# LoRaWAN Applications Workshop

Part 1 – LoRa Communications through The Things Network

## 1 Workshop Goal

To understand the way in a full LoRa network is configured using The Things Network services, from LoRa end-devices to the LoRa application server, how LoRa end-devices transmit data and the technical limitations of this technology.

## 2 Tools

Each group will make use of the following hardware:

- One configurable LoRa end-device supported by RIOT-OS (Board B-L072Z-LRWAN1)
- One Picocell Lora Gateway Board
- One Raspberry Pi with Picocell drivers and Packet Forwarder installed
- One PC with RIOT-OS configured with the toolchain for the LoRa end-device, and with internet access.

## 3 Introduction

This document is composed of several sections, each of which provides instructions for different tasks. The tutor will guide you through each one of the tasks during the workshop class.

Each task has an objective, a short theoretical background, a description of the steps to follow, a procedure for verifying that the objective of a section was achieved, and a set of questions to be answered by the students. At the end of each task, a questions and discussion session will be open during the workshop class for cross-verification of the achieved results of each task.

You should not continue to the next section if the verification procedure was not successful.

## 4 The Things Network

The Things Network (TTN) is a cloud platform that freely provides an operating LoRa Application server, LoRa Network server and other services for integrating external applications with a world-wide LoRa network.

The objective of this task is to set-up your account in TTN. TTN will be used in the upcoming two sections for setting up a LoRa Gateway and LoRa end-device.

Configuration steps: Go to https://www.thethingsnetwork.org and click on the “Sign up” button (usually on the right top corner of the website). Follow the instructions for creating and activating your account.

Verification steps: verify that you access TTN by logging with your credentials to your TTN account.

# 5 LoRa Gateway

The LoRa Gateway will provide to LoRa end-device access to TTN. The LoRa Gateway will forward all frames between the LoRa Network server and LoRa end-devices. The LoRa Gateway runs different process for this. The LoRa Packet Forwarder is the process running in the LoRa Gateway in charge of relaying frames between the LoRa Network server and LoRa end-devices.

The objective of this task is to configure both a LoRa Gateway and your TTN account for allowing the transmission of LoRa frames between your LoRa Gateway and the TTN’s Lora Network Server.

## 5.1 Configuration steps

### *5.1.1 Access the LoRa gateway.*

1. The tutor will provide you the IP address of the LoRa gateway for your team.
2. Access the LoRa gateway using ssh. Username: iot, Password: iot. An example of a command to be executed in the console of your laptop is provided below. You should replace the IPv4 address for the one given by your tutor.

```
ssh iot@192.168.0.168
```

3. If the previous step fails, verify that you can ping the LoRa gateway using the ping command of your operating system.

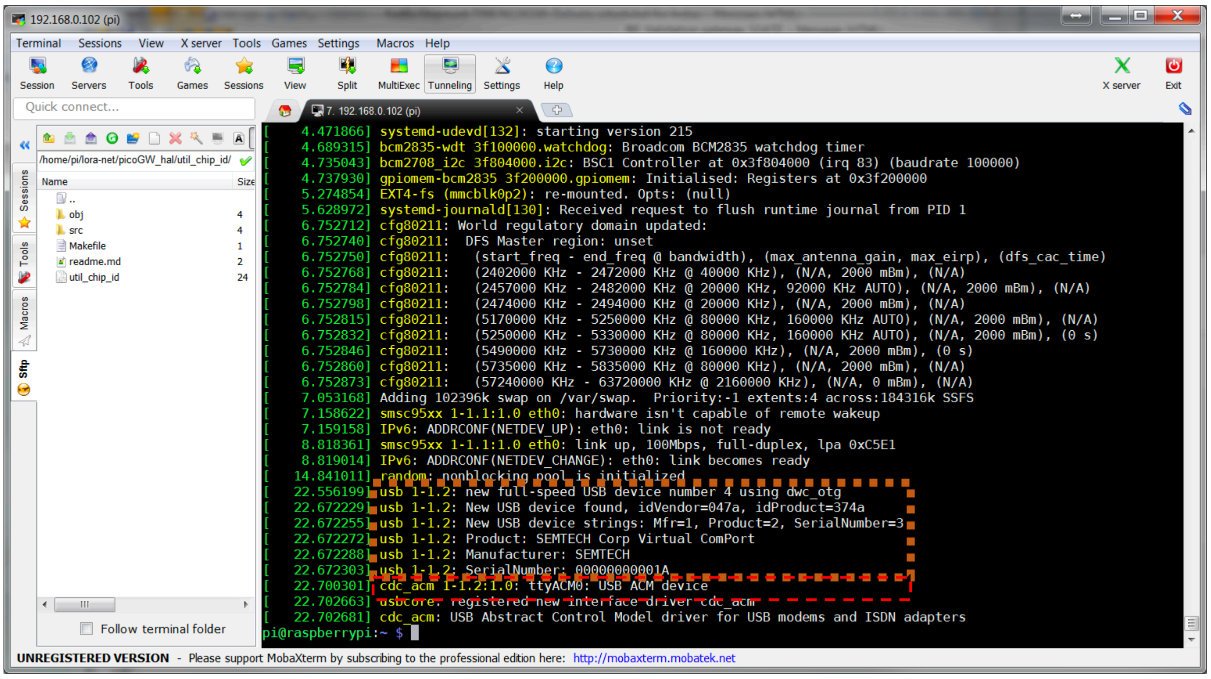
### *5.1.2 Checking the ID of the LoRa radio board of the LoRa Gateway*

