)		
				Compteur TOUCH		
				Etat TOUCH		
				Distance (PROXIR)		

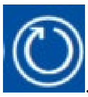
- **NFC Configuration**

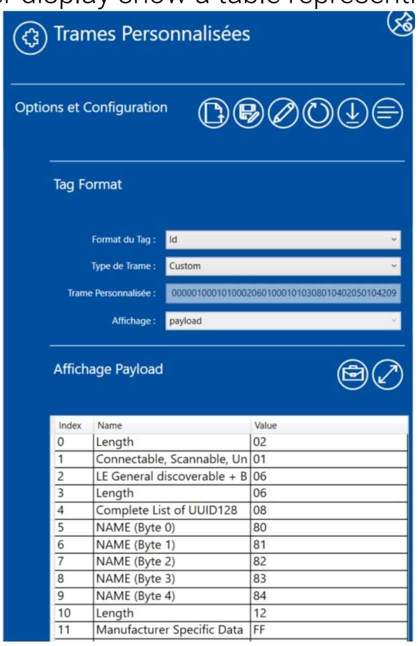
Custom frame can be configured using the "***Custom Frame"*** tool accessible from the icon 



1. Click on the  incon to read and refresh custom frame configuration

2. Select the frame type to ***"Custom***" and click the "***Write"*** button 

3. Remove the beacon from the NFC reader wait for 2 seconds, put it back on the NFC reader and click the refresh icon . The frame type should display ***"Custom"*** and the custom frame buffer display show a table representing the new frame format



The ***"Import"*** icon can be used to load pre-saved (in Device Manager Software) custom frame format.

Upon request a new frame format can be define by ELA Innovation and share using a specific .elacf file format. Use the icon ***"Custom Frame environment"*** to save this format with Device Manager Software and makes it available for import.

- **Examples**

TEMPERATURE SENSOR WITH ELA T SERVICE DATA FRAME TYPE PLUS BATTERY VOLTAGE

In this first example, the advertising frame is specified as follow:

Frame type	Byte Num	Block	Conditions
			Always
Advertising	1	Header	Longueur : 0x02
	2		Type : 0x01
	3		Donnée : 0x06
	4	Block 1	Longueur : 0x05
	5		Type : 0x16
	6		Carac. T° LSB : 0x6E
	7		Carac. T° MSB : 0x2A
	8		Donnée T° (0,01°C) LSB
	9		Donnée T° (0,01°C) MSB
	10	Block 2	Longueur : ≤ 0x0E
	11		Type : 0x09
	12		Nom[0]
	13		Nom[1]
	14		Nom[2]
	15		Nom[3]
	16		Nom[4]
	17		Nom[5]
	18		Nom[6]
	19		Nom[7]
	20		Nom[8]
	21		Nom[9]
	22		Nom[10]
	23		Nom[11]
	24		Nom[12]
	25		Nom[13]
	26	Block 3	Longueur: 0x04
	27		Type : 0x16
	28		Carac. Voltage LSB 0x18
	29		Carac. Voltage MSB 0x2B
	30		Batt. Voltage (mV) LSB
	31		Batt. Voltage (mV) MSB

This is the same frame format as the standard ELA T with service data format but with battery voltage present at the end of the advertising frame. Note, that the length of the name is reduce to 14 Bytes (-1 compared to ELA T standard format) and the "voltage" (generic) characteristic identifier (0x2B18) define in the Bluetooth SIG specification is used for the battery voltage as a specific identifier for battery voltage does not exists in the specs.

As a result the advertising payload will be the following (27°C):

Raw data:
0x0201060516 6E2AAB0A0F09454C4120494E4E4F564154494F4E

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x6E2AAB0A
14	0x09	0x454C4120494E4E4F564154494F4E
4	0x16	0x182BAE09

0x6E2A : *Temperature* service

T° data:

- 0xAB : LSB
- 0x0A : MSB

T° = 0AAB = 2731 * 0.01 = 27.31°C

Name (ASCII) : ELA INNOVATION

0x2B18 : *Voltage* service

Voltage data:

- 0xAE : LSB
- 0x09 : MSB

V = 09AE = 2478 mV

MAGNETIC SENSOR DATA WITH "MANUFACTURER SPECIFIC DATA" DATA TYPE (ELA SPECIFIC), NAME WITH "SERVICE DATA TYPE" AND CONDITIONAL NAME BLOCK

This example illustrates the use of conditional advertising according to the sensor value. Door opening monitoring can be achieved using beacon with magnetic sensor (MAG) stuck on a fixed surface and magnet fixed on the door such that in the closed position the magnet is brought very close to the beacon. In this scenario it possible to configure the beacon such that it advertises the ASCII code for "OPEN" when no magnet is detect and the ASCII code for "CLOSE" when a magnet is detected.

Frame type	Byte Num	Block	Conditions	
			MAG State = 0	MAG State = 1
Advertising	1	Header	Longueur : 0x02	Longueur : 0x02
	2		Type : 0x01	Type : 0x01
	3		Donnée : 0x06	Donnée : 0x06
	4	Block 1	Longueur : 0x05	Longueur : 0x05
	5		Type : 0x16	Type : 0x16
	6		Carac. T° LSB : 0x06	Carac. T° LSB : 0x06
	7		Carac. T° MSB : 0x2A	Carac. T° MSB : 0x2A
	8		MAG Data (cn + state) LSB	MAG Data (cn + state) LSB
	9		MAG Data (cn + state) MSB	MAG Data (cn + state) MSB
	10	Block 2	Longueur : ≤ 0x05	Longueur : ≤ 0x06
	11		Type : 0x09	Type : 0x09
	12		Data[0] : O	Data[0] : C
	13		Data[1] : P	Data[1] : L
	14		Data[2] : E	Data[2] : O
	15		Data[3] : N	Data[3] : S
	16		Not Used	Data[4] : E
	17		Not Used	Not Used
	18		Not Used	Not Used
	19		Not Used	Not Used
	20		Not Used	Not Used
	21		Not Used	Not Used
	22		Not Used	Not Used
	23		Not Used	Not Used
	24		Not Used	Not Used
	25		Not Used	Not Used
	26		Not Used	Not Used
	27		Not Used	Not Used
	28		Not Used	Not Used
	29		Not Used	Not Used
	30		Not Used	Not Used
	31		Not Used	Not Used

As a result the advertising payload will be the following:

Case 1: The magnetic sensor detect a magnetic field

Raw data:

0x02010606FF570732FB0A0609434c4f5345

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0xFF	0x570732FB0A
6	0x09	0x434c4f5345

CLOSE (ASCII)

0x5707 : *ELA Innovation* Compagny Identifier
0x32 : *Magnetic sensor data* (ELA Mfr Specific data type
MAG data:

- 0xFB : LSB
- 0x0A : MSB

Hexa.	0	A	F	B
Binary	0000	1010	1111	1011

⇨ 1: instantaneous sensor state (magnet present)
⇨ 1010 1111 101: event counter value on 15 bits, 1405 in this example

Case 2: The magnetic sensor detects a magnetic field

Raw data:

0x02010606FF570732FA0A05094f50454e

Details :

LEN.	TYPE	VALUE
2	0x01	0x06
5	0xFF	0x570732FA0A
5	0x09	0x4f50454e

OPEN (ASCII)

0x5707 : *ELA Innovation* Compagny Identifier
0x32 : *Magnetic sensor data* (ELA Mfr Specific data type
MAG data:

- 0xFA : LSB
- 0x0A : MSB

Hexa.	0	A	F	A
Binary	0000	1010	1111	1010

⇨ 0: instantaneous sensor state (magnet present)
⇨ 1010 1111 101: event counter value on 15 bits, 1405 in this example

EMULATING PROPRIETARY FRAME FORMAT

It is not uncommon for some solution to use proprietary frame format and it is, of course, impossible to embed in the beacon memory all existing proprietary format. This example illustrates the capability of the custom frame feature to implement a proprietary frame format. ELA tag provides as a pre-set the Eddystone frame format. However, this format does not provide any sensor or battery information as per its specification. There are two solutions to overcome this issue:

- Includes the sensor info in the scan response frame
- Use the Eddystone TLM format according to its specification

Eddystone format with Battery voltage and temperature data in the scan response

Frame type	Byte Num	Block Num	Condition Always	Frame type	Byte Num	Block Num	
Advertising	1	HEADER	Length : 0x02	Scan Response	1	Block 1	Length : 0x06
	2		Type : 0x01		2		Type : 0xFF
	3		Data : 0x06		3		ELA_CIN_LSB : 0x57
	4	BLOCK 1	Length : 0x03		4		ELA_CIN_MSB : 0x07
	5		Type : 0x03		5		BATT_DATA_ID : 0xF2
	6		Eddystone_UUID_LSB : 0xAA		6		Bat Voltage (mV) LSB
	7		Eddystone_UUID_MSB : 0xFE		7		Bat Voltage (mV) MSB
	8	BLOCK 2	Length : 0x17		8	Block 2	Length : 0x05
	9		Type : 0x16		9		Type : 0x16
	10		Eddystone_UUID_LSB : 0xAA		10		Carac. T° LSB : 0x6E
	11		Eddystone_UUID_MSB : 0xFE		11		Carac. T° MSB : 0x2A
	12		Frame type UUID : 0x00		12		Donnée T° (0,01°C) LSB
	13		Power TX à 0m		13		Donnée T° (0,01°C) MSB
	14		NID[0]		14		Not Used
	15		NID[1]		15		Not Used
	16		NID[2]		16		Not Used
	17		NID[3]		17		Not Used
	18		NID[4]		18		Not Used
	19		NID[5]		19		Not Used
	20		NID[6]		20		Not Used
	21		NID[7]		21		Not Used
	22		NID[8]		22		Not Used
	23		NID[9]		23		Not Used
	24		BID[0]		24		Not Used
	25		BID[1]		25		Not Used
	26		BID[2]		26		Not Used
	27		BID[3]		27		Not Used
	28		BID[4]		28		Not Used
	29		BID[5]		29		Not Used
	30		Reserved		30		Not Used
	31		Reserved		31		Not Used

Eddystone TLM format

Type trame	Byte Num	Block Num	Condition	
			Frame count -> Even	Frame count -> Odd
Octets trame	1	HEADER	Length : 0x02	Length : 0x02
	2		Type : 0x01	Type : 0x01
	3		Data : 0x06	Data : 0x06
	4	Block 1	Length : 0x03	Length : 0x03
	5		Type : 0x03	Type : 0x03
	6		Eddystone_UUID_LSB : 0xAA	Eddystone_UUID_LSB : 0xAA
	7		Eddystone_UUID_MSB : 0xFE	Eddystone_UUID_MSB : 0xFE
	8	Block 2	Length : 0x17	Length : 0x17
	9		Type : 0x16	Type : 0x16
	10		Eddystone_UUID_LSB : 0xAA	Eddystone_UUID_LSB : 0xAA
	11		Eddystone_UUID_MSB : 0xFE	Eddystone_UUID_MSB : 0xFE
	12		Frame type UUID : 0x00	Frame type UUID : 0x20
	13		Power TX à 0m	DATA[0] : 0x00
	14		NID[0]	Batt. voltage (mV) MSB
	15		NID[1]	Batt. voltage (mV) LSB
	16		NID[2]	T° Data (°C) MSB
	17		NID[3]	T° Data (°C) LSB
	18		NID[4]	Frame Count (unint32 |24-31])
	19		NID[5]	Frame Count (unint32 |16-23])
	20		NID[6]	Frame Count (unint32 |8-15])
	21		NID[7]	Frame Count (unint32 |0-7])
	22		NID[8]	Second Count (unint32 |24-31])
	23		NID[9]	Second Count (unint32 |16-23])
	24		BID[0]	Second Count (unint32 |8-15])
	25		BID[1]	Second Count (unint32 |0-7])
	26		BID[2]	Not used
	27		BID[3]	Not used
	28		BID[4]	Not used
	29		BID[5]	Not used
	30		Reserved	Not used
	31		Reserved	Not used

5.8. DATA VIZUALISATION USING DEVICE MANAGER

The ELA Innovation *Device Manager* application can perform BLE scans in order to view advertising data from BLE ELA Innovation products:

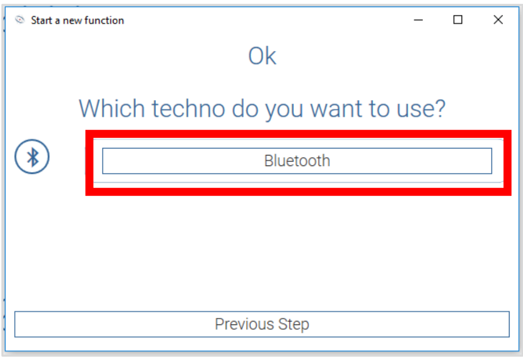
1. Enable internal Bluetooth or connect a Bluetooth device (typ. Dongle) to your PC

