Embit Binary Interface

-

LoRa® – Specific Documentation

Embit s.r.l.

Document information

Versions & Revisions

Revision	Date	Author	Comments
0.9	2015-10-29	Embit	Preliminary
1.0	2016-10-14	Embit	
1.1	2017-12-11	AS	Updated module definitions
1.2	2018-02-05	AS	Added OTAU state notifications
1.3	2019-02-13	JH	Added multicast and watchdog definitions
1.4	2019-06-24	JH	Added Service commands paragraph
1.6	2019-11-15	MD - JH	Added New LoRaEMB commands Added options for command "Send Data"
1.7	2021-03-15	AG	Added LoRaWAN Class B commands
1.8	2021-04-21	AS	Added LR1280 identifier
1.9	2023-05-23	AS	Added LoRa 2.4 Ghz specific details
2.0	2025-05-22	Embit	Added LR1121-e specification
2.1	2025-07-09	Embit	Error Correction

Index

1 Introduction

This document is an extension of the *Embit Binary Interface Overview* document [1] and describes the EBI protocol for the Embit wireless modules that support the LoRa™ modulation. This document is intended as a reference manual, although it also provides a simple usage example.

This document refers to the EBI-LoRa firmware version ***01 51 00 8A*** and upwards.

It is important to specify that, although the EBI protocol abstracts and simplifies some aspects of LoRa™ wireless networks, a good knowledge of LoRa™ concepts is useful to understand how to use EBI-LoRa.

Note that in this document the term *host* and the term *module* refer to the customer system hosting the Embit wireless module and the Embit wireless module itself, respectively. An overview of the interaction between the *host* and the *module*, using the EBI protocol, is shown in Figure 1.

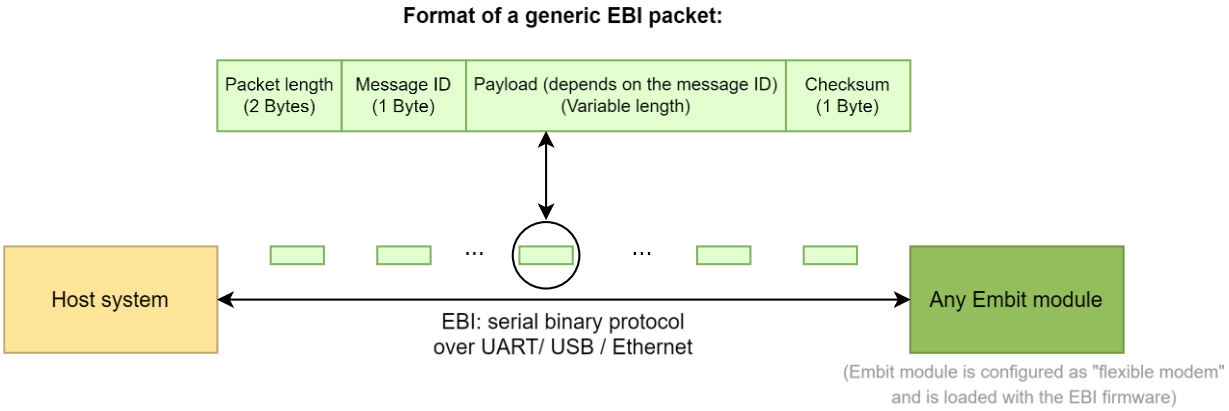


Figure 1: EBI Protocol.

Note that in this document the term *host* and the term *module* refer to the customer system hosting the Embit wireless module and the Embit wireless module itself, respectively. An overview of the interaction between the *host* and the *module*, using the EBI protocol, is shown in Figure 1.

In the following chapter a usage example to get started with LoRa® is detailed step by step, and is useful to new users.

2 EBI – LoRa® Network Structure

This chapter provides some information on the type network topologies supported by EBI – LoRa®, how to exchange data in such networks and how LoRa concepts are mapped in Ebi – LoRa®.

2.1 Network Topologies Supported

EBI – LoRa supports the same network topologies supported by LoRa, That is:

- ***LoRaWAN® Network***: An end device can be connected to the public/private networks actually in deployment in several countries. Contact Embit or refer to https://lora-alliance.org/ for further information;
- ***Embit custom network (LoRaEMB):*** Using this custom radio protocol, it is possible to establish direct connections between nodes and/or between nodes and a concentrator/coordinator, using a simple network protocol similar to the IEEE 802.15.4. This solution is suitable for local networks and for point to point communications (e.g., cable replacement).

2.2 End Device Activation (LoRaWAN®)

To participate in a LoRaWAN® network, each end-device has to be personalized and activated.

The end devices can follow a join procedure (*Over-The-Air Activation - OTAA*) prior to participating in data exchanges with the network server, using following information:

- DevEUI;
- JoinEUI;
- AppKey.

At the end of the procedure, the following parameters are set by the Network Server:

- DevAddr;
- NwkSKey;
- AppSKey.

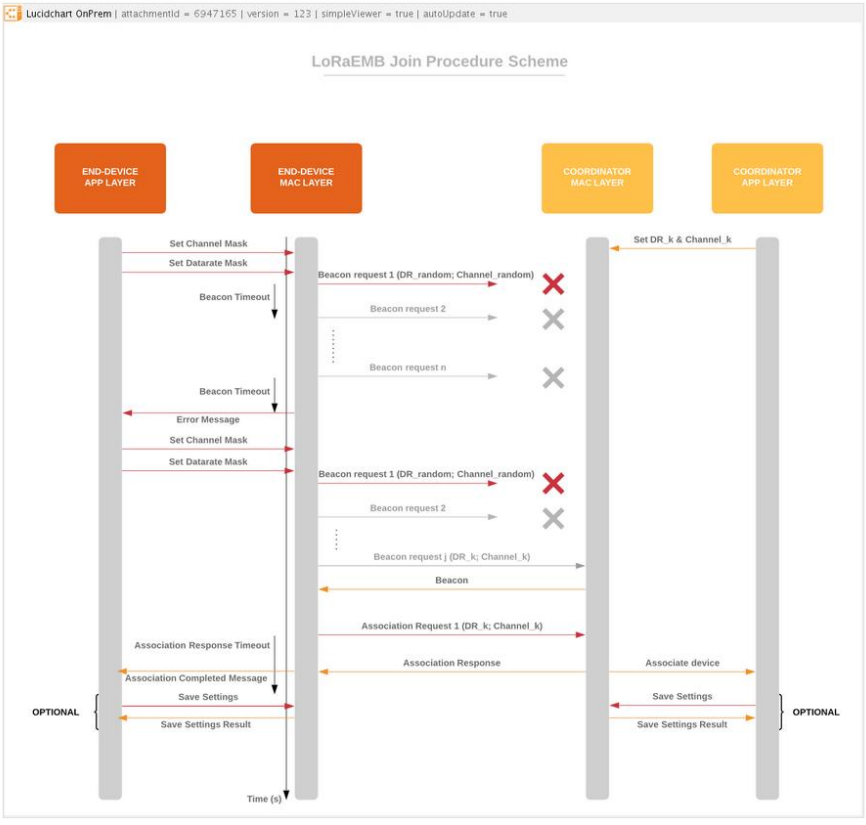
For Further information *LoRaWAN® Specification* V1.0 section 6 *End-Device Activation*.

2.3 End-Device Activation (LoRaEMB)

To participate in a star-type LoRaEMB network (when OTAA join mode is enabled), each end-device has to be perform the following Join procedure:

Each end-device sends a **Beacon Request** to the coordinator. If the End-Device has no information on the radio parameters of the coordinator (channel, SF, BW), it sends a set of Beacon Requests varying each time the channel, spreading factor and bandwidth (that can be set by the user) and waits for a **Beacon** from the coordinator in order to know its radio configuration.

After that, the End-Device sends an **Association Request** in which it specifies its IEEE address. When the coordinator receives an Association Request, it can accept the Request by sending back an **Association Response** or can reject it (based on a white/black list of IEEE device addresses). The coordinator saves the information related to each End-Device into a volatile device list. Only associated End-Devices can communicate with the coordinator in the created network.



3 EBI – LoRa® Usage Example

3.1 Introduction

This chapter provides a guide useful to get started with EBI for LoRa®. The step by step instructions provided here will guide the user through the creation of a network; moreover, some test data will be exchanged between two Embit EMB-EVB boards.

Prior to get started, it is useful to read the generic documentation about the EBI protocol (EBI documentation [1]).

3.2 Getting Started

To follow step by step this guide, you only need:

1. An EMB-EVB board equipped with the EMB-LR1121-e module programmed with EBI-LoRa.
2. The Embit EBI LoRa serial tester software, which is shipped in Embit evaluation kit's disks as *ebi-LoRa-serial-tester.exe*;
3. A PC connected to the EMB-EVB boards (through Micro – USB cables) and running the Embit EBI LoRa® serial tester software.

This guide will provide a list of command to be sent sequentially to the boards in order to establish a communication. The format of these command will be as follow:

Command ➔ Command Description

Command payload content

Where *command payload content* will be the string to copy & paste in the Payload text field of the Embit EBI LoRa® serial tester software. For almost all commands listed later in this document, the Embit module will reply to the command it receives with some bytes in the format

00 05 XX 00 YY

which will appear in the Embit EBI LoRa® serial tester software (see Figure 2); numbers are expressed in hex base. In this specific case, the 00 05 indicates the

message length of 5 bytes; the *XX* field is a code identifying the message sent by the module to the PC (it can be ignored for now) and the *YY* field is a checksum (it is ignored in the following).

The 00 field indicates the content of the response coming from the module. It depends on the specific Command ID (refer to the specific section of this document), and can be interpreted in two ways:

- A command was sent, the 00 field is the so-called *execution status byte* [1] and is important because it denotes that the command was successful (e.g., Network start); in case another value appears, then an error occurred;
- If a value was requested from the module (e.g. the network address), the value is returned.

Please refer to the EBI documentation [1] while reading this document for further details on the format of the commands sent and the notifications received.

