WIREPAS RANGE SENSORS 2.X.X – USER GUIDE

Specifications subject to change without notice. Non-contractual document.

Table of contents

1 GENERAL INFORMATION ON MESH NETWORKS

Mesh networks

A mesh network is a network topology (wired or wireless) in which all hosts are connected “peer-to-peer” without a centralized hierarchy, thus creating a net-type structure. With this architecture, every node can send, receive, and relay data. This eliminates the presence of “backbone” points that can isolate parts of the network in case of malfunction. If a host stops working, data simply takes another route to its destination. A mesh network can relay data via “flooding” (broadcasting data so that it is received by all nodes within direct wireless range). It can also use predefined routes, in which case the network must plan for uninterrupted connections or alternative routes.

Wirepas Mesh

The Wirepas Mesh protocol is a wireless network protocol that uses a multi-jump, self-organizing, and decentralized design. Decentralized network topology enables extremely dense network deployment.

Wirepas focuses on providing a connectivity solution that is highly **reliable**, **optimized**, **scalable**, and **easy to deploy**.

This solution was specifically designed to meet two major challenges facing wireless mesh networks: network reliability regardless of its size and density; and low energy consumption by router devices in the network.

Information about Wirepas Mesh technology is available here:

www.wirepas.com

2 PRODUCTS: BEACONS

BLUE MESH SENSORS

Blue PUCK **T MESH**	Blue PUCK **RHT MESH**	Blue PUCK **DO MESH**
Blue PUCK **MAG MESH**	Blue PUCK **MOV MESH**	Blue PUCK **PIR MESH**
Blue PUCK **AI MESH**	Blue PUCK **DI MESH**	

3 PRODUCTS: BLUE MESH GATEWAY

Raspberry Pi Gateway and Wirepas Wireless Dongle (2.4 GHz)

Raspberry Pi3 B+ or Pi4

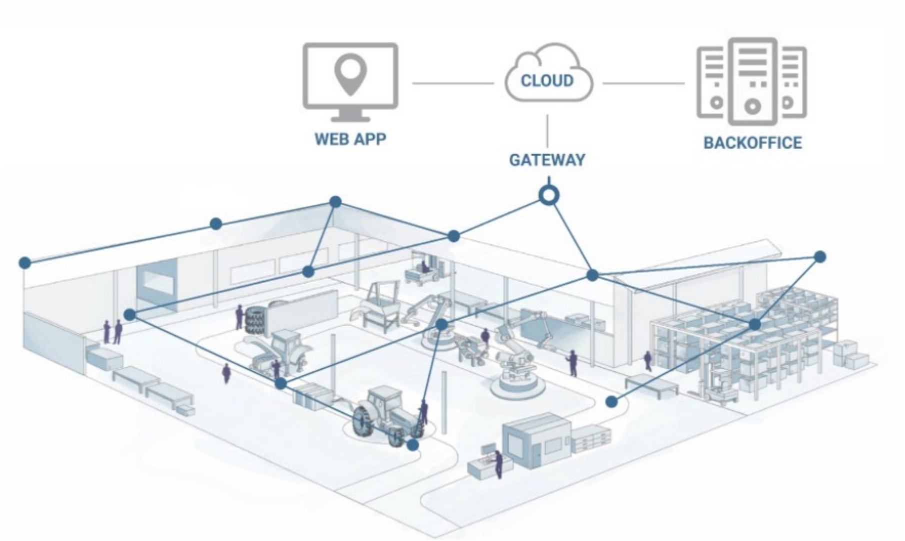
Wirepas Mesh 2.4 GHz wireless dongle

SolidRun Gateway

SolidSense N6

4 BLUE MESH SENSOR OPERATION

4.1 WIREPAS MESH SENSOR NETWORK DIAGRAM



Network components	Products
● MESH sensor	BLUE PUCK T MESH - RHT MESH - MAG MESH - MOV MESH - PIR MESH - AI MESH - Digi IN MESH - Digi OUT MESH
○ GATEWAY	ELA Innovation MESH Gateway

The mesh sensor network comprises two different types of devices: sensor tags and one or more gateways. Each element may be added to an existing network. Adding new devices to an existing network is made easier with automatic connection and integration into the mesh network. Once connected, each element in the network can send its sensor information to a gateway. Mesh connectivity between tags, and message forwarding, are handled automatically and dynamically.

SENSOR TAG

A sensor tag is a tag in the BLUE PUCK MESH product line equipped with a sensor that periodically transmits its sensor data to an MQTT broker. Connection between tags and the relaying of sensor information to the gateway is handled automatically and dynamically in the Wirepas Mesh network.

GATEWAY

A software gateway links the BLUE Mesh tag network to the server. The gateway receives a stream of data sent by tags and forwards it to an MQTT broker. The received data is then processed by an application to display and/or further process sensor data.

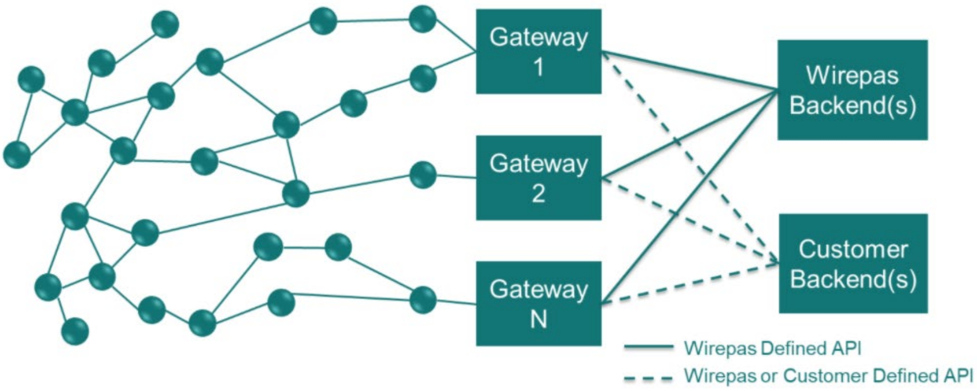
4.2 OPERATING MODE

Sensor Tag in Autorole mode

The Sensor Tag is configured in a mode referred to as "Autorole". This mode enables all tags in the network to manage themselves. Each tag can send its sensor data over the network, and each tag can relay information flowing across the network. Management is dynamic and automatic. The only constraint is for the tag to be at an acceptable connection distance to one or more neighboring tags.

