EVALUATION BOARD

Average reading time: 7 minutes.

LoRaWAN® protocol emerged with the popularization of technologies that provide connectivity to meet the market's need for **Internet of Things** (IoT) requirements. First, it is important to understand that Lora and LoraWan are different technologies. **LoRa** is a wireless modulation technique based on **Chirp Spread Spectrum** (CSS) technology that enables long-distance communication with low power consumption.

On the other hand, **LoRaWAN** is a LPWAN (Low Power, Wide Area Network) protocol designed to connect wireless devices, built on LoRa modulation. In the tutorial, we will be using the **SiP HTLRBL32L** microcontroller that comes with a LoRA radio transceiver, aiming to send data over the Lora network.

Tools:

- ❑ **Wise Studio** IDE to compile the code;
- ❑ **Termite** to visualize the board's serial;
- ❑ **RF-flasher** software to write the firmware to the board;
- ❑ Git installer.
- ❑ HTLRBL32L board;
- ❑ FTDi module to connect the board to the computer.

1. LoRaWAN Architecture

Before starting the tutorial it is important to understand how the architecture of the protocol works. **LoRaWAN** network is deployed in a star-star topology, which typically consists of the following elements, as shown in image 1:

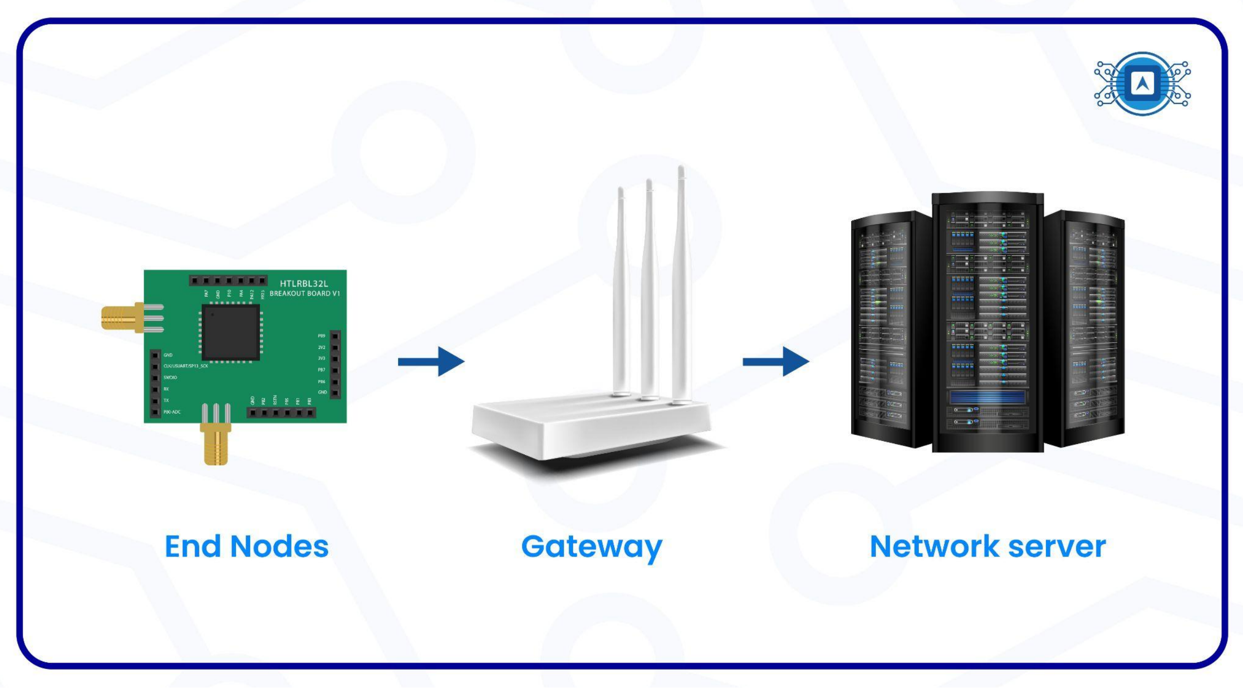


Image 1: LoRaWAN network architecture. Source: *The author*.

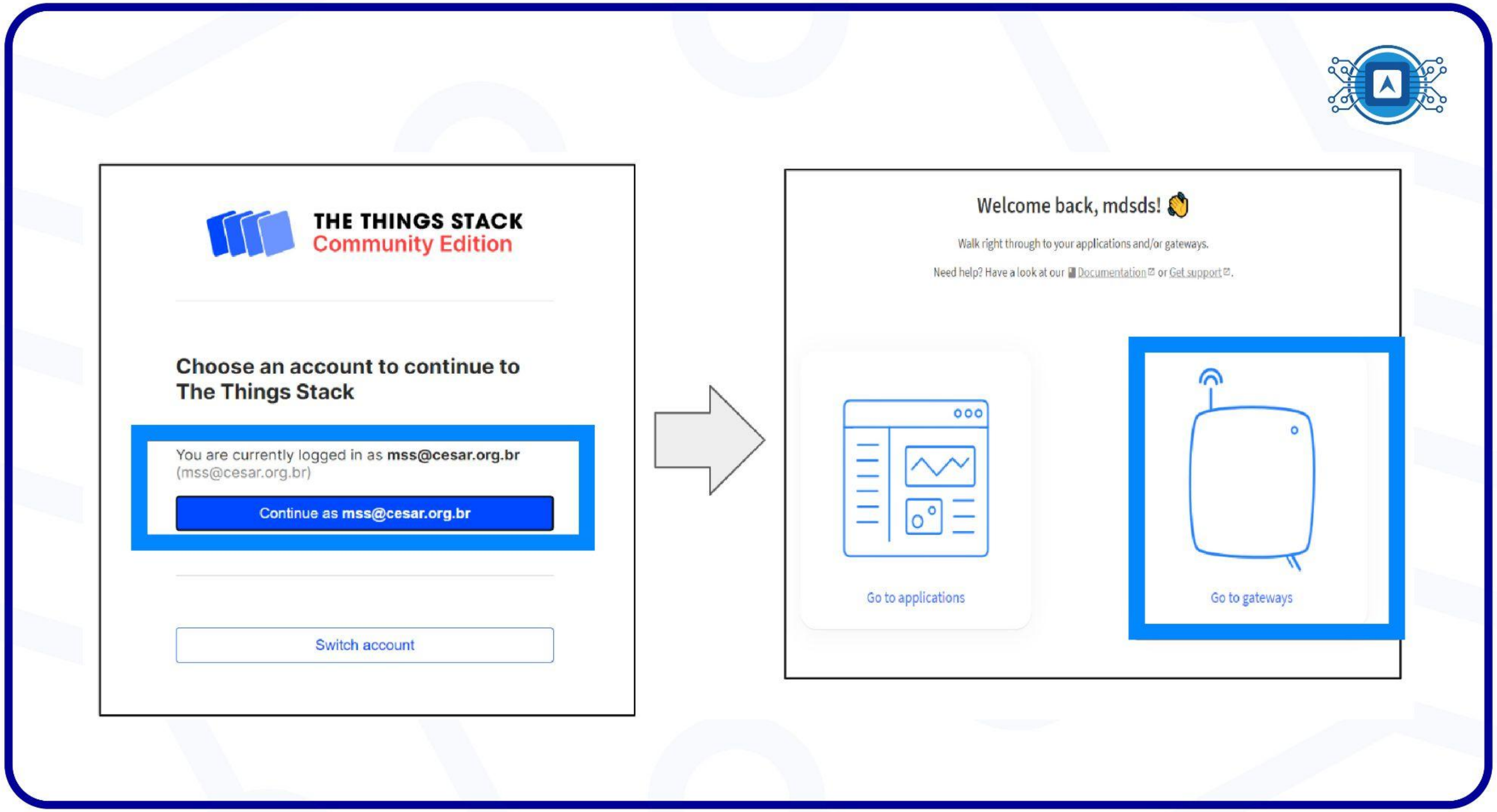
- **End Nodes** - Sensors or actuators send wireless messages with the LoRaWAN protocol to the gateways or receive the messages back from the gateways.
- **Gateways** - They are the receivers of the messages sent by the **End Nodes** and forward these messages to the NetWork server.
- **Network Server** - Manages the information sent by the gateways.