To get started, just connect the EMB-EVB boards to a PC and open the instances of Embit EBI – LoRa® serial tester software in order to setup each module. Select the virtual serial port to each program instance and press *Connect* to initiate the communication. Before sending any commands, make sure the correct command set is selected; in this case, the *ebi LoRa* set must be chosen.

In the payload text field write the commands as indicated in the following guide pressing the *Send data* button (or ENTER key) to send the command.

Figure 2 shows a screenshot of the application, highlighting the area where the EBI commands can be entered in.

In order to verify the connection of the boards to the PC, please send the command

01

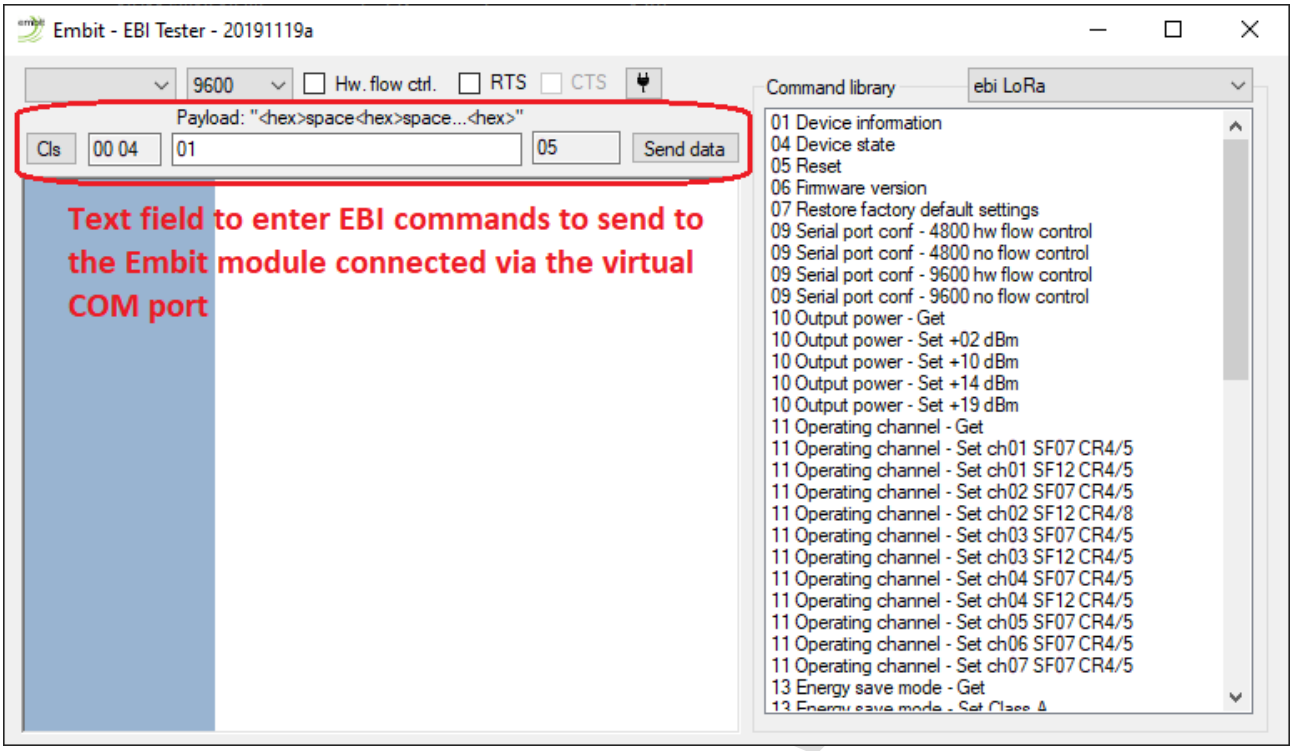


Figure 2: Serial Tester Window.

that request the *device information.* The devices will reply with a command formatted as follow:

00 0E 81 WW XX YY YY YY YY YY YY YY YY ZZ

where:

- *WW* is the protocol in use (50 for LoRa®)*;*
- *XX* identifies the module (see Chapter 3 for more details);
- *YY* are the 8 bytes of the Embit UUID (see the comments below);
- ZZ is the checksum of the packet.

If no reply is received please check the connection with the board, the baudrate set in Embit EBI LoRa® serial tester software (default 9600 baud) and the hardware flow control (typically disabled).

3.3 Conventions

In the following the Command ID is highlighted in red. In all the following examples, the first two bytes (length field) and the last one (checksum) are omitted but they shall be present in every EBI packet.

3.4 Quick Start

In order to enable the radio communication using LoRaEMB, send the following EBI command sequence:

- Set Network preferences – Set LoRaEMB
- Set Operating channel
- Set Output power
- Set Network address
- Set Network ID
- Set Energy save mode (the default value is *TX_ONLY*, set *RX_ALWAYS)*
- Start Network

In order to join a LoRaWAN® network, send the following EBI command sequence:

- Set Network Preferences – Set LoRaWAN® – Join & ADR
- Set LoRaWAN® Class (by default, the modem will remain in Class A, see command *0x13*, Section 4.10)
- Set Network Security (set AppKey)
- Set Physical Address (DevEUI, JoinEUI)
- Set Region (default WW 2G4)
- Start Network

In the next section a detailed description of each command is given.

3.5 LoRaEMB Example

3.5.1 Introduction

In this example a more complete sequence to get the module working is illustrated. The communication will be done between two Embit modules.

3.5.2 Stop network

To make sure that the start command will take effect as desired (some commands can be executed only when the network is stopped), we first stop any network process on the attached board:

30 ➔ Network Stop

Note that if the module is already in stop state (e.g., it is just powered on or reset), an execution status byte different than 00 is received. It can be safely ignored.

3.5.3 Network preference

The network preference on each device can be set manually with the associated command:

25 ➔ Network preference – Set LoRaEMB

25 00

Please repeat this command on any board involved in the network to choice of communication protocol.

3.5.4 Operating channel

In order to set the operating channel

- Channel: [868.1 MHz (ch1); 868.3 MHz (ch2); 868.3 MHz (ch3); 869.525 (ch4)](*)
- Spreading Factor: [7;12]
- Bandwidth: [125 kHz; 250 kHz]
- Coding rate: [4/5; 4/6; 4/7; 4/8]

In this example, the first method is used. For simplicity, only the channel 1, Spreading Factor 7, bandwidth 125 kHz and coding rate 4/5 is enabled by sending:

11 ➔ Operating channel – Set ch01 SF7 CR4/5

11 01 07 00 01

Please repeat this command on any board involved in the network.

Note(*): To see more channel please refer to the Operating Channel command (0x11) Section 4.9

3.5.5 Output power

The second thing to do is to set an appropriate output power according to the regulations (please check the appropriate documentation for details). As a starter let's set the output power to 10 dBm:

10 ➔ Output power – set +10 dBm

10 A0

Please repeat this command on any board involved in the network.

3.5.6 Network address

The first thing to do is to set an appropriate Network address for each device, please send:

21 ➔ Network address – Set

21 xx xx

where *xx* are the bytes of address to be set.

3.5.7 Network ID

The network ID on each device can be set manually with the associated command.

22 ➔ Network ID – Set 0x0001

22 00 01

Please repeat this command on any board involved in the network.

3.5.8 Energy save mode

Select the command

13 ➔ Energy save mode – Set always rx on

13 02

The module EMB-LRx now is enable to received data continuous.

3.5.9 Initialization of the radio interface

In the next sections the command described will start the radio interfaces of the Embit modules and initiate communications over the air. Note that at every power cycle, even if the previous parameters have been saved, the network must be started again, with the commands detailed below.

3.5.10 Start network

The network can now be restarted by issuing the command:

31 ➔ Network start

31

On the EMB-EVB board. The acknowledge of the start network command execution may take some time but will be successful (execution status byte equal to 00).

3.5.11 Exchange data over – the – air

When the network is up and running, data can be exchanged between the two boards by using the following command:

50 ➔ Send data – LoRaEMB – broadcast

50 00 00 FF FF D0 D1 D2 D3

This command will send the payload *01 02 03 04 05 06 07 08* to all devices on the network, using the broadcast network address *FF FF* and options *00 00*.

Once the data has been sent, the destination device(s) will be notified; in practice, a line like

Received data notification

50 xx xx xx xx xx xx D0 D1 D2 D3 yy

will appear on the Embit EBI LoRa® serial tester software instance attached to the module which received the data ('xx' denotes bytes which can be ignored for now). Such byte sequence is a *received data notification*; please refer to Chapter 3 for information on how this data is formatted. Note that the payload bytes which have been sent over-the-air using the other Embit serial tester instance (in this example: 01 02 03 04 05 06 07 08) are visible in such notification message.

Note:

Any EMB-EVB boards is configured with reception always enabled, so it is able to receive both direct messages and broadcast messages. The end device usually employs energy save mode (will be implemented in the future).

3.6 LoRaWAN® Example

3.6.1 Introduction

To execute this example, a LoRaWAN® gateway should be employed and connected to a LoRaWAN® server (like iot.semtech.com or other provider). The network connection will be from the module to the cloud server through the BS.

3.6.2 Network preference

The network preference on the device needs to be set manually with the associated command

25 ➔ Set Network Preferences – Set LoRaWAN® – Join & ADR

25 E0

3.6.3 Encryption keys

Set the encryption key (AppKey and NwkKey)

26 01 ➔ Set AppKey

26 01 XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

26 00 ➔ Set NwkKey

26 00 YY YY YY YY YY YY YY YY YY YY YY YY YY YY YY YY

3.6.4 Set physical address

Set DevEUI and JoinEUI

20 ➔ Set physical address

20 XX XX XX XX XX XX XX XX YY YY YY YY YY YY YY YY

The first 8 bytes identify the JoinEUI, the last 8 bytes identify the DevEUI.