The gateway receives network data and transmits it to one or more back-end servers.

The example below shows the connection between tags, gateway(s), and back-end(s).



There are two ways to retrieve data:

- Connect to the Wirepas API and retrieve the data stream respecting the format required by Wirepas.
- Develop your own API and handle message collection yourself.

Sensor data must be retrieved from the various topics presented in the table on the next page.

Messages are received in a generic format described in Wirepas documentation and encoded in protocol buffer format: https://developers.google.com/protocol-buffers.

Complete information related to message reception and encoding is provided in the following section of the Wirepas GitHub:

https://github.com/wirepas/backend-apis/tree/master/gateway_to_backend

Sensor Data and MQTT topics

Data is sent by the gateway to a specific MQTT broker the server. The list of topics corresponding to the various sensors in the BLUE MESH product line are provided in the following table.

Specifications subject to change without notice. Non-contractual document.

Functionality	Endpoint source/ destination	Data			Results	
		Type	Length	Received data	Interpreted data	Value
Battery level	11/ 11	01	02	BD 0B	0x0BBD	3.005 v
Temperature ***BLUE PUCK T MESH***	100/ 100	02	02	92 0B	0x0B92	2962 -> 29.62°
Temperature and Humidity ***BLUE PUCK RHT MESH***	110/ 110	03	04	27 00 BA 0B	0x0027 0x0BBA	39% 3002 -> 30.02°
Digital input ***BLUE PUCK DI MESH***	120/ 120	04	06	01 00 2A 00 00 00 00 00 2A 00 00 00	00 01 00 00 00 2A 00 00 00 00 00 2A	Input activated: 01 Counter: 42 Input deactivated: 00 Counter: 42
Digital output ***BLUE PUCK DO MESH***	130/ 130	05	06	01 00 01 36 00 00 00 00 01 36 00 00	00 01 00 00 36 01 00 00 00 00 36 01	Output activated: 01 Counter: 13825 Output deactivated: 00 Counter: 13825
Tag Removed (alerts) ***BLUE SLIM ID+ MESH***	140/ 140	06	06	01 00 12 00 00 00 00 00 12 00 00 00	00 01 00 00 00 12 00 00 00 00 00 12	Tag removed: 01 Counter: 18 Tag not removed: 00 Counter: 18
Magnet detection ***BLUE PUCK MAG MESH***	150/ 150	07	06	01 00 B5 00 00 00 00 00 B5 00 00 00	00 01 00 00 00 B5 00 00 00 00 00 B5	Magnet detected: 01 Counter: 181 Magnet not detected: 00 Counter: 181
Movement detection ***BLUE PUCK MOV MESH*** ***MOV format***	160/ 160	08	06	01 00 2A 00 00 00 00 00 2A 00 00 00	00 01 00 00 00 2A 00 00 00 00 00 2A	Movement present Counter: 42 Movement absent Input deactivated Counter: 42
Acceleration ***BLUE PUCK MOV MESH*** ***ANG format***	170/ 170	09	06	B8 00 58 FF 8E 04	00 B8 FF 58 04 8E	X: 00 B8 → 184 milliG Y: FF 58 → -168 milliG Z: 04 8E → 1166 milliG
Analog reading ***BLUE PUCK AI MESH***	180/ 180	0A	02	C7 09	09 C7	Level: 09 C7 → 2.503 v

Specifications subject to change without notice. Non-contractual document.

Detection Presence **BLUE PUCK PIR MESH**	200/ 200	0C	06	01 00 B5 00 00 00 00 00 B5 00 00 00	00 01 00 00 00 B5 00 00 00 00 00 B5	Presence detected: 01 Counter: 181 No detection: 00 Counter: 181
Button press **BLUE LITE TOUCH MESH**	210/ 210	08	06	01 00 05 00 00 00 00 00 05 00 00 00	00 01 00 00 00 05 00 00 00 00 00 05	Button pressed: 01 MOV counter: 5 Button released: 00 MOV counter: 5

All frames sent to MQTT topics are comprised of sensor data followed by battery level data, in TLV format as described in the above table.

Commands and MQTT topics

These commands can only be used for devices that are always connected to the network (Anchor, Mobile HC and Sensor). Mobile beacons in NRLS mode (Mobile or Mobile+) cannot receive these commands due to the network disconnection during their sleep phase between two measurement updates.

LED, buzzer and DO activation commands

Functionality	Product	Endpoint source	Endpoint destination	Description	
				Command	ACK
LED ON	BLUE PUCK ID MESH BLUE PUCK BUZZ MESH BLUE COIN ID MESH BLUE ANCHOR	20	20	LED_ON	OK: 00 NOK: 01
LED OFF		20	20	LED_OFF	OK: 00 NOK: 01
LED ON Time *** "Time" in seconds**		20	20	LED_ON 10 *10 seconds	OK: 00 NOK: 01
BUZZ ON	BLUE PUCK BUZZ MESH	20	20	BUZZ_ON	OK: 00 NOK: 01
BUZZ OFF		20	20	BUZZ_OFF	OK: 00 NOK: 01
BUZZ ON Time *** "Time" in seconds**		20	20	BUZZ_ON 10 *10 seconds	OK: 00 NOK: 01

LEDBUZZ ON	BLUE PUCK BUZZ MESH	20	20	LEDBUZZ_ON	OK: 00 NOK: 01
LEDBUZZ OFF		20	20	LEDBUZZ_OFF	OK: 00 NOK: 01
LEDBUZZ ON Time *** "Time" in seconds**		20	20	LEDBUZZ_ON 10 *10 seconds	OK: 00 NOK: 01
DIGITAL Output ON	BLUE PUCK DO MESH	130	130	DIGI_ON	See "previous page" table
DIGITAL Output OFF		130	130	DIGI_OFF	See "previous page" table
DIGITAL Output ON Time *** "Time" in seconds**		130	130	DIGI_ON 10	See "previous page" table

