Embit Binary Interface

-

LoRa™-specific Documentation

embit s.r.l.

Document information

Versions & Revisions

Revision	Date	Author	Comments
0.9l	2015-10-29	Embit	Preliminary
1.0	2016-10-14	Embit	
1.0.1	2016-11-07	Embit	

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

Index

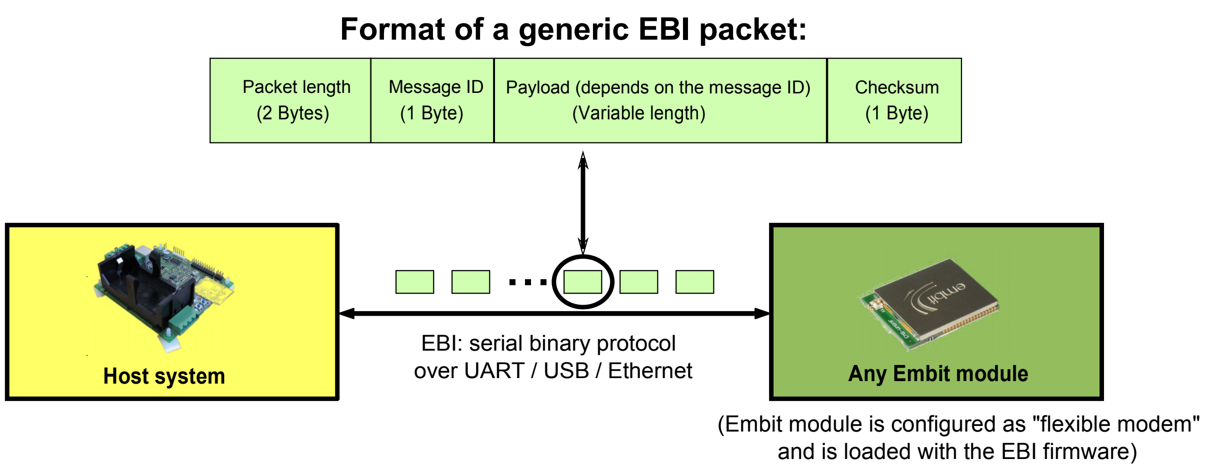
1 Introduction

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 the following figure:



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

In Chapter 3 the list of commands specifically supported by LoRa™ is provided, in the form of reference manual.

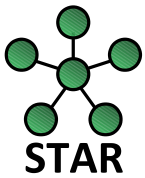
2 EBI-LoRa Network Structure

This chapter provides some information on the type of 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 http://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, 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 , AppEUI , AppKey*

At the end of the procedure, the following parameters are set:

- *DevAddr , NwkSKey , AppSKey*

Without the use of the join procedure, the above parameters must be set before sending any data (*Activation By Personalization - ABP*).

For further information please refer to the document "*LoRaWAN™ Specification*" V1.0 section 6 "*End-Device Activation*".

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. two EMB-EVB boards (mounting Embit modules programmed with EBI-LoRa, like the EMB-LR1272);
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 two EMB-EVB boards (through 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 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 Img.1); 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:

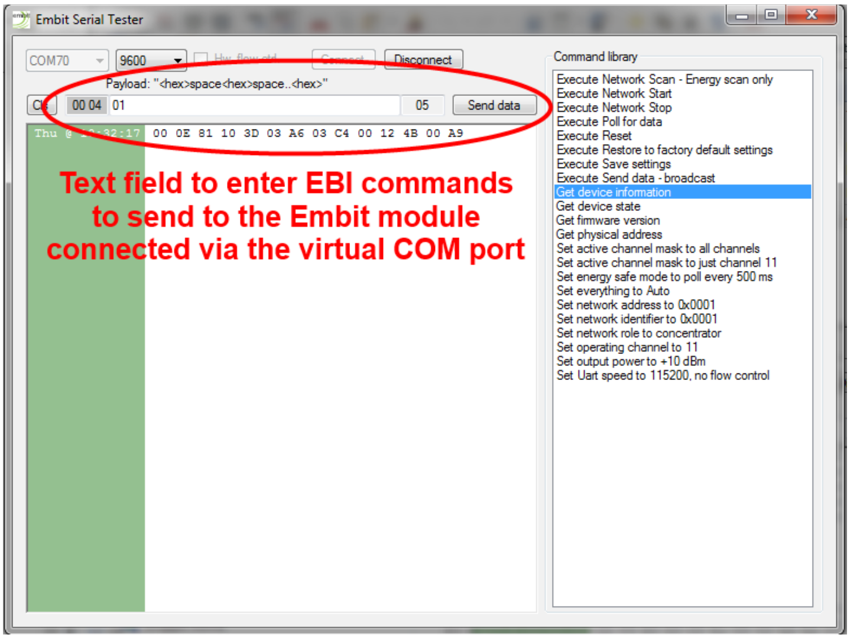
- if a command was sent, the 00 field is the so-called “execution status byte” [1] and is important because it denotes that the command was successful (e.g., Network start); in case another value appears, then an error occurred;
- if a value was requested from the module (e.g., the network address), 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 two EMB-EVB boards to a PC and open two 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" button to initiate the communication.

