LoRa Software Modem Evaluation Kit: User Guide

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This user guide introduces the Semtech LoRa Software Modem and describes the steps to set up an End-Node to Cloud demonstration.

1.1 System Architecture

The major components of the End-Node to Cloud demo consist of the End-Node (carrying the LoRa Software Modem and the Application Host), the LoRaWAN infrastructure (made of a Gateway, a Network Server, and an Application Server) and the LoRa Cloud.

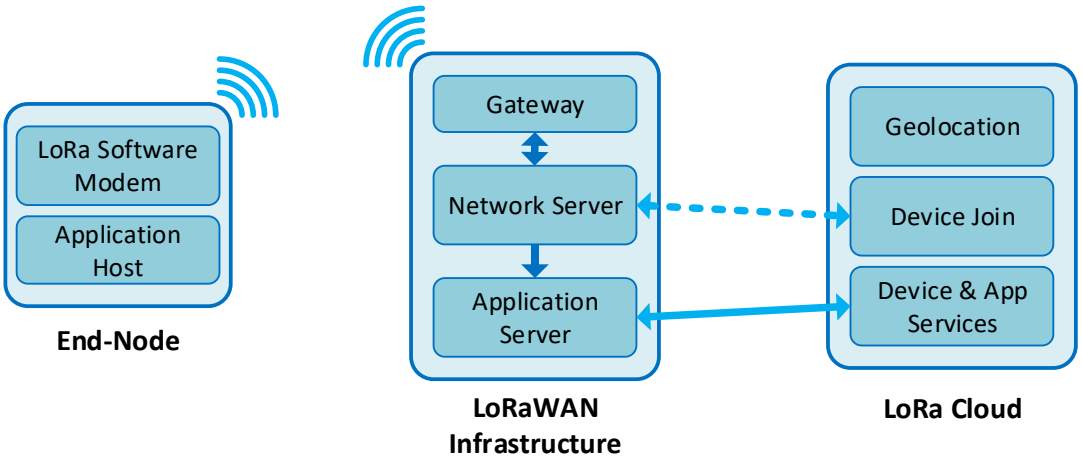

Figure 1. High level diagram of all critical components of End-Node to Cloud demo

1.2 The End-Node

The End-Node consists of two of boards: the Murata Module Shield board that runs the LoRa Software Modem and a ST Nucleo Development board that hosts the end-node application.

1.2.1 Murata Module Shield

The LoRa Software Modem is a firmware developed by Semtech that provides complete physical and MAC layer functions within the MuRata LoRa Module (CMWX1ZZABZ-104). It is designed to support both Class A and C devices, and meet all LoRaWAN specifications. In addition, the LoRa Software Modem firmware supoorts services offered by Semtech LoRa Cloud. This module is mounted on an Mbed shield format board.

Figure 2. CMWX1ZZABZ-104 mbed shield

1.2.2 ST Nucleo Development Board

To facilitate the evaluation of the LoRa Software Modem, an application layer software is available to run on a separate application processor (Application HOST). The chosen platform for the Application HOST is the ST Nucleo L073RZ5 development board which uses the STM32L073RZ[1] microcontroller.

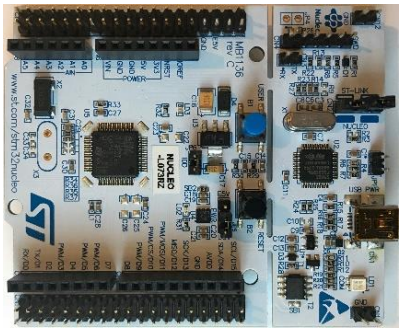

Figure 3. NUCLEO L073RZ5 Development Board

1.3 Gateway

The gateway is a device which forms a bridge between the End-Node and the LoRaWAN Network Server. Messages are sent wirelessly by the End-Node to the gateway using the LoRaWAN protocol. The gateway then forwards the packets to the Network Server through Ethernet, WiFi, or cellular backhaul. LoRa gateways commonly come in 8, 16, or 64 channels.

To complete the example presented in this User Guide, it's expected that a LoRaWAN gateway is within range of the End-Node.

1.4 LoRa Cloud

1.4.1 Geolocation Server

The geolocation server will not be covered in this document; it allows easy estimated location of any LoRaWAN-based device[2]

1.4.2 Device & Application Services

It is a set of full life-cycle management features for LoRaWAN®-based devices. These features include but not limited to:

- Comprehensive device telemetry
- Device and application configuration
- Clock synchronization
- Advanced data transport services with configurable robustness against packet loss and transparent data fragmentation.

[1] More information at: https://www.st.com/en/microcontrollers-microprocessors/stm32l073rz.html

[2] See https://www.loracloud.com/documentation/geolocation?url=gettingstarted.html

The purpose of this service is to dramatically simplify the process of developing managed endpoint solutions and to make LoRa Technology more accessible to application developers.

All Device & Application Services (DAS) frames are sent at application level, which means that if the user chooses to forward uplinks from the network server to the DAS, the DAS does not need to know the devices keys.

1.4.3 Device Join Server

LoRa Cloud Device Join Server allows claiming pre-provisioned LoRaWAN® modem-based devices. The standard LoRaWAN join server interfaces directly to LoRaWAN network servers in order to provide fully secured End-Node on boarding and network join.

For those wishing to move their devices to another join server, there is also the possibility to extract the AppKey and to re-key the End-Node.

1.5 LoRa Network Server

For the demo presented in this document, The Things Networks (TTN) will be used as the LoRa Network Server. Other LoRaWAN network servers can be used if desired.

To learn more about the TTN LoRaWAN network server and how to create an account and use the services, please visit https://www.thethingsnetwork.org/docs/network/

1.6 Application Server

For the demo presented in this document, Node-RED will be used for LoRa Application server. Other Application servers can be used if desired. For more information on Node-RED, please visit https://nodered.org/

2 Quick Start Guide

2.1 Required Hardware

- 1x CMWX1ZZABZ-104 Mbed Shield
- 1x Nucleo-L073RZ Development Board
- 1x Linx ANT-916-CW-HW or Linx ANT-868-CW-HW Antenna
- 1x Mini USB Cables
- 1x PC with Windows 7 or above

2.2 Required Server Accounts

- Semtech LoRaWAN Cloud Services – https://www.loracloud.com/
- The Things Network LoRaWAN Network Server - https://www.thethingsnetwork.org/

Detailed information on the creation of these accounts are provided in Sections 3.1 and 3.4.

2.3 Required Program Installation

The following programs or equivalent are required to complete the examples described in this document.

- Python 3.0 or above[3]
- Teraterm
- NodeRED Application Server

2.4 Hardware Connection

All of the communication between the LoRa Software Modem Shield board and a serial terminal on a Host PC are done through the Application HOST (the Nucleo board).

Between the Shield board and the Nucleo board, communication is made over UART via the Arduino connector. And between the Nucleo board and User Serial Terminal, the communication is made through Mini USB. This connection is only necessary for sending commands to and reading responses from the LoRa Software Modem.

Steps:

1. Connect the Shield board to the Arduino connector on the Nucleo board
2. Connect a Mini USB cable from the Host PC to the Nucleo board.
3. The Mini USB connection also provides power to both Shield board and Nucleo board
4. Make sure all four jumpers on the Shield board are arranged as in Figure 4

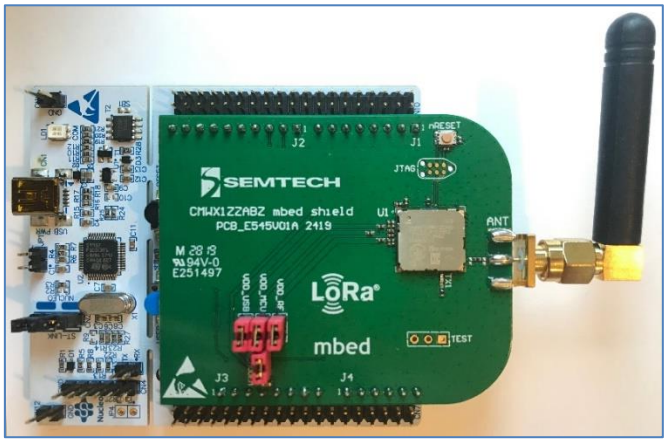

Figure 4 Connections between the LoRa Software Modem and the Application HOST

[3] https://www.python.org/downloads/release/python-374/

2.5 Getting the Host PC Ready

A COM terminal software on a Host PC is necessary to send commands to, and read from the Application processor or LoRa Software Modem. The tools commonly used for this communication are: Tera Term[4], Putty[5] or MobaXterm[6]. For this document, Tera Term is chosen as the example terminal.

To bring up the serial terminal on the Host PC:

1. Connect the Nucleo board USB port to the Host PC via a mini USB cable
2. Open the Tera Term tool
3. Select "Serial" and the COM port occupied by the Nucleo board
4. Under "Setup" select "Serial port…"
5. Configure the COM port connection to:
    a. Baud Rate: 115200
    b. Data Bits: 8
    c. Parity: None
    d. Stop Bits: 1
    e. Flow control: None
6. Under "Setup" select "Terminal"
7. Configure the terminal page as shown in Figure 12:

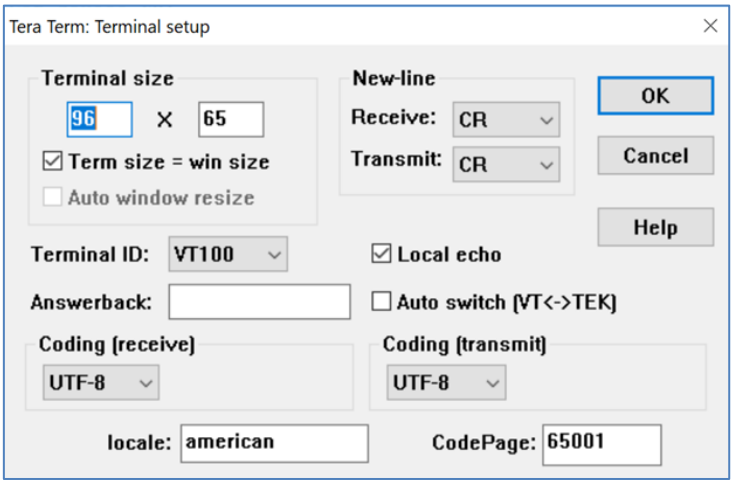

Figure 5 Terminal configuration example for TeraTerm

[4] Open source, more information available at: https://ttssh2.osdn.jp/index.html.en (careful of click bait)

[5] Open source, more information available at: https://www.putty.org/

[6] Free for personal use, more information available at: https://mobaxterm.mobatek.net/

2.6 Application Firmware on the Nucleo Board

The application firmware residing on the Nucleo board supports two modes of operation:

- Evaluation Mode: this mode is intended for evaluation of the API commands supported by the LoRa Software Modem and for radio conformance test.
- Demo Mode: this mode is intended for the demonstration of the LoRa Software Modem in a simple LoRaWAN temperature sensor application.