1.1 Gateway

The gateways are connected to the network server through standard IP connections and act as a data bridge between the **End Nodes** and the Internet network, by simply converting RF (Radio Frequency) packets into IP packets and vice versa.

2. Basic gateway configuration

To set up a gateway you will need to create an account on thethingsnetwork platform. After creating the account two options will appear, click on the option “**go to gateways**”, as indicated in image 2.



THE THINGS STACK
Community Edition

Choose an account to continue to The Things Stack

You are currently logged in as **mss@cesar.org.br** (mss@cesar.org.br)

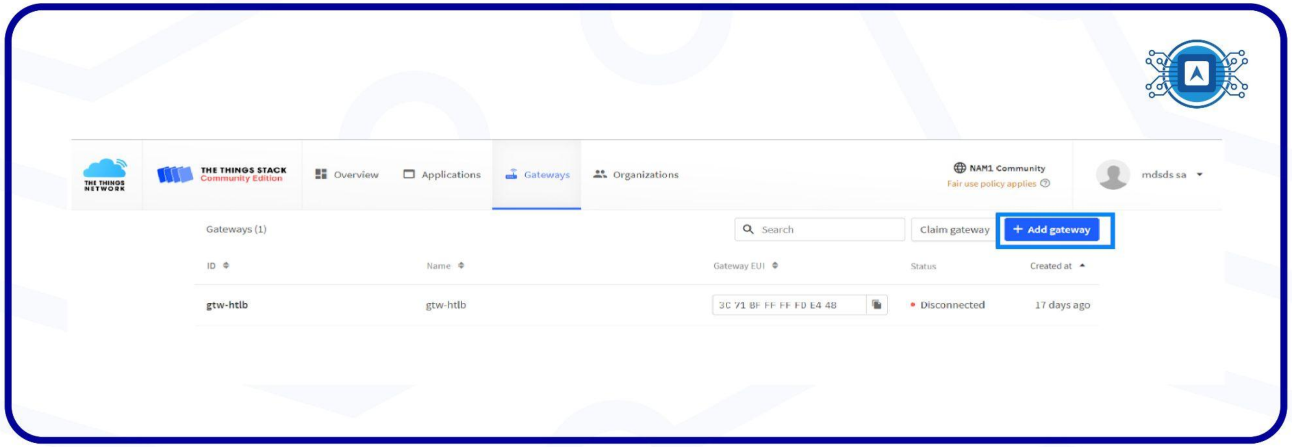
Welcome back, mdsds! 👋

Walk right through to your applications and/or gateways.

Need help? Have a look at our Documentation or Get support.

Image 2: TTN Platform. Source: *The author*.

Then, click on “ **add gateway**”.

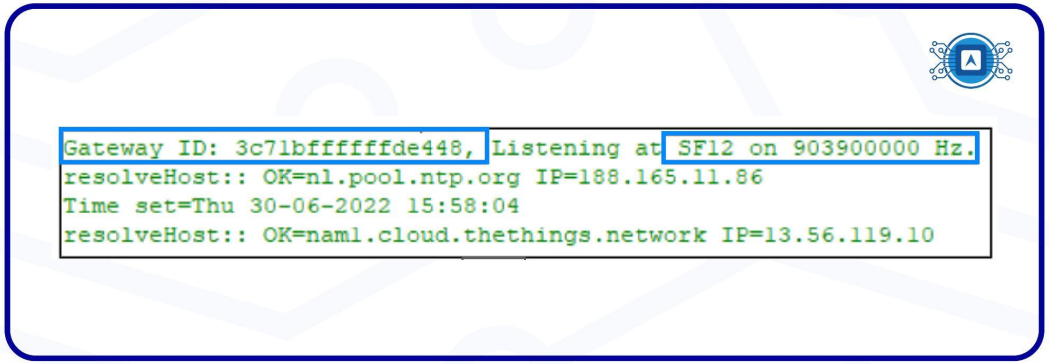


Gateways (1)

ID	Name	Gateway EUI	Status	Created at
gtw-htlb	gtw-htlb	3C 71 BF FF FF FD E4 48	• Disconnected	17 days ago

Image 3: Add Gateway. Source: *The author*.

