

HOW TO SET UP THE 3000150xBxx TO WORK WITH THE THINGS NETWORK

USER GUIDE



Description

This document will guide you through an example setup of the Things Network console.

Contents

1. Overview	3
2. Software setup	4
2.1. Module reset	5
Command example	5
2.2. Creating the application	6
2.3. Register an end device.....	7
Command example	8
2.4. Configure the radio.....	10
Command example	10
2.5. Enable the public network.....	11
Command example	11
2.6. Join the network.....	12
Command example	12
3. Sending a confirmed message	13
Command example	14
4. Sending an unconfirmed message	14
Command example	14
5. Receiving a message from the server.....	15
Command example	15
6. Revision History	16

1. Overview

The LoRaWAN module will be controlled by a PC software connecting via the serial port.
The device will join the network with the OTAA (Over The Air Activation) method.

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 3200150xBxx command reference.

2. Software setup

2.1. Starting the GUI

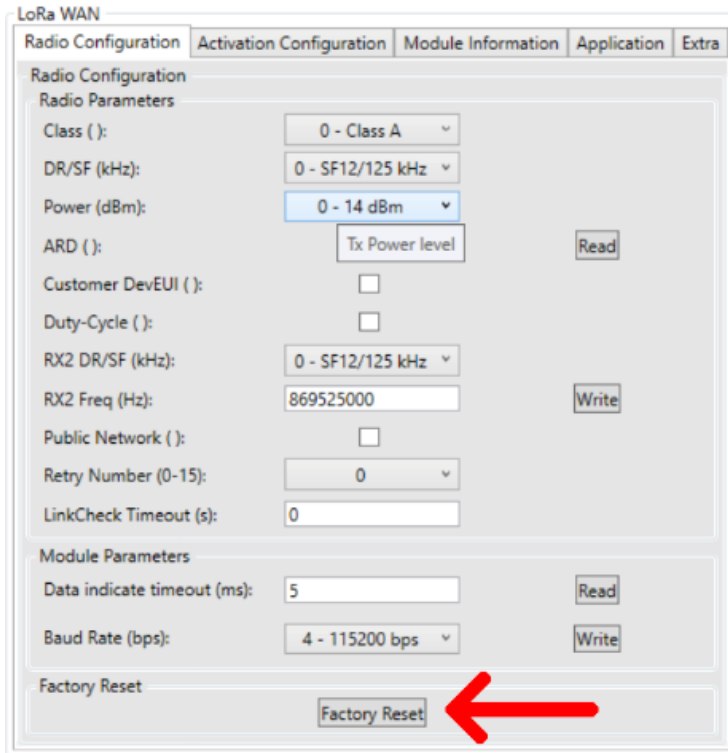
Once all connections are made and the module are powered, start the LoRaMiP GUI. At startup, it will scan available serial ports looking for connected devices. It is possible to connect multiple devices to the same PC.



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.

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.



The screenshot shows the LoRa WAN GUI with the 'Radio Configuration' tab selected. The 'Factory Reset' button is located at the bottom of the 'Radio Parameters' section and is highlighted with a red arrow.

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

Parameter	Value
Class	0 (Class A)
DR/SF	0 (SF12, 125 kHz)
Power	0 (14 dBm)
ADR	1 (Enabled)
Duty Cycle Control	1 (Enabled)

¹ 923300000 Hz for the BUS version

² This parameter will be changed to work with The Things Network

Parameter	Value
Unconfirmed Tx repetitions	0
Enable customer EUI	0 (Disabled)
RX2 Data Rate	0 (SF12, 125 kHz)
RX2 Frequency	869525000 Hz ¹
Public Network Enable	0 (Private network) ²

