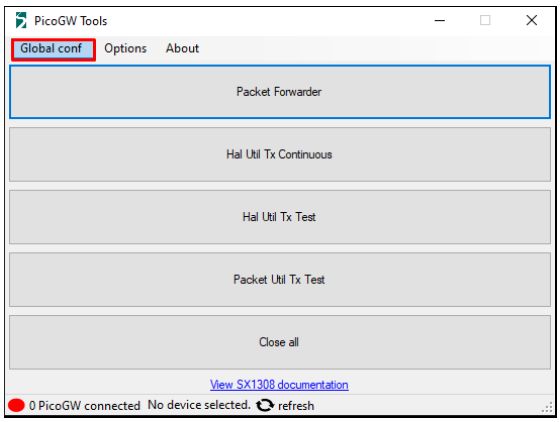
Gateway, Network Server, and Application Configuration

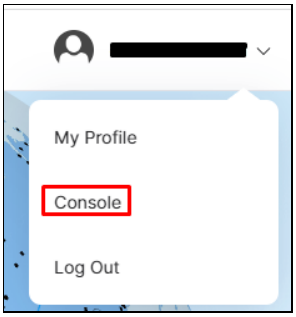
This document will teach you how to configure your gateway, network server, and application to run any LoRaWAN firmware example using the HTLRBL32L SiP. The gateway mentioned in this document is Semtech's Picocell, but the example will work with any other. The example uses two different network servers, TTN (The Things Network) and LoRaCloud, which integrates Cayenne for easy data display.

Picocell Gateway Configuration (TTN)

1. Install the 'PicoGW_UI' software and open it. **WARNING: Do not install PicoGW_UI with your gateway device connected to the computer.**
2. Access "Global conf" screen, tick the "Use custom gateway ID" checkbox and copy your gateway ID.

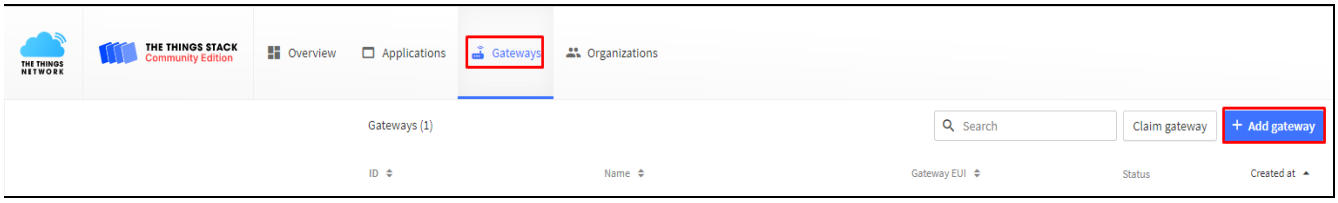


3. Access your The Things Network account, click on your user ID on the top right and access the console.

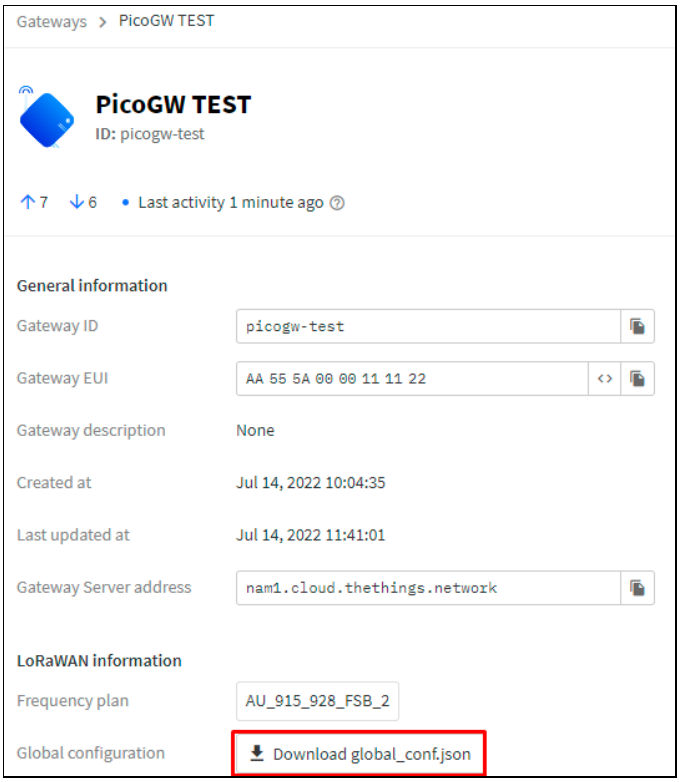


4. Select your cluster of preference (eg. North America cluster (nam1)).

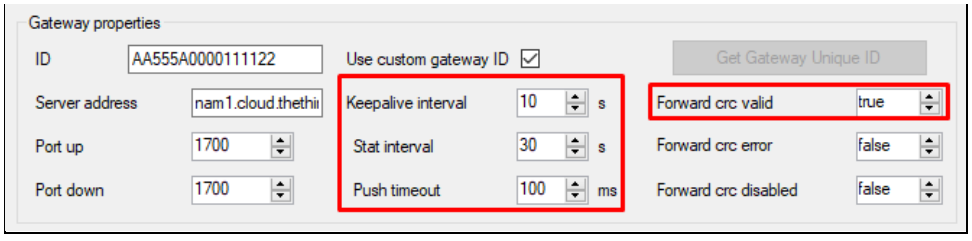
5. Access gateways screen and click on “add gateway”, make a unique ID and a name for your gateway, then paste the gateway ID from step 2 into the “Gateway EUI” field. Select your area’s frequency plan (eg. AU915-928 FSB2).



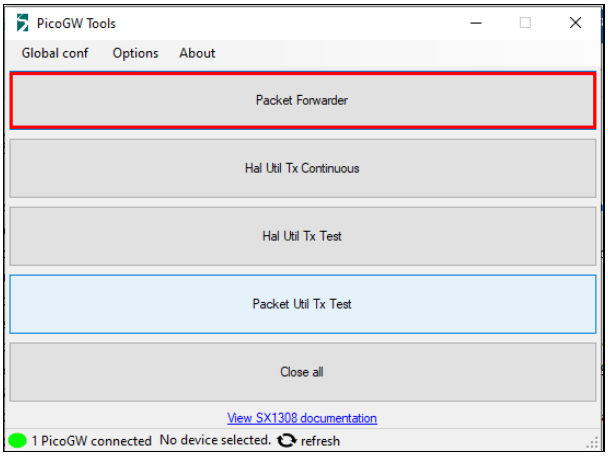
6. Confirm the gateway creation, then access your gateway’s page and click on “download global_conf.json”.

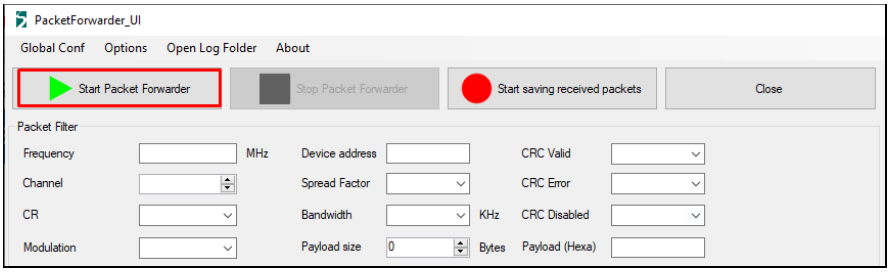


7. Paste the file downloaded in step 6 to “C:\Users*user*\AppData\Roaming\Semtech\PicoGW_UI” and overwrite the existing file.
8. Open the PicoGW_UI ‘global conf’ screen and alter the values of the following fields: ‘Keepalive interval’ to 10s, ‘Stat interval’ to 30, ‘Push timeout’ to 100ms and ‘forward crc valid’ to true.

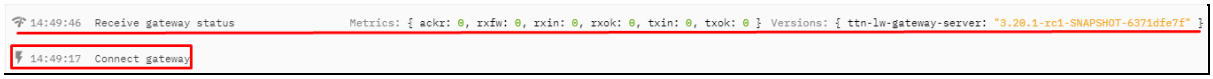


9. Connect your gateway device to your computer and click on the “packet forwarder” button, then click on “start packet forwarder”.





