kerlink
communication is everything

Wanesy™ Wave

Interfaces

PTA
KERLINK

Wanesy™ Wave
Interfaces

	Redaction	Approbation	Validation
Trigram	FRO	DLR	MBR
Date	2022-06-17	2022-06-24	2022-06-24
Signature			

Version	Edits
V0.1	2020-07-24 Creation from product description
V0.2	2020-12-09 Update after internal review
V0.3	2020-01-12 Corrections and reorder of chapters
V0.4	2021-02-08 Update to match firmware v1.8.0 features
V1.0	2021-02-08 Validation to release document version V1.0
V1.1	2021-09-29 BLE Asset Monitoring Add-Up
V1.2	2022-02-14 Dwell Time Applicative Configuration
V1.3	2022-05-31 Add WiFi Sniffer Mode

TABLE OF CONTENT

Reference

Reference	Document / link	Description
[1]	Wanesy™ Wave Product Description	Description of Wanesy™ Wave product

Glossary

Abbreviation	Description
BLE	Bluetooth Low Energy
CBOR	Concise Binary Object Representation
IoT	Internet of Things
LPIoT	Low Power Internet of Things
LoRaWAN	Long Range Wide-Area Network
Wi-Fi	Wireless network protocols, based on the IEEE 802.11 family of standards

1 Introduction

The Wanesy™ Wave is a product based on Kerlink Low Power IoT Reference Design solution. It consists of a Wi-Fi and BLE device with a LoRaWAN™ backhaul. The device notably provides the following functionalities:

- Communicate in LoRaWAN™
- Several Wi-Fi services
- BLE services

The main purpose of the device is tracking, using Wi-Fi probe requests collection from Smartphones when doing people counting or Bluetooth Low Energy scanning and zoning when doing asset tracking.

This document describes the interfaces provided by the Wanesy™ Wave product, refer to **[1]** for a full description of the product.

2 LoRaWAN interface

2.1 LoRaWAN backhaul configuration

The LoRaWAN backhaul configuration is hardcoded in the firmware. Some LoRaWAN parameters can be modified using LoRaWAN Mac messages. More information is available in the LoRaWAN specification. LoRaWAN ADR mode is activated.

Uplink LoRaWAN messages periodicity can be configured in the application parameters, see chapter §3.

2.2 Uplink message frame structure for BLE scan

Uplink LoRaWAN messages concerning BLE scan are sent on FPort:**85.**
Uplink BLE scan LoRaWAN message content is detailed in the CBOR-based protocol**Erreur ! Source du renvoi introuvable.**, see chapter §5.

2.3 Uplink message frame structure for BLE asset monitoring

Uplink LoRaWAN messages concerning BLE asset monitoring are sent on FPort:**88**.
Uplink BLE asset monitoring LoRaWAN message content is detailed in the CBOR-based protocol, see chapter §5.

2.4 Uplink message frame structure for Wi-Fi COUNT

Uplink LoRaWAN messages concerning Wi-Fi COUNT are sent on FPort:**80**.
Uplink WI-FI COUNT LoRaWAN message content is detailed in the COUNT Protocol, see chapter §5.

2.5 Uplink message frame structure for Wi-Fi SNIFFING

Uplink LoRaWAN messages concerning Wi-Fi SNIFFING are sent on FPort:**75**.
Uplink WI-FI SNIFFING LoRaWAN message content is detailed in the SNIFFING Protocol, see chapter §5.

2.6 Behavior in case of LoRaWAN emission not acknowledged

If a LoRaWAN frame is not acknowledged by the LoRa Network Server, the frame upload will be retried until the maximum number of retries is reached, or sooner if the frame is acknowledged.

2.7 Clock synchronization

Uplink LoRaWAN messages concerning Wanesy Wave clock synchronization with the LoRa Network Server are sent on FPort:**202**.

This message is described in LoRaWAN Application Layer Clock Synchronization specification in the following link:
https://lora-alliance.org/resource_hub/lorawan-application-layer-clock-synchronization-specification-v1-0-0

2.8 Uplink status message

Uplink LoRaWAN messages concerning Wanesy Wave status are sent on FPort:**32.**
Status message provides power supply connection information and LoRaWAN acknowledge message failures.

Data of uplink status message have the following structure:

Field	Type	Description
USB_connection_state	uint8	State of power supply connection with: • USB_STATE_DISCONNECTED = 0 • USB_STATE_CONNECTED = 1 • USB_STATE_DISCONNECTION = 2 • USB_STATE_CONNECTION = 3
uptime	uint32	Date of status message creation
event_time	uint32	Date of power supply connection change: • Disconnection date if conn_state equals to USB_STATE_DISCONNECTION • Connection date if conn_state equals to USB_STATE_CONNECTION • 0 for other conn_state values
ack_failure	uint16	Number of ACK message KO in case of LoRaWAN message ACK, else 0.

2.9 Dwell Time applicative configuration

The device’s LoRaWan dwell time feature can be deactivated through an applicative command sent on FPort:**68**. This setting will remain effective until a reactivation command is issued (ie. It is persistent to a device reboot and becomes part of the device configuration file).
This feature is only effective for AS923 devices.

3 Configuration with LoRaWAN

3.1 ESP32 application configuration

The ESP32 is the embedded slave chipset with the Wi-Fi and BLE capabilities.
The STM32 is the master chipset with the control of the LoRaWAN communication (refer to **[1]** for a full description of the product).
The STM32 will automatically forward the AT commands to the ESP32.

Wanesy™ Wave remote configuration is done using LoRaWAN downlink messages. The strategy used for the Wanesy Wave product is the transmission of AT commands using the format defined in AT protocol chapter, refer to §4.

Answers received from the ESP32 are forwarded to the server using LoRaWAN uplink messages.

