Co-funded by the
Erasmus+ Programme
of the European Union

**WirelessUP!**

**UPraising VET skills for innovation in European electrotechnical sector**

Project number: 2017-1-HR01-KA202-035434

# WirelessUP! Toolkit

# Module 2:
# Connecting Devices to IoT via Wireless Mesh Networks

**Intellectual Output 3**

**February 2019**

## Content

# Build your IQRF network

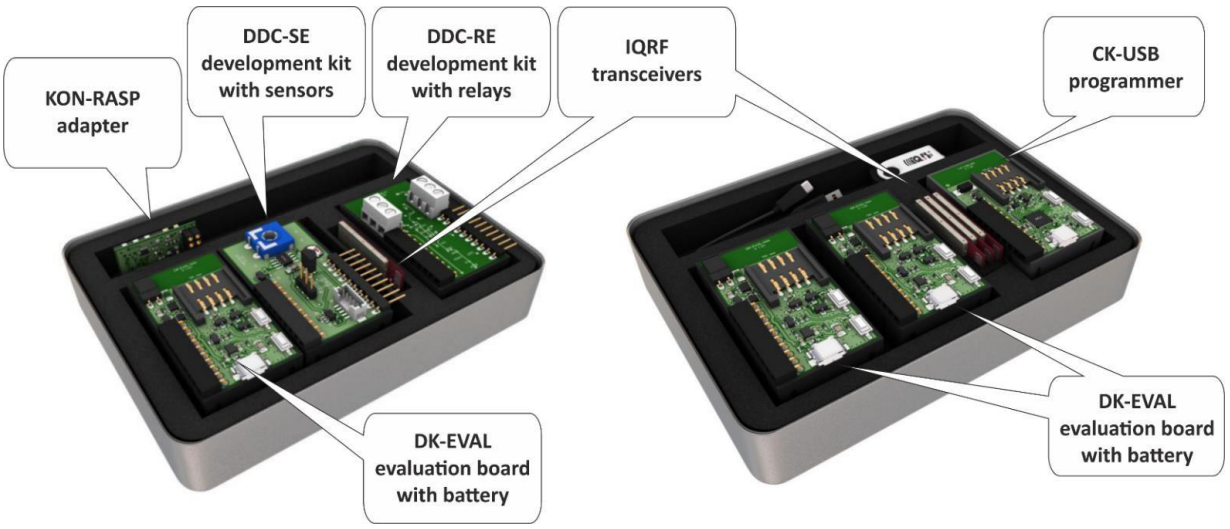
### 1.1 Understand the components in the Development Set for IoT (DS-IOT-01)

1. Set the box with the UP board aside for use in **Step 2: Install Your IQRF Gateway**.
2. Open the two small boxes containing the **IQRF technology** components (DS-IOT-01).


3. Review the key features, switches and connectors for each component.

## 1.1.1 IQRF components

Inside the two IQRF boxes, you will find:



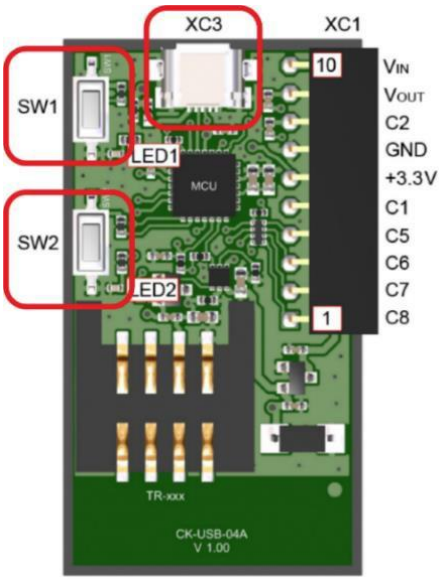

- one gray box called **CK-USB** – the programmer,
- three black boxes called **DK-EVAL** – evaluation kits for powering wireless transceivers,
- one **DDC-SE** development kit with three sensors - the Dallas thermometer, the light sensor and the potentiometer,
- one **DDC-RE** development kit with two relays,

- four **IQRF transceivers** for creating a basic IQRF network,
- one **micro USB cable** for CK-USB connection to a computer,
- one **KON-RASP** adapter for the connection of an IQRF transceiver to the UP board.

#### 1.1.1.1 CK-USB

This development kit is for programming and debugging of IQRF transceivers. You will connect this tool to a computer USB port with the micro USB cable connected to an XC3 connector.

**SW1** and **SW2** are pushbuttons for USB mode selection and other purposes. Find details about it here: https://www.iqrf.org/products/development-tools/development-kits/ck-usb-04a

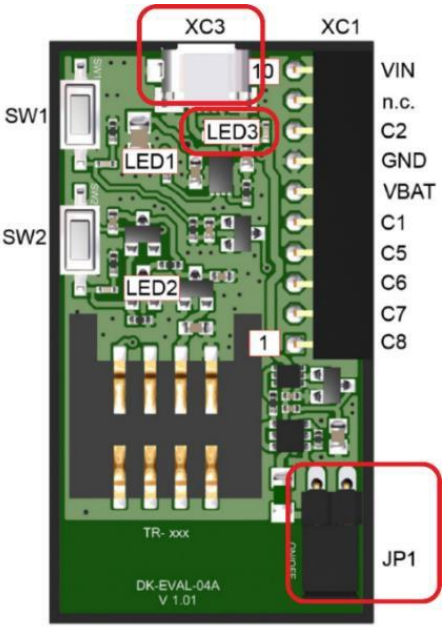



**Caution**: The IQRF transceiver can be plugged into / unplugged from the SIM connector while powered off only. The SIM connector is not powered while the SW2 pushbutton is held. Press and hold it always when you are plugging an IQRF transceiver to or you are unplugging it from the SIM connector of CK-USB.

#### 1.1.1.2 DK-EVAL

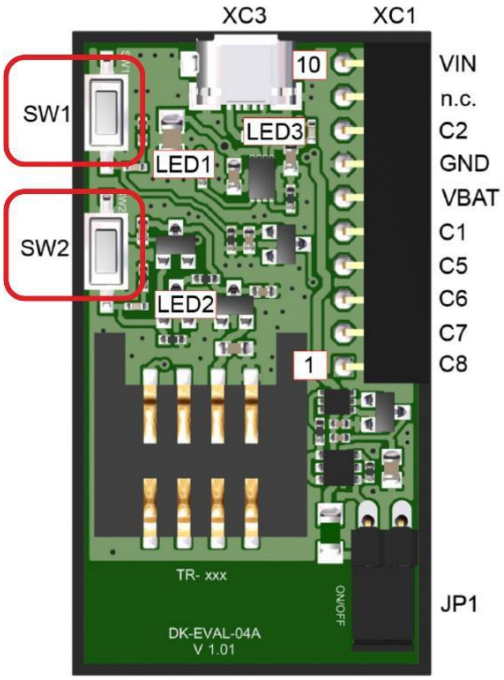
This development kit is supplied from the internal accumulator (battery) or from an external power source via micro USB connector **XC3** which also serves as a charger. Charging is indicated by red **LED3**. The accumulator (battery) should be kept charged. Charging takes up to 8 hours when the battery was empty. You can use a power supply expansion slot (https://www.iqrf.org/products/accessories/power-supplies/dk-pwr-01) so you can charge all batteries at once.

The TR transceiver is supplied when jumper **JP1** is turned on.



**SW1** is a **User** pushbutton which is connected to pin C5 of the transceiver. It will be used here for bonding (adding to a network) in next steps.

**SW2** is a **Reset** pushbutton. A transceiver is disconnected from the power supply when the SW2 pushbutton is pressed.

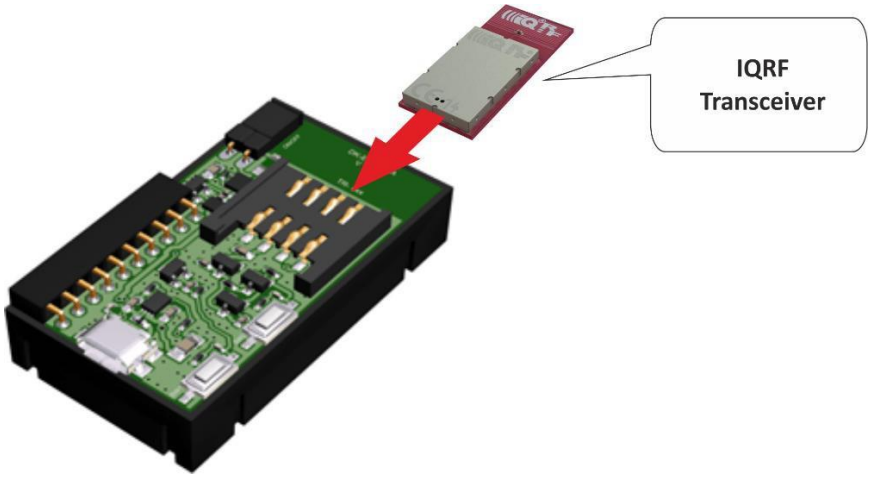



**Caution:** SW2 must be pressed always when you are plugging or unplugging the transceiver to/from the SIM connector.

Find details about this tool here: https://www.iqrf.org/products/development-tools/development-kits/dk-eval-04a.

#### 1.1.1.3 IQRF Transceiver

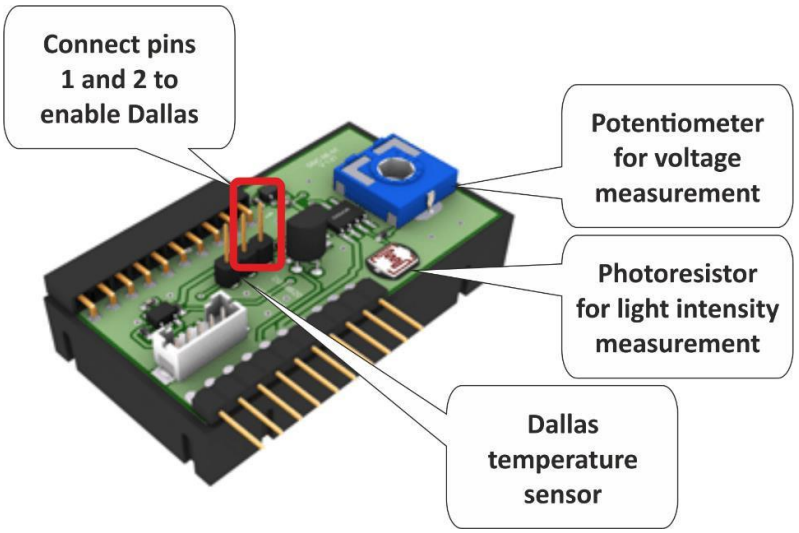
**TR-72DAT** (version of IQRF transceivers with temperature sensor and an on-board antenna) is a family of IQRF transceiver modules operating in the 868 MHz and 916 MHz license-free ISM frequency band. It is a highly integrated ready-to-use design containing MCU, RF circuitry, integrated LDO regulator, serial EEPROM, temperature sensor and an on-board antenna.




Press and hold the SW2 button (Reset button) on a CK-USB or DK-EVAL always when you are plugging an IQRF transceiver to or you are unplugging it from the SIM connector. Be sure you connect an IQRF transceiver into a CK-USB or DK-EVAL in the right direction (the antenna is outside the SIM connector).

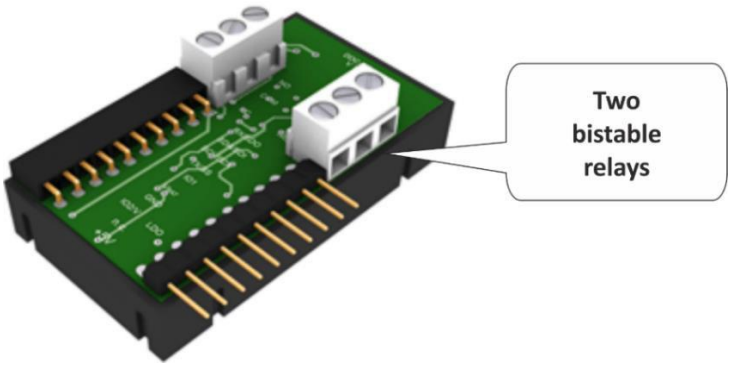
#### 1.1.1.4 DDC-SE

A sensor development kit contains a potentiometer for voltage measurement, a photoresistor for light intensity measurement and a Dallas 18B20 temperature sensor. Connect pins 1 and 2 with a jumper to select reading values fromthe Dallas 18B20 sensor. Connect DDC-SE with DK-EVAL to be charged and to read values from it through the IQRF transceiver. It is compatible with other DDC (Development Daisy Chain) kits.




#### 1.1.1.5 DDC-RE

A relay development kit contains two bistable (latching) relays. Connect this tool to DK-EVAL to be charged and with the IQRF transceiver to control the relays. It is compatible with other DDC (Development Daisy Chain) kits.



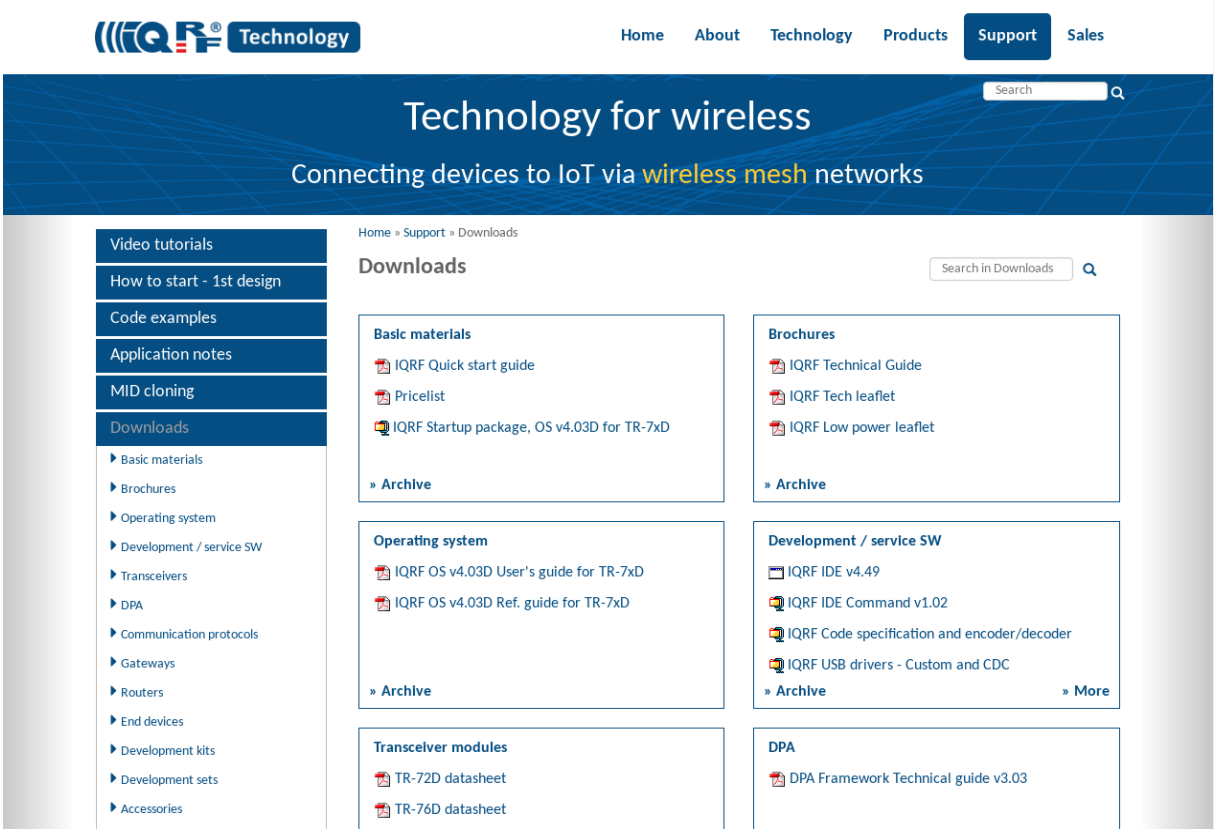
#### 1.1.1.6 KON-RASP

This is an adapter for a connection of an IQRF wireless transceiver to the UP board. It will be described in more detail in Part 2 – IQRF Gateway.

## 1.2 Create the IQRF Network

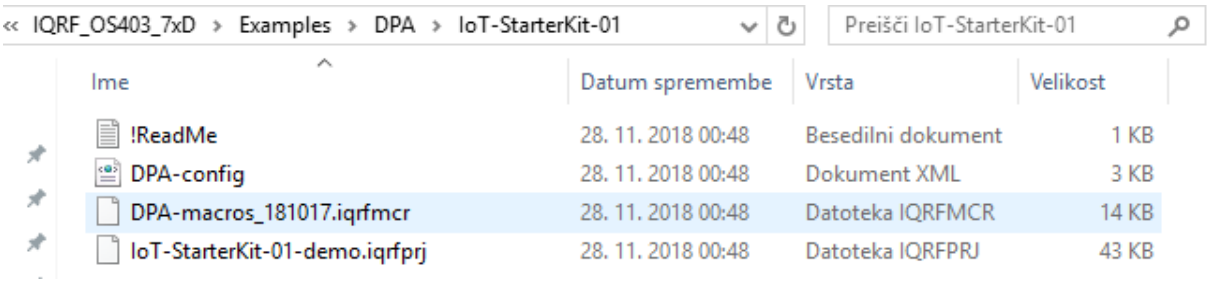
### 1.2.1 IQRF IDE installation

If you haven't done it yet, download the startup-package from [www.iqrf.org/support/download](www.iqrf.org/support/download) and install the last version of the IQRF IDE. There are two downloads – the IQRF Startup package and the IQRF IDE, and the IQRF IDE needs to be installed.




In the startup package, in the **Examples/DPA/IoT-Starter-Kit-01** folder, you will find the **IoT-StarterKit-01-demo** file.

Double-click the file to launch the IQRF IDE with all necessary files.