10. Check the TTN’s gateway live data tab to see if the gateway communicates with TTN.



End Node Configuration (TTN)

1. Open the project’s LoRaWAN configuration file (lorawandefines.h).
2. Configure keys to be downloaded onto the device (in hexadecimal) as follows:

LORAWAN_DEVICE_EUI { 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX }

LORAWAN_JOIN_EUI { 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX }

LORAWAN_NWK_KEY { 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX }

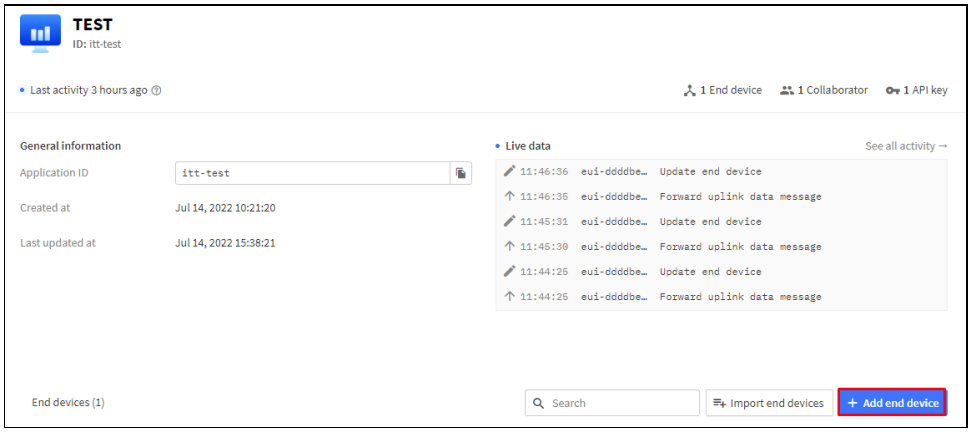
```
lorawandefines.h*
74
75  #define LORAWAN_DEVICE_EUI                              {0xDD,0xDD,0xBE,0xEF,0xBE,0xCF,0xEE,0xEB}
76
77  #endif
78  // { IEEE_OUI, 0x01, 0x01, 0x01, 0x01, 0x01 }
79
80  /*!
81   * App/Join server IEEE EUI (big endian)
82   */
83  #define LORAWAN_JOIN_EUI                                { 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01 }
84
85  /*!
86   * Application root key
87   */
88  #define LORAWAN_APP_KEY                                 { 0x12, 0x34, 0x56, 0x78, 0x90, 0xAB, 0xCD, 0xEF,0x12, 0x34, 0x56, 0x78, 0x90, 0xAB, 0xCD, 0xEF }
89
90  /*!
91   * Network root key
92   * WARNING: FOR 1.0.x DEVICES IT IS THE \ref LORAWAN_APP_KEY
93   */
94  #define LORAWAN_NWK_KEY                                  { 0x12, 0x34, 0x56, 0x78, 0x90, 0xAB, 0xCD, 0xEF,0x12, 0x34, 0x56, 0x78, 0x90, 0xAB, 0xCD, 0xEF }
95
```

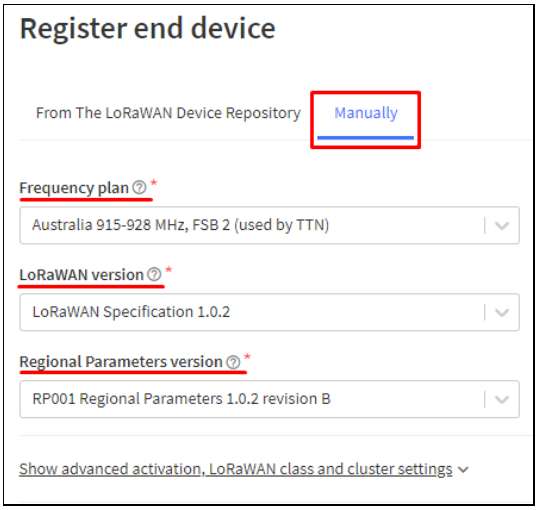
Obs.: Depending on your LoRaWAN version, the keys are given different names. For 1.0.x, the Application Key is defined in LORAWAN_NWK_KEY. For 1.1.x, LORAWAN_NWK_KEY defines the Network Key and LORAWAN_APP_KEY defines the Application Key. In this example we are using LoRaWAN 1.0.2.

```
*******************************************************************************
********************************** WARNING ************************************
*******************************************************************************
 The crypto-element implementation supports both 1.0.x and 1.1.x LoRaWAN
 versions of the specification.
 Thus it has been decided to use the 1.1.x keys and EUI name definitions.
 The below table shows the names equivalence between versions:
              +--------------------+-------------------------+
              |       1.0.x        |          1.1.x          |
              +====================+=========================+
              | LORAWAN_DEVICE_EUI | LORAWAN_DEVICE_EUI      |
              +--------------------+-------------------------+
              | LORAWAN_APP_EUI    | LORAWAN_JOIN_EUI        |
              +--------------------+-------------------------+
              | N/A                | LORAWAN_APP_KEY         |
              +--------------------+-------------------------+
              | LORAWAN_APP_KEY    | LORAWAN_NWK_KEY         |
              +--------------------+-------------------------+
              | LORAWAN_NWK_S_KEY  | LORAWAN_F_NWK_S_INT_KEY |
              +--------------------+-------------------------+
              | LORAWAN_NWK_S_KEY  | LORAWAN_S_NWK_S_INT_KEY |
              +--------------------+-------------------------+
              | LORAWAN_NWK_S_KEY  | LORAWAN_NWK_S_ENC_KEY   |
              +--------------------+-------------------------+
              | LORAWAN_APP_S_KEY  | LORAWAN_APP_S_KEY       |
              +--------------------+-------------------------+
*******************************************************************************
```

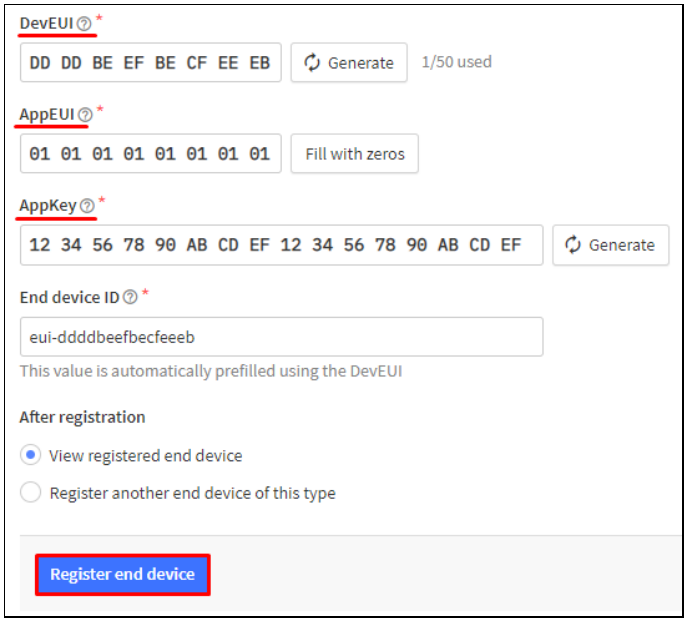
3. Compile project and download it to the end node device.
4. Access the TTN console, open the Applications tab and click on “Add Application”.

5. Access your application’s page and click on “Add End Device”, then select your device from the list. If your device is not listed in the TTN’s device repository, you may manually insert its specifications in the “Manually” tab (to find what these specifications are, you must verify your device’s datasheet/user guide). Ex: select the US902-928 FSB2 frequency plan, select the LoRaWAN version 1.0.2, select the RP001 Regional Parameters 1.0.2 revision B regional parameter version.