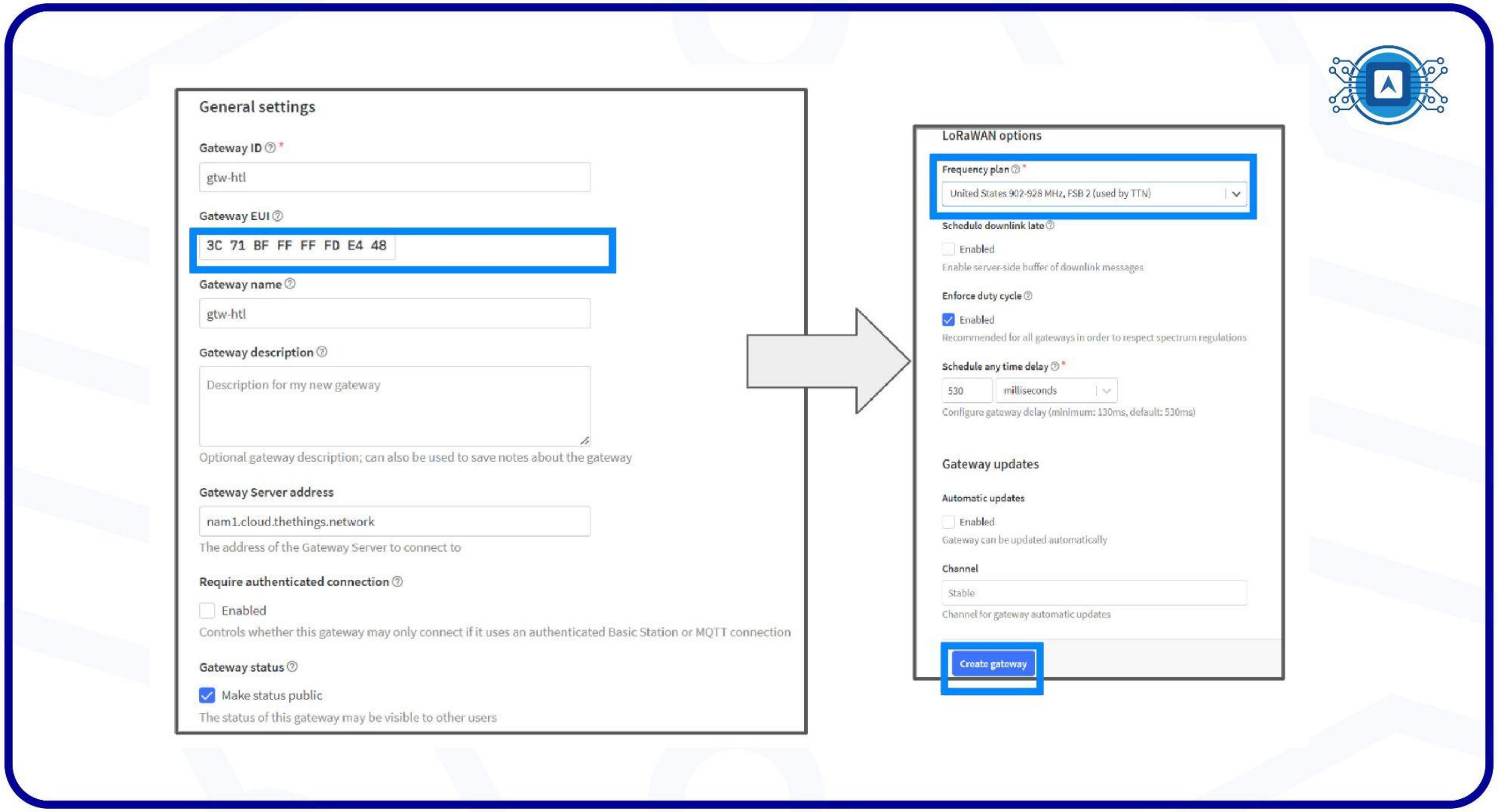
Next, it is necessary to turn on the gateway so as to get the required information to input on to “**the things network”**, such as GatewayID, GatewayEUI and frequency information, according to image 4.



```
Gateway ID: 3c71bfffffde448, Listening at SF12 on 903900000 Hz.
resolveHost:: OK=nl.pool.ntp.org IP=188.165.11.86
Time set=Thu 30-06-2022 15:58:04
resolveHost:: OK=nam1.cloud.thethings.network IP=13.56.119.10
```

Image 4: Gateway Information. Source: *The author*.

Then take the information provided by the gateway and enter it according to image 5. Remember that the regulated frequency range, for this example, in the Brazil region is 902-928Mhz. Because of this, we will add in the "**Frequency plan**" the information "United States 902-928 Mhz". To finish click "**Create gateway**".



General settings

Gateway ID ⓘ *

gtw-htl

Gateway EUI ⓘ

3C 71 BF FF FF FD E4 48

Gateway name ⓘ

gtw-htl

Gateway description ⓘ

Description for my new gateway

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

Gateway Server address

nam1.cloud.thethings.network

The address of the Gateway Server to connect to

Require authenticated connection ⓘ

□ Enabled

Controls whether this gateway may only connect if it uses an authenticated Basic Station or MQTT connection

Gateway status ⓘ

☑ Make status public

The status of this gateway may be visible to other users

LoRaWAN options

Frequency plan ⓘ *

United States 902-928 MHz, FSB 2 (used by TTN)

Schedule downlink late ⓘ

□ Enabled

Enable server-side buffer of downlink messages

Enforce duty cycle ⓘ

☑ Enabled

Recommended for all gateways in order to respect spectrum regulations

Schedule any time delay ⓘ *

530 milliseconds

Configure gateway delay (minimum: 130ms, default: 530ms)

Gateway updates

Automatic updates

□ Enabled

Gateway can be updated automatically

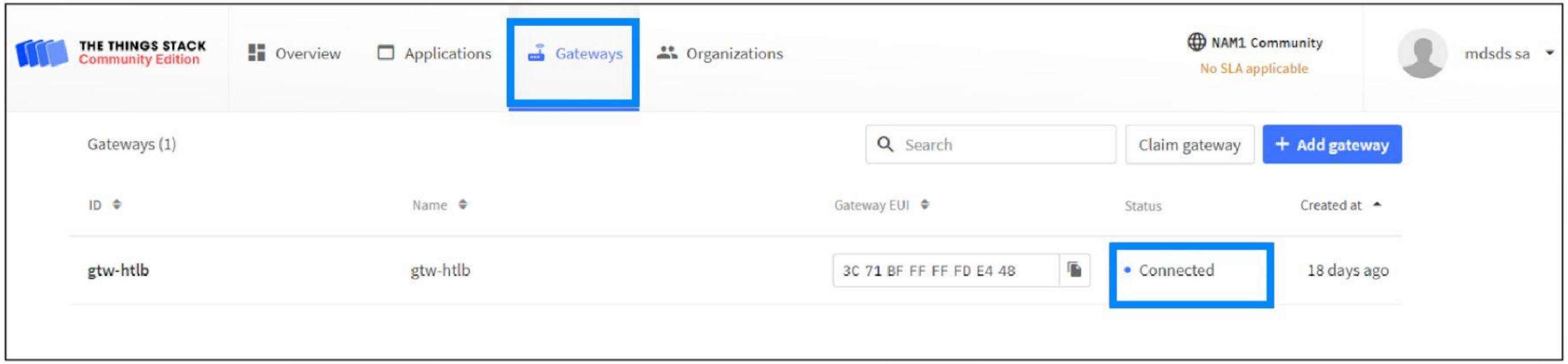
Channel

Stable

Channel for gateway automatic updates

Image 5: Gateway Information. Source: *The author*.

Back to the "**gateway**" tab, we can see the Gateway connected, as can be seen in image 6.