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.

Following is a screenshot of the application, with highlighted the area where the EBI commands can be entered:



Img. 1: Serial Tester Window

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

"01 Device information"
01

The devices will reply with a command formatted as follows:

Device information response
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 is the module (see Chapter 3 for 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 fist 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, send the following EBI command sequence:

LoRaEMB operating mode:

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

LoRaWAN operating mode:

- Stop Network
- Set Network preferences → LoRaWAN
- Set LoRaWAN classes (default class A)
- Set Network address
- Set Network Security (set encryption keys AppSKey, NwkSKey)
- 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 the shortest command sequence to get the module working is illustrated. The communication will be done between two Embit modules.

3.5.2 Energy save mode

Select the command:

"13 Energy save mode – Set always rx on"

13 02

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

3.5.3 Start network

The network can now be restarted by issuing the command:

"31 Network start"
31

on the two 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).

3.5.4 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
E0 xx xx xx xx xx xx D0 D1 D2 D3 xx

For more details about advanced EBI commands and features, please refer to the following Chapter of this document.

3.6 LoRaEMB Advanced Example

3.6.1 Introduction

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

3.6.2 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”
30

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

3.6.3 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.

3.6.4 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 13 dBm:

“10 Output power - Set +10 dBm”
10 0A

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

3.6.5 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.

3.6.6 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.

3.6.7 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.

3.6.8 Energy save mode

Select the command:

"13 Energy save mode - Set always rx on"

13 00

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

3.6.9 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.

3.6.10 Start network

The network can now be restarted by issuing the command:

"31 Network start"
31

on the two 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).

3.6.11 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

where:

- “50” is the send command ID;
- “00 00” are the send options;
- “FF FF” is the destination address (broadcast);
- “D0 D1 D2 D3” is example data which will be sent over-the-air (it can be changed with any other data up to the packet size limit).

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

Received data notification
E0 xx xx xx xx xx xx D0 D1 D2 D3 xx

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).

3.7 LoRaWAN Example

3.7.1 Introduction

To execute this example, a LoRaWAN Base Station 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.

3.7.2 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"
30

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

3.7.3 Network preference

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

"25 Network preference – Set LoRaWAN - noJoin"
25 A0

3.7.4 Network address

Set the network address:

"21 Network address – Set"
21 xx xx xx xx

where XX are the bytes of address to be set.

3.7.5 Encryption keys

Set the encryption keys (NwkSKey and AppSKey)

"26 Network security – Set NwkSKey"
26 10 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx

and

"26 Network security – Set AppSKey"
26 11 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx

where XX are the bytes of 128-bit key to be set.

3.7.6 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.

3.7.7 Start network

The network can now be restarted by issuing the command:

"31 Network start"
31

on the two 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).

3.7.8 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 – LoRaWAN - ACK"
50 0C 00 06 D0 D1 D2 D3

where:

- "50" is the send command ID;
- "0C 00" are the send options (require acknowledge);
- "06" is the destination port;
- "D0 D1 D2 D3" is example 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:

Send data response
D0 00 03 FF EC

where:

- "00" is the execution status byte (success)
- "03" is the number of retries;
- "FFEC" is the RSSI of the acknowledge (int16 big endian = -20 dBm)

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

Received 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 Binary 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 Byte	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 Byte	1 Byte	8 Bytes

The “EBI protocol” field is divided in two sub-fields of 4 bits as follows:

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 the “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 = Wireless M-Bus
- 0x50 = LoRa™

The “Embit Module” field is divided in three sub-fields as follows:

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

Any of 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-Z2530PA
  - 0x38 = EMB-Z2531x
    - 0x3A = EMB-Z2531PA-USB
  - 0x3C = EMB-Z2538x
    - 0x3D = EMB-Z2538PA
- 0x40 = EMB-WMBx
  - 0x44 = EMB-WMB169x
    - 0x45 = EMB-WMB169T
    - 0x46 = EMB-WMB169PA
  - 0x48 = EMB-WMB868x
    - 0x49 = EMB-WMB868
- 0x50 = EMB-LRx
    - 0x54 = EMB-LR1272

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 for more information) and also with the unique IEEE address of the module (all such identifiers are 8 bytes long).