2. Launch the "**Device Manager**" desktop application

3. Start the **"Bluetooth" widget** by clicking button

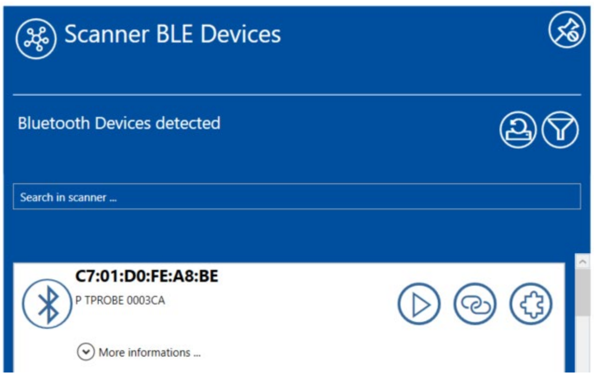
4. Start the **BLE device** **search**

5. Click on the found device. The ***Bluetooth*** windows appears

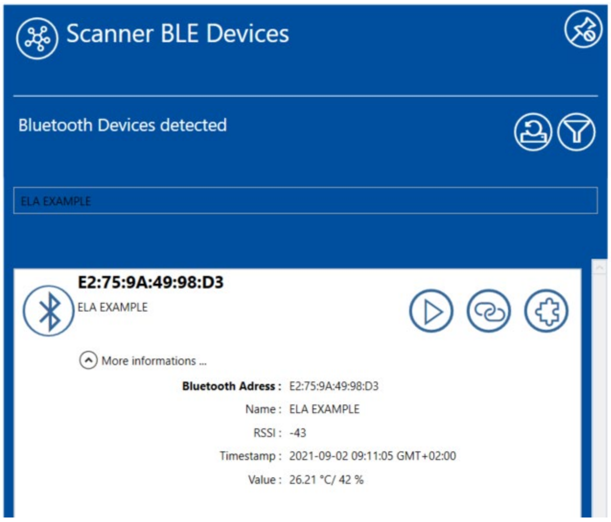


Once reader has been chosen, this window appears

6. Start the ***BLE Device scanner***



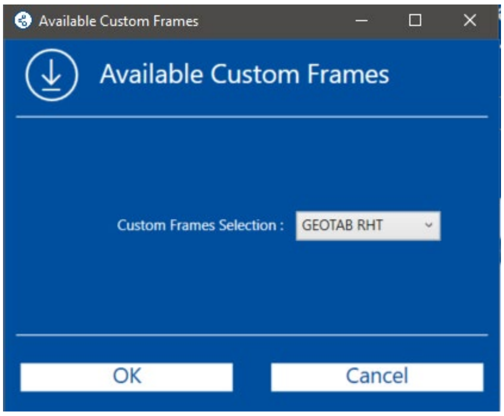
On this window, it is possible to search for a Name or MAC Address from the entire drop-down list (see next page)



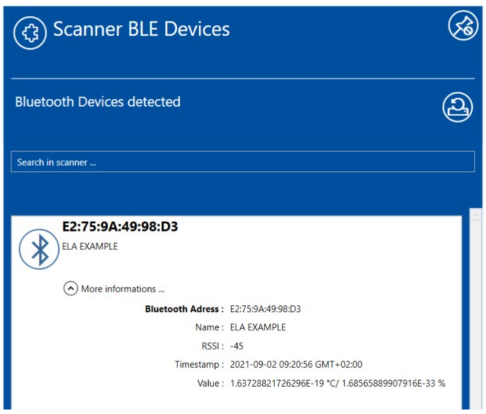
Tag name filtered
« More information » field extended
Right click: copy device name or mac address

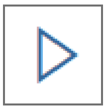
For tag with frame format other than ELA, IBeacon and Eddystone, decoded payload information can be maid accessible by clicking the  icon.

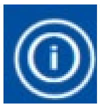
Select the predefine frame format:

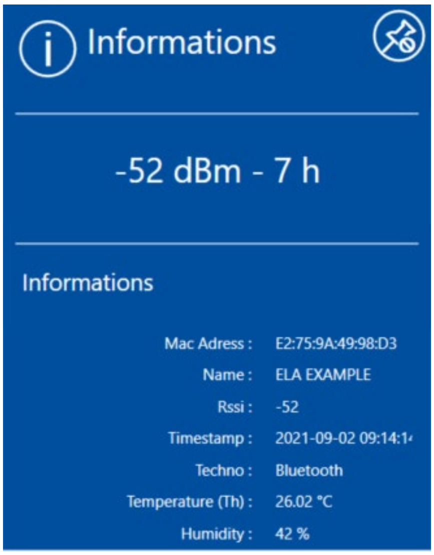


The corresponding tag info are now decoded according to this frame format:



7. Click on the ***Tag info visualization***  button. This window opens:

8. It is possible to view the tag data with the Information button.  This opens a window displaying tag Name, RSSI and MAC address, and sensor info:



6. CONNECTED MODE OPERATION

In "***Connected Mode***", a link is established between two devices and only they can communicate and exchange with each other. You may establish a connection using a smartphone or a mobile application, or with a PC equipped with the ELA "*Device Manager*" application (provided you activated Bluetooth or connected a BLE dongle to the PC).

- The *Advertising Recurrence* must be less than or equal to 3 seconds to be able to establish a connection.
- Once you enter "*Connected Mode*", "***Advertising*" is stopped by default**.
- It is possible to send commands to the tag to perform special actions or read data.

It is possible to get a record of saved data (***Datalogger***) using *Connected Mode*. This datalogger will contain sensor data saved at a defined period with a timestamp for each data

6.1. *CONNECTED MODE LIST OF COMMAND*

COMMANDS	***ACTIONS***	***MINIMUM FIRMWARE VERSION***
LED_ON	Turn ON the LED (infinite Blink)	**≥1.0.0**
LED_OFF	Turn OFF the LED	**≥1.0.0**
LED_ON XX	Turn ON the LED (for XX seconds)	**≥2.0.0**
BUZZ_ON	Turn ON the buzzer (Repeated beep)	**≥1.0.0**
BUZZ_OFF	Turn OFF the buzzer	**≥1.0.0**
BUZZ_ON XX	Turn ON the buzzer (for XX seconds)	**≥2.0.0**
DIGI_ON	Turn Digital Output to "ON" state	**≥2.1.0**
DIGI_OFF	Turn Digital Output to "OFF" state	**≥2.1.0**
DIGI_ON XX	Turn Digital Output to "ON" state for XX seconds	**≥2.1.0**
RAZ_COUNT	Counter reset	**≥2.0.0**
LOG_DL	Download datalogger values in chronological order	**≥2.0.0 (non-EN12830)**
LOG_SO_DL	Download values in reverse chronological order	**≥4.0.0 (non-EN12830)**
LOG_RST	Erase datalogger values and timestamp	**≥2.0.0 (non-EN12830)**
LOG_SP_DL XX YY	Download datalogger values from the index XX to index YY in chronological order (index 00 is the oldest value)	**≥4.0.0 (non-EN12830)**
LOG_SP_INV_DL XX YY	Download datalogger values from the index XX to index YY in reverse chronological order (index 00 is the newest value)	**≥4.0.0 (non-EN12830)**
LOG_SP_DL_DATE DD/MM/YYY HH :MM Dd/mm/yyyy hh :mm	Download datalogger values from the date DD/MM/YYYY HH:MM to the date dd/mm/yyyy hh:mm in chronological order	**≥4.1.0 (non-EN12830)**

LOG_SP_INV_DL_DATE DD/MM/YYY HH:MM Dd/mm/yyyy hh :mm	Download datalogger values from the date DD/MM/YYYY HH:MM to the date dd/mm/yyyy hh:mm in reverse chronological order	**≥4.1.0 (non-EN12830)**
LOG_SET_CURR_DATE DD/MM/YYYY HH:MM + TZ:TZ	Set the current date reference of the beacon to DD/MM/YYYY HH:MM TZ:TZ	**≥4.1.0 (non-EN12830)**
LOG_READ_CURR_DATE	Read current date according to the beacon clock calendar	**≥4.1.0 (non-EN12830)**
LOG_READ_START_DATE	Read the data logger start date according to the beacon clock callendar	**≥4.1.0 (non-EN12830)**
GET_BATT_VOLTAGE	Return battery voltage in mV	**≥2.1.0**
GET_SENSOR_DATA	Return the last measured sensor value	**≥2.2.0**

6.2. *SIMPLE DATA LOGGER (relative time)*

Ela innovation sensor tags can operate the "*Datalogger*" feature. The datalogger is a record of saved data, memorized while advertising, with each value associated with a time stamp, to be able to recover the moment when it was measured.

For firmware version strictly before V4.0.0, the datalogger can be retrieved in *Connected mode* using the "***LOG_DL***" command. The simple datalogger (non-EN12830) formatting for a temperature sensor with a log interval of 30 seconds is the following:

Temperature LOG:
DATA_START
0d0h0m30s:2712 ← Temperature measured 30 seconds after the tag has booted (here 27.12°C)
0d0h1m0s:2730
0d0h1m30s:2695
…
1d3h25m30s :1505 ← Temperature measured 1 day, 3 hours, 25 minutes and 30 seconds after the tag has booted (here 15.05°C)
END_OF_DATA

The simple datalogger formatting is the same for all ELA Innovation sensor products (xx**d**xx**h**xx**m**xx**s** followed by sensor data). The sensor data is the same as the one transmitted in advertising frames.

The "***LOG_RST***" command is used to delete datalogger data content.

For firmware version equal or above V4.0.0 (non EN 12830), it is possible to download only a subset of logged value in order to reduce the number of data downloaded and thus downloading time.

The connected command "**LOG_SP_DL XX YY**" is used to download logged value between index XX and index YY in chronological order, index 00 being the oldest value logged.

The connected command "**LOG_SP_INV_DL XX YY**" is used to download logged value between index XX and index YY in reverse chronological order, index 00 being the newest value logged.

For a simple datalogger (non-EN12830) of a temperature sensor with a log interval of 30 seconds, the results of the three downloading command are exemplified below:

LOG_DL	LOG_SP_DL 03 12	LOG_SP_INV_DL 03 12
Temperature LOG: **DATA_START** **0d0h0m30s:2712** **0d0h1m0s:2730** **0d0h1m30s:2695** **…** **1d3h24m30s:1617** **1d3h25m0s:1500** **1d3h25m30s :1505** **END_OF_DATA**	**Temperature LOG:** **DATA_START** **0d0h1m30s:2695** **0d0h2m0s:2700** **0d0h2m30s:2705** **…** **0d0h5m0s:2902** **0d0h5m30s:2875** **0d0h6m0s :2822** **END_OF_DATA**	**Temperature LOG:** **DATA_START** **1d3h25m30s:1505** **1d3h25m0s:1500** **1d3h24m30s:1617** **…** **1d3h20m30s:1200** **1d3h20m00s:1102** **1d3h19m30s :1015** **END_OF_DATA**

6.3. ABSOLUTE TIME DATA LOGGER

For firmware version strictly before V4.1.0, time stamp of the datalogger can be retrieved in absolute time with the formatting DD/MM/YYYY HH:MM:SS Tz:Tz using the command "***LOG_SP_DL_DATE".*** This requires to fix set a date reference in the beacon with the "**LOG_SET_CURR_DATE"** commands any time between the start of the data log (last "***LOG_RST"*** command or reboot of the beacon) and the ***LOG_SP_DL_DATE*** command.

Absolute timing data logger formatting for a temperature sensor with a log interval of 30 seconds is the following:

Temperature LOG:
DATA_START
0d0h0m30s:2712 ← Temperature measured 30 seconds after the tag has booted (here 27.12°C)
0d0h1m0s:2730
0d0h1m30s:2695
…
1d3h25m30s :1505 ← Temperature measured 1 day, 3 hours, 25 minutes and 30 seconds after the tag has booted (here 15.05°C)
END_OF_DATA

Like for the simple data logger it is possible to download only a part of the datalogger by specifying the start and stop date of interest using the **LOG_SP_DL_DATE DD/MM/YYY HH :MM dd/mm/yyyy hh :mm.** command. Likewise, it is possible to download the data logger entry in reverse chronological order (newest first) using the **LOG_SP_INV_DL_DATE** command for which a start and stop date can also be defined (partial download).

For an absolute time datalogger (non-EN12830) of a temperature sensor with a log interval of 30 seconds, the results of the three downloading commands are shown below:

LOG_DL	LOG_SP_DL 03 12	LOG_SP_INV_DL 03 12
Temperature LOG: **DATA_START** **0d0h0m30s:2712** **0d0h1m0s:2730** **0d0h1m30s:2695** **…** **1d3h24m30s:1617** **1d3h25m0s:1500** **1d3h25m30s :1505** **END_OF_DATA**	**Temperature LOG:** **DATA_START** **0d0h1m30s:2695** **0d0h2m0s:2700** **0d0h2m30s:2705** **…** **0d0h5m0s:2902** **0d0h5m30s:2875** **0d0h6m0s :2822** **END_OF_DATA**	**Temperature LOG:** **DATA_START** **1d3h25m30s:1505** **1d3h25m0s:1500** **1d3h24m30s:1617** **…** **1d3h20m30s:1200** **1d3h20m00s:1102** **1d3h19m30s :1015** **END_OF_DATA**

6.4. EN12830 DATA LOGGER (BLUE PUCK T EN12830 & BLUE PUCK TPROBE)

The EN12830 format has several new features:

- EN12830 Data logger
- Calibration by 2nd-degree polynomials of temperature values
- Saving tag calibration values (Target values – measured values)

These EN12830 (2018) dedicated functionalities are protected by a BLE password. This password is inserted by the NFC configuration. The EN12830 tag configuration options are only available from *Device manager* version 1.3.0.

The PUCK T EN12830 dedicated documentation can found here [BLUE PUCK T ZN12830 Application Note]

6.5. CONNECTED MODE RESTRICTIONS

- During a NFC configuration, datalogger data is erased from the tag memory.
- If the tag is in *Connected Mode* and goes under an ***NFC-field***, then the tag will restart.