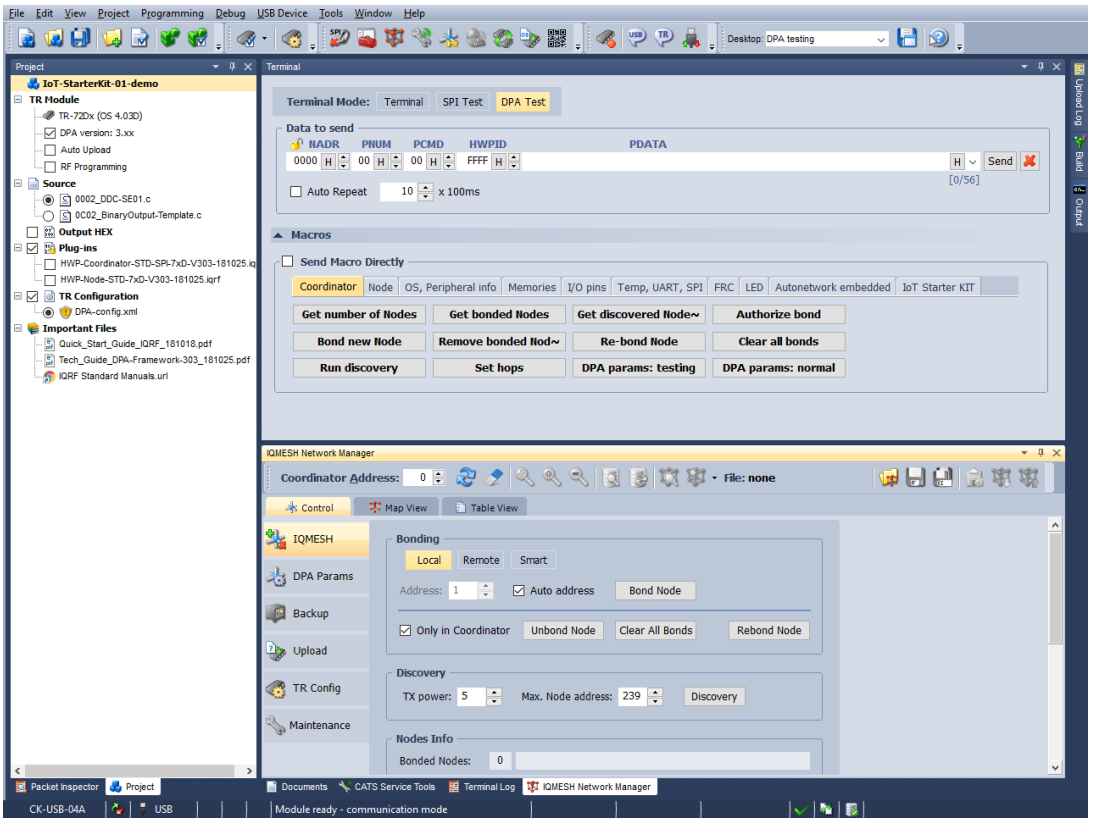

**Note**: The IQRF IDE environment is tested for Windows 10, Windows 8.x, Windows 7 and Vista. Windows installation in a virtual machine is not tested and it is not recommended. The following minimum configuration is required to run IQRF IDE:

- Processor PC-compatible running on 1 GHz or higher
- 512 MB memory
- 30 MB of hard disk space
- 1 USB port
- Vista, Windows 7 (32 bit, 64 bit), Windows 8.x (32 bit, 64 bit), Windows 10 (32 bit, 64 bit)
- Internet Explorer 7.0 or higher or other suitable browsers for Help

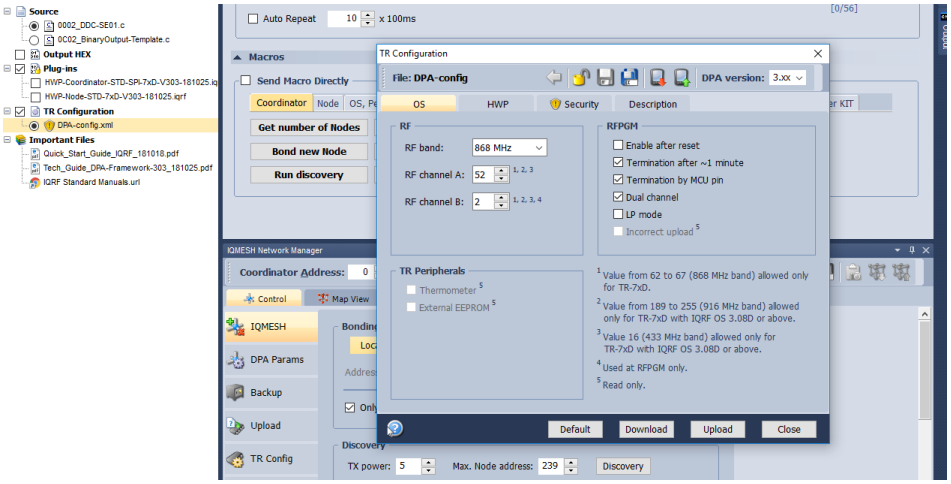
## 1.2.2 IQRF Nodes

### 1.2.2.1 Node #1 – connected to sensors

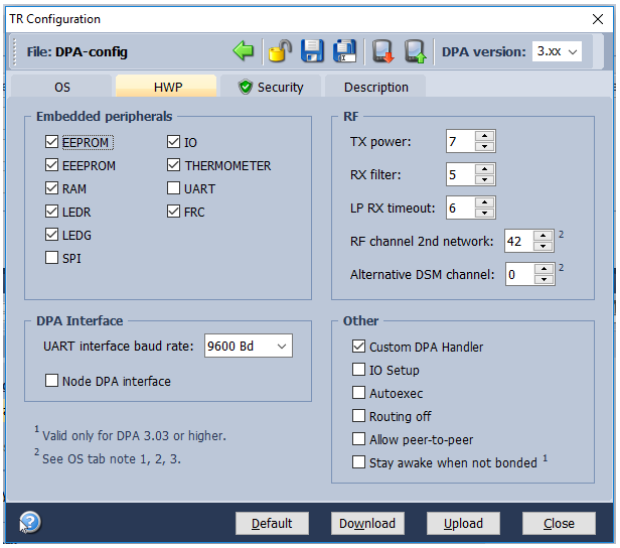
Press the SW2 button (Reset button) on a CK-USB always when you are plugging an IQRF transceiver to or you are unplugging it from the SIM connector. Connect the CK-USB programmer to your computer (on the picture below it is marked with a red box) and insert the first transceiver.



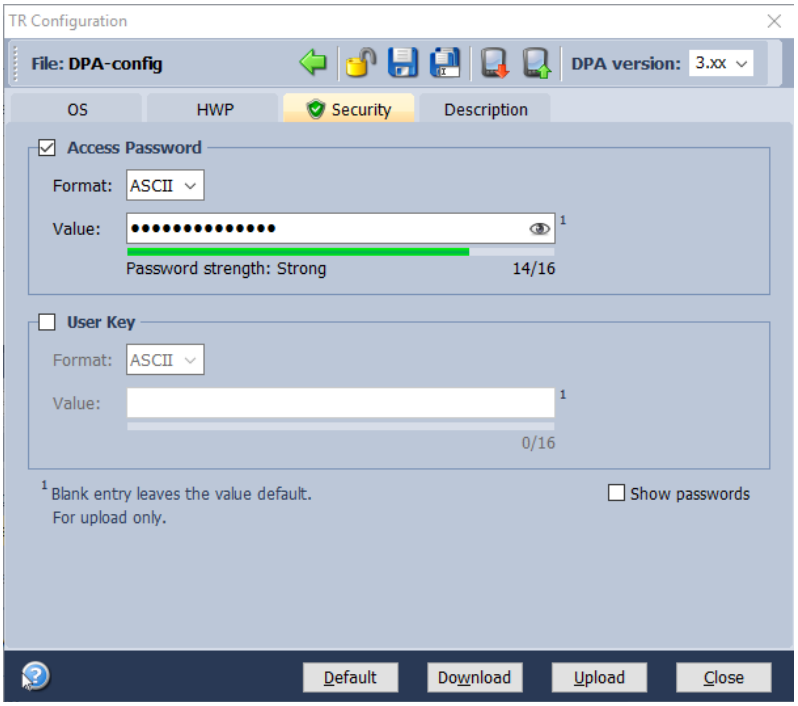

On the left-hand side, you double-click on the configuration. Don't change the selected channel 52 because all devices you will connect to the network later will have the default channel 52 as well.




On the **DPA** tab, allow the usage of a **Custom DPA Handler** because you will upload one into the transceiver in one of the next steps.

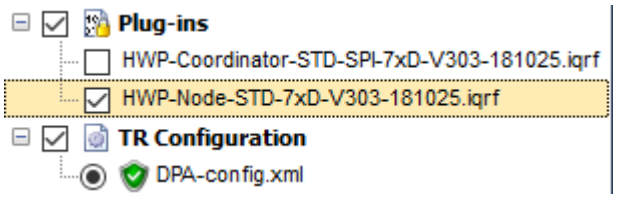



On the **Security** tab, you can set your access password. Don't forget that the same access password must be configured in all devices in your network including the coordinator. The user key is for optional payload data encryption, but this is something we will not use now.

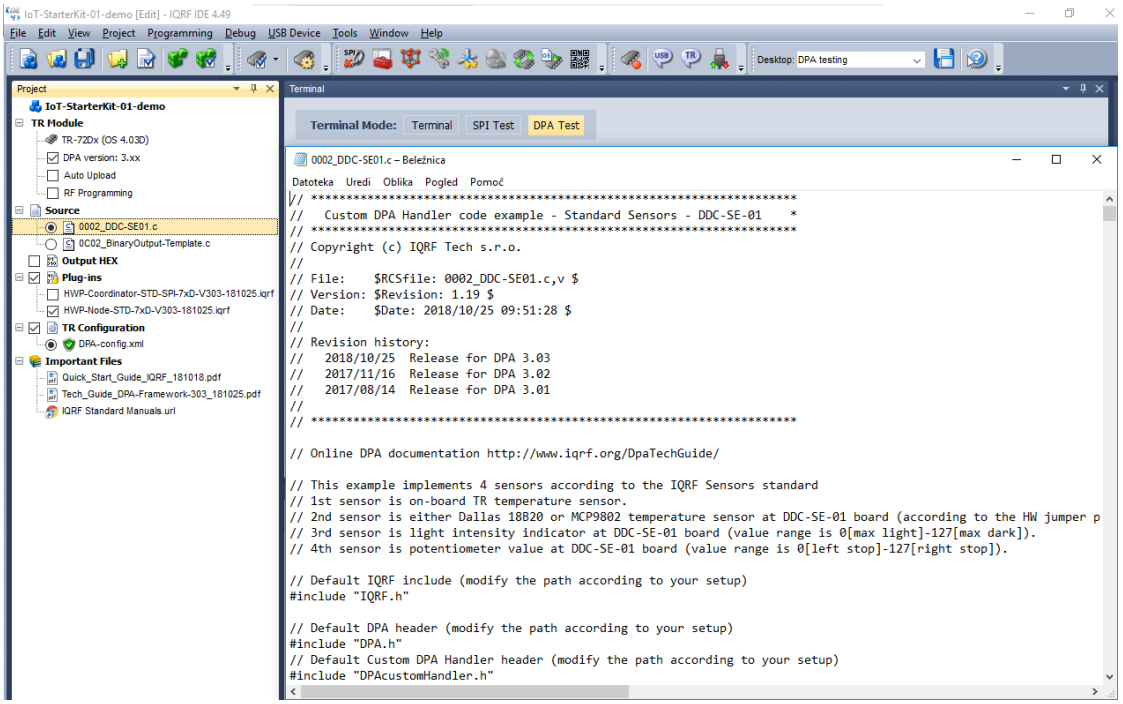



Save the configuration by clicking on the Save button and close it.

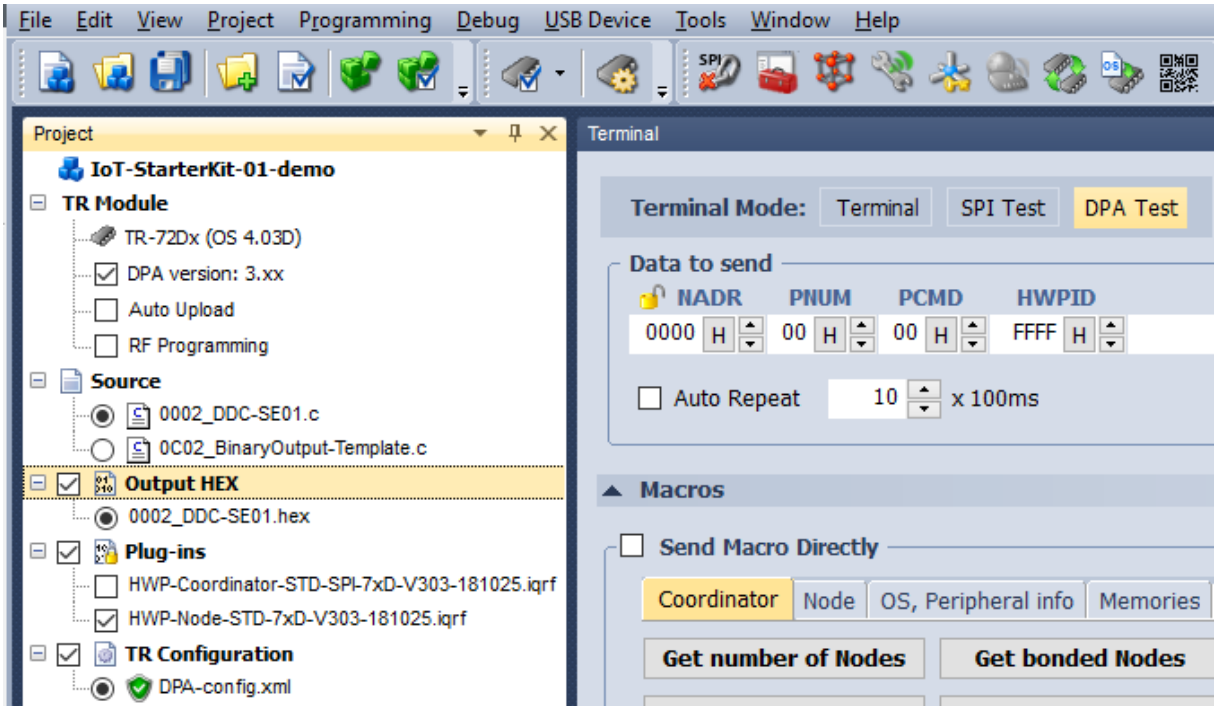
Next, select the Node plugin (HWP-Node-…) so the transceiver supports the DPA protocol and features.



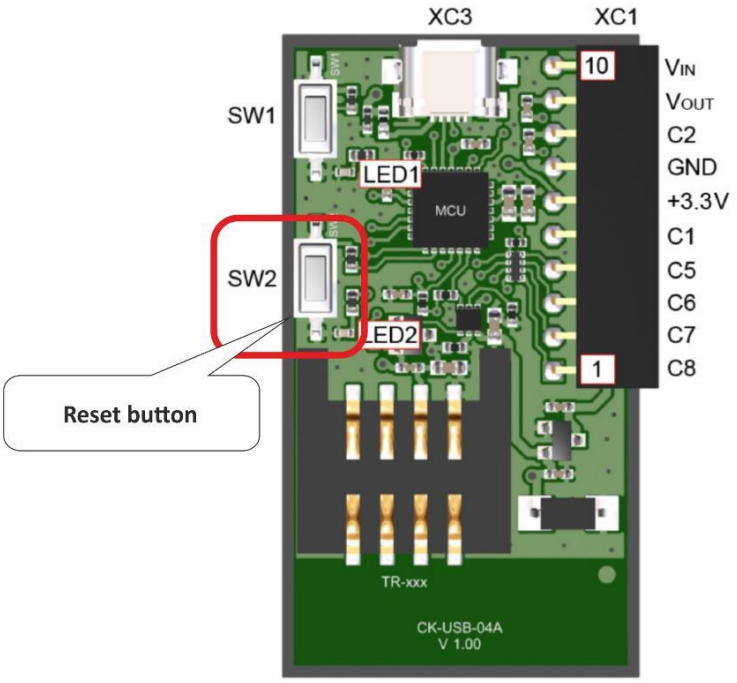
**Custom DPA Handler** is used to customize the behavior of a transceiver. In the **DDC-SE01.c** Custom DPA Handler "c file" you will find the source code that is written according to the IQRF Interoperability standard. To compile the source code, click on the **Build button** or push **F10**.




Make sure you have all three files selected - the **HEX** file of the DDC-SE01 Custom DPA handler, the **Node hardware profile** and the **configuration**. Upload the selected files using the **Upload button** or by pushing **F5**.

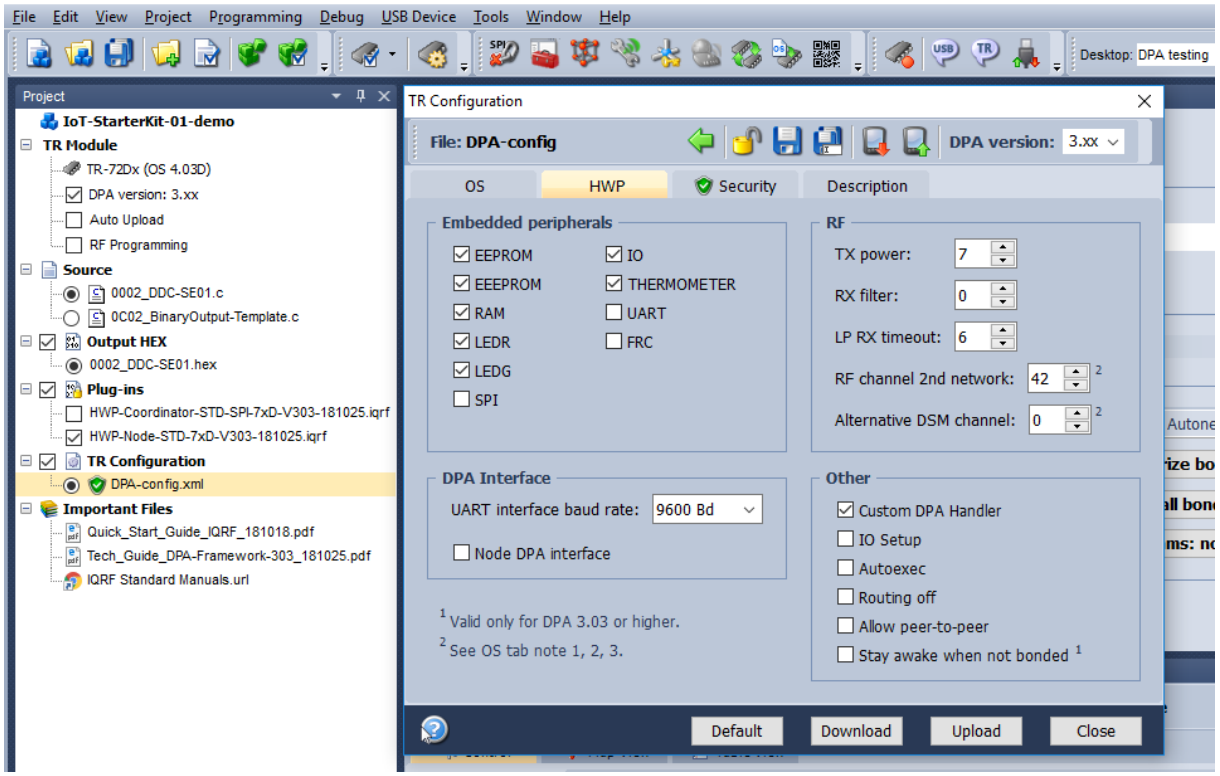


For a safe replacement of a transceiver, press and hold the Reset button (**SW2**) on your programmer. Now remove the connected transceiver and place it next to the Sensor kit.



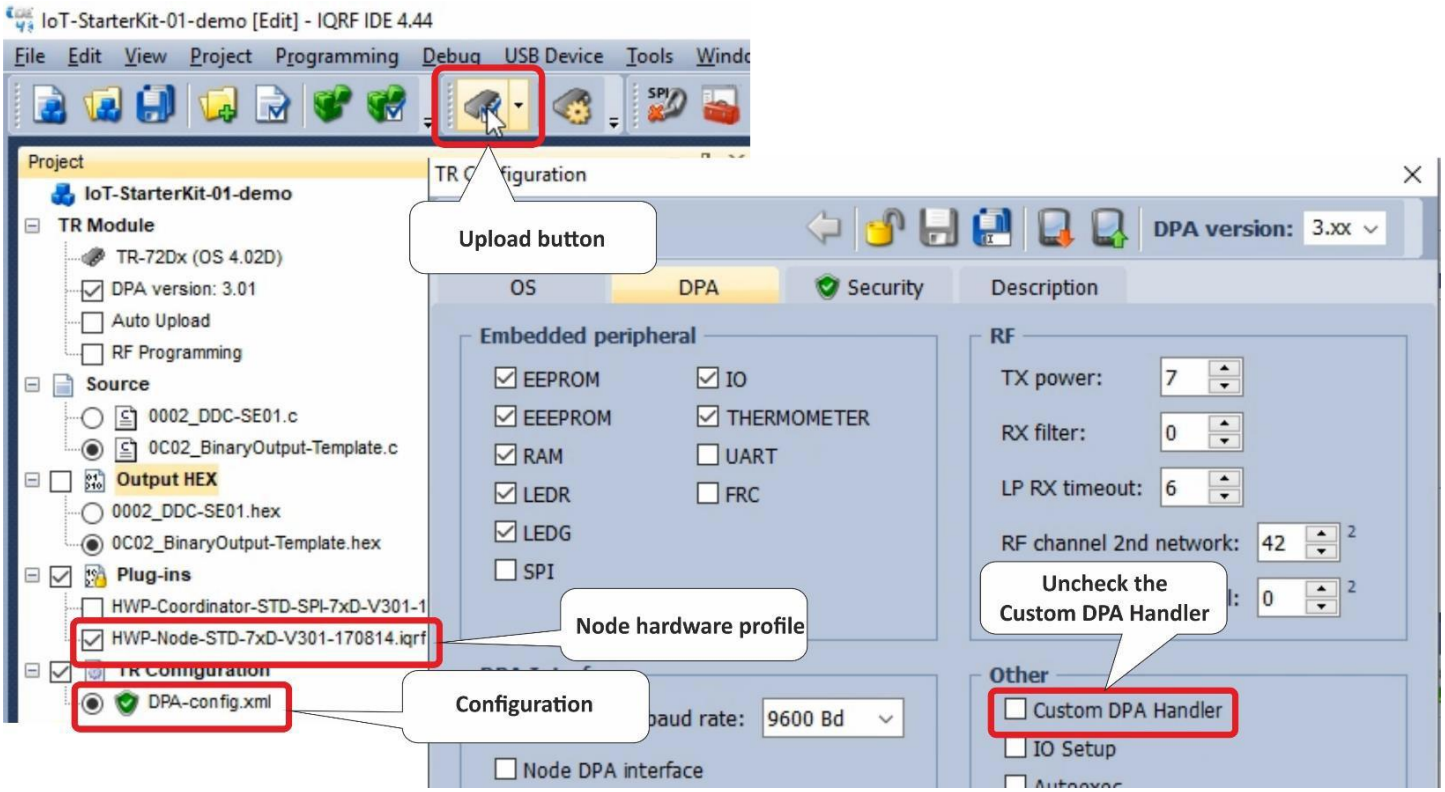

### 1.2.2.2 Node #2 – connected to relays

Insert the second transceiver. Keep the same **configuration** and the same **hardware profile**. Compile the **BinaryOutput Custom DPA Handler** designed to control the relay kit and upload these three files to the second transceiver.