For the **EMB-ZRF2xx** family (i.e., EMB-ZRF231PA, EMB-ZRF212B) the UUID is a unique sequence of 8 bytes which is NOT necessarily a valid IEEE address (even if it is used as default IEEE/physical address); this is due to the fact that the devices of the EMB-ZRF2xx family do not have a physical/IEEE address hardwired in their memory. For such a reason the physical address of the module on the EMB-ZRF2xx family must be set after start up with the “Physical Address (0x20)” command. Embit provides a valid physical/IEEE address on the printed label stick on the module.

Finally, for the **EMB-WMBx** and **EMB-LRx** family (i.e., EMB-WMB169PA, EMB-WMB868 and **EMB-LR1272**) the UUID is a unique sequence of 8 bytes which identifies the device but that does not necessarily coincides 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 (0x84)

Direction: host ← module.

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 of receiving window

Notes:

The module will send a notification after booting up with a "Ready" state (0x10 or 0x11) and then will switch to "Offline" or "Online" state (depending on the result of the startup operations, see "Auto network creation" bit in the "Network automated settings" command).

The module will also send a notification when the "Offline" 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).

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:

Parameter	Default value 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

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 settings (operating channel, transmission power, serial port settings, addresses, etc) in the module's internal non-volatile memory.

Once the settings have been saved, they 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.

IMPORTANT NOTE: THIS FEATURE IS NOT CURRENTLY SUPPORTED!

4.6.1 Save settings response (0x88)

Direction: host ← module.

Payload format: 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 Byte	1 Byte

The “baudrate” field specifies the baudrate in use as follows:

Support on modules:	EMB-Lr1272
0x00 = Maintain current speed	V
0x01 = 1200 baud/sec.	V
0x02 = 2400 baud/sec.	V
0x03 = 4800 baud/sec.	V
0x04 = 9600 baud/sec.	V
0x05 = 19200 baud/sec.	V
0x06 = 38400 baud/sec.	V
0x07 = 57600 baud/sec.	V
0x08 = 115200 baud/sec.	V
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 MCU 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, especially with the speed values marked with an asterisk (*). 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 MCU and this will wake up the Embit 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.

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 retrieve the current output power, the packet must be sent with an empty payload.

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

Accepted values:

EMB-LR1272	[+2, +19] dBm

Notes:

When joined to a LoRaWAN network, the output power is automatically selected by the concentrator in order to optimize the power consumption, so it is possible only to retrieve it.

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 retrieve the current channel the packet must be sent with an empty payload.

To set the channel the payload are a 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-LRx
0x02	2	868.300	
0x03	3	868.500	
0x04	4	869.525	
Spreading Factor [hex]	**Spreading Factor [dec]**	**Chips/symbol**	
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	
Coding rate [hex]	**Coding rate [dec]**	**Coding Rate**	
0x01	1	4 / 5	
0x02	2	4 / 6	
0x03	3	4 / 7	
0x04	4	4 / 8	

Notes:

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

When using the LoRaWAN network, the channel is automatically assigned by the concentrator. It is possible only to retrieve the actual channel number.

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].

4.10 Energy save (0x13)

Direction: host → module.

Valid when: network is stopped.

Payload format:

To retrieve the current energy save options, the packet must be sent with an empty payload. To set the energy save options, the payload is as follow:

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)
0x01 = RX WINDOW (LoRaWAN™ Class A / LoRaEMB receive after transmit)
0x02 = TX ONLY (LoRaEMB transmit only)

ALWAYS ON: The reception is always enabled and it is possible to transmit and receive data anytime.

RX WINDOW: 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.

TX WINDOW: The module is in mono-directional transmission mode, after each transmission the radio goes automatically in sleep-mode.

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 Physical address (0x20)

Direction: host → module.

Valid when: network is stopped.

Payload format:

Field	AppEUI	DevEUI
Length	8 Bytes	8 Bytes

- *AppEUI*: 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.

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

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).