The LoRaWAN port used for the uplink/downlink messages for AT command is FPort:**65**
The response of the ESP32 (data + OK/ERROR) is encapsulated into the next LoRaWAN Uplink frame on the same FPort:**65**.
If an error occurs (AT timeout), the FPort:**66** is used.

The AT commands listed below are detailed in AT protocol chapter, refer to §4.

3.1.1 BLE scan application

3.1.1.1 BLE scan major list

The "BLE scan major list" AT command is used to program the major list of BLE iBeacons that will be scanned. If iBeacon major is not part of the list, it will not be selected.

Payload text: (AT command)
AT+BLE/MLIST=XX,YYY,… with XX and YYY are major values in decimal (from 0 to 65534)

When AT command is received by ESP32, only BLE iBeacons with major value which equals to one of listed major values will be detected during scan.
Major value 0xFFFF is forbidden in the system, only values from 0 to 0xFFFE are valid.
A maximum of 10 major parameters can be entered in the list.

Each AT command "BLE/MLIST" replaces the previous list by the new one.
If no XX and YYY parameters are filled in (AT+BLE/MLIST=), the list is removed and BLE scan accepts all iBeacons.

3.1.1.2 BLE scan confirm message delay

The "BLE scan confirm message delay" AT command is used to configure the amount of time a iBeacon must be detected by the Wanesy Wave to be considered in the "confirmed" state.

Payload text: (AT command)
AT+BLE/CNF=XX with XX is time in second

When AT command is received by ESP32, BLE iBeacon presence is confirmed by a message if it is detected for XX seconds.
If XX is not present or equal to 0, the parameter comes back to default value: 300s (5 minutes).

3.1.1.3 BLE scan RSSI filter threshold

The "BLE scan RSSI filter threshold" AT command is used to fix RSSI threshold to filter BLE iBeacons during scan.

Payload text: (AT command)
AT+BLE/RSSI=XX,YY with
XX is RSSI level threshold (between -1 and -94)
XX is RSSI level hysteresis to exit BLE scan (between 0 and -20)

When AT command is received by ESP32, only BLE tags with RSSI level higher than XX are collected.
The already collected BLE tags are removed if RSSI level is lower than XX+YY.
If XX is equal to 0, the threshold is removed and no RSSI filter applies.

3.1.1.4 BLE scan identifier configuration

iBeacon identifier filtering can be processed on one of the two identifier parameters.

The "BLE scan identifier configuration" AT command is used to select the BLE tag identifier in the scan report message.

BLE tag identification can be done on:
- BLE tag MAC address.
- BLE tag MAJOR+MINOR.

Payload text: (AT command)
AT+BLE/CFG=X with X is the identifier configuration.

Configuration mode:
- If identifier configuration = 0, MAC address is used in the scan report.
- If identifier configuration = 1, MAJOR+MINOR is used in the scan report.

3.1.1.5 UID filtering

The "BLE tag UUID filtering" AT command is used to allow only BLE tags with matching UUID listed in the UUID list.

Payload text: (AT command)
AT+BLE/UUID=X,Y,Z,… with X,Y and Z as vendor specific UUID on 16 bytes, each UUID is separated by coma.

Up to five UUID can be managed in the list.
Using the AT command without parameter deletes the UUID list, so no more UUID filter is applied.

3.1.2 Wi-Fi Probe Request Count/Sniffing application

3.1.2.1 Wi-Fi device RSSI threshold

The "Wi-Fi device RSSI threshold" AT command is used to limit Wi-Fi COUNT/SNIFFING to devices with RSSI value higher than defined threshold.

Payload text: (AT command)
AT+WPR/TH=XX with XX is RSSI threshold.

Only Wi-Fi probe requests with RSSI higher than XX are accounted for.
If XX is equal to 0, the threshold is removed and no RSSI filter applies.

3.1.3 BLE Asset Monitoring application

The "BLE monitor" AT command is used to configure the entirety of the BLE asset monitoring feature.

Payload text: (AT command)

AT+BLE/MON=AAAAAAAAAAAA/RR,MMMM,TT,BBBB with
- AAAAAAAAAAAA is the base mac address to apply the feature on
- RR is the address range to expand the base address on (CIDR like addressing)
- MMMM is the BLE manufacturer ID to apply the feature on
- TT is the minimum TX interval to respect between uplinks
- BBBB is a bitmask indicating which significant bytes must be sent

3.2 STM32 application configuration and basic Device Management

The LoRaWAN port used for the downlink message to the STM32 application configuration is FPort:**68**.
Responses to the commands are uplink frames sent on FPort:68.

Raw data command is used for STM32 rather than AT command type used to communicate with the ESP32.

3.2.1 Payload description:

STM32 configuration payload sent in downlink configuration frames is composed as following:

- The first byte is the command.
- Following bytes (optional) are the data of the command.

STM32 configuration payload must be sent in hexadecimal (binary) mode.

3.2.2 List of commands

3.2.2.1 Test command

0x00: test command - no effect (test response only)

Command response: TEST OK

3.2.2.2 Reboot command

0x01: reboot command – with the data [0x62, 0x6F, 0x6F, 0x74], the end-device will reboot after 10 seconds. [0x62, 0x6F, 0x6F, 0x74] is the ascii value for the characters "boot".

Command response:

- REBOOT OK: success
- REBOOT NONE: failure

3.2.2.3 Format File System command

0x02: format File System command – with the data [0x66, 0x6F, 0x72, 0x6d, 0x61, 0x74], the end-device will format the file system. [0x66, 0x6F, 0x72, 0x6d, 0x61, 0x74] is the ASCII values for the characters "format".