Diagnostic commands

Functionality	Product	Endpoint source	Endpoint destination	Description	
				Command	Information
Battery level	BLUE PUCK ID MESH BLUE PUCK BUZZ MESH BLUE COIN ID MESH BLUE ANCHOR	50	50	GET_BATT_VOLTAGE	Returns the battery level
Hardware version		50	50	HW_VERS	Returns hardware version
Firmware version		50	50	FW_VERS	Returns firmware version
Sequence Number		50	50	SCRATCHPAD_INFO	Returns the value of the scratchpad sequence number

Response diagnostic commands

Functionality		Data	Results

	Endpoint source/ destination	Type	Length	Received data	Interpreted data	Value
Battery level	11/ 11	01	02	BD 0B	0x0BBD	3.005 v
Hardware version	50/ 50	-	-	13 05 24 10	0x10240513	(used for internal diagnostics)
Firmware version	50/ 50	-	-	02 00 40 70 04 01	0x0200 0x4070 0x0401	FW version 200 Stack version 4.0.70 PosApp version 4.0.1
Sequence Number	50/ 50	-	-	03 00 01 00	0x00 03 0x00 01	Scratchpad sequence number: 3 Processed scratchpad sequence number: 1

Request Sensor Data

It is possible to send a command to a device to demand it to immediately send its sensor data (in addition to the periodic data transmission):

Sensor	Endpoint source	Endpoint destination	Command
T	30	30	T_DATA
RHT	30	30	RHT_DATA
MOV	30	30	MOV_DATA
ANG	30	30	ANG_DATA
MAG	30	30	MAG_DATA
AI	30	30	AI_DATA
DI	30	30	DI_DATA
PIR	30	30	PIR_DATA
TOUCH	30	30	TOUCH_DATA

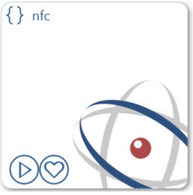
5 CONFIGURATION VIA NFC

1. Connect NFC reader to PC (ELA reference: NFC R/W 01 - ref. ACIOM177)

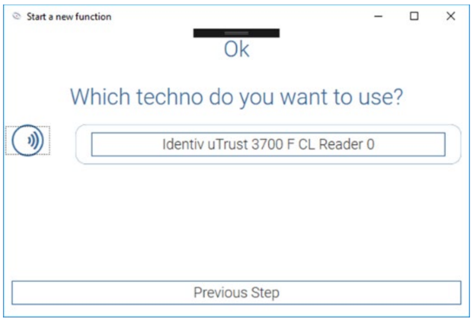
2. Start the **Device Manager** application (installed prior to use).

The application can be downloaded from the elainnovation.com website, in Support → Downloads

3. Launch the **NFC widget**



4. Click on the button to select the proposed **NFC reader**.


Start a new function

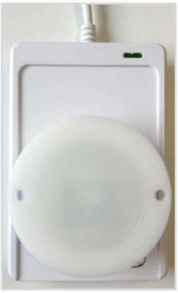
Ok

Which techno do you want to use?

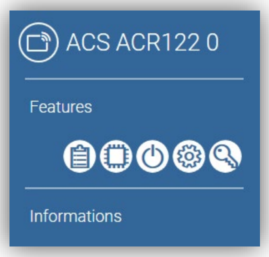
Identiv uTrust 3700 F CL Reader 0

Previous Step

5. Place the tag on the NFC reader as shown below.



6. Click on the **Configuration** icon to read tag parameters.



ACS ACR122 0

Features

Informations

Display raw configuration data.

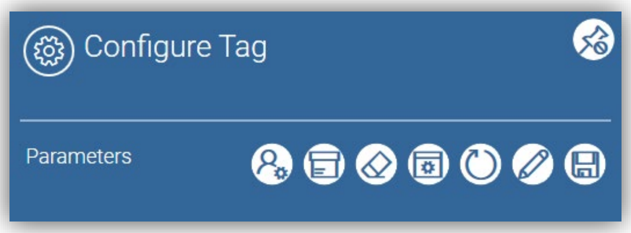
NFC memory information.

Write configuration operation locked.

Display configuration data.

Activate / deactivate tag.

1. **Configuration:** example of read parameters



Configure Tag

Parameters

Read parameters

Write parameters

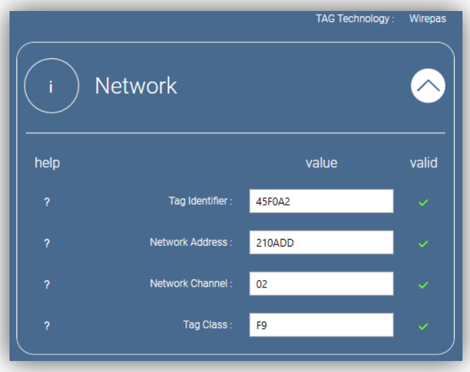
Clear the entire configuration (attention, automatically writes default parameters)

Display Wirepas network fields in hexadecimal or decimal format

Display all configuration parameters, including limits

Factory settings (default parameters)

Save configuration

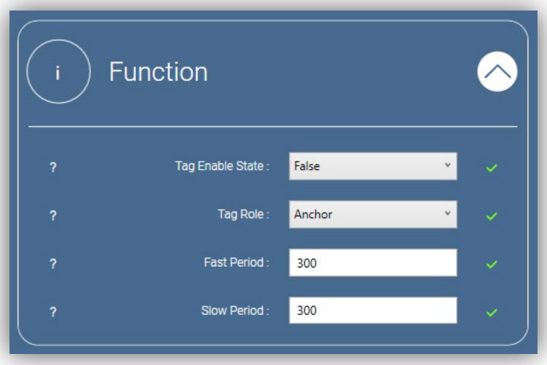


TAG Technology : Wirepas

Network

help		value	valid
?	Tag Identifier :	45F0A2	✓
?	Network Address :	210ADD	✓
?	Network Channel :	02	✓
?	Tag Class :	F9	✓