3.6.5 Set region

Note: This command is only for EMB-LR1121-e module

Set the operating region

19 ➔ Set region

19 XX

Use:

- 00: EU868
- 01: US915
- 02: WW 2G4

3.6.6 Initialization of the radio interfaces

In the next sections the command described will start the radio interfaces of the Embit modules and initiate communications over the air. Note that at every power cycle, even if the previous parameters have been saved, the network must be started again, with the commands detailed below.

3.6.7 Start Network

The network can now be restarted by issuing the command:

31 ➔ Network Start

31

on the EMB – EVB board. The acknowledge of the start network command execution may take some time but will be successful (execution status byte equal to 00).

3.6.8 Exchange data over – the – hair

When the network is up and running, data can be exchangedbetween the board and the Network Server by using the following command:

50 ➔ Send data – LoRaWAN® – Ack

50 0C 06 D0 D1 D2 D3

where:

- 50: command ID;
- 0C 00: options (require ACK);
- 06: port destination
- D0 D1 D2 D3: data which will be sent over-the-air (it can be changed with any other data up to the packet size limit).

The transmission may employ some time (the first data exchange with the server may last for up to 1 minute, due to some handshake with the server). At the end of TX a Send data response (like the following) is received:

D0 ➔ Send data response

D0 XX YY ZZ

and the parameters are defined as follows:

- *XX*: excecution status byte (00 → Success);
- *YY*: number of ACKs received;
- *YY*: confirms the receipt of ACK (00 → ACK received, 01 → ACK not received).

Once the data has been sent, if the server has some data to send back to the module:

E0 ➔ Receive data notification

E0 XX XX XX XX XX D0 D1 D2 D3

will appear on the Embit EBI LoRa® serial tester software instance attached to the module.

4 EBI – LoRa® Commands

EBI binary commands allow to control each aspect of the network and the wireless module module behavior. They target embedded hosts that require advanced networking features or complex network topologies.

This chapter provides details on the format of the payload for each different packet. As detailed in [1], the generic packet format for EBI packets is:

Field	Packet length	Message ID	Payload (specific data for each message ID)	Checksum
Length	2 Bytes	1 Bytes	Variable	1 Byte

In the following sections the *Message ID* for each EBI-LoRa® command is provided, in hexadecimal format, at the end of the section name; note that the message IDs in this document match the message IDs reported in [1].
The ***Payload format*** paragraphs provide the specification for the variable-length *Payload* field.

Finally, the ***Direction*** paragraphs identify whether the packets are commands sent to the module (host → module) or are replies/notifications sent to the host (host ← module).

4.1 Device information (0x01)

Direction: host → module.

Valid when: always.

Payload format: no payload.

4.1.1 Device information response (0x81)

Direction: host ← module

Payload format:

Field	EBI protocol	Embit Module	Embit UUID
Length	1 Bytes	1 Bytes	8 Bytes

The *EBI protocol* field is divided in two sub-fields of 4 bits each as follow:

Field	EBI protocol family	EBI protocol Variant
Length	4 bits	4 bits

Any nibble can be zero indicating unknown value. This is the list of all possible values for *EBI protocol* field:

- 0x00: unknown
- 0x01: proprietary
- 0x10: 802.15.4
- 0x20: ZigBee
  - 0x21: ZigBee 2004 (1.0)
  - 0x22: ZigBee 2006
  - 0x23: ZigBee 2007
  - 0x24: ZigBee 2007 – Pro
- 0x40: Wirless M-Bus
- 0x50: LoRa®

The *Embit Module* field is divided in three sub-field is divided in three sub-field as follows:

Field	Embit Module Family	Model	Revision
Length	4 Bits	2 Bits	2 Bits

Any if these sub-fields can be zero indicating unknown information.

The list of all possible values for the *Embit module field* is:

- 0x00: unknown
- 0x10: reserved
- 0x20: EMB – ZRF2XX
    - 0x24: EMB – ZRF231XX
        - 0x26: EMB – ZRF231PA
    - 0x28: EMB – ZRF212XX
        - 0x29: EMB – ZRF212B
- 0x30: EMB – Z253X
    - 0x34: EMB – Z2530X
        - 0x36: EMB – Z2560PA
    - 0x38: EMB – Z2531X
        - 0x3A: EMB – Z2531PA – USB
    - 0x3C: EMB – Z2538X
        - 0x3D: EMB – Z2538P
- 0x40 = EMB-WMBx
    - 0x44: EMB-WMB169x
        - 0x45: EMB-WMB169
        - 0x46: EMB-WMB169PA
    - 0x48: EMB-WMB868x
        - 0x49 = EMB-WMB86
- 0x50 = EMB-Lrx
    - 0x54 = EMB-LR1272
    - 0x55 = EMB-LR1276S
    - 0x56 = EMB-LR1276S (CRYPTO)
    - 0x58 = EMB-LR1280
    - 0x5C = EMB-LR1121-e
- 0x60 = EMB-AERx

The *Embit UUID* field contains the so-called Embit Universally Unique Identifier (known as *UUID* or just *UID*) that is an 8-bytes sequence identifying the module universally.

For the **EMB-Z253x** family (i.e., EMB-Z2530PA, EMB-Z2538PA, EMB-Z2531PA-USB) the Embit UUID coincides with the concept of physical address (see the Physical Address (0x20) command, Section 4.15, for more information) and also with the unique IEEE address of the module (all such identifiers are 8 bytes long).

Finally, for the **EMB-WMBx** and **EMB-LRx** family (i.e., EMB-WMB169PA, EMB-WMB868, EMB-LR1276 and **EMB-LR1121-e**) the UUID is a unique sequence of 8 bytes which identifies the device but that does not necessarily coincide with the physical address; this is due to the fact that the devices of the EMB-WMBx family do not have a physical address hardwired in their memory. For such a reason the physical address of the module on the EMB-WMBx family must be set after start up with the *Physical Address* (0x20) command. Embit provides a valid physical address on the printed label stick on the module. Note that for the EBI-WMBus variant employed on EMB-WMBx devices there is no concept of IEEE address, since Wireless M-Bus does not employ such type of addressing.

4.2 Device state (0x04)

Direction: host → module.

Valid when: always.

Payload format: no payload.

4.2.1 Device state response / Event notification (0x48)

Direction: module → host.

Payload format: Single byte indicating the device's state:

- 0x00: booting
- 0x01: inside bootloader
- 0x10: ready (startup operations completed successfully)
- 0x11: ready (startup operations failed)
- 0x20: offline
- 0x21: connecting
- 0x22: transparent mode startup

- 0x30: online
- 0x40: disconnecting
- 0x50: reserved
- 0x51: end if receiving window
- 0x71: firmware update over the air started
- 0x72: firmware update over the air completed (reset required to switch to new fw)

Notes:

The module will send a notification after booting up with a *Ready* state (0x10 or 0x11) and then will switch to *Offline* state.

The module will also send a notification when the *Offlin*e state is entered directly from the *Online* state (indicating that the device is orphan).

A *device state notification* might also indicate that the device exited the energy save mode (see associated command, Section 4.2).

4.3 Reset (0x05)

Direction: host → module.

Valid when: always.

Payload format: no payload.

4.3.1 Reset response (0x85)

Direction: host ← module.

Payload format: single byte in the *execution status byte* format [1].

***Notes*:**

Please wait for the *device state notification* message that comes at startup after receiving the confirmation in order to allow the module to perform the hardware reset and initialize everything again.

4.4 Firmware version (0x06)

Direction: host → module.

Valid when: always.

Payload format: no payload.

4.4.1 Firmware version response (0x86)

Direction: host → module.

Payload format: 4 bytes identifying the firmware version.

4.5 Restore factory default settings (0x07)

Direction: host → module.
Valid when: network is stopped.
Payload format: no payload.
Notes:
The default settings are the following:

Paramter	Default valure for EMB – LRx modules
Operating channel	1 (868.1 MHz; SF = 7; BW = 125 kHz; CR = 4 / 5)
Transmission power	+13 dBm
Serial port settings	9600 baud, 8, N, 1, no hardware flow control
Network role	End device
Network preference	LoRaEMB custom radio network
Energy save mode	Default

Notes:

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

4.5.1 Restore factory default settings response (0x87)

Direction: host ← module.

Payload format: single byte in the *execution status byte* format [1].

Notes: The module will turn off networking when executing this command and will perform a system reset right after sending this response. Please wait for the *device state notification* message that comes at startup after receiving the response in order to allow the module to perform the hardware reset and initialize everything again.

4.6 Save Settings (0x08)

Direction: host → module.

Valid when: network is stopped.

Payload format: no payload

Notes:

Saves the currently selected serial port settings in the module's internal non-volatile memory.

Once the settings have been saved, it will be used by the module each time the module is (re)started, in place of the factory default settings. Note that the factory default settings can always be restored using EBI command ID 0x07.

Notes:

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

4.6.1 Save settings response (0x88)

Direction: host ← module.

Valid when: single byte in the *execution status byte* format [1].

4.7 Serial port configuration (0x09)

Direction: host → module.

Valid when: network is stopped.

Payload format:

The payload is 2 bytes long formatted as follows:

Field	Baudrate	Flow Control
Length	1 Bytes	1 Bytes

The *baudrate* field specifies the baudrate in use as follows:

Support on modules / Baudrate	EMB – LR1272	EMB – LR1280	EMB – LR1121 – e
0x00 = Maintain current speed.	✓	✓	✓
0x01 = 1200 baud/sec.	✓	✓	✓
0x02 = 2400 baud/sec.	✓	✓	✓
0x03 = 4800 baud/sec.	✓	✓	✓
0x04 = 9600 baud/sec.	✓	✓	✓
0x05 = 19200 baud/sec.	✓	✓	✓
0x06 = 38400 baud/sec.	✓	✓	✓
0x07 = 54600 baud/sec.	✓	✓	✓
0x08 = 115200 baud/sec.	✓	✓	✓
0x09 = 230400 baud/sec.		✓	✓
0x0A = 460800 baud/sec.			
0x0B = 921600 baud/sec.			