6.6. CONNECTING TO A ELA INNOVATION BLE TAG

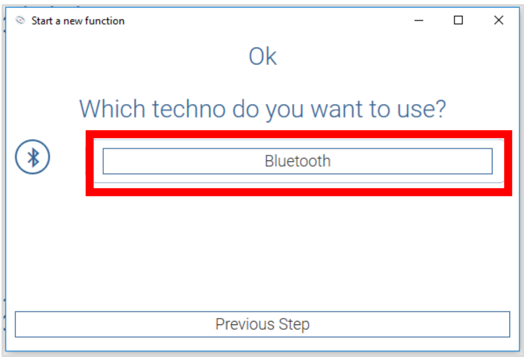
Enable internal Bluetooth or connect a Bluetooth device (typ. Dongle) to your PC

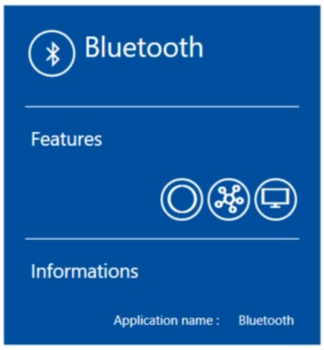
1. Launch the "**Device Manager**" desktop application 

2. Start the **"Bluetooth" widget** by clicking  button

3. Start the **BLE device search** 

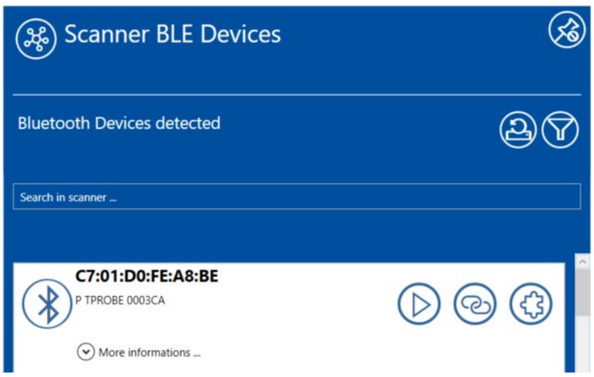
4. Click on the found device. The ***Bluetooth*** windows appears





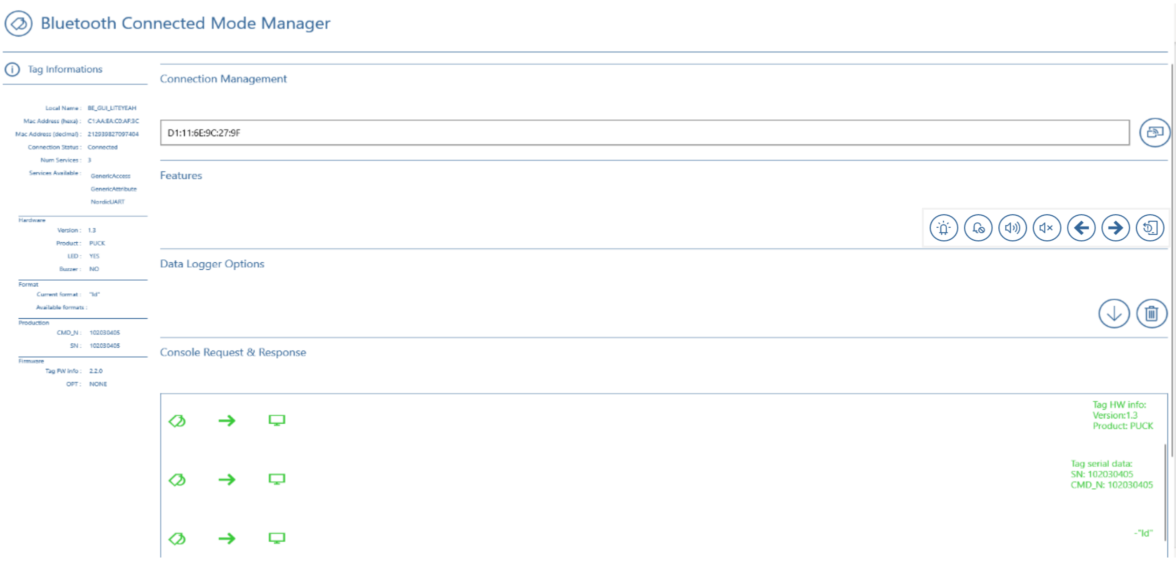
Once reader has been chosen, this window appears

5. Start the BLE Device scanner 



On this window, it is possible to search for a Name or MAC Address from the entire drop-down list (see next page)

6. Start the connection by  pressing icon. The *Device Manager Connector* window opens:



- **Features:** Commands to send to the tag (see next page for commands syntax and use)
- **Informations :** Name –Mac Adress – Connection status – Available services
- **Hardware – Format - ...:** Services details and tag options

« COMMANDS » DESCRIPTION

ICONS	*COMMANDS*	*ACTIONS*
	LED_ON	Turn ON the LED (infinite Blink)
	LED_OFF	Turn OFF the LED
	BUZZ_ON	Turn ON the buzzer (repeated beep)
	BUZZ_OFF	Turn OFF the buzzer
	DIGI_ON	Turn Digital Output to "ON" state
	DIGI_OFF	Turn Digital Output to "OFF" state
	RAZ_COUNT	Counter reset (for MAG, MOV and DI formats)
	LOG_DL	Download datalogger values
	LOG_RST	Erase datalogger values and timestamp

- **LED & BUZZER commands:**

For lifetime constraints, LED and BUZZER commands cannot be turned ON at the same time.

- **Datalogger download :**

The « **LOG_DL** » command is used to download the recorded log data.
Detailed Data according to sensor can be found on the application note on the ELA website.

7. PRODUCT SPECIFIC OPERATION

7.1. *OVER THE AIR PROGRAMMING (OTAP) SOFTWARE UPDATE*

OTAP (Over-The-Air Programming) is a method used to update a software, data or settings of a product without having to disassemble it and do it in a completely wireless way.

ELA Innovation products programmed with firmware version >3.0.0 can use OTAP Mechanism to update the tag embedded firmware, which can be done without having to return the product to ELA Innovation.

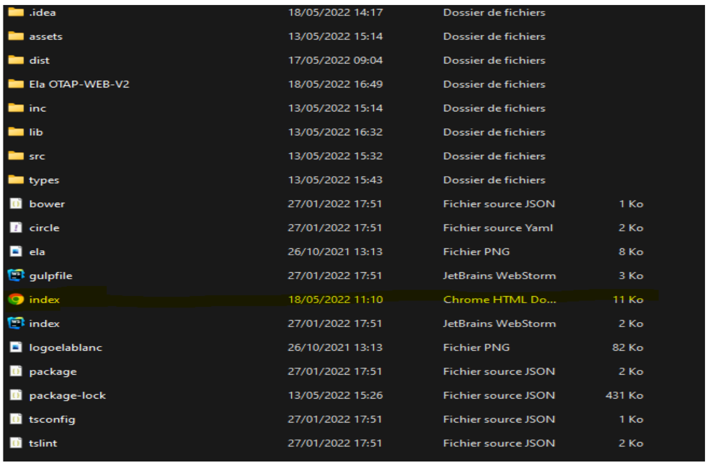
The OTAP procedure is secured by 2 methods:

- The switch into OTAP mode of ELA Innovation products is protected by a password that can be set by the user with NFC configuration
- The firmware update package is signed by a SHA256 private key.

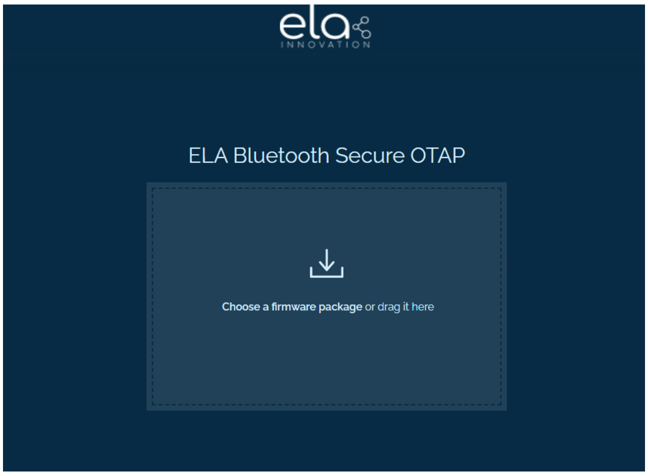
STEP BY STEP PROCEDURES

- Contact ELA support to receive the FIRMWARE OTAP package for the desired FW version
- Download the ZIP file of the OTAP tools from ELA web site
- Unzip the file with winrar,7Zip… and open the newly created folder "Ela-Otap-WEB VX"

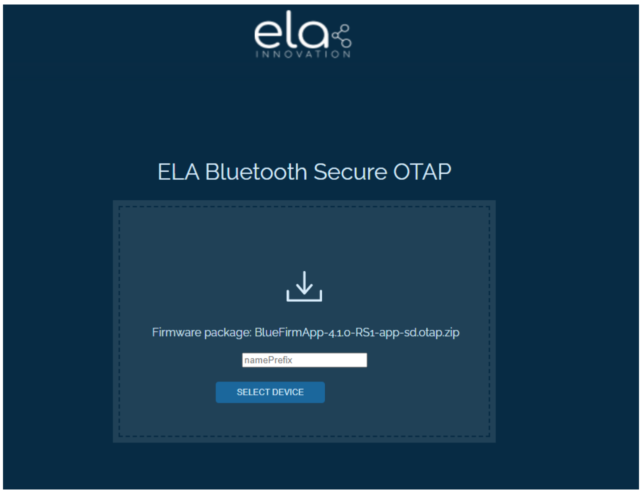
- Open the index.html with Google Chrome or Microsoft Edge



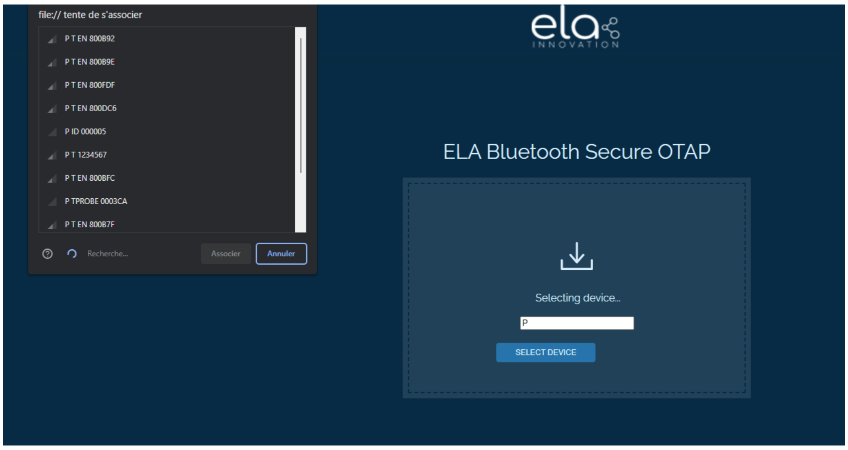
- The OTAP tool display the page below



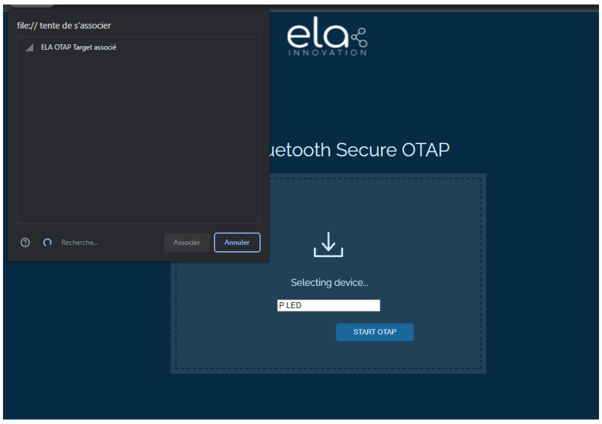
- Choose the FIRWMARE OTAP package file or drag it on the app web

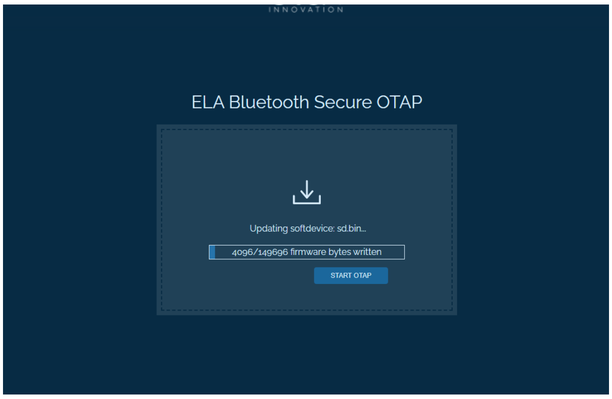


- Enter beginning (prefix) of the tag name (Case sensitive), click on "select device"
- A small pop up opens on the side with the different scanned devices, choose your device



- Wait for the message "Otap Mode Enable you can select "Ela OTAP Target"
- Click on “Start OTAP”, a small pop up opens on the side choose ELA OTAP TARGET





- Wait for the OTAP procedure to completed and the tag should reboot (blinking red LED)

7.2. BLUE PUCK T EN12830 & BLUE PUCK TPROBE

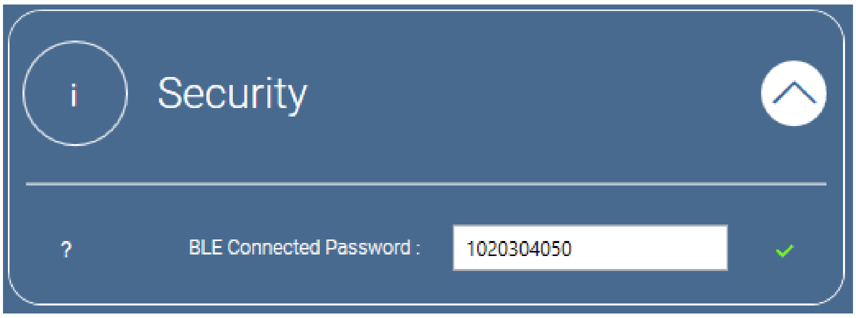
7.2.1. Password configuration

Designed for the EN12830 standard (temperature recorders for the transport, storage, and distribution of temperature sensitive goods), these features are protected by a Bluetooth Low Energy (BLE) access password. The password is added during tag configuration using NFC.