#### 1.2.2.3 Node #3 - repeater

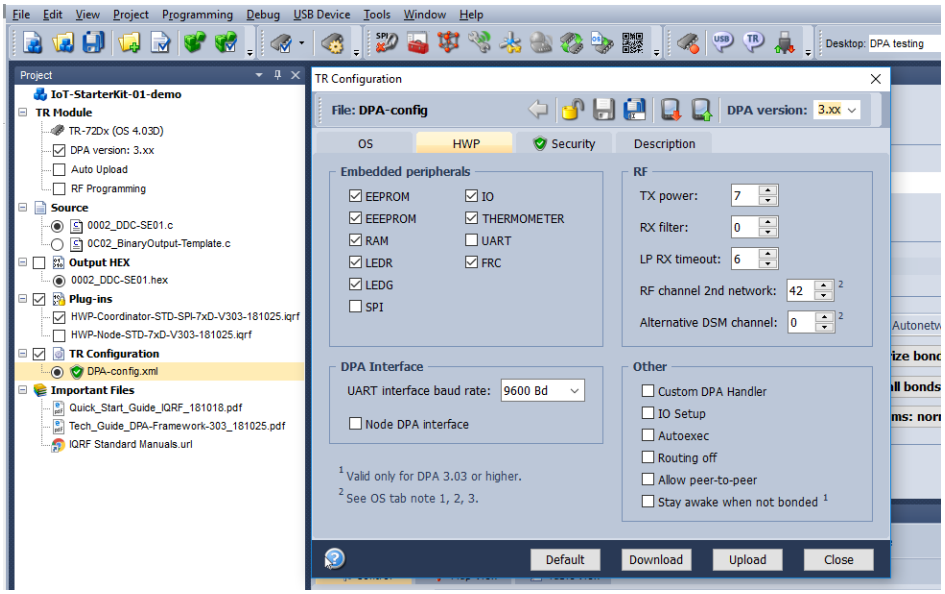
For a safe replacement of a transceiver, press and hold the Reset button (SW2) on your programmer. Replace the second transceiver with the third one This transceiver will work only as a repeater, so it won't contain any Custom DPA Handler. Uncheck the Custom DPA Handler checkbox in the Configuration window. Don't change any other parameters here. Upload the **configuration** and the **Node hardware profile** to the transceiver. Do not upload a HEX file here.



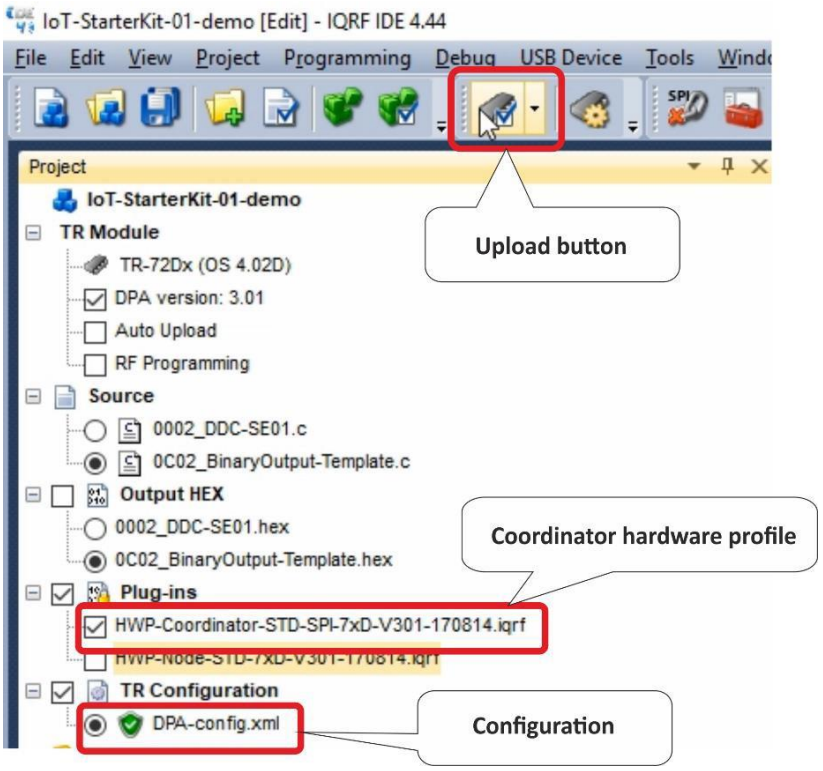

### 1.2.3 IQRF Coordinator

For a safe replacement of a transceiver, press and hold the Reset button (SW2) on your programmer. Now remove the third transceiver and insert the last one which will work as a Coordinator.

In the configuration, enable the **FRC** - Fast Response Command used for fast data collection. This is a peripheral of the coordinator, so it doesn't make sense to enable it in the Nodes. We will not upload any Custom DPA Handler to the Coordinator, so there is no need to enable it. Save the configuration and close it.




Select the **coordinator hardware profile** and the **configuration** and upload them.

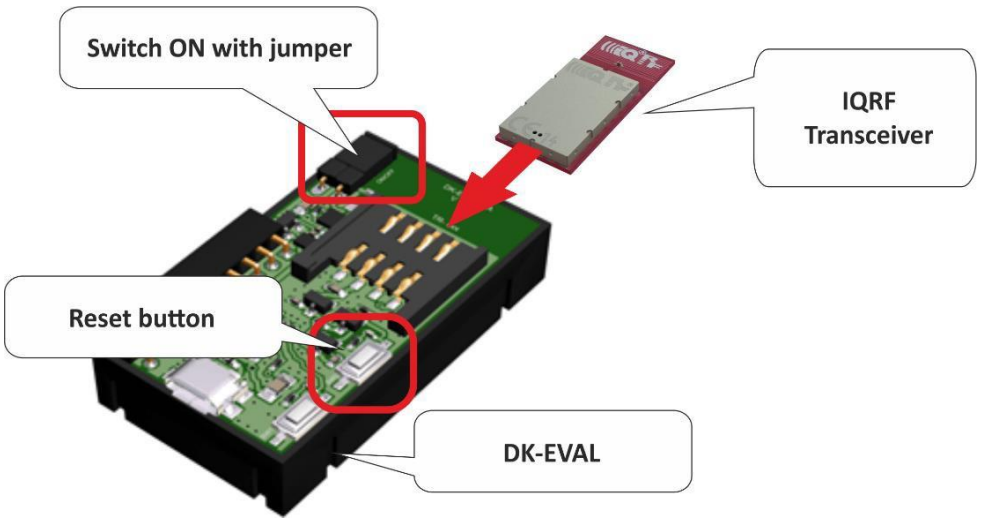



Now you have your coordinator ready so leave it connected to your computer through the programmer.

### 1.2.4 Bonding and unbonding

Adding a node to a network is called **bonding**. Removing a node from a network is called **unbonding**.

For a safe replacement of a transceiver, press and hold the Reset button (SW2) on the evaluation board. Put the prepared transceivers into the evaluation boards and switch them on with the jumpers.

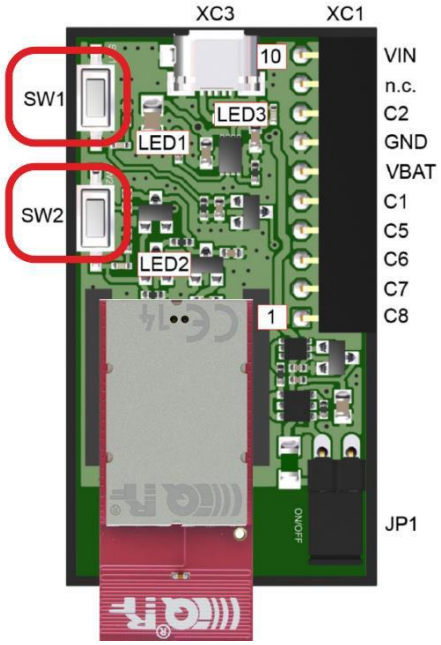



If the red LED on the IQRF transceiver is flashing, it means it has no previous bonding information stored. If this is not the case, you must unbond the node. We will do it here manually.

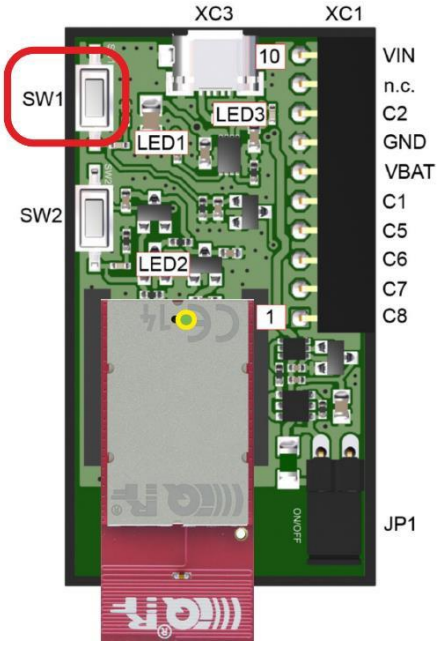
#### 1.2.4.1 Unbonding

You can unbond the node by following this procedure: press both Reset (SW2) and user (SW1) buttons on the evaluation board, release the Reset button. The green LED now lights for 1 second. Once it goes out you have half a second to release the user button. If the Red LED starts blinking, your node was successfully unbonded.

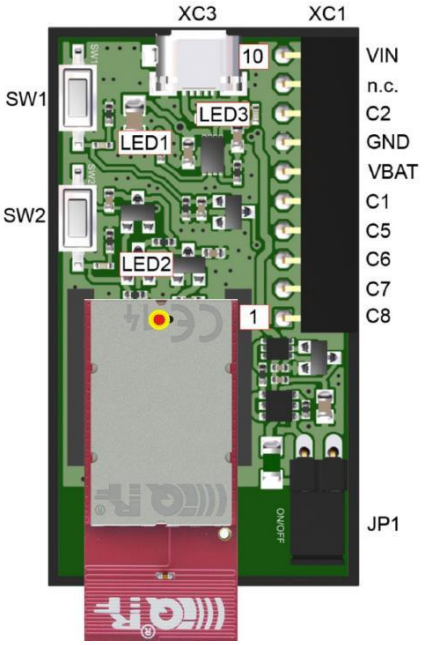
**Step 1**
**- hold both buttons**



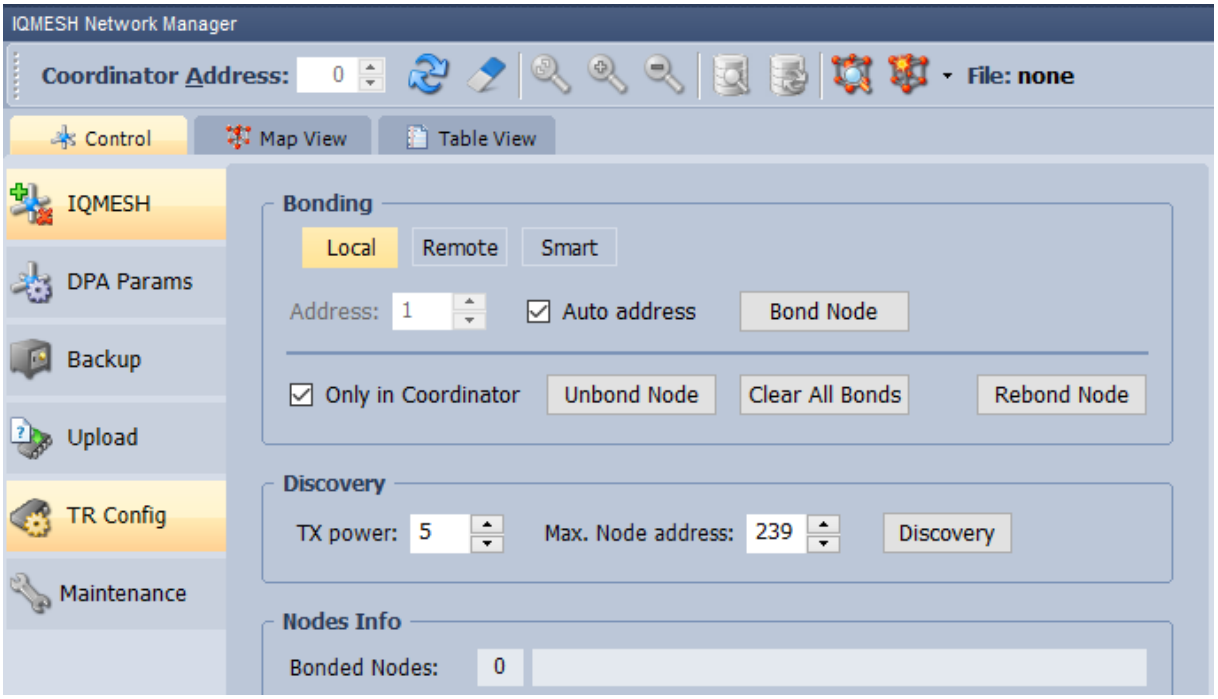

**Step 2**
**- release the SW2 button**
**- green LED lights**




**Step 3**
**- release the SW1 button as soon as green LED turns off (within 0.5 s)**
**- red LED is blinking for 10 s**

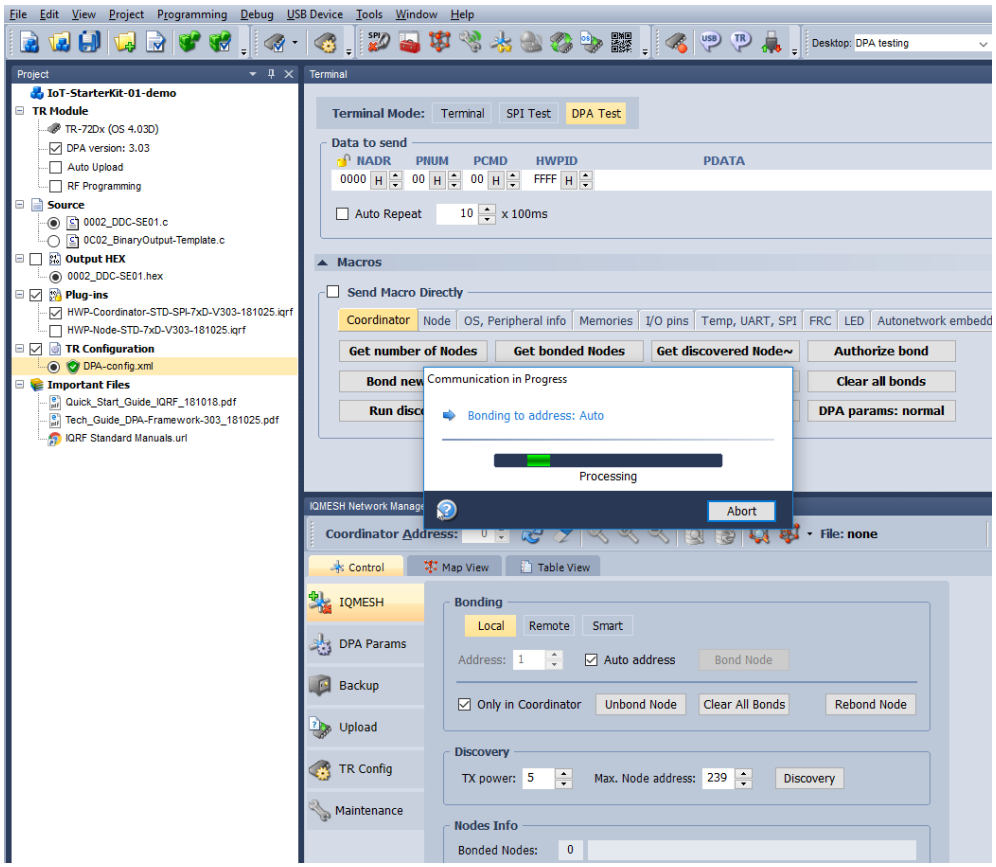



Once you have all three nodes ready, delete any residual bonding information from the **coordinator**. Click the **Clear All Bonds** button on the **IQMESH Network Manager** – **Control** tab.

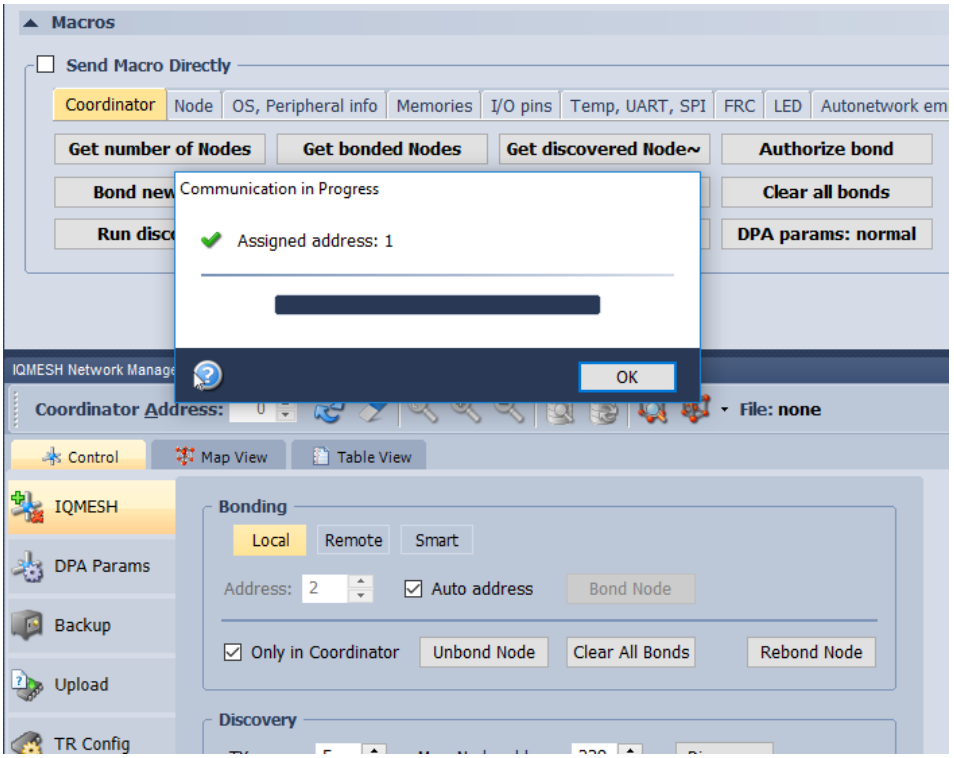


### 1.2.4.2 Bonding

Now build your wireless network. Click on the **Bond node** button to run the Coordinator listening for a new Node request.

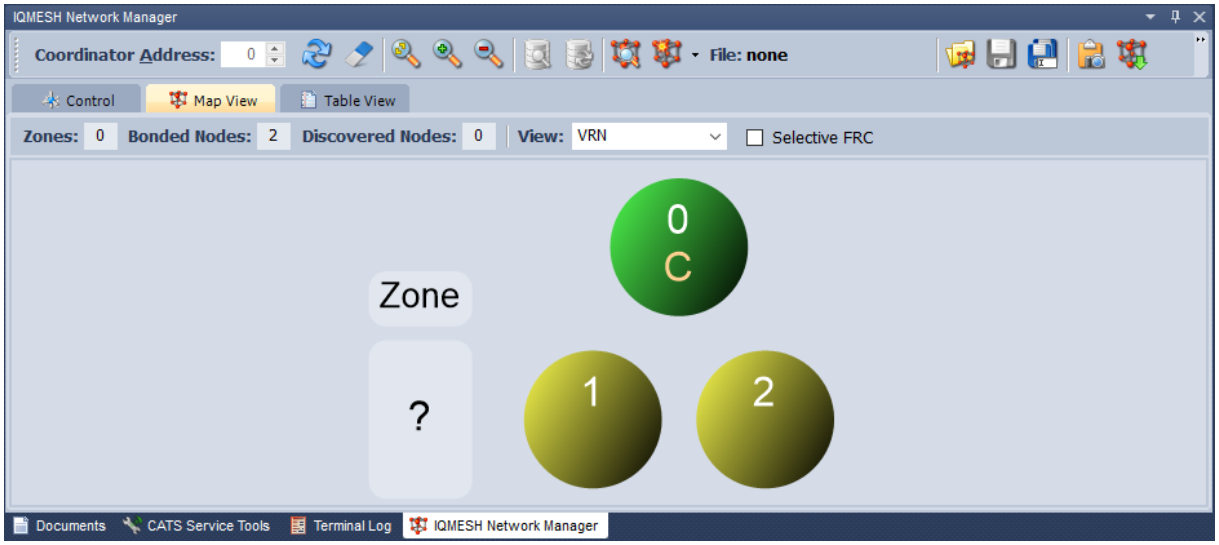



During this ten-second period, press the User (SW1) button on the evaluation board with the transceiver configured for communication with the sensor kit.