Gateways (1)

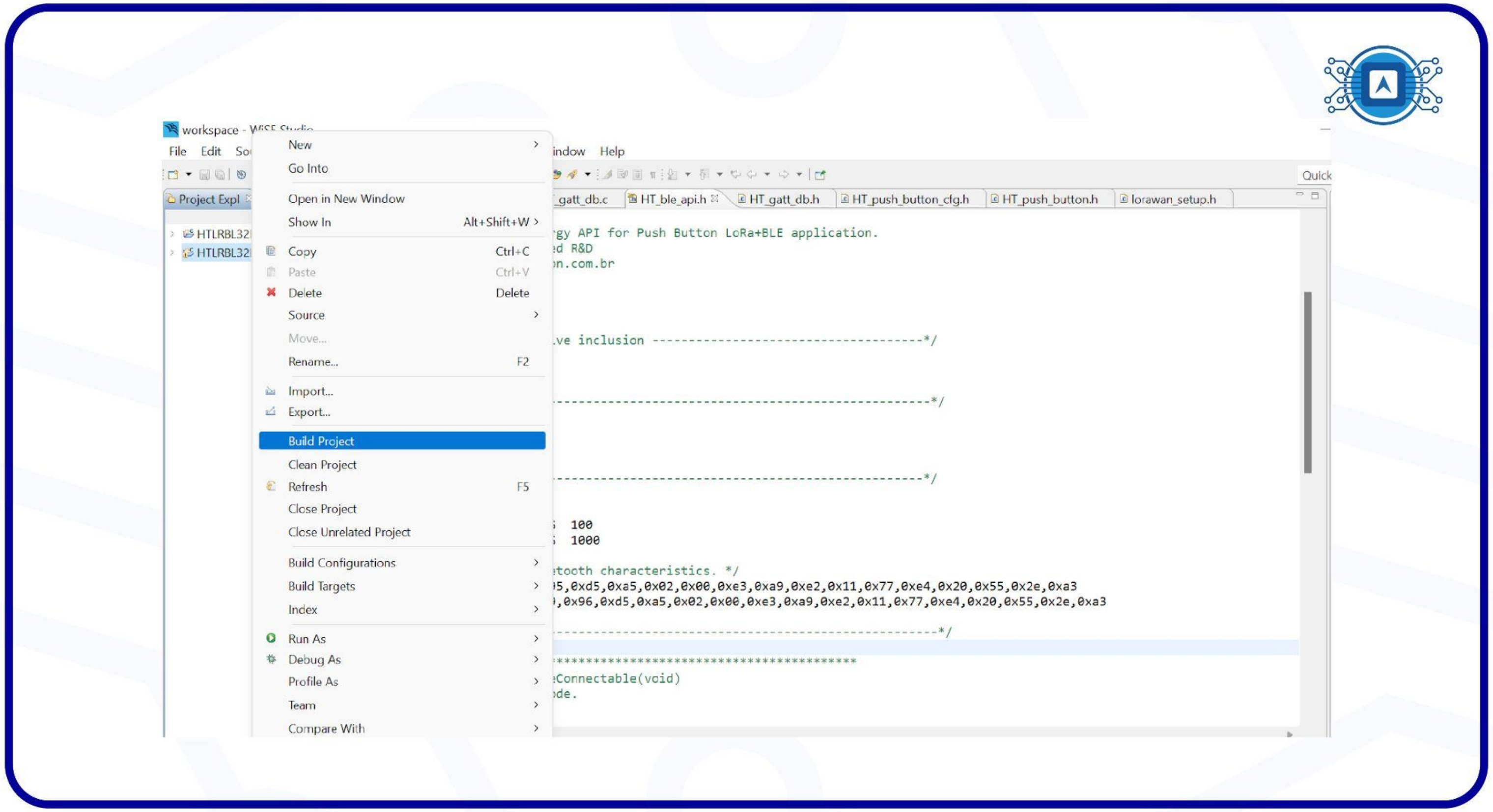
ID	Name	Gateway EUI	Status	Created at
gtw-htlb	gtw-htlb	3C 71 BF FF FF FD E4 48	• Connected	18 days ago

Image 6: Gateway Connected. Source: *The author*.

3. Get the sample code provided on GitHub

To proceed with the next steps, you will need to download or clone the "**LoRaWAN_TagoIO_DashBoard**" project available in this repository. Review the cloning process in the Getting Started with Git material located in the Fundamentals Track. After the cloning procedure, use the Wise Studio **IDE** to compile the code and generate the binary. To perform this procedure, open the "**File**" tab in Wise Studio, click on "**Open projects from file System**". Then, click "**Directory**", select the downloaded or cloned folder and then click **Finish**.

After that, right-click on the project's main folder and click on “**Build Project**”, as seen in image 7. This operation will compile all the code and therefore build the firmware binary.

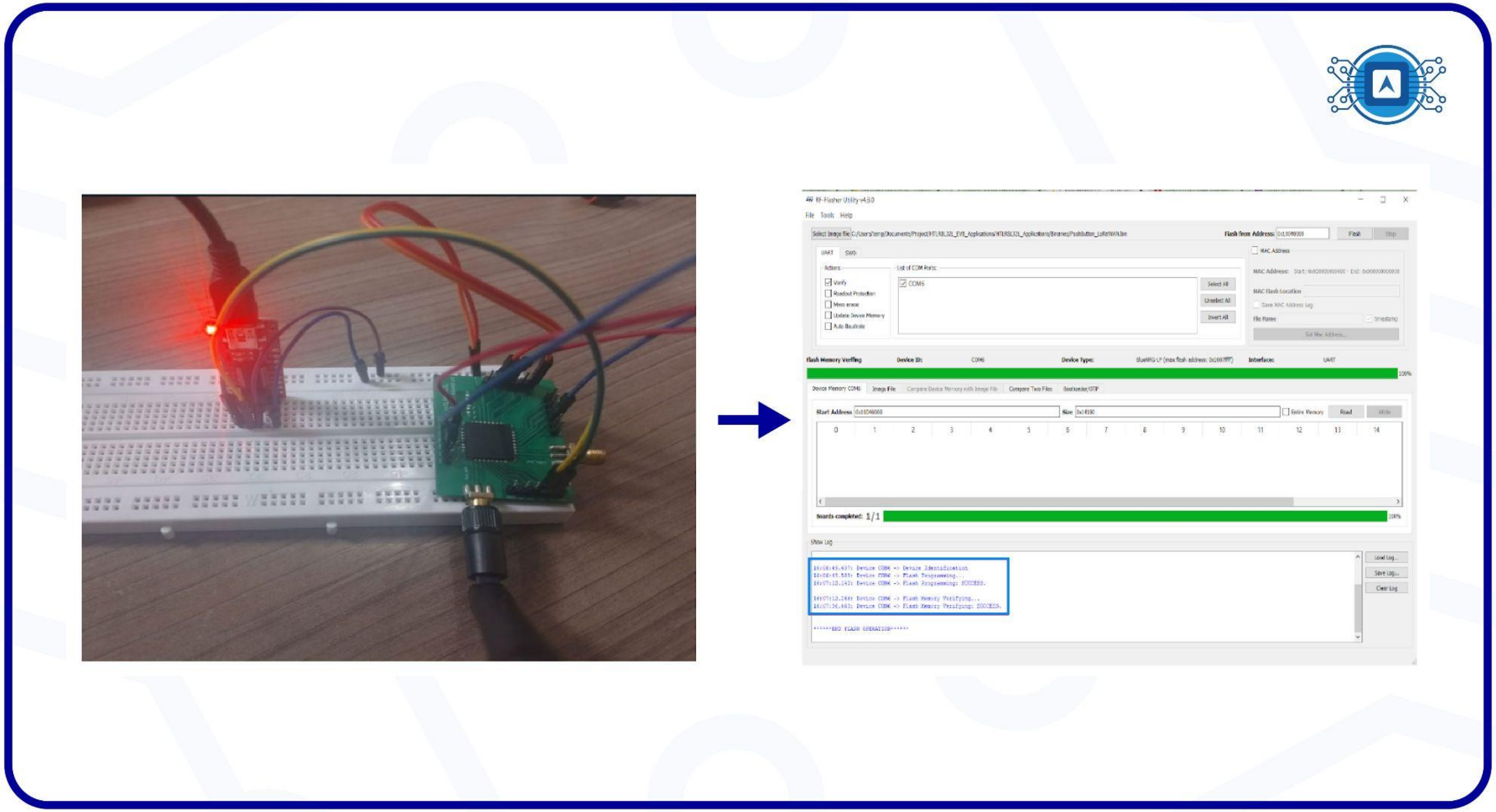


```
rgy API for Push Button LoRa+BLE application.
ed R&D
on.com.br



.ve inclusion -----------------------------------*/


--------------------------------------------------*/



-------------------------------------------------*/


5  100
5  1000

etooth characteristics. */
95,0xd5,0xa5,0x02,0x00,0xe3,0xa9,0xe2,0x11,0x77,0xe4,0x20,0x55,0x2e,0xa3
9,0x96,0xd5,0xa5,0x02,0x00,0xe3,0xa9,0xe2,0x11,0x77,0xe4,0x20,0x55,0x2e,0xa3

---------------------------------------------------*/

*****************************************
eConnectable(void)
ode.
```