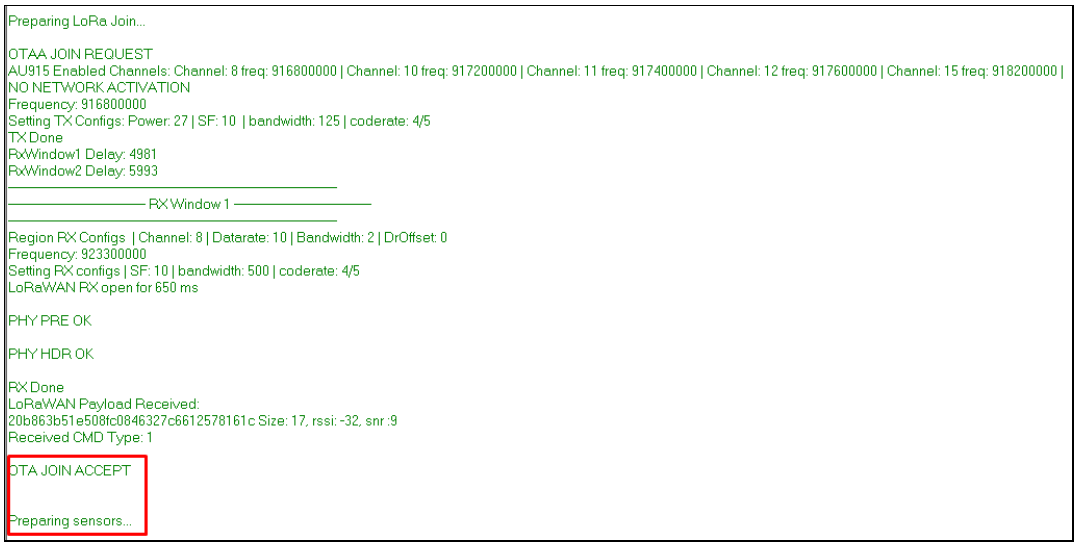


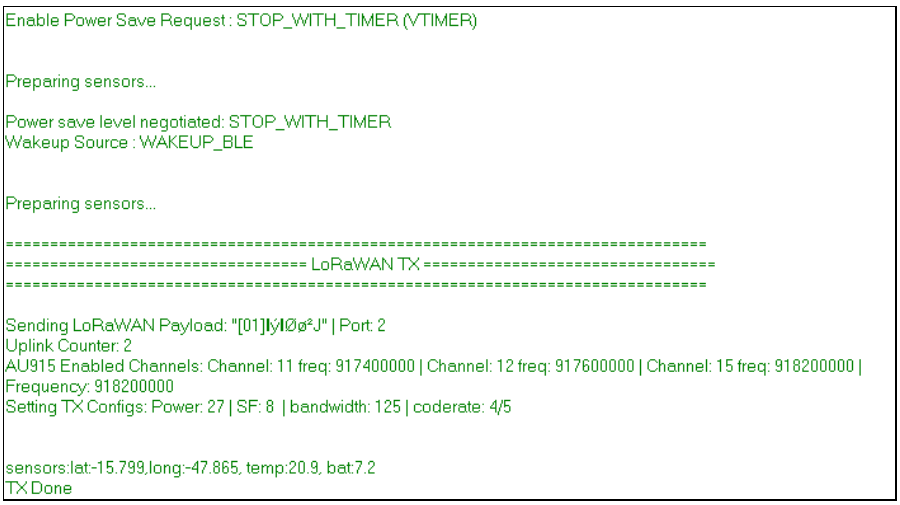


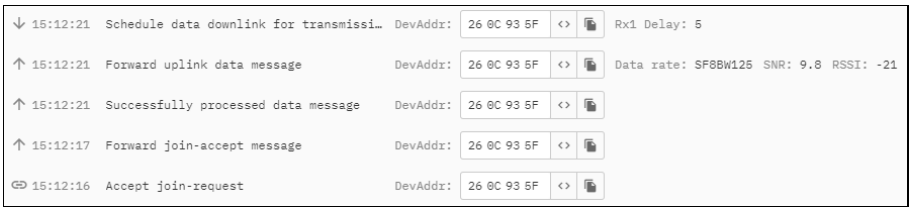
6. Insert in the DevEUI field the device key (LORAWAN_DEVICE_EUI), do the same with the AppEUI (LORAWAN_JOIN_EUI) and AppKey (as per the example, since we are using LoRaWAN version 1.0.2, then we will insert LORAWAN_NWK_KEY) fields.



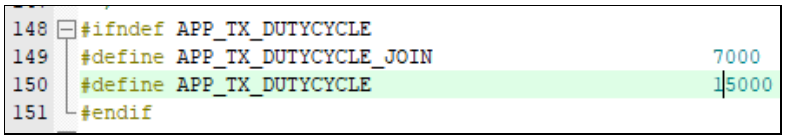
7. Connect the gateway to your computer, start the packet forwarder, connect your end node to the power supply (i.e. the USB port) and wait for it to connect to the LoRaWAN. You can monitor the serial port of the end node to check this. Once the device is connected, wait for it to send data to the TTN (device's "live data" tab).





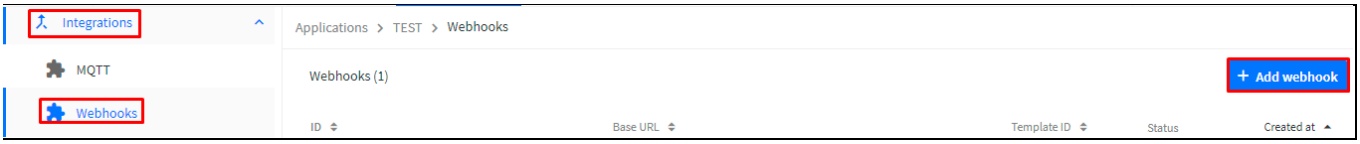


Obs.: The 'LORAMAC_STATUS_BUSY' error is caused by an overlap in the Tx and Rx times in the end node. Adjusting the 'APP_TX_DUTYCYCLE' value (in lorawandefines.h) from 5000 (5s) to 15000 (15s), for example, should correct this.

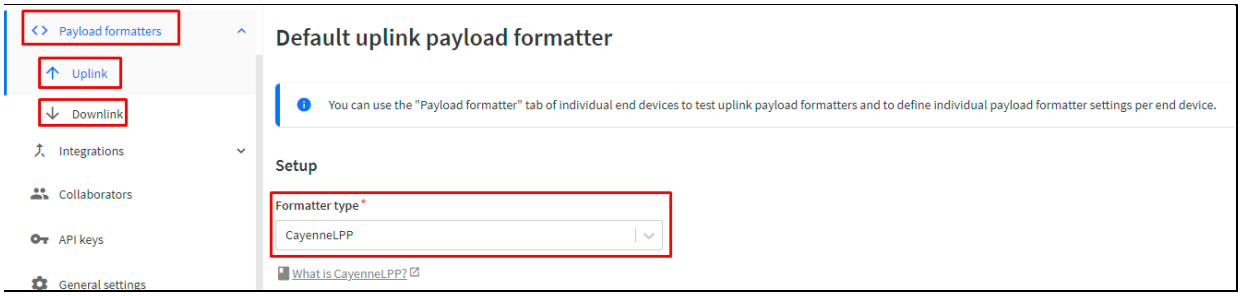


Cayenne Integration (TTN)

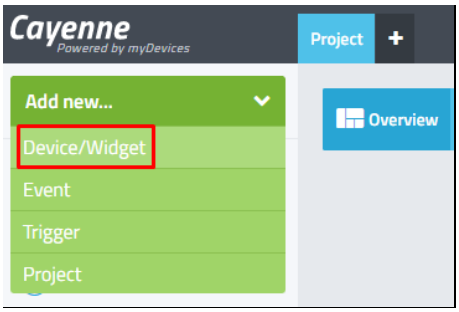
1. Open the application page in TTN. Go into “integrations”, “webhooks” then add a new webhook. Select Cayenne and create an ID for the webhook.

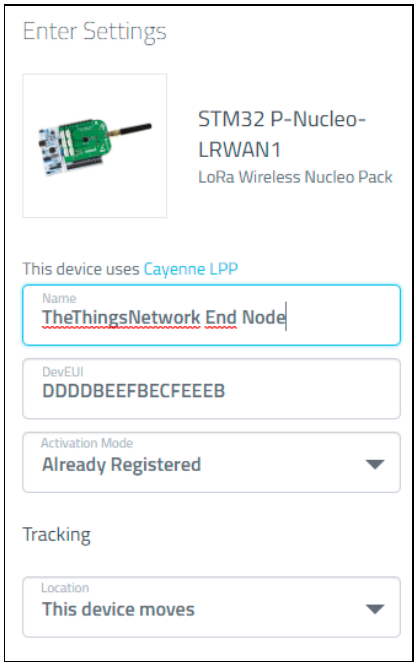


2. In the “Payload Formatters” tab, select CayenneLPP for both downstream and upstream.

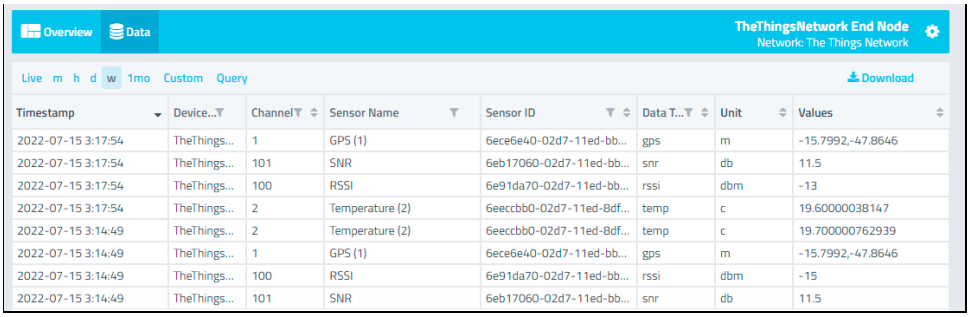


3. Access the Cayenne dashboard (https://mydevices.com/ -> Login -> Cayenne).
4. In your Cayenne dashboard, click on “Add new...” then select ‘Device/Widget’. Expand the LoRa device list, then open the TTN list. Select your device (ex.: STM32 P-Nucleo-LRWAN1), then insert its DevEUI into the field.