1. Execute the `dmesg` command in the console with access to the LoRa Gateway.

```
dmesg
```

This will allow you to check the Linux kernel message logs. More information about this command can be found here.

2. Verify that the output of the command contains a sequence of lines, all starting with “`usb`” referring to a SEMTECH device (as shown in the orange box below). Then, find the first line starting with “`cdc_acm`” (red box in the figure below) after the “`usb`” block – this may be located lines further down. Write down in a piece of paper the name of the terminal port used by Linux for communicating with the LoRa radio board. It shall start with the prefix `tty`, and it is usually `ttyACM0`.


192.168.0.102 (pi)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Settings Help X server Exit
Quick connect...
7. 192.168.0.102 (pi)
/home/pi/lora-net/picoGW_hal/util_chip_id/
Name Size
..
obj 4
src 4
Makefile 1
readme.md 2
util_chip_id 24
Follow terminal folder
Sessions Tools Macros Sftp
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net

```
[    4.471866] systemd-udevd[132]: starting version 215
[    4.689315] bcm2835-wdt 3f100000.watchdog: Broadcom BCM2835 watchdog timer
[    4.735043] bcm2708_i2c 3f804000.i2c: BSC1 Controller at 0x3f804000 (irq 83) (baudrate 100000)
[    4.737930] gpiomem-bcm2835 3f200000.gpiomem: Initialised: Registers at 0x3f200000
[    5.274854] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[    5.628972] systemd-journald[130]: Received request to flush runtime journal from PID 1
[    6.752712] cfg80211: World regulatory domain updated:
[    6.752740] cfg80211:  DFS Master region: unset
[    6.752750] cfg80211:   (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp), (dfs_cac_time)
[    6.752768] cfg80211:   (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[    6.752784] cfg80211:   (2457000 KHz - 2482000 KHz @ 20000 KHz, 92000 KHz AUTO), (N/A, 2000 mBm), (N/A)
[    6.752798] cfg80211:   (2474000 KHz - 2494000 KHz @ 20000 KHz), (N/A, 2000 mBm), (N/A)
[    6.752815] cfg80211:   (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2000 mBm), (N/A)
[    6.752832] cfg80211:   (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2000 mBm), (0 s)
[    6.752846] cfg80211:   (5490000 KHz - 5730000 KHz @ 160000 KHz), (N/A, 2000 mBm), (0 s)
[    6.752860] cfg80211:   (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 2000 mBm), (N/A)
[    6.752873] cfg80211:   (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 0 mBm), (N/A)
[    7.053168] Adding 102396k swap on /var/swap.  Priority:-1 extents:4 across:184316k SSFS
[    7.158622] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[    7.159158] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[    8.818361] smsc95xx 1-1.1:1.0 eth0: link up, 100Mbps, full-duplex, lpa 0xC5E1
[    8.819014] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[   14.841011] random: nonblocking pool is initialized
[   22.556199] usb 1-1.2: new full-speed USB device number 4 using dwc_otg
[   22.672229] usb 1-1.2: New USB device found, idVendor=047a, idProduct=374a
[   22.672255] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   22.672272] usb 1-1.2: Product: SEMTECH Corp Virtual ComPort
[   22.672288] usb 1-1.2: Manufacturer: SEMTECH
[   22.672303] usb 1-1.2: SerialNumber: 00000000001A
[   22.700301] cdc_acm 1-1.2:1.0: ttyACM0: USB ACM device
[   22.702663] usbcore: registered new interface driver cdc_acm
[   22.702681] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
pi@raspberrypi:~ $
```

