HOW TO SET UP THE 3000150xDxx TO WORK WITH THE DEMO GUI

USER GUIDE

Description

This document will guide you through an example setup of the LoRaMiP GUI.

Contents

1. Overview

The LoRa modem module will be controlled by a PC software connecting via the serial port.

The device will exchange some messages with a similar device.

The software LoRaMiP GUI can be downloaded directly from the official Mipot website, link below.

https://mipot.com/en/

After logging in, in the product category, search for the product 32001505xxx (where xxx corresponds to the same version of the DevKit you have). Here, in the documents section, there are two versions (32 and 64 bit) of the software.

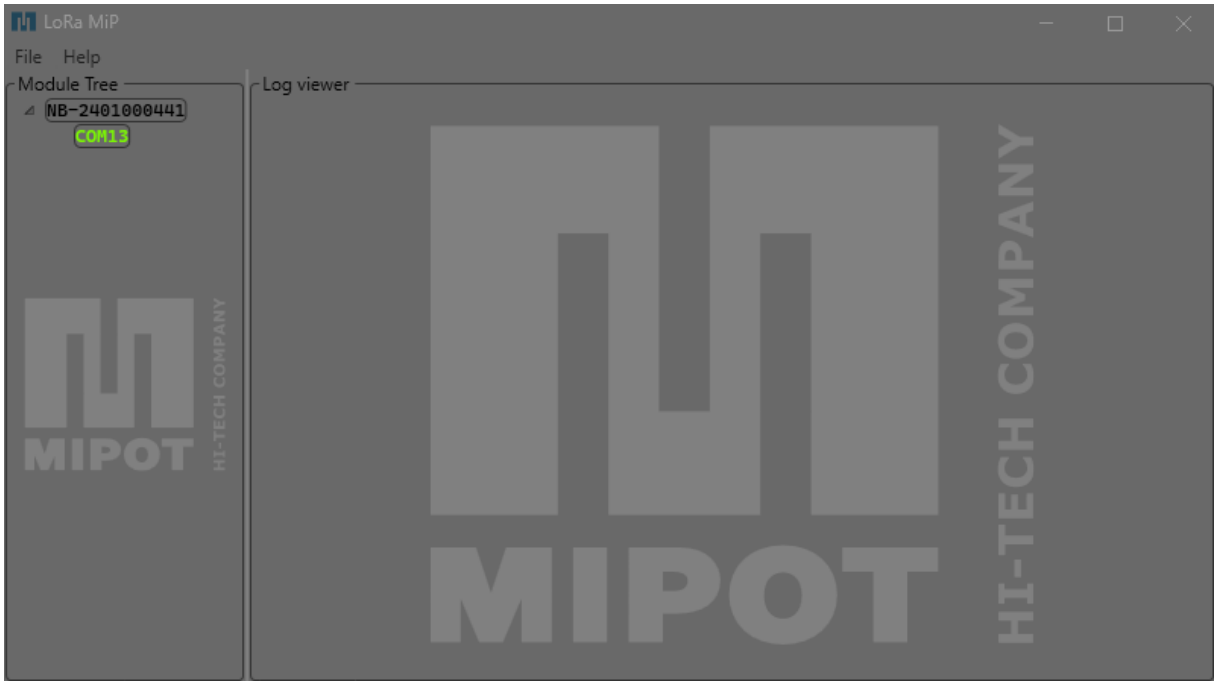
As regards the hardware setup of the DevKit, refer to the document AN_HWS001.

For details about the commands, please see the 3200150xDxx command reference.