Image 7: Build Project. Source: *The author*.

3.1 Writing Firmware to the Board

With the binary created, we will write it using the **RF-flasher** software, as shown in image 8. The procedure to write the firmware using the RF-flasher is in the text Firmware Recording and Running Tests.



Select Image file: C:/Users/temp/Documents/Project/HTLRBL32L_EVB_Applications/HTLRBL32L_Applications/Binaries/PushButton_LoRaWAN.bin

Flash from Address: 0x10040000

Actions: ☑ Verify ☐ Readout Protection ☐ Mass erase ☐ Update Device Memory ☐ Auto Baudrate

List of COM Ports: ☑ COM6

☐ MAC Address

MAC Address: Start: 0x0200000000000 - End: 0x0000000000000

MAC Flash Location

☐ Save MAC Address Log

File Name ☑ timestamp

Flash Memory Verifing Device ID: COM6 Device Type: BlueNRG-LP (max flash address: 0x1007ffff) Interface: UART 100%

Start Address 0x10040000 Size 0x4180 ☐ Entire Memory

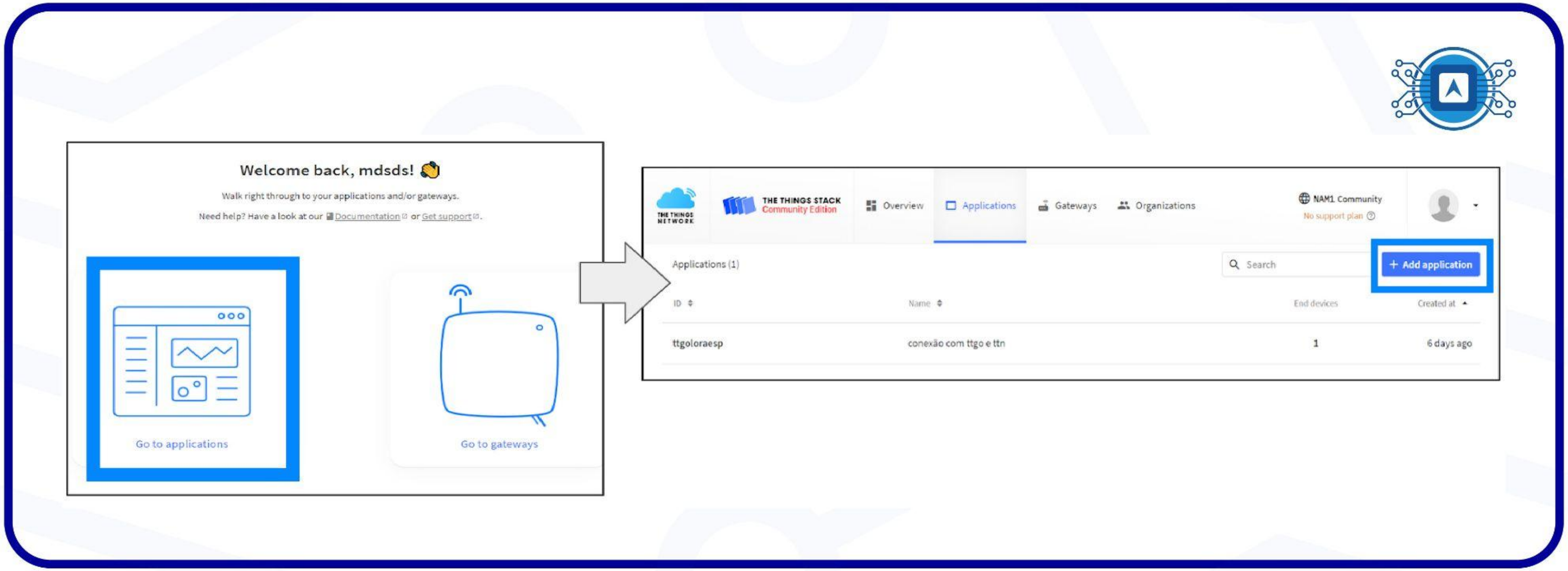
Boards completed: 1/1 100%

```
16:06:49.407: Device COM6 -> Device Identification
16:06:49.585: Device COM6 -> Flash Programming...
16:07:12.142: Device COM6 -> Flash Programming: SUCCESS.

16:07:12.266: Device COM6 -> Flash Memory Verifying...
16:07:36.463: Device COM6 -> Flash Memory Verifying: SUCCESS.

******END FLASH OPERATION******
```

Image 8: Recording the firmware. Source: *The author*.

4. Creating an application on TTN.

In this step, add our “**microcontroller**” with the LoraWan communication protocol. When you return to the home page, click the “**go to Applications**” option and then click “**Add application**”. As seen in image 9.



Welcome back, mdsds!