Parameters	Description	Default values
Tag identifier	Unique Tag identifier (24 bits, modifiable)	Assigned at tag startup
Network Address	Wirepas Network Address (24 bits)	0x210ADD
Network Channel	Wirepas Network Channel (8 bits)	0x02
Tag Class	Tag group management (8 bits) [0xF9-0xFF]	0xF9



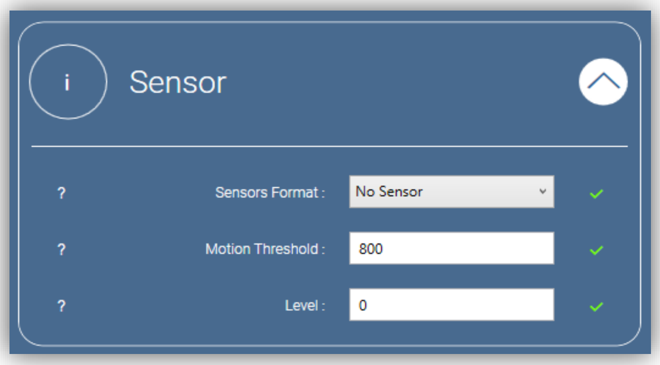
Function

?	Tag Enable State :	False	✓
?	Tag Role :	Anchor	✓
?	Fast Period :	300	✓
?	Slow Period :	300	✓

Parameters	Description	Default values

Tag Enable State	Tag activation and deactivation: True/False	False: Deactivation
Tag Role	Tag role in Wirepas network:[1] Anchor: Anchor with location function Mobile: Long-life mobile beacon with location function Mobile+: Long-life mobile beacon with ID+ location function Mobile HC: Mobile beacon, fast response with location function **Sensor: Automatic management of sensor function role**	Anchor
Fast Period	Period for sending location and/or sensor data **(*Fast used in ID+)**	300
Slow Period	Period for sending location and/or sensor data **(*Slow used in ID+)**	300

*** Fast and Slow are reserved for the location and ID+ function. To send Sensor Data, the transmission period must be the same in both “Period” fields. If these periods are not equal, the beacon will automatically rewrite the same value in Fast and Slow Period by the value entered in Slow Period.**



Sensor

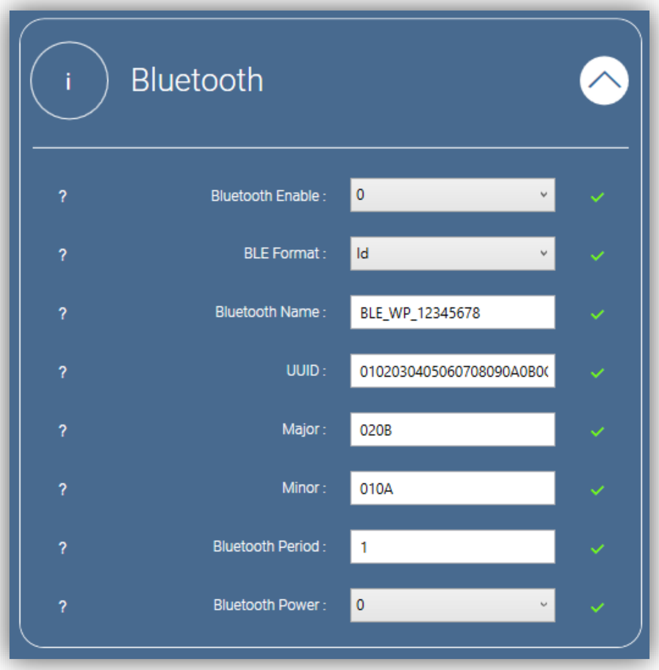
? Sensors Format : No Sensor ✓

? Motion Threshold : 800 ✓

? Level : 0 ✓

Parameters	**Description**	**Default values**
Sensors format	Choice of format associated with sensor reading if tag is equipped: [No Sensor – T – RHT – MAG – MOV – ANG – AI – Digi IN – Digi OUT – AT – PIR – Touch]	**No sensor***
Motion Threshold	Acceleration limit in milliG [32 – 8000] Used by ID+ location function and by MOV sensor function.	1000 milliG
Level	4 level used only for PIR Sensor, Sensibility of presence detection.	0 (Less sensitive)

[1] The Tag Roles available depend on the Firmware version. A complete list of the roles available for each version can be found in the annex of this document.



Bluetooth Enable : 0 ✓

BLE Format : Id ✓

Bluetooth Name : BLE_WP_12345678 ✓

UUID : 0102030405060708090A0B0(✓

Major : 020B ✓

Minor : 010A ✓

Bluetooth Period : 1 ✓

Bluetooth Power : 0 ✓

Note

The Bluetooth advertising function cannot be used in Sensor Role. This function is related to the location application and is therefore not activated for Sensor Role only. This function may be activated only if the tag is not configured in Sensor Role but in Anchor or Mobile Role with a sensor selected.