This application firmware can easily be loaded into the Nucleo board by following these steps:

1. Ensure that when the Nucleo board is connected to the Host PC, a temporary drive is immediatedly created. If the temporary drive doesn't show, then it's possible that the necessary ST-Link driver is missing. To obtain the latest driver, visit https://www.st.com/en/evaluation-tools/nucleo-l073rz.html#tools-software
2. Locate the application firmware bin file (ModemApp.bin) and drag-and-drop it into the Nucleo board temporary drive. If the application firmware bin file is not available, please download it from the following Github site: https://github.com/lwmodem/firmware

2.7 LoRa Software Modem API Commands

To access the Modem API commands, the LoRa Software Modem must be placed into the Evaluation Mode by following these steps:

a. Press and hold down the User button (blue) on the Nucleo board.
b. While holding down the User button, press and release the Reset button (black)
c. Continue to hold down User button for at least 5 seconds and release the User button

If done correctly, the serial terminal shoud look like the following:

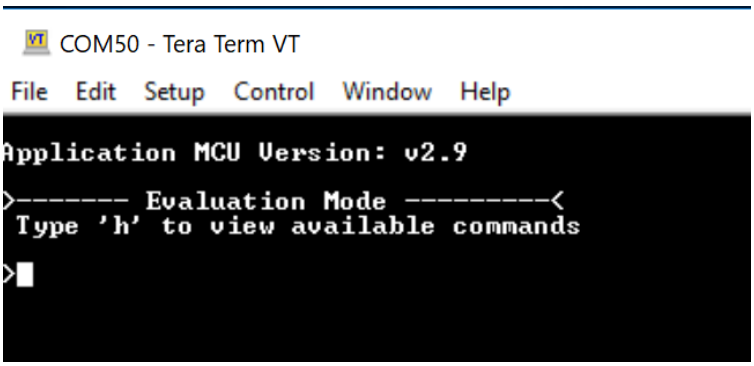

Figure 6. Serial Terminal in Evaluation Mode

Note: Commands are entered in ASCII format and are not case sensitive.

Figure 7 illustrates the list available commands from for the Validate application firmware. This list is accessible by typing h on the serial terminal.

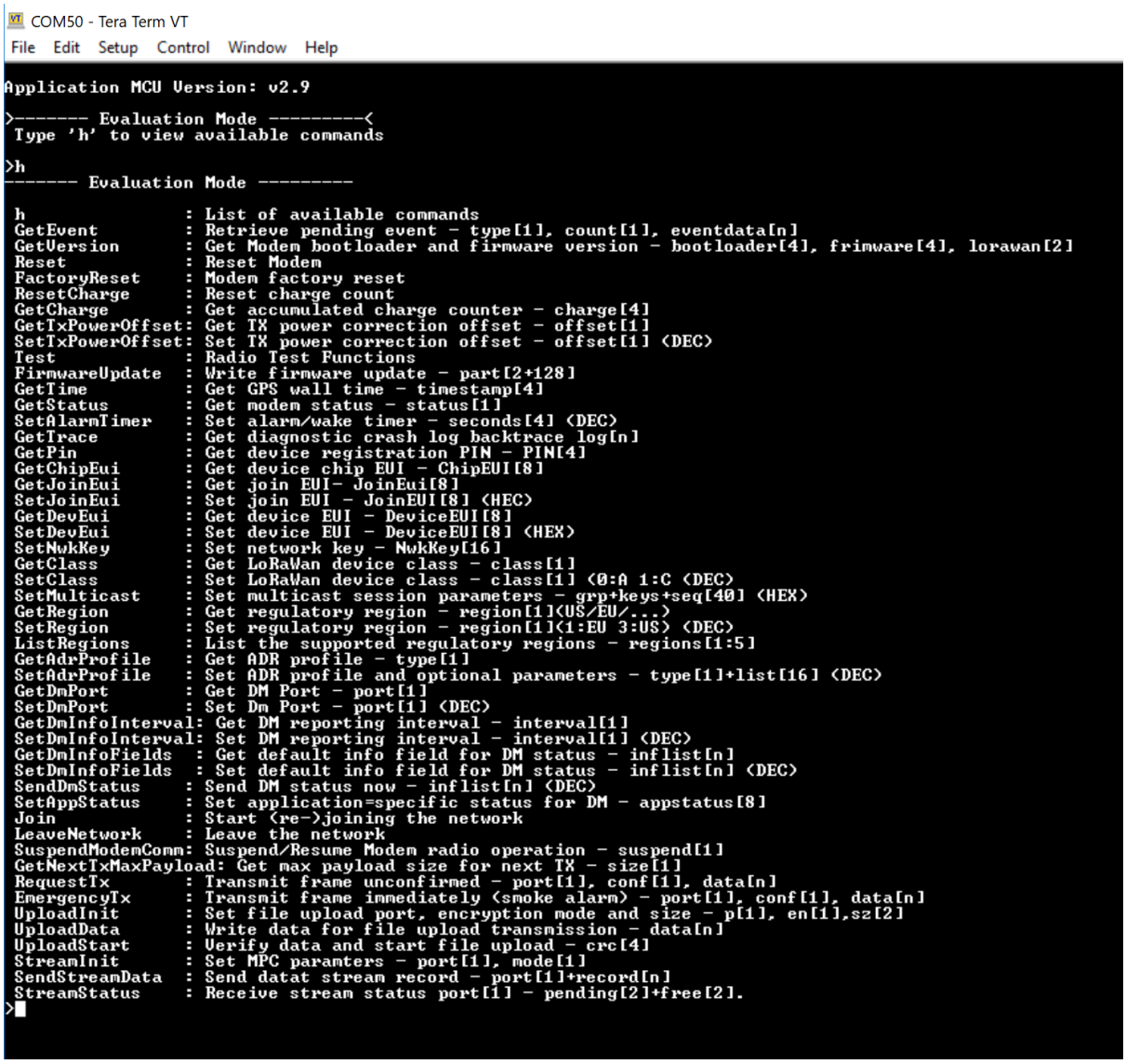

Figure 7. Supported Commands under Evaluation Mode

2.8 LoRa Software Modem Commands and Events

For convenience, the list of available LoRa Software Modem commands are listed in Table 1. For the latest updates and detailed information on all of the available commands, please refer to the LoRa Software Modem Command Reference Manual[7].

Note:

- A multi-byte parameter should be converted to a list of bytes, which is input by decimal number 0-255.
- The input parameter is LSB first. For example: “setalarmtimer 10 0 0 0” is the command to set the alarm to 10 seconds.

Table 1. List of LoRa Software Modem API Commands and Responses

Name	Code	Description	Input	Output
GetEvent	0x00	Retrieve pending events		type[1], count[1], eventdata[n]
GetVersion	0x01	Get version		bootversion[4], firmware version[4], lorawan[2]
Reset	0x02	Reset modem		
FactoryReset	0x03	Perform modem factory reset		
ResetCharge	0x04	Reset charge counter		
GetCharge	0x05	Get accumulated charge offset		Charge[4]
GetTxPowerOffset	0x06	Get power correction offset		offset[1]
SetTxPowerOffset	0x07	Set TX power correction offset	Offset[1]	
Test	0x08	Radio Test function	Refer to Reference Manual	Refer to Reference Manual
FirmwareUpdate	0x09	Write firmware update	part[4+128]	
GetTime	0x0A	GetGPS wall time		timestamp[4]
GetStatus	0x0B	Get modem status		status[1]
SetAlarmTimer	0x0C	Set alarm / wakeup timer	seconds[4]	
GetTrace	0x0D	Get diagnostic crash log		backtrace log[n]
GetPin	0x0E	Get device registration PIN		PIN[4]
GetChipEui	0x0F	Get device chip EUI		ChipEUI[8]

[7] https://github.com/lwmodem/firmware

GetJoinEui	0x10	Get join EUI		JoinEUI[8]
SetJoinEui	0x11	Set join EUI and derive keys	JoinEUI[8]	
GetDevEui	0x12	Get device EUI		DeviceEUI[8]
SetDevEui	0x13	Set device EUI and derive keys	DeviceEUI[8]	
SetNwkKey	0x14	Set network key	NwkKey[16]	
GetClass	0x15	Get the LoRaWAN device Class		class[1]
SetClass	0x16	Set LoRaWan class A/C	class[1]	
SetMulticast	0x17	Set multicast session parameters	param[40]	
GetRegion	0x18	Get regulatory region		region[1]
SetRegion	0x19	Set regulatory region	region[1]	
ListRegions	0x1A	List the supported regulatory regions		region[1-5]
GetAdrProfile	0x1B	Get ADR profile		type[1]
SetAdrProfile	0x1C	Set ADR profile and optional parameters	type[1]+list[16]*	
GetDmPort	0x1D	Get DM port		port[1]
SetDmPort	0x1E	Set DM port	port[1]	
GetDmInfoInterval	0x1F	Get DM reporting interval		interval[1]
SetDmInfoInterval	0x20	Set DM reporting interval	interval[1]	
GetDmInfoFields	0x21	Get default info field for DM status		inflist[n]
SetDmInfoFields	0x22	Set default info field for DM status	inflist[n]	
SendDmStatus	0x23	Send DM status now	inflist[n]	
SetAppStatus	0x24	Set application-specific status for DM	appstatus[8]	
Join	0x25	Start joining the network		
LeaveNetwork	0x26	Leave the network		
SuspendModemComm	0x27	Suspend/resume radio operations	suspend[1]	
GetNextTxMaxPayload	0x28	Get max payload size for next TX		size[1]
RequestTx	0x29	Transmit frame unconfirmed or confirmed	port[1], conf[1], data[n]	
EmergencyTx	0x2A	Transmit frame immediately (smoke alarm)	port[1], conf[1], data[n]	

UploadInit	0x2B	Set file upload port, encryption mode, size	p[1],en[1],sz[2]	
UploadData	0x2C	Write data for file upload transmission	data[n]	
UploadStart	0x2D	Verify data and start file upload	crc[4]	
StreamInit	0x2E	Set data stream parameters	port[1],mode[1]	
SendStreamData	0x2F	Send data stream record	port[1]+record[n]	
StreamStatus	0x30	Retrieve stream status	port[1]	pending[2]+free[2]

The LoRa Software Modem is capable of sending an asynchronous event to the application processor. This event will also be displayed on the serial terminal.

Table 2. List of the Events

Event	Code	Description	Output
Reset	0x00	Modem has been reset	reset count[2]
Alarm	0x01	Alarm timer expired	
Joined	0x02	Network successfully joined	
TxDone	0x03	Frame transmitted	status[1]
DownData	0x04	Downlink data received	port[1], downdata[n]
UploadDone	0x05	File upload completed	status[1]
SetConf	0x06	Config has been changed by DM	info tag[1]
Mute	0x07	Modem has been muted or unmuted by DM	mute[1]
StreamDone	0x08	Data Stream Fragments Sent	
LinkStatus	0x09	Netwrok connectivity status changed	status[1]
JoinFail	0x0A	Attempt to join network failed	

2.8.1 Sending Commands

Commands will be constructed with the corresponding opcode and parameters to the LoRa Software Modem board through the UART.

The general command format is:

“command” “parameters”

- The “command” is the name of operations, for example, “join” or “requesttx”
- The “parameters” is a list of the parameters sent to the LoRa Software Modem. If it is a hex string, it is converted raw bytes. Space must be added between each byte.
  For example, we can use the command “requesttx 100 0 1 2 3 4” to send an uplink frame of using FPort 100, no acknowledge and 4 bytes of data (1, 2, 3, and 4).

Two types of commands are supported

- Synchronous command: commands sent to the LoRa Software Modem with an instant response expected, the response time should be within 50 ms.

- Asynchronous command: commands sent to the LoRa Software Modem with an additional non-instant response expected. In this case, response is trigger by the LoRa Software Modem through the interrupt pin EVENT.

For example, when an asynchronous command like “setalarmtimer” is sent to the LoRa Software Modem, there will be no immediated response from the LoRa Software Modem after the command has been sent. A response will only be sent by the LoRa Software Modem after the command has been executed, in which case the interrup EVENT pin will be pulled high.

Proper network coverage is required so that the LoRa Software Modem can join the network, send uplink frames and receive downlink frames.

2.8.2 Reading Responses

On the console, if a response message is available, the Application HOST will pull the data, validates the checksum, and display the response data and status.

A successful synchronous command sending and response is shown in Figure 8.