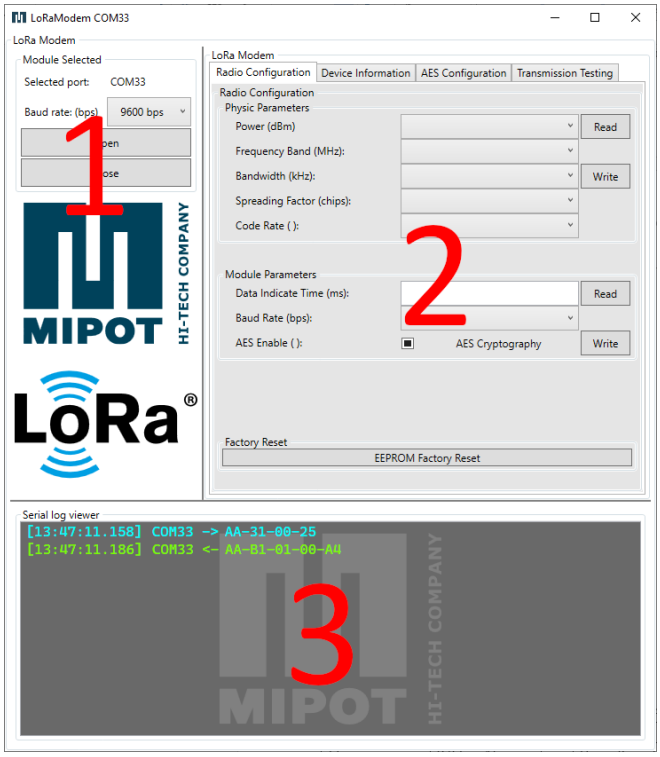
2. Software setup

2.1. Starting the GUI

Once all connections are made and the module are powered, start the LoRaMiP GUI. At startup, it will scan available serial ports looking for connected devices. It is possible to connect multiple devices to the same PC.



Once the scan is finished, the serial ports with the module connected will be highlighted in green. To use one of the module, double click on the appropriate COM port.



The view that will open is divided in 3 main zones:

1)Serial port control

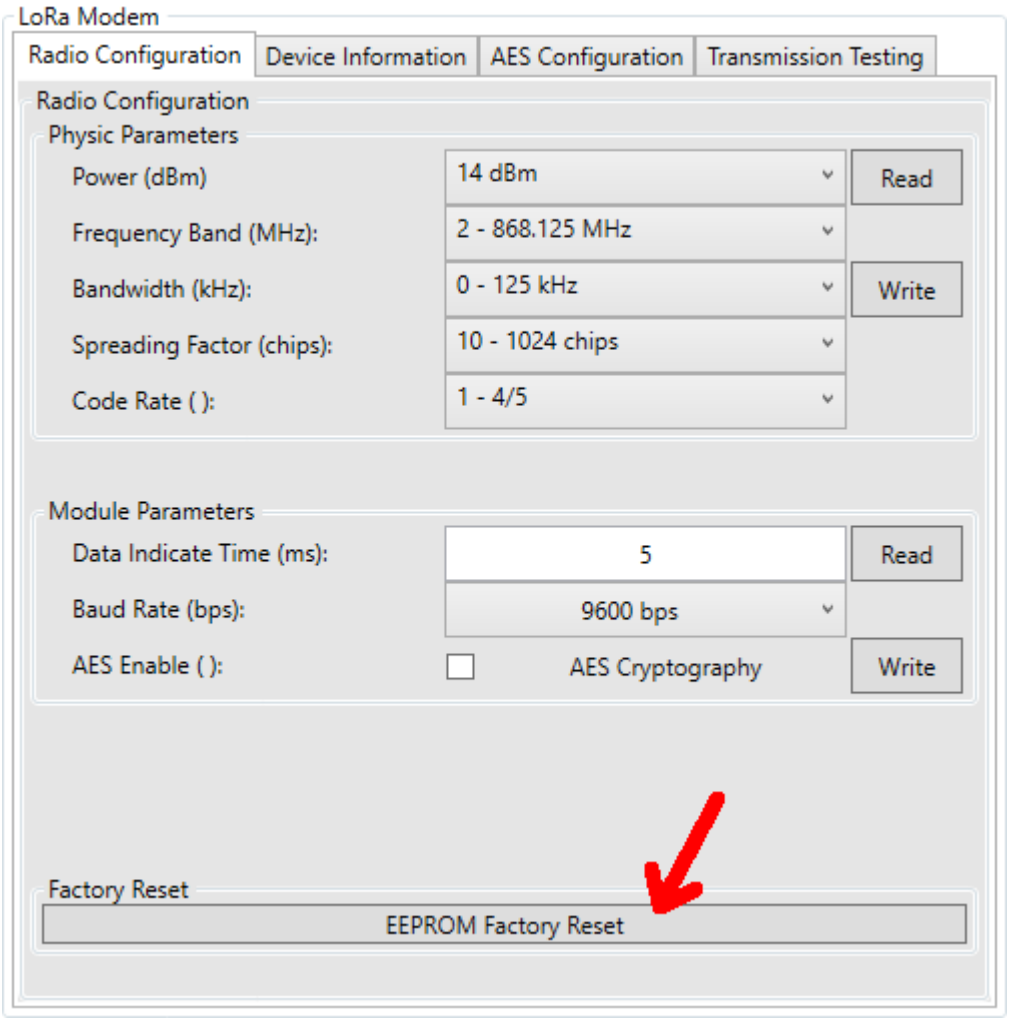
2)Module control

3)Log of the messages exchanged on the serial port.

In the log, the cyan text is about messages from the pc to the module and are indicated with an arrow pointed right, while the green messages are from the module and have an arrow pointing left.

2.2. Module reset

To start with a known configuration reset the module using the "*Factory Reset*" button in the "*Radio Configuration*" tab of the GUI.



This will configure the module with the default parameters as shown in the next table.

Parameter	Value
Power	14 (dBm)
Frequency Band	2 (868.125 MHz)
Bandwidth	0 (125 kHz)
Spreading Factor (chips)	10 (1024)
Code Rate	1 (4/5)

Parameter	Value
Data Indicate Time	5 (ms)
Baud Rate	0 (9600)
AES enable	0 (disabled)

Before set a parameter, read them with the appropriate button so all the fields are filled with the values configured in the module.

Command example

Reset the module to start with a known configuration of the module with the command FACTORY_RESET_CMD (0x31)