Command response:

- FORMAT OK: success
- FORMAT NONE: failure

3.2.2.4 Wi-Fi count/sniffing application reconfiguration

0x10: Wi-Fi count/sniffing reconfiguration, the payload respects the following structure:

Field	Type	Description	Accepted Values
period_sec_cfg	uint32	Periodicity of Wi-Fi counts in seconds / Sniffing redundancy period	from 60 to 86400
counter_max	uint32	Maximum number of Wi-Fi counts before sending a message	from 100 to 5000
clock_sync_period	uint32	Periodicity of clock sync messages in seconds	from 120 to 86400

All fields are in little-Endian.

Submode configuration:
The 2 most significant bits of the period_sec_cfg field are used to select the WiFi submode :
bit 31: ‘0’ = Count Mode, ‘1’ = Sniffing Mode
bit 30: ‘0’ = Apply hashing function onto MAC addresses, ‘1’ = Send original MAC addresses (used in sniffing mode only)

<u>Examples:</u>

#1: WiFi Count
Downlink payload: 102C010000E803000008070000

Detailed frame:

Command	period_sec_cfg	counter_max	clock_sync_period
10	2C010000	E8030000	08070000
0x10: Wi-Fi mode	h(0x0000012C)= d(300)	h(000003E8)=d(1000)	h(00000708)=d(1800)

- period_sec_cfg = Count mode (bit31=0), 300 seconds (5 min)
- counter_max = 1000 count maximum
- clock_sync_period = 1800 seconds (30 min)

Default configuration of Wi-Fi Probe Request Count is:

- period_sec_cfg = 0 (disabled)
- Counter of maximum Wi-Fi target to reach to send a LoRaWAN frame: counter_max = 2000
- Clock synchronization period clock_sync_period = 60min (3600s)

Command response:

- WIFI PARAM OK: success
- WIFI PARAM KO: failure

#2: WiFi Sniffing with hashed addresses
Downlink payload: 102C010080

Detailed frame:

Command	period_sec_cfg
10	2C010080
0x10: Wi-Fi mode	h(0x80**00012C**)= d(**300**)

- period_sec_cfg = Sniffing mode (bit31=1) with addresses hashing (bit30=0), 300 seconds (5 min) of redundancy period

#3: WiFi Sniffing with original addresses
Downlink payload: 102C0100C0

Detailed frame:

Command	period_sec_cfg
10	2C0100C0
0x10: Wi-Fi mode	h(0xC0**00012C**)= d(**300**)

- period_sec_cfg = Sniffing mode (bit31=1) without addresses hashing (bit30=1), 300 seconds (5 min) of redundancy period

3.2.2.5 BLE scan application reconfiguration

0x11: BLE scan application reconfiguration, the payload respects following structure:

Field	Type	Description	Accepted Values
scan_periodicity	uint32	Periodicity of BLE scans in seconds	from 15 to 7200
scan_duration	uint32	Duration of BLE scans in seconds	from 10 to 7199
Frame_periodicity	uint32	Periodicity of LoRaWAN frame upload	from 5 to 300

All fields are in little-Endian.

Example:
Downlink payload: 11780000000C00000028000000

Detailed frame:

Command	scan_periodicity	scan_duration	frame period
11	78000000	0C000000	28000000
0x11: BLE scan	h(00000078)= d(120)	h(0000000C)=d(12)	h(00000028)=d(40)

- scan_periodicity = 120 secondes (2mn)
- scan_duration = 12 secondes
- frame period = 40 seconds

Default configuration of BLE Scan Count is:

- scan_periodicity = 40 seconds
- scan_duration = 30 seconds
- frame period= 40 seconds

Command response:

- BLE PARAM OK: success
- BLE PARAM KO: failure

3.2.2.6 Dwell Time applicative configuration

0x20: Dwell Time applicative configuration, the payload respects the following structure:

Field	Type	Description
dwell_time_activated	uint8	State of the dwell time feature

Example:
Downlink payload: 2000

Detailed frame:

Command	dwell_time_activated
20	00

- dwell_time_activated = 0: the dwell time feature will be deactivated (if allowed by current DataRate and LoRaWan region)

Default configuration of Dwell Time feature is: 01 (enabled).

Command response:

- DT CFG OK: success
- DT CFG KO: failure

3.2.2.7 Other command values

Other values: reserved for future use.

4 AT protocol

4.1 Protocol description

The protocol is inspired by AT commands: it is fully described on the following section.

On the Wanesy Wave, the LoRaWAN port used for the uplink/downlink messages for AT command is FPort:**65**
The response (data + OK/ERROR) is encapsulated into the next LoRaWAN Uplink frame on the same FPort:65.
If an error occurs (AT timeout), the port used will be on the FPort:**66.**

4.1.1 AT commands syntax

4.1.1.1 Abbreviations

Symbol	Meaning
<CR>	Carriage return character
<LF>	Line Feed character
<...>	Name enclosed in angle brackets is a syntactical element. Brackets themselves do not appear in the command line.
[...]	Optional subparameter of a command. Brackets themselves do not appear in the command line.

4.1.1.2 Simple command

Simple command syntax:

- AT+CMD0<CR>

Simple answer syntax:

- <CR><LF>OK<CR><LF>
- <CR><LF>ERROR<CR><LF>

4.1.1.3 Read command

Read command for checking current subparameter values syntax:

- AT+CMD1?<CR>

Associated response syntax:

- <CR><LF>+CMD1: <subparameter1>,<subparameter2>,… <CR><LF>

4.1.1.4 Command with subparameter

Command with subparameter syntax:

- AT+CMD2=<subparameter1>,<subparameter2><CR>

Technical table

Parameter	Format	Description
State	ASCII	State of the feature: • ON: enable the feature • OFF: disable the feature
RAW data	HEXASCII	Raw data
CBOR data	HEXASCII	Data, CBOR formatted
URL	ASCII	Relative URL (can be "404" for undefined URLs)
Web Content	ASCII	Web page source content
RSSI Threshold	Signed Integer	RSSI threshold: under this threshold, collected messages are discarded Use value 0 to disable this filter Value from -94 to -1
MAC	HEXASCII 48bit	MAC Address
MaxFrameSize	Integer	Maximum frame size allowed in data responses
SSID	ASCII	SSID Wi-Fi Access Point
Key	HEXASCII	Password Wi-Fi Access Point (can be void)
Channel	Integer	Channel Wi-Fi Access Point (can be void for 'auto')
BLE Period	Integer	BLE period that schedules each beacon transmission
BLE Content	HEXASCII	BLE beacon raw content that is sent on each transmission
Data mode	Integer	Set the data mode that will be used (see AT+DATA for details)
Time	Integer	Current time in seconds, aligned on the GPS epoch

4.1.2 Basic AT commands

Command	**Meaning**
AT	Basic AT test command
+V	Get software and hardware versions
+F	Reset Factory Modem + Reboot

4.1.3 Wi-Fi Access Point AT commands

Command	**Meaning**
+WAP/EN	Manage the Wi-Fi Access Point feature

4.1.4 Wi-Fi Probe Request Collector AT commands

Command	**Meaning**
+WPR/EN	Manage the Wi-Fi Probe Request Collector feature
+WPR/TH	Manage the Wi-Fi Probe Request Collector RSSI Threshold

4.1.5 Bluetooth AT commands

Command	**Meaning**
+BLE/MLIST	Manage BLE Scan “major” list
+BLE/CNF	Delay of confirmation presence of iBeacon
+BLE/RSSI	RSSI Level Threshold and Hysteresis
+BLE/CFG	Select BLE MAJOR+MINOR or BLE MAC address for report
+BLE/UUID	Manage BLE scan UUID list
+BLE/MON	Configure BLE Asset Monitoring

4.2 Details of AT commands

4.2.1 Basic AT commands details

4.2.1.1 AT

AT	**Basic AT command**
Command	AT
Response	OK

4.2.1.2 +V

+V	**Get versions command**
Command	AT+V?
Response	+V: <software version>,<hardware version> OK / ERROR

4.2.1.3 +F

+F	**Reset Factory Modem + Reboot**
Command	AT+F
Response	OK/ERROR

4.2.1.4 +WAP/EN

+WAP/EN	**Activate or de-activate the WiFi Access Point**
Command	AT+WAP/EN=<MODE>
Parameter	**mode**: ON: activates WAP OFF: de-activates WAP
Response	OK/ERROR

4.2.2 Wi-Fi Probe Request Collector AT commands details

4.2.2.1 +WPR/EN

+WPR/EN	**Activate or de-activate the Wi-Fi Probe Request Collector**
Command	AT+WPR/EN=<MODE>
Parameter	**mode**: ON: activates PRC OFF: de-activates PRC HASH : activates addresses hashing function RAW : de-activates addresses hashing function
Response	OK/ERROR

4.2.2.2 +WPR/TH

+WPR/TH	**Manage the Wi-Fi Probe Request Collector RSSI Threshold filter**
Command	AT+WPR/TH=<threshold>
Parameter	**threshold**: Wi-Fi RSSI level detection threshold. Default: 0 (deactivated) Min.: -1 Max.: -94
Response	OK/ERROR

4.2.3 Bluetooth AT commands details

4.2.3.1 +BLE/MLIST

+BLE/MLIST	**Manage BLE Scan "Major" list**
Command	AT+BLE/MLIST=<MAJOR1>,<MAJOR2>,<MAJORn>
Parameter	**MAJORn:** list of major parameters of allowed iBeacon (in decimal). Each AT command BLE/MLIST removes the previous list. List: Min.: Empty list Default: Empty list Max.: 10 majors in the list MAJORn: Min.: 0 Max.: 65534 If no major in parameter ("AT+BLE/MLIST="), major list is removed.
Response	OK/ERROR

4.2.3.2 +BLE/CNF

+BLE/CNF	**Delay of confirmation presence of iBeacon**
Command	AT+BLE/CNF=<delay>
Parameter	**delay:** Delay of confirmation presence in seconds of iBeacon. If delay = 0, delay is reset to default value (300 seconds). Min.: 15 (and 0 as an extra value) Default: 300 Max.: 172800
Response	OK/ERROR

4.2.3.3 +BLE/RSSI

+BLE/RSSI	**RSSI Level Threshold and Hysteresis**
Command	AT+BLE/RSSI=<threshold>,<hysteresis>
Parameter	**threshold:** BLE RSSI level threshold. Default: 0 (deactivated) Min.: -1 Max.: -94 **hysteresis:** RSSI level hysteresis to exit BLE scan. Default: 0

	Min.: 0
	Max.: -20
	If threshold = 0 (“AT+BLE/RSSI=0”), RSSI filter is removed.
Response	OK/ERROR

4.2.3.4 +BLE/CFG

+BLE/CFG	**Select BLE MAJOR+MINOR or BLE MAC address report mode**
Command	AT+BLE/CFG=<identifier configuration mode>
Parameter	**Identifier configuration mode:** If configuration mode = 0, MAC address is used in the scan report. If configuration mode = 1, MAJOR+MINOR is used in the scan report.
Response	OK/ERROR

4.2.3.5 +BLE/UUID

+BLE/UUID	**Manage BLE Scan UUID list**
Command	AT+BLE/UUID=<UUID1>,<UUID2>,<UUIDn>,…
Parameter	**UUIDn:** list of UUID parameter of allowed iBeacon. Min.: 0 Max.: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF Each AT command BLE/UUID removes the previous list. Max. item in the list: 5 UUID maximum in list. If No UUID in parameter (“AT+BLE/UUID=”), UUID list is removed.
Response	OK/ERROR

4.2.4 BLE Asset Monitoring AT commands details

+BLE/MON	**Configure BLE Asset Monitoring**
Command	AT+BLE/MON=<MON1>/<MON2>,<MON3>,<MON4>,<MON5>
Parameter	**MON1:** base MAC address (6 bytes in hexadecimal). **MON2:** MAC address filter size (1 byte in decimal). **MON3**: Manufacturer ID (2 bytes in hexadecimal) **MON4**: TxInterval in minutes (up to 300, in decimal) **MON5**: Significant bytes bitmap (1 to 4 bytes, in hexadecimal)
Response	OK/ERROR

5 Application data exploitation

BLE scan application uses a CBOR-based protocol.
Wi-Fi Probe request Count mode uses a custom protocol.
Wi-Fi Probe request Sniffing mode also uses a CBOR-base protocol.