BLE connected password configuration

The BLE connected password must be 10 characters long. It may contain special characters (*Space*, !,.”, #, $, %, &, ‘, (,), *, +, -, /, :, ;, <, =, >, ?, @) Here is an example of BLE password configuration in an EN12830 tag using the Device Manager PC application.

1. Open Device Manager and go into “Programmers” to select the reader “*ACS ACR122*”.
2. Open the Configuration panel. 
3. Place the tag on the NFC reader and click on “Refresh”. 
4. Click on the “Security options” icon. 
5. The “Security” window is displayed:



6. Enter the BLE password in this window and then click on “Write”. 
7. When the write pop-up window is displayed, you may remove the tag from the NFC reader. The BLE password is now operational.

ATTENTION:

- A password cannot be read from the device.
- You can replace a password with a new password.
- Temperature data stored on the tag is erased if the password is changed.

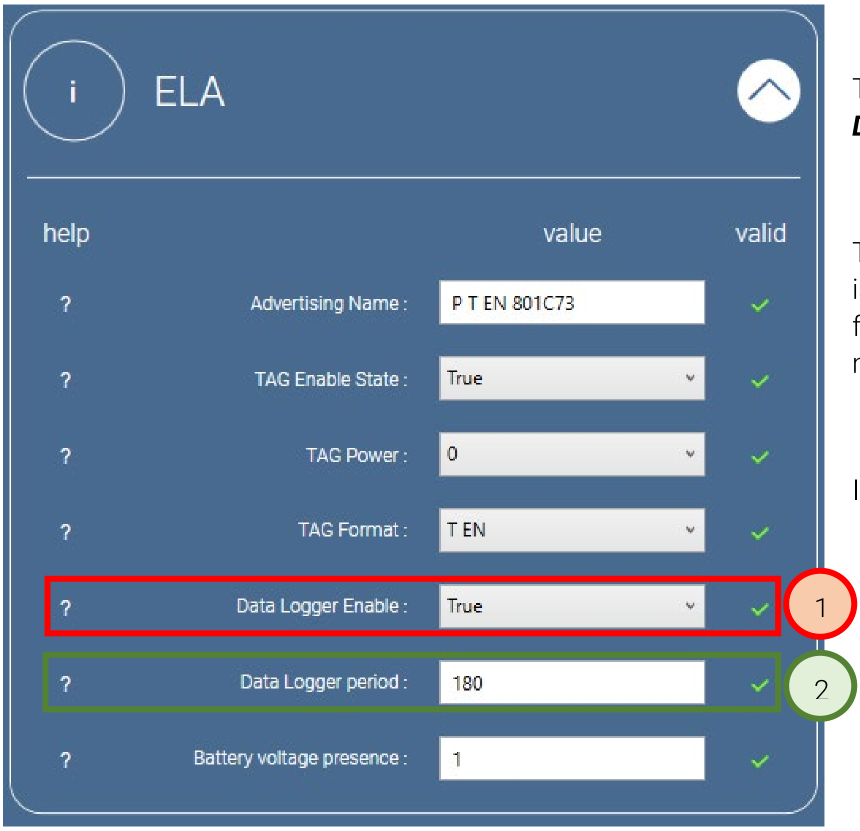
7.2.2. EN12830 data logger

7.2.2.1. General information

Firmware version > 2.1.0 tags implements a data logger that is fully compatible with the *EN12830 Data Recorder v2018 standard*. The data logger contains timestamp and identification information for stored values and implements a data control mechanism to ensure integrity.

7.2.2.2. NFC configuration

Datalogger fields are configured using the Device Manager application:



To use the data logger, you must set the ***Data Logger Enable*** field to ***True***. **(1)**

The storage interval for temperature values is configured using the ***Data Logger period*** field. The value expressed in this field is in mili***seconds***. **(2)**

In this example, the data logger is ***activated*** with an interval of ***10 seconds*** between readings.

NOTE: To be able to connect to the tag and download the datalogger, it is recommended to configure an ***Advertising Interval less than or equal to 3 seconds.*** If the advertising period is greater than this value, establishing *Connected Mode* may take more time and several attempt.

7.2.2.3. Configuration of BLE connected mode

Starting datalogging

To start storing temperature readings in tag memory, you must send the start date/time to the tag. That start date/time will be used as the basis for time-stamping data. After the command is sent, the data logger will start monitoring and storing values for the period defined in NFC configuration.

The command to start the data logger and send the date is as follows: ***DATALOGGER_START***. You must then provide the BLE password for the command to be considered by the tag. Lastly, you must provide the date in the following format: ***DD***/***MM***/***YYYY HH***:***mm***:***SS*** +***hh***:***gg***, where:

- ***DD***: day on which logging starts, written with 2 digits (ex. 01, 08, 15...)
- ***MM***: month in which logging starts, written with 2 digits (ex. 01, 05, 11...)
- ***YYYY***: year in which logging starts, written with 4 digits (ex. 2019...)
- ***HH***: hour at which logging starts, in 24-hour format, written with 2 digits (ex. 02, 16, 23...)
- ***mm***: minute at which logging starts, written with 2 digits (ex. 01, 26, 54...)
- ***SS***: seconds at which logging starts, written with 2 digits (ex. 05, 18, 56...)
- ***hh***: UTC time zone hour used to start the logging, written with 2 digits (ex. 00, 03…).
- ***gg***: UTC time zone minutes used to start the logging, written with 2 digits (ex. 00, 30...)

If the command syntax is valid, the tag returns ***DATALOGGER_START: Success.*** The date is then stored in the tag.

Command example:

DATALOGGER_START **PASSWORD_1** *05/06/2019 11:20:00 +01:00*

Tag response if password is valid: *DATALOGGER_START: Success*

Tag response if password is invalid: *DATALOGGER_START: ACCESS DENIED*

Tag response if date is invalid: *DATALOGGER_START: WRONG PARAMETERS*

When this command is sent to the tag, the first sensor measure will be performed after the Logging period entered in the NFC configuration.

For example, if the Datalogger period is 30 seconds, the first data logger data will be measured and stored 30 seconds after sending the *DATALOGGER_START* command.

IMPORTANT: When this command is sent to the tag, all previous content stored on the data logger is deleted and datalogging restarts from zero.

Stopping datalogging

You may stop datalogging on the tag. This is done using the ***DATALOGGER_STOP*** command. You must provide the BLE password the command to be considered by the tag. The data already logged is not erased by this command.

Command example:

DATALOGGER_STOP ***PASSWORD_1***

Tag response:

- If the password is valid and datalogging is running: *DATALOGGER_STOP: Success*
- If the password is valid but datalogging is not running: *DATALOGGER_STOP: LOG not started!*
- If the password is invalid: *DATALOGGER_STOP: ACCESS DENIED*

The data stored in data logger memory may be retrieved (***READ_DATA*** command) until you start another datalogging session or restart the tag.

7.2.2.4. Retrieving and verifying data

Reading all data logger values

IMPORTANT: You may read stored data at any time, without having to stop datalogging.

4 connected mode commands are available for retrieving datal logger datas:

- **READ_DATA**: download the complete data Logger in chronological order (oldest first)
- **READ_INV_DATA:** download the complete data logger in reverse order (newest first)
- **READ_SP_DATA DD/MM/YYYY HH:MM dd/mm/yyyy hh:mm**: download the complete data Logger in chronological order between the dates DD/MM/YYYY HH:MM and dd/mm/yyyy hh:mm
- **READ_SP_INV_DATA DD/MM/YYYY HH:MM dd/mm/yyyy hh:mm**: download the complete data Logger in reverse chronological order between the dates DD/MM/YYYY HH:MM and dd/mm/yyyy hh:mm

For each commands the password should be specified. If the command syntax is valid, the tag returns ***READ_DATA: Success.*** The tag then begins to transmit its data.

Command example:

READ_DATA ***PASSWORD_1***

READ_INV_DATA ***PASSWORD_1***

READ_SP_DATA ***DD/MM/YYYY HH:MM dd/mm/yyyy hh:mm PASSWORD_1***

READ_SP_INV_DATA ***DD/MM/YYYY HH:MM dd/mm/yyyy hh:mm PASSWORD_1***

Tag response:

- If password is valid: *READ_DATA: Success*
- If password is invalid: *READ_DATA: ACCESS DENIED*
- If the password is valid but datalogging is not running: *READ_DATA: LOG not started!*

Once the command is received, the tag transmits all the data stored in data logger memory. Data is in the following format:

---DOWNLOAD_START---
Firmware version: 2.1.0*n*
MacAddress: 01:02:03:04:05:FE*n*
Name: TAG_LOCAL_NAME*n*
Unit: Celsius degrees*n*
Start date: 01/04/2019 11:26:33 +01:00*n*
<DATA_START>*n*
01/04/2019 11:26:33+01:00: 23.34\n
01/04/2019 11:26:43+01:00: 23.44\n
[...]
<DATA_END>*n*
CRC16: 0x0D06*n*
---DOWNLOAD_END---

The newline characters (*n*) are not visible, but they must be considered for CRC calculation.

Field explanation:

- --- *DOWNLOAD_START*--- : marker indicating the start of transmission by data logger
- *Firmware version*: data logger (tag) firmware version
- *MacAddress*: tag's unique ID number
- *Name*: tag's name as configured in NFC memory
- *Unit*: unit used for temperature values (°C in this example)
- *Start date*: datalogging start as sent by the ***DATALOGGER_START*** command
- *<DATA_START>*: marker indicating the start of temperature data transmission
- 01/04/2019 11:26:33+01:00: 23.34: Complete example of temperature data with timestamp
- *<DATA_END>*: marker indicating the end of temperature data transmission
- *CRC16:* circular redundancy check calculated from transmitted data
- --- *DOWNLOAD_END*--- : marker indicating the end of transmission by data logger

Reading temperature data

In the previous example, temperature readings are formalized as follows:

01/04/2019 11:26:33+01:00: 23.34

- ***01/04/2019*** corresponds to the date on which the temperature was read: 1 April 2019 in this example.
- ***11:26:33*** corresponds to the time at which the temperature was read.
- ***+01:00*** corresponds to the UTC time zone, provided with the DATALOGGER_START command.
- ***23.34*** corresponds to the temperature value expressed in the unit transmitted by the data logger, 23.34°C in this example.

Checking data logger CRC

Datalogger data is controlled by a CRC16 calculation (*Cyclic Redundancy Check).* CRC details are as follows:

- CRC-16-CCITT algorithm
- 0x1021 polynomial
- Initialization value: 0xFFFF

Calculation example: Input data (ASCII type input) 0123456789ABCDEF

  - Result: 0x2C1F

The data logger CRC calculation is performed on all the data contained between the markers: ***—DOWNLOAD_START—*** (not included) and the character string ***CRC16: 0x***.(included).

Complete data logger example:

---DOWNLOAD_START---
Firmware version: 2.1.0\n
MacAddress: FA:FD:50:39:A1:2C\n
Name: BE_TEST_T3\n
Unit: Celsius degrees\n
Start date: 14/06/2019 12:00:00 +01:00\n
<DATA_START>\n
14/06/2019 12:00:10 +01:00: 26.62\n
14/06/2019 12:00:20 +01:00: 26.62\n
<DATA_END>\n
CRC16: 0xDF91\n
---DOWNLOAD_END---

The CRC value is calculated on all the data shown above in red. For this example, the value is thus 0xDF91.
A CRC calculator is available online at this website address: http://www.tahapaksu.com/crc/. Use the result contained in the CRC-CCITT field (0xFFFF).

IMPORTANT:
- Using the READ_DATA function does not erase the recorded and transmitted data.
- Datalogging is stopped until the transmission is not completed ("***—DOWNLOAD_END—***" marker). It continues automatically as soon as data transmission is finished.

Reading the datalogging start date

The command to read the datalogging start date that was sent to the tag is ***READ_START_DATE***. You must provide the BLE password for the command to be considered by the tag.

Command example:

READ_START_DATE ***PASSWORD_1***

Tag response:

- If the password is valid but datalogging is not running:
  READ_START_DATE: LOG not started!
- If the password is valid and datalogging is running:
  READ_START_DATE: DD/MM/YYYY HH:MM:SS +UU:UU
- If the password is invalid: *READ_START_DATE: ACCESS DENIED*

7.2.3. Calibration

7.2.3.1. General information

Calibration

You may calibrate the temperature readings measured by the tag. Calibration uses the format $\boldsymbol{a}T^2+\boldsymbol{b}T+\boldsymbol{c}$, where ***a***, ***b*** and ***c*** are configurable coefficients. These coefficients may only be written with tag configuration via NFC. The coefficients may be read in *connected mode*.

They are transmitted in the format ***X***e***Y***, where ***X*** is an integer between -32768 and 32767, followed by a superscript ***Y*** from -128 to 127. ***X***e***Y*** is equivalent to X.10^Y. Examples:

- 125e-5 = 0.00125
- 1e-2 = 0.01
- 12e-1 = 1.2

Examples of complete calibration:

- Sensor reading before calibration = 25.00°C. Calibration polynomial [c, b, a]: [5e-1, 1e0, 0e0]. The calculated value is therefore: T_{cal} = 25.5°C
- Sensor reading before calibration = 25.00°C. Calibration polynomial [c, b, a]: [0e0, 101e-2, 0e0]. The calculated value is therefore: T_{cal} = 25.25°C

It is not possible to change the calibration state (activated/deactivated) in connected mode when datalogging is running.