Host: 0xAA, 0x31, 0x00, 0x25

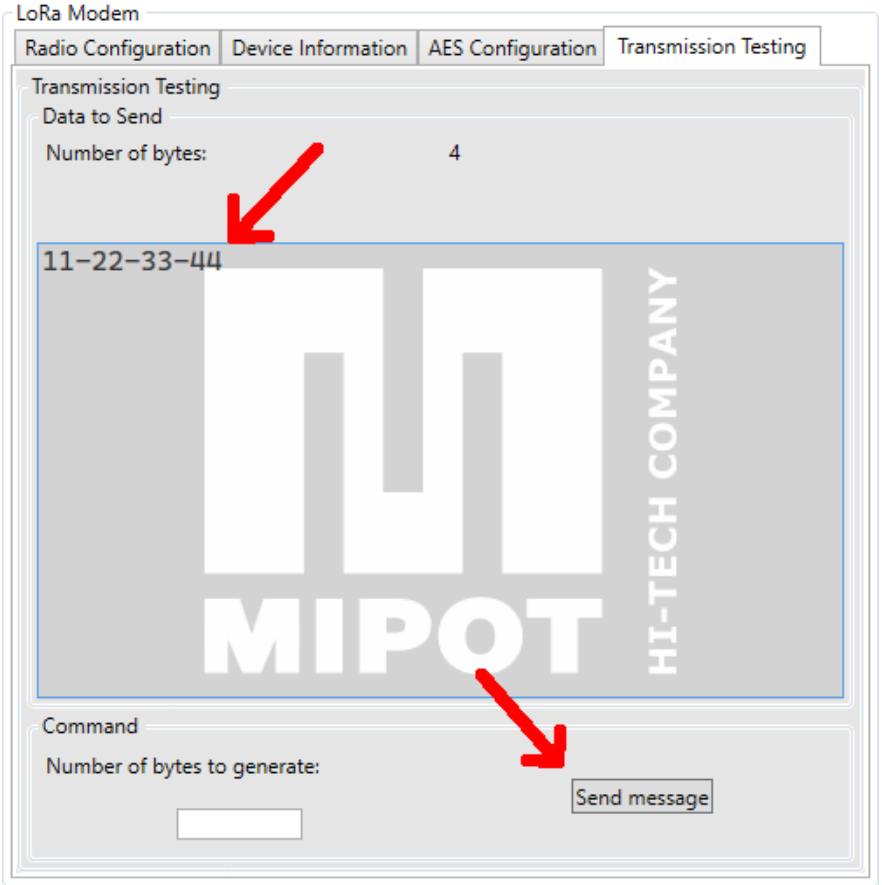
Device:0xAA, 0xB1, 0x01, 0x00, 0xA4

3. Sending and receiving a message

For this operation connect two DevKits and open the GUI for each one.

The radio parameters should be the same on both devices.

Use the "*Transmission Testing*" tab to send a message.



Write the payload (hex format) in the text box and click "*Send message*".

The module will then send a message and signal the completion with an indicate message.

On the GUI opened on the second module, an indication message will contain the transmitted payload.

Command example

Send a message with the TX_MSG_CMD (0x50).
For example to send the payload "**0x11, 0x22, 0x33, 0x44**":

Host: 0xAA, 0x50, 0x04, 0x11, 0x22, 0x33, 0x44, 0x58
Device:0xAA, 0xD0, 0x01, 0x00, 0x85

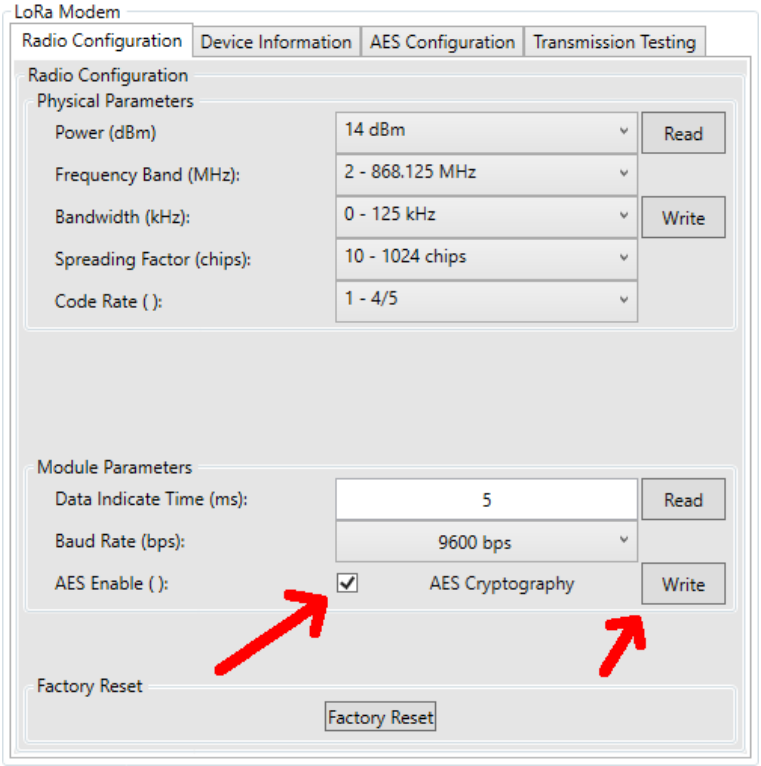
Once the transmission is complete, the module sends TX_MSG_IND (0x52):