5.1 BLE Scan CBOR-based Protocol

5.1.1 Presentation

Wanesy Wave BLE scan application uses a CBOR-based protocol.

CBOR is a binary data serialization, close to a JSON representation. The transmission of data object contains key-value pairs by definition. The sender and the receiver must agree on the interpretation of these key-value pairs, but it will not break the compatibility if other pairs are added later (read IETF RFC7049 for more details about CBOR).

The CBOR data model used in the Wanesy Wave is explained below in order to be decoded by the application servers interfaced with the LoRa Network Server.

ESP32 sends BLE scan data to the STM32. The payload is transmitted in hexascii mode with a CBOR data representation. This hexascii payload is forwarded to the LoRa Network Server via LoRAWAN message.

5.1.2 BLE scan use-case

Main feature of the BLE Scan application is to collect periodically BLE iBeacon identifier and RSSI within a close range. iBeacons information collected by the Wanesy Wave BLE scanner will be then forwarded to a LoRaWAN server.

To be usable by the server, a caught BLE iBeacon must contain:

- One Identification parameter among the two following:
  - MAJOR+MINOR iBeacon parameter (4 bytes data – 2 MSB for Major and 2 LSB for minor)
  - MAC address parameter (16 bytes)
- Bluetooth RSSI level

5.1.3 Data items description

- New tag:
  - Key “n” is used to store an array of new BLE identifier since previous scan. Each data is composed of: MAJOR+MINOR iBeacon parameter + RSSI.
  - Key “m” is used to store an array of new BLE identifier since previous scan. Each data is composed of: MAC Address iBeacon parameter + RSSI.
  - When the list of new tags is composed of several elements, optimization is processed on the identifiers value, except on the first which stays unoptimized.
- Confirmed tag:

-
    - Key "c" is used to store an array of confirmed BLE identifier after a delay of presence. Each data is composed of: MAJOR+MINOR iBeacon parameter + RSSI.
    - Key "b" is used to store an array of confirmed BLE identifier after a delay of presence. Each data is composed of: MAC Address iBeacon parameter + RSSI.
    - When the list of confirmed tags is composed of several elements, optimization is processed on the identifiers value, except for the first identifier which stays unoptimised.
- Removed tag:
    - Key "r" is used to store an array of not detected (removed) BLE identify from previous scan (MAJOR+MINOR identifier used).
    - Key "q" is used to store an array of not detected (removed) BLE identify from previous scan (MAC Address identifier used).

In order to be more efficient, each data field is optimized, and described below:
For "n" and "c" values (or "m" and "b" values when MAC Address identifier is used):

- Array of 3 parameters:
    - MAJOR+MINOR iBeacon value (or MAC Address value),
    - RSSI
    - Moving Status.

For "r" value (or "q" value when MAC Address identifier is used):

- Tag identifier value with no array (MAJOR+MINOR iBeacon value/ MAC Address value only).

5.1.3.1 MAJOR+MINOR identifier

Entry in the table is a 4 bytes word with MAJOR in MSB (2 Bytes) + MINOR in LSB (2 bytes).

Each couple MAJOR+MINOR parameter is filtered by ascending value.

An optimization is done for the second and the following iBeacon MAJOR+MINOR for one specific format ('c', 'n' and 'r') with only the delta.

BLE tag major or minor value equals to 0xFFFF (65535) are not supported. 0xFFFF value is forbidden.

Example:

- 0x08020245 => Major is 0x802 (2050 in decimal) and Minor is 0x245 (581 in decimal)
- If a second element in the same format is 0x08030102 (Major 0x803 and Minor 0x102) the value of MAJOR+MINOR parameter will be 0xFEBD (0x08030102 - 0x08020245 = 0x0000FEBD)

5.1.3.2 MAC address identifier

BLE tag identifier report can be done using the MAC address value instead of MAJOR+MINOR value. Keys of the BLE zoning fields are switched to ‘b’, ‘m’ and ‘q’ instead of ‘c’, ‘n’ and ‘r’.

MAC address identifier is casted into a 6 bytes hexadecimal word.

An optimization is done for the second and the following iBeacon MAC address of the list by the substraction of the current unoptimized MAC address binary word representation by the previous unoptimized MAC address binary word representation.

Calculation : <current MAC> - <previous MAC> = <encoded MAC>

Example:

- The list is composed of the following sorted MAC address :
  40:D5:77:CB:CF:D9
  40:D5:77:CB:CF:**E4**
  40:D5:77:CB:**D5:F6**

- The MAC address 40:D5:77:CB:CF:D9 is represented in the hexadecimal word of 6 bytes 40D577CBCFD9.

- The second MAC address element is 40:D5:77:CB:CF:E4, represented in the hexadecimal word of 6 bytes 40D577CBCFE4, the optimized identifier value will be 061F5A0C084E.

  Calculation in hexadecimal: 0x40D577CBCFE4 – 0x40D577CBCFD9 = 0xB

- The third MAC address element is 40:D5:77:CB:D5:F6 represented in the hexadecimal word of 6 bytes 606A8E6379E0, the optimized identifier value will be 40D577CBD5F6.