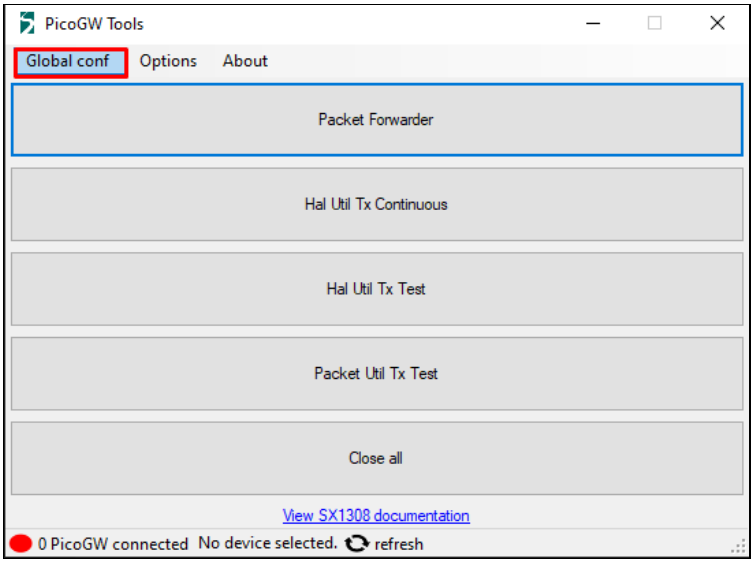
5. Connect the end device to the LoRaWAN through the gateway and check if Cayenne is receiving data from the device.

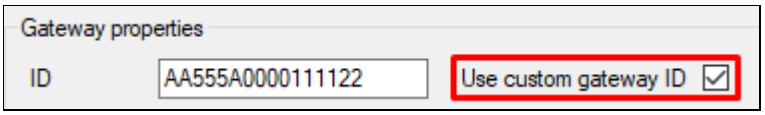


Timestamp	Device...	Channel	Sensor Name	Sensor ID	Data T...	Unit	Values
2022-07-15 3:17:54	TheThings...	1	GPS (1)	6ece6e40-02d7-11ed-bb...	gps	m	-15.7992,-47.8646
2022-07-15 3:17:54	TheThings...	101	SNR	6eb17060-02d7-11ed-bb...	snr	db	11.5
2022-07-15 3:17:54	TheThings...	100	RSSI	6e91da70-02d7-11ed-bb...	rssi	dbm	-13
2022-07-15 3:17:54	TheThings...	2	Temperature (2)	6eeccbb0-02d7-11ed-8df...	temp	c	19.60000038147
2022-07-15 3:14:49	TheThings...	2	Temperature (2)	6eeccbb0-02d7-11ed-8df...	temp	c	19.700000762939
2022-07-15 3:14:49	TheThings...	1	GPS (1)	6ece6e40-02d7-11ed-bb...	gps	m	-15.7992,-47.8646
2022-07-15 3:14:49	TheThings...	100	RSSI	6e91da70-02d7-11ed-bb...	rssi	dbm	-15
2022-07-15 3:14:49	TheThings...	101	SNR	6eb17060-02d7-11ed-bb...	snr	db	11.5

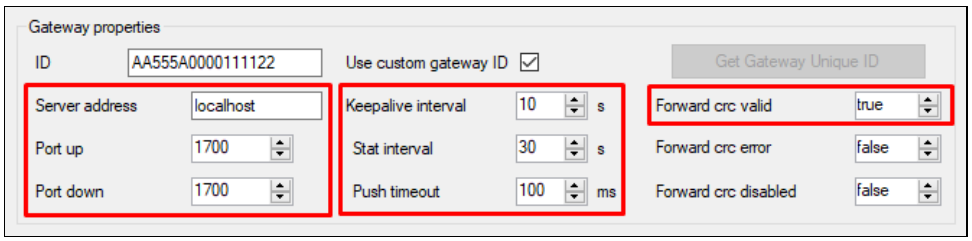
Picocell Gateway Configuration (LoRaCloud)

1. Install the 'PicoGW_UI' software and open it. **WARNING: Do not install PicoGW_UI with your gateway device connected to the computer.**
2. Access "Global conf" screen, tick the "Use custom gateway ID" checkbox and copy your gateway ID.

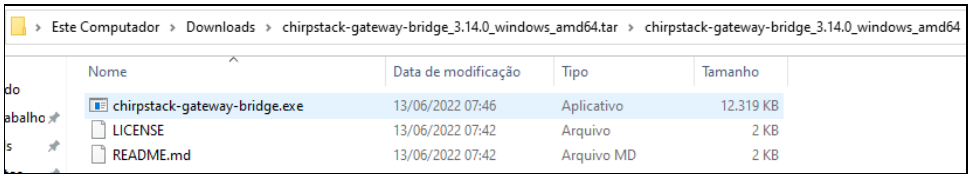




3. Change the "server address" to 'localhost', and "port up/down" to 1700. Change the "Keepalive interval" value to 10s, "Stat interval" to 30s, "push timeout" to 100ms and "forward crc valid" to true.

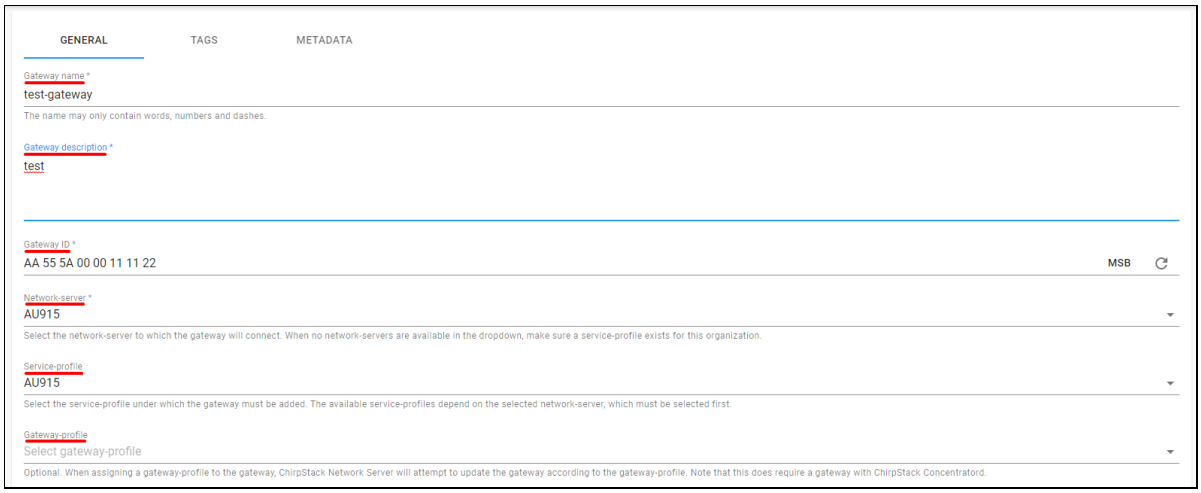


4. In "SX1301 Properties", adjust the settings according to your area's frequency plan and frequency sub-band (FSB) of your choice (ex.: AU915-928 FSB2).
5. Install the ChirpStack gateway bridge in your system. In this guide we'll be installing it on a Windows system to which the gateway will be connected. In such case, download chirpstack-gateway-bridge_3.14.0_windows_amd64.tar.gz, extract the file using 7zip and once again extract the files from the extracted .tar file. Place the .exe file in a folder for itself.



6. Open [LoRaCloud,](#) access your organization's portal with your accound and go into "gateways". Then click "Create".

7. Give your gateway a name and description, in "Gateway ID" insert the Gateway ID copied in step 2, select the 'network-server' according to your location, select the 'service-profile' based on what was selected in the previous field, **leave the "gateway-profile" field blank,** confirm the gateway's creation.