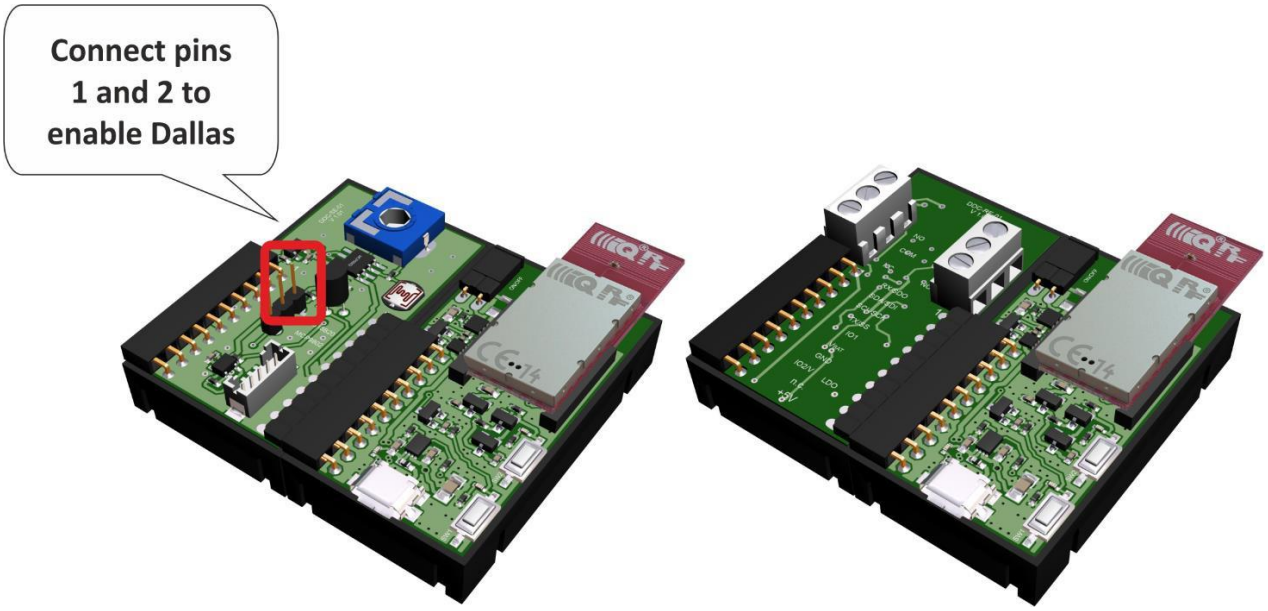

Bond the transceiver with the Custom DPA Handler for communication with the relays as node number 2 following the same procedure. The last transceiver will be bonded as number 3.

Click on the Refresh button (rounded arrows) at the top of the IQMESH Network Manager window. Then you can see the current network in the **Map View**.



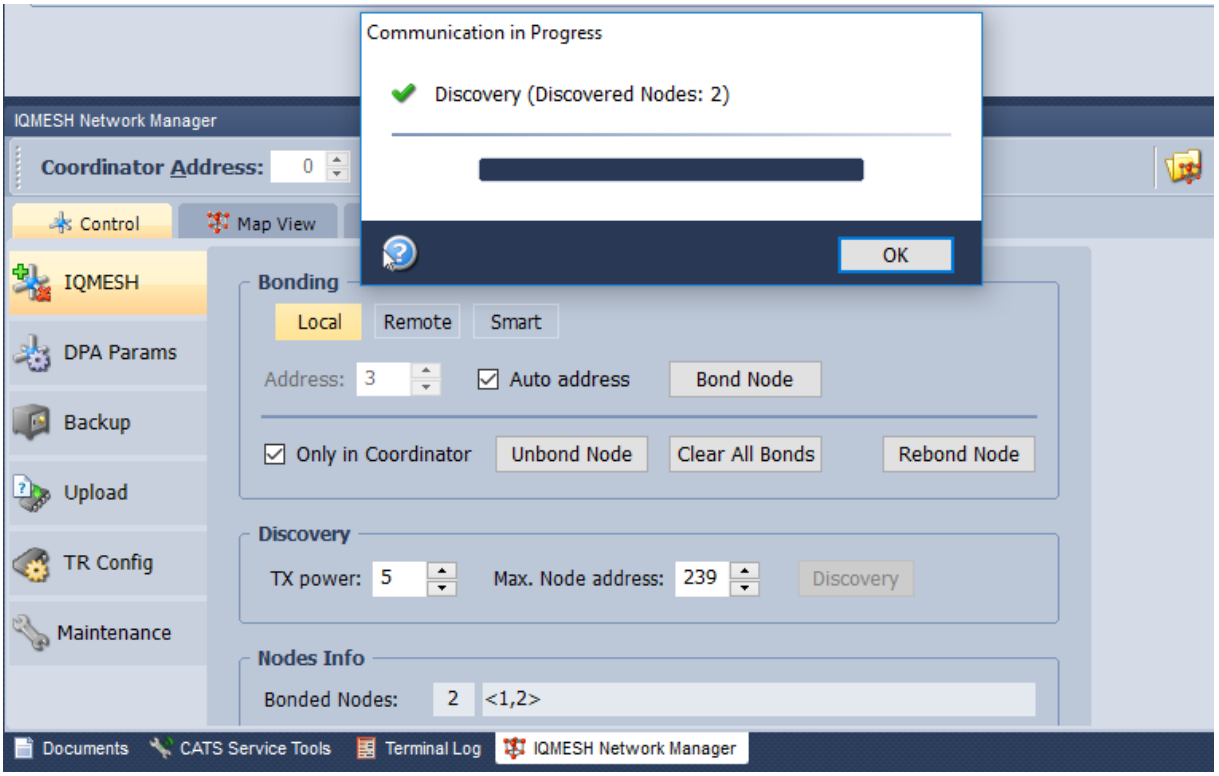

### 1.2.5 DDC kits adding

Connect node number 1 to the sensor kit and node number 2 to the relay kit. Connect pins 1 and 2 on the sensor kit with the jumper to select the Dallas temperature sensor.

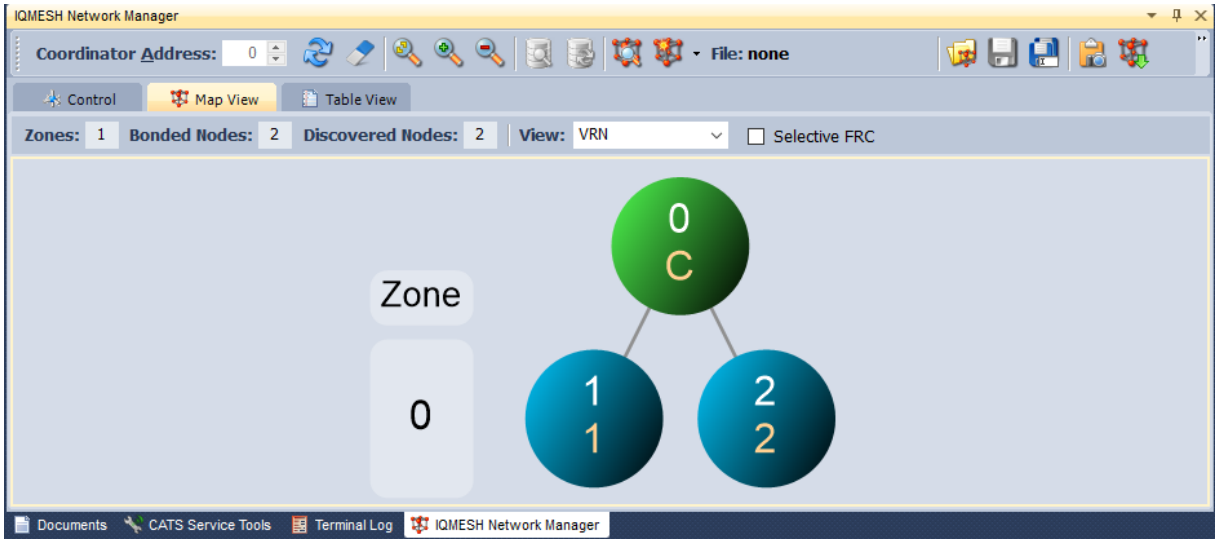



### 1.2.6 Discovery

Now place the nodes on their final destination and run discovery. Discovery will automatically set up routing topology of the network.



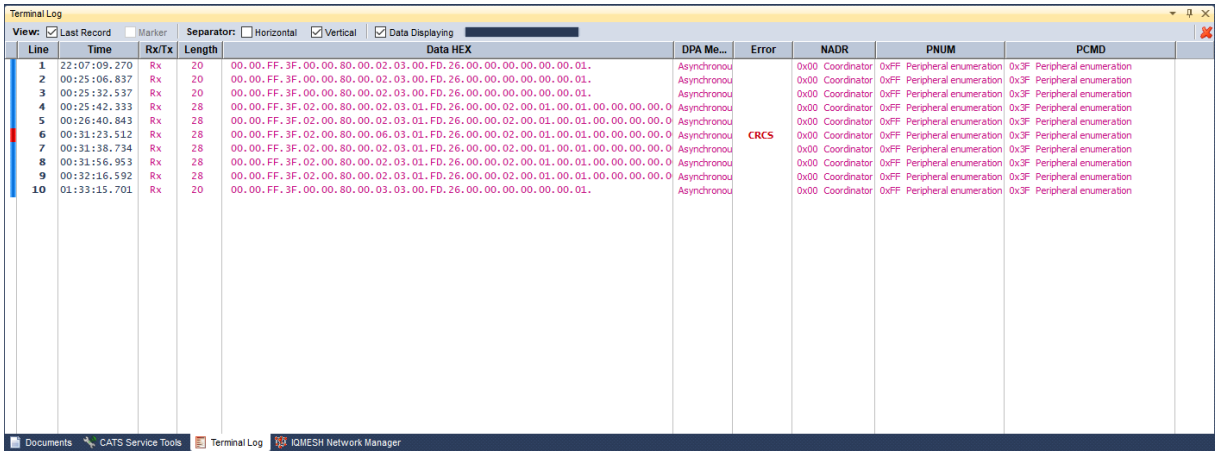

Check the **Map View** again. Discovered nodes have virtual routing addresses and are marked with blue color.




## 1.2.7 Test the wireless communication

### 1.2.7.1 Terminal log

Go to the Terminal log located at the bottom of the IQRF IDE next to the IQMESH Network manager and clear the current log.

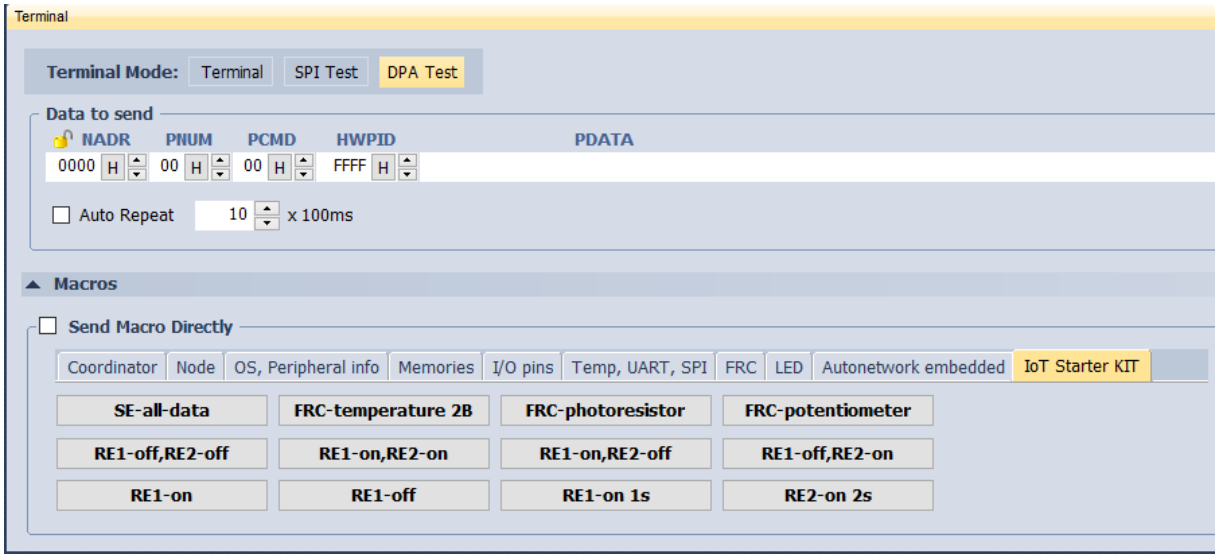



### 1.2.7.2 Macros

On the **IoT Starter Kit macro** tab, there are prepared macros containing commands for Custom DPA Handlers used in the Starter Kit.

Usually, you have to resize the upper panel manually or none of the command buttons will be visible. After a command button is pressed, you must hit the **Send** button or you can select the checkbox in the macro window to **Send Macro Directly**.

Select the first macro that contains the command to read all sensory data from the sensor kit connected to node number 1.

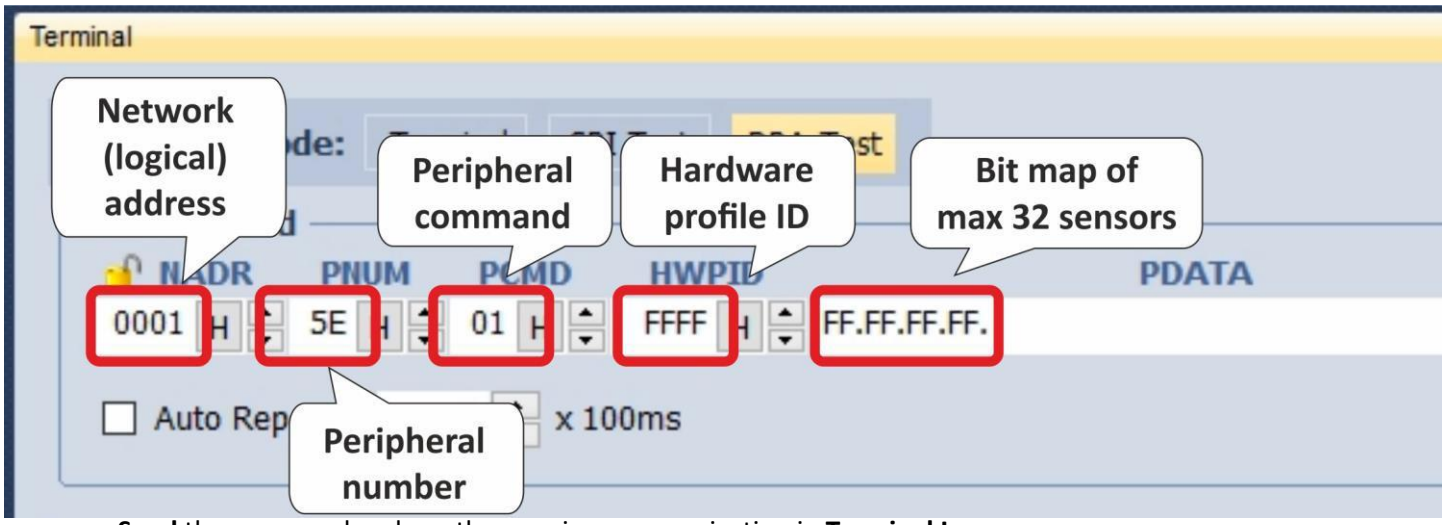



#### 1.2.7.3 DPA packet parts

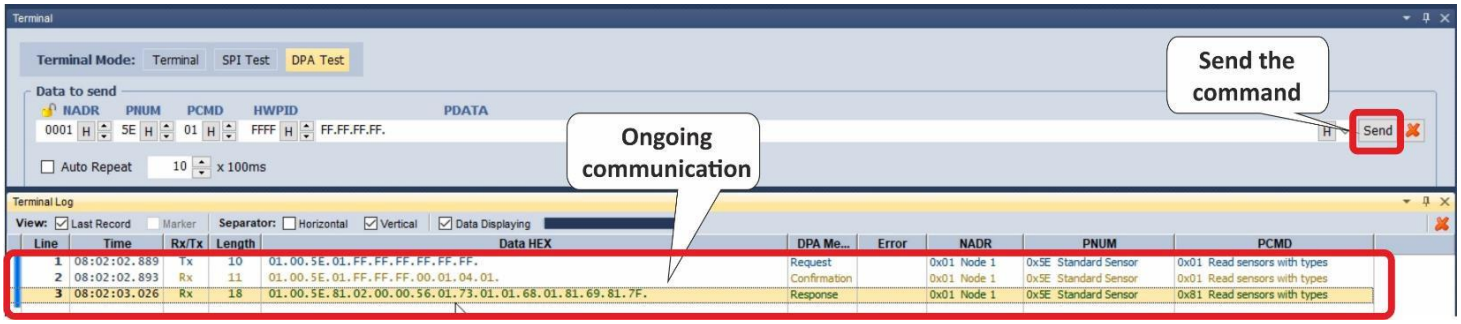
Note the individual parts of the DPA packet. The first part is the **network (logical) address** of the node you want to communicate with. Here we use the logical address **#1** which corresponds to the node with the sensor kit.

**5E** is the hexadecimal representation of the **Standard sensor peripheral**. The number **01** in the **peripheral command** field corresponds to the command for collecting types of sensors and their values. The **FF.FF** in the **hardware profile ID** field indicates that there is no hardware profile filtering.

In the **PDATA** field, there is a bitmap of **maximum 32 sensors** you want to communicate with on the specific node. Here we want to read all sensors available.



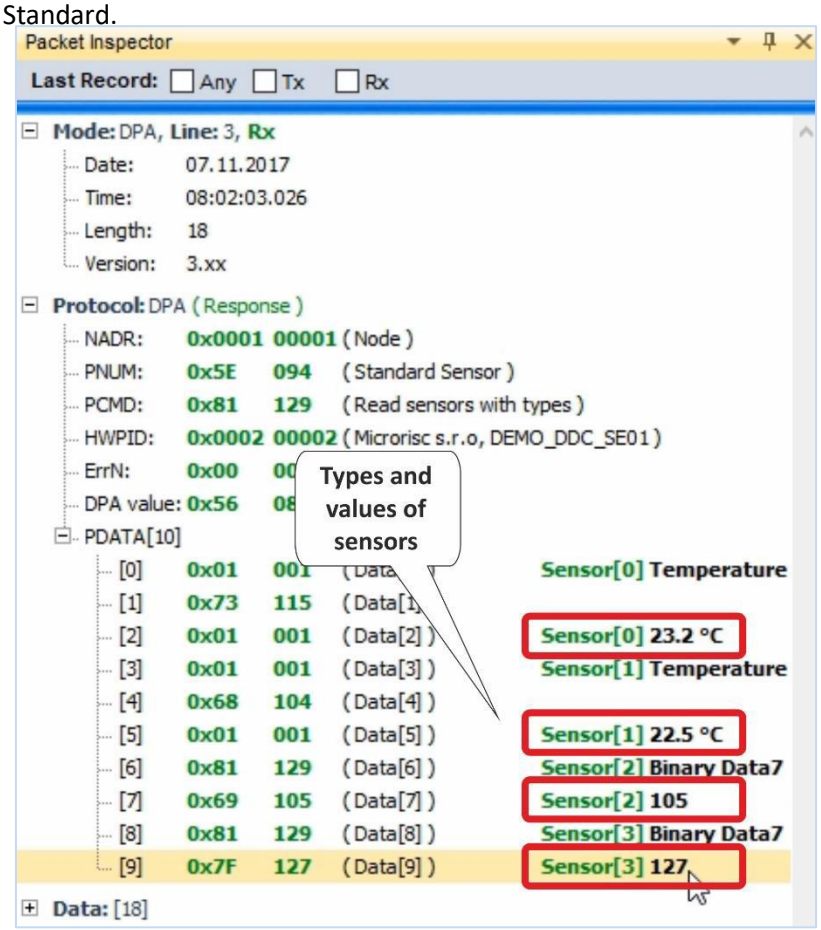

**Send** the command and see the ongoing communication in **Terminal Log**.



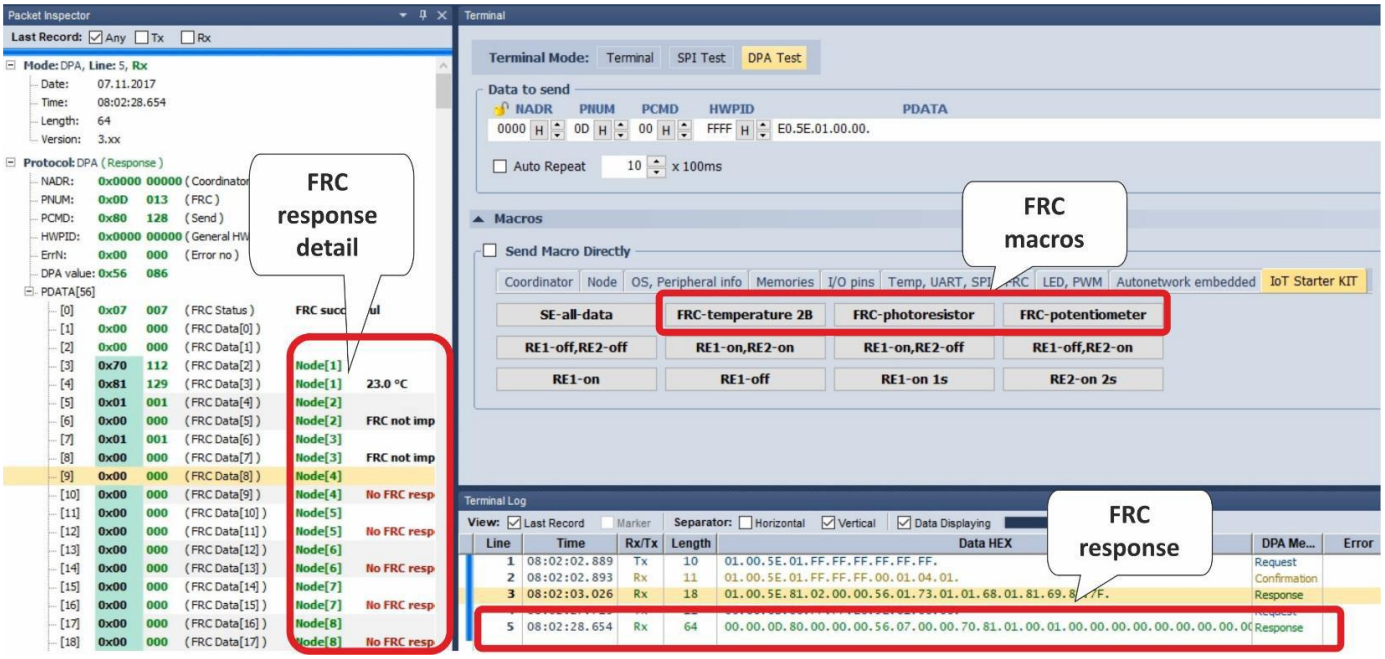

**Note**: DPA protocol is in detail described here: https://www.iqrf.org/support/download&kat=54&ids=511. The IQRF Standard manual is available here: www.iqrfalliance.org/techDocs/.