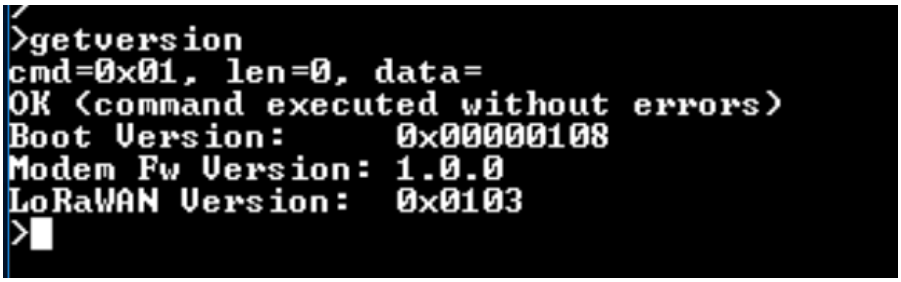

Figure 8. Message Exchange for Info Command

Sometimes the terminal displays “NO RESPONSE FROM MODEM”. This indicates the LoRa Software Modem is busy and not ready to process the request within the pre-defined 200ms timeout.

2.9 Retrieving LoRa Software Modem Information

To provision the Lora Software Modem to the LoRaWAN network server, the unique AppKey of the device must be obtained from the LoRa Cloud Device Join Server. And in order to do that, the DEVEUI, CHIPEUI, JOINEUI, and PIN must be retrieved from the module using the following commands:

1. Enter “getdeveui” – This command will get 8 bytes of Dev EUI from LoRa Software Modem.
2. Enter “getchipeui” – This command will get 8 bytes of Chip EUI from the LoRa Software Modem.
3. Enter “getpin” – This command will get 4 bytes of Dev PIN from the LoRa Software Modem.
4. Enter “getjoineui” – This command will get 8 bytes of Chip EUI from the LoRa Software Modem.

These three keys and pin are then entered into the LoRa Cloud Device Join Server to obtain the AppKey.

3 LoRa Cloud and First Application Setup

3.1 LoRa Cloud Setup

To open an account on LoRa Cloud, follow these steps :

1. Open https://www.loracloud.com/ in a browser
2. Click on "LOGIN or REGISTER" link

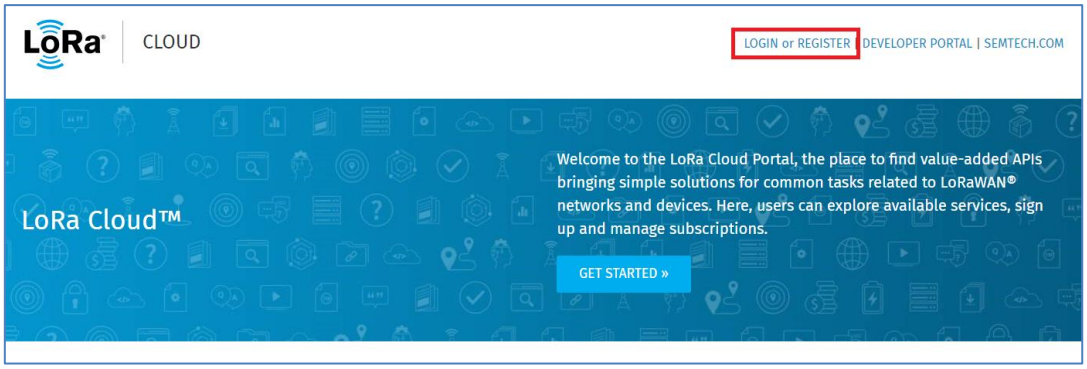

Figure 9. LoRa Cloud Account Creation Step 2

3. Click on "Sign Up" link
4. Fill the required fields and validate your email address

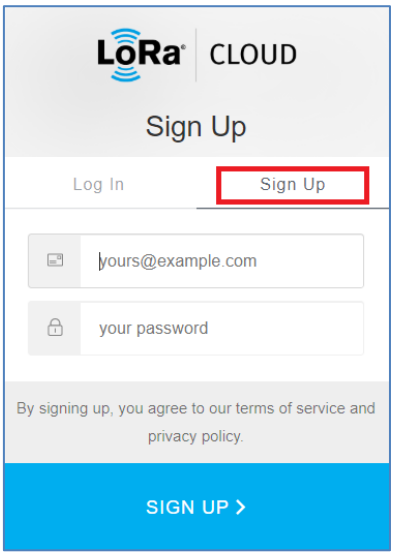

Figure 10. LoRa Cloud Account Creation Step 3

3.2 Device Join Server

There are two ways to use the LoRa Cloud Device Join Server, depending if your network server is listed in https://www.loracloud.com/portal/join_service/network_servers .

1. The LoRaWAN network server is in the list, so it is linked to the LoRa Cloud join server: users only needs to claim the LoRa Software Modem in the LoRa Cloud join server and keys will be exchanged in a secured way between the LoRaWAN network server and the LoRa Cloud join

server. If users do not want to disclose the AppKey to the LoRaWAN network server, they should set a wrapping algorithm and provide its associated key in the LoRa Cloud join server[8]

2. The LoRaWAN network server is not in the list, so it is not known by the LoRa Cloud join server: users have to claim the LoRa Software Modem, then export the AppKey from the LoRa Cloud join serverto be able to register the LoRa Software Modem against the LoRaWAN network server. This way there is no communication between the LoRaWAN network server and LoRa Cloud join server, the key exchange is only between the LoRaWAN network server and the device

To claim a device it is required to get the DevEUI, ChipEUI, JoinEUI and PIN code from the LoRa Software Modem (see Section 2.9 Retrieving LoRa Software Modem Information), and then follow the procedure provided in the following link to claim the device and obtain the AppKey from the LoRa Cloud join server. The AppKey will be used in Section 3.4.2 Edit Application EUI: https://www.loracloud.com/documentation/join_service?url=quickstart.html.

Note that if the network server is not known by the join server, follow the end of the procedure until AppKey is exported. If the LoRaWAN network server is known by the LoRa Cloud join server, follow these steps:

1. Go to https://loracloud.com, click on “LoRa Cloud™ Device Join” then “NETWORK SERVERS”

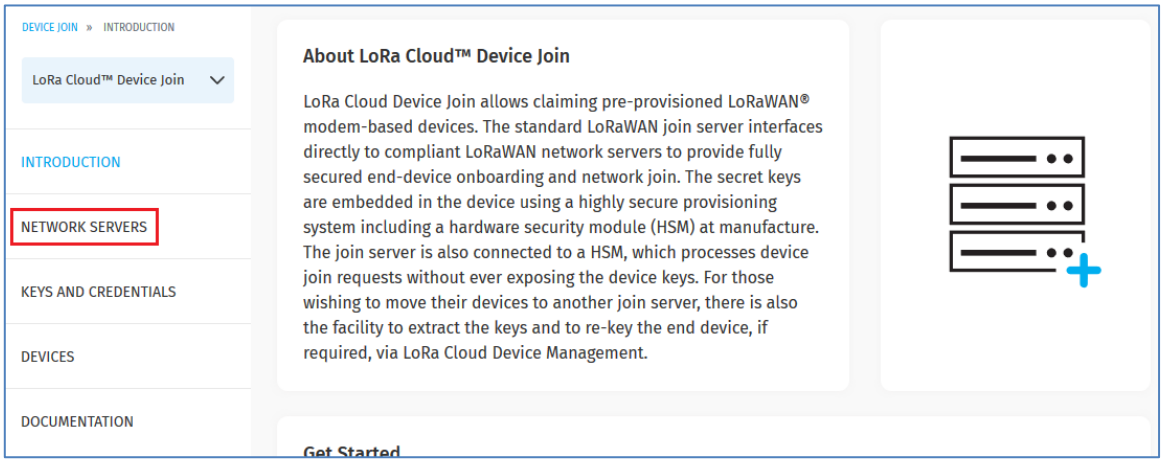

Figure 11. LoRa Cloud Join Server – NS Selection Step 1

[8] https://www.loracloud.com/documentation/join_service?url=appskey.html

2. Select network server in the drop down list, ADD and SAVE NETWORK SERVERs

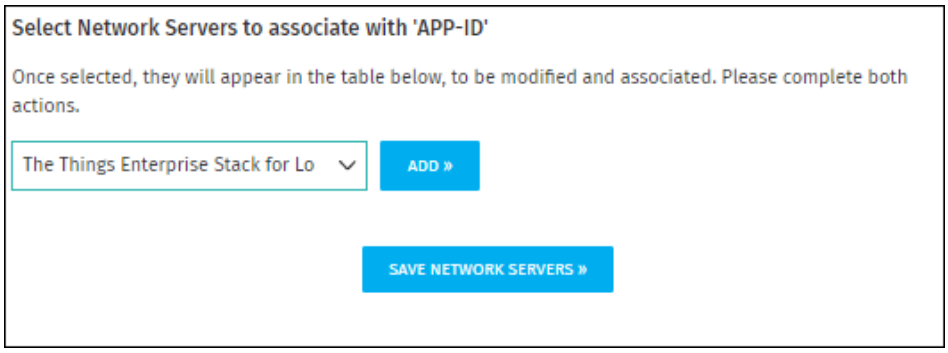

Figure 12. LoRa Cloud Join Server- LoRaWAN Network Server Selection Step 2

The LoRa Cloud join server is now ready to handle join requests coming from the LoRaWAN network server

Note: A single LoRA Cloud join server account can be associated with multiple LoRaWAN network servers at a time.

3.3 Device Management Server

The procedure in the following link explains how the LoRa Software Modem is registered in the Device Management Server:
https://www.loracloud.com/documentation/device_management?url=gettingstarted.html

3.4 Network Server

The device must be registered on a network server. Any network server can be used, but for this example The Things Network will be used.

1. If needed, create an account at https://account.thethingsnetwork.org/register.
2. Login at http://console.thethingsnetwork.org/