Command example

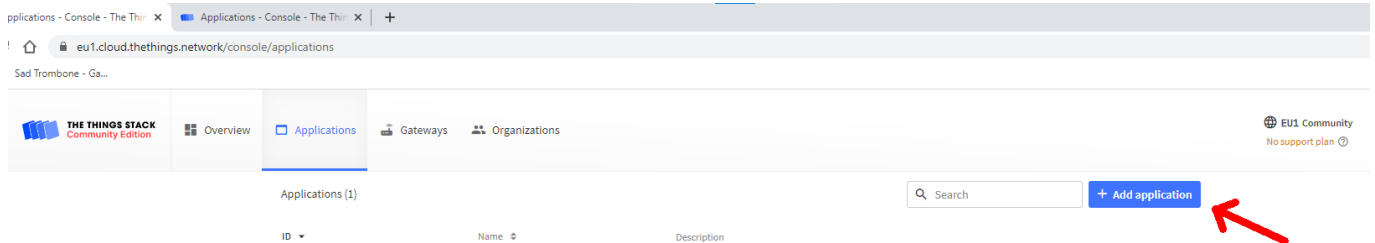
Reset the module to start with a known configuration of the module with the command `FACTORY_RESET_CMD (0x31)`

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

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

2.3. Creating the application

Log in in the Things Network Console, open the “Applications” section and click on “Add Application”.



Write an application ID and click “Create application”.

Add application

Application ID *

Application name

Description

Optional application description; can also be used to save notes about the application

The only mandatory information is the “Application ID” and must contain only lowercase letters, numbers and dashes (-)

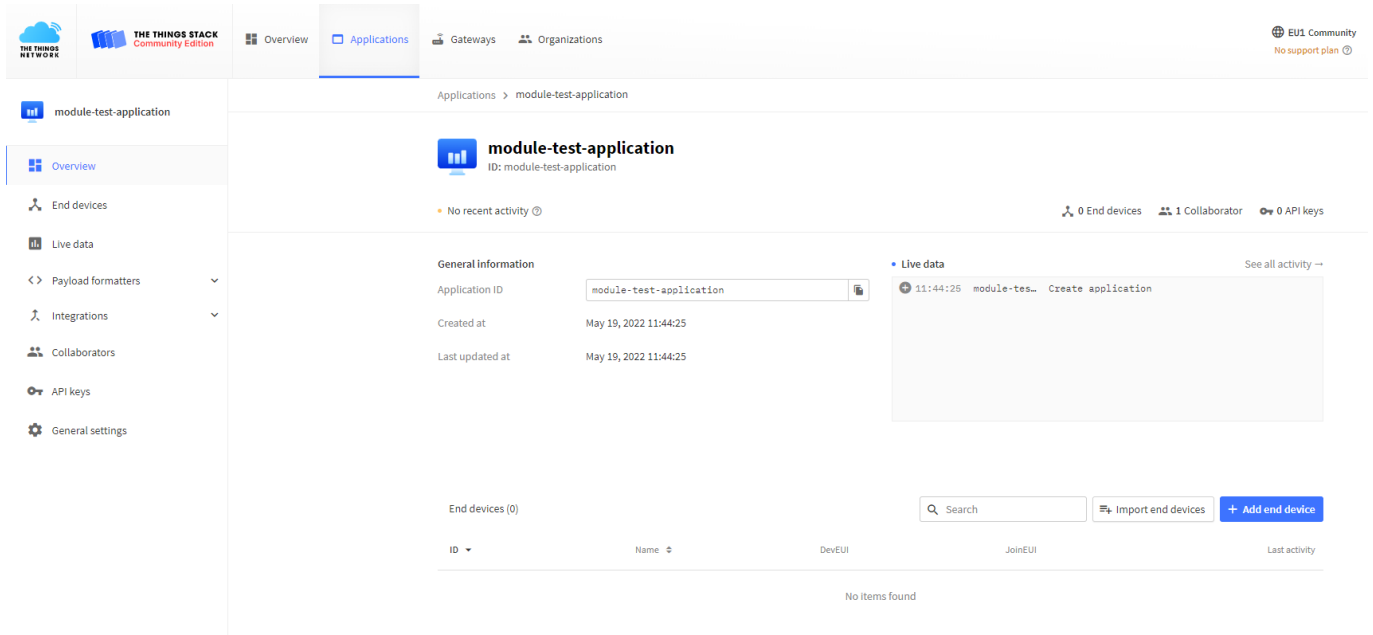
It’s not possible to reuse an application id for more than one app, even if the old app is deleted.

The other fields can contain any text and are shown in the application list.