The *Flow control* field specifies the behavior of the RTS and CTS pins of the Embit module:

- 0x00: Flow control disabled;
- 0x01: Hardware flow control enabled in modem mode;

The *modem mode* hardware flow control means that, to communicate with the Embit module, the host should assert the RTS line and then wait for the CTS line assertion (transition from logic high to logic low) from the Embit module before sending data over the UART interface.

Notes:

The use of high baudrate values may introduce errors on the UART communications. Please keep the baudrate as low as possible when few data bytes are exchanged.

The flow control mode affects the way the module wakes up from energy save mode. While in energy save mode, the module will not be able to receive data over the UART. If modem mode hardware flow control is used, the RTS will be asserted before

sending data by the host and this will wake up the Embit's module from energy save mode. The module will then assert CTS and start receiving the data. This means that in modem mode, the device can serve commands also during energy save mode.

If you need to flash the device with EBI, make sure you close all other connections that have the same COM.

4.7.1 Serial port configuration response (0x89)

Direction: host ← module.

Payload format: Single byte in the *execution status byte* format [1].

Notes:

If the execution is acknowledged with a *success* response, the module will switch to the new settings immediately after the response has been sent, otherwise it will remain with current settings. Please wait at least 25 ms after the reception of the response to allow the module to switch to the new baudrate.

4.8 Output power (0x10)

Direction: host → module.

Valid when: network is stopped.

Payload format:

To set the output power of the module, the payload is a single signed byte indicating the output power to be used in dBm. The accepted values are:

- EMB – LR1272: [0 ; +14] dBm
- EMB – LR1276S: [0 ; +14] dBm
- EMB – LR1121-e:
  - EU 868: [0 ; +14] dBm
  - US 915: [0 ; +22] dBm
  - WW 2G4: [0 ; +11] dBm

Notes:

When the LoRaWAN® network has been selected, the output power is automatically selected by the Network Server in order to optimize the power consumption, so it is

possible only to retrieve it. **If the EMB-LR1121-e module is used, this parameter cannot be read**.

4.8.1 Output power response (0x90)

Direction: host ← module

Payload format:

If getting the current output power, the payload is a single signed byte indicating the output power in use in dBm.

If setting the channel, the payload is a single byte in the *execution status byte* format [1].

4.9 Operating channel (0x11)

Direction: host → module.

Valid when: Network is stopped.

Payload format:

To set the channel, the payload are four unsigned byte indicating the channel to be used:

Field	Channel	Spreading Factor	Bandwidth	Coding rate
Length	1 Byte	1 Byte	1 Byte	1 Byte

Channel [Hex]	Channel [Dec]	Freq [MHz]	Valid for:
0x01	1	868.100	EMB – LR127X (868 MHz)
0x02	2	868.300	
0x03	3	868.500	
0x04	4	869.525	
0x01	1	902.700	EMB – LR127X

0x02	2	902.900	(915 MHz)
0x03	3	903.100	
0x01	1	2403	EMB – LR1280 (2.4 GHz); LoRaEMB & LoRa 2.4GHz
0x02	2	2425	
0x03	3	2479	
0x04	4	2415	EMB – LR1280 (2.4 GHz) LoRaEMB only
0x05	5	2420	
0x06	6	2430	
0x07	7	2435	
0x08	8	2440	
0x09	9	2445	
0x0A	10	2450	
0x0B	11	2455	
0x0C	12	2460	
0x0D	13	2465	
0x0E	14	2470	
0x0F	15	2475	
0x10	16	2480	

Spreading Factor [Hex]	Spreading Factor [Dec]	Chirps/Symbol	Valid for
0x05	5	32	EMB – LRx (2.4 Gz)
0x06	6	64	
0x07	7	128	EMB – LRx (868 MHz)
0x08	8	256	
0x09	9	512	

0x0A	10	1024	
0x0B	11	2048	
0x0C	12	4096	
Bandwidth [Hex]	Bandwidth [Dec]	Bandwidth [kHz]	
0x00	0	125	EMB – LRx (868 MHz)
0x01	1	250	
0x00	0	200	EMB – LRx (2.4 Gz)
0x01	1	400	
0x02	2	800	
0x03	3	1600	
Coding rate [Hex]	Coding rate [Dec]	Coding Rate	
0x01	1	4 / 5	EMB – LRx
0x02	2	4 / 6	
0x03	3	4 / 7	
0x04	4	4 / 8	

Notes:

The operating channel can only be changed when network is stopped.

If the **EMB-LR1121-e** module is used and the device operates within a LoRaWAN® network, the *0x11* command allows setting the data rate at which the device transmits. When using the LoRaWAN® network, the channel is automatically assigned by the Network Server. This command must only be called after Join no ADR has been set using the *network preference* command (see command 0x25, Section 4.19), otherwise it will return an error.

This command can be used before or after the join procedure. If the custom data rate is set before the join, the module will perform the join with the user data rate.

Refer to RP002-1.0.4 Regional Parameters to verify the available settings for each region. If using the LoRaWAN® 2G4 channel plan, only the SF will be considered, as the allowed BW is limited to 812 kHz, and can be set only before the join.

For LoRaWAN® 2G4, the gateways provide only single Spreading Factor feature, so before set the datarate check the configuration of your gateway, although join will fail. The other fields bytes must be set to *00*.

If no data rate has been set, the module will use DR = 0 by default.

Spreading Factor [Hex]	Spreading Factor [Dec]	Chirps/Symbol
0x05	5	32
0x06	6	64
0x07	7	128
0x08	8	256
0x09	9	512
0x0A	10	1024
0x0B	11	2048
0x0C	12	4096

Bandwidth [Hex]	Bandwidth [Dec]	Bandwidth [kHz]
0x00	0	125
0x01	1	250
0x02	2	500
0x03	3	200
0x04	4	400
0x05	5	800

When the LoRaEMB network has been selected, channel setting depends on the join mode selected. In case of ABP join (see the Network Join mode command, Section 4.21), channels 0x01 to 0x10 are dedicated to the EU_868 region, channels 0x11 to 0x20 are dedicated to the US 915 region, and channels 0x21 to 0x30 are dedicated to the WW 2G4 region. Please refer to the table below for more information.

EMB-LR1121-e					
EU 868		US 915		WW 2G4	
0x01	868.1 MHz	0x11	902.7 MHz	0x21	2.403 GHz
0x02	868.3 MHz	0x12	902.9 MHz	0x22	2.425 GHz
0x03	868.5 MHz	0x13	903.1 MHz	0x23	2.479 GHz
0x04	869.525 MHz	0x14	903.9 MHz	0x24	2.415 GHz
0x05	867.1 MHz	0x15	904.1 MHz	0x25	2.420 GHz
0x06	867.3 MHz	0x16	904.3 MHz	0x26	2.430 GHz
0x07	867.5 MHz	0x17	904.5 MHz	0x27	2.435 GHz
0x08	867.7 MHz	0x18	904.7 MHz	0x28	2.440 GHz
0x09	867.9 MHz	0x19	904.9 MHz	0x29	2.445 GHz
0x0A	868.8 MHz	0x1A	905.1 MHz	0x2A	2.450 GHz
0x0B	869.0 MHz	0x1B	905.3 MHz	0x2B	2.455 GHz
0x0C	869.2 MHz	0x1C	904.6 MHz	0x2C	2.460 GHz
0x0D	866.1 MHz	0x1D	907.3 MHz	0x2D	2.465 GHz
0x0E	866.3 MHz	0x1E	910.7 MHz	0x2E	2.470 GHz
0x0F	866.5 MHz	0x1F	913.5 MHz	0x2F	2.475 GHz
0x10	866.7 MHz	0x20	914.7 MHz	0x30	2.480 GHz

4.9.1 Operating channel response (0x91)

Direction: host ← module.

Payload format:

If getting the current channel, the payload is a single byte indicating which channel is currently being used.

If setting the channel, the payload is a single byte in the *execution status byte* format [1].

EMB – LR1121-e: When the LoRaWAN network has been selected, if getting the current channel, the payload is a single byte indicating which data rate is currently being used.

4.10 Energy save(0x13)

Direction: host → module.

Valid when: Network is stopped.

Payload format:

Field	Module sleep policy
Length	1 Byte

The *module sleep policy* field specifies the selected energy save mode:

- 0x00 always on (LoRaWAN® Class C / LoRaEMB reception anytime): The reception is always enabled and it is possible to transmit and receive data anytime.
- 0x01 rx window (LoRaWAN® Class A /LoRaEMB receive after transmit): The module opens a *reception window* for an amount of time after each transmission before going in low power mode. It allows to receive a response to the packet just sent.
- 0x02 tx only (LoRaEMB): The module is in mono–directional transmission mode, after each transmission the radio goes automatically in sleep–mode.

Notes:

To retrieve the current energy save options, the packet must be sent with an empty payload.

If you wish to enable Class C, the class must be strictly set before the join, as during the initial handshake, the device communicates the initial class (whether Class C or A) to the Network Server. If you wish to reactivate Class A, you need to stop the network, set Class A, and perform the join procedure again. If the LoRaWAN network has been selected, or activate Class B functionalities, the device must re-execute the join in Class A (if the device is operating in Class C).

4.10.1 Energy save response (0x93)

Direction: host ← module.

Payload format:

If getting the current energy save options, the payload is a single byte as described in the request.

If setting the energy save options, the payload is a single byte in the *execution status byte* format [1].

4.11 Channel mask (0x16)

Direction: host → module.

Valid when: network is stopped. Applies to LoRaWAN® 915 MHz band (not valid for 868 MHz). Applies to LoRaEMB Network.

Notes:

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

LoRaWAN®

Payload format:

Field	Channel Mask Control	Channel Mask
Length	1 Byte	2 Bytes

Channel mask control:

- 0: Channel mask applies to channels 0-15
- 1: Channel mask applies to channels 16-31

- 4: Channel mask applies to channels 64-71
- 5: Reserved
- 6: All 125 kHz channels ON. Channel mask applies to channel 64-71
- 7: All 125 kHz channel OFF. Channel mask applies to channel 64-71

Channel mask:

- b_n = 0: channel disabled
- b_n = 1: channel enabled

where n = 0 . . . 15.