#### 1.2.7.4 Inspect the packet

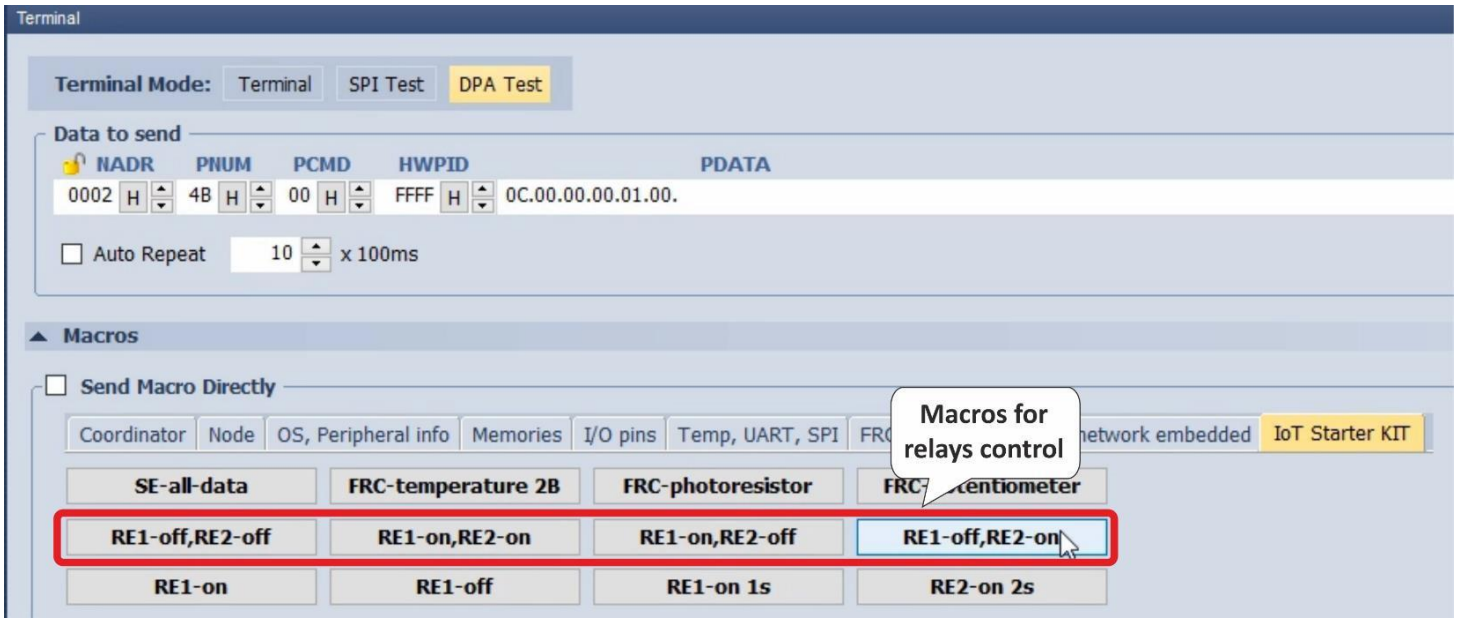
Double-click on the response to see the details of the sensor data. In the Packet Inspector, you can see the values measured by the individual sensors and types of the sensors according to the IQRF Interoperability Standard.




If you have more sensors of the same type in the network, you can collect the measured values at once using the **FRC** (Fast Response Command). In the macros, three **FRC** commands are prepared to collect **temperatures**, **light intensity** from **photoresistors** and **potentiometer** values.

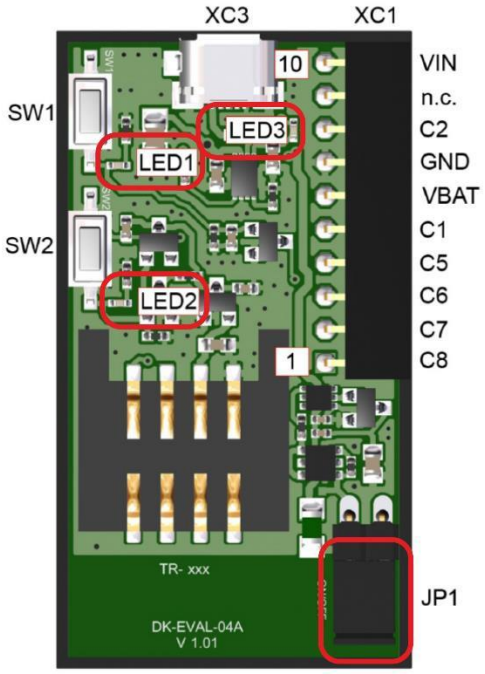



The other macros prepared here are used to **control the two relays** on the relay kit. You can test individual commands and inspect the ongoing records in the terminal log. If everything works well, you should be able to hear the clicks of your relays.



## 1.3 Status of the evaluation board (DK-EVAL)

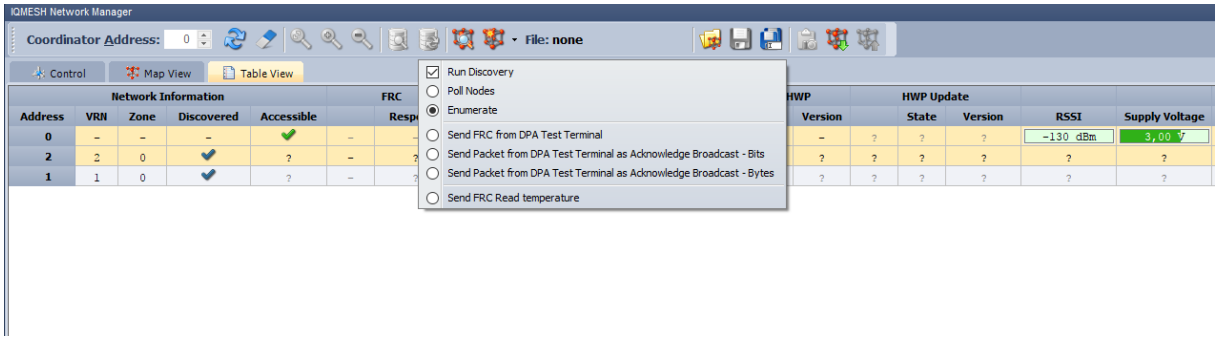
If the evaluation board DK-EVAL is charged and turned on (the jumper JP1 is set) and you press the pushbutton (SW1 or SW2), the appropriate red LED (LED1 or LED2) will light on. Otherwise, DK-EVAL is discharged. LED3 is on during charging and switched off when fully charged.




When you have your network created, you can use the features of the IQRF IDE environment to show the supply voltage of the accumulator (battery) inside the DK-EVAL.

In the IQMESH Network Manager, click the arrow to drop the menu **Perform selected operation** and select the **Enumerate** item. Then press the adjacent button to execute the command. Your network will be enumerated and the nodes will be asked for detailed information.

In the **Table View** tab, you will find all detailed information about your network. In the **Supply Voltage** column, you will find the information about the accumulator (battery) inside the DK-EVAL boards. If the color is red (supply voltage less than 2.9 V), DK-EVAL should be charged.



## 1.4 Summary

You have your IQRF network working and it is controlled by the IQRF IDE. Be sure you can collect sensory data from DDC-SE and control the relays – see the chapter 2.7.

The next step is to make the UP board working as an IQRF Gateway. The installation and configuration of the gateway is the subject of Part 2 – IQRF Gateway.

Share your ideas and solve problems with others on the IQRF Forum.

# Install your IQRF Gateway

This step-by-step guide is prepared for the UP bord. However, with minor modifications you can use the same process for any other Linux computer. First, we install an operating system on the UP board. Then we will install and configure basic services. Last, we will read data and control the development kits that are part of the UP-IQRF IoT Starter Kit.

## 2.1 Operating system

### 2.1.1 Install Ubilinux

To install Linux, prepare a USB flash drive with a capacity of at least 4 GB, a keyboard, a mouse, a monitor with an HDMI cable, and a connection to the Ethernet network.

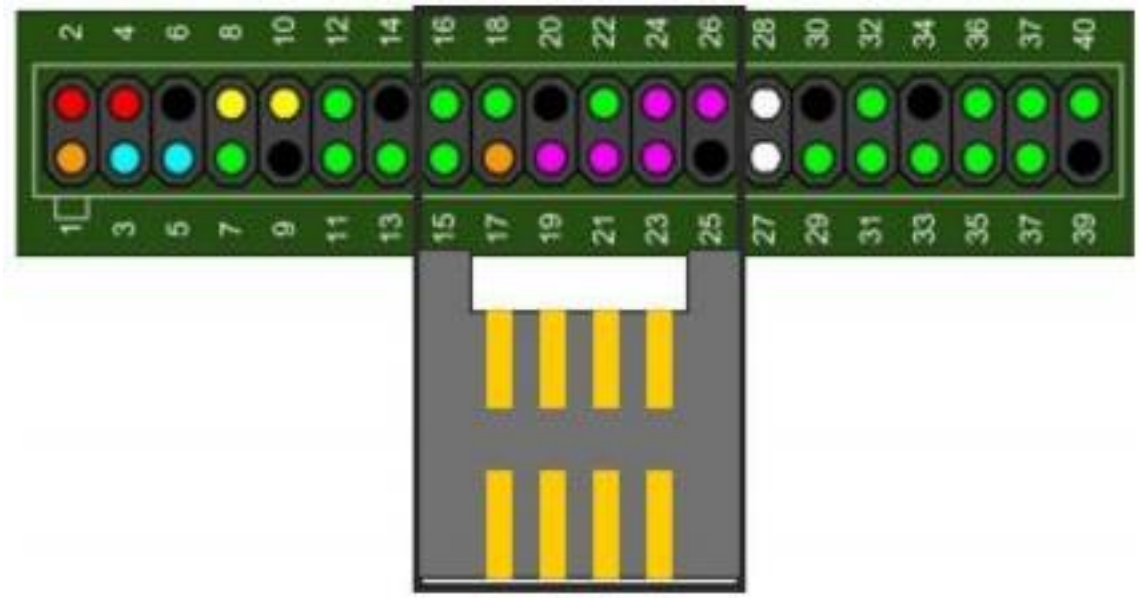
Download Ubilinux 4.0 for UP board and save it to your disk.

Then, download the Etcher software for burning the image of the operating system to a USB flash drive and install it.

After starting Etcher, select the image of the operating system and choose your USB flash drive to burn it on.


After burning the image to the USB flash drive, connect it to the UP board. Your monitor, keyboard and mouse should already be connected.

Connect the IQRF SPI board (adapter) to the GPIO pins right in the middle of the header of the UP board (you don't needto insert the coordinator here now) and turn it on.




If you have the operating system installed on your UP board already, press **F7** at the beginning and select booting from the USB flash drive. If there's nothing on your UP board, the installation will start automatically.

The installation continues automatically and doesn't last for more than 4 minutes.

```
Space in use:   1.3 MB = 2592 Blocks
Free Space:   535.5 MB = 1045984 Blocks
Block size:   512 Byte
Elapsed: 00:00:02, Remaining: 00:00:00, Completed: 100.00%, Rate:  39.81MB/min,
current block:       7872, total block:    1048576, Complete: 100.00%
Total Time: 00:00:02, Ave. Rate:   39.8MB/min, 100.00% completed!
Syncing... OK!
Partclone successfully restored the image (-) to the device (/dev/disk/by-partlabel/ESP)
Cloned successfully.
Partclone v0.2.89 http://partclone.org
Starting to restore image (-) to device (/dev/disk/by-partlabel/root)
Calculating bitmap... Please wait... done!
File system:  EXTFS
Device size:    5.0 GB = 1220608 Blocks
Space in use:   3.4 GB = 824482 Blocks
Free Space:     1.6 GB = 396126 Blocks
Block size:   4096 Byte
Elapsed: 00:02:06, Remaining: 00:00:00, Completed: 100.00%, Rate:   1.61GB/min,
current block:    1154391, total block:    1220608, Complete: 100.00%
Total Time: 00:02:06, Ave. Rate:    1.6GB/min, 100.00% completed!
Syncing... OK!
Partclone successfully restored the image (-) to the device (/dev/disk/by-partlabel/root)
Cloned successfully.
- Growing root partition...
e2fsck: Cannot continue, aborting.


~/.automated_script.sh  80.00s user 33.53s system 80% cpu 2:21.71 total
8 root@ubilinux4-installer ~ #
```

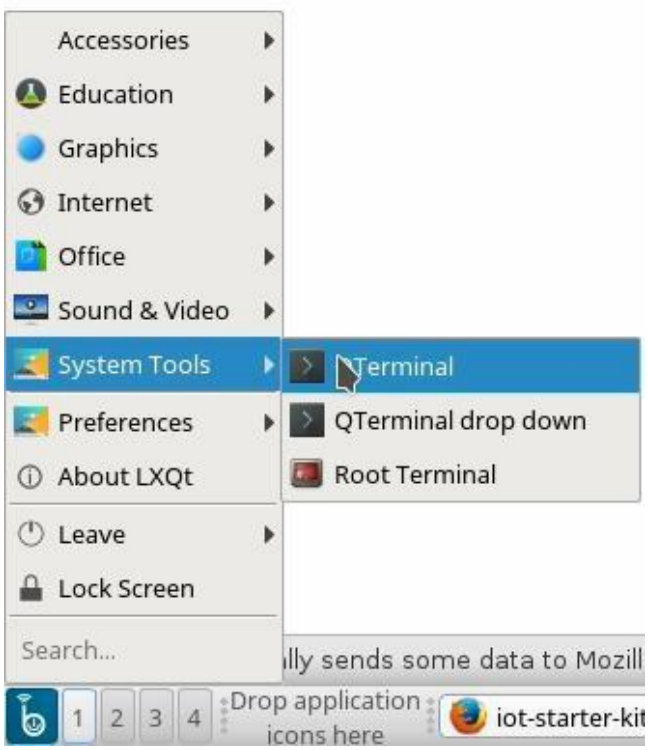
After the operating system is installed, the UP board turns off. Then, remove the USB flash drive, connect the UP board to the Ethernet network and turn it on again.

### 2.1.2 Update UbiLinux

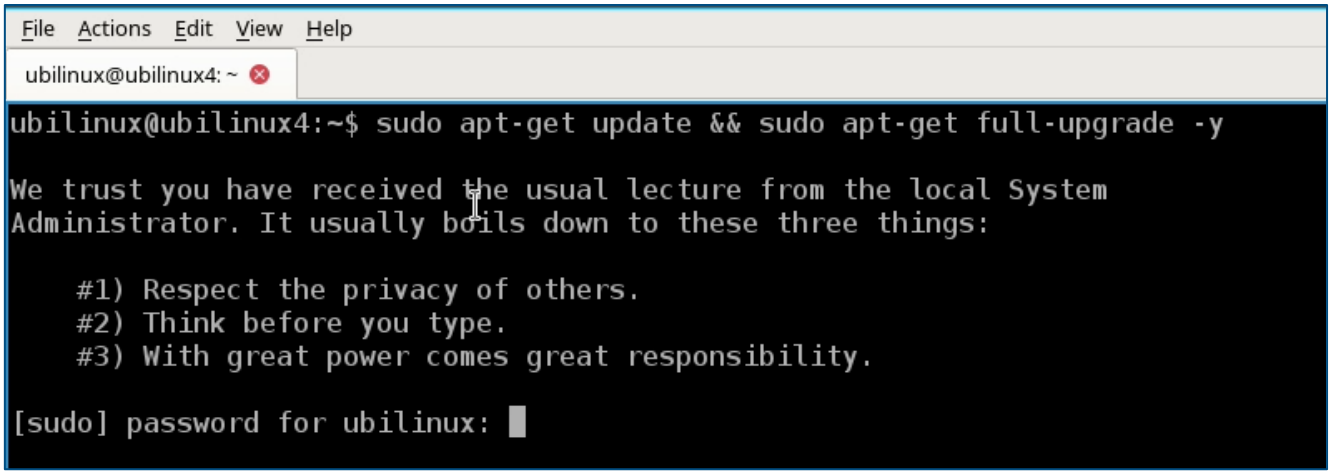
At this point, we have already installed the operating system. Log in to it with a default password – ubilinux.


We need to get the operating system updated. Copy the command for the update and paste it into the terminal.

**sudo apt-get update && sudo apt-get -y full-upgrade**



Enter the default password – ubilinux for user ubilinux.




## 2.2 MQTT Broker

### 2.2.1 Install MQTT Broker

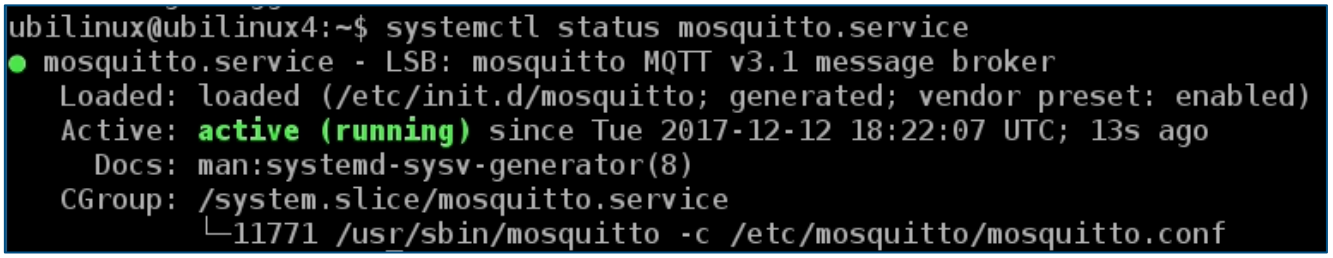
Install the MQTT Broker by using this command.

**sudo apt-get install -y mosquitto mosquitto-clients**

### 2.2.2 Confirm the MQTT Broker is running

Verify that the MQTT Broker is running.

**systemctl status mosquitto.service**




## 2.3 IQRF Gateway Daemon

### 2.3.1 Install the IQRF Gateway Daemon

Install the IQRF Gateway Daemon. There are four commands that you need to enter into the terminal. The time of the installation mostly depends on the speed of your internet connection.

**sudo apt-get install -y dirmngr**

**sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys**

**9C076FCC7AB8F2E43C2AB0E73241B9B7B4BD8F8E echo "deb http://repos.iqrfsdk.org/debian**

**stretch stable" | sudo tee -a /etc/apt/sources.list.d/iqrf-daemon.list sudo apt-get update &&**

**sudo apt-get install -y iqrf-daemon**

### 2.3.2 Confirm IQRF Gateway Daemon is running

Verify that the IQRF Gateway Daemon is running. Press Q to leave the listing.

**systemctl status iqrf-daemon.service**

```
ubilinux@ubilinux4:~$ systemctl status iqrf-daemon.service
● iqrf-daemon.service - IQRF daemon iqrf_startup
   Loaded: loaded (/lib/systemd/system/iqrf-daemon.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2017-12-12 18:23:37 UTC; 16s ago
 Main PID: 13048 (iqrf_startup)
    Tasks: 11 (limit: 4915)
   CGroup: /system.slice/iqrf-daemon.service
           └─13048 /usr/bin/iqrf_startup /etc/iqrf-daemon/config.json
```

## 2.4 IQRF Gateway Daemon WebApp

### 2.4.1 Install IQRF Gateway Daemon WebApp

Now install the web application for the IQRF Gateway Daemon configuration. Copy and paste the commands one after the other.