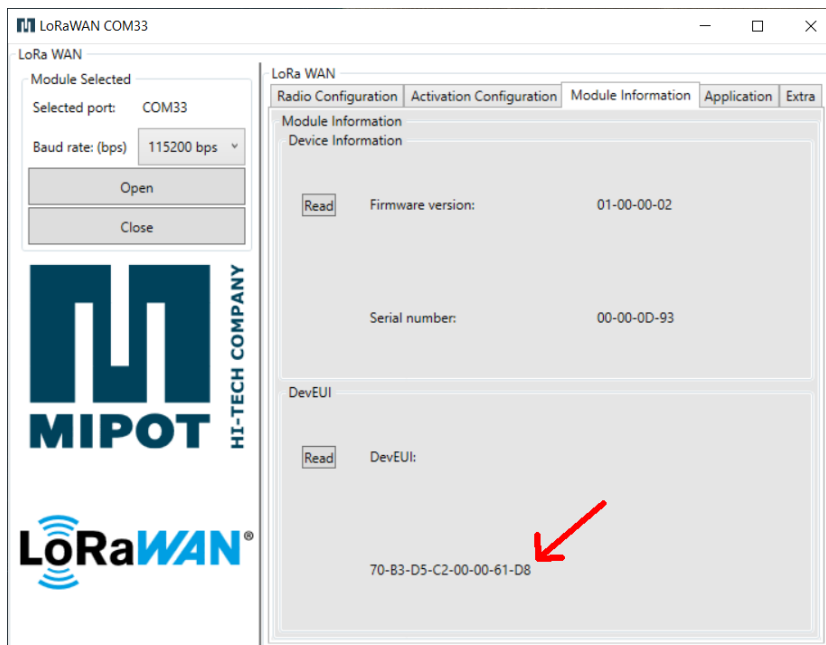
The “Application name” and “Description” can be changed later in the application “General settings”.

2.4. Register an end device

Once created, the application page is loaded and from there it's possible to add a new device by clicking to "Add end device".



For the next step, it's necessary to know the DevEUI of the end node, which can be read from the Mipot LoRaWAN Demo GUI in the "Module Information" tab.



This is the DevEUI of the module itself, but if needed the module can be configured to use a custom one.

Command example

The DevEUI can be read with the GET_DEVEUI_CMD (0x36)