You may only define a first-degree polynomial (***b***T + ***c***), or a 0-degree polynomial (only ***c***, which can be used for testing). Coefficients are always sent in the following order: [c, b, a].

You may save a report containing calibration values (with the pair: setpoint value + measured value and calibration result) in tag memory. The report can be read in *connected mode*. Report example:

- Setpoint value 1: 25.00°C
- Measured value 1: 25.21°C
- Setpoint value 2: 75.00°C
- Measured value 2: 75.56°C
- Report result: 1 *(OK)*

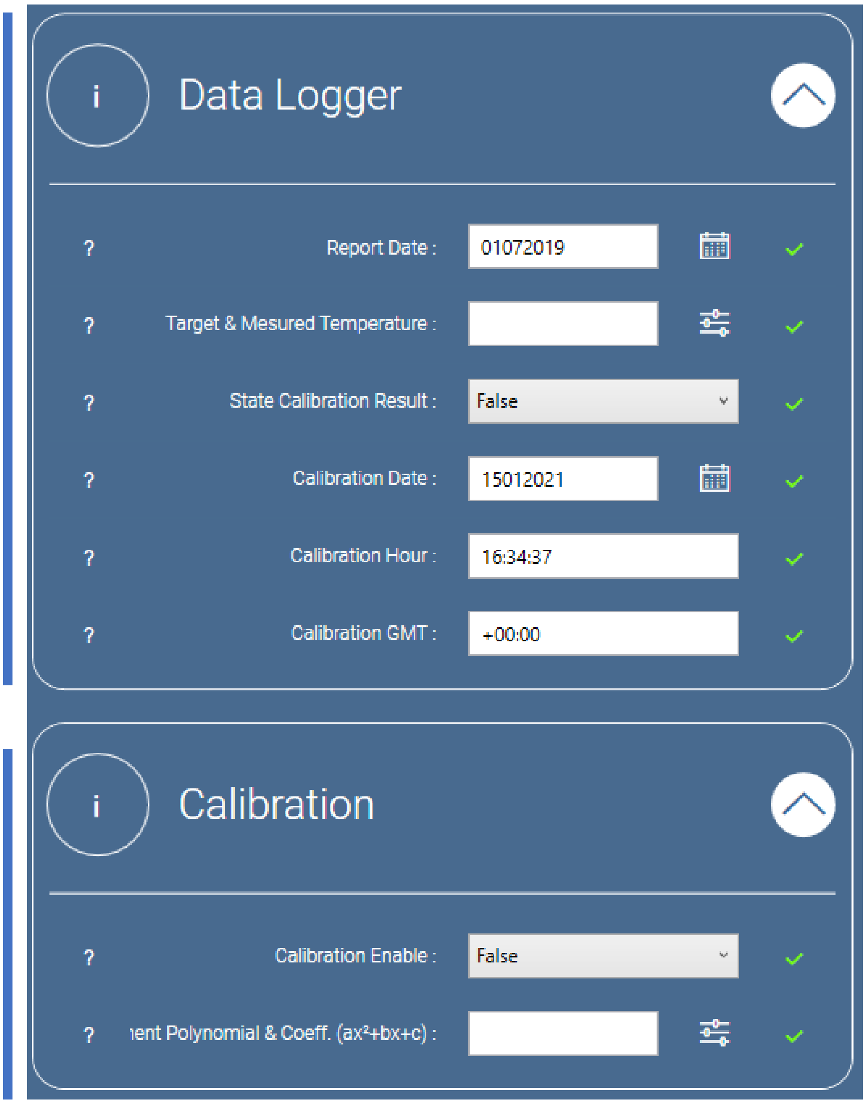
7.2.3.2. NFC configuration

The fields for calibration and the calibration report are configured using the Device Manager application.

The windows for configuring calibration and report fields are accessible via the ***Data Logger*** icon and via the ***Calibration*** icon.

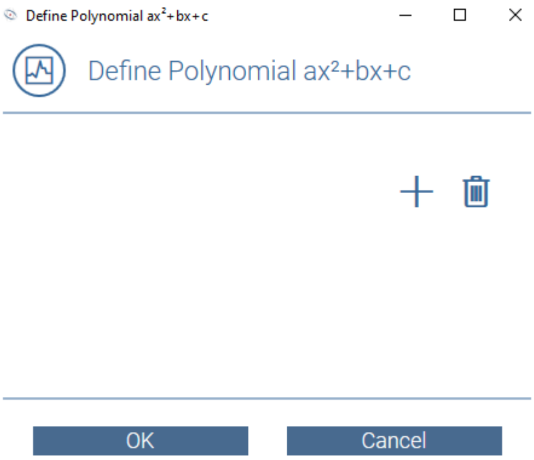
Dedicated EN12830 section: Fields related to Values report (values and report date) and Calibration date

Dedicated Calibration section: available for regular and EN12830 temperature products



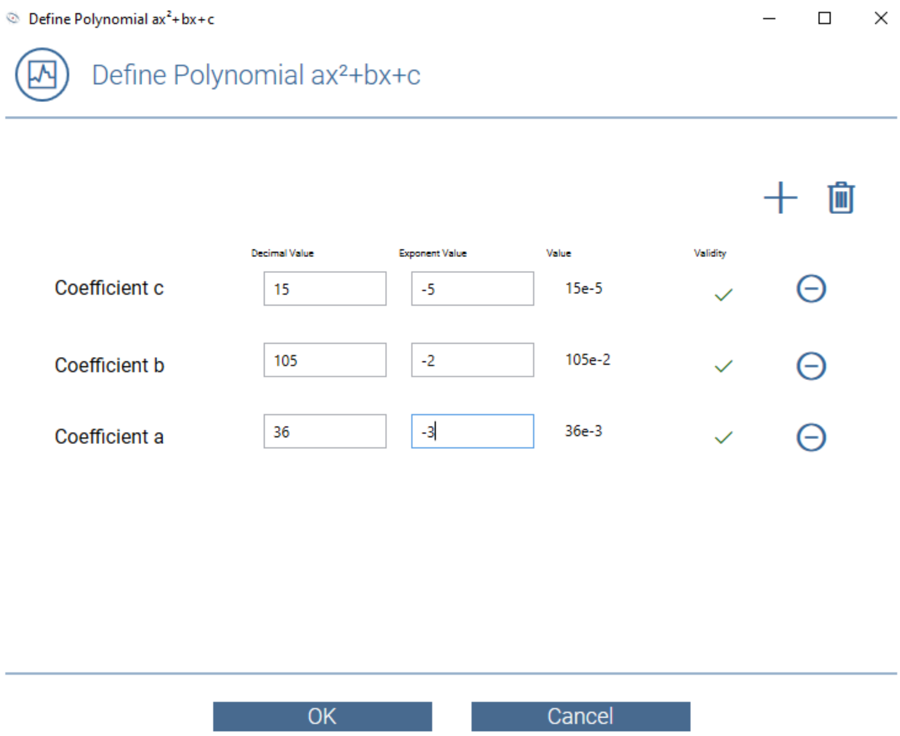
Configuring calibration coefficients

To activate calibration (use of the temperature correction polynomial), you must enter open the ***Calibration*** section of NFC parameters and set the parameter to ***Enable***. You can then click on the ***Coefficients configuration*** button to open the window:

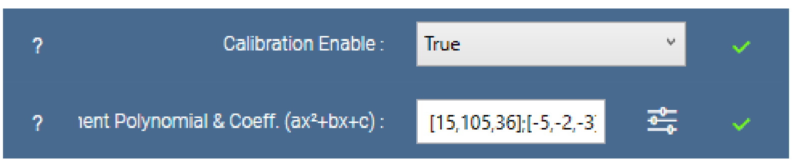


You can add a calibration coefficient by clicking on the **+** button.
You can add up to 3 coefficients.

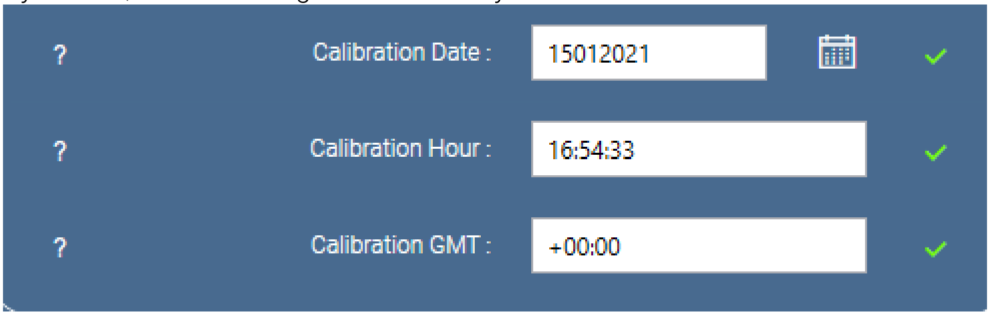
The window will check the coefficients and exponent values for integrity.



Click on ***OK*** when the values are set. They will appear on the previous window:



On the Data Logger window, you must then enter the current data, which is used to know when the calibration coefficients were modified.
By default, Device Manager automatically fills in the three date fields with the current date.



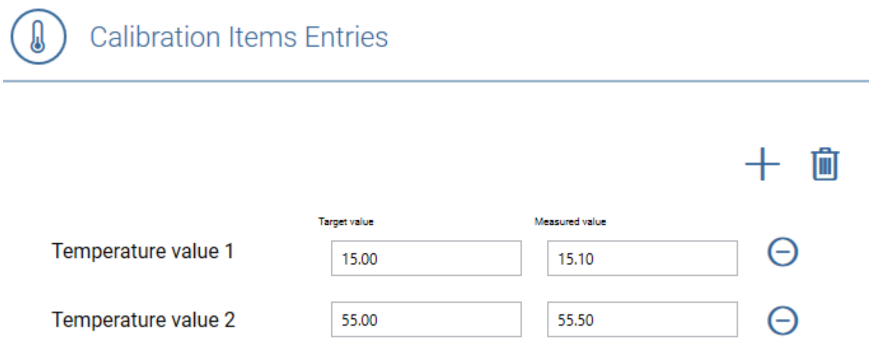
The date is in the format DDMMYYYY for the day; hh:mm:ss for the time; and +HH:MM for the time zone.

Do not forget to write the complete NFC configuration to the tag.

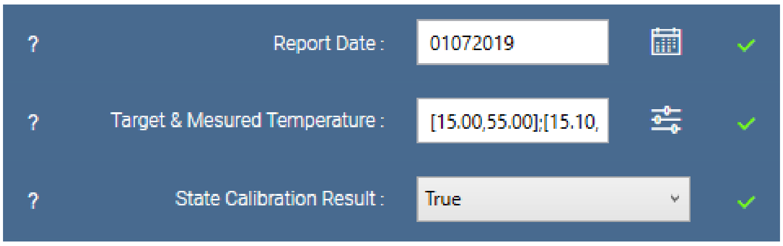
When the write pop-up window is displayed, you may remove the tag from the NFC reader. Calibration parameters will now be considered.

Writing the calibration value report

To save a report with calibration values, select the date in the calendar in the ***Data Logger*** window. Fill in the target/measured values using the icons and to the ***Target & Measured Temperatures*** field.



When you click on OK, the coefficients are displayed like this:



Lastly, enter State Calibration Result (true/false).

Do not forget to write the complete NFC configuration to the tag.

When the write pop-up window is displayed, you may remove the tag from the NFC reader. The calibration report is now stored in tag memory and accessible in connected mode.

7.2.3.3. Configuration of BLE connected mode

Activating / deactivating calibration

The command to activate temperature calibration is ***SET_CALIB_EN***. You must provide the BLE password for the command to be considered by the tag. Then you must provide a Boolean 1 (calibration activated) or 0 (calibration deactivated).

Command example:

SET_CALIB_EN ***PASSWORD_1*** *1*

Tag response:

- If the password is valid and the command contains "1": *SET_CALIB_EN: 1*
- If the password is valid and the command contains "0": *SET_CALIB_EN: 0*
- If the password is invalid: *SET_CALIB_EN: ACCESS DENIED*
- If the data logger is already running: *SET_CALIB_EN: LOG already started*

Reading coefficients

The command to read the tag's calibration coefficients is ***READ_CALIB_COEF***. You must provide the BLE password for the command to be considered by the tag.

Command example:

READ_CALIB_COEF ***PASSWORD_1***

Tag response:

- If the password is valid and no coefficients are used: *READ_CALIB_COEF: No polynomial values used*
- If password is valid: *READ_CALIB_COEF: Success*
- If the password is invalid: *READ_CALIB_COEF: ACCESS DENIED*

The READ_CALIB_COEF command starts a download of the information contained in the stored calibration coefficient report. The format is as follows:

Calibration coefficient date: 05072019 14:43:18 +01:00*n*
Calibration coefficients: c=1e0 b=3e-2 a=125e-5

Reading calibration status

The command to read the tag's calibration status is ***READ_CALIB_EN***. You must provide the BLE password for the command to be considered by the tag.

Command example:

READ_CALIB_EN ***PASSWORD_1***

Tag response:

- If the password is valid and calibration is deactivated: *READ_CALIB_EN: 0*
- If the password is valid and calibration is activated: *READ_CALIB_EN : 1 c=1e0 b=3e-2 a=125e-5*
- If the password is invalid: *READ_CALIB_EN: ACCESS DENIED*

Retrieving the calibration value report

Retrieving report values

The command to read the values in the tag's calibration report is ***READ_REPORT_VAL***. You must provide the BLE password for the command to be considered by the tag.