**cd /home/ubilinux**

**git clone https://github.com/iqrfsdk/iqrf-**

**daemon-webapp.git cd iqrf-daemon-**

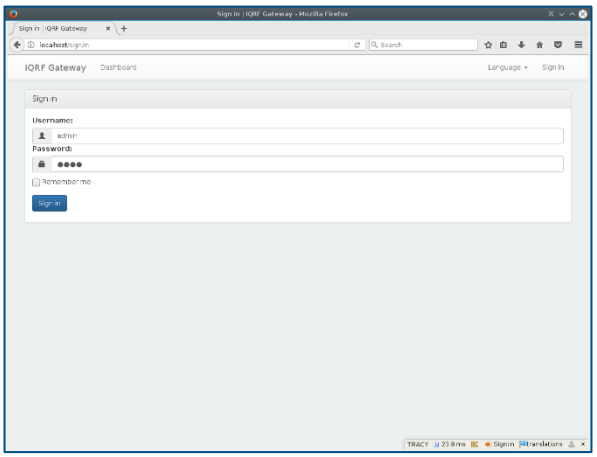
**webapp/install/**

**sudo python3 install.py -d debian -v 9**

### 2.4.2 Confirm IQRF Gateway Daemon WebApp is running

Verify that the web application is running by typing a localhost address in your web browser on the UP board. Log in as **admin** with the password **iqrf**.

**http://localhost/en**

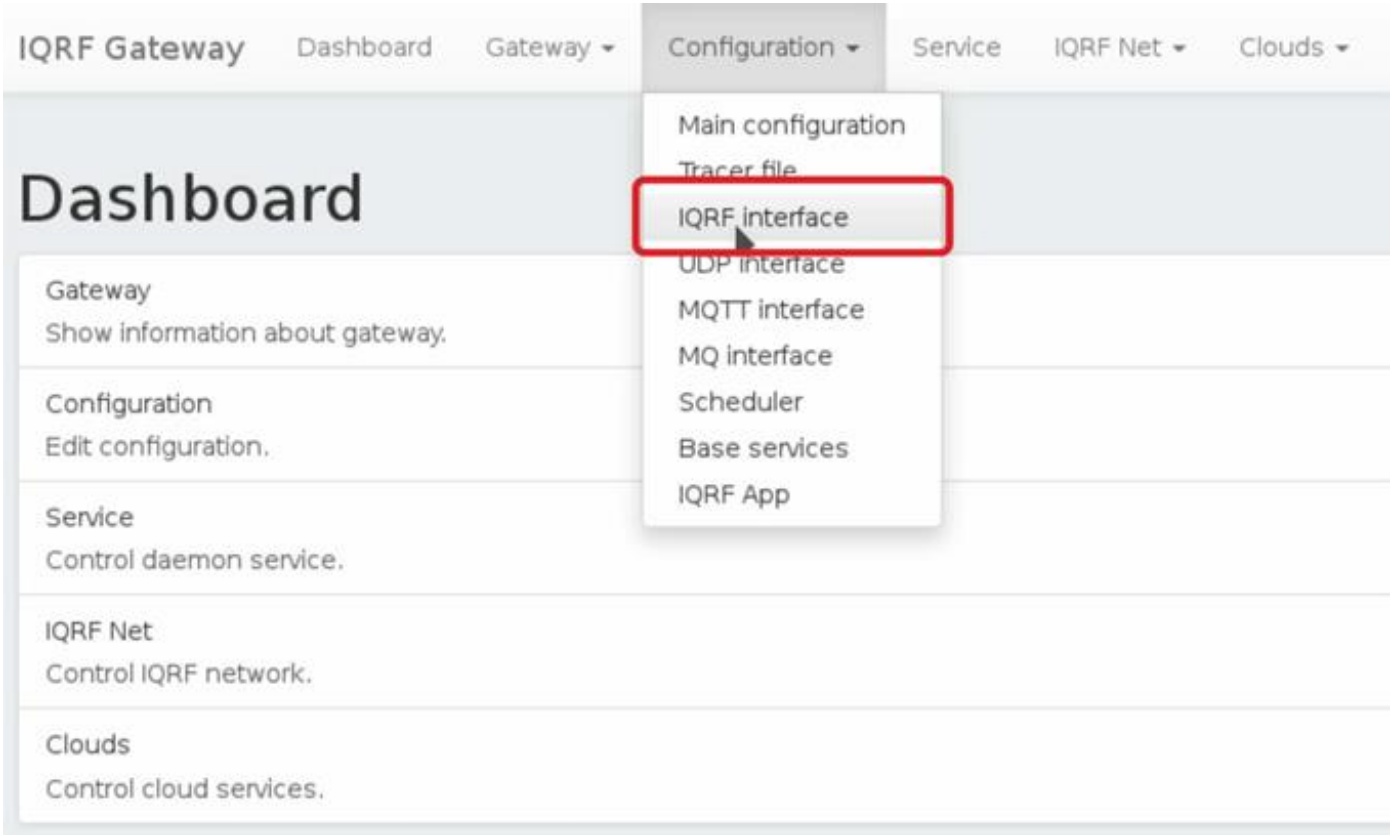


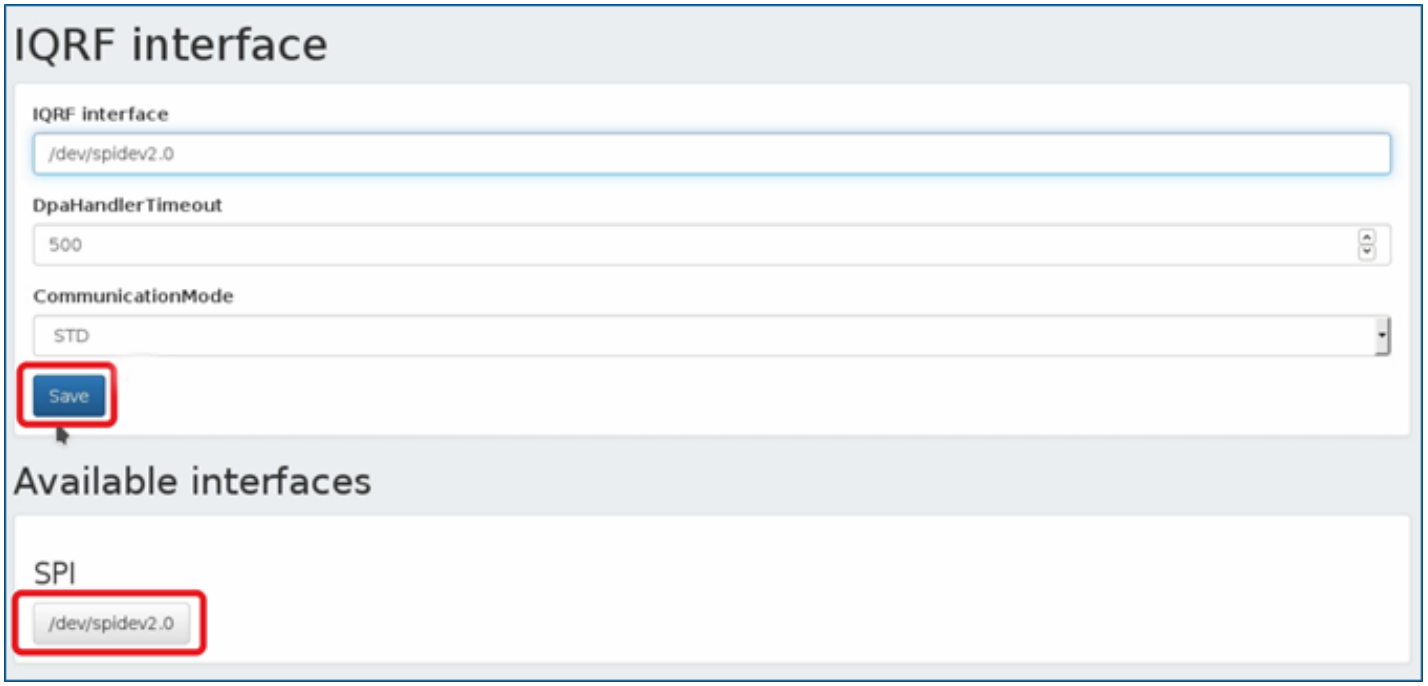
## 2.5 SPI interface

### 2.5.1 Configure IQRF SPI interface

Now configure the connection to the IQRF network through the SPI interface. Click on the **IQRF interface** in the **Configuration** menu, then click on the available **SPI** interface (in the picture below it is marked with number 1) and save the configuration by clicking on the **Save** button.

**http://localhost/en/config/iqrf**

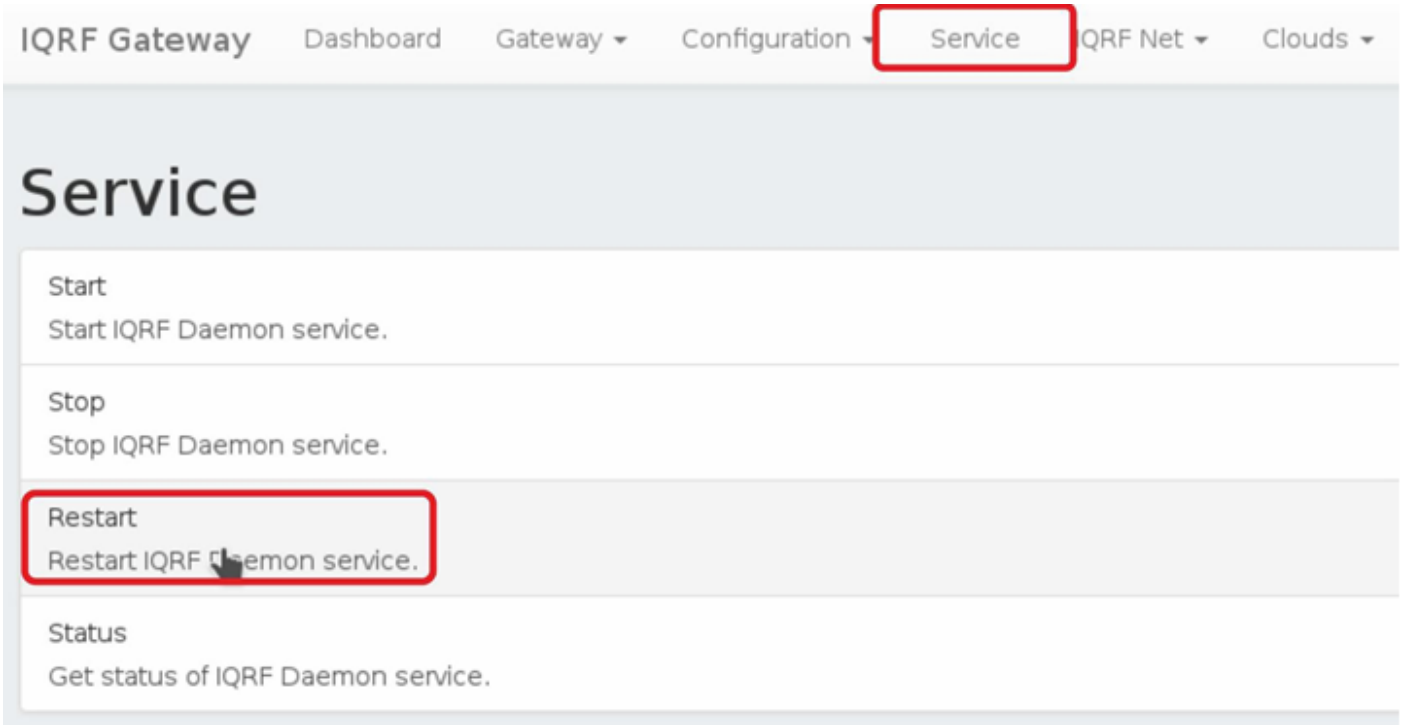



IQRF interface
IQRF interface
/dev/spidev2.0
DpaHandlerTimeout
500
CommunicationMode
STD
Save
Available interfaces
SPI
/dev/spidev2.0

### 2.5.2 Restart IQRF Gateway Daemon

Restart the IQRF Gateway Daemon by clicking on **Restart** in the **Service** menu. You can see here that the daemon has been restarted.

**http://localhost/en/service**




## 2.6 Node.js

### 2.6.1 Install Node.js

Now install the **Node.js**. This is done by a set of commands you copy and paste one by one into the terminal.

**cd /home/ubilinux**

**git clone https://github.com/iqrfsdk/iot-starter-kit.git**

**cd iot-starter-kit/install** Enter ubilinux as a passwor

**sudo cp etc/lsb-release-debian**

**/etc/lsb-release sudo apt-get install**

**curl**

**curl -sL**

**https://deb.nodesource.com/setup_6.x |**

**sudo -E bash - sudo apt-get install nodejs**

**sudo cp etc/lsb-release-ubilinux /etc/lsb-release**

## 2.7 Node-RED

### 2.7.1 Install Node-RED

Now install Node-red. Copy the two prepared commands and paste them into the terminal.

**sudo npm install -g --**

**unsafe-perm node-red**

**sudo npm install -g**

**pm2**

### 2.7.2 Start Node-RED

Run Node-RED with these two commands.

**cd /home/ubilinux**

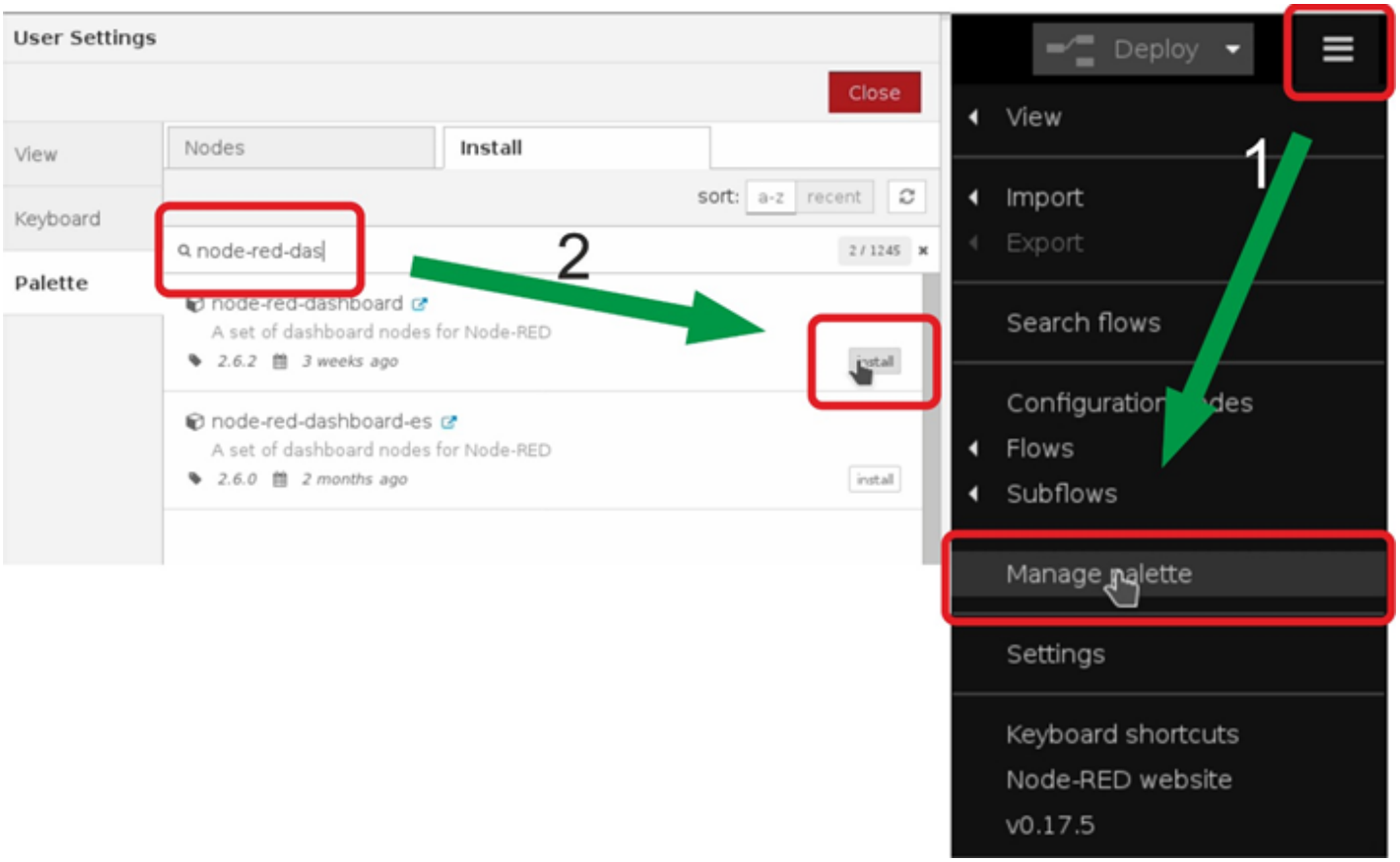
**pm2 start /usr/bin/node-red**

### 2.7.3 Add Node-RED dashboard

Now create a **Node-RED** dashboard environment.

In the internet browser of the UP board, enter the localhost address with the port **1880** and select the **Manage palette** item from the menu. Find the **node-red-dashboard** and install it.

**http://localhost:1880**


User Settings
Close
View
Keyboard
Palette
Nodes
Install
sort: a-z recent
node-red-das
2 / 1245
node-red-dashboard
A set of dashboard nodes for Node-RED
2.6.2 3 weeks ago
install
node-red-dashboard-es
A set of dashboard nodes for Node-RED
2.6.0 2 months ago
install
2
Deploy
View
Import
Export
Search flows
Configuration nodes
Flows
Subflows
Manage palette
Settings
Keyboard shortcuts
Node-RED website
v0.17.5
1

### 2.7.4 Run IoT-Starter-Kit flow

Run the prepared example for the UP-IQRF IoT Starter Kit. The acquired data will be visualized in the dashboard and the two relays can be controlled by using buttons.

**cd /home/ubilinux/iot-starter-kit/install**

**cp up-board/node-red/* /home/ubilinux/.node-red**

**pm2 restart node-red**

### 2.7.5 Allow Node-RED to run after reboot

Use these prepared commands to set up Node-RED to start automatically after switching on the UP board.

**p**

**m**

**2**

**s**

**a**

**v**

**e**

**p**

**m**

**2**

**s**

**t**

**a**

**r**

**t**

**u**

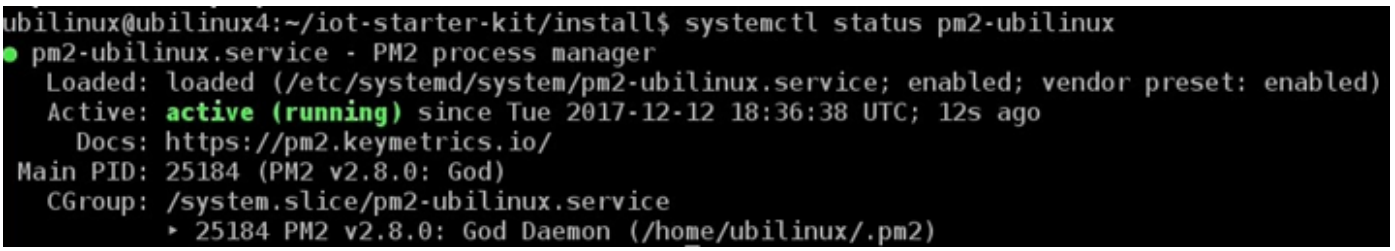
**p**

**sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u ubilinux --hp /home/ubilinux**

### 2.7.6 Confirm Node-RED is running

Verify that Node-RED is running.

**systemctl status pm2-ubilinux**




### 2.7.7 Check Node-RED dashboard

**Now you need to connect the IQRF network to the UP board.**

***Caution:*** *The IQRF transceiver can be plugged/unplugged into/from the SIM connector on the IQRF SPI board (adapter) while powered off only.* If you haven't done it yet, insert the IQRF coordinator to the IQRF SPI board and turn on the UP board.