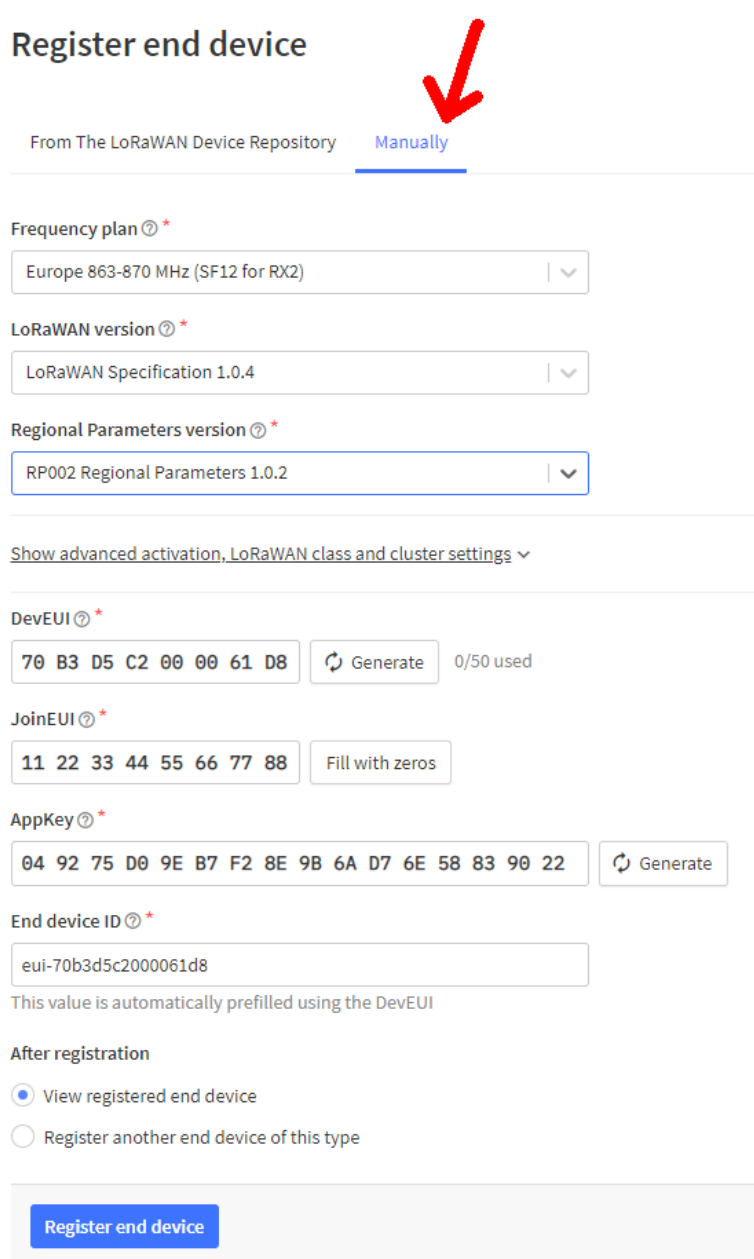
Host: 0xAA, 0x36, 0x00, 0x20

Device: 0xAA, 0xB6, 0x08, **0xD8, 0x61, 0x00, 0x00, 0xC2, 0xD5, 0xB3, 0x70**, 0xA5

In this case, the DevEUI is **70 B3 D5 C2 00 00 61 D8**; if needed the module can be configured to use a custom one.

On the TTN's Console, configure the parameters of the new device using the manual configuration.

Register end device



From The LoRaWAN Device Repository **Manually**

Frequency plan *

Europe 863-870 MHz (SF12 for RX2) | v

LoRaWAN version *

LoRaWAN Specification 1.0.4 | v

Regional Parameters version *

RP002 Regional Parameters 1.0.2 | v

[Show advanced activation, LoRaWAN class and cluster settings](#) v

DevEUI *

70 B3 D5 C2 00 00 61 D8 Generate 0/50 used

JoinEUI *

11 22 33 44 55 66 77 88

AppKey *

04 92 75 D0 9E B7 F2 8E 9B 6A D7 6E 58 83 90 22 Generate

End device ID *

eui-70b3d5c2000061d8

This value is automatically prefilled using the DevEUI

After registration

☒ View registered end device

☐ Register another end device of this type

Register end device

Select the manual configuration to open the view with all the parameters.

Configure the “*Frequency plan*”, “*LoRaWAN version*”, and “*Regional Parameters version*” as in the image.

For the EU version SF12 for RX2 is the default configuration of the module, and indicates the spreading factor to use by the gateway when sending a message in the RX2 window.

For the US version, select the frequency plan appropriate for your locale.

The Dev EUI is the value taken from the Module's GUI.

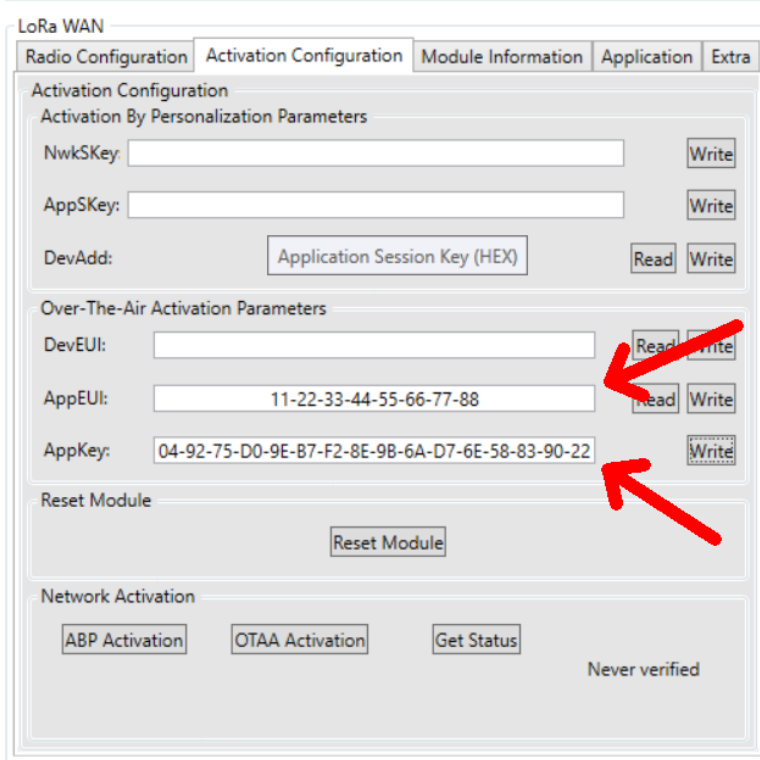
The JoinEUI is a value used to identify the join server. Since the module is programmable, it's possible to write any number except all zeros. Take note of this number as it will be needed to configure the module.