8. Open the newly created gateway's page, access the "Certificate" tab and click on "generate certificate". Go to the folder where the gateway bridge .exe is located and create three files: ca.pem, cert.pem and key.pem. Open these files with a text editor and paste the "CA Certificate", "TLS Certificate" and "TLS Key" keys into ca.pem, cert.pem and key.pem respectively.

Obs.: Copy the whole fields, including the '-----BEGIN CERTIFICATE-----' and '-----END CERTIFICATE-----' file.

9. Open the Command Prompt, go into the folder ChirpStack is located in and run the following command:

chirpstack-application-server.exe configfile > chirpstack-application-server.toml

This will generate a configurations file for the gateway bridge with the name ‘chirpstack-application-server.toml’.

10. Open this file with a text editor and locate the line containing “[backend.semtech_udp]”. A few lines below, edit the one containing ‘udp_bind’ to ‘udp_bind = "0.0.0.0:1700"’. The backend configs should look like this:

```
[backend]

# Backend type.
#
# Valid options are:
#   * semtech_udp
#   * concentratord
#   * basic_station
type="semtech_udp"


  # Semtech UDP packet-forwarder backend.
  [backend.semtech_udp]

  # ip:port to bind the UDP listener to
  #
  # Example: 0.0.0.0:1700 to listen on port 1700 for all network interfaces.
  # This is the listener to which the packet-forwarder forwards its data
  # so make sure the 'serv_port_up' and 'serv_port_down' from your
  # packet-forwarder matches this port.
  udp_bind = "0.0.0.0:1700"

  # Skip the CRC status-check of received packets
  #
  # This is only has effect when the packet-forwarder is configured to forward
  # LoRa frames with CRC errors.
  skip_crc_check = false

  # Fake RX timestamp.
  #
  # Fake the RX time when the gateway does not have GPS, in which case
  # the time would otherwise be unset.
  fake_rx_time=false
```

11. Locate the line containing “[integration.mqtt.auth]”. Below it, in the line containing 'servers=[', insert the LoRaCloud MQTT server address for your frequency band. You may use the image below for reference, or verify in the [Semtech documentation:]

Frequency	Gateway OS (MQTT) Endpoint	Basics Station Endpoint
EU868	eu868.mqtt.loracloud.com:8883	eu868.basicstation.loracloud.com:443
US915	us915.mqtt.loracloud.com:8883	us915.basicstation.loracloud.com:443
CN470	cn470.mqtt.loracloud.com:8883	cn470.basicstation.loracloud.com:443
AU915	au915.mqtt.loracloud.com:8883	au915.basicstation.loracloud.com:443
AS923-1	as923-1.mqtt.loracloud.com:8883	as923-1.basicstation.loracloud.com:443
AS923-2	as923-2.mqtt.loracloud.com:8883	as923-2.basicstation.loracloud.com:443
KR920	kr920.mqtt.loracloud.com:8883	kr920.basicstation.loracloud.com:443
IN865	in865.mqtt.loracloud.com:8883	in865.basicstation.loracloud.com:443

In our example, as we are using the AU915 frequency band, our file will look like this:

```
# Generic MQTT authentication.
[integration.mqtt.auth.generic]
# MQTT servers.
#
# Configure one or multiple MQTT server to connect to. Each item must be in
# the following format: scheme://host:port where scheme is tcp, ssl or ws.
servers=[
  "ssl://au915.mqtt.loracloud.com:8883"
]
```

12. Further below, in the ‘ca_cert=””’ line, you must insert between the quotation marks the filesystem path to the faile containing the CA Certificate (ca.pem). In our example, as we are using Windows, we must replace the ‘\’ divisions for ‘/’:

C:/Users/*user*/Downloads/chirpstack/ca.pem

Do the same for tls_cert and tls_key:

```
# CA certificate file (optional)
#
# Use this when setting up a secure connection (when server uses ssl://...)
# but the certificate used by the server is not trusted by any CA certificate
# on the server (e.g. when self generated).
ca_cert="C:/Users/*user*/Downloads/chirpstack/ca.pem"

# mqtt TLS certificate file (optional)
tls_cert="C:/Users/*user*/Downloads/chirpstack/cert.pem"

# mqtt TLS key file (optional)
tls_key="C:/Users/*user*/Downloads/chirpstack/key.pem"
```

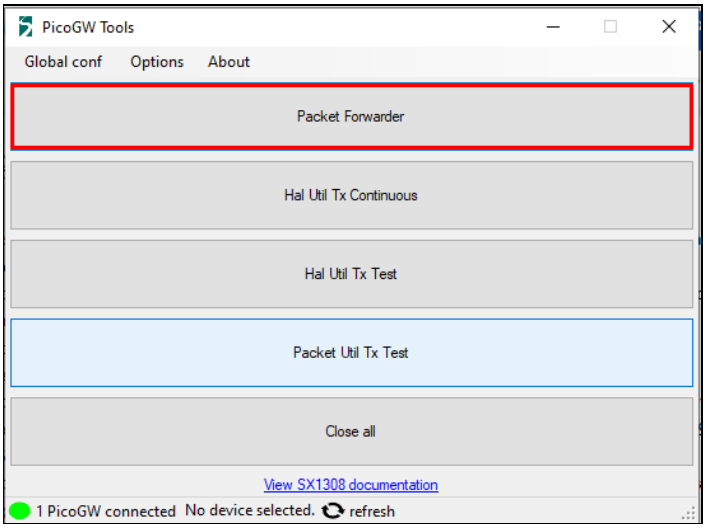
13. Save the file, **open the Command Prompt with administrator privileges** and run the chirpstack-gateway-bridge.exe:

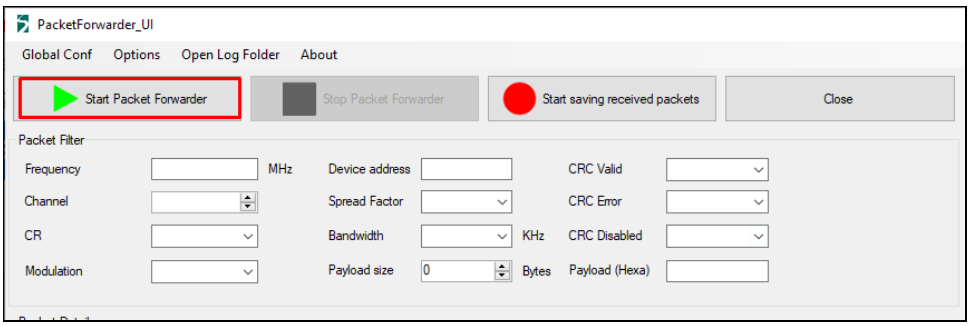
```
C:\Users\    *user*     \Downloads\chirpstack>chirpstack-gateway-bridge.exe
time="2022-07-19T11:43:39.2767812-03:00" level=info msg="starting ChirpStack Gateway Bridge" docs="https://www.chirpstack.io/gateway-bridge/" version=3.14.0
time="2022-07-19T11:43:39.2787739-03:00" level=info msg="backend/semtechudp: starting gateway udp listener" addr="0.0.0.0:1700"
time="2022-07-19T11:43:40.4276946-03:00" level=info msg="integration/mqtt: connected to mqtt broker"
time="2022-07-19T11:43:40.4285996-03:00" level=warning msg="[store]    memorystore wiped" module=mqtt
```

If you get the "integration/mqtt: connected to mqtt broker" message as in the image above, the gateway bridge has been properly configured and it has connected to the LoRaCloud MQTT broker.

Obs.: In case the Command Prompt is not opened with administrator privileges, it is possible that it will not be able to listen to the port and, as a consequence, receive the gateway packets.

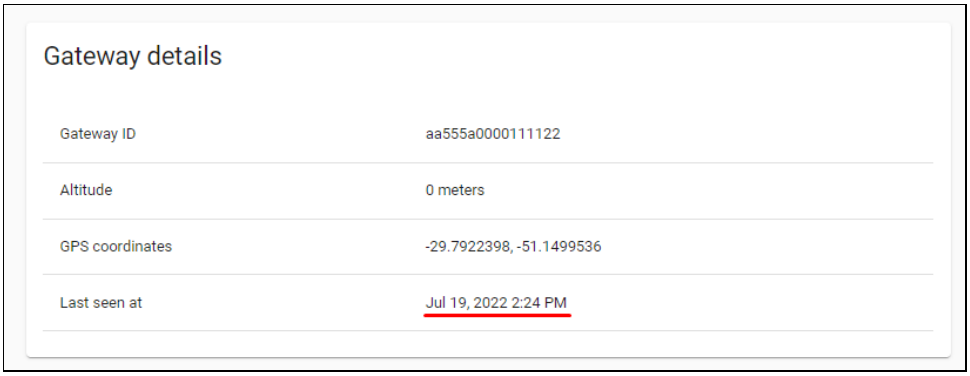
14. Connect the gateway to your computer, open the Packet Forwarder program from PicoGW_UI and start it.