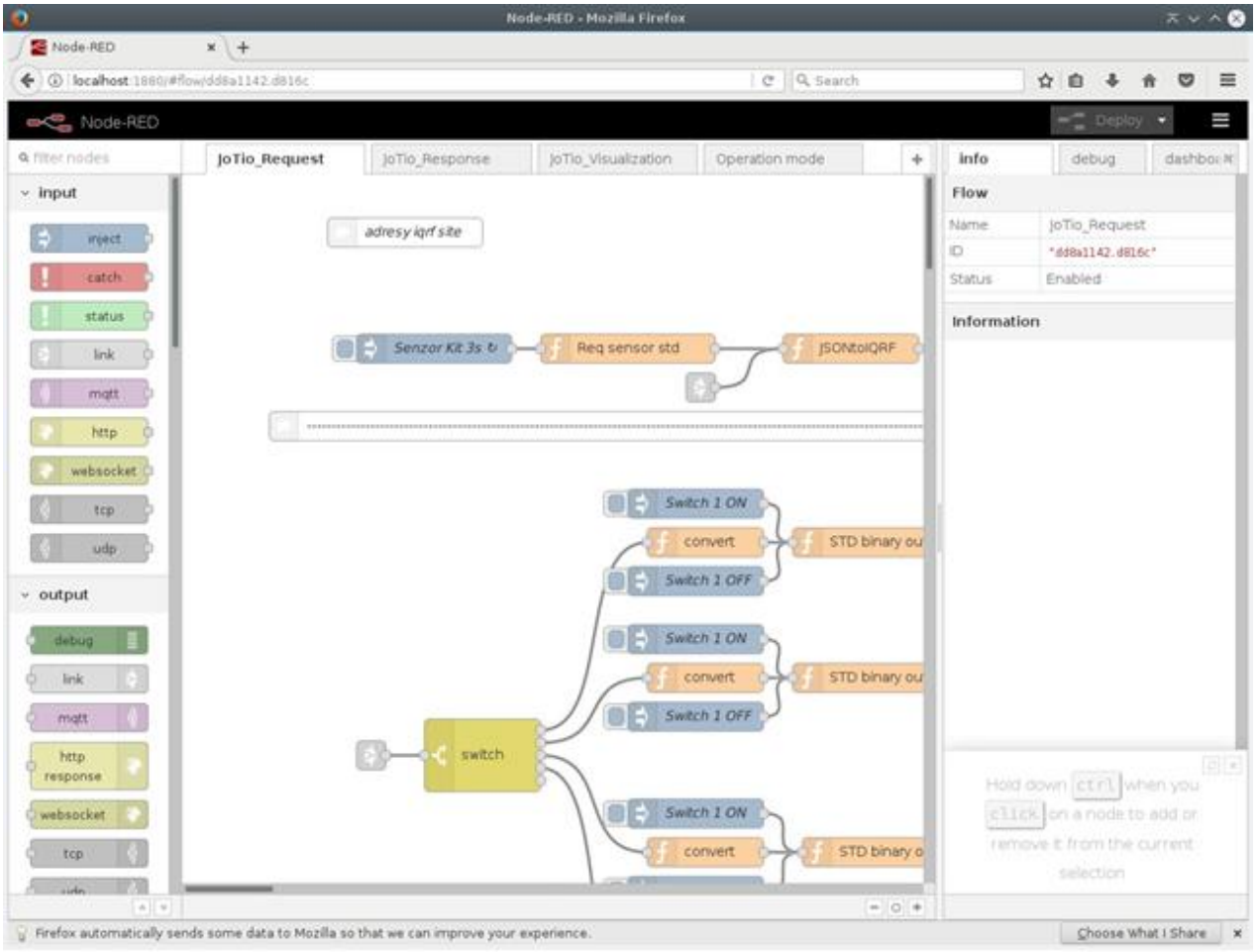
**Check the dashboard at the localhost address with the port 1880/ui. If you have your IQRF network with the sensor and relay kit ready, you can see the measured values on the dashboard and switch the relays on and off.**

**http://localhost:1880/ui**


### 2.7.8 Check Node-RED flow

At the localhost address, port 1880, the Node-RED administration environment can be used to modify your flows and dashboards.

**http://localhost:1880**

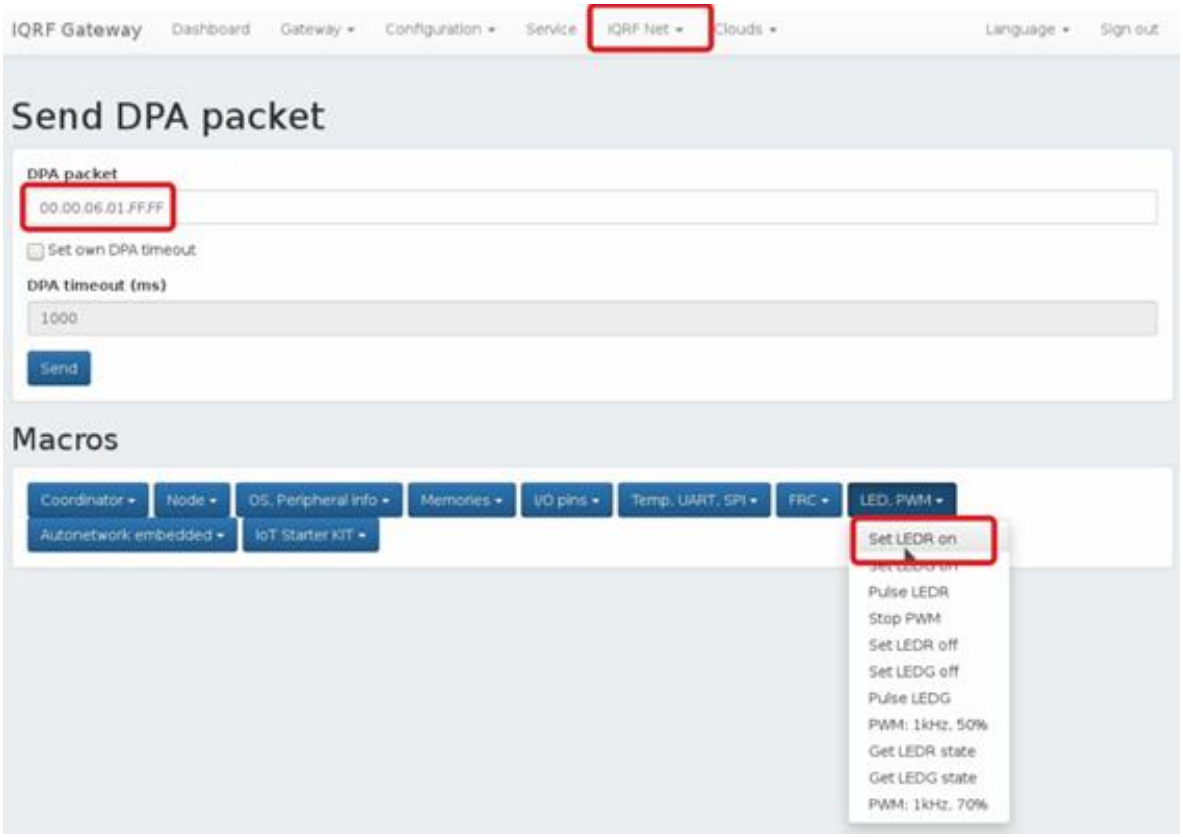

Node-RED - Mozilla Firefox
Node-RED
localhost:1880/#flow/dd8a1142.d816c
Search
Node-RED
Deploy
filter nodes
JoTio_Request
JoTio_Response
JoTio_Visualization
Operation mode
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
output
debug
link
mqtt
http response
websocket
tcp
adresy iqrf site
Senzor Kit 3s t
Req sensor std
JSONtoIQRF
Switch 1 ON
convert
STD binary ou
Switch 1 OFF
Switch 1 ON
convert
STD binary ou
Switch 1 OFF
switch
Switch 1 ON
convert
STD binary o
info
debug
Flow
Name
JoTio_Request
ID
"dd8a1142.d816c"
Status
Enabled
Information
Hold down ctrl when you click on a node to add or remove it from the current selection
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share

## 2.8 Test the functionality

### 2.8.1 Send DPA Packet

**Verify the functionality of controlling the IQRF network from the web application. Click on the Send DPA Packet in the IQRF Net menu and select any command here, such as turning on the red LED on the coordinator. You can also modify the command.**

**http://localhost/en/iqrfnet/send-raw**




You can easily double check that the command has been executed.


3V
CR2032HF-90
AMI-Aptio
PASS
CE

### 2.8.2 Inspect JSON messages between Node-RED and IQRF Gateway Daemon

Check up the DPA commands in JSON format running between Node-RED and the IQRF Gateway Daemon.

Listen for all JSON DPA RAW Requests:
**mosquitto_sub -t Iqrf/DpaRequest**

Listen for all JSON DPA RAW Responses
**mosquitto_sub -t Iqrf/DpaResponse**

Insert the command into the terminal to observe the ongoing communication.

```
ubilinux@ubilinux4:~/iot-starter-kit/install$ mosquitto_sub -t Iqrf/DpaRequest
{"ctype":"dpa","type":"raw","request":"01.00.5e.01.ff.ff.ff.ff.ff.ff","timeout":1000}
{"ctype":"dpa","type":"raw","request":"01.00.5e.01.ff.ff.ff.ff.ff.ff","timeout":1000}
```

## 2.9 Check more examples

**cd /home/ubilinux**

**git clone https://github.com/iqrfsdk/iqrf-daemon-examples.git cd iqrf-daemon-examples**

## 2.10 Summary

**We made an IQRF Gateway from an UP board. Be sure you can control your IQRF network from the UP board – see chapter 8. Apparently, you can connect this gateway to any cloud solution such as Microsoft Azure, IBM Cloud Platform, Amazon Web Services or anything else. How to do that, is the topic of the following part.**

# UP-IQRF IoT Starter Kit – Part 3:

## Connect to the cloud – AWS IoT

The IoT Starter Kit is designed in the way to be connectable to different clouds via bidirectional MQTT channel. So, you can collect, store, process and visualize data in a cloud or you can send your commands to the IQRF network remotely. In this part, we will configure the UP board to communicate with the Amazon Web Services (AWS) through the MQTT channel.

### 3.1 Local network

Connect your UP board to your local network so it can obtain an IP address using DHCP. In the following steps, you will enter this address into your web browser on your computer (which is in the same local network as the UP board) and configure your gateway through the IQRF Daemon Web application.

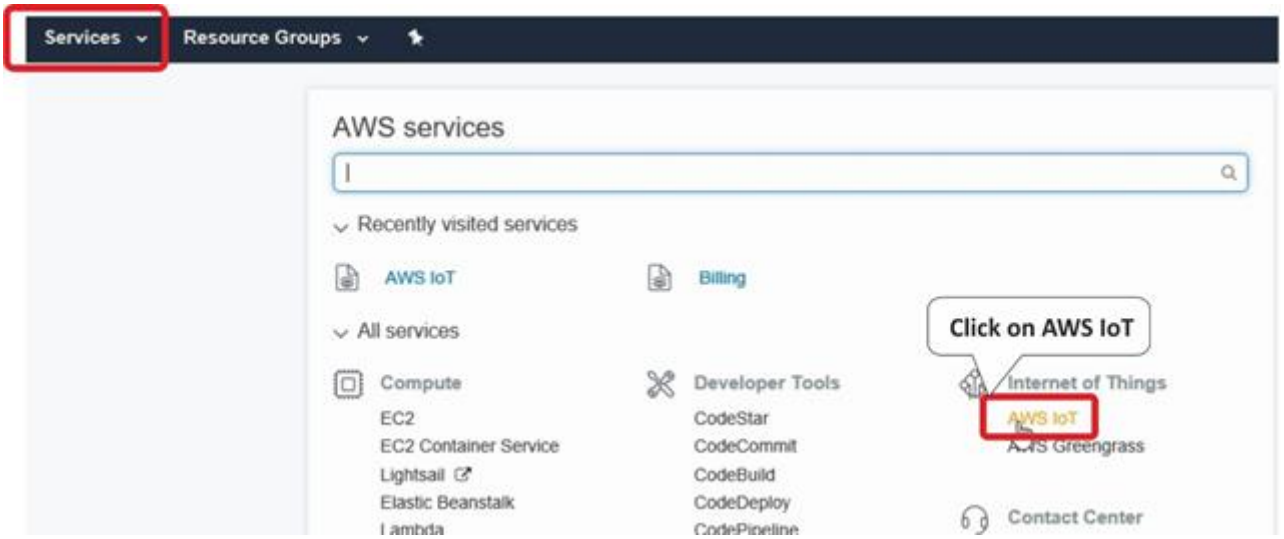
## 3.2 Amazon Web Services account

First, create an Amazon Web Services account (aws.amazon.com). You must fill in your personal or company data and add your credit card details. Your credit card will be used for payments in a case you exceed the limits of the selected services.

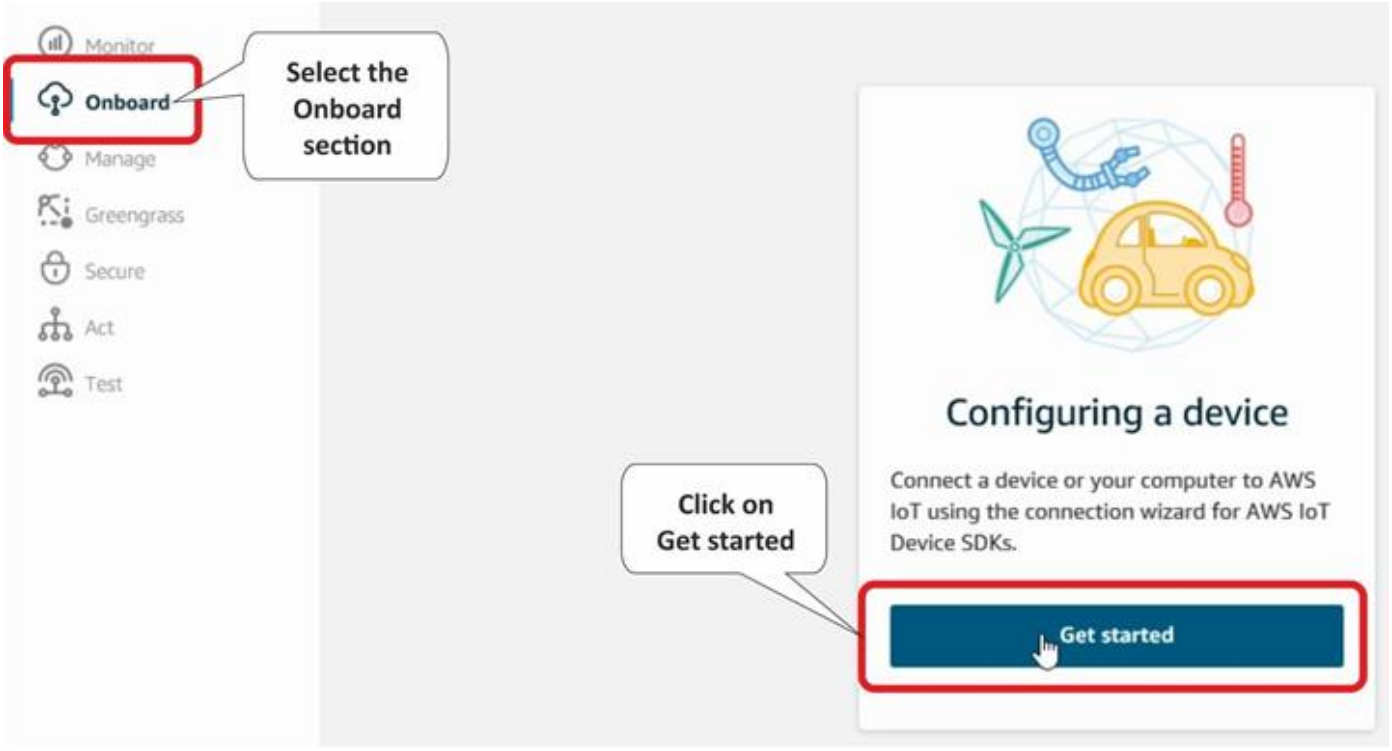
## 3.3 Set up the connection

To set up the connection between AWS and your UP board, you need to do some configuration steps on both sides. In **Services**, in the **Internet of Things** section of AWS, you will find **AWS IoT**.

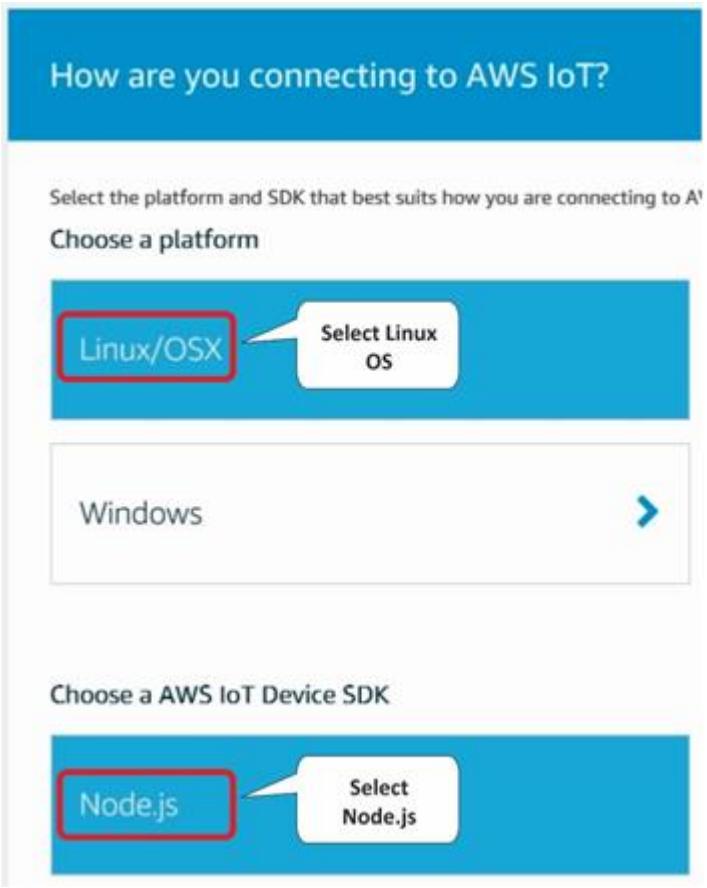
***Note**: the environment of AWS may look different because of often changes and its personalization. This guide shows the status of March 2018. You need to look for appropriate items to configure the MQTT connection.*




Click on **Get started** in the **Onboard** section. You will register your device, download the connection kit, and configure and test the connection with your device.

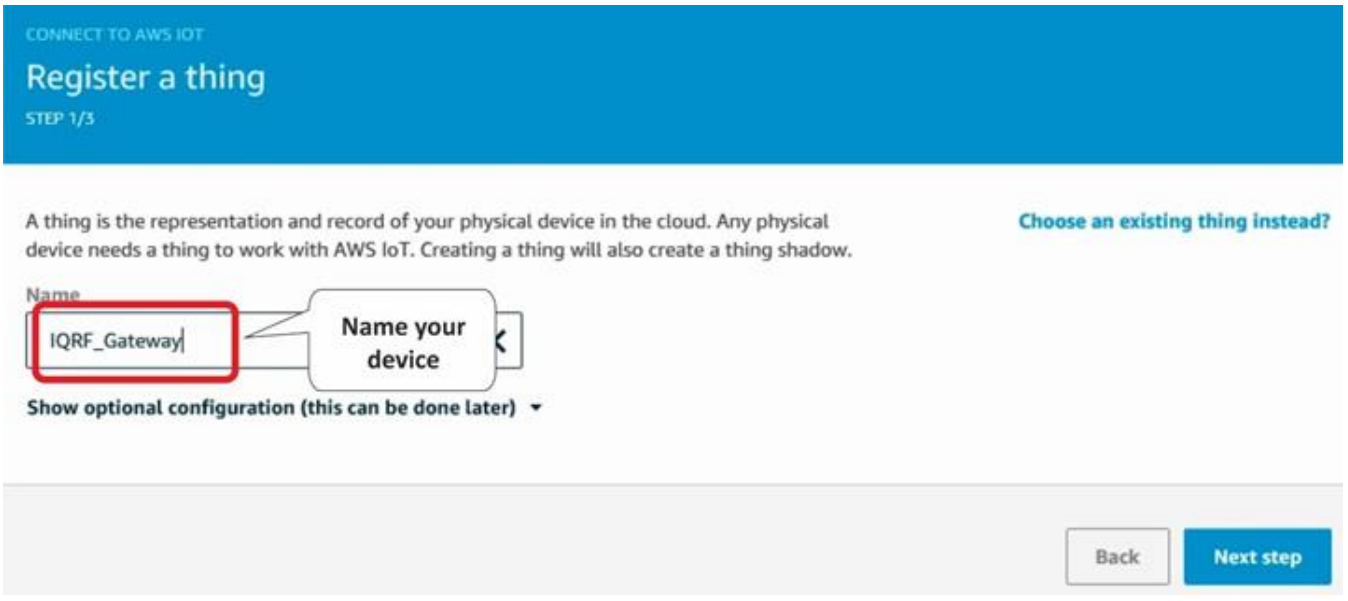


Set up how you will be connected to the AWS IoT. Select the **Linux** operating system and **Node.js** as the AWS IoT Device SDK.