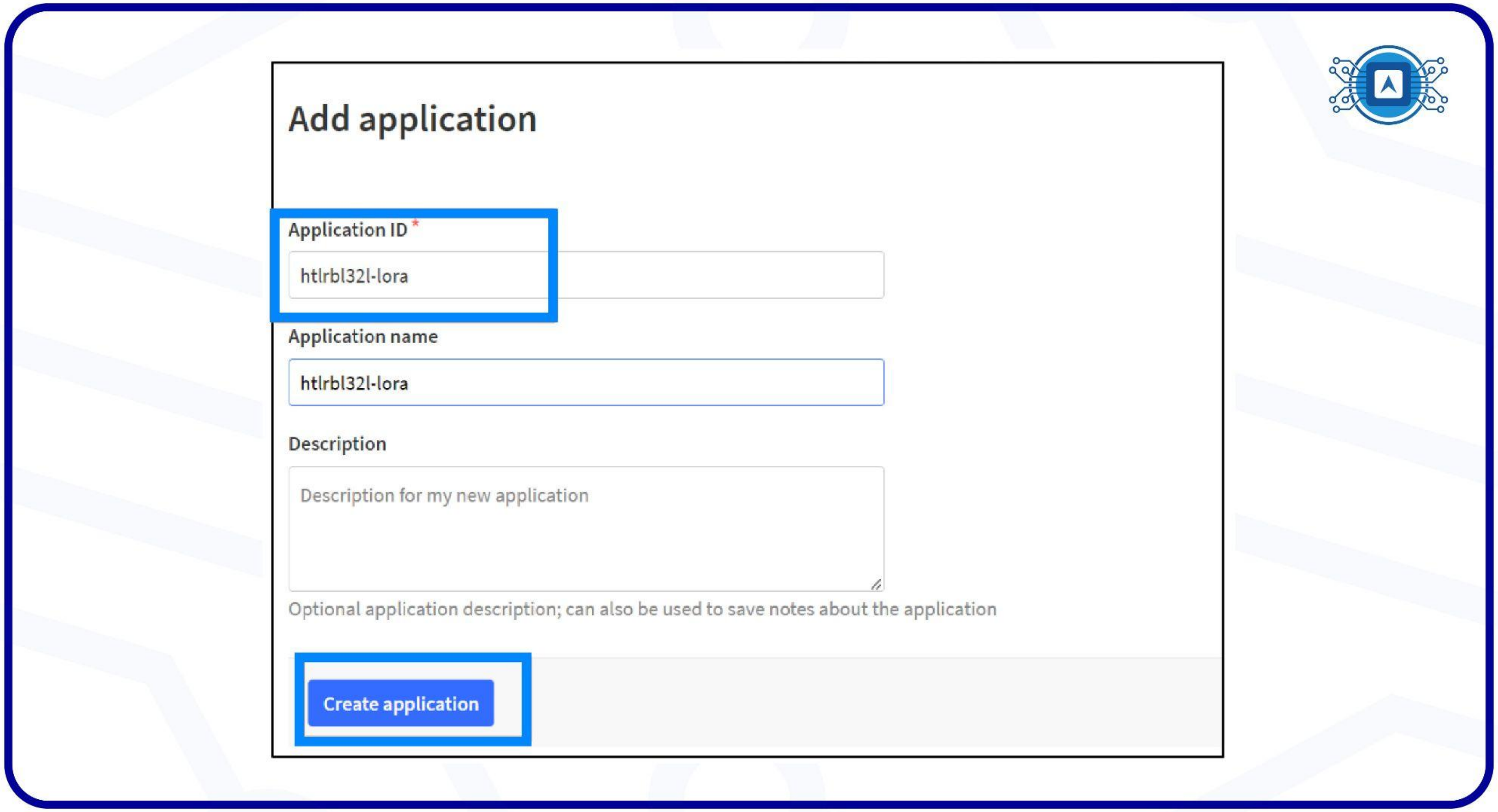
Walk right through to your applications and/or gateways.

Need help? Have a look at our Documentation or Get support.

ID	Name	End devices	Created at
ttgoloraesp	conexão com ttgo e ttn	1	6 days ago

Image 9: Add application. Source: *The author*.

On the next screen give a name to your application (**microcontroller**), which in the case of image 10 is named "**htlrbl32l-lora**", then click "**Create application**".



Add application

Application ID *

htlrbl32l-lora

Application name

htlrbl32l-lora

Description

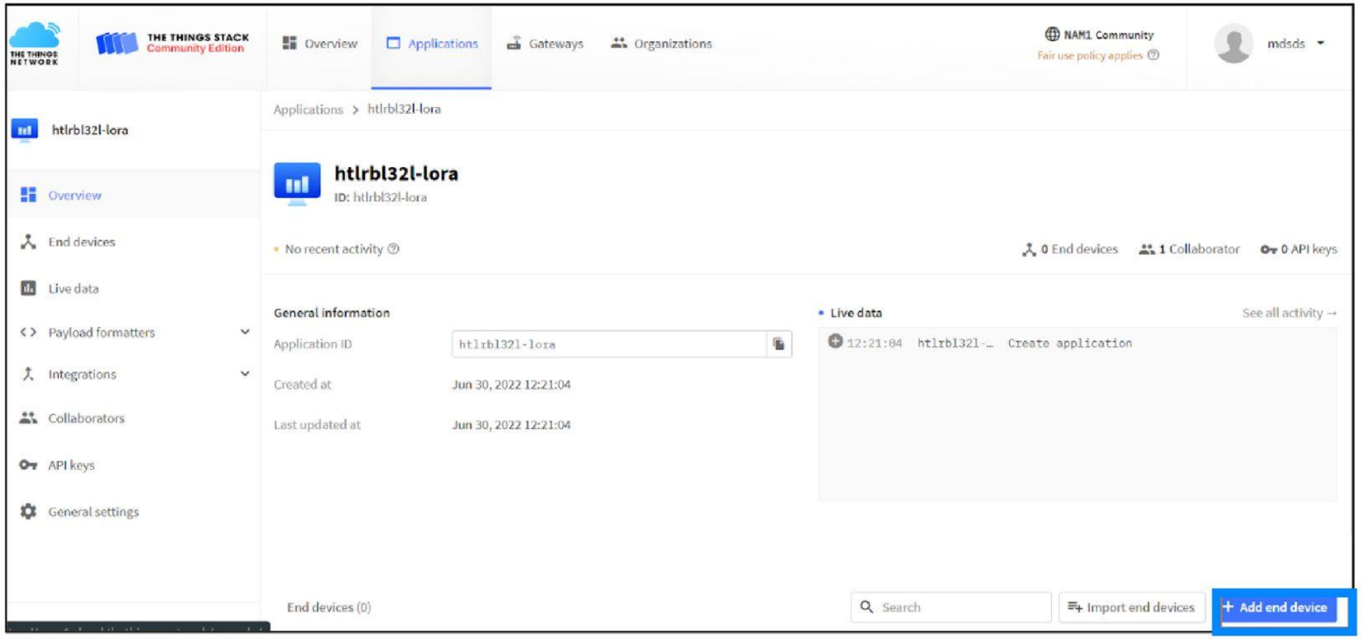
Description for my new application

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

Image 10: Create application. Source: *The author*.

5. Creating a device.

On the next screen, click on the “**Applications**” created and add a **device**. To do this, click on “**Add end device**”, as shown in image 11.