3.4.1 Creating an Application

1. Create an Application by “Applications”

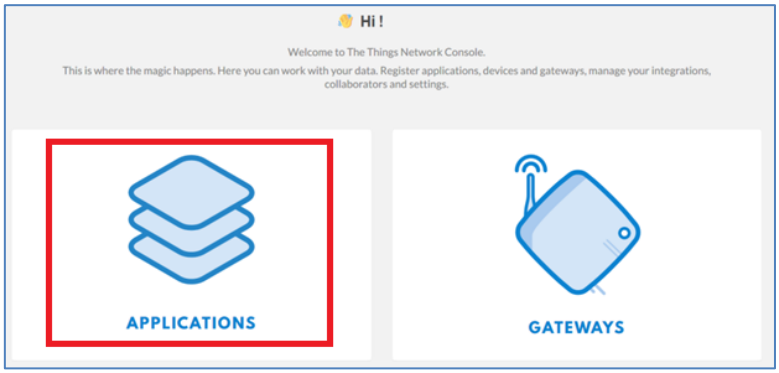

Figure 13. TTN Application - Step 1

2. Click on “Add Application”

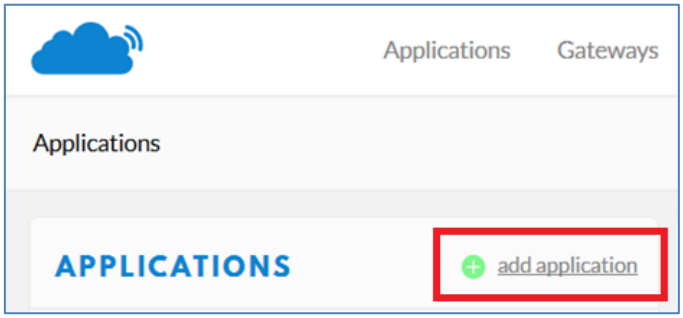

Figure 14. TTN Application - Step 2

3. Fill in the Application ID, Description, and Handler Registration
4. Click Add Application

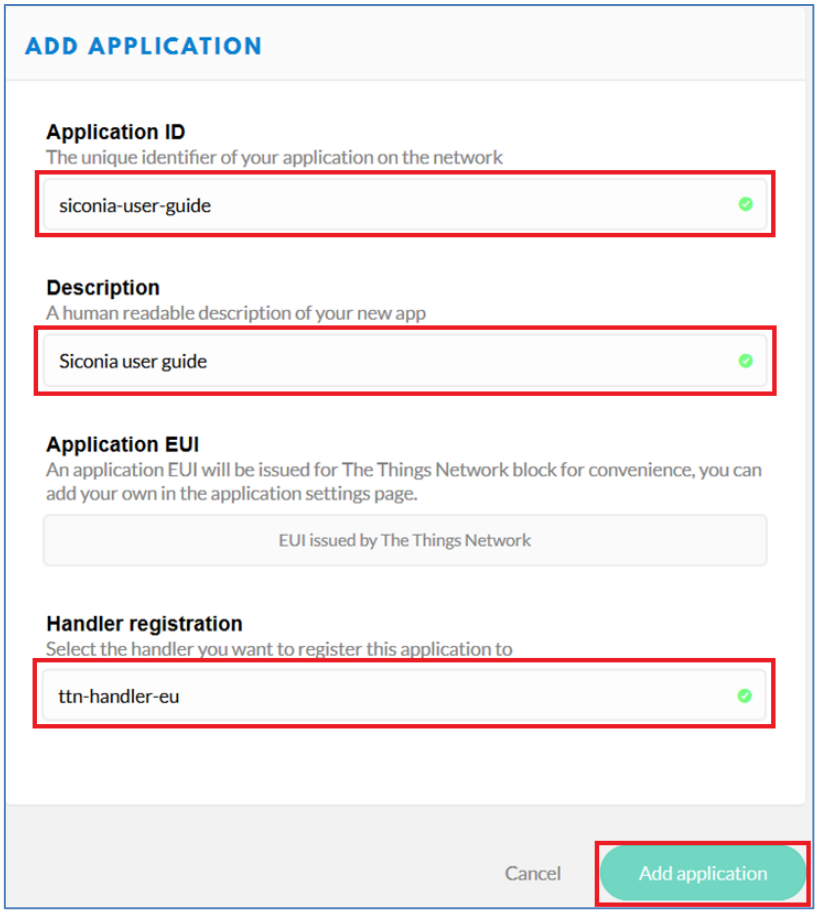

Figure 15. TTN Application - Step 3

3.4.2 Edit Application EUI

TTN automatically assigned an Application EUI for the Application. The Application EUI must be changed to match the joinEUI retrieved from the LoRa Software Modem. Follow these steps to do so:

1. Under the Application landing page, click on “Settings”

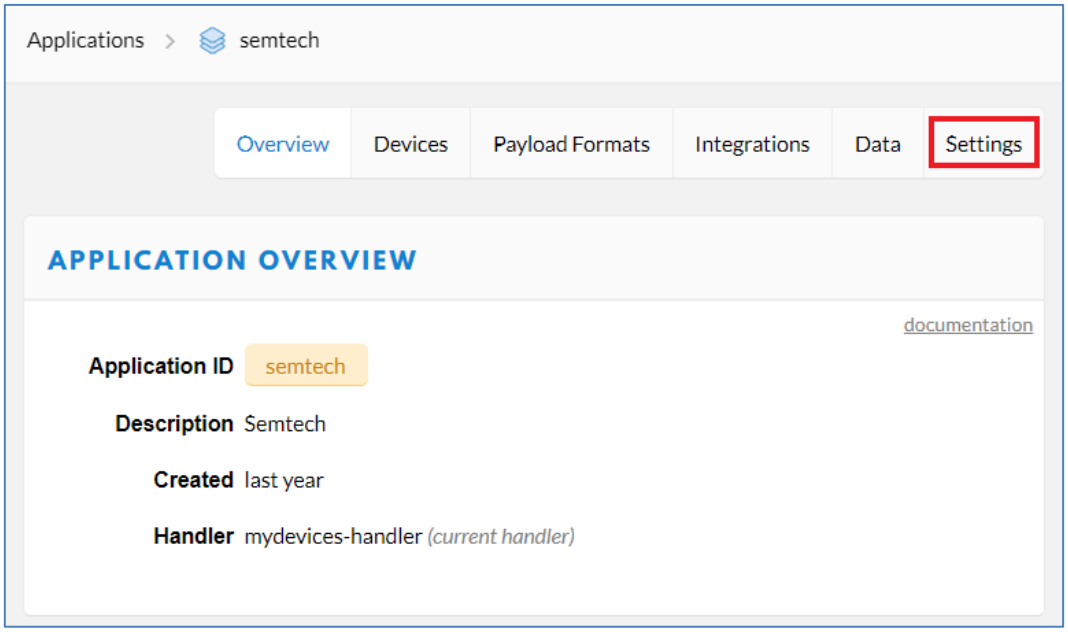

Figure 16. TTN Application EUI – Step 1

2. Under the Settings page, select “EUIs” and then click on “add EUI”

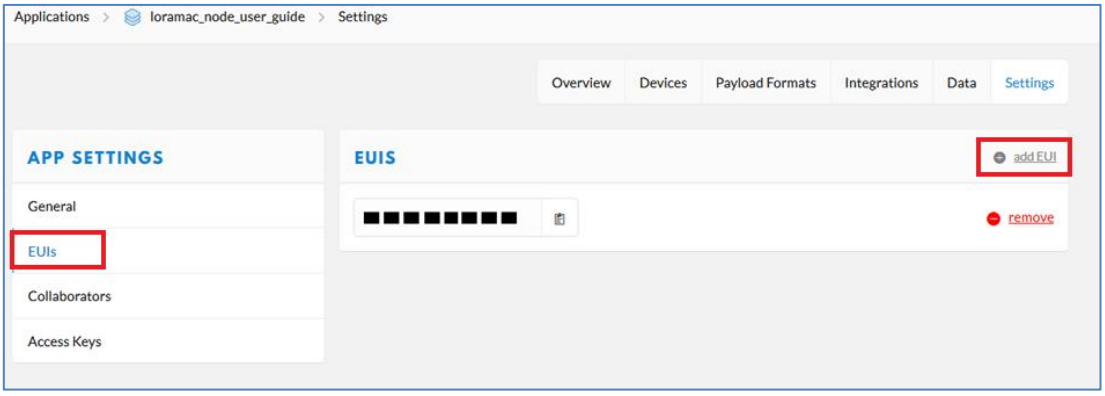

Figure 17. TTN Application EUI – Step 2

3. Under the EUIS pop-up window, click on the customize pencil
4. Enter the joinEUI key retrieved from the LoRa Software Modem
5. Click on “Add EUI” to complete

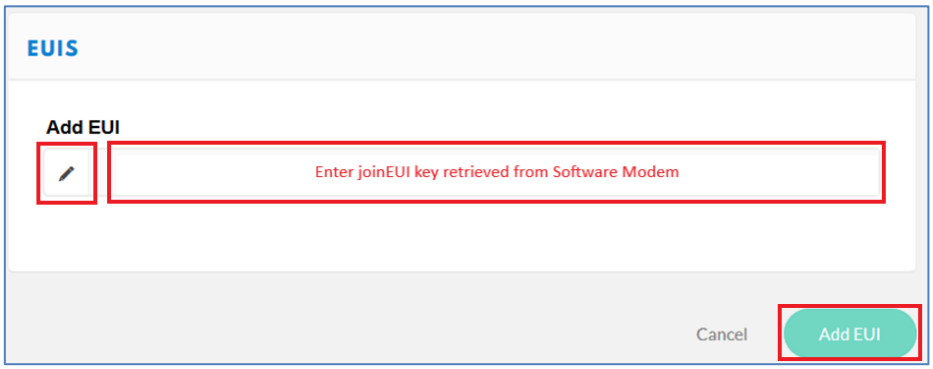

Figure 18. TTN Application EUI – Step 3

3.4.3 Provision End-Node to Network Server

1. Under the “DEVICES” section of the Application page, click on “register device”

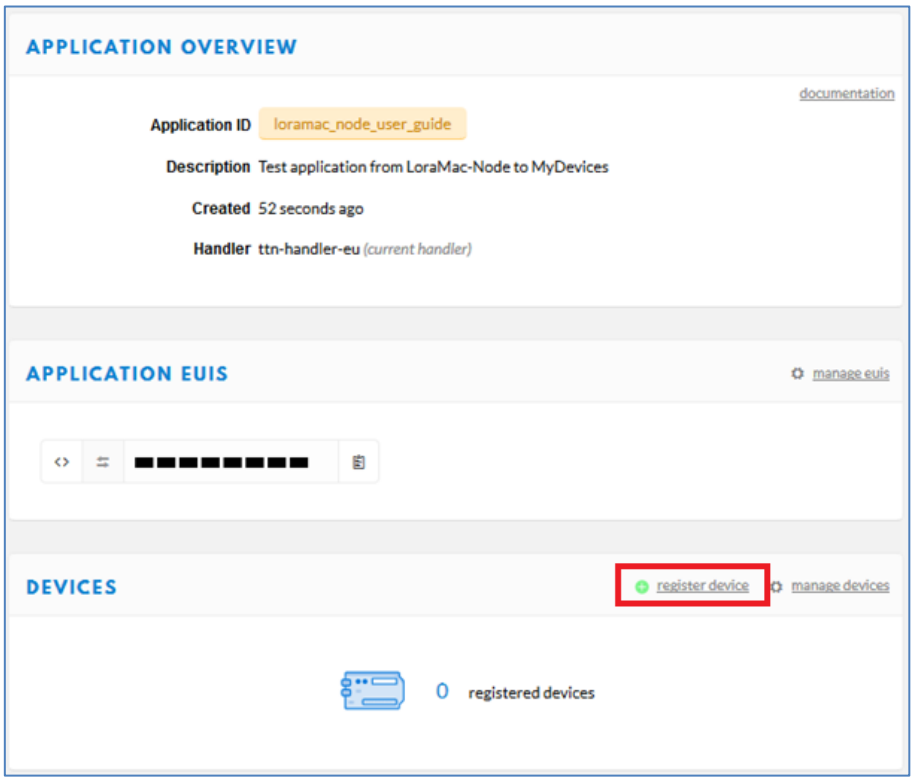

Figure 19. TTN Add Device - Step 1

2. Enter a name or description into the Device ID field
3. Enter the DevEUI retrieved from the LoRa Software Modem into the Device EUI field
4. Enter the AppKey retrieved from the LoRa Cloud Device Join Server into the App Key field
5. Select the correct App EUI (matching JoinEUI key) into the App EUI field
6. Click on “register device” to provision the device.