The AppKey is used to create the session key and must be a random number. Take note of this number as it will be needed to configure the module.

Once everything is configured, click on “*Register end device*” to complete the procedure.

2.5. Configure the radio

The module needs to be configured with the correct values of JoinEUI and AppKey to join the network server.



The screenshot shows the 'LoRa WAN' configuration interface with the 'Activation Configuration' tab selected. Under 'Activation By Personalization Parameters', there are fields for NwkSKey, AppSKey, and DevAdd. Under 'Over-The-Air Activation Parameters', there are fields for DevEUI, AppEUI (containing '11-22-33-44-55-66-77-88'), and AppKey (containing '04-92-75-D0-9E-B7-F2-8E-9B-6A-D7-6E-58-83-90-22'). Red arrows point to the AppEUI and AppKey fields. At the bottom, there are buttons for 'ABP Activation', 'OTAA Activation', and 'Get Status', with a 'Never verified' status indicator.

Using the GUI, write the AppEUI and the AppKey in the appropriate fields in the “Activation Configuration”.

Command example

The AppEUI value is stored in the EEPROM memory starting at the address 0x08 and can be written using the EEPROM_WRITE_CMD (0x32).

In the previous page, on the TTN console the JoinEUI/AppEUI has been set to **11 22 33 44 55 66 77 88**.

Host: 0xAA, 0x32, 0x09, 0x08, **0x88, 0x77, 0x66, 0x55, 0x44, 0x33, 0x22, 0x11**, 0xAF

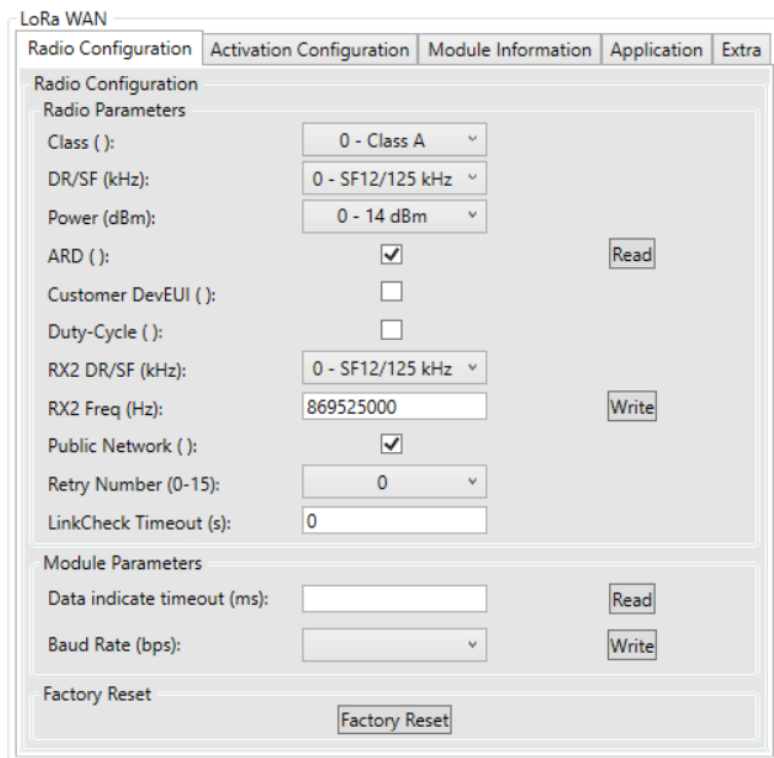
Device: 0xAA, 0xB2, 0x01, 0x00, 0xA3

The AppKey is written with the SET_APP_KEY_CMD (0x43).