If all is correctly configured, the gateway bridge will start receiving the packets from the gateway's packet forwarder and send them to LoRaCloud.

```
time="2022-07-19T14:24:22.9979957-03:00" level=info msg="integration/mqtt: subscribing to topic" qos=0 topic="gateway/aa555a0000111122/command/#"
time="2022-07-19T14:24:23.3723477-03:00" level=info msg="integration/mqtt: publishing state" gateway_id=aa555a0000111122 qos=0 state=conn topic=gateway/aa555a0000111122
/state/conn
time="2022-07-19T14:24:53.2800679-03:00" level=info msg="integration/mqtt: publishing event" event=stats qos=0 stats_id=43f8a897-461c-4c6b-9c8b-651aeb279000 topic=gatew
ay/aa555a0000111122/event/stats
```



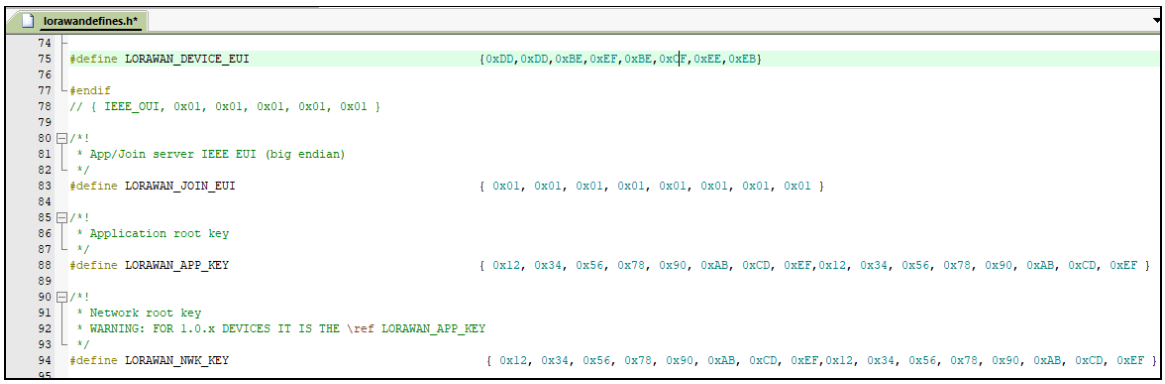
End Node Configuration (LoRaCloud)

1. Open the project's LoRaWAN configurations file (lorawandefines.h).
2. Configure keys to be downloaded onto the device (in hexidecimal) as follows:

LORAWAN_DEVICE_EUI { 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX }

LORAWAN_JOIN_EUI { 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX }

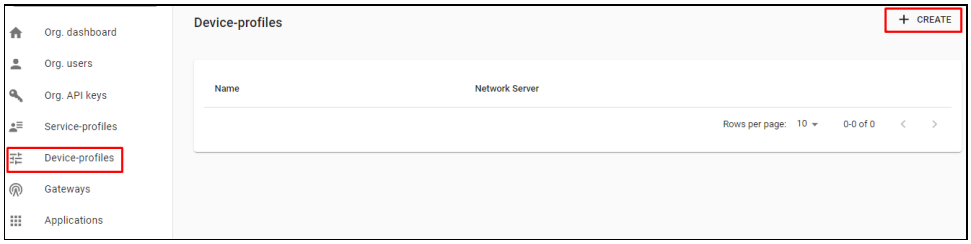
LORAWAN_NWK_KEY { 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX, 0xXX }



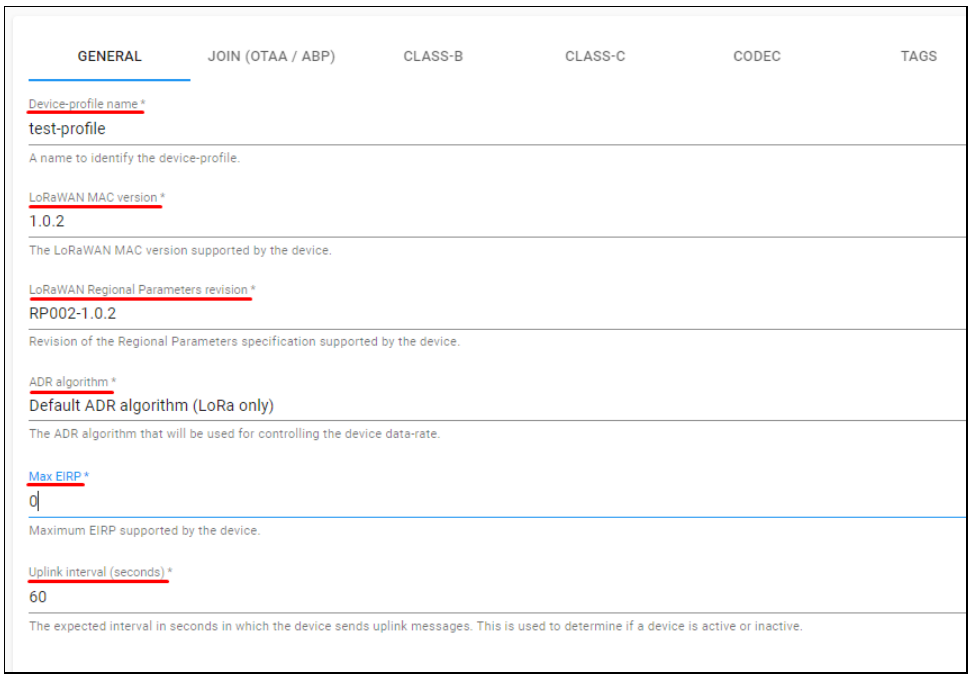
Obs.: Depending on your LoRaWAN version, the keys are given different names. For 1.0.x, the Application Key is defined in LORAWAN_NWK_KEY. For 1.1.x, LORAWAN_NWK_KEY defines the Network Key and LORAWAN_APP_KEY defines the Application Key. In this example we are using LoRaWAN 1.0.2.

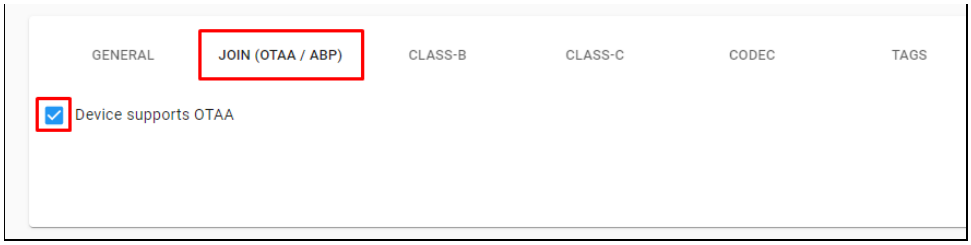
```
*****************************************************************************
********************************** WARNING **********************************
*****************************************************************************
The crypto-element implementation supports both 1.0.x and 1.1.x LoRaWAN
versions of the specification.
Thus it has been decided to use the 1.1.x keys and EUI name definitions.
The below table shows the names equivalence between versions:
             +------------------+--------------------------+
             |       1.0.x      |          1.1.x           |
             +==================+==========================+
             | LORAWAN_DEVICE_EUI| LORAWAN_DEVICE_EUI      |
             +------------------+--------------------------+
             | LORAWAN_APP_EUI   | LORAWAN_JOIN_EUI        |
             +------------------+--------------------------+
             | N/A               | LORAWAN_APP_KEY         |
             +------------------+--------------------------+
             | LORAWAN_APP_KEY   | LORAWAN_NWK_KEY         |
             +------------------+--------------------------+
             | LORAWAN_NWK_S_KEY | LORAWAN_F_NWK_S_INT_KEY |
             +------------------+--------------------------+
             | LORAWAN_NWK_S_KEY | LORAWAN_S_NWK_S_INT_KEY |
             +------------------+--------------------------+
             | LORAWAN_NWK_S_KEY | LORAWAN_NWK_S_ENC_KEY   |
             +------------------+--------------------------+
             | LORAWAN_APP_S_KEY | LORAWAN_APP_S_KEY       |
             +------------------+--------------------------+
*****************************************************************************
```

3. Compile project and download it to the end node device.
4. Open the LoRaCloud portal, open the "Device-profiles" tab and click on "Create".