Note:

The physical address is used only when joining to a LoRaWAN network.

4.11.1 Physical address response (0xA0)

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 physical address of the module, the last eight indicate AppEUI.

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

4.12 Network address (0x21)

Direction: host → module.

Valid when: network is stopped.

Payload format:

Field	Address
Length	0 / 2 / 4 Bytes

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

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 4 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.

LoRaEMB Network:

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

The address 0xFFFF indicates the broadcast address. Addresses form 0xFFF0 to 0xFFFE are reserved.

4.12.1 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].

4.13 Network identifiers (0x22)

Direction: host → module.

Valid when: network is stopped.

Payload format:

Field	Nework 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:

In LoRaWAN network, the EBI "network identifier" is 4 bytes long and corresponds to the "Network ID" identifier.

LoRaEMB Network:

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

4.13.1 Network identifiers response (0xA2)

Direction: host ← module.

Payload format:

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

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

4.14 Network role (0x23)

Direction: host → module.

Valid when: network is stopped.

Payload format:

To retrieve the selected network role, the packet must be sent with an empty payload.

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

0x00 = Coordinator
0x01 = *Reserved*
0x02 = End Device

Note: only the End Device role is supported on LoRaWAN networks.

4.14.1 Network role response (0xA3)

Direction: host ← module.

Payload format:

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].

4.15 Network automated settings (0x24)

Direction: host → module.

Valid when: network is stopped.

Payload format:

The payload can be empty for reading the current setting or formatted as follows for writing it:

Field	Reserved	Reserved
Length	1 Byte	1 Byte

Notes:

Reserved for future use.

4.15.1 Network automated settings response (0xA4)

Direction: host ← module.

Payload format:

If getting the network automated settings, the payload is as specified in the request.

If setting the network automated settings, the payload is a single byte in the “execution status byte” format [1].

4.16 Network preference (0x25)

Direction: host → module.

Valid when: network is stopped.

Payload format:

To retrieve the active network preferences, the packet must be sent with an empty payload.

To set the network preferences, the payload is formatted as follows:

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

The meaning of the fields is the following:

b7(MSB)→	0 = LoRaEMB network protocol 1 = LoRaWAN network protocol
b6 →	0 = Auto Joining Disabled (*Activation By Personalization - ABP*) 1 = Auto Joining Enabled (*Over-The-Air Activation - OTAA*)
b5 →	0 = disable ADR (Adaptive Data Rate) 1 = enable ADR (Adaptive Data Rate)
b4...b0→	Reserved

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

The ADR bit enables the Adaptive Data Rate control, managed automatically by the base station.

IMPORTANT:

This command shall be sent as the first configuration command.

When 'Auto Joning' is enabled the Network Join Mode (0x27) command configuration will be ignored.

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].

4.17 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 Key ID indicates key used for the encryption (AES 128 bit):

0x00 = *Reserved*
0x01 = AppKey
0x10 = NwkSKey
0x11 = AppSKey

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.

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.

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].

LoRaEMB network:

Not implemented.

4.17.1 Network security response (0xA6)

Direction: host ← module.

Payload format:

For security reasons, the security settings cannot be read.

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

4.18 Network join mode (0x27)

Direction: host → module.

Valid when: network is stopped.

Payload format:

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

To set the network join, the payload is formatted as follows:

Field	Mode
Length	1 Byte

Mode parameter value can be:

0x00 = ABP (activation by personalisation)
0x01 = *OTAA (over-the-air-activation)*

4.19 RX2 channel (0x2A)

Direction: host → module.

Valid when: network is stopped.

Payload format:

To retrieve the rx2 datarate and frequency, the packet must be sent with an empty payload.

To set the network join, the payload is formatted as follows:

Field	Datarate	Channel frequency
Length	1 Byte	4 bytes

The datarate field accept values from 0 to 6 representing the data rate for the second Receive window.

The Channel frequency field indicates the channel frequency in Hz for the second Receive window.

Note:

The rx2 channel frequency is in big endian order.

4.20 Battery level (0x2B)

Direction: host → module.

Valid when: always.

Payload format:

To retrieve the battery level, the packet must be sent with an empty payload.

To set the battery level, the payload is formatted as follows:

Field	Battery level
Length	1 Byte

The battery level field indicates the level of the battery, from 0 to 255.

0: The end-device is connected to an external power source.

1..254: The battery level, 1 being at minimum and 254 being at maximum.

255: The end-device was not able to measure the battery level.