Parameters	Description	Default values
Bluetooth Enable	0: BLE frame deactivated. 1: BLE frame activated. 2: BLE frame activated when tags are offline (disconnected from mesh network)	0
BLE Format	Id: Advertising with Bluetooth Name. iBeacon: Advertising in Apple format with UUID – Major – Minor.	Id
Bluetooth Name	15 Characters maximum.	BLE_WP_12345678
UUID (iBeacon format)	16 bytes in hexadecimal value	0102030405060708090A0B0C0D0E0F10
Major (iBeacon format)	2 bytes in hexadecimal value	020B
Minor (iBeacon format)	2 bytes in hexadecimal value	010A
Bluetooth period	Advertising period in seconds [1 to 10]	1
Bluetooth power	Power [-8, -4, 0, 4]	0

8. Version

ELA Firmware	WIREPAS Stack	Positioning Application	WIREPAS stack version compatibility	GATEWAY
v0.0.6	v3.4.47	v3.40	Incompatible with 4.x.x versions	v1.1.0
v1.0.0	v4.0.50	v4.0.0	Compatible with all 4.x.x versions	v1.3.0
v1.0.3	v4.0.70	v4.0.0	Compatible with all 4.x.x versions	v1.3.0
v2.0.0	v4.0.70	v4.0.1	Compatible with all 4.x.x versions	v1.3.0
v2.1.0	v4.0.70	v4.0.1	Compatible with all 4.x.x versions	v1.3.0
v2.1.1	v4.0.70	v4.0.1	Compatible with all 4.x.x versions	v1.3.0
v2.2.0	v4.0.70	v4.0.1	Compatible with all 4.x.x versions	v1.3.0
v2.2.1	v4.0.70	v4.0.1	Compatible with all 4.x.x versions	v1.3.0
v2.2.2 (PIR only)	v4.0.70	v4.0.1	Compatible with all 4.x.x versions	v1.3.0

The latest version (at the time of this writing) is v2.2.2.

There is no incompatibility between the gateway version and the firmware version in tags on the network. The software environment for the Wirepas gateway evolves according to customer requests and for general bug fixes and improvements.

Details regarding parameters for other versions are available in the Annex of this document.

6 CONFIGURATION OVER THE NETWORK: APPLICATION CONFIGURATION DATA

Command syntax:

[Class] [Type] [Length] [Value]

➔ to send multiple commands at once the syntax is as follows:

[Class] [Type$_1$] [Length$_1$] [Value$_1$] [Type$_2$] [Length$_2$] [Value$_2$] … [Type$_N$] [Length$_N$] [Value$_N$]

Command field details:

[Class]: This parameter enables you to differentiate and address tag groups.

Sent commands are addressed to one or more Classes. It is important to know the tag class when sending commands.

[Type]: The type identifies the chosen functionality or parameter to modify.

** see complete list on next page*

[Length]: Data size (length).

[Value]: Value to be updated or activation / deactivation data.

** If "Length" is greater than 1, the value must be written with the least significant byte first.*

Information and good practices

- The application configuration data can be up to 80 bytes.
- The class indicated in the following tables is provided as an example. You must enter the class corresponding to your tags when sending commands. Possible values are from 0xF9 to 0xFF.
- The identifier 0x00FA1221 indicated in the following tables is provided as an example. You must enter the identifier for your tag for an individual command addressed to a tag. For a general command addressed to all tags, simply replace the identifier with 0x00000000.
- When [Value] is larger than 1 byte, bytes must be in order with the least significant byte first (little endian).
- A sent command is persistent in the network. Each tag in the network receives this command, and each new tag entering the network also receives the command.
- However, only one command is persistent in the network at a time. As soon as a new command is sent, the previous command is overwritten. A tag that has not yet received the previous command will therefore never receive. It is therefore important to wait a sufficient amount of time, when sending multiple commands in a row, to ensure for message propagation in the entire network. This is especially important in a network with tags in Mobile/Mobile+ mode (NRLS), because these tags only receive commands upon wakeup.
- A broadcast or unicast "ON" command to activate the LED and/or buzzer, either continuous or timed, must be followed by an "OFF" deactivation command of the same type to cancel that same command.

- If a tag is configured for "TAG NRLS" Long-life Mobile Beacon, it will receive the command and apply it upon wakeup. Its maximum standby time is equal to the configured data transmission period.
- No information is returned by tags in the network to know whether the command was properly received. You must calculate and wait the necessary time to be sure that all your tags receive the command.

Configuration commands

Command	Class	Type	Length	Identifier	Data *example	Information
Fast Period	0xF9	0x10	0x08	0x2112FA00	0x20FD0000	[0x01 – 0xFD20] from 1 to 64800 seconds
Slow Period	0xF9	0x12	0x08	0x2112FA00	0x20FD0000	[0x01 – 0xFD20] from 1 to 64800 seconds
Acceleration limit	0xF9	0x14	0x08	0x2112FA00	0x401F0000	[0x20 – 0x1F40] from 32 to 8000 milliG
Activate BLE	0xF9	0x16	0x08	0x2112FA00	0x01000000	[0x00 – 0x02] OFF – ON – Specific ON
BLE period	0xF9	0x18	0x08	0x2112FA00	0x01000000	[0x01 – 0x0A] From 1 to 10 seconds
BLE power	0xF9	0x20	0x08	0x2112FA00	0x01000000	[0x01 – 0x04] -8, -4, 0, +4 dBm
Deactivate tag	0xF9	0x91	0x08	0x2112FA00	0x01000000	Reactivate only via NFC

Diagnostic commands

Command	Class	Type	Length	Identifier	Data *example	Information
Sequence Number	0xF9	0xA0	0x08	0x2112FA00	0x01000000	Returns the value of "Sequence Number"
Hardware version	0xF9	0xA1	0x08	0x2112FA00	0x01000000	Returns hardware version
Firmware version	0xF9	0xA2	0x08	0x2112FA00	0x01000000	Returns firmware version
Battery level	0xF9	0xA3	0x08	0x2112FA00	0x01000000	Returns the battery level

LED and buzzer activation commands

Command	Class	Type	Length	Identifier	Data *example	Information
LED	0xF9	0xE0	0x08	0x2112FA00	0x01000000	LED activation / deactivation
BUZZER	0xF9	0xE1	0x08	0x2112FA00	0x01000000	BUZZER activation / deactivation
LED & BUZZER	0xF9	0xE2	0x08	0x2112FA00	0x01000000	LED & BUZZER activation / deactivation

Data:
1st byte: 01 ➔ Activation – 00 ➔ Deactivation

2nd byte and 3rd byte: 0000 for infinite period.
May be replaced by a value to activate for a few seconds or minutes, for example:

C8 00 ➔ Activation for **200** seconds (0x00C8) (little endian)

10 0E ➔ Activation for **3600** seconds (0x0E10) (little endian)

4th byte: Not used

Commands described here correspond to version v2.0.0 and above. Commands for versions previous are provided in the Annex of this document.