Device:0xAA, 0x52, 0x05, 0x00, 0xCF, 0x00, 0x00, 0x00, 0x30

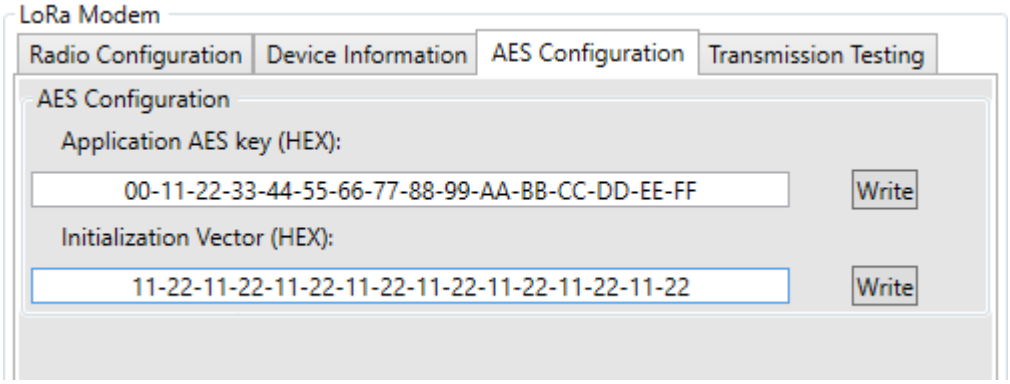
The second module, indicates the reception with the RX_MSG_IND (0x53):

Device:0xAA, 0x53, 0x08, 0x00, 0xF2, 0xFF, 0x09, **0x11, 0x22, 0x33, 0x44**, 0x57

4. Using AES ciphering



Enable the AES ciphering in the "*Radio Configuration*" tab and write the configuration in both modules.



Write the AES key and the Initialization Vector on both modules using the "*AES Configuration*" tab.

From now on, every message sent from the module will be ciphered with this key, and will be decoded correctly only by those modules having the same key and initialization vector.

Since the AES ciphering is a block cipher, the payload will be padded with 0x00 to reach a length that is a multiple of the block length (16 bytes).

Command example

Write the AES key and the Initialization Vector:

For example to use the key "**0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF**" use the SET_AES_KEY_CMD (0x58):

Host: 0xAA, 0x58, 0x10, **0xFF, 0xEE, 0xDD, 0xCC, 0xBB, 0xAA, 0x99, 0x88, 0x77, 0x66, 0x55, 0x44, 0x33, 0x22, 0x11, 0x00**, 0xF6

Device:0xAA, 0xD8, 0x01, 0x00, 0x7D

To set the initialization vector "**0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22**" use the : SET_IV_CMD (0x59):

Host: 0xAA, 0x59, 0x10, **0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11, 0x22, 0x11**, 0x55

Device:0xAA, 0xD9, 0x01, 0x00, 0x7C

Enable the cipher writing 1 to the AppEnAES parameter using the EEPROM_WRITE_CMD (0x32):

Host: 0xAA, 0x32, 0x02, 0x07, 0x01, 0x1A

Device:0xAA, 0xB2, 0x01, 0x00, 0xA3

The above commands should be issued to all modules in the network.

To send a message, proceed as usual.

Host: 0xAA, 0x50, 0x04, 0x11, 0x22, 0x33, 0x44, 0x58
Device:0xAA, 0xD0, 0x01, 0x00, 0x85

Once the transmission is complete, the module sends TX_MSG_IND (0x52):

Device:0xAA, 0x52, 0x05, 0x00, 0x21, 0x01, 0x00, 0x00, 0xDD

The second module, indicates the reception with the RX_MSG_IND (0x53):

Device:0xAA, 0x53, 0x14, 0x00, 0xEE, 0xFF, 0x09, 0x11, 0x22, 0x33, 0x44, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4F

Contrary to the previous case, now the receiver shows the payload with the padding included.

5. Revision History

Revision	Date	Description
0.1	11.11.2022	First version
0.2	22.11.2022	Change image and product code in the title
0.3	06.12.2022	Change the image on the first page Change the name of the document to “User Guide”
0.4	17.05.2024	Add configuration for module programming Change image and product code in the title
0.5	20.01.2025	Title changed Hardware setup removed