General information

Application ID htlrbl321-lora

Created at Jun 30, 2022 12:21:04

Last updated at Jun 30, 2022 12:21:04

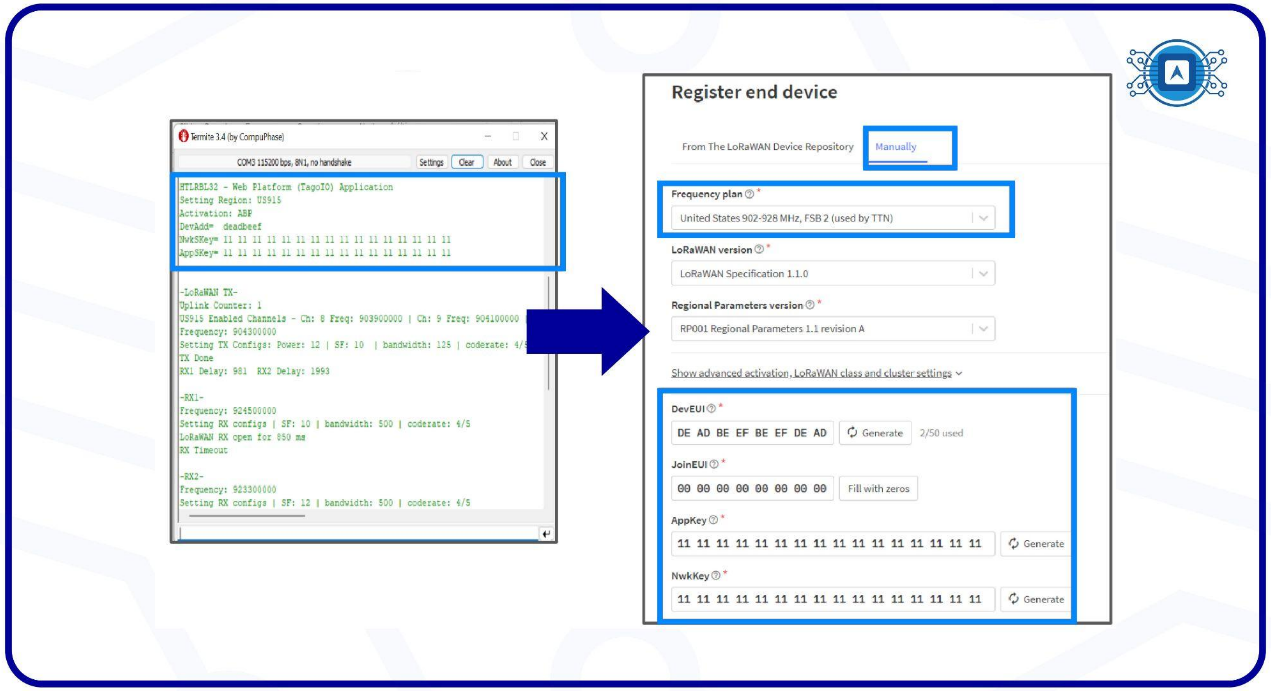
Live data — See all activity →

12:21:04 htlrbl321-… Create application

Image 11: Add end device. Source: *The author*.

5.1 Security key set.

Click “**Manually**” to add the security keys. LoRaWAN 1.0 specifies several security keys: **NwkSKey, AppSKey and AppKey**. All keys have a length of 128 bits, similar to the algorithm used in the wifi standard 802.15.4. The **AppKey** application key is only known to the device and the application. While the **NwkSKey** is shared with the network, the **AppSKey** is kept private. However, these keys are provided from the microcontroller’s source code to **The things network**. To view this information coming from the microcontroller, we are going to need the help of the “termite” software. When you open the thermite and reset the board, you will get the region and key information. After adding all the keys, finish the procedure by clicking on “**Register end device**”.



Termite 3.4 (by CompuPhase)
COM3 115200 bps, 8N1, no handshake
Settings
Clear
About
Close

```
HTLRBL32 - Web Platform (TagoIO) Application
Setting Region: US915
Activation: ABP
DevAdd=  deadbeef
NwkSKey= 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
AppSKey= 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11

-LoRaWAN TX-
Uplink Counter: 1
US915 Enabled Channels - Ch: 8 Freq: 903900000 | Ch: 9 Freq: 904100000
Frequency: 904300000
Setting TX Configs: Power: 12 | SF: 10  | bandwidth: 125 | coderate: 4/5
TX Done
RX1 Delay: 981  RX2 Delay: 1993

-RX1-
Frequency: 924500000
Setting RX configs | SF: 10 | bandwidth: 500 | coderate: 4/5
LoRaWAN RX open for 850 ms
RX Timeout

-RX2-
Frequency: 923300000
Setting RX configs | SF: 12 | bandwidth: 500 | coderate: 4/5
```

Register end device

Frequency plan *: United States 902-928 MHz, FSB 2 (used by TTN)

LoRaWAN version *: LoRaWAN Specification 1.1.0

Regional Parameters version *: RP001 Regional Parameters 1.1 revision A

Show advanced activation, LoRaWAN class and cluster settings

DevEUI *: DE AD BE EF BE EF DE AD

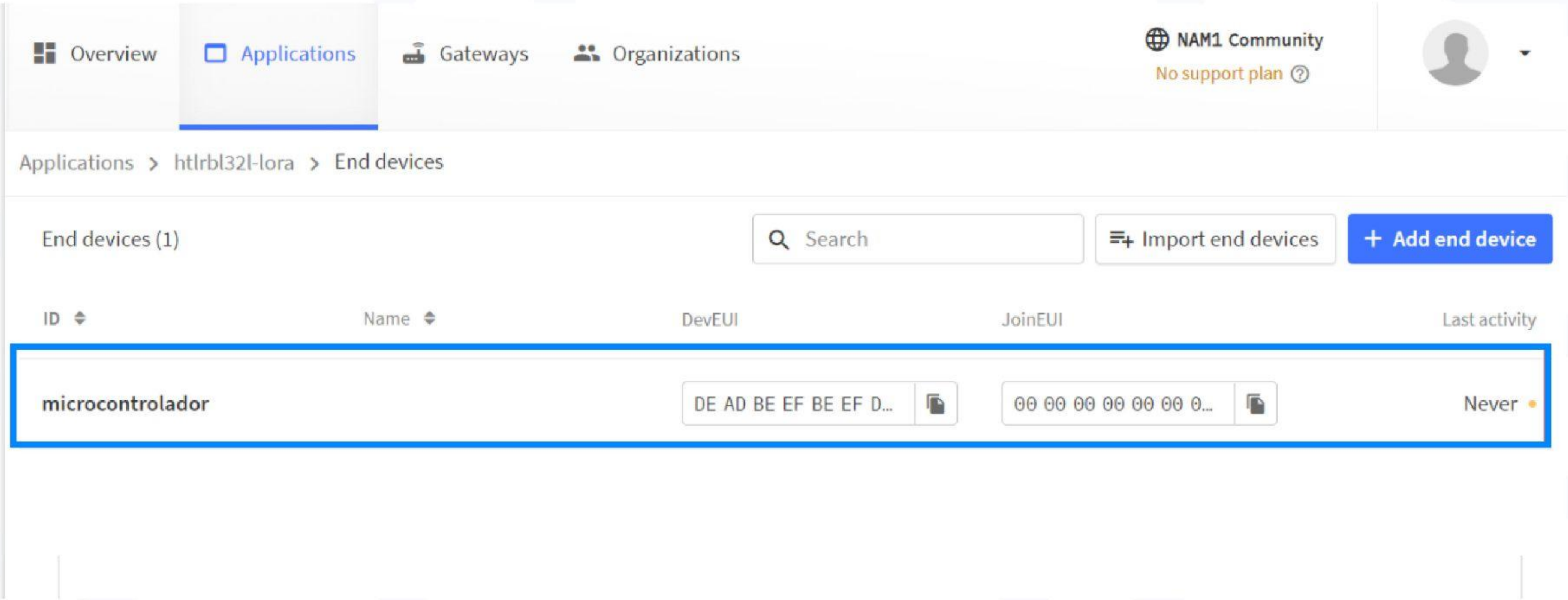
JoinEUI *: 00 00 00 00 00 00 00 00

AppKey *: 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11

NwkKey *: 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11

Image 12: Security Key Configuration. Source: *The author*.