3. Type in the shell the following command. Note: you should replace ‘ttyACM0’, if the result of the previous step showed you a different terminal port.

```
cd   ~/lora-net/picoGW_hal/util_chip_id
```

And then

```
./util_chip_id -d /dev/ttyACM0
```

The command will show you the Gateway EUI (universal identifier of the Gateway), which is a 64-bit number written as 16 hexadecimal characters. Write down in a piece of paper the Gateway EUI for further usage.

### 5.1.3 *Configuring the gateway in TTN*

1. On the main TTN webpage, click on “Start building”, select “Europe”, select “Go to gateways” and click on the “Add gateway” button.
2. Provide a name for your gateway in the “Gateway ID” field. You may name it as you wish.
3. In the Gateway EUI field, type in the serial number of the radio board, as obtained in section 4.1.2 step 3.
4. In the frequency plan field, choose the SF9 option for Europe.
5. Make sure the “Schedule downlink” and the “Enforce duty cycle” options are selected as enabled.
6. Leave all other fields untouched and save your configuration by pressing on “Create gateway” button.

**Once you finalize your laboratory you MUST delete the configured Gateway from your TTN account. If this is not done, other students will not be able to use it for the laboratories. To remove it, go to Gateways -> (Select your gateway) -> General settings -> Delete gateway.**

### 5.1.4 Setting up the Packet Forwarder process of the LoRa Gateway

1. On TTN, click on the "Gateways" tab on the top of the window. Your newly configured gateway should appear on the list.
2. Click on your newly configured gateway in the list.
3. TTN can generate for you the configuration file for the Packet Forwarder process allowing it to register and communicate with TTN. Click on "Download global_conf.json" to obtain it.
4. You should copy this file from your Windows machine to the LoRa Packet Forwarded folder on the LoRa Gateway. You have several options to do this. If you do not know how to do this, you may follow the following procedure. You open a terminal window in your Windows system (not on the LoRa gateway), go to the folder where the global_conf.json file was downloaded, and use the `scp` command to copy it to the "/home/iot/lora-net/picoGW_packet_forwarder/lora_pkt_fwd" of the LoRa Gateway. An example of `scp` for copying a file "file.txt" from a laptop to a remote host with IP address "remote_ip_address" in a folder "/remote/directory/" is provided below. More information about this command can be found here.

```
scp {filename} {username}@{remote_ip_address}:{/remote/directory/}
```

5. On the LoRa Gateway, go to the LoRa Packet Forwarder folder using the `cd` command, which is located in "/home/iot/lora-net/picoGW_packet_forwarder/lora_pkt_fwd"
6. Start the LoRa Packet Forwarder process on the LoRa Gateway with the following command:

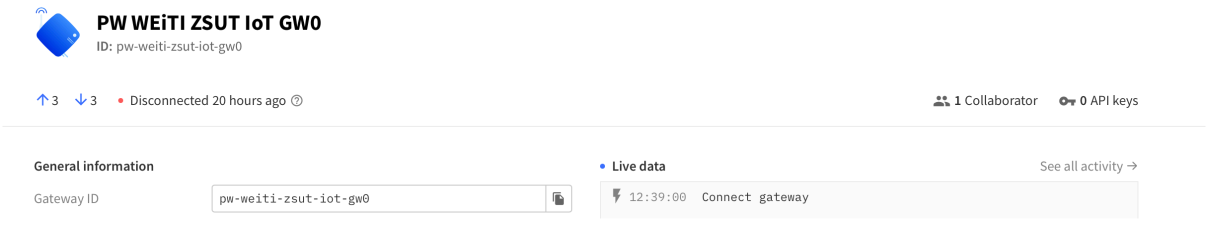
```
./lora_pkt_fwd -d /dev/ttyACM0
```

You should replace the ttyACM0 word in the last parameter in case the Linux OS is using a different terminal for connecting to the radio board (check section 4.1.2 step 2).