If the command syntax is valid, the tag returns ***READ_REPORT _VAL: Success***. The tag then begins to transmit its data.

Command example:

READ_REPORT_VAL ***PASSWORD_1***

Tag response:

- If password is valid: *READ_REPORT_VAL: Success*
- If the password is invalid: *READ_REPORT_VAL: ACCESS DENIED*
- If the password is valid but no report is stored on the tag: *READ_REPORT_VAL: No values entered!*

The *READ_REPORT_VAL* command starts a download of the information contained in the stored calibration report. The format is as follows:

Calibration report date: 01012019\n
Nb of measures: 3\n
TargetVal1: 25.00\n
MeasVal1: 24.95\n
TargetVal2: 5.00\n
MeasVal2: 4.79\n
TargetVal3: 50.00\n
MeasVal3: 49.90\n

A newline character (\n) is placed at the end of each line.

Retrieving report results

The command to read the tag's calibration results state is ***READ_REPORT_RES***. You must provide the BLE password for the command to be taken into account by the tag.

Command example:

READ_REPORT_RES ***PASSWORD_1***

Tag response:

- If the password is valid and the result is "1": *READ_REPORT_RES: 1*
- If the password is valid and the result is "0": *READ_REPORT_RES: 0*
- If the password is invalid: *READ_REPORT_RES: ACCESS DENIED*

7.2.3.4. Retrieving and verifying data

Reading all data logger values

IMPORTANT: You may read stored data at any time, without having to stop datalogging.

4 connected mode commands are available for retrieving datal logger datas:

- **READ_DATA**: download the complete data Logger in chronological order (oldest first)
- **READ_INV_DATA:** download the complete data logger in reverse order (newest first)
- **READ_SP_DATA DD/MM/YYYY HH:MM dd/mm/yyyy hh:mm**: download the complete data Logger in chronological order between the dates DD/MM/YYYY HH:MM and dd/mm/yyyy hh:mm
- **READ_SP_INV_DATA DD/MM/YYYY HH:MM dd/mm/yyyy hh:mm**: download the complete data Logger in reverse chronological order between the dates DD/MM/YYYY HH:MM and dd/mm/yyyy hh:mm

For each commands the password should be specified. If the command syntax is valid, the tag returns ***READ_DATA: Success.*** The tag then begins to transmit its data.

Command example:

READ_DATA ***PASSWORD_1***

READ_INV_DATA ***PASSWORD_1***

READ_SP_DATA ***DD/MM/YYYY HH:MM dd/mm/yyyy hh:mm PASSWORD_1***

READ_SP_INV_DATA ***DD/MM/YYYY HH:MM dd/mm/yyyy hh:mm PASSWORD_1***

Tag response:

- If password is valid: *READ_DATA: Success*
- If password is invalid: *READ_DATA: ACCESS DENIED*
- If the password is valid but datalogging is not running: *READ_DATA: LOG not started!*

Once the command is received, the tag transmits all the data stored in data logger memory. Data is in the following format:

---DOWNLOAD_START---
Firmware version: 2.1.0*n*
MacAddress: 01:02:03:04:05:FE*n*
Name: TAG_LOCAL_NAME*n*
Unit: Celsius degrees*n*
Start date: 01/04/2019 11:26:33 +01:00*n*
<DATA_START>*n*
01/04/2019 11:26:33+01:00: 23.34\n
01/04/2019 11:26:43+01:00: 23.44\n
[...]
<DATA_END>*n*
CRC16: 0x0D06*n*
---DOWNLOAD_END---

The newline characters (*n*) are not visible, but they must be considered for CRC calculation.

Field explanation:

- --- *DOWNLOAD_START*--- : marker indicating the start of transmission by data logger
- *Firmware version*: data logger (tag) firmware version
- *MacAddress*: tag's unique ID number
- *Name*: tag's name as configured in NFC memory
- *Unit*: unit used for temperature values (°C in this example)
- *Start date*: datalogging start as sent by the ***DATALOGGER_START*** command
- *<DATA_START>*: marker indicating the start of temperature data transmission
- 01/04/2019 11:26:33+01:00: 23.34: Complete example of temperature data with timestamp
- *<DATA_END>*: marker indicating the end of temperature data transmission
- *CRC16:* circular redundancy check calculated from transmitted data
- --- *DOWNLOAD_END*--- : marker indicating the end of transmission by data logger

Reading temperature data

In the previous example, temperature readings are formalized as follows:

01/04/2019 11:26:33+01:00: 23.34

- ***01/04/2019*** corresponds to the date on which the temperature was read: 1 April 2019 in this example.
- ***11:26:33*** corresponds to the time at which the temperature was read.
- ***+01:00*** corresponds to the UTC time zone, provided with the DATALOGGER_START command.
- ***23.34*** corresponds to the temperature value expressed in the unit transmitted by the data logger, 23.34°C in this example.

Checking data logger CRC

Datalogger data is controlled by a CRC16 calculation (*Cyclic Redundancy Check).* CRC details are as follows:

- CRC-16-CCITT algorithm
- 0x1021 polynomial
- Initialization value: 0xFFFF

Calculation example: Input data (ASCII type input) 0123456789ABCDEF

- Result: 0x2C1F

The data logger CRC calculation is performed on all the data contained between the markers: ***—DOWNLOAD_START—*** (not included) and the character string ***CRC16: 0x***.(included).

Complete data logger example:

---DOWNLOAD_START---
Firmware version: 2.1.0\n
MacAddress: FA:FD:50:39:A1:2C\n
Name: BE_TEST_T3\n
Unit: Celsius degrees\n
Start date: 14/06/2019 12:00:00 +01:00\n
<DATA_START>\n
14/06/2019 12:00:10 +01:00: 26.62\n
14/06/2019 12:00:20 +01:00: 26.62\n
<DATA_END>\n
CRC16: 0xDF91\n
---DOWNLOAD_END---

The CRC value is calculated on all the data shown above in red. For this example, the value is thus 0xDF91.
A CRC calculator is available online at this website address: http://www.tahapaksu.com/crc/. Use the result contained in the CRC-CCITT field (0xFFFF).

IMPORTANT:

- Using the READ_DATA function does not erase the recorded and transmitted data.
- Datalogging is stopped until the transmission is not completed (“***—DOWNLOAD_END—***” marker). It continues automatically as soon as data transmission is finished.

Reading the datalogging start date

The command to read the datalogging start date that was sent to the tag is ***READ_START_DATE***. You must provide the BLE password for the command to be considered by the tag.

Command example:

READ_START_DATE ***PASSWORD_1***

Tag response:

- If the password is valid but datalogging is not running: *READ_START_DATE: LOG not started!*
- If the password is valid and datalogging is running: *READ_START_DATE: DD/MM/YYYY HH:MM:SS +UU:UU*
- If the password is invalid: *READ_START_DATE: ACCESS DENIED*

7.2.4. Summary of BLE commands in connected mode

Order	Information	
DATALOGGER_START	**Description**	Starts data logger and records start date/time
	Example	DATALOGGER_START **PASSWORD_1** 05/06/2019 11:20:00 +01:00
DATALOGGER_STOP	**Description**	Stops datalogging on the tag
	Example	DATALOGGER_STOP **PASSWORD_1**
READ_DATA	**Description**	Retrieves all data logger values
	Example	READ_DATA **PASSWORD_1**
READ_START_DATE	**Description**	Reads data from the start date/time defined in the tag
	Example	READ_START_DATE **PASSWORD_1**
READ_CALIB_COEF	**Description**	Reads calibration polynomial coefficients
	Example	READ_CALIB_COEF **PASSWORD_1**
SET_CALIB_EN	**Description**	Activates / deactivates calibration
	Example	SET_CALIB_EN **PASSWORD_1** 1
	Description	Returns calibration status (activated / deactivated)

READ_CALIB_EN	**Example**	READ_CALIB_EN **PASSWORD_1**
READ_REPORT_VAL	**Description**	Reads values contained in calibration report
	Example	READ_REPORT_VAL **PASSWORD_1**
READ_REPORT_RES	**Description**	Reads calibration report results
	Example	READ_REPORT_RES **PASSWORD_1**

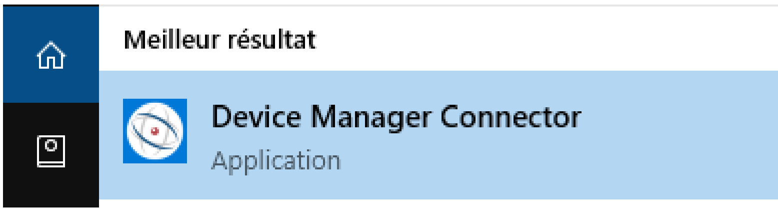
7.2.5. Example using Device Manager Connector

The **Device Manager Connector** application executes the commands for ELA Innovation Bluetooth® tags. This software offers a user interface that enables you to test features in a tag's BLE connected mode. You can use Device Manager Connector to send basic commands to control tags in ELA Innovation's Blue product line.

The application is available for free download from the Microsoft Store. Version 1.2.0 (and higher) enables you to manage data logger functions.

Launching the application

Open Device Manager Connector by clicking on the associated icon or by entering the application name in the Windows search bar: **Device Manager Connector**.

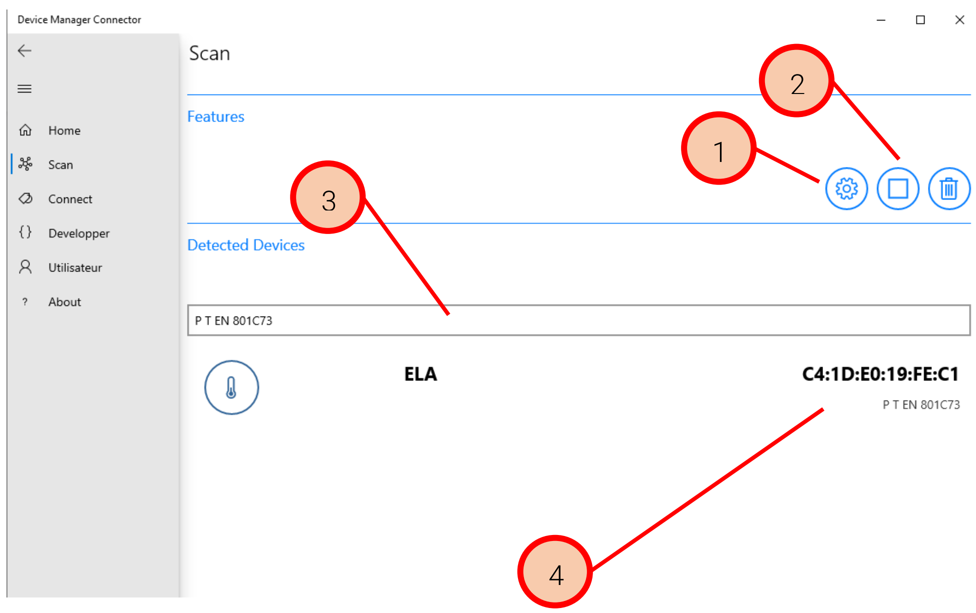


Once the application is running, we recommend that you use the "**Scanner**"  function to discover tags within wireless range and select the tag that you want to use. There are two ways to access this function:

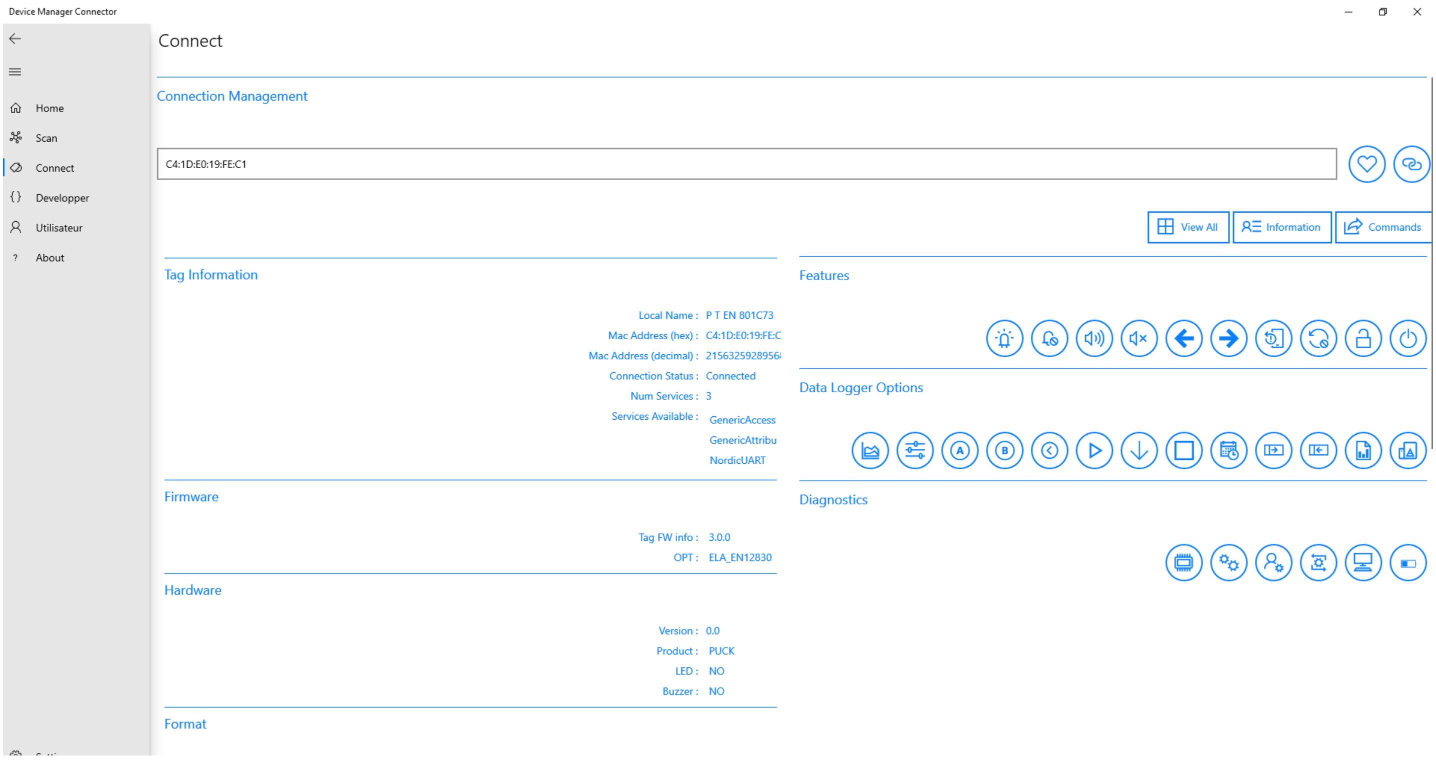
- The IoT icon **(1)** in the left-hand menu bar
- The **Scan** button on the application's home page **(2)** (Navigate menu)

With the Scanner function, you can start "listening" for Bluetooth communications by clicking on the Bluetooth Scan button. At any time, you may define a filter to more precisely target the tag that you want to access. The search bar enables you to browse through all the detected tags and refine your selection based on MAC address, tag name, etc.