Once the security key has been set up, you can see, as highlighted in image 13, the **device** created.



Applications > htlrbl32l-lora > End devices

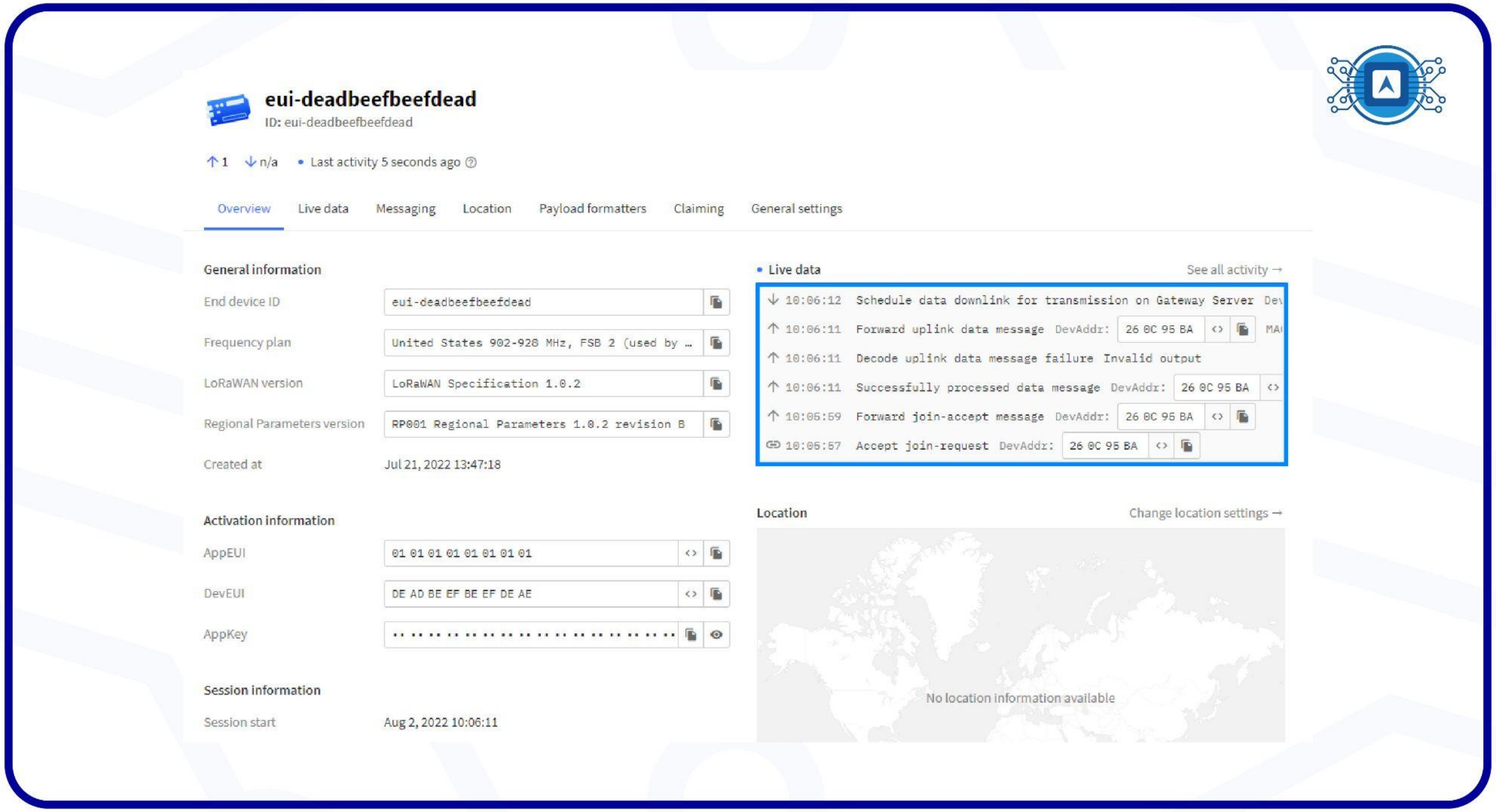
End devices (1)

ID	Name	DevEUI	JoinEUI	Last activity
microcontrolador		DE AD BE EF BE EF D...	00 00 00 00 00 00 0...	Never

Image 13: Add end device. Source: *The author*.

5. Sending LoRa data to the gateway

Since the gateway acts as a data bridge between the “End nodes” and the Internet network, as image 14 highlights, the **GateWay** sends the data received from the LoRa communication to the LoRaWan network through an Internet connection.



eui-deadbeefbeefdead

ID: eui-deadbeefbeefdead

↑ 1 ↓ n/a • Last activity 5 seconds ago

General information

End device ID: eui-deadbeefbeefdead

Frequency plan: United States 902-928 MHz, FSB 2 (used by …

LoRaWAN version: LoRaWAN Specification 1.0.2

Regional Parameters version: RP001 Regional Parameters 1.0.2 revision B

Created at: Jul 21, 2022 13:47:18

Activation information

AppEUI: 01 01 01 01 01 01 01 01

DevEUI: DE AD BE EF BE EF DE AE

AppKey: •• •• •• •• •• •• •• •• •• •• •• •• •• •• •• ••

Session information

Session start: Aug 2, 2022 10:06:11

Live data

```
↓ 10:06:12  Schedule data downlink for transmission on Gateway Server  Dev
↑ 10:06:11  Forward uplink data message  DevAddr: 26 0C 95 BA  MA
↑ 10:06:11  Decode uplink data message failure  Invalid output
↑ 10:06:11  Successfully processed data message  DevAddr: 26 0C 95 BA
↑ 10:05:59  Forward join-accept message  DevAddr: 26 0C 95 BA
  10:05:57  Accept join-request  DevAddr: 26 0C 95 BA
```

Location

Image 14: transmission test. Source: *The author*.

References

AWS. **What is LoRaWAN?.** 2022**.** Available at: **<** https://docs.aws.amazon.com/iot/latest/developerguide/connect-iot-lorawan-what-is-lorawan.html **>.** Accessed on july 08th 2022.

GIT. **Git.** 2022**.** Available at: **<** https://git-scm.com/**>.** Accessed on july 28th 2022.

GIT-HUB. **Git guides.** 2022**.** Available at: **<** https://github.com/git-guides **>** Accessed on july 28th 2022.

THETHINGSNETWORK. **LoRaWAN Architecture**. Available at: **<** https://www.thethingsnetwork.org/docs/lorawan/architecture/ **>** Accessed on july 08th 2022.