## 5.2 Verification procedure

### 5.2.1 Verify activity on TTN

On the "Gateways" tab of TTN, verify that your gateway shows the status as "Connected" on the last column of the table. You may, in addition, click on your gateway and check on the live data window on the right for the "Connect gateway" message log, as shown below:



**PW WEiTI ZSUT IoT GW0**
ID: pw-weiti-zsut-iot-gw0

↑3 ↓3 • Disconnected 20 hours ago — 1 Collaborator — 0 API keys

General information
Gateway ID: pw-weiti-zsut-iot-gw0

• Live data — See all activity →

```
12:39:00  Connect gateway
```

### 5.2.2 Verify activity of the Packet Forwarder

On the Lora Gateway console, the Lora Packet Forwarder process will continue running. Verify that the log shows periodically an information message "INFO: [down] PULL_ACK received in XX ms", which confirms that there is a two-way communication between the LoRa Gateway and the TTN LoRa Network Server.

## 5.3 Questions and Analysis

Alter the Gateway_EUI in the global_conf.json file, so it has a different value. Does the gateway connect to TTN? Does TTN shows activity from your Gateway? After finalizing this task, please make sure to revert to its original value the Gateway_EUI field of the global_conf.json file.

# 6 LoRa end-device

The LoRa end-device will host a test application (built application at RIOT/test/pkg_semtech-loramac) for the LoRa stack from RIOT-OS.

The objective of this task is to configure both a LoRa end-device and your TTN account for allowing the uplink and downlink transmission of frames, up to the TTN's LoRa Application Server.

## 6.1 Configuration steps

### *6.1.1 Setting up a LoRa application on TTN*

1. On the TTN website, go to the "Applications" tab.
2. You should first configure a LoRa Application, before configuring LoRa end-devices. For this, click on the "Add application" button.
3. Fill in the "Application ID" field with your own name for your application, and then click on "Create application".
4. On the list of Applications, click on the newly created application.
5. Press the "Add end device" button, and the select the "Manually" tab.
6. Set the frequency plan field to the same value set for your LoRa Gateway (see Section 4.1.3 step 4).
7. Set LoRaWAN version to 1.0.0.
8. Click on "Generate" for the DevEUI
9. Click on "Fill with zeros" for the AppEUI field.
10. Click on "Generate" for the AppKey field
11. Write down the DevEUI, AppKey and AppEUI in a piece of paper for future use in section 5.1.2.
12. Click on "Register end device"

### *6.1.2 Setting up and connecting your LoRa end-device*

1. Open a new terminal to the console of the B-L072Z-LRWAN1 board, which will be used as a LoRa end-device. You may do so using the putty application in your laptop. Instruction for connecting to the console of this board were given during the RIOT-OS Laboratory. For the following steps, be aware that pasting text from the clipboard on the console of the B-L072Z-LRWAN1 board may block it. To avoid problems, type the text instead.
2. Set up the DevEUI of the LoRa end-device by typing in the RIOT-OS console the following command

   ```
   loramac set deveui 0011223344556677
   ```
   Where 0011223344556677 shall be the DevEUI generated in Section 5.1.1 step 8.
3. Set up the AppKey on the LoRa end-device by typing in the RIOT-OS console the following command

   ```
   loramac set appkey 0011223344556677
   ```

Where 0011223344556677 shall be the AppKey generated in Section 5.1.1 step 8.

4. Set up the AppEUI on the LoRa end-device by typing in the RIOT-OS console the following command

```
loramac set appeui 0011223344556677
```

Where 0011223344556677 shall be the AppEUI set in Section 5.1.1 step 9.