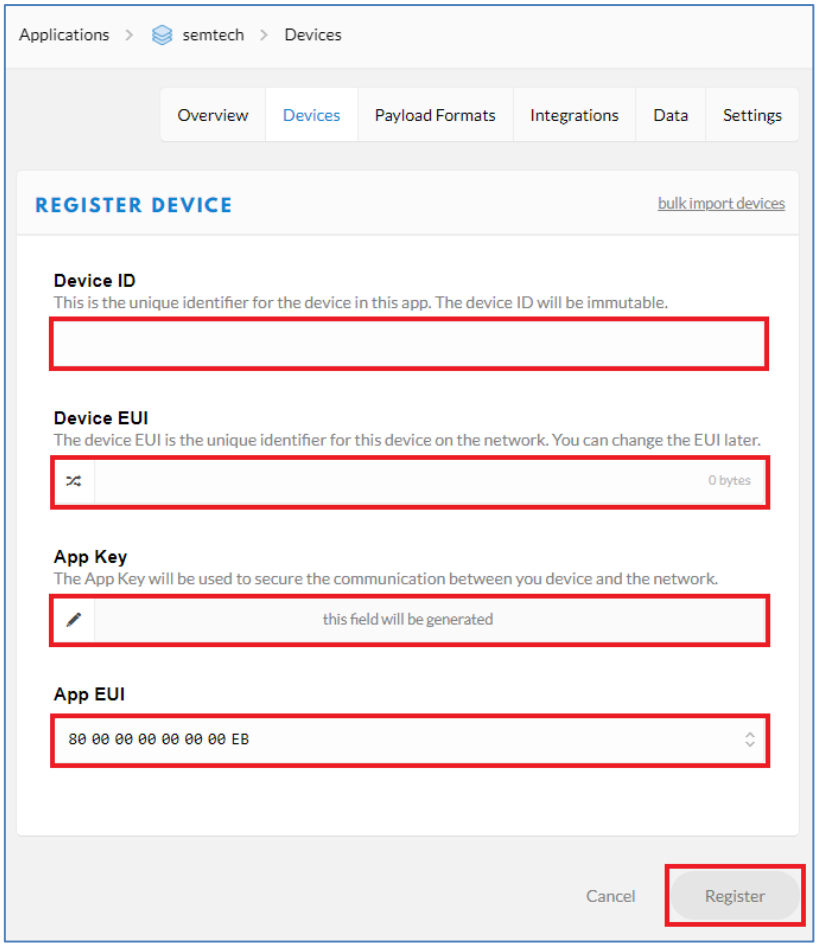

Figure 20. TTN Add Device – Step 2

Now that everything is setup, the LoRa Software Modem should be able to join the network server.

3.4.4 Network Server Coverage

To ensure the operating frequencies of the LoRa gateway matches the frequency plan supported by TTN, visit https://www.thethingsnetwork.org/docs/lorawan/frequency-plans.html.

3.5 Connect LoRa Software Modem to LoRa Network Server

In-order to connect the LoRa Software Modem to LoRaWan Network Server, there are serier of commands needed to enter in a following order:

1. Place the LoRa Software Modem into Evaluation Mode.
    a. Press and hold down the User button (blue) on the Nucleo board.
    b. While holding down the User button, press and release the Reset button (black)
    c. Continue to hold down User button for at least 5 seconds and release the User button
2. Enter “listregions” to get a list of region supporting by the LoRa Software Modem.
3. Enter “getregion” to obtain current region setting.
4. Enter “setregion 3” to set region to US915 or “setregion 1” to set region to EU868.
5. Enter “getregion” to confirm desired region.

6. Enter “join” to command the LoRa Software Modem to send a join request to the Network Server. When the LoRa Software Modem successfully joined the Network Server, a message “Network successfully joined” will be displayed on the serial terminal.
    a. Before entering “join” command, the LoRa Software Modem must be provisioned to the Network Server. If not, refer to Sections 3.3 and 3.4.
    b. While waiting for the network server to accept the join request, no command will be accepted by the LoRa Software Modem.
    c. To confirm the device has successfully joined the Network Server, visit the Device section under the TTN Console (https://console.thethingsnetwork.org/applications/)
7. Enter “requesttx 102 0 72 69 76 76 79” to send a string “HELLO” message to the LoRaWAN Network Server. Note that the command only can be entered after the LoRa Software Modem has successfully joined to the LoRa Network Server, and the command payload must be in decimal value.
8. To view the “HELLO” message received by the TTN LoRaWAN network server, go to https://console.thethingsnetwork.org/applications/ to view all uplink and downlink messages at the TTN LNS.

3.6 Sample Temperature Application

One of the two applications supported by the Application firmware residing on the Nucleo board is the Demo Mode. This demo mode is a simple temperature application that was developed to illustrating the ease of setting up an end-node to cloud demo. This application can easily be activated by:

1. Ensure the application firmware is properly loaded on the Nucleo board by pressing the Reset button (black). Upon pressing on the Reset button, the serial terminal should display the following. If not, refer back to Section 2.5.

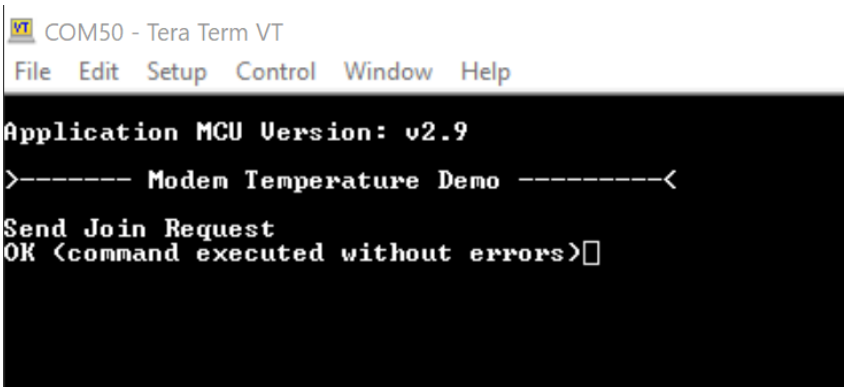

Figure 21. Serial Terminal when Modem in Temperature Application in US915

2. If the application is successfully executed after the Reset button (black) is pressed, te LoRa Software Modem will automatically join the network server. After it has successfully joined, the device will send temperature to the network.
3. Prior to visualizing the temperature data, an Application Server must first be set up as described in Section 3.7. Node-RED is used in this sample application.

3.7 Application Server (Node-RED)

The application server needs to handle application data and device management frames: there is no background connection between LNS and DAS, all DM uplink frames have to be forwarded by the application server to the DAS.

The demo presented in this document uses Node-RED[9] as the Application server.

To install Node-RED and the necessary dashboard, following these general steps:

1. Download the “Recommended For Most Users” Node.JS JavaScript runtime from nodejs.org
2. Install Node.JS on the Host PC
3. Open a Windows Command Prompt on the Host PC
    a. To install Node-RED type: npm install -g --unsafe-perm node-red
    b. To install Node-RED Dashboard type: npm install node-red-dashboard

Next, the pre-defined flow provided in Annex 1 must be imported by following these steps:

1. Open a new Windows Command Prompt and type in node-red to execute Node-RED.
2. Open a Windows internet browser and type in: localhost:1880
3. Click on the “three horizontal bar” button on the upper right corner, select “Import”, and select “Clipboard”.
4. Open the Nodered_modem_as_v0.3.json file from the following Github site: https://github.com/lwmodem/firmware
5. Copy and paste the contents in the Nodered_modem.json into the clipboard and click on “Import” as shown in Figure 22.

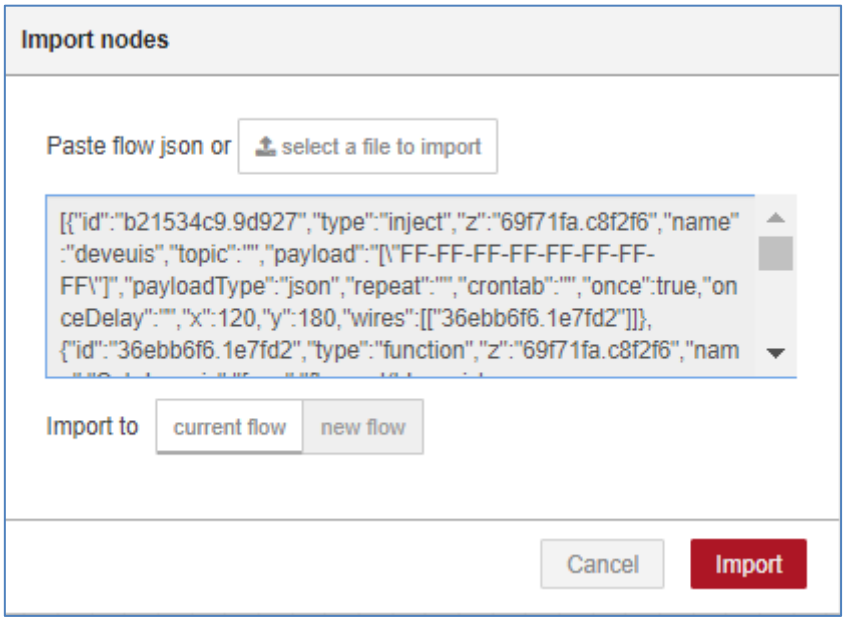

Figure 22. Node-RED Import Clipboard

[9] Graphical prototyping tool based on JavaScript, installation guides: https://nodered.org/docs/getting-started/

If done correctly, the Node-RED flow should look like Figure 23.

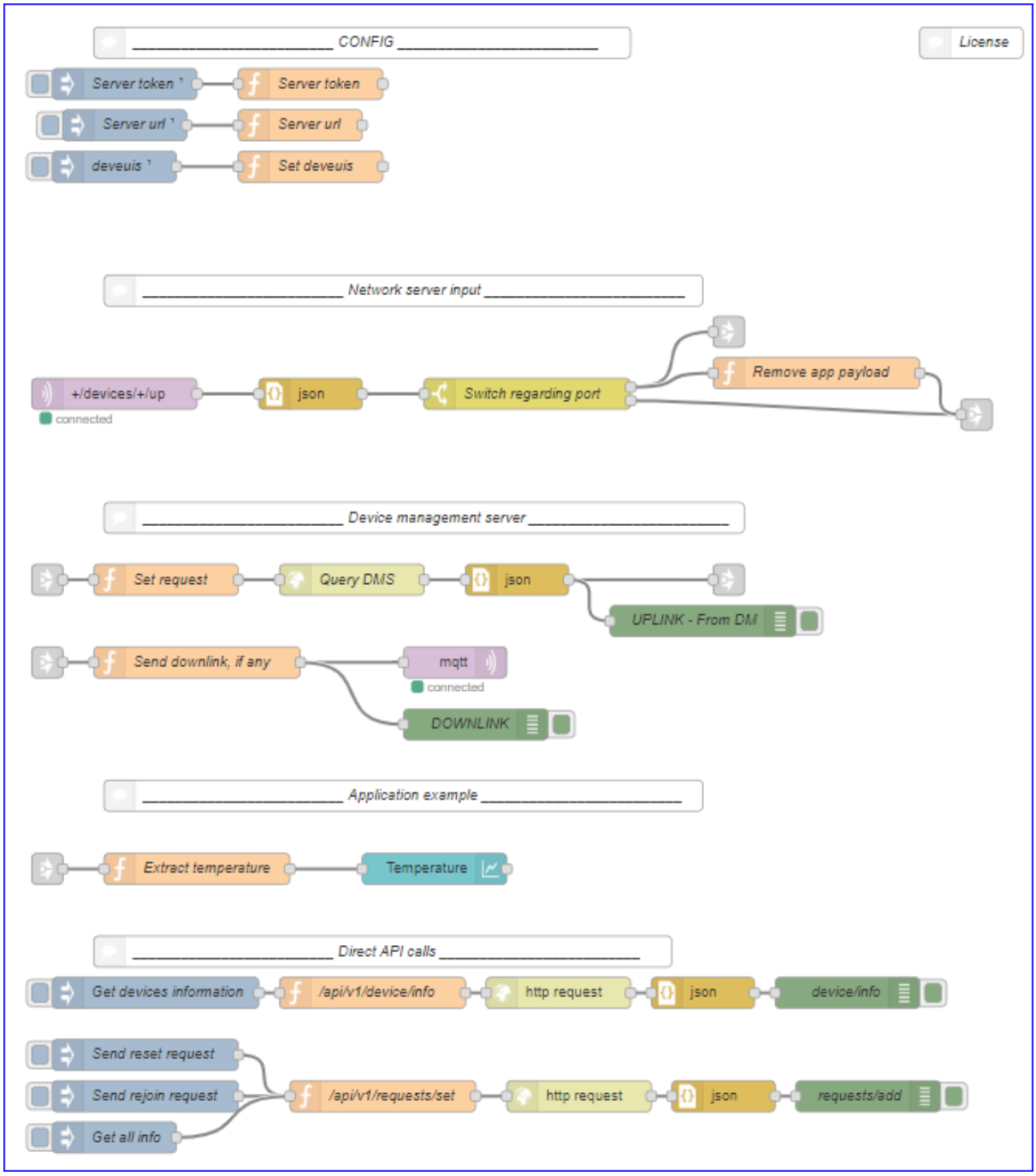

Figure 23 - Node-RED application server

3.7.1 Configure the Flow

Configure “Server token”, “Server url”, and “deveuis” by double clicking on the blue boxes:

3.7.1.1 Server Token

1. Open a browser and go to this page: https://www.loracloud.com/portal/device_management/tokens
2. Under LoRa Cloud Device Management page, click on Token Management
3. Create new Device Owner by entering Name under “New Owner” and click on “Add New Owner>>”

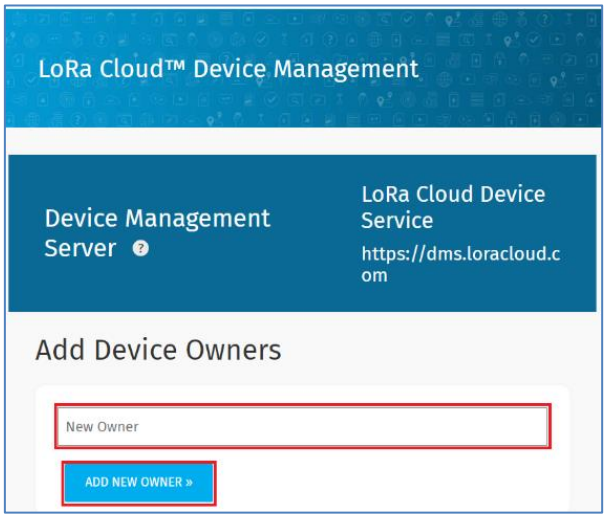

Figure 24. AS – Server Token Step 3

4. Under the “Device Owner” page, click on “Add New Token>>” to add Device ID

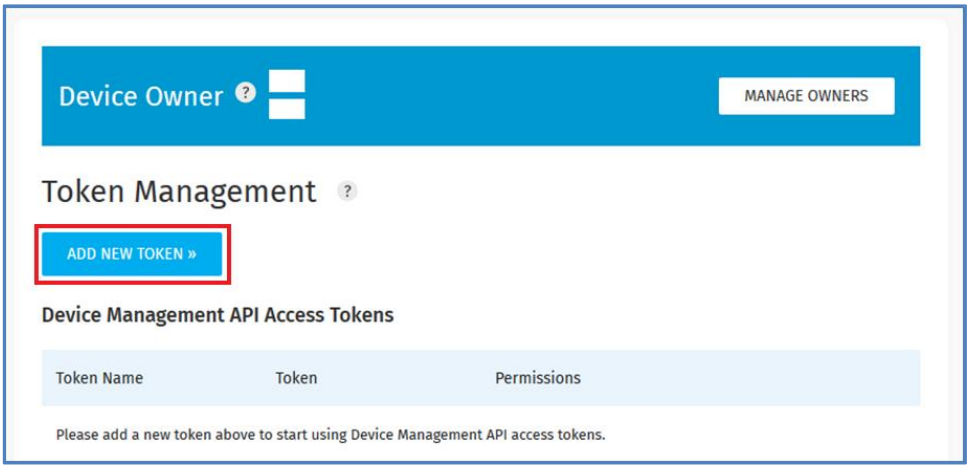

Figure 25. AS – Server Token Step 4

5. Fill in the desired Token Name, Select at least three Permissions, and click “Add New Token”

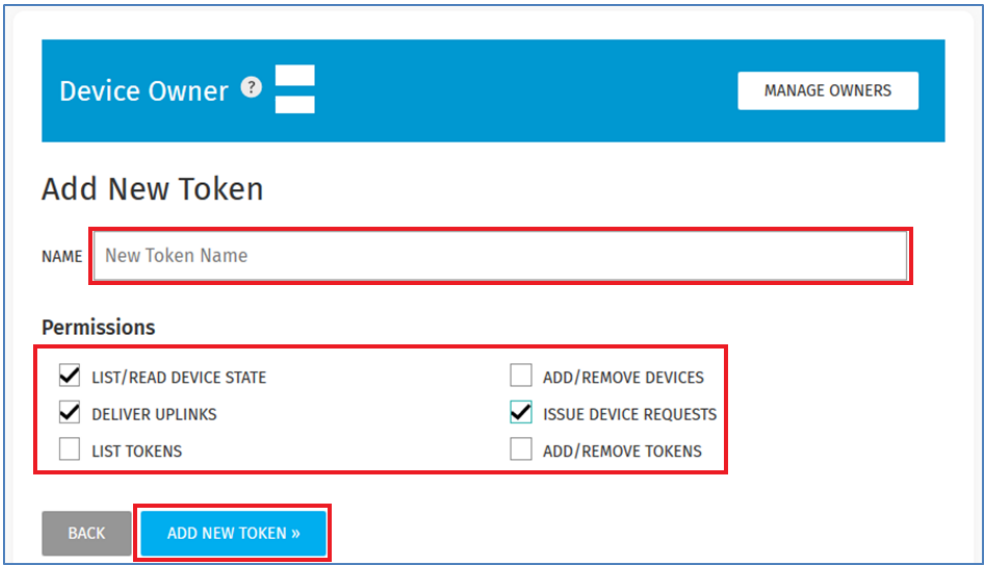

Figure 26. AS – Server Token Step 5

6. Copy the Token by clicking on the “Copy” button

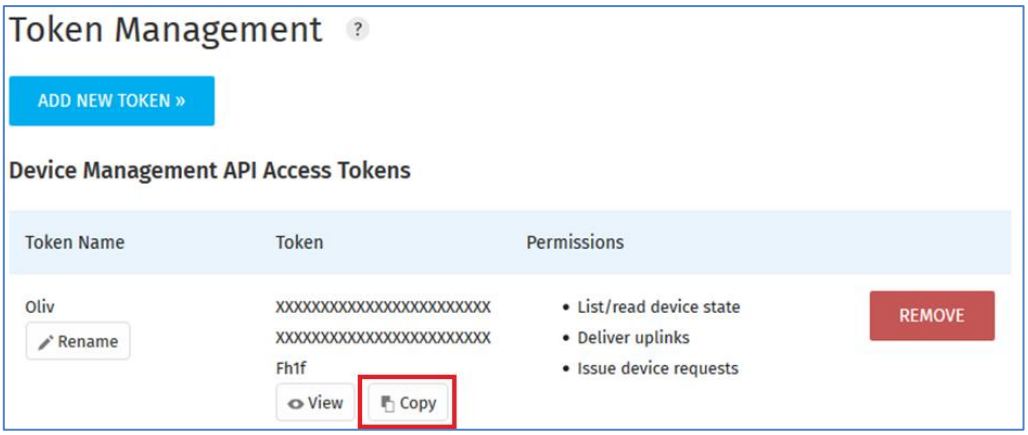

Figure 27. AS – Server Token Step 6

7. Return to Node-RED and Paste the Token into the “Payload” field and hit “Done”

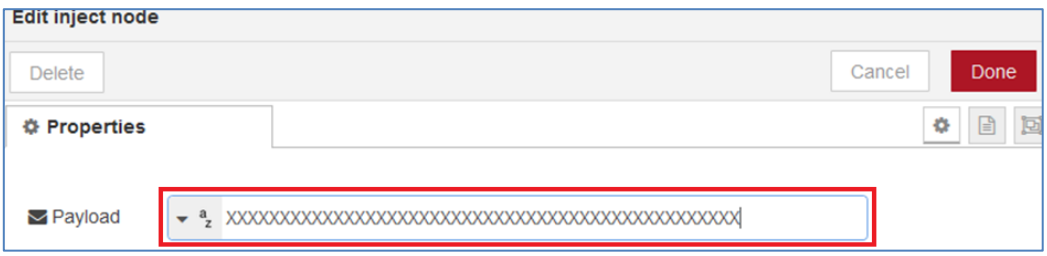

Figure 28. AS – Server Token Step 7

3.7.1.2 Server URL

1. Enter the https://dms.loracloud.com URL into the Payload field and hit “Done”

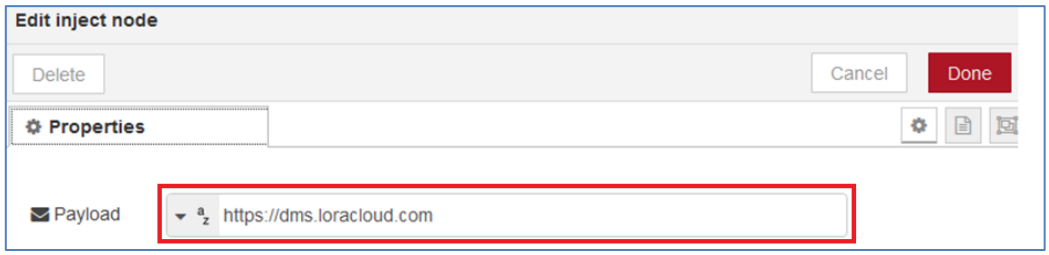

Figure 29. AS – Server URL

3.7.1.3 DevEUI

Figure 30. AS – DevEUI

3.7.1.4 Network Server Input

1. Double click on the mqtt input box under the Network Server Input.

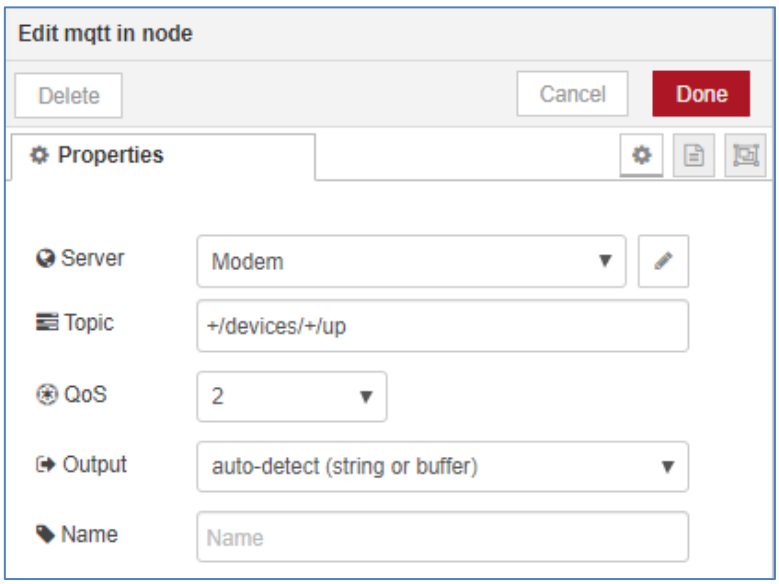

Figure 31. AS – Network Server Property Settings

2. Click on the pencil symbol next to the “Server” field
3. Under Connection tab, make sure the server address points to the correct LoRaWAN network server