  Calculation in hexadecimal: 0x40D577CBD5F6 – 0x40D577CBCFE4 = 0x612

- Resulting optimized identifier list is the following:
  40D577CBCFD9
  B
  612

5.1.3.3 RSSI optimization

All RSSI measurements have a negative value included in the range [-1;-94]. In order to be more efficient on the CBOR encoding, the following algorithm is applied on the RSSI value:

```
(RSSI + 47) / 2 = encoded RSSI
```

5.1.3.4 Moving/Status

The Moving/Status value provides the information about the iBeacon motion (and reserved bits). The value 0 means the iBeacon is not moving and the value 1 means the iBeacon is moving.

E9 Dear Beacon BLE tag by MINEW Company is the one selected to detect motion by accelerometer.

If the motion state “Moving/Status” toggles, the BLE tag Major+Minor information is sent in a extra dedicated frame under the key confirm “c”. The extra frame is sent at the end of the scan period.

5.1.3.5 Example of BLE scan

Steps starting from scanned values until optimized and encoded CBOR message is detailed below:

<u>BLE Scan (n-1): Previous tab</u>

Timestamp	BLE MAJOR	BLE MINOR	RSSI	Move
10	2048 (0x800)	3056 (0xBF0)	-52	0
10	2048 (0x800)	3057 (0xBF1)	-64	0
10	2048 (0x800)	18069 (0x4695)	-88	0
10	2049 (0x801)	45 (0x2D)	-41	0

<u>New BLE Scan (n): Tab</u>

Timestamp	BLE MAJOR	BLE MINOR	RSSI	Move
60	2047 (0x7FF)	3 (0x3)	-56	1
60	2048 (0x800)	3056 (0xBF0)	-50	0
60	2048 (0x800)	3057 (0xBF1)	-80	0
60	2048 (0x800)	25024 (0x61DE)	-40	0

The new tab (n) with complete STATUS of each iBeacon:

Timestamp	BLE MAJOR	BLE MINOR	RSSI	Move	Status
60	2047 (0x7FF)	3 (0x3)	-56	1	New
60	2048 (0x800)	3056 (0xBF0)	-50	0	Confirm
60	2048 (0x800)	3057 (0xBF1)	-80	0	Confirm
10	2048 (0x800)	18069 (0x4695)	-88	0	Remove
60	2048 (0x800)	25024 (0x61DE)	-40	0	New
10	2049 (0x801)	45 (0x2D)	-41	0	Remove

The new tab (n) with RSSI optimization:

Timestamp	BLE MAJOR	BLE MINOR	RSSI	Move	Status
60	2047 (0x7FF)	3 (0x3)	-4	1	New
60	2048 (0x800)	3056 (0xBF0)	-1	0	Confirm
60	2048 (0x800)	3057 (0xBF1)	-16	0	Confirm
10	2048 (0x800)	18069 (0x4695)	Not used	Not used	Remove
60	2048 (0x800)	25024 (0x61DE)	+3	0	New
10	2049 (0x801)	45 (0x2D)	Not used	Not used	Remove

The same dataset is split in 3 specifics keys (”new”, ”confirm”, ”remove”):

Optimization is processed on the BLE tag identifier.

```
{"n": [[0x7FF0003, -4,1], [0x161DB, 3,0]],

"r": [0x8004695, 0xB998],

"c": [[0x8000BF0, -1,0], [0x1, -16,0]]}
```

The same with decimal value:

```
{"n": [[134152195, -4,1], [90587, 3,0]],

"r": [134235797, 47512],

"c": [[134220784, -1,0], [1, -16,0]]}
```

The CBOR representation sent is the following:

A3616E82831A07FF00032301831A000161DB03006172821A0800469519B998616382831A0
8000BF0200083012F00

The online tool cbor.me can be used to decode the payload.

5.1.3.6 Example with a moving tag

<u>BLE Scan (n-1): Previous tab</u>

Timestamp	BLE MAJOR	BLE MINOR	RSSI	Move
10	2048 (0x800)	3056 (0xBF0)	-52	0
10	2048 (0x800)	3057 (0xBF1)	-64	0
10	2048 (0x800)	18069 (0x4695)	-88	0
10	2049 (0x801)	45 (0x2D)	-41	0

<u>New BLE Scan (n): Tab</u>

Timestamp	BLE MAJOR	BLE MINOR	RSSI	Move
60	2048 (0x800)	3056 (0xBF0)	-52	0
60	2048 (0x800)	3057 (0xBF1)	-64	1
60	2048 (0x800)	18069 (0x4695)	-88	0
60	2049 (0x801)	45 (0x2D)	-41	0

The new tab (n) with complete STATUS of each iBeacon:

Timestamp	BLE MAJOR	BLE MINOR	RSSI	Move	Status
60	2048 (0x800)	3056 (0xBF0)	-50	0	Not sent
60	2048 (0x800)	3057 (0xBF1)	-80	1	Confirm
60	2048 (0x800)	18069 (0x4695)	-88	0	Not sent
60	2049 (0x801)	45 (0x2D)	-41	0	Not sent

The new tab (n) with RSSI optimization in each iBeacon not removed:

Timestamp	BLE MAJOR	BLE MINOR	RSSI	Move	Status
60	2048 (0x800)	3056 (0xBF0)	Not used	0	Not sent
60	2048 (0x800)	3057 (0xBF1)	-16	1	Confirm
60	2048 (0x800)	18069 (0x4695)	Not used	0	Not sent
60	2049 (0x801)	45 (0x2D)	Not used	0	Not sent

The same dataset is put in 3 specifics keys (“new”,”confirm”,”remove”):

```
{"c": [[0x8000BF1, -1,1]]}
```

The same with decimal value:

```
{"c": [[134220784, -1,1]]}
```

The CBOR representation is the following:

```
A1616381831A2001
```

The online tool cbor.me can be used to decode the payload.