5. Execute the LoRa join procedure for connecting the end-device to the LoRa network by executing the following command.

```
loramac join otaa
```

## 6.2 Verification procedure

The TTN "Live data" window of the end-device *may* show "Accept join-request" followed by "Forward join-accept message", indicating a successful processing of the join request frame.

## 6.3 Questions and Analysis

Do you need to specify in the LoRa end-device the identifier of the Gateway that it should connect to?Explain.

The LoRa Packet Forwarder process log should display a message confirming the reception of the LoRa join frame. The log message may look like as follows

```
##### 2022-05-10 12:35:28 GMT #####
### [UPSTREAM] ###
# RF packets received by concentrator: 6
# CRC_OK: 83.33%, CRC_FAIL: 16.67%, NO_CRC: 0.00%
# RF packets forwarded: 5 (115 bytes)
# PUSH_DATA datagrams sent: 2 (1066 bytes)
# PUSH_DATA acknowledged: 100.00%
```

Does the LoRa Packet Forwarder always show the join of your device frame? Does the LoRa Packet Forwarder show the join of your device frame only?

# 7 Uplink Transmission and Duty Cycle

Up to this point, the LoRa end-device has successfully join the LoRa network, enabling communication with the LoRa Application Server.

The objective of this task is to test the LoRa frame transmission from the LoRa end-device to the LoRa application server (uplink) and empirically evaluate its limitations.

## 7.1 Task steps

### *7.1.1 Sending a frame*

To send a frame from the end-device use the following command on the RIOT-OS console

```
loramac tx msg
```

Where msg is the text message (without spaces) you want to send.

#### 7.1.2 Checking which LoRa Gateways received your frame and the airtime

On the "Live data" subtab, click on the "Forward uplink data message". A log in JSON format is displayed. Find the JSON object `data -> uplink` -> `rx_metada`, and the list of gateways within in. Check whether your gateway received this message. For each gateways, there is an RSSI, an SNR and a timestamp. In `data -> uplink -> consumed_airtime` you can observe how long time it took the radio transmission of your frame. In `data -> uplink -> settings -> data_rate` you can see the transmission parameters used by your LoRa radio: spreading factor and bandwidth.

#### 7.1.3 Checking the duty cycle

The rate of transmission of LoRa frames is constraint in time. This means that it is not possible for an end-device to continuously transmit frames without interruptions. To empirically check what the duty cycle of your LoRa application, execute several times the send frame command and make a pause between the execution of these commands. Use a time stopwatch or countdown timer for counting the time between frame transmissions. If you do not have access to one, you may use one from the internet here.

## 7.2 Verification procedure

#### 7.2.1 On the LoRa end-device

The RIOT-OS console should show two messages "Received ACK from network" and "Message sent with success", after successful transmission of the frame.

#### 7.2.2 On the LoRa Gateway

The LoRa Packet Forwarder should display "JSON up: {"**rxpk**"…" log for each message received from the end-device.

#### 7.2.3 On TTN

The "Live data" box on the end-device should display "Forward uplink data message". The transmitted text should be displayed in ASCII HEX format as the MAC payload in the same line.

## 7.3 Questions and Analysis

Discuss in your group the answer to the following questions:

1. What is the sent payload text you sent in Section 6.1.1?
2. Based on the data rate observed in Section 6.1.2, calculate the airtime. Does the calculated airtime match with the one reported on the logs? Explain.
3. What is the duty cycle period you observed in Section 6.1.3?

# 8 TTN's Integration and MQTT

TTN provides a service for interoperability with external applications. These services are called integrations. One of the supported integrations allows the usage of the MQTT protocol for accessing an API for managing LoRa end-devices and gateways.

The objective of this task is to test the MQTT integration for communicating with external applications.

## 8.1 Configuration steps

### 8.1.1 On TTN – MQTT connection settings and key

1. Go to the "Applications" tab, and choose your application from the list
2. On the side-menu bar, click on "Integrations" and "MQTT"
3. Write down the "Public address" and "Username". We will refer to the later field as the MQTT username in this guide. It normally is composed as your TTN username and the suffix "@ttn".
4. Click on "Generate new API key"
5. Copy the content of the generated "Password" field for further usage – you will not see it again ever in TTN!