- **(1)**: scan preferences (show only tags with a default name)
- **(2)**: start / stop scan
- **(3)**: search bar
- **(4)**: list of detected tags

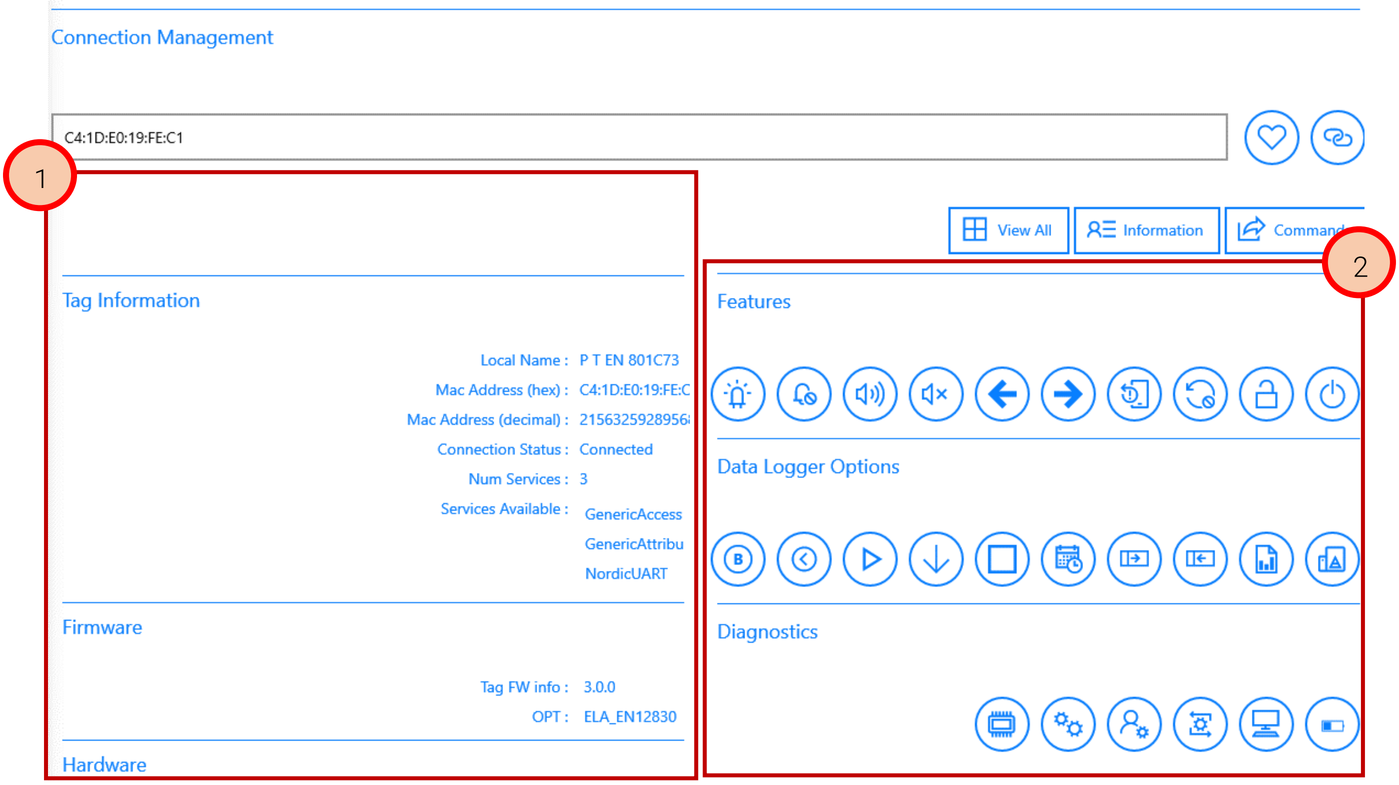


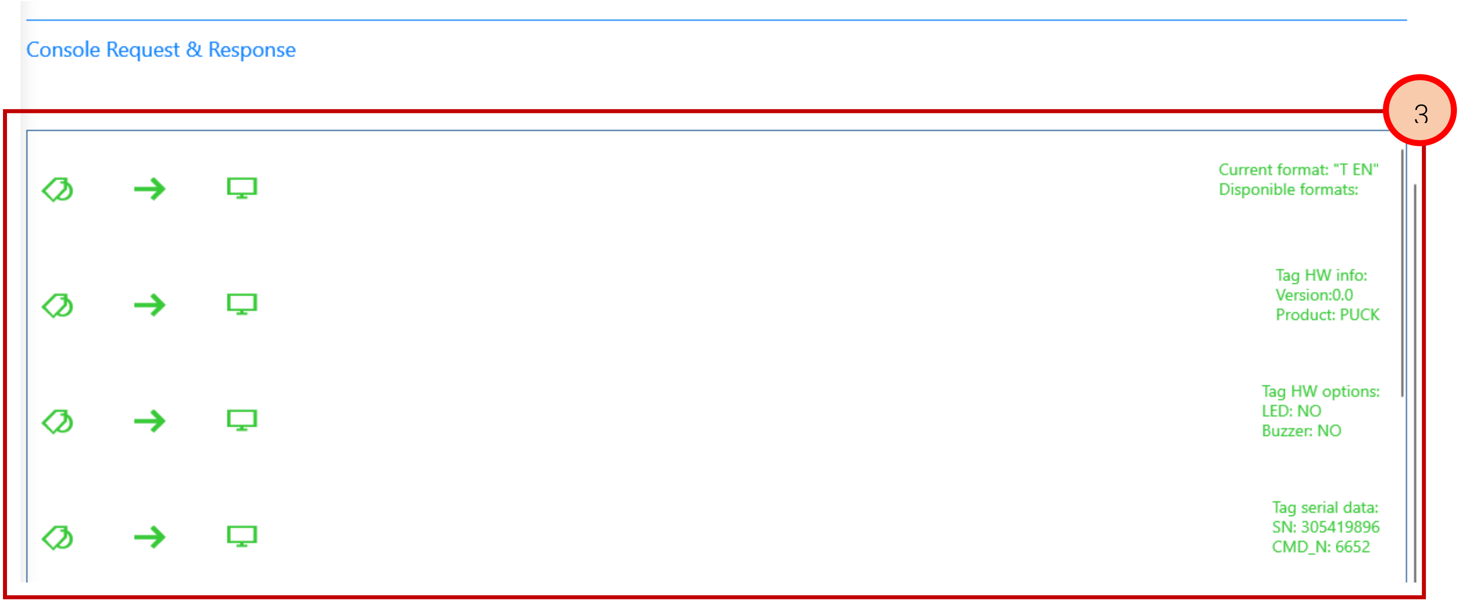
- Simply click on a tag in the list to connect to that tag. The application then switches to the connection screen, which displays all the information and available commands for the selected tag.



The tag connection screen is divided into three main sections:

- **(1)**: general information
- **(2)**: all available commands
- **(3)**: command console (queries & responses)





General information

The information section displays all tag-related data. Each part is updated based on the data present (or not present) on the tag. The firmware version for **EN 12830** tags is **2.1.0** or **3.0.0**, with an OPT option: **ELA_EN12830**.

Commands

All the commands available for the data logger are available in the **Data Logger Options** section. The list below describes the buttons and their functions.

Order	Information	
DATALOGGER_START		Starts data logger and records start date/time
DATALOGGER_STOP		Stops datalogging on the tag
READ_DATA		Retrieves all data logger values
READ_START_DATE		Reads data from the start date/time defined in the tag
READ_CALIB_COEF		Reads calibration polynomial coefficients

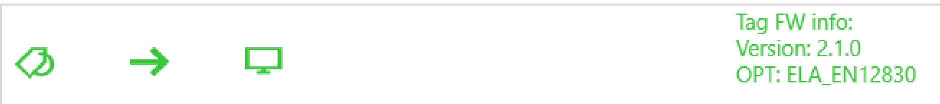
SET_CALIB_EN		Activates / deactivates calibration
READ_CALIB_EN		Returns calibration status (activated / deactivated)
READ_REPORT_VAL		Reads values contained in calibration report
READ_REPORT_RES		Reads calibration report results

Console

The console section shows the transmission/reception actions for Bluetooth® commands. This display is for informational purposes, enabling you to follow:

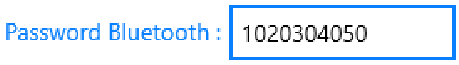
- Commands sent to the tag (in the image below, the pictograms represent the sending action; the command being sent is indicated on the right).

- Commands received by the tag (in the image below, the pictograms represent the receiving action; the information received is indicated on the right).



For each command sent to the data logger, you must enter the password to confirm whether the command should be sent. In Device Manager Connector, the password can be configured in the ***Settings Window***.

Go to the Bluetooth Configuration section to define the ***Bluetooth Password***:



You can also define some more parameters.

The password to use here is the one that was defined during tag configuration. For more information, see Chapter 3 - General (Security).

Starting the data logger

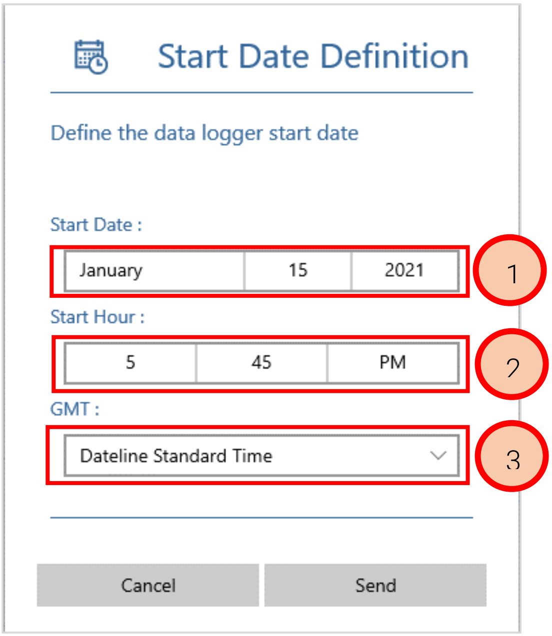
To use the application to retrieve data from a data logger, the data logger must first have been started (with the DATALOGGER_START command). To do this from the tag connection screen, click on the PLAY button to start the data logger.

The application prompts you to enter the password, which is required to execute commands.

A pop-up window enables you to define the start time. Fill in the following fields:

- **(1)**: Data logger start date
- **(2)**: Data logger start time
- **(3)**: Time zone in which the data logger is started

Then click on **Send** to transmit the settings or **Cancel** to return to the previous screen without taking any action.

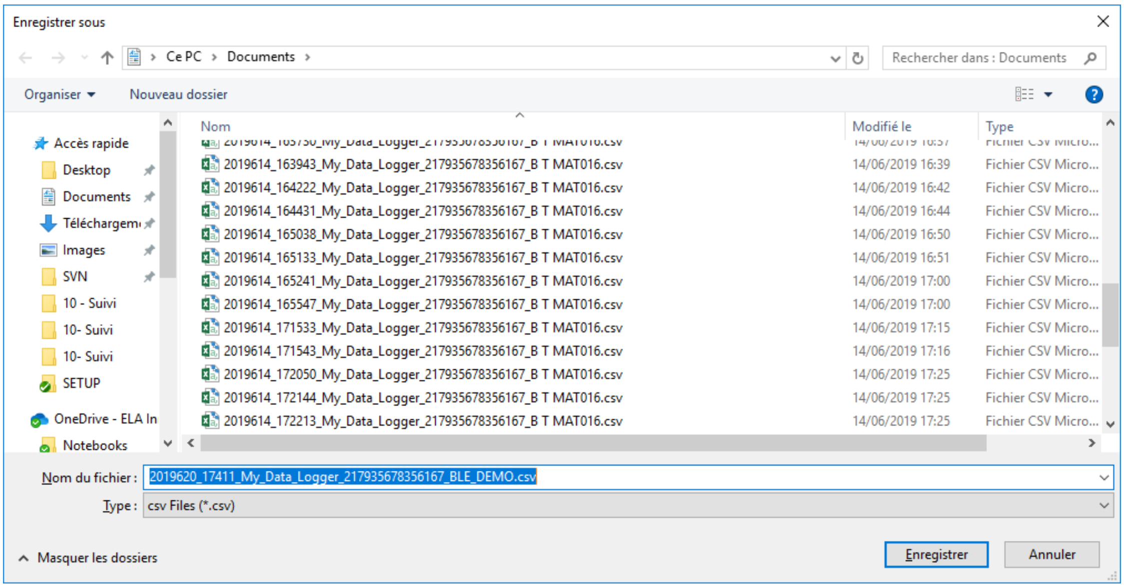


Stopping the data logger

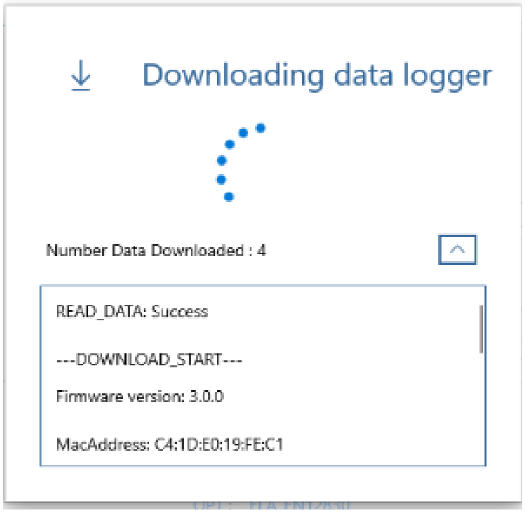
To stop data logger recording, click on the Stop button. When you click on the button, the password pop-up window opens. Enter the password to continue.