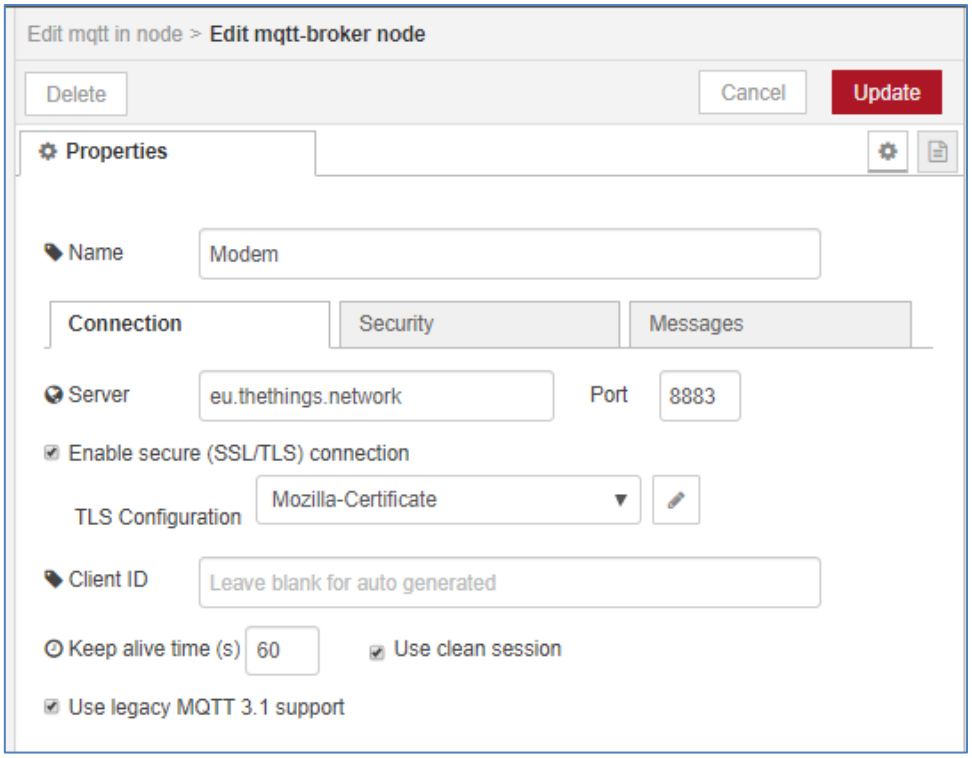

Figure 32. AS – Network Server Connection URL

4. Under the Security tab, enter the network server Application ID into the Username field, and Access Key into the Password field.

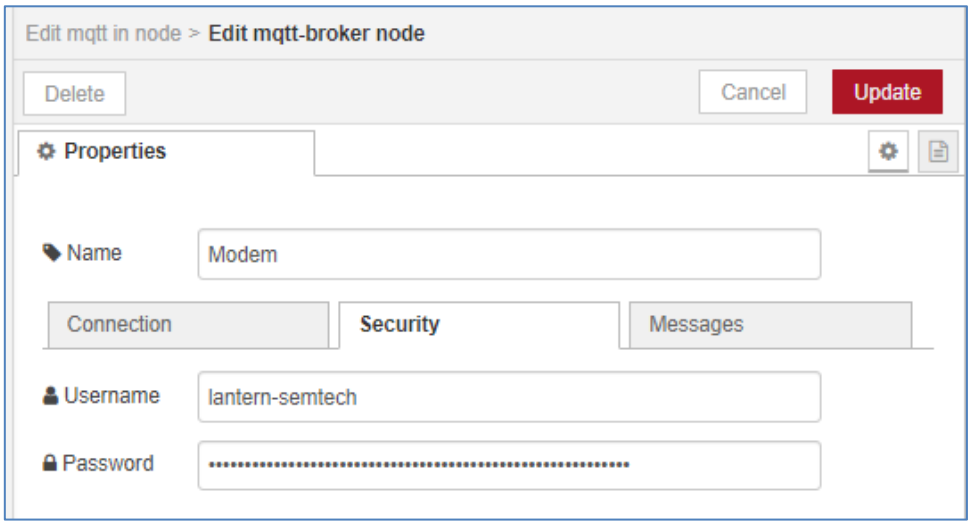

Figure 33. AS – Network Server Application Name and Access Code

Once all of the above are done, click on Node-RED button “Deploy” to validate the changes. If done correctly, there should be a green square with the word Connected under the Network Server Input.

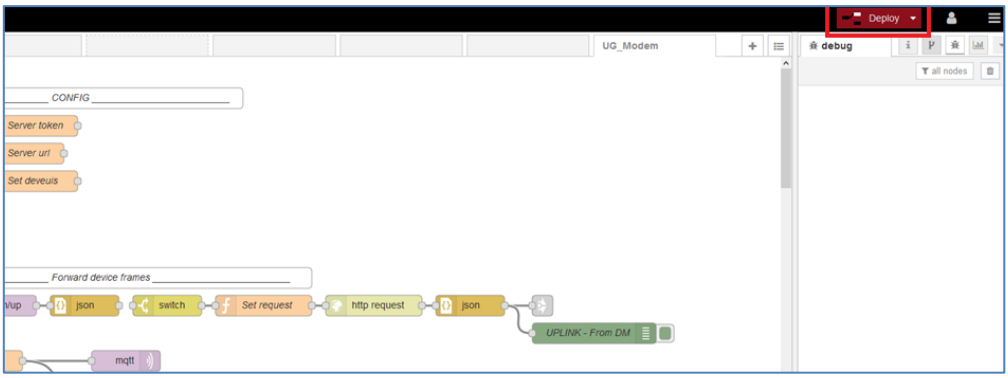

Figure 34. AS – Deploy Flow

Lastly, open a Windows Command Prompt and type in node-red to execute Node-RED application.

Note: It is recommended to send all uplinks to the DAS, payload can be removed for any privacy concerns. It will allow DAS to have more downlink opportunities and accurate statistics

3.8 Explore Data

Congratulations, everything is set up!

3.8.1 Temperature Graph

Go to the Node-RED ui (http(s)://yourip/ui) to see incoming temperatures plots in the graph

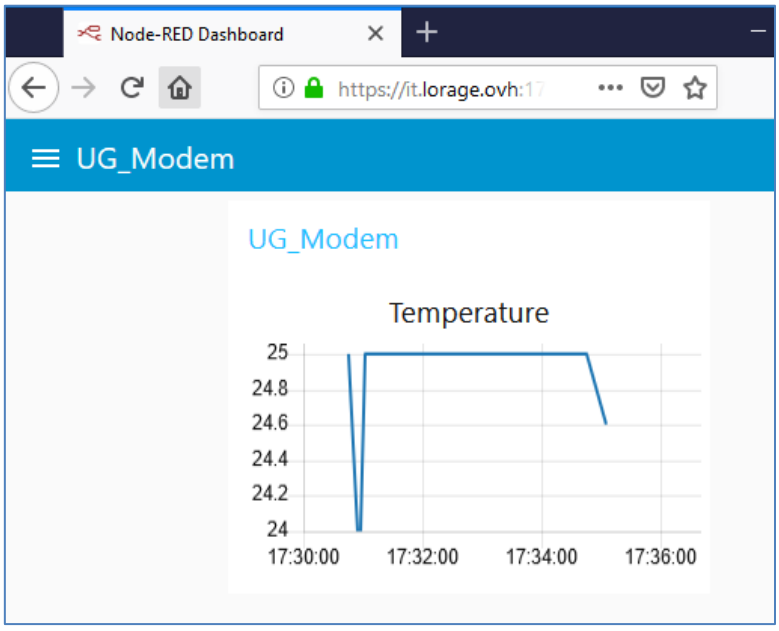

Figure 35. Temperature Graph

3.8.2 Device & Application Services

1. Go to LoRa Cloud by clicking on https://www.loracloud.com/portal/device_management
2. Under LoRa Cloud Device Management page, click on “Devices”
3. Click on the arrow on the right to see all available device data

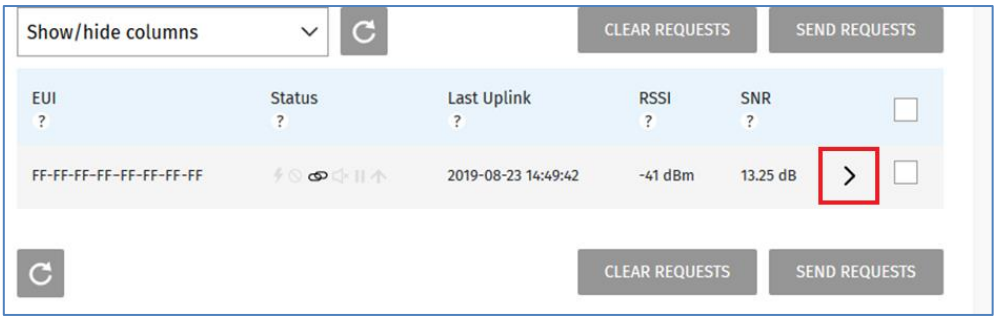

Figure 36. Device Management Data

A modal window will open with add device data known by the DAS

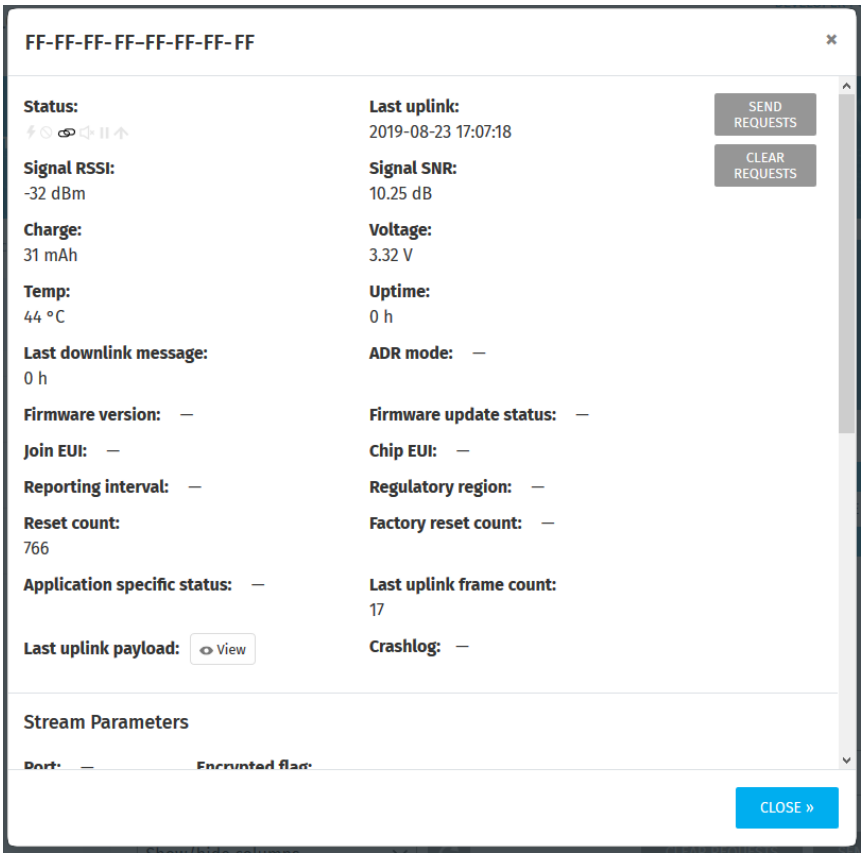

Figure 37. Device Management – Device Window

3.9 Send a Request to the LoRa Software Modem

Request can be sent to the LoRa Software Modem for several purposes, this chapter will explain how to send a reset request to the LoRa Software Modem.

1. A request is added to the queue thanks to DAS API
2. On next device management frame from the LoRa Software Modem, it is sent to the DAS by the application server.
3. The DAS will answer the downlink payload to be send to the LoRa Software Modem.
4. The application server will queue this downlink in NS
5. On next opportunity, the NS will send the downlink to the LoRa Software Modem
6. The LoRa Software Modem will reset!