Host: 0xAA, 0x43, 0x10, **0x22, 0x90, 0x83, 0x58, 0x6E, 0xD7, 0x6A, 0x9B, 0x8E, 0xF2, 0xB7, 0x9E, 0xD0, 0x75, 0x92, 0x04**, 0x7C

Device: 0xAA, 0xC3, 0x01, 0x00, 0x92

2.6. Enable the public network



The screenshot shows the 'LoRa WAN' configuration window with the 'Radio Configuration' tab selected. The 'Radio Parameters' section includes the following settings:

- Class (): 0 - Class A
- DR/SF (kHz): 0 - SF12/125 kHz
- Power (dBm): 0 - 14 dBm
- ARD (): ☒
- Customer DevEUI (): ☐
- Duty-Cycle (): ☐
- RX2 DR/SF (kHz): 0 - SF12/125 kHz
- RX2 Freq (Hz): 869525000
- Public Network (): ☒
- Retry Number (0-15): 0
- LinkCheck Timeout (s): 0

The 'Module Parameters' section includes:

- Data indicate timeout (ms):
- Baud Rate (bps):

At the bottom, there is a 'Factory Reset' button.

To set the radio configuration, open the “Radio Config” tab and read the configuration.

Configure the parameters as shown in the image on the side and write them in the module.

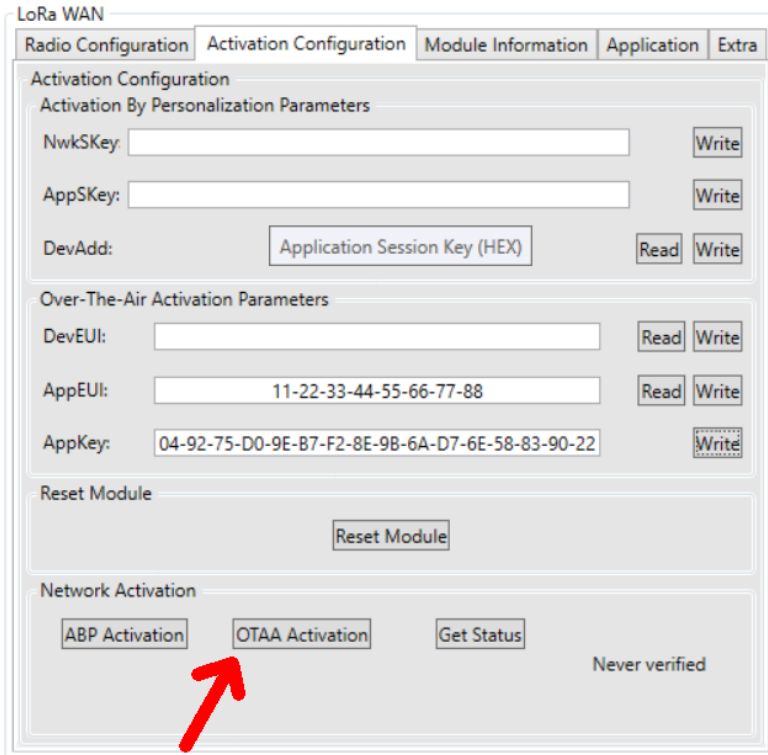
Command example

To enable the public network sync word writing the parameter in the EEPROM:

Host: 0xAA, 0x32, 0x02, 0x2E, 0x01, 0xF3

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

2.7. Join the network



To join the network server, click on “*OTAA Activation*” button in the “*Activation Configuration*” tab.

Command example

To join the network server, use the JOIN_CMD (0x40)

Host: 0xAA, 0x40, 0x01, 0x01, 0x14

Device: 0xAA, 0xC0, 0x01, 0x00, 0x95

On confirmation the module sends the JOIN_IND(0x41)

Device: 0xAA, 0x41, 0x01, 0x00, 0x14

On the end node page from the thing network console, it’s possible to see the join event.