For further details please refer to LoRaWAN® Specification [2].

To retrieve the channel mask of a module, the packet must be sent with an empty payload.

To set the channel mask of a module, the payload is a 3 bytes field (most significant byte sent first) indicating the channel mask configuration and on which specific channels it will apply to.

Notes:

The channel mask is used only when joining to a 915 MHz LoRaWAN® network.

LoRaEMB

Payload format:

Field	Channel Mask
Length	2 Byte

Channel Mask:

- b_n = 0: Channel disabled
- b_n = 1: Channel enabled

Notes:

The region must be set before setting the channel mask!

The channel mask is in LSB first format i.e the first channel is represented using the first least significant bit (b_0).

To retrieve the channel mask of a module, the packet must be sent with an empty payload.

To set the channel mask of a module, the payload is a 2 bytes field (most significant byte sent first) indicating the channel mask configuration.

4.11.1 Channel mask response (0x69)

Direction:I host ← module.

Payload format:

LoRaWAN® Network

If retrieving the channel mask, the payload is a 10 bytes field indicating the overall channel mask configuration.

If setting the channel mask, the payload is a single byte in the *execution status byte* format [1].

LoRaEMB Network

If retrieving the channel mask, the payload is a 2 bytes field indicating the channel mask configuration.

If setting the channel mask, the payload is a single byte in the *execution status byte* format [1].

4.12 Watchdog configuration (0x17)

Direction: host → module.

Valid when: always.

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

Payload format:

Field	Options	Timeout
Length	1 Byte	4 bytes

Options:

- b0 (LSB)
  - 0: Disable watchdog
  - 1: Enable watchdog
- b1
  - 0: refresh on UART disable
  - 1: refresh on UART enable
- b2
  - 0: refresh on OTA (over the air) disabled
  - 1: refresh on OTA (over the air) enabled
- b3 – b7: reserved

The *timeout* field indicates the time interval after which the watchdog will be activated and the system will reset.

Notes:

To retrieve watchdog configuration of a module, the packet must be sent with an empty payload.

To set watchdog configuration of a module, the payload is a 5 bytes field (sent most significant byte first).

4.12.1 Watchdog configuration response (0x97)

Direction: host ← module.

Payload format:

If getting watchdog configuration, the payload is a 5 bytes field: the first most significant byte indicates the options field, the remaining four indicate the timeout.

If setting watchdog configuration, the payload is a single byte in the *execution status byte* format [1].

4.13 Data rate mask (0x18)

Direction: host → module.

Valid when: LoRaEMB network.

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

Payload format:

Field	Spreading factor	Bandwidth mask
Length	1 Byte	1 bytes

Spreading factor mask specifies the enabled spreading factors:

- b0 (LSB) SF7:
  - 0: disabled
  - 1: enabled
- b1 SF8:
  - 0: disabled
  - 1: enabled

  . . .
- b5 SF12:
  - 0: disabled
  - 1: enabled
- b6 – b7: reserved

and the bandwidth mask specifies the enabled bandwidths:

- b0 (LSB) 125 kHz:
  - 0: disabled
  - 1: enabled
- b1 250 kHz:
  - 0: disabled

    - 1: enabled
- b2 500 kHz
    - 0: disabled
    - 1: enabled
- b3 – b7: reserved

Notes:

To retrieve the data rate mask of a module, the packet must be sent with an empty payload.

To set the data rate of a module, the payload is a 2 bytes field (sent most significant byte first).

4.13.1 Data rate mask response (0x98)

Direction: host ← module.

Payload format:

If getting the data rate mask, the payload is a 2 bytes field: the first most significant byte indicates the spreading factor mask, the remaining byte indicates the bandwidth mask.

If setting the data rate mask, the payload is a single byte in the *execution status byte* format [1].

4.14 Region (0x19)

Direction: host → module.

Valid when: Network is stopped.

Payload format:

Field	Region
Length	1 Byte

LoRaWAN® Network:

To set the region of a module, the payload is a 1 byte field (sent most significant byte first) that specifies the operating region of the module:

- 0x00: EU 868
- 0x01: US 915
- 0x02: WW 2G4

Notes:

The Modem-E will save the set parameters if the join procedure is successfully completed.

LoRaEMB Network:

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

To set the region of a module, the payload is a 1 byte field (sent most significant byte first) that specifies the operating region of the module:

- 0x00: EU 868
- 0x01: US 915
- 0x02: WW 2G4

Notes:

To retrieve the region of operation of a module, the packet must be sent with an empty payload.

To set the region of a module, the payload is a 1 byte field (sent most significant byte first).

4.14.1 Region response (0x99)

Direction: host ← module.

Payload format:

If getting the region, the payload is a 1 byte field containing the ID of the LoRa region band.

If setting the region, the payload is a single byte in the *execution status byte* format [1].

4.15 Physical address (0x20)

Direction: host → module.

Valid when: Network is stopped.

Payload format:

Field	JoinEUI	DevEUI
Length	8 Bytes	8 Bytes

LoRaWAN Network:

To set the physical address of a module, the payload is an 8 bytes field (sent most significant byte first) indicating the physical address to be used (any value accepted).

- JoinEUI: is a global application ID in IEEE EUI64 address space that uniquely identifies the application provider (i.e., owner) of the end device.
- DevEUI: global end-device ID in IEEE EUI64 address space that uniquely identifies the end device.

Notes:

The Modem-E will save the set parameters if the join procedure is successfully completed.

4.15.1 Physical address response (0x0A)

Direction: host ← module.

Payload format:

If getting the physical address, the payload is a 16 bytes field: the first eight most significant bytes indicate the JoinEUI, the last eight indicate the DevEUI of the module.

If setting the physical address, the payload is a single byte in the *execution status byte* format [1].

4.16 Network address (0x21)

Direction: host → module

Valid when: network is stopped

Payload format:

Field	Address
Length	0 / 2 / 4 Bytes

LoRaWAN® Network

Note that the EBI *network address* corresponds, in the context of LoRaWAN® networks, to the *device address* (DevAddr). To set the network addresses, the payload is a bytes field (sent most significant byte first) indicating the network identifier to be used.

When using Over – The – Air activation, this parameter is set during network start.

Notes:

If the EMB-LR1121-e module is used, this parameter cannot be read in LoRaWAN configuration.

LoRaEMB Network

When using the LoRaEMB protocol, the network address is 2 bytes long.

The address range from 0xFFF0 to 0xFFFE is reserved, and 02xFFFF is also reserved as it is used for broadcast communication.

Network address response (0xA1)

Direction: host ← module

Payload format:

If getting the network address, the payload is a 2 / 4 bytes field (sent most significant byte first) indicating the network address in use.

If setting the network address, the payload is a single byte in the *execution status*

byte format [1].

4.17 Network identifier (0x22)

Direction: host → module

Valid when: network is stopped

Payload format:

Field	Network ID
Length	0 / 2 / 4 Bytes

To retrieve the network ID in use on the module, this packet must be sent with an empty payload.

LoRaWAN® Network:

Every LoRaWAN® network server is configured with one or more network IDs. The Network Server sets the prefix of the DevAddr it assigns to end–device based on a Net ID.

In LoRaWAN® network, the ebi *network identifier* is 4 byte long and corresponds to the *Network ID* identifier.

Notes:

If the EMB-LR1121-e module is used, this parameter cannot be read in LoRaWAN configuration.

LoRaEMB Network:

In Embit custom radio networks, the *Network ID* correspond to the *PAN ID* and identifies the specific network in use.

4.17.1 Network identifier response (0xA2)

Direction: host ← module

Payload format:

If getting the network identifier, the payload is a 2 / 4 bytes field (sent most significant byte first) indicating the network address in use.

If setting the network address, the payload is a single byte in the *execution status Byte* format [1].

4.18 Network role (0x23)

Direction: host → module.

Valid when: network is stopped.

Payload format:

To set the network role the payload is a single unsigned byte with the following meaning:

- 0x00: Coordinator
- 0x01: Reserved
- 0x02: End Device

Notes:

If using the device in a LoRaWAN® network, you can only read this parameter.

If the EMB-LR1121-e module is used, the only available setting is *End Device*.

4.18.1 Network role response (0xA3)

Direction:

Payload format:

If If getting the network role, the payload is a single unsigned byte as specified in the request packet.

If setting the network role, the payload is a single byte in the *execution status byte* format [1].

4.19 Network preference (0x25)

Direction: host → module.

Valid when: Network is stopped.

Payload format:

Field	Protocol	Auto joining	ADR	Reserved
Length	1 Bit	1 Bit	1 Bit	5 Bit

LoRaWAN® Network:

The meaning of the fields is the following:

- b7 (MSB):
  - 1 ➔ LoRaWAN® network;
  - 0 ➔ LoRaEMB network;

- b6:
  - 0 ➔ Auto Joining Disabled (Activation By Personalization – ABP)
  - 1 ➔ Auto Joining Enabled (Over–The–Air Activation – OTAA)
- B5:
  - 0 ➔ Disable ADR (Custom Data Rate)
  - 1 ➔ Enable ADR (Adaptive Data Rate)
- b4 – b0: reserved (must be set to 0)

The Auto Joining bit enables the *Over-the-Air Activation OTAA* procedure on network start. It is defined in [2]. If this procedure is enabled, the *DevEUI*, JoinEUI and *AppKey* must be set prior to the Network start.

The ADR bit enables the Adaptive Data Rate control, managed automatically by the Network Server. The ADR is available only in sub-GHz region (such as EU868 or US915).

In "Disable ADR" settings, use "operating channel (0x11)" command to set the desired Spreading Factor and Bandwidth.

Notes:

This command shall be sent as the first configuration command.

The EMB-LR1121-e supports only OTAA for LoRaWAN networks, and only ABP for LoRaEMB networks.