It can be tested in Node-RED thanks to the “Send reset request button” under “Direct API calls”

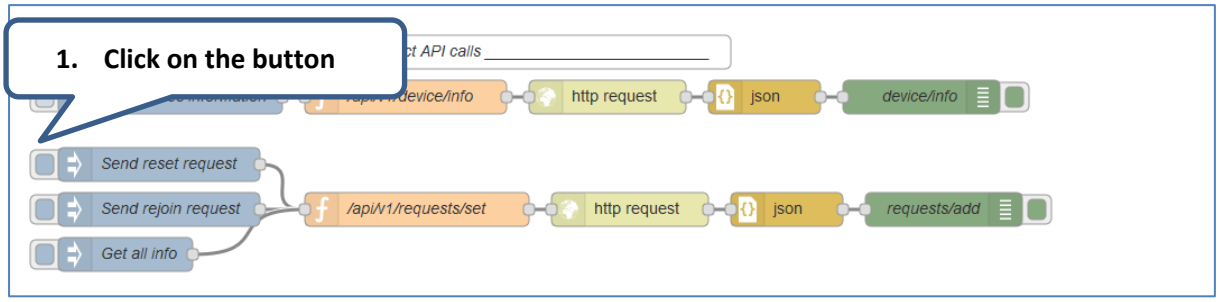

Figure 38. Send Request to LoRa Software Modem

4 LoRa Software Modem Firmware Update

Firmware update for the LoRa Software Modem is done through the Nucleo board USB port.

4.1 Python Program and Script Installation

To complete the firmware update process, it's necessary to install Python (version 3.0 or later) and pyserial module.

Note : "Add Pythong 3.x to PATH" box must be checked during the Python installation.

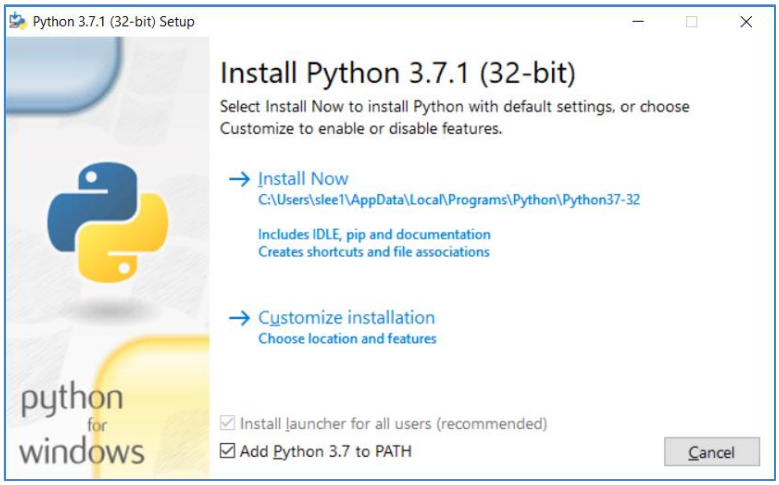

Figure 39. Add Python 3.x to PATH check box

Then install the pyserial module as shown below:

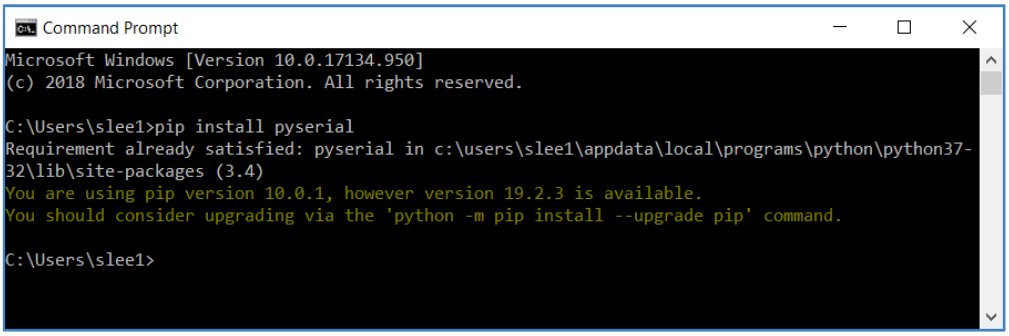

Figure 40. Pyserial module installation

4.2 Firmware Update

To execute the firmware update:

1. Locate the Python script (modem_fw_update.py) for transferring the modem firmware from PC to LoRa Software Modem through the Nucleo application processor.
2. Locate the Modem firmware binary file (i.e. modem.up128).

The Python script and Modem firmware files can be downloaded from https://github.com/lwmodem/firmware

Within the firmware folder, there are four types of Modem firmware files:

1. flash.hex: This file is used by the ST-LINK V2.

2. flash.bin: This file is used by the ST-LINK V2.
3. modem.up128: This modem firmware update file used by Application. Remaining bytes of the last block filled with 0's.
4. modem.up: This modem firmware update file used by Application. Remaining bytes of the last block not filled with 0's.

3. Ensure mini USB cable is connected between PC and Nucleo and Shield boards
4. Open a serial terminal (Tera Term) from the PC. Refer to Section 2.5 for more information.
5. Place the LoRa Software Modem into Evaluation Mode.
    a. Press and hold down the User button (blue) on the Nucleo board.
    b. While holding down the User button, press and release the Reset button (black)
    c. Continue to hold down User button for at least 5 seconds and release the User button
6. Confirm the LoRa Software Modem is in Evaluation mode by typing "getversion" in the serial terminal

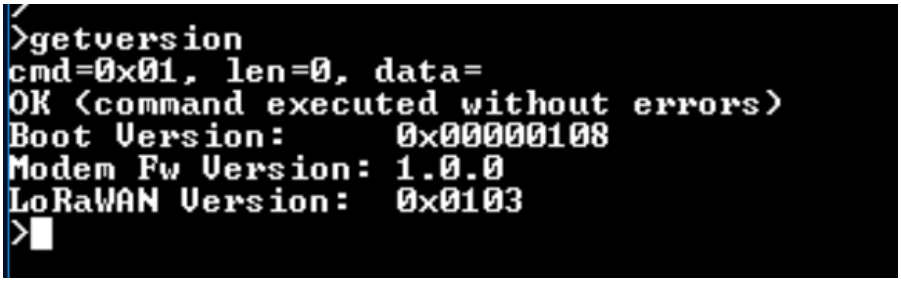

Figure 41. Confirm LoRa Software Modem Firmware Version on Serial Terminal

7. Close the serial terminal if it's open
8. Open a new Windows Command Prompt
    a. Type cd to the directory where the new firmware is located
    b. Type python modem_fw_update.py modem.up128 COM< >
        i. COM<> is the COM port used by the Nucleo board
9. Hit Return. If done successfully, the Windows Command Prompt should look like Figure 39

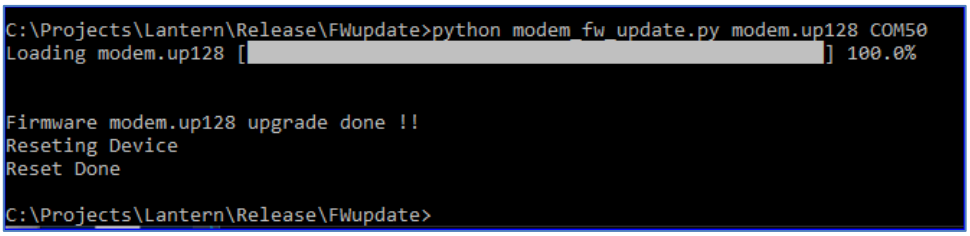

Figure 42. LoRa Software Modem Firmware Update Status

10. LED near the module on the mbed shield will led up in RED, wait until this LED turn OFF, then set the LoRa Software Modem into Evaluation Mode as from step 5.
11. To verify that the LoRa Software Modem firmware have been updated
    a. Open a new serial terminal (Tera Term)
    b. Repeat Step 5 of this section to re-enter Evaluation Mode
    c. Type "getversion" to confirm the version number is correct

5 Terminology

LoRa
The radio protocol itself is called LoRa, which stands for Long Range. It provides a chirp-based system (more on that later under Bandwidth) to allow 2 devices to communicate over great distances.

LoRaWAN
LoRaWAN builds on LoRa to provide a complete end-to-end protocol stack to allow Internet of Things (IoT) devices to communicate with an application server and each other. It provides Standard device classes (A, B, and C) with defined behaviors:

- **Class A** - allow bi-directional communications, but downlink messages can only be transmitted after an uplink transmission by the node. This class involves the lowest power devices
- **Class B** - allow bi-directional communication with scheduled receive slots
- **Class C** - with continuously open receive windows, only closed when transmitting

End Device, Node, Mote
An object with an embedded low-power communication device.

DevEUI: End-Device Identifier
DevEUI is a global end-device ID in IEEE EUI64 address space that uniquely identifies the end-device, assigned to the device by the chip manufacturer.

AppEUI: Application Identifier
The AppEUI is a global application ID, 16 characters long, in IEEE EUI64 address space that identifies the application provider (i.e. owner) of the end-device. The AppEUI is stored into the end-device before the execution of the activation procedure. ResIOT administration panel can automatically generate AppEUI values.

AppKey: Application Key
The AppKey is an AES-128 application key specific for the end-device which is assigned by the application owner to the end-device. It is derived from an application-specific root key, exclusively known by the application provider. Whenever an end-device joins a network via *over-the-air activation*, the AppKey is used to derive the specific session keys (NwkSKey and AppSKeyfor) that end-device has to encrypt and verify into the network communication and application data.

Application
A piece of software, running on a server.

Uplink Message
A message from a Device to an Application.

Downlink Message

A message from an Application to a Device.

Security Tokens:
Security Tokens are used to authenticate programmatically requests to ResIOT services. At the moment they are used for:

- Sigfox access
- API access
- HTTP Receivers access

It's important to understand each security token can be configured in order to work only with the desired services.

Join
LoRaWAN 1.0 knows three distinct 128-bit security keys. The application key AppKey is only known by the device and by the application. When a device joins the network (this is called a join or activation), an application session key AppSKey and a network session key NwkSKey are generated. The NwkSKey is shared with the network, while the AppSKey is kept private. These session keys will be used for the duration of the session.

NwkSKey
The NwkSKey is used to validate the integrity of each message by its Message Integrity Code (MIC). This MIC is similar to a checksum, except that it prevents intentional tampering with a message. For this, LoRaWAN uses AES-CMAC. The AppSKey is used for encryption of the application payload.

Regions
LoRaWAN uses lower radio frequencies with a longer range that tries to be as uniform as possible in all different regions of the world. As a result, LoRaWAN is specified for a number of bands for these regions.

- EU 863-870 MHz
- US 902-928 MHz
- Australia 915-928 MHz
- China 779-787 MHz and 470-510 MHz

6 Revision History

Release	Date	Edit By	Comments
1.0	Oct 22, 2019	N. Lam	Final version 1.0
1.1	Jan 16, 2020	N. Lam	Updated to match the changes in the modem milestone 17
1.2	Jan 22, 2020	N. Lam	Updated document Header, Footer and changed Network Server connect to TTN instead of Actility
1.3	Jan 27, 2020	N. Lam	Removed Watermark & corrected reference error in the footer

Important Notice

Information relating to this product and the application or design described herein is believed to be reliable, however such information is provided as a guide only and Semtech assumes no liability for any errors in this document, or for the application or design described herein. Semtech reserves the right to make changes to the product or this document at any time without notice. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Semtech warrants performance of its products to the specifications applicable at the time of sale, and all sales are made in accordance with Semtech's standard terms and conditions of sale.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS, OR IN NUCLEAR APPLICATIONS IN WHICH THE FAILURE COULD BE REASONABLY EXPECTED TO RESULT IN PERSONAL INJURY, LOSS OF LIFE OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK. Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

Contact Information

Semtech Corporation

Wireless & Sensing Products

200 Flynn Road, Camarillo, CA 93012

E-mail: sales@semtech.com

Phone: (805) 498-2111, Fax: (805) 498-3804

www.semtech.com