Examples:

In the following some examples are provided to demonstrate the functionality:

- Deactivate LEDs of an entire class (0xF9):

  F9 E0 08 00 00 00 00 00 00 00 00

- Activate the LED of a tag with ID 16388641 (=0xFA1221) in class 0xF9 for 30 s (=0x1E):

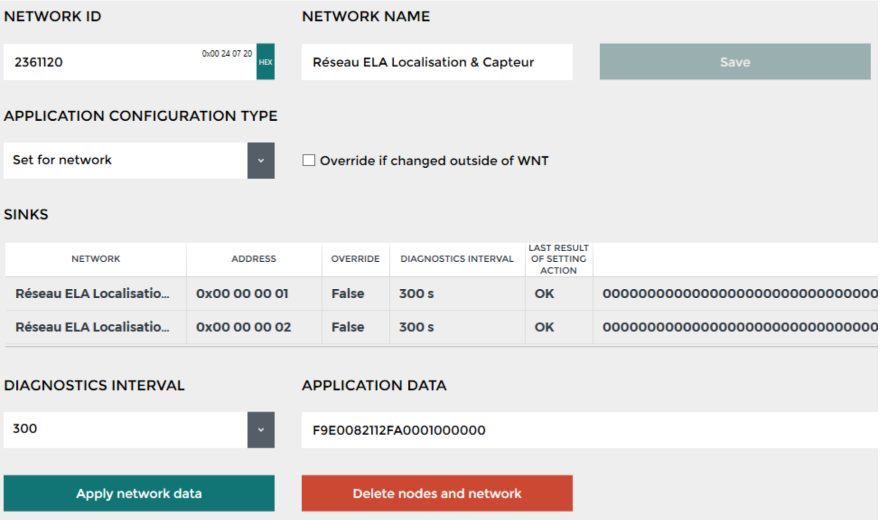
  F9 E0 08 21 12 FA 00 01 1E 00 00

- Multi command: Change Fast Period to 12s (=0x0C) and Slow Period to 3600s (0x0E10) for class 0xFB:

  FB 10 08 00 00 00 00 0C 00 00 00 12 08 00 00 00 00 10 0E 00 00

Sending a command via the Wirepas Network Tool Client

- Open the **Settings** menu and select the **Network** tab.
- Choose your network.Enter the command in the "Application Data" field and then click on "Apply network data".



NETWORK ID

2361120 0x00 24 07 20

NETWORK NAME

Réseau ELA Localisation & Capteur

APPLICATION CONFIGURATION TYPE

Set for network

□ Override if changed outside of WNT

SINKS

NETWORK	ADDRESS	OVERRIDE	DIAGNOSTICS INTERVAL	LAST RESULT OF SETTING ACTION	
Réseau ELA Localisatio...	0x00 00 00 01	False	300 s	OK	0000000000000000000000000000000
Réseau ELA Localisatio...	0x00 00 00 02	False	300 s	OK	0000000000000000000000000000000

DIAGNOSTICS INTERVAL

300

APPLICATION DATA

F9E0082112FA0001000000

The command sent above activates the LED on a tag whose identifier is 0x00FA1221 (little endian), with a Class of 0xF9.

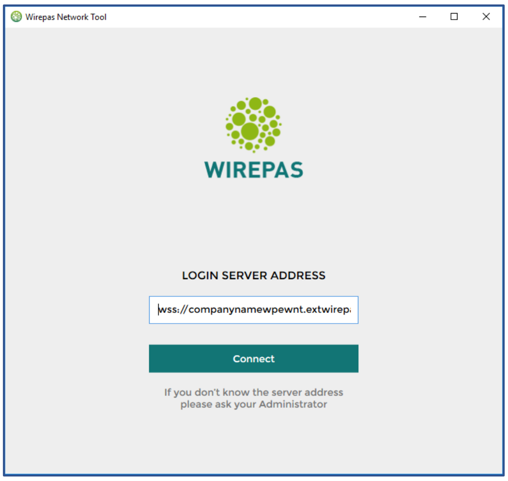
7 VIEWING TOOLS

Wirepas Network Tool (WNT)

- Run the provided **WNT** software installation tool.
- Launch the **WNT** application.

- Fill in the field “**LOGIN SERVER ADDRESS**” with your “login server” provided with your kit. This information corresponds to your “instance”.

  Example: **wss://yourcompanywpewnt.extwirepas.com:8813**



LOGIN SERVER ADDRESS

wss://companynamewpewnt.extwirep:

If you don't know the server address please ask your Administrator

- Enter your provided login name and password and then click on **Log in**.

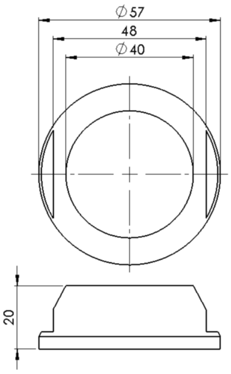
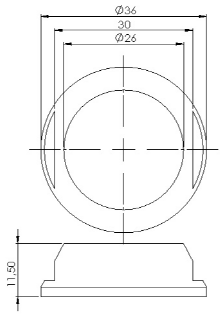
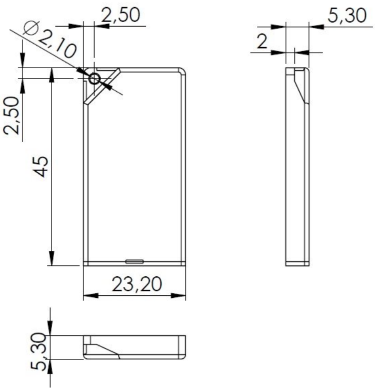
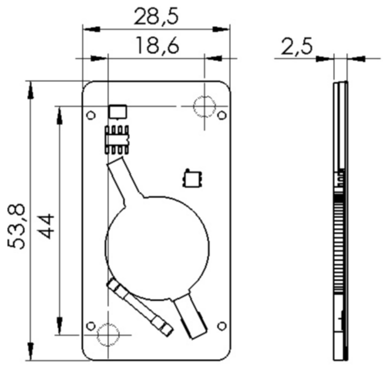
This application enables you to monitor your network(s), view diagnostics data for your objects, but not to view specific ELA Innovation sensor data.