Reading the data logger

To read the data stored on the data logger, click on the READ_DATA button or on Download. When you click on the button, the password pop-up window opens. Enter the password to continue. The "Save as" window opens so you can save the data locally in a .CSV format file. The file name field contains a proposed default name for the file.



A download progress window shows the status of the file download. If the data logger is not currently running, an error message is displayed. Otherwise, data is retrieved progressively, as shown below.



7.3. BLUE PUCK MAG

If you wish to put the tag to a metallic surface, we highly recommend you use our **TAG HOLDER** to avoid any disturbances the metal could cause to the radio frames Transmission. Note that you can use other magnets, but we only guarantee the functioning with our products.

7.3.1. Sensor installation

The MAG sensors use a hall effect sensor to detect the presence or not of a magnet. In order to be properly detected, the magnet needs to be side by side with the top of the tag, near our logo. On the case, you will see an arrow to show you on which side the magnet should be.

On the pictures below, you will find some use cases examples for the MAG sensors.

7.3.2. Tag Operation

These sensors are used to detect if the industrial tools are open or closed. If the magnet is not detected, the sensor state within its frame will be 0, and will change to 1 when the magnet is detected. When nothing is happening, the tag will send a frame each to a certain period: the BLE Emit Period in the NFC settings. In the case of an event (When the magnet is detected in this case), the tag will trigger the **fast event frame functionality**.

• This functionality sends data to a faster recurrence (equal to one tenth of the advertising tag recurrence set in NFC). Data contained in this frame is the same as that contained in the simple advertising frame, but its recurrence varies.
• Fast frames appear during a period equal to the advertising period, and with a recurrence equal to one tenth of it. Thus, there are 10 frames.

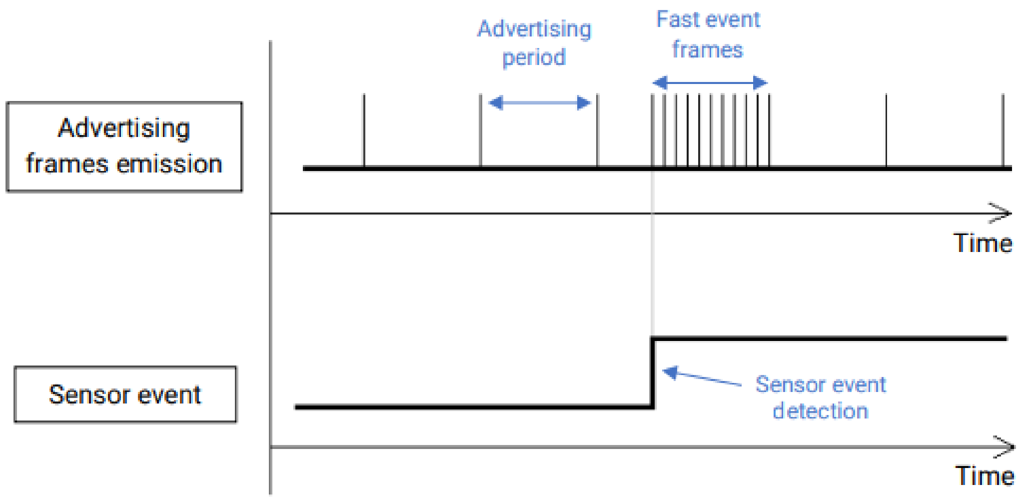


Figure 2: Demonstration diagram of fast event frames occuring during an event.

7.4. Blue PUCK PIR

Blue PUCK PIR (IDF25249X) is adapted for presence or movement detection of people in a 20 cm to 5 m range (configurable) with 120° Field of View angle.

Blue PUCK PIR uses a differential Pyroelectric Infrared Radial (PIR) sensor and a Fresnel lens. It senses infrared light flux fluctuation. In this way, warm object (body or body parts) that naturally emits infrared light, can be detected provided that they are in the detection cone and sufficiently mobile (still target won't be detected).

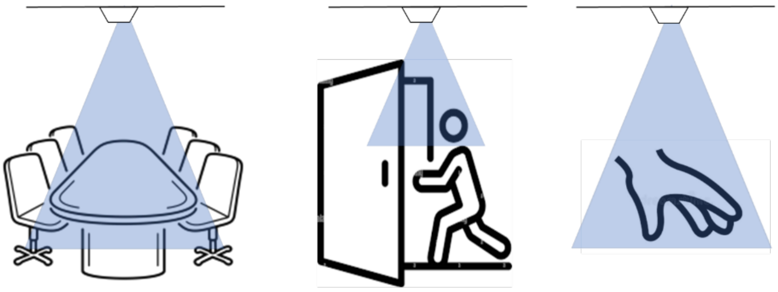
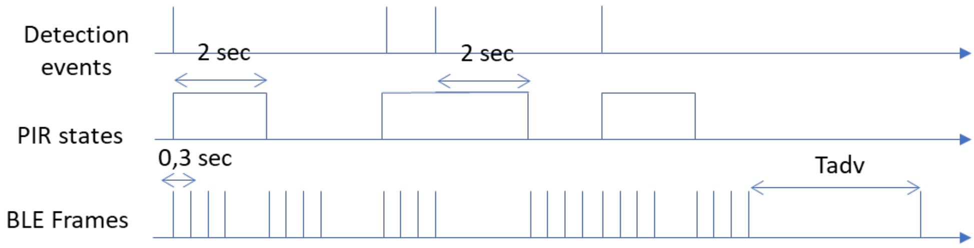


Figure 7: PROXIR use case example: meeting room occupancy detection, entrance detection, hand picking detection.

7.4.1. Operation – Configuration

The standard BLE frame format of the Blue PUCK PIR is given in the table below. The PIR state is a 1 bit value equals to 1 if a detection has occurred. This value is hold up for 2 seconds after the detection and until further detection occurs (see figure 2).

In its standard configuration, The Blue Puck PIR transmits a burst of 10 advertising frame (periods 0,3 seconds) on each PIR state value change. If the PIR state has not change for a duration equals to the advertising period advertising h frame is sent.



Frame type		Service Data	Mfr Spec. Data
Version		≥3.0.1	≥3.0.1
Octets trame	1	Length: 0x02	Length: 0x02
	2	Type : 0x01	Type : 0x01
	3	Data : 0x06	Data : 0x06
	4	Length: 0x05	Length: 0x06
	5	Type : 0x16	Type: 0xFF
	6	*Rainfall* Carac. LSB : 0x78	ELA_CIN_LSB : 0x57
	7	*Rainfall* Carac. MSB : 0x2A	ELA_CIN_MSB: 0x07
	8	PIR Data (cnt + state) LSB	PIR_DATA_ID: 0x92
	9	PIR Data (cnt + state) MSB	PIR Data (cnt + state) LSB
	10	Length: ≤0x0F	PIR Data (cnt + state) MSB
	11	Type : 0x09	Length: ≤0x0F
	12	Name[0]	Type : 0x09
	13	Name[1]	Name[0]
	14	Name[2]	Name[1]
	15	Name[3]	Name[2]
	16	Name[4]	Name[3]
	17	Name[5]	Name[4]
	18	Name[6]	Name[5]
	19	Name[7]	Name[6]
	20	Name[8]	Name[7]
	21	Name[9]	Name[8]
	22	Name[10]	Name[9]
	23	Name[11]	Name[10]
	24	Name[12]	Name[11]
	25	Name[13]	Name[12]
	26	Name[14]	Name[13]
	27	Not used	Name[14]
	28	Not used	Not used
	29	Not used	Not used
	30	Not used	Not used

7.4.2. Sensitivity and angle of view

The sensitivity of the BLUE PUCK PIR can be tuned to optimize its operation. High sensitivity will allow detecting smaller target at higher range but can lead to higher wrong detection rates. Sensitivity level can be configured according to four predefine level. Following table gives the correspondence between sensitivity level and detection range for a complete human body.

Sensitivity level	Detection range (full human body)
0	20 cm
1	50 cm
2	1 m
3	2 m
4	5 m

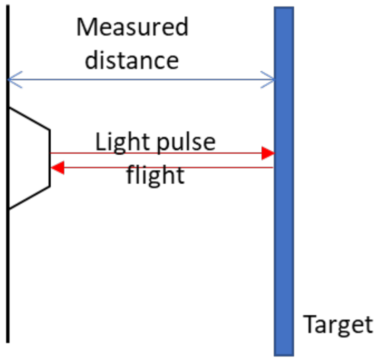
The Blue PUCK PIR has a 120° angle of view (isotropic cone of detection thanks to Fresnel lense). Following table gives the correspondences between covered area and distance for maximum sensitivity:

Distance	Covered area radius
20 cm	34 cm
50 cm	86 cm
1 m	1,73 m
2 m	3,46 m
5 m	8,6 m

7.5. PROXIR

Blue PUCK PROXIR (IDF25252X) is adapted for detection and distance measurement of any reflecting object in a 1 mm to 80 cm range with 15° Field of View angle.

Blue PUCK PIR uses a Time Of Flight (TOF) infrared sensor. It emits a infrared laser pulse and measure the time taken by this pulse to reach the target, be reflected and comes back to the sensor.



7.5.1. Sensor installation

Measurement precision is highly dependent on the reflected light intensity. Therefore, distance measurement precision will depend on the target reflectivity, distance, and orientation. Best installation practices will thus consist in insuring that:

- The target to be detected is made of a highly reflecting material (see table below)
- The target to be detected is at a distance lower between 20 and 80 cm
- The tag top surface (with the IR window) is parallel to the target surface.

Direct sun light illumination is avoided.

The sensor is installed vertically to avoid dust and moisture accumulation should be preferred

Following table give typical precision depending on measure range and target material.

	1 mm – 20 cm	20 cm -40 cm	40 cm - 80 cm
Wood	15%	1%	2%
Rubber (black)	> 20%	10 %	5 %
Copper	15 %	1%	1%
PVC (black)	> 20%	3%	4%

7.5.2. Sensor output

The standard BLE frame format of the Blue PUCK PROXIR is given in the table below. The PROXIR data consist in a 2 Bytes value with 15 bits coding for the measured distance and 1 bit coding for the measurement integrity.

Measurement integrity is equal to 0 if the measurement is trustworthy and equals to 1 otherwise. Lack of integrity can be explained by a too large detection range, too low target reflectivity or two high ambient light.

Frame type		Service Data	Mfr Spec.Data
Version		≥4.0.0	≥4.0.0
Octets trame	1	Length: **0x02**	Length: **0x02**
	2	Type : 0x01	Type : 0x01
	3	Data : 0x06	Data : 0x06
	4	Length: **0x05**	Length: **0x06**
	5	Type : 0x16	Type: 0xFF
	6	*Altitude* Carac. LSB : 0x8E	ELA_CIN_LSB : 0x57
	7	*Altitude* Carac. MSB : 0x2A	ELA_CIN_MSB: 0x07
	8	Distance Data (mm) + integrity bit LSB	PROXIR_DATA_ID: 0xA2
	9	Distance Data (mm) MSB	Distance Data (mm) + integrity bit LSB
	10	Length: **≤0x0F**	Distance Data (mm) MSB
	11	Type : 0x09	Length: **≤0x0F**
	12	Name[0]	Type : 0x09
	13	Name[1]	Name[0]
	14	Name[2]	Name[1]
	15	Name[3]	Name[2]
	16	Name[4]	Name[3]
	17	Name[5]	Name[4]
	18	Name[6]	Name[5]
	19	Name[7]	Name[6]
	20	Name[8]	Name[7]
	21	Name[9]	Name[8]
	22	Name[10]	Name[9]
	23	Name[11]	Name[10]
	24	Name[12]	Name[11]
	25	Name[13]	Name[12]
	26	Name[14]	Name[13]
	27	Not used	Name[14]
	28	Not used	Not used
	29	Not used	Not used
	30	Not used	Not used
	31	Not used	Not used

8. NORMS & STANDARDS

FCC Statement

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

1. This device may not cause harmful interference; and
2. This device must accept any interference received, including interference that may cause undesired operation.

Industry Canada Statement

This device complies with ISED's licence-exempt RSSs. Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation.

CE Mark

- FCC Mark

- RoHS Certified

- Bluetooth 4.2