Applications > Module test application > End devices > eui-70b3d5c2000061d8


eui-70b3d5c2000061d8
ID: eui-70b3d5c2000061d8

↑ n/a ↓ n/a • Last activity 14 minutes ago ©

Overview Live data Messaging Location Payload formatters Claiming General settings

General information

End device ID
eui-70b3d5c2000061d8

Description
This end device has no description

Created at
May 20, 2022 10:17:25

Activation information

JoinEUI
11 22 33 44 55 66 77 88

DevEUI
70 B3 D5 C2 00 00 61 D8

Root key ID
n/a

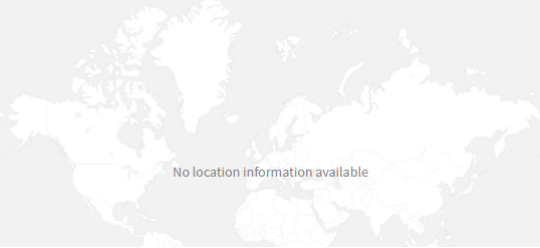
AppKey
.....

NwkKey
n/a

Session information
No data available

Live data See all activity →

12:10:43 Console: Stream reconnected The stream connection has been reconnected
12:10:39 Forward join-accept message
12:10:37 Console: Stream connection closed The connection was closed by the server
12:10:37 Accept join-request
10:17:25 Create end device

Location Change location settings →


Now that the device has joined the network, it's possible to send and receive messages.

3. Sending a confirmed message

LoRa WAN

Radio Configuration Activation Configuration Module Information Application Extra

Application

Port (byte): 100

Data (HEX)
AA-BB-CC-DD-EE-FF

☒ Confirmed Message

Send Message

Success

Use the “*Application*” tab to send a message.

Here it's possible to write the payload, select the port number and whether the message should be confirmed or not.

Command example

Send a message with the TX_MSG_CMD (0x46) with the “reliable data transmission” option.
For example to send the payload “0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF”:

Host: 0xAA, 0x46, 0x08, 0x01, 0x64, **0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF**, 0xA8

Device: 0xAA, 0xC6, 0x01, 0x00, 0x8F

Once the procedure is complete, the module sends TX_MSG_CONFIRMED_IND (0x47) followed by a RX_MSG_IND (0x49):

Device: 0xAA, 0x47, 0x05, 0x00, 0x00, 0x00, 0x01, 0x01, 0x08

Device: 0xAA, 0x49, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0xBC, 0xFF, 0x05,
0x00, 0x40

4. Sending an unconfirmed message

To send an unconfirmed message from the GUI, remove the relative tick.

Command example

Send a message with the TX_MSG_CMD (0x46) with the “unreliable data transmission” option.

For example to send the payload “0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF”:

Host: 0xAA, 0x46, 0x08, 0x00, 0x64, **0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF**, 0xA9

Device: 0xAA, 0xC6, 0x01, 0x00, 0x8F

Once the procedure is complete, the module sends TX_MSG_UNCONFIRMED_IND (0x48) followed by a RX_MSG_IND (0x49):

Device: 0xAA, 0x48, 0x03, 0x00, 0x05, 0x00, 0x06

Device: 0xAA, 0x49, 0x0C, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x00, 0xBA, 0xFF, 0x0E,
0x00, 0x35

5. Receiving a message from the server

In this example the module is configured as a Class A device, so the module will open its receiving windows only after sending a message. If a downlink message has been scheduled, it will be returned in the RX_MSG_IND (0x49)

Command example

For example, the payload “**0x11, 0x22, 0x33**” has been scheduled for downlink.

Host: 0xAA, 0x46, 0x08, 0x00, 0x01, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF, 0x0C

Device: 0xAA, 0xC6, 0x01, 0x00, 0x8F

Device: 0xAA, 0x48, 0x03, 0x00, 0x05, 0x00, 0x06

Device: 0xAA, 0x49, 0x0F, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x01, 0xA5, 0xFF, 0x0B,
0x01, **0x11, 0x22, 0x33**, 0xE2

6. Revision History

Revision	Date	Description
0.1	6.6.2022	First version
0.2	7.10.2022	-Add command examples -Update screenshots to new GUI
0.3	06.12.2022	Change the image on the first page Change the name of the document to “User Guide”
0.4	20.01.2025	Title changed Hardware setup removed