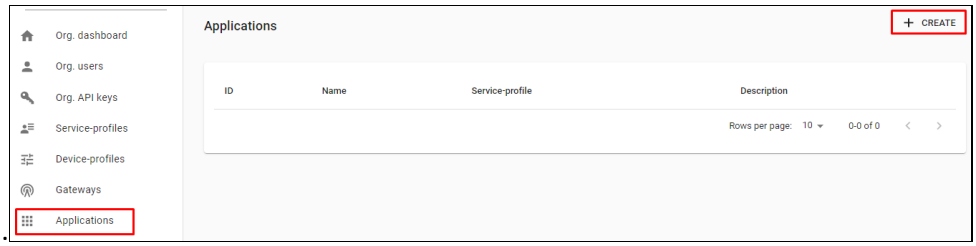


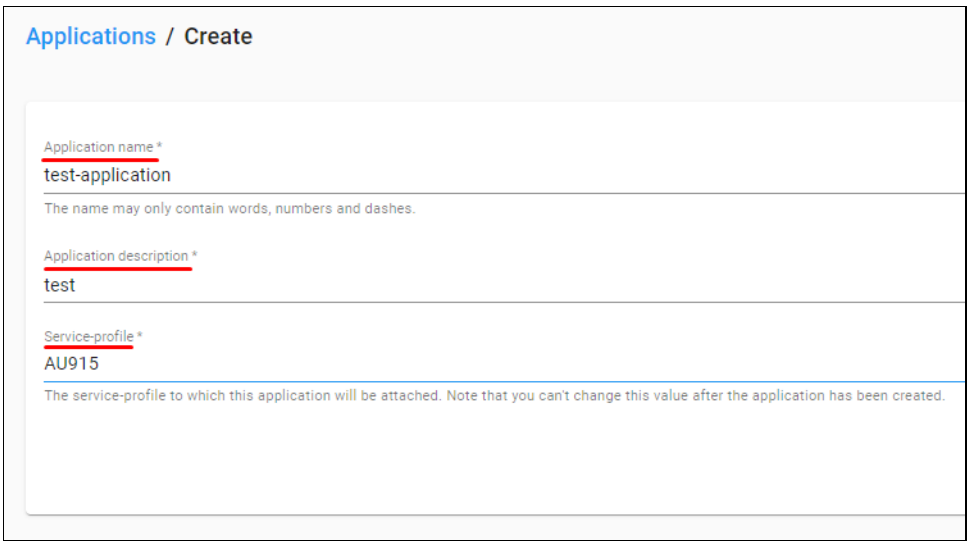
5. Create a name for your device profile, select the same network-server as defined in the gateway creation, select the LoRaWAN MAC version of your device, in “ADR algorithm” select ‘Default ADR algorithm (LoRa only)’. In the “Uplink interval (seconds)” type the interval in seconds that your device will wait between messages to the gateway. Open the “Join (OTAA/ABP)” tab and check the “Device supports OTAA” box. Confirm the creation of the device profile.



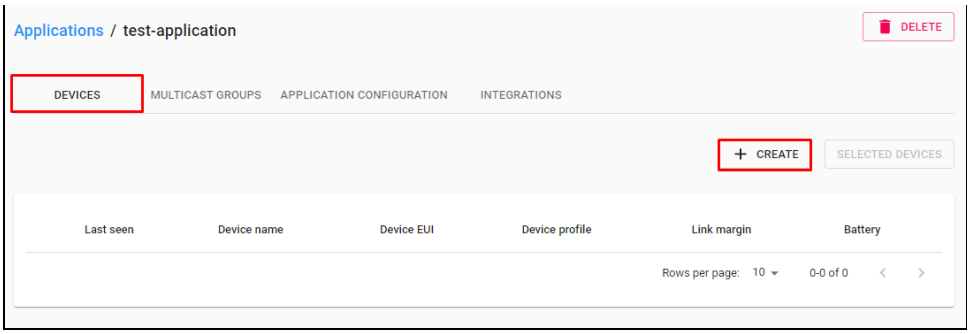


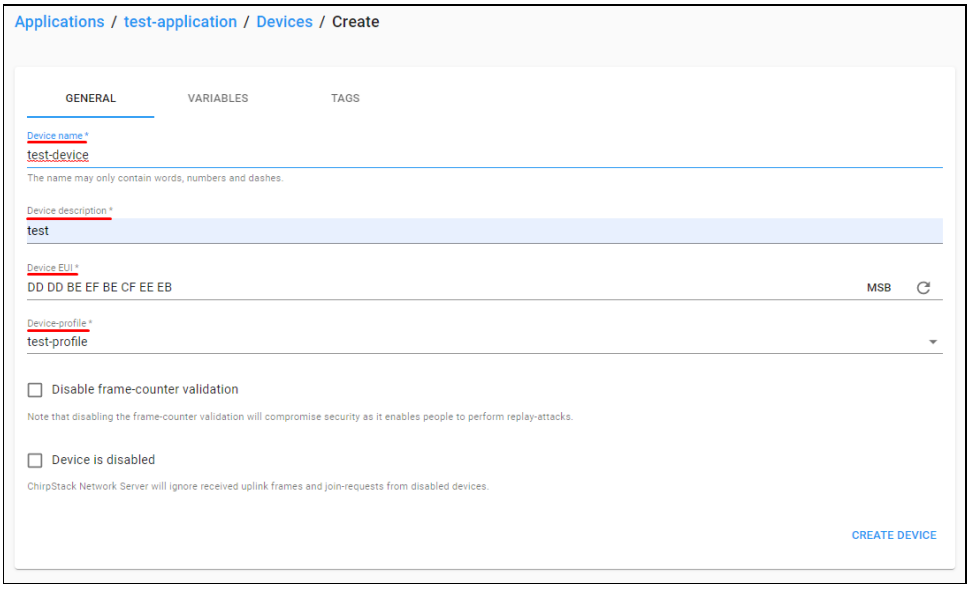
6. In the “Applications” tab, click on “Create”. Create a name and description for your application, and select the same service-profile as the one selected during the gateway creation and confirm the application’s creation



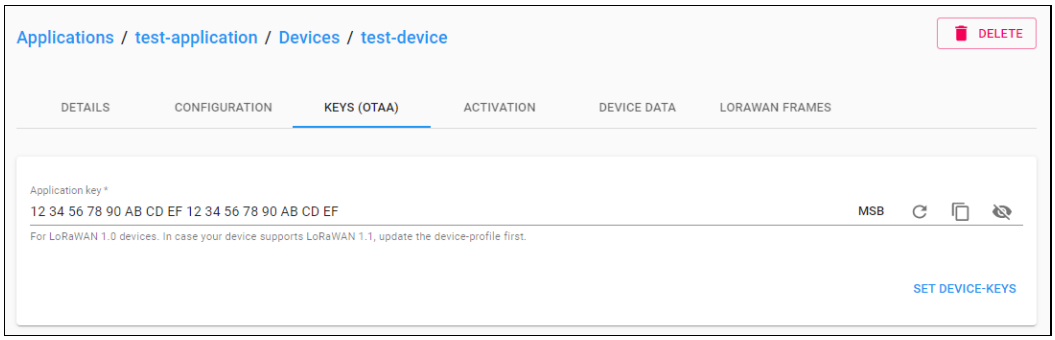


7. In the created application's page, in the "Devices" tab click on "Create". Create a name and description for your device, insert the Device EUI downloaded into the end node on step 2 in the field, then select the device profile created on step 5. Confirm the device's creation.