4.21 Network type (0x2D)

Direction: host → module.

Valid when: network is stopped.

Payload format:

To retrieve the LoRaWAN network type public or private, the packet must be sent with an empty payload.

To set the LoRaWAN network type to public or private, the payload is formatted as follows:

Field	LoRaWAN Type
Length	1 Byte

0x00: LoRaWAN Public Network.

0x01: LoRaWAN Private Network.

The LoRaWAN type field indicates the LoRaWAN network type, which configures mainly syncword that the transceiver going to use for transmission, 0x34 for public network and 0x12 for private network.

4.22 MAC counters (0x2E)

Direction: host → module.

Valid when: network is started.

Payload format:

To retrieve the LoRaWAN MAC counters uplink and downlink, the packet must be sent with an empty payload. The response is formatted as follows:

Field	Uplink counter	Downlink counter
Length	4 Bytes	4 Bytes

Note:

The device's uplink and downlink counter are in big endian order, read-only and can not be set.

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 and the whole modules goes lo low power mode (the power consumption will depend on the selected UART Hardware Flow Control).

4.23.1 Network stop response (0xB0)

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 stopped.

Payload format:

The packet has no payload.

LoRaWAN network:

If the "Auto Joining" in network preferences is set, the module will execute the "Over-the-Air Activation" as described in [2]. In this case, before network start, the DevEUI AppEUI and AppKey parameters shall be set (using "physical address" and "network security" EBI commands). At the end of the procedure, the DevAddr, NwkSKey and AppSKey parameters are automatically set.

In the other case, if "Auto Joining" is not set, before network 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].

Note:

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 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	Retries	Application data
Length	2 Bytes	1 Byte	0 / 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 (Confirmed uplink)
b9 → - Enable retries field (only LoRaWAN confirmed uplink)
b8 → - Return Tx information
b09 ↔ b0 - Reserved

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.

The “retries” field indicates the number of transmission retries.

LoRaEMB Network:

Field	Options	Dst Address	Application data
Length	2 Bytes	2 Bytes	Variable

The “Options” field indicates to the module which option to apply to the request:

b15 ↔ b12 - Reserved
b11 → - Shall be set to “0”
b10 ↔ b0 - Reserved

The “Dst Address” is the network address of the destination device. The address 0xFFFF is the broadcast address.

4.25.1 Send data response (0xD0)

Direction: host ← module.

Payload format:

Field	Execution status byte	Retries	RSSI of the Acknowledge	Tx Channels Mask	Tx Datarates Mask	TxPower (dBm)	Waiting Time (milliseconds)
Length	1 Byte	1 Byte	2 Bytes	0 / 2 Bytes	0 / 1 Byte	0 / 1 Byte	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:

- 00 : Command successful
- 01 : Generic error (e.g., network not started)
- 02 : Invalid parameter (e.g., invalid options field)
- 03 : Timeout (No Acknowledgment form destination device)
- 04 : No Memory (reserved)
- 05 : Unsupported (e.g., unsupported option)
- 06 : Busy (e.g., detected radio traffic, transmission denied to avoid collision)
- 07 : Notify the packet can not be sent due to duty-cycle restriction
- 08 : Busy in join OTAA operation
- 09 : The device is shut down by LoRaWAN server.

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.

The “Tx Datarates Mask” field indicates the datarates used in the packet transmissions.

Tx Datarates Mask format:

Bit	7	6	5	4	3	2	1	0
Datarate	0	DR6 (SF7 - BW250)	DR5 (SF7 - BW125)	DR4 (SF8 - BW125)	DR3 (SF9 - BW125)	DR2 (SF10 - BW125)	DR1 (SF11 - BW125)	DR0 (SF12 - BW125)

The “Tx Channels Mask” field indicates the channels used in the packet transmissions.

Tx Channels Mask format:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

The “TxPower” field indicates the output power used in dBm.

The “Waiting Time” field indicates the waiting time in milliseconds to send another LoRaWAN packet, due to duty cycle restriction.

4.26 Received data (0x60)

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

4.26.1 Received data notification (0xE0)

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	RSSI	FPort	Application Data
Length	2 Byte	2 Byte	1 Bytes	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	Src Address	Dst Address	Application Data
Length	2 Byte	2 Byte	2 Bytes	2 Bytes	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 “Src Address” field is the network address of the source device. The address 0xFFFF is the broadcast address.

The “Dst 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 Annex

5.1 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.

5.2 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.