5.1.3.7 Example of BLE scan reporting MAC address

Changing the type of identification to MAC address can be done thanks to the AT command "AT+BLE/CFG=0".

BLE Scan: Previous tab (n-1)

Timestamp	BLE MAC address	RSSI	Move
10	40:D5:77:CB:CF:E4	-52	0
10	40:D5:77:CB:D5:F6	-64	0
10	40:D5:77:D5:03:03	-88	0
10	F3:AE:6B:39:02:B3	-41	0

New BLE Scan (n): Tab

Timestamp	BLE MAC address	RSSI	Move
60	40:D5:77:CB:CF:D9	-56	1
60	40:D5:77:CB:CF:E4	-50	0
60	40:D5:77:CB:D5:F6	-80	0
60	40:D5:77:D5:57:12	-40	0

The new tab (n) with complete STATUS of each iBeacon:

Timestamp	BLE MAC address	RSSI	Move	Status
60	40:D5:77:CB:CF:D9	-56	1	New
60	40:D5:77:CB:CF:E4	-50	0	Confirm
60	40:D5:77:CB:D5:F6	-80	0	Confirm
10	40:D5:77:D5:03:03	-88	0	Remove
60	40:D5:77:D5:57:12	-40	0	New
10	F3:AE:6B:39:02:B3	-41	0	Remove

The new tab with RSSI optimization in each iBeacon not removed:

Timestamp	BLE MAC address	RSSI	Move	Status
60	40:D5:77:CB:CF:D9	-4	1	New
60	40:D5:77:CB:CF:E4	-1	0	Confirm
60	40:D5:77:CB:D5:F6	-16	0	Confirm
10	40:D5:77:D5:03:03	Not used	Not used	Remove
60	40:D5:77:D5:57:12	+3	0	New
10	F3:AE:6B:39:02:B3	Not used	Not used	Remove

The same dataset is put in 3 specifics keys (m for “new”, b for “confirm”, q for “remove”) and the MAC address representation is optimized (hexadecimal values).

```
{"m": [[h'40D577CBCFD9', -4, 1], [98739, 3, 0]],
 "q": [h'40D577D50303', h'B2D8F363FFB0'],
 "b": [[h'40D577CBCFE4', -1, 0], [612, -16, 0]]}
```

The CBOR representation is the following:

```
A3616D82834640D577CBCFD92301831A000181B3030061718246 40D577D5030346B2D8F363
FFB0616282834640D577CBCFE42000831902642F00
```

The online tool cbor.me can be used to decode the payload.

Decoding of the MAC addresses:

m key: (0x98739 + 0x40D577CBCFD9) = 0x40D577D55712
(=> 40:D5:77:D5:57:12)

q key: (0xB2D8F363FFB0+0x40D577D50303) = 0xF3AE6B3902B3
(=>F3:AE:6B:39:02:B3)

b key: (0x612 + 0x40D577CBCFE4) = 0x40D577CBD5F6
(=> 40:D5:77:CB:D5:F6)

5.1.4 BLE Asset Monitoring use-case

The role of the BLE Asset Monitoring is to monitor the manufacturer specific data advertised by BLE tags and to assemble LoRaWAN messages when these data fulfil a set of conditions. This feature is fed with the results of the various BLE scans done by the Wanesy Wave device, it will consequently be inactive if a Wi-Fi count only configuration is used.

5.1.4.1 Configuration downlink example

The following configuration command:

```
AT+BLE/MON=AC233FA383D2/24,956,10,0330
```

Is interpreted as follows:

- Only tags with a MAC address starting with AC233F (top 24 bits) are targeted
- Only tags with a manufacturer ID of 0x956 are targeted
- A minimum interval of 10 minutes will be respected between two LoRaWAN uplinks for a given tag
- Bytes 4,5,8 and 9 of the advertisement data will be included in the LoRaWAN message.

5.1.4.2 Uplinks contents

BLE Asset Monitoring uplinks are sent on fPort 88 and use a CBOR format.
Each message is an array of elements.
Each element contains:

- A tag mac address
- The significant bytes bitmask used (present only in the first element of each message)
- The RSSI of the last received advertisement data from this tag
- An array of significant bytes

5.1.4.3 Uplink example

The following uplink on fPort 88
CBOR representation:
838446ac233fa383d2190330384084020 10a048346ac233fa5bdd238588402000b18f48346ac233fa5bdd438548402000b18fc
CBOR decoding with http://cbor.me/ (warning, http://cbor.me/ shows all values in decimal format):
[[h'AC233FA383D2', 816, -65, [2, 1, 10, 4]], [h'AC233FA5BDD2', -89, [2, 0, 11, 244]], [h'AC233FA5BDD4', -85, [2, 0, 11, 252]]]

should be interpreted as:

- 3 tags are concerned by the message
- A bitmask of 0x330 (816 decimal) was used to select the significant bytes (ie. Bytes 4,5,8 and 9 of the manufacturer specific data)
- AC233FA383D2: last message was received with an RSSI of -65 dB, and manufacturer specific data bytes 4,5,8,9 are respectively: 2, 1, 10 and 4

- AC233FA5BDD2: last message was received with an RSSI of -89 dB and bytes were 2,0,11,244
- AC233FA5BDD4: last message was received with an RSSI of -85 dB and bytes were 2,0,11,252

5.2 WiFi COUNT Protocol

5.2.1 Presentation

Wanesy Wave Wi-Fi count application uses the COUNT protocol.