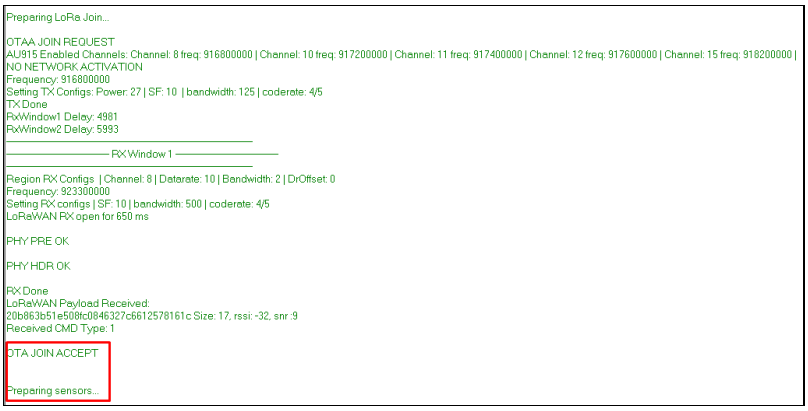




8. After confirming the device creation, it will open its page on the “Keys (OTAA)” tab. Insert the Application Key downloaded to the device on step 2 into the field:



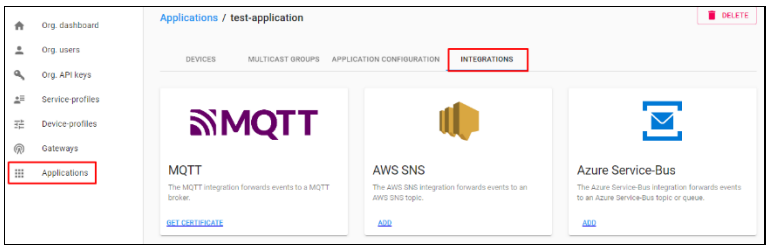
9. Start ChirpStack as instructed in the previous guide, connect your gateway to your computer, start the packet forwarder, and connect your end node to the power supply (i.e. the USB port). Observe the device through a serial port monitor software to see if it is able to connect to LoRaWAN, and if so then go to the “Device Data” tab in your device’s page on the LoRaCloud portal to see if it displays the device’s data.

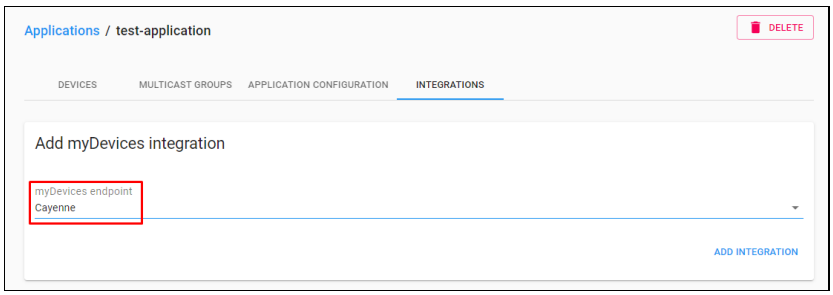


```
================================================================================
================================== LoRaWAN TX ==================================
================================================================================

Sending LoRaWAN Payload: "[01]kÝlØø²J" | Port: 2
Uplink Counter: 42
AU915 Enabled Channels: Channel: 8 freq: 916800000 | Channel: 9 freq: 917000000 | Channel: 11 freq: 917400000
Frequency: 917000000
Setting TX Configs: Power: 27 | SF: 8  | bandwidth: 125 | coderate: 4/5


sensors:lat:-15.799,long:-47.865, temp:21.9, bat:7.1
TX Done
RxWindow1 Delay: 678
RxWindow2 Delay: 1693
```

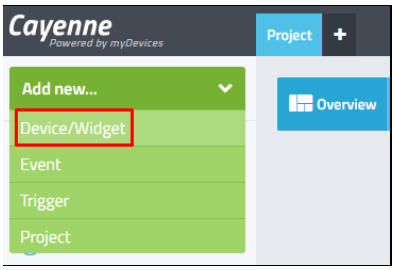
Cayenne Integration (LoRaCloud)

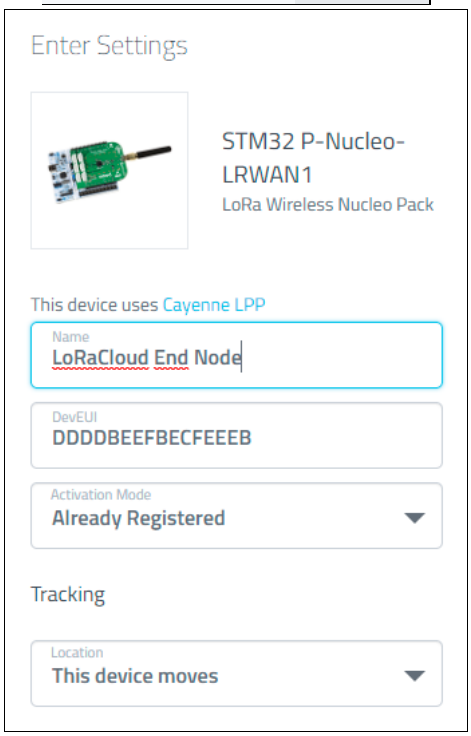
1. Open your application's page in the LoRaCloud portal and access the "Integrations" tab. Find the "myDevices" box and click on "Add". Select Cayene in the "myDevices endpoint" field.



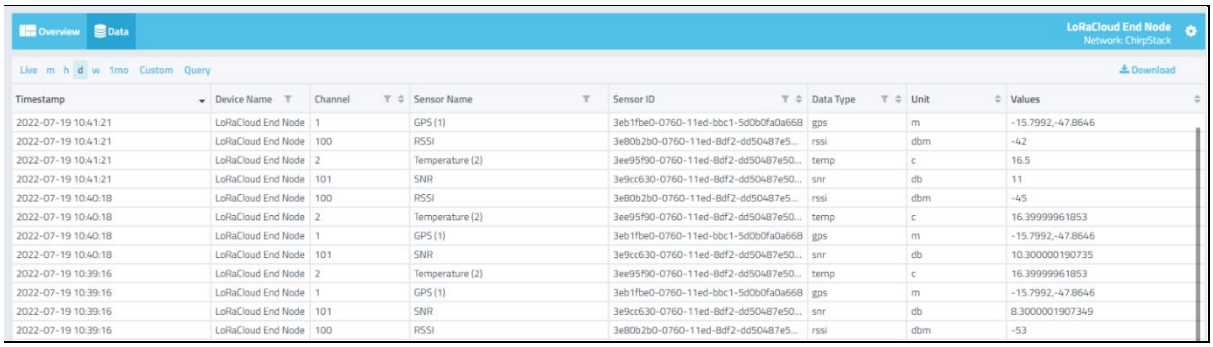


2. Access the Cayenne dashboard (https://mydevices.com/ -> Login -> Cayenne).
3. In your Cayenne dashboard, click on "Add new..." then select 'Device/Widget'. Expand the LoRa devices list, then open the ChirpStack list. Select the "STM32 P-Nucleo-LRWAN1" device and insert its DevEUI in the field.





4. Connect the device to LoRaWAN through the gateway and wait for Cayenne to receive its data.



Timestamp	Device Name	Channel	Sensor Name	Sensor ID	Data Type	Unit	Values
2022-07-19 10:41:21	LoRaCloud End Node	1	GPS (1)	3eb1fbe0-0760-11ed-bbc1-5d0b0fa0a668	gps	m	-15.7992,-47.8646
2022-07-19 10:41:21	LoRaCloud End Node	100	RSSI	3e80b2b0-0760-11ed-8df2-dd50487e5...	rssi	dbm	-42
2022-07-19 10:41:21	LoRaCloud End Node	2	Temperature (2)	3ee95f90-0760-11ed-8df2-dd50487e50...	temp	c	16.5
2022-07-19 10:41:21	LoRaCloud End Node	101	SNR	3e9cc630-0760-11ed-8df2-dd50487e50...	snr	db	11
2022-07-19 10:40:18	LoRaCloud End Node	100	RSSI	3e80b2b0-0760-11ed-8df2-dd50487e5...	rssi	dbm	-45
2022-07-19 10:40:18	LoRaCloud End Node	2	Temperature (2)	3ee95f90-0760-11ed-8df2-dd50487e50...	temp	c	16.39999961853
2022-07-19 10:40:18	LoRaCloud End Node	1	GPS (1)	3eb1fbe0-0760-11ed-bbc1-5d0b0fa0a668	gps	m	-15.7992,-47.8646
2022-07-19 10:40:18	LoRaCloud End Node	101	SNR	3e9cc630-0760-11ed-8df2-dd50487e50...	snr	db	10.300000190735
2022-07-19 10:39:16	LoRaCloud End Node	2	Temperature (2)	3ee95f90-0760-11ed-8df2-dd50487e50...	temp	c	16.39999961853
2022-07-19 10:39:16	LoRaCloud End Node	1	GPS (1)	3eb1fbe0-0760-11ed-bbc1-5d0b0fa0a668	gps	m	-15.7992,-47.8646
2022-07-19 10:39:16	LoRaCloud End Node	101	SNR	3e9cc630-0760-11ed-8df2-dd50487e50...	snr	db	8.300000190735
2022-07-19 10:39:16	LoRaCloud End Node	100	RSSI	3e80b2b0-0760-11ed-8df2-dd50487e5...	rssi	dbm	-53