For more information about using the Wirepas Network Tool application, please see the user guide available upon request from.

WP-UG-421 - Wirepas Network Tool - Client User Guide

8 HARDWARE SPECIFICATIONS

PUCK	COIN

 Ø57 48 Ø40 20	 Ø36 30 Ø26 11,50
LITE	**SLIM**
 Ø2,10 2,50 2,50 45 23,20 5,30 2 5,30	 28,5 18,6 2,5 53,8 44

9 SALES REFERENCES

DESCRIPTION	PRODUCT REFERENCE	DESCRIPTION
Blue PUCK **T MESH**	IDF25441	Temperature sensor
Blue PUCK **RHT MESH**	IDF25442	Temperature and humidity sensor
Blue PUCK **MAG MESH**	IDF25443	Magnetic detection sensor
Blue PUCK **MOV MESH**	IDF25444	Movement sensor
Blue PUCK **AI MESH**	IDF25448	Power level sensor
Blue PUCK **DI MESH**	IDF25446	Digital input sensor
Blue PUCK **DO MESH**	IDF25447	Digital output sensor
Blue PUCK **PIR MESH**	IDF25649	Detection presence sensor

10 STANDARDS AND CERTIFICATIONS

- CE logo

- FCC mark

- IC Mark

- RoHS certified

- Wirepas Mesh

11 ANNEX

The Annex provides a complete list of the Tag Roles available for the different Firmware version 2.x.x as well as the settings for BLUE MESH versions, and the LED and buzzer command set for versions prior to v2.0.0.

Tag Roles – v2.x.x

v2.0.0	v2.1.0/ v2.1.1	v2.2.0/ v2.2.1/ v2.2.2
ANCHOR	Anchor	Anchor
TAG NRLS	Mobile	Mobile
-	Mobile+ (new)	Mobile+
TAG AS	Mobile HC	Mobile AS
AUTOROLE	Sensor	Sensor AR
-	-	Sensor R (new)
-	-	Sensor NR (new)

Specifications subject to change without notice. Non-contractual document.

Parameters for version v1.0.3 – Wirepas stack 4.0.70

Parameters	Restrictions	Default values	Description
Tag identifier	6 characters (required): [0-9] [A-F] * hexadecimal value	24-bit identifier Ex: 52A6F9	Tag identifier in MESH network, user modifiable.
Tag activation	True/False	False	Tag activation / deactivation (storage mode)
Mode	NRLS tag: 1 * *Non-Router Long Sleep* Autoscan tag: 2 Autoscan anchor: 3 Opportunistic anchor: 4 * decimal value	3	Tag mode in Wirepas network: - Location function Anchor: 4 Mobile tag, NRLS: 1 - Sensor function Autoscan anchor: 3 - Sensor and Location Anchor: 4 Mobile tag, NRLS: 1
Class	Possible values from 0xF9 to 0xFF * hexadecimal value	FA	Creation of tag groups Example: - PUCK: class 0xFA - COIN: class 0xFB - SLIM: class 0xFC
Node role	0x01: Router 0x02: Non-router 0x41: Router with Autorole mode 0x42: Non-router with Autorole mode * hexadecimal value	41	Tag function in Wirepas network - Location function Anchor: 01 Mobile tag: 02 - Sensor function Autorole: 41
Network channel	2 characters (required): [0-9] [A-F] * hexadecimal value	04	MESH network channel (8 bits)
Network address	6 characters (required): [0-9] [A-F] * hexadecimal value	123ADD	MESH network address (24 bits)

Minimum location period	Period of rapid Location frame emission From 15 seconds to 64800 seconds (18 hrs.)	300 (seconds)	Period of rapid tag position updates, in seconds: automatic change of slow/rapid period possible only with **ID+ MESH** models.
Maximum location period	Period of slow Location frame emission From 15 seconds to 64800 seconds (18 hrs.)	300 (seconds)	Period of slow tag position updates, in seconds: automatic change of slow/rapid period possible only with **ID+ MESH** models.
Threshold	Possible values from 0001 to 07FF	0050	Threshold used by accelerometer when Min Period is different from Max Period. Min Period must always be less than Max Period.
BLE beacon name	BLE beacon tag name	BLUETAGID ELAWP	BLE tag name: 15 characters
BLE BEACON activation	Activation of BLE Beacon mode 0: deactivation, BLE BEACON OFF 1: activation, BLE BEACON ON 2: activation, BLE BEACON ON WHEN OFFLINE	0	Activation / Deactivation of BLE Beacon function 0: deactivation, BLE BEACON OFF 1: activation, BLE BEACON ON 2: activation, BLE BEACON ON WHEN OFFLINE, the BLE Advertising Beacon is activated only when the tag is in OFFLINE mode, that is, when the tag is activated but not connected to its mesh network.
BLE transmission interval	BLE Beacon advertising period	1 (second)	BLE advertising period in seconds, range 1 to 10
BLE power	BLE Beacon tag power:	0	BLE transmission power: [-8, -4, 0, 4]
Format	Tag format	ID	Available formats: ID – T – RHT – MAG – DI – DO - AT
Location activation	True/False	False	Activation / deactivation of Location function Anchors and mobile tags in a Location network must have this parameter set to "True". This parameter must be set to "False" for a sensor network that does not use the Location function.