4.19.1 Network preference response (0xA5)

Direction: host ← module.

Payload format:

If getting the network preferences, the payload is as specified in the request packet.

If setting the network preferences, the payload is a single byte in the *execution status byte* format [1].

4.20 Network Security(0x26)

Direction: host → module.

Valid when: Network is stopped.

Payload format:

Field	Key ID	Payload
Length	1 Byte	16 Byte

LoRaWAN® Network:

The *kei ID* indicates key used for the encryption (AES 128 bit):

- 0x00: This field is only applicable for the EMB-LR1121-e device (reserved for other devices). It is a device-specific encryption key used to derive the FNwkSIntKey, SNwkSIntKey, and NwkSEncKey in LoRaWAN® 1.1. **When a LoRaWAN® 1.1 capable device connects to a LoRaWAN® 1.0.x Network Server which does not support dual root keys (NwkKey and AppKey), the NwkKey value must be set equal as the AppKey value;**
- 0x01 – AppKey: is an AES-128 application key specific for the end device, the AppKey is used to derive the session keys NwkSKey and AppSKey specific for that end device communications;

- 0x10 – NwkSKey: is a network session key specific for the end device. It is used to calculate and verify the MIC and encrypt and decrypt the payload field of MAC-only data messages;
- 0x11 – AppSKey: is a network session key specific for the end device. It is used to encrypt and decrypt the payload field of application-specific data messages and used to calculate and verify an application-level MIC.

For further information please refer to [2].

Notes:

If the EMB-LR1121-e module is used, only NwkKey and AppKey can be set, as the device does not support Activation By Personalization.

LoRaEMB Network:

The *kei ID* indicates key used for the encryption (AES 128 bit):

- 0x01 – AppKey: is an AES-128 application key specific for the end device, the AppKey is used to encrypt and decrypt the payload field of application-specific data messages for *OTAA Join* mode in LoRaEMB network;
- 0x11 – AppSKey: is an application session key specific for the end device. It is used to encrypt and decrypt the payload field of application-specific data messages for *No Join* mode in LoRaEMB network.

4.20.1 Network security response (0xA6)

Direction: host ← module.

Payload format:

If setting the network preferences, the payload is a single byte in the *execution status byte* format [1].

For security reasons, the security settings cannot be read.

4.21 Network Join mode (0x27)

Direction: host → module.

Valid when: LoRaEMB network.

Payload format:

Field	Join Mode
Length	1 Byte

The *join mode* field specifies the join mode:

- 0x00: Join mode *OTAA*
- 0x01: Join mode *No Join*

To retrieve the join mode of a module, the packet must be sent with an empty payload.

To set the join mode of a module, the payload is a 1 byte field (sent most significant byte first).

Notes:

If the EMB-LR1121-e module is used, only ABP is supported.

4.21.1 Network join mode response (0xA7)

Direction: host ← module.

Payload format:

If getting the network join mode, the payload is a single unsigned byte as specified in the request packet.

If setting the network join mode, the payload is a single byte in the *execution status byte* format [1].

4.22 Multicast group (0x29)

Direction: host → module

Valid when: network is stopped. Only for LoRaWAN® network. Only Class C end devices. Each LoRaWAN® end–device can be part of at most 4 multicast groups at the same time.

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

Payload format:

Field	Group addresses	AppSKey	NwkSKey
Length	4 Byte	16 Byte	16 Bytes

LoRaWAN® network:

The meaning of the fields are:

- Group Address: it corresponds, in the context of LoRaWAN® networks, to the network address of the multicast group common to all end-devices of the group. To set the network address, the payload is a 4 bytes field (sent most significant byte first) indicating the network identifier to be used.
- AppSKey & NwkSKey: multicast group keys. These keys are multicast group specific (different for every multicast group), but all end-devices of a given multicast group have the same keys associated to this group. To set the network multicast keys, the payload is a 32 (16+16) bytes field (sent most significant byte first) indicating the network multicast security keys to be used.

For further information please refer to [2].

4.22.1 Multicast group response (0xA9)

Direction: host ← module.

Payload format:

If retrieving multicast parameters of the end-device, the response is a field containing 4/8/12/16 bytes (depending on the number of the previously configured groups)

indicating the specific address of each group. If no multicast groups are configured, the payload is a single byte in the *execution status byte* format [1]. For security reasons, the multicast security keys cannot be read.

If adding a multicast group or clearing, the payload is a single byte in the *execution status byte* format [1].

4.23 Network stop (0x30)

Direction: host → module.

Valid when: Network is started.

Payload format: The packet has no payload.

This commands stops the network, so the transceiver will close its reception. (the power consumption will depend on the selected UART Hardware Flow Control).

4.23.1 Network stop response (0x31)

Direction: host ← module.

Payload format:

The payload is a single byte in the *execution status byte* format [1].

4.24 Network start (0x31)

Direction: host → module.

Valid when: Network is started.

Payload format: The packet has no payload.

LoRaWAN® Network:

The module will execute the *Over-the-Air Activation* as described in [2]. In this case, before network start, the DevEUI, JoinEUI, AppKey and NwkKey parameters and the region shall be set (using *physical address*, *network security* and *region* EBI commands). At the end of the procedure, the DevAddr, NwkSKey and AppSKey parameters are automatically set.

In order case, if *auto joining* is not set, before start the parameters DevAddr, NwkSKey and AppSKey shall be set using the *network address* and *network security* commands.

LoRaEMB Network:

The network will start automatically.

4.24.1 Network start response / notification (0xB1)

Direction: host ← module.

Payload format:

The payload is a single byte in the *execution status byte* format [1].

Notes:

The cause of a possible failure in a stating transmission network may be due to a missing or incorrect setting of all the required parameters.

4.25 Add device to list (0x38)

Direction: host → module

Valid when: only for LoRaEMB network.

Payload format:

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

Field	Options	IEEE Address	Receive Window (ms)	App Key
Length	1 Byte	8 Byte	2 Bytes	16 Bytes

The meaning of the various fields is as follows:

- Options: RFU
- IEEE Address: contains the extended address of the associated device

- Receive window: The module opens a *reception window* for an amount of time after each transmission before going to low power mode. It allows the reception of a response to the packet just sent.
- App Key: an AES-128 application key specific for the end device communications.

4.25.1 Add Device to list response (0XB8)

Direction: host ← module.

Payload format:

The payload is a single byte in the *execution status byte* format [1].

4.26 Clear Associated Device list (0x39)

Direction: host → module.

Valid when: only for LoRaEMB network.

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

Payload format:

This is an empty payload command. The entire list containing the associated devices is cleared.

4.26.1 Clear associated device list response (0xB9)

Direction: host ← module.

Valid when: only for LoRaEMB network.

Payload format:

The payload is a single byte in the *execution status byte* format [1].

4.27 Address translation (0x40)

Direction: host → module.

Valid when: only for LoRaEMB network.

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

Payload format:

This command can be used to retrieve the association/mapping between a network address (2 bytes long) and an extended IEEE address (8 bytes long) or vice versa. This command can be sent only to a module that has started the network as coordinator which has an internal table of associated devices.

The payload for this packet is the address of the device. The address can be a network address (2 bytes) or an extended IEEE address (8 bytes). The module will detect which address is sent based on the packet length and will respond with the translated/mapped address.

4.27.1 Address translation response (0xC0)

Direction: host ← module.

Valid when: only for LoRaEMB network.

Payload format:

If the response payload is one byte long, it is an *execution status byte* response.

Otherwise, the payload is in the following format:

Field	Execution status byte	Network Address	IEEE Address
Length	1 Byte	2 Byte	8 Bytes

The meaning of the various fields is as follows:

- Execution status byte: indicates if the data was found or retrieved correctly
- Network address: is the short address of the device.
- IEEE address: is the extended address of the device

Notes:

This packet will return a negative execution status response if the module fails to find an address association in its tables and with over the air requests.

4.28 Associating device (0x41)

Direction: host ← module.

Valid when: LoRaEMB Network.

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

Payload format:

Field	IEEE address
Length	1 Byte

This is a notification message sent from the module to the host whenever the module has started the network as coordinator and an end-device is trying to associate with it.

The *IEEE address* field contains the extended address of the device requiring association.

4.28.1 Address translation response (0xC1)

Direction: host → module.

Payload format:

The payload is a single byte in the *execution status byte* format [1].

This packet must be sent from the host to the module, to indicate if the device attempting the association is allowed to associate or not.

The *execution status byte* indicates if the device is to be associated (0x00) or rejected (anything else).

4.29 Associated device list (0x42)

Direction: host → module.

Valid when: LoRaEMB network.

THIS FEATURE IS NOT CURRENTLY SUPPORTED ON EMB-LR1121-e

Payload format:

This command has no payload. It is valid only when the module has started the network as coordinator and can be used by the host to retrieve the list of devices that have been associated so far.

4.29.1 Associated device list response (0xC2)

Direction: host → module.

Payload format:

Field	Number of associated devices	Options	IEEE Address	Receive Window (ms)	App Key
Length	1 Byte	1 Byte	8 Byte	2 Bytes	16 Bytes

The meaning of the various fields is as follows:

- Number of associated devices: indicates the total number of devices present on the list
- Options: RFU
- IEEE Address: contains the extended address of the associated device
- Network Address: is the short LoRaEMB address of the associated device
- Rx Window: The module opens a *reception window* for an amount of time after each transmission before going to low power mode. It allows the reception of a response to the packet just sent.

The same fields as described previously are repeated n-times for all the other

associated devices with n being the number of associated devices.

- App Key: is an AES-128 application key specific for the end device communications

Notes:

Sending the *Associated device list* command to a node that has started the network as router or end-device will result in an *Execution status byte* equal to 0x02.

If the coordinator has no associated devices the response will result in an *execution status byte* equal to 0x01.

4.30 Send data (0x50)

Direction: host → module.

Valid when: Network is started.

Payload format:

The payload of the send data command depends on the network protocol chosen.