### 8.1.2 On Virtual Box

1. Run the „IoT Virtual Lab Light" virtual machine in VirtualBox
2. Log in with username and password: osboxes and osboxes.org
3. Go to folder "/usr/sbin"
4. Run the mosquito subscribe command using the following syntax

```
mosquito_sub -h {ip_address} -t {topic} -u {username} -P {password}
```

5. The format of the MQTT topic string for upstream messages is "v3/{mqtt_username}/devices/{device_id}/up". Example: "v3/fs-lz30@ttn/devices/eui-70b3d57ed00504aa/up"

Information about other MQTT end-points and topics for TTN can be found here.

## 8.2 Verification procedure

Send a frame from your LoRa end-device, as explained in previous tasks. Make sure your frame is received in JSON format in your MQTT client.

## 8.3 Questions and Analysis

- Is there any way that an MQTT client be notified of a LoRa device joining the network? How? (hint: check the list of TTN's end-points for MQTT)?
- How could an MQTT client be notified of a LoRa join event for *any* device of this application (especially when the MQTT client does not know all device IDs)?
- Could an MQTT client be notified when a selected device joins any LoRa application (especially when the MQTT client does not know all possible application IDs)?
- Is there any way that an MQTT client be notified of a LoRa device leaving the network? Why?

# 9 LoRaWAN Downlink Transmission and Timeout from TTN

The objective of this section is to test the transmission of frames from the LoRa Application Server to the LoRa end-device and evaluate its restrictions.

## 9.1 Configuration steps

#### *9.1.1 On TTN*

1. Open the tab "Messaging" of your end-device in TTN.
2. Select the "Downlink" subtab
3. Type a hex string in the payload field.
4. Select the "Confirmed downlink" checkbox
5. Leave all other options untouched
6. Press the "Schedule downlink" button

#### *9.1.2 On the LoRa end-device*

Send one (uplink) LoRa frame from the end-device by following the procedure of section 6.1.1.

## 9.2 Verification procedure

#### *9.2.1 On the LoRa end-device*

After transmission of the uplink frame, RIOT-OS should include one line "Data received: xxx, port: 1", where xxx is the string representation of the hex string typed in Section 7.1.1 step 3.

#### *9.2.2 On the LoRa Gateway*

The LoRa Packet Forwarder of your Gateway *may* display "JSON up: {"**txpk**"…" log for each message received from the end-device.

## 9.3 Questions and Analysis

Discuss in your group the answer to the following questions:

1. What class of end-device are we using?
2. What is the queue timeout period?
3. Did you see a log message in your gateway about the message transmission (point 8.2.2)?

# 10 MQTT Downlink Client

The objective of this section is to test TTN API for downlink LoRa frame transmissions.

## 10.1 Task steps

#### *10.1.1 On Virtual Box*

1. Run the „IoT Virtual Lab Light" virtual machine in VirtualBox
2. Log in with username and password: osboxes and osboxes.org
3. Go to folder "/usr/sbin"

4. The format of the MQTT topic string for downstream message transmission is “v3/{mqtt_username}/devices/{device_id}/down/push”. Example: “v3/fs-lz30@ttn/devices/eui-70b3d57ed00504aa/down”
5. The message to be published should have the following JSON format

   ```
   {
    "downlinks": [{
     "f_port": 15,
     "frm_payload": "vu8=",
     "priority": "NORMAL"
    }]
   }
   ```

   For f_port you may write any number between 1 and 233. Frm_payload should use base64 encoding. You may use this tool for base64 encoding. In the example above, the hexadecimal payload BE EF was encoded.
6. Run the mosquito subscribe command using the following syntax

   ```
   mosquito_pub -h {ip_address} -m “{message}” -t {topic} -u {username} -P
   {password}
   ```

## 10.2 Verification procedure

In TTN, go to the “Applications” (top menu bar), select your application and then click on the “Live data” option (left menu bar). A new item “Receive downlink data message” should appear, with the payload encoded in hexadecimal, once the MQTT publish request is received and internally queued for later transmission.

Once TTN finds the proper time for sending the queued message, it will display in the “Live data” window of the end-device “Schedule data downlink for transmission”.

## 10.3 Questions and Analysis

N.A.

Remember to delete your Gateway from your TTN account once you have finished this workshop.