5.2.2 Payload format

Data sent in the LoRaWAN payload using the COUNT protocol has the following structure:

Field	Type	Description
period_sec	uint32	Period in second of probe request collection
counter	uint32	Number of different MAC address by probe request collect
rssi_tab[10]	uint16	Table composed of 10 Wi-Fi counter value (in 2 bytes each) with threshold RSSI interval: • RSSI value is included in interval 0 to -100 dBm • Index 0 => number of Wi-Fi collects with RSSI value between 0 to -9dbm • Index 1 => number of Wi-Fi collects with RSSI value between -10dbm to -19dbm
counter_new	uint16	Number of new MAC address detected from previous period to actual period of probe request.
counter_remove	uint16	Number of removed MAC address present on the previous period of probe request collection but absent on the actual period.

All fields are in little-Endian.

5.2.3 Example

Example of uplink payload:

840300004E020000010002000500110022005400230004017000280062004300

According to the payload structure described above, the example payload is split like this:

Period_sec	counter	Rssi_tab[10]										new	rem
84030000	4E020000	0100	0200	0500	1100	2200	5400	2300	0401	7000	2800	6200	4300

- ⇨ Collect period is 900s (0x00000384)
- ⇨ Number of counted Wi-Fi devices is 590 (0x0000024E)*
- ⇨ RSSI table with 10 values

Index	Rssi interval	Rssi[i]
0	[0, -9]	1 (0x0001)
1	[-10, -19]	2 (0x0002)
2	[-20, -29]	5 (0x0005)
3	[-30, 39]	17 (0x0011)
4	[-40, -49]	34 (0x0022)
5	[-50, -59]	84 (0x0054)
6	[-60, -69]	35 (0x0023)
7	[-70, -79]	260 (0x0104)
8	[-80, -89]	112 (0x0070)
9	[-90, -100]	40 (0x0028)

- ⇨ Number of new MAC address with previous collect is 98 (0x0062)
- ⇨ Number of remove MAC address with previous collect is 67 (0x0043)

5.3 WiFi Sniffing CBOR based protocol

5.3.1 Introduction

Wanesy Wave WiFi sniffing application uses a CBOR-based protocol.
CBOR is a binary data serialization, close to a JSON representation. The transmission of data object contains key-value pairs. The sender and the receiver must agree on the interpretation of these key-value pairs, but it will not break the compatibility if other pairs are added later (read IETF RFC7049 for more details about CBOR).
The CBOR data model used in the Wanesy Wave is explained below in order to be decoded by the application servers interfaced with the LoRa Network Server.
ESP32 sends WiFi sniffed data to the STM32. The payload is transmitted in hexascii mode with a CBOR data representation. This hexascii payload is forwarded to the LoRa Network Server via LoRaWAN message.

5.3.2 WiFi Sniffing use-case

The purpose of the WiFi sniffing mode is to retrieve surrounding WiFi adapters MAC addresses and RSSI levels. These data are timestamped and sent regularly to the LoRa Network Server where they can be fed to a back-end processing unit (Application Server).

5.3.3 Data items description

Each WiFi sniffing payload contains a “d” key, representing the collected data. These data are aggregated in an array whose each element contains:

- A 6-Byte identifier (original MAC address or its hash)
- A timestamp indicator (absolute or relative)
- An RSSI level (encoded)

5.3.3.1 Address identifier

In each payload, the address identifier of the very first element of the data array corresponds to the exact address (or its hashed value) retrieved by the device. The other elements address identifier parameters only indicate their difference from the previous element’s identifier.

5.3.3.2 Timestamp indicator

In each payload, the timestamp parameter of the very first element of the data array corresponds to the number of seconds elapsed since January 6th 1980 (GPS Epoch reference). The other elements timestamp parameters only indicate their difference from the previous element’s date.

5.3.3.3 RSSI optimization

All RSSI measurements have a negative value included in the range [-1;-94]. In order to be more efficient on the CBOR encoding, the following algorithm is applied on the RSSI value:

(RSSI + 47) / 2 = encoded RSSI

5.3.3.4 Example of WiFi sniffing payload decoding

Here is an example of a valid WiFi sniffing uplink payload in its hexadecimal representation:

a161648583462d2000406e3d1a4fbfa4e82d834664ab0c4bbe50142383463011947153112a0
483460bea31df817e381d2183461eb415efd9a418182d

It can be CBOR decoded (using the online tool cbor.me for instance), to get its human readable contents:

{"d": [[h'2D2000406E3D', 1337959656, -14], [h'64AB0C4BBE50', 20, -4], [h'301194715311', -11, 4], [h'0BEA31DF817E', -30, -2], [h'1EB415EFD9A4', 24, -14]]}

In a second time, this 5 element data array can de decoded, starting with the first element:

- MAC address (or its hash): 2D:20:00:40:6E:3D
- Timestamp of collection: 1337959656 in GPS Epoch => 2022-05-30 15:27:18 UTC (converted using online tool GPS Time Converter)
- RSSI of probe request: (-14 x 2) - 47 = -75

The second element can then be decoded, using the first element's values and the given differences held in the second element data:

- MAC address (or hash): 0x2D2000406E3D + 0x64AB0C4BBE50 ➔ 91:CB:0C:8C:2C:8D
- Timestamp: 2022-05-30 15:27:18 UTC + 20sec ➔ 2022-05-30 15:27:38 UTC
- RSSI: (-4 x 2) - 47 = -55

The remaining three elements follow the same mechanism, leading to the following values:

Third:	C1:DC:A0:FD:7F:9E on 2022-05-30 15:27:27 UTC @ -39 dBi
Fourth:	CD:C6:D2:DD:01:1C on 2022-05-30 15:26:57 UTC @ -51 dBi
Fifth:	EC:7A:E8:CC:DA:C0 on 2022-05-30 15:27:21 UTC @ -75 dBi

End of Document