LoRaWAN® Network

Field	Options	FPort	Application Data
Length	2 Byte	1 Byte	Variable

The *options* field indicates to the module which option to apply to the request:

- b15 – b12: reserved
- b11: shall be set to *1*
- b10: request acknowledge
- b09 – b0: reserved

Notes:

The FPort field accept values from 1 to 223 (0x01..0xDF); it is application specific. Other values 224..255 (0xE0..0xFF) are reserved for future extensions.

LoRaEMB Network

Field	Options	Destination Address	Application Data
Length	2 Byte	2 Byte	Variable

The *options* field indicates to the module witch option to apply to the request:

- b15 – b12: reserved
- b11: shall be set to *0*
- b10: request acknowledge
- b09 – b0: reserved

The The *destination address* is the network address of the destination device. The address *0xFFFF* is the broadcast address.

4.30.1 Send data response (0xD0)

Direction: host ← module.

Payload format:

Field	Execution status byte	Retries	RSSI of the ACK	Tx Channels Mask	Tx Data rates Mask	Tx Power (dBm)	Waiting time (ms)
Length	1 Byte	1 Byte	2 Bytes	0 / 2 Bytes	0 / 1 Bytes	0 / 1 Bytes	0 / 4 Bytes

The *execution status byte* field indicates if the data transmission was successful or not. In the following a (non exhaustive) list of possible errors:

- 0x00: Command successful
- 0x01: Generic error (e.g., network not started)
- 0x02: Invalid parameter (e.g., invalid options field)
- 0x03: Timeout (no ACK from destination device)
- 0x04: No Memory (reserved)
- 0x05: Unsupported (e.g., unsupported option)
- 0x06: Busy (e.g., detected radio traffic, transmission denied to avoid collision)

- 0x07: Notify the packet can not be sent due to duty–cycle

The *retries* field indicates the number of transmission retries (after first primary attempt) carried out while attempting to deliver data to the destination device (when an Acknowledge / retries are required).

The *RSSI of the acknowledge* field (signed integer) provides the RSSI level (in dBm) of the acknowledge received from the destination device. This field is valid only if the *Execution status byte* indicates success.

Tx datarates mask indicates the datarates used in the packet transmissions. The Tx datarates mask format is:

Datarate	0	DR6(SF7 – BW250)	DR5 (SF7–BW125)	DR4(SF8-BW125)	DR3(SF9-BW125)	DR2(SF10-BW125)	DR1(SF11-BW125)	DR0(SF125-BW125)
Bit	7	6	5	4	3	2	1	0

Tx channels mask indicates the channels used in the packet transmissions. The Tx channels mask format is:

Channel	Index 15	Index 14	Index 13	Index 12	Index 11	Index 10	Index 9	Index 8	Index 7	Index 6	Index 5	Index 4	Index 3	868.5 MHz	868.3 MHz	868.1 MHz
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TxPower field indicates the output power used in dBm and *waiting time* indicates the waiting time in milliseconds to send another LoRaWAN® packet, due to duty cycle restriction.

Notes:

If the EMB-LR1121-e module is used and the LoRaWAN Network has been selected, the payload format is:

Field	Execution status byte	ACKs counter	ACK received
Length	1 Byte	1 Byte	Variable

The *execution status byte* field indicates if the data transmission was successful or not. In the following a (non exhaustive) list of possible errors:

- 0x00: command successful
- 0x01: generic error (e.g. network not started)
- 0x02: invalid parameter (e.g. invalid options field)
- 0x03: timeout (no ACK from the Network Server)
- 0x04: no memory (reserved)
- 0x05: unsupported (e.g. unsupported option)
- 0x06: busy (e.g detected radio traffic, transmission denied to avoid collision)

The *number of ACK* received field indicates the number of the total ACKs received. If an uplink with ACK is requested, the value of the byte is incremented by 1 if the ACK is received; otherwise, the non-incremented value is displayed. The count starts from 0.

The *ACK received* field indicates acknowledge receipt confirmation. If *00* is displayed, the ACK has been received; if *01* is displayed, the ACK has not been received.

4.31 Receive data (0x60)

This packet should not be sent by the host to the module (if sent, it is ignored).

4.31.1 Received data notification (0xE0)

Direction: host ← module.

Valid when: Network is started.

Payload format:

LoRaWAN® Network

Field	Options	RSSI	FPort	Application Data
Length	2 Byte	2 Byte	1 Byte	Variable

The *options* field indicates to the host which fields are present in the packet and which options the received packet was implementing:

- b15 – b0: reserved

The *RSSI* field indicates the received signal strength in dBm (signed integer) and is always present.

The *FPort* field is application specific value.

The *Application data* field contains the payload sent by the originating device.

LoRaEMB Network

Field	Options	RSSI	Source Address	Destination Address	Application Data
Length	2 Byte	2 Byte	2 Byte	2 Byte	Variable

The *oprions* field indicates to the host which fields are present in the packet and which options the received packet was implementing:

- b15 – b0: reserved

The *RSSI* field indicates the received signal strength in dBm (signed integer) and is always present.

The *Source Address* field is the network address of the source device. The address 0xFFFF is the broadcast address.

The *Destination Address* field is the network address of the destination device. The address 0xFFFF is the broadcast address.

The *Application data* field contains the payload sent by the originating device.

5 EBI – LoRa® Class B Binary Commands

5.1 Class B introduction

The purpose of LoRaWAN® Class B is to have devices available for reception at predictable time in addiction to reception windows opened after uplink transmissions of Class A end-devices. In order to operate in Class B the device must be synchronized to the network and this synchronization can be achieved by having the gateway sending a Beacon on a regular basis. The gateway which intends to provide Class B operability must feature GPS connection to retrieve the correct network common timing. The Beacon is sent after every Beacon Period which has a fixed length of 128s. The Beacon other than network timing reference, carries some useful information that can be used by the application layer of the end-device. Once the end-device is synchronized with the network, it can periodically open one or multiple *ping slots*, scheduled within the Beacon Period. The end-device always starts with a join procedure as a Class A device and it is up to the application layer of the end-device to perform Class A to Class B switching. The LoRaWAN® MAC layer provides a set o MAC commands which allow the end-device to facilitate the synchronization with the network, the Beacon locking and the customization of the ping slot periodicity. When the first Beacon is locked the end-device must send an uplink message to inform the network server that from now on the end-device is operating in Class B (the uplink message will have a specific bit set in its Frame Control field). The end-device needs to periodically search for and receive the Beacon in order to minimize any drift of its internal clock time base with respect to network timing. The end-device might be unable to receive the Beacon for a period of time called *Beacon-less Period*. During this period the end-device cannot rely on the Beacon and on the ping slots to cancel internal drifts so it will automatically enlarge Beacon and ping slot reception windows at each iteration in order to increase the probability of regaining network synchronization. Ping slot downlink frequency and data rate are defined by the network server which may change them by sending a specific MAC command to the end device. Downlink messages can be in unicast or multicast and a single end-device can belong to a maximum four multicast groups. In order to

receive multicast commands an end-device must share the same multicast encryption key and address. Each multicast group will also share the same ping slot periodicity which is independent from the unicast ping slot periodicity. In case of multicast and unicast ping slot collision, the end-device will listen to the multicast slot. The end-device can switch back to Class A at every time. To do so, the end-device must unset the Class B bit in the Frame Control field of the next uplink message. (Both set and unset of this bit are handled by the end-device MAC layer). When an end-device is operating in class B, all the Class A functionalities are kept unchanged with a single exception: when the devices tries to send an uplink message which collides with either a ping slot or Beacon reception slot, the transmission will be aborted (this failure in transmission is signaled to the application layer by means of an error message). For more details related to LoRaWAN® Class B, please consult the LoRaWAN® official documentation [3].

Note: The Class B functionalities (and the following commands) are available only for Sub-GHz.