Parameters for version v1.0.0 – Wirepas stack 4.0.50

Parameters	Restrictions	Default values	Description

Tag identifier	6 characters (required): [0-9] [A-F] * hexadecimal value	24-bit identifier Ex: 52A6F9	Tag identifier in MESH network, user modifiable.
Tag activation	True/False	False	Tag activation / deactivation (storage mode)
Class	Possible values from 0xF9 to 0xFF * hexadecimal value	FA	Creation of tag groups Example: - PUCK: class 0xFA - COIN: class 0xFB - SLIM: class 0xFC
Node role	0x01: Router 0x02: Non-router Not used for location function: *0x41: Router with Autorole mode* *0x42: Non-router with Autorole mode* *0x11: Router with low latency mode* *0x12: Non-router with low latency mode* * hexadecimal value	01	Tag function in Wirepas network - Location function Anchor: ROUTER, value 01 Mobile tag: NON-ROUTER, value 02
Mode	NRLS tag: 1 ** Non-Router Long Sleep* Autoscan tag: 2 Autoscan anchor: 3 Opportunistic anchor: 4 * decimal value	4	Tag mode in Wirepas network: - Location function Anchor: ROUTER, value 4 Mobile tag: NON-ROUTER, value 1
Network channel	2 characters (required): [0-9] [A-F] * hexadecimal value	04	Mesh network channel
Network address	6 characters (required): [0-9] [A-F] * hexadecimal value	001234	Mesh network address
Minimum location period	Emission period of location frame From 15 seconds to 64500 seconds	300 (seconds)	Required: The minimum period must be the same as the maximum period
Maximum location period	Emission period of location frame From 15 seconds to 64500 seconds	300 (seconds)	Required: The minimum period must be the same as the maximum period
Threshold	Possible values from 0001 to 07FF	0050	Not used
BLE beacon name	BLE beacon tag name	BLUETAGID ELAWP	BLE tag name: 15 characters

BLE BEACON activation	Activation of BLE Beacon mode 0: deactivation, BLE BEACON OFF 1: activation, BLE BEACON ON 2: activation, BLE BEACON ON WHEN OFFLINE	0	Activation / Deactivation of BLE Beacon function 0: deactivation, BLE BEACON OFF 1: activation, BLE BEACON ON 2: activation, BLE BEACON ON WHEN OFFLINE, the BLE advertising beacon is activated only when the tag is in OFFLINE mode, that is, when the tag is activated but not connected to its mesh network. Not operational for this version Recommendation: do not activate this functionality
BLE transmission interval	BLE Beacon advertising period	1 (second)	BLE advertising period in seconds, range 1 to 10
BLE power	BLE Beacon tag power:	0	BLE transmission power: [-8, -4, 0, 4]

Parameters for version v0.0.6 – Wirepas stack 3.4.47

Parameters	Restrictions	Default values	Description
Tag identifier	6 characters (required): [0-9] [A-F] * hexadecimal value	24-bit identifier Ex: 52A6F9	Tag identifier in MESH network, user modifiable.
Tag activation	True/False	False	Tag activation / deactivation (storage mode)
Node role	0x01: Router 0x02: Non-router Not used for location function: *0x41: Router with Autorole mode* *0x42: Non-router with Autorole mode* *0x11: Router with low latency mode* *0x12: Non-router with low latency mode* * hexadecimal value	01	Tag function in Wirepas network - Location function Anchor: ROUTER, value 01 Mobile tag: NON-ROUTER, value 02
Flag	0x00: Low energy 0x40: Autorole 0x10: Latency mode * hexadecimal value	40	Tag function in Wirepas network
Network channel	2 characters (required): [0-9] [A-F] * hexadecimal value	04	Mesh network channel
Network address	6 characters (required): [0-9] [A-F] * hexadecimal value	001234	Mesh network address
Location function	True/False	True	Activation / deactivation of location function
Mode	NRLS tag: 1 ** Non-Router Long Sleep* Autoscan tag: 2 Autoscan anchor: 3 Opportunistic anchor: 4 * decimal value	4	Location tag mode in Wirepas network - Location function Anchor: ROUTER, value 4 Mobile tag: NON-ROUTER, value 1
Location period	Emission period of location frame From 15 seconds to 1800 seconds	300 (seconds)	Tag position update period in seconds
BLE BEACON activation	Activation of BLE Beacon mode True/False	False	Activation / Deactivation of BLE Beacon function Not operational for this version. Recommendation: do not activate this functionality
BLE name	BLE beacon tag name	BLUE TAG ID WPE	BLE tag name: 15 characters
BLE power	BLE Beacon tag power	0	BLE transmission power: [-8, -4, 0, 4]
Transmission interval	BLE Beacon advertising period	1000 (milliseconds)	BLE advertising period in milliseconds, range 100 to 10000

Commands for Data Configuration Application version [v1.0.0 – v1.0.3]

Configuration commands

** Not taken into account for NFC tag configuration and global commands addressed to all tags*

Command	Class	Type	Length	Data	Example
Scan period	0xF9	0x01	0x02	0x0258	0258 → 5802 = 600 seconds
Mode	0xF9	0x02	0x01	0x01	NRLS mode: 01
Class	0xF9	0x0A	0x01	0xFB	Class change from F9 to FB

Global LED and buzzer activation commands (a.k.a. "Broadcast")

Command	Class	Type	Length	Data	Information
LED	0xF9	0xE0	0x04	0x01000000	The two bytes shown in red enable you to enter a time in seconds for LED and/or buzzer activation. 300 seconds: 0x012C0100 **Little endian*
BUZZER	0xF9	0xE1	0x04	0x01000000	
LED & BUZZER	0xF9	0xE2	0x04	0x01000000	

Individual LED and buzzer activation commands (a.k.a. "Unicast")

ID example → 0x1215FA

Command	Class	Type	Length	Data	Information
LED	0xF9	0xE3	0x08	0x01FA151200000000	The two bytes shown in red enable you to enter a time in seconds for LED/ buzzer activation. 300 seconds: 0x012C0100 **Little endian*
BUZZER	0xF9	0xE4	0x08	0x01FA151200000000	
LED & BUZZER	0xF9	0xE5	0x08	0x01FA151200000000	

Specifications subject to change without notice. Non-contractual document.