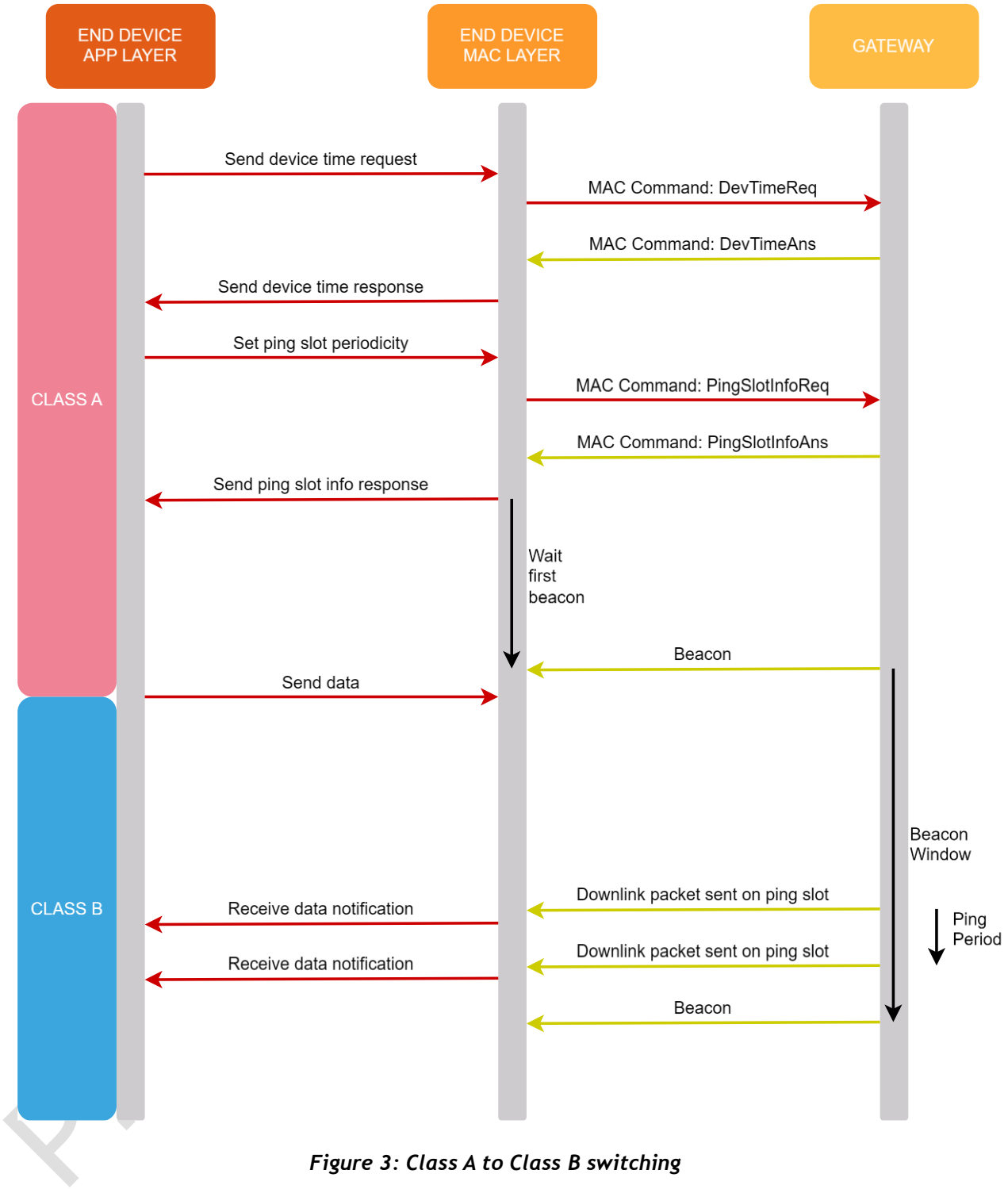


Figure 3: Class A to Class B switching

5.2 Send device time request (0x45)

Direction: host → module.

Valid when: LoRaWAN® network & network is started

Payload format:

Field	Options
Length	1 Byte

The *options* field indicates to the module which option to apply to the request:

- b7 – b0: RFU (must be all 0)

This command is used to send the LoRaWAN® MAC command *DeviceTimeReq*.

The end device will immediately try to send an uplink message with no payload embedding the MAC request. Note that the module will answer with a Send data response (0xD0) after the Class B Send device time response (0xC5) as a consequence of the data transmission attempt.

Notes:

If the EMB-LR1121-e is used, it's not necessary to send this command if you want to activate Class B, as the Modem-E will automatically send the MAC *DeviceTimeReq* command. For more information, see the Start Beacon Acquisition command (0x48,Section 5.5).

5.2.1 Send device time response (0xC5)

Direction: host → module.

Valid when: LoRaWAN® network & network is started

Payload format:

The payload is a single byte in the *execution status byte* format[1].

This command is sent twice. The first Send device time response indicates the execution of the EBI command. The second one, after the Send data response

(0xD0), notifies the reception of the ***DeviceTimeAns.*** After that the end-device is synchronized with the Network Server time reference.

Notes:

If The EMB-LR1121-e is used, the second response must have the following structure:

- 00 00 00

If no response is received, the message will have the following structure:

- 00 00 01

5.3 Set ping slot periodicity (0x46)

Direction: host → module.

Valid when: LoRaWAN® network & network is started

Payload format:

Field	Unicast Ping Slot Periodicity	Multicast Ping Slot Periodicity (Group 0)	Multicast Ping Slot Periodicity (Group 1)	Multicast Ping Slot Periodicity (Group 2)	Multicast Ping Slot Periodicity (Group 3)
Length	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte

The actual *Ping Slot Periodicity* will be approximately $0.96 \times 2^{PING\_SLOT\_PERIODICITY}$ with

$$0 \leq PING\_SLOT\_PERIODICITY \leq 7$$

$PING\_SLOT\_PERIODICITY = 0$ means that the end-device will open a Ping Slot approximately every second, while $PING\_SLOT\_PERIODICITY = 7$ means that the end-device will open a Ping Slot approximately every 128 seconds, which is the maximum Ping Slot periodicity supported.

Notes:

Multicast Session are not supported yet by the EMB-LR1121-e, so only the Unicast Ping Slot Periodicity will be considered.

5.3.1 Set ping slot periodicity response (0xC6)

Direction: host ← module.

Valid when: LoRaWAN® network & network is started

Payload format:

If getting the current Ping Slot Periodicity configuration, the payload is five bytes indicating the Ping Slot periodicity for the unicast and for each of the four multicast groups.

If setting the Ping Slot Periodicity configuration, the payload is a single byte in the *execution status byte* format [1].

5.4 Send ping slot info request (0x47)

Direction: host → module.

Valid when: LoRaWAN® network, network is started & operating in Class A

Payload format:

Field	Options
Length	1 Byte

The *options* field indicates to the module which option to apply to the request:

- b7 – b0: RFU (must be all 0)

This command is used to send the LoRaWAN® MAC command *PingSlotInfoReq*.

The end-device will immediately try to send an uplink message with no payload embedding the MAC request. Note that the module will answer with a Send data response (0xD0) after the Send device ping slot info response (0x83) as a consequence of the data transmission attempt.

Notes:

If the EMB-LR1121-e is used, the notification of the *Ping Slot Info Request* is automatically communicated to the Network Server by the Modem-E when the Start Beacon Acquisition Request (0x48,Section 5.5) is executed.

5.4.1 Send device ping slot info response (0xC7)

Direction: host ← module.

Payload format:

The payload is a single byte in the *execution status byte* format [1].

This command is sent twice. The first Send device ping slot info response indicates the execution of the EBI command. The second one, after the Send data response (0xD0), notifies the reception of the ***PingSlotInfoAns.***

5.5 Start beacon acquisition request (0x48)

Direction: host → module.

Valid when: LoRaWAN® network, network is started & operating in Class A

Payload format:

The packet has no payload.

The end-device starts to periodically open receive windows in order to lock the Beacon, when the first Beacon is locked the device automatically switches to class B. In order to inform the Network Server about the Class B switching, an uplink message must be sent after first Beacon acquisition.

Notes:

If the EMB-LR1121-e is used, the end-device will automatically send the *DeviceTimeReq* command and the *PingSlotInfoReq* command to the Network Server. When the first Beacon is locked the device automatically switches to class B. In order to inform the Network Server about the Class B switching, an uplink message with ACK request is sent automatically after the first Beacon acquisition in order to enable Class B functionality. If the end node does not detect any beacon for a duration of 2 hours, the device will stop the configuration for Class B or, if already in Class B, it will switch to Class A. In any situation, if the device attempts to enable Class B functionalities, it will send an uplink when switching back to Class A.

5.5.1 Start beacon acquisition response (0xC8)

Direction: host ← module.

Payload format:

The payload is a singe byte in the *execution status byte* format [1].

5.5.2 Beacon notification (0xC8)

Direction: host ← module

Payload format:

Field	Beacon State	Beacon RSSI	Beacon SNR	Beacon Data Rate	Beacon Frequency	Beacon Time	Beacon Info Descriptor	Beacon Info
Length	1 Byte	2 Byte	1 Byte	1 Byte	4 Byte	4 Byte	1 Byte	6 Byte

Once the end-device started Class B acquisition, at each Beacon period, the module updates the host with a Beacon notification.

The *beacon state* indicates the state of Beacon acquisition:

- 0x00: Beacon locked for the first time
- 0x01: Beacon locked
- 0x02: Beacon miss
- 0x03: Beacon lost

And the meaning of the other fields is as follows:

- Beacon RSSI: indicates the received signal strength in dBm (signed integer)
- Beacon SNR: indicates the signal-to-noise ratio in dB (signed integer)
- Beacon Data Rate: indicates the Beacon Data Rate
- Beacon Frequency: indicates the Beacon frequency in Hz
- Beacon Time: indicates the Beacon timing in seconds (Epoch Unix).
- Beacon Info Descriptor & Beacon Info: those fields provide additional information about the gateway and are therefore Gateway specific, if this information is not present in the Beacon all the bytes of these two fields are equal to 0x00.

Notes:

In case of *beacon miss* or *beacon lost*, the other fields related to Beacon info are not present. The Beacon lost state notifies the expiration of Beacon-less period and the end-device automatically switches back to Class A.

THIS FEATURE IS NOT SUPPORTED ON EMB-LR1121-e

5.6 Switch back to class A (0x49)

Direction: host → module.

Valid when: LoRaWAN® network, network is started & operating in Class B

Payload format:

The packet has no payload.

The end-device switches back to Class A. To inform the Network Server about the Class A switching, an uplink message must be sent.

5.6.1 Switch back to Class A response (0xC9)

Direction: host ← module.

Payload format:

The payload is a single byte in the *execution status byte* format [1].

5.7 Sleep management in Class B

The Sleep management of a device operating in Class B is essentially the same of a device operating in Class A. The only difference is that even when hardware flow control is enabled and the device is in sleep state, the physical layer automatically wakes up the module for the duration of every Beacon or Ping Slot period in order to allow the reception of the packet. As soon as the receiving window expires or the downlink is received, the physical layer puts again the module into sleep state.

6 Annex

6.1 Disclaimer

The information provided in this and other documents associated to the product might contain technical inaccuracies as well as typing errors. Regulations might also vary in time. Updates to these documents are performed periodically and the information provided in these manuals might change without notice. The user is required to ensure that the documentation is updated and the information contained is valid. Embit reserves the right to change any of the technical/functional specifications as well as to discontinue manufacture or support of any of its products without any written announcement.

6.2 Trademarks

Embit is a registered trademark owned by Embit s.r.l..

All other trademarks, registered trademarks and product names are the sole property of their respective owners.

6.3 References

Ref	Version	Date	Author	Title
[1]	Rev 2.1	2015	Embit	Embit Binary Interface Overview
[2]	V1.0	2015	LoRa® Alliance (http://lora-alliance.org/)	LoRaWAN® Specification
[3]	V1.03	2018	LoRa® Alliance (http://lora-alliance.org/)	LoRaWAN® Specification