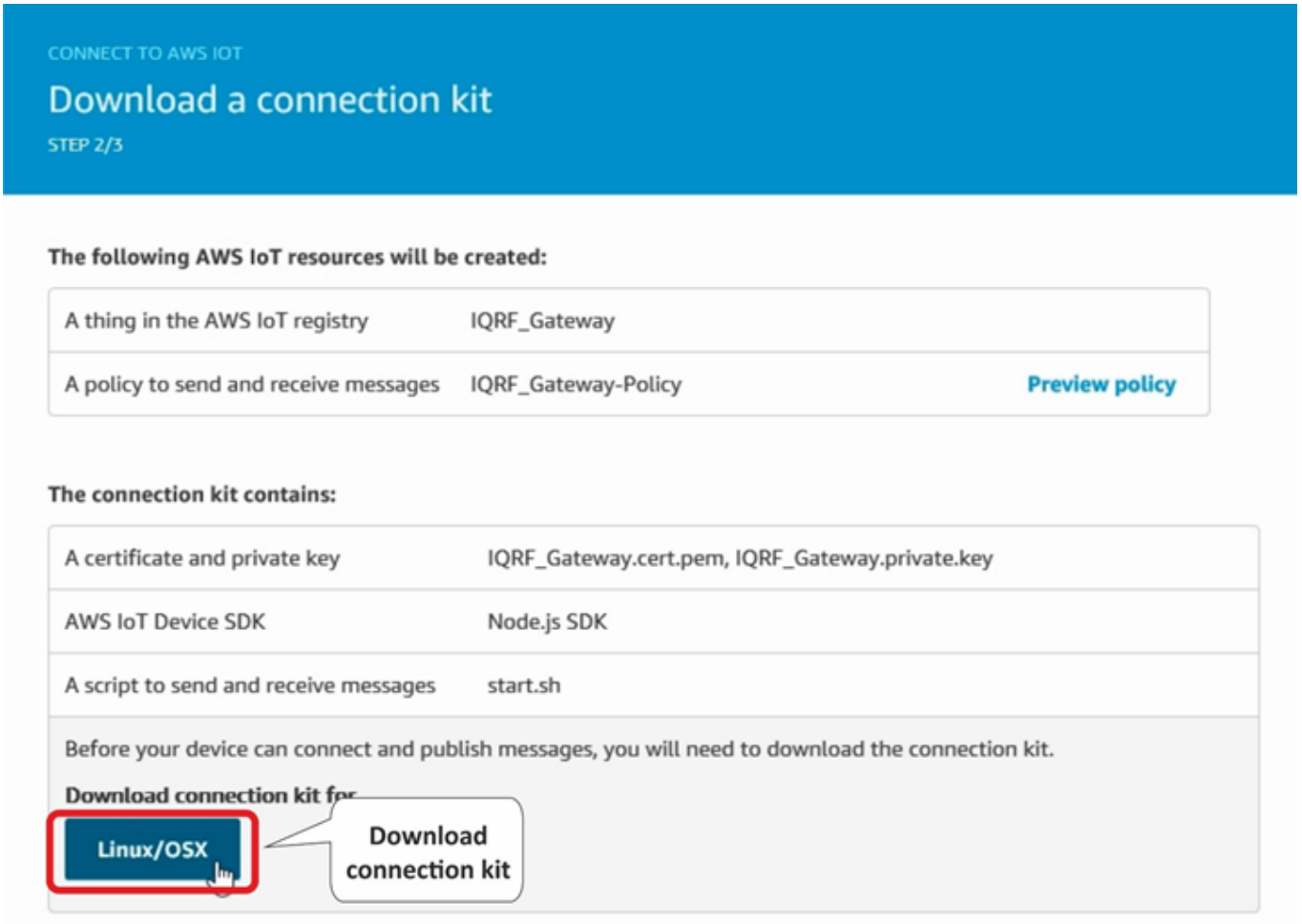



Enter the name of your connected device.

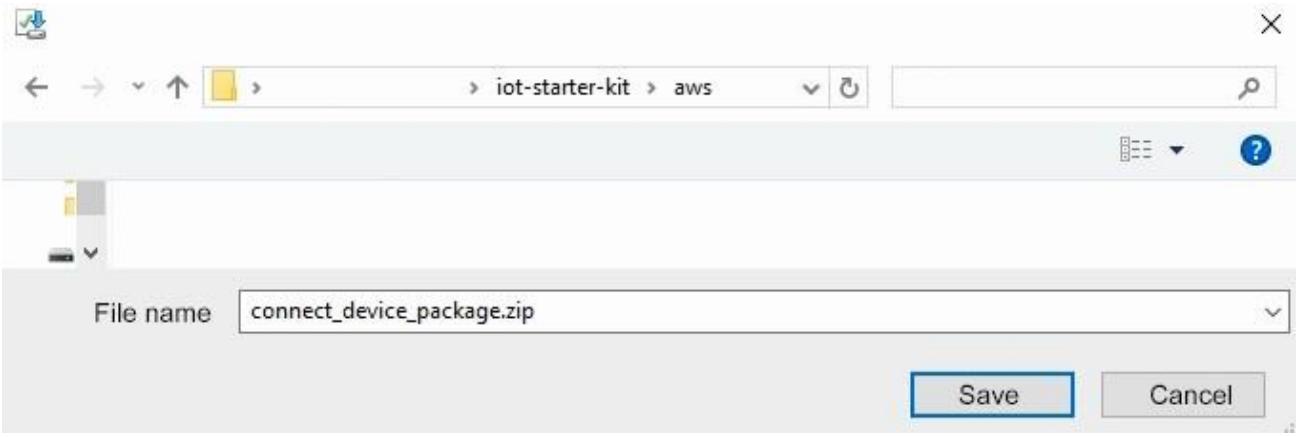
*Note*: You can choose your own name. In that case in later steps, you need to use the given name.



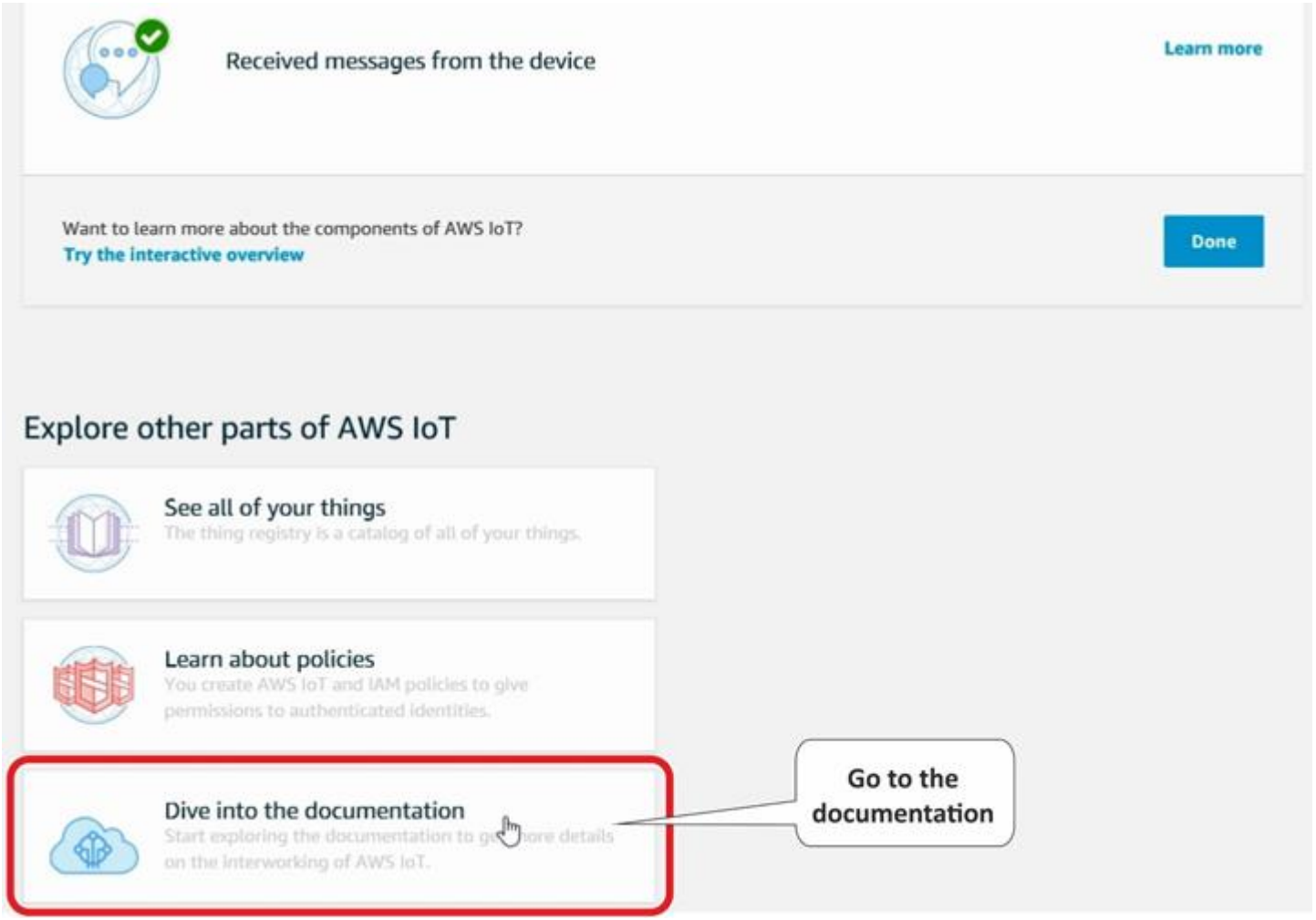
Download the connection kit to get a certificate and keys for a secure MQTT connection.




Save and unzip this file to your computer. Store the certificate and the keys for further use.



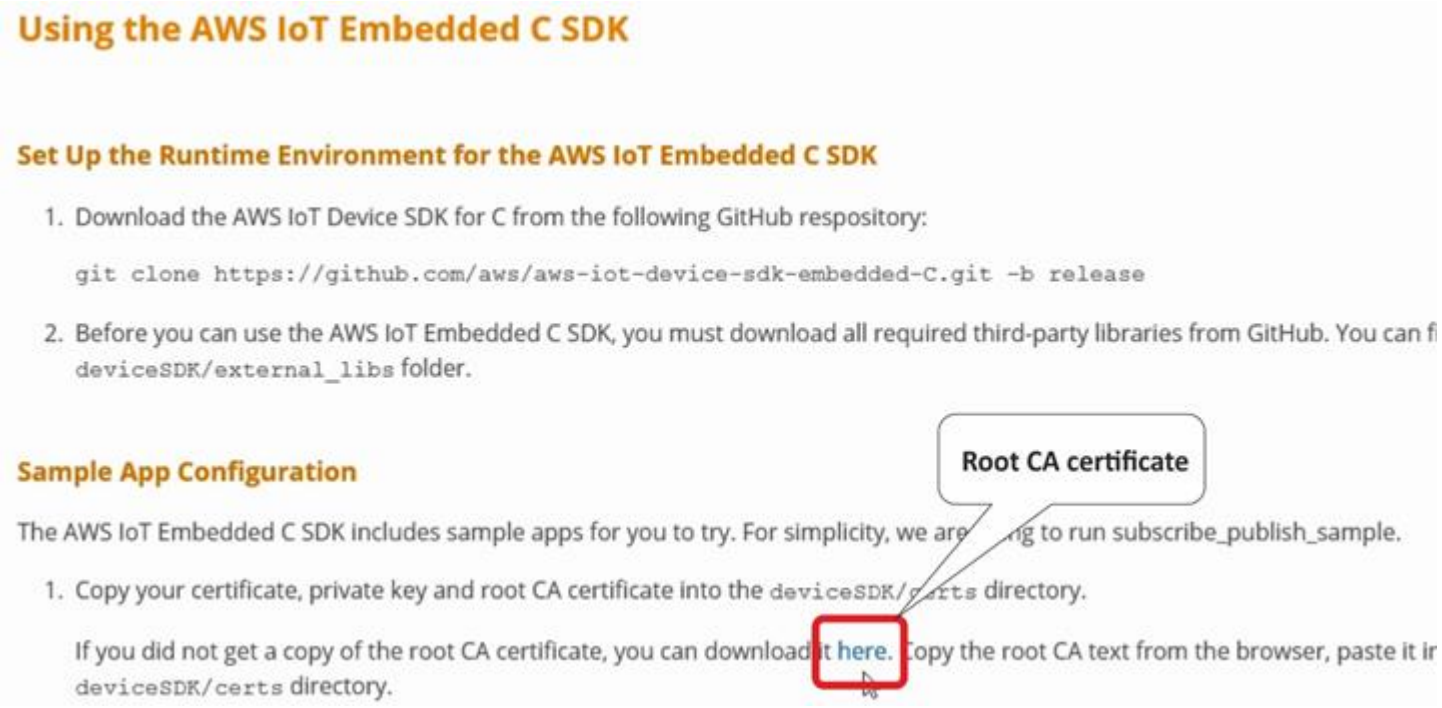
After the saving process, go to the documentation.




Here, look up the **Download root CA** string. Search in the **Entire site** to be sure to find it.

In the search results, find the article **Using the AWS IoT Embedded C SDK**. The number of records in a search result can exceed the page limit so you need to go through more pages.

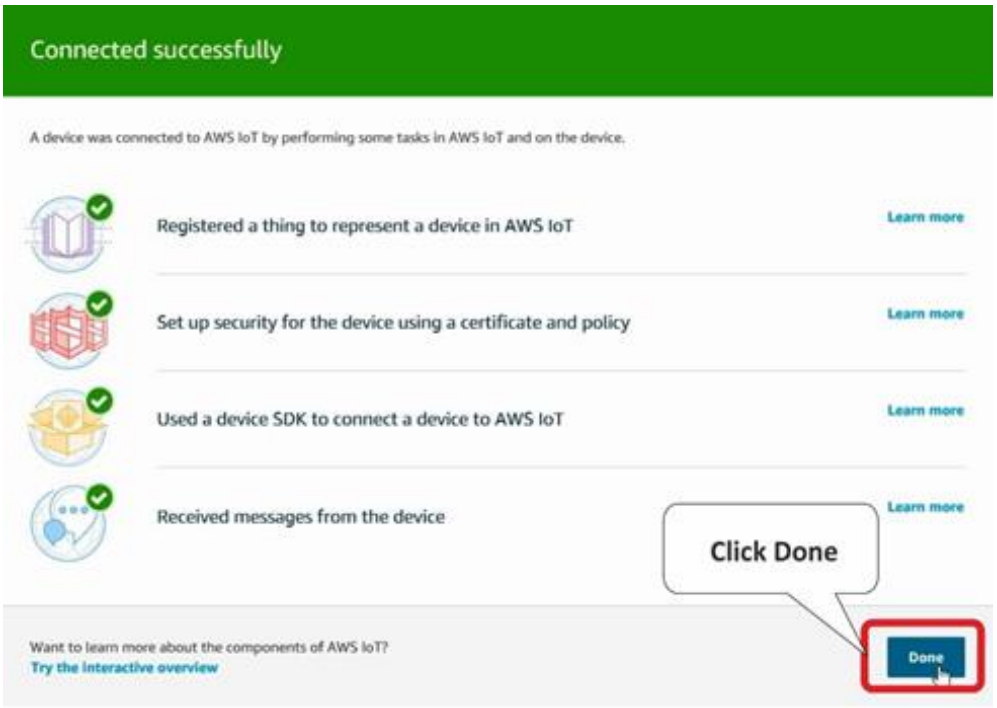

Here you can find the root certificate.



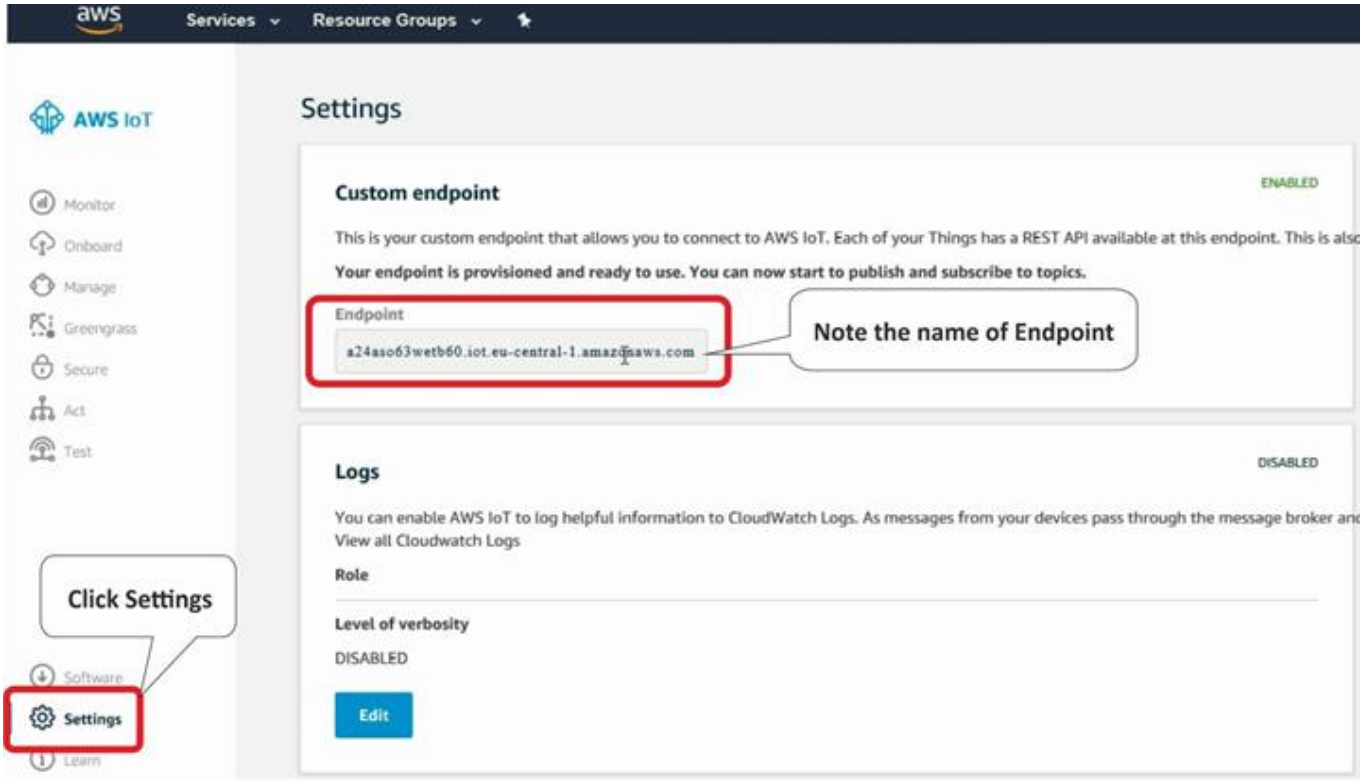
Copy the string to a text file and save it as **rootCA.pem** to the directory with other certificates on your computer.

*Note*: You can choose your own name. In that case in later steps, you need to use the given name.


The message Connected successfully is shown automatically after finishing the process of configuring a device. Next, click Done.



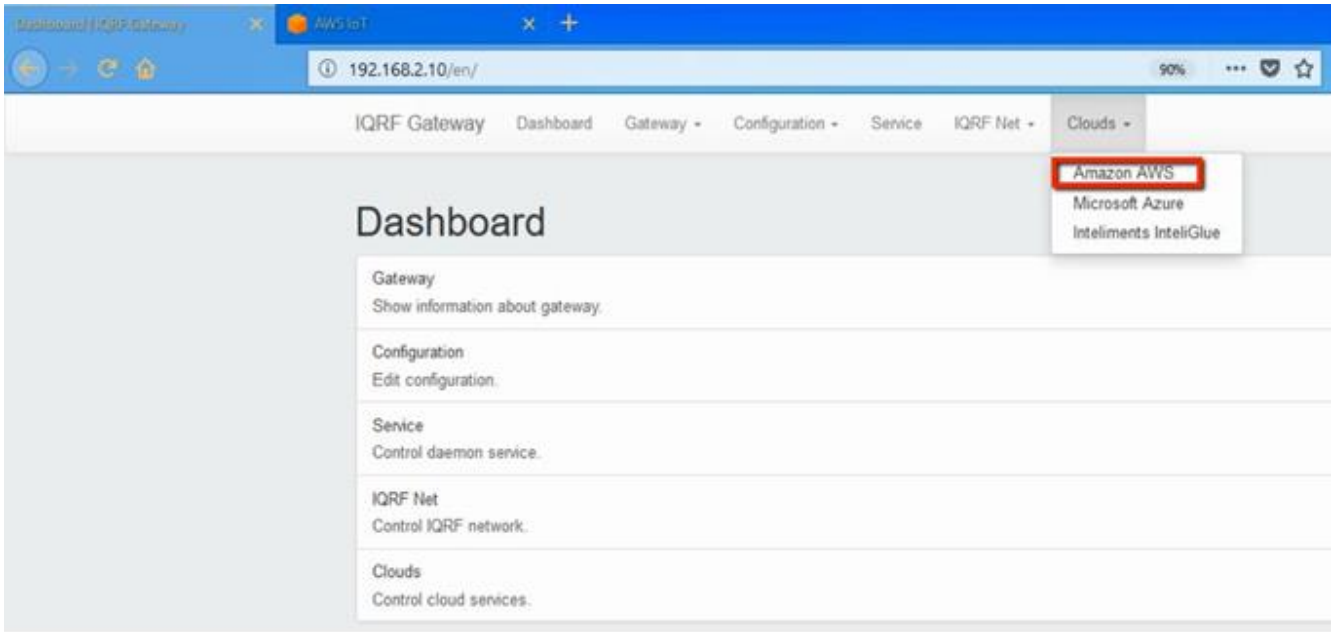
In the **Settings**, note the name of your **endpoint** because you will need it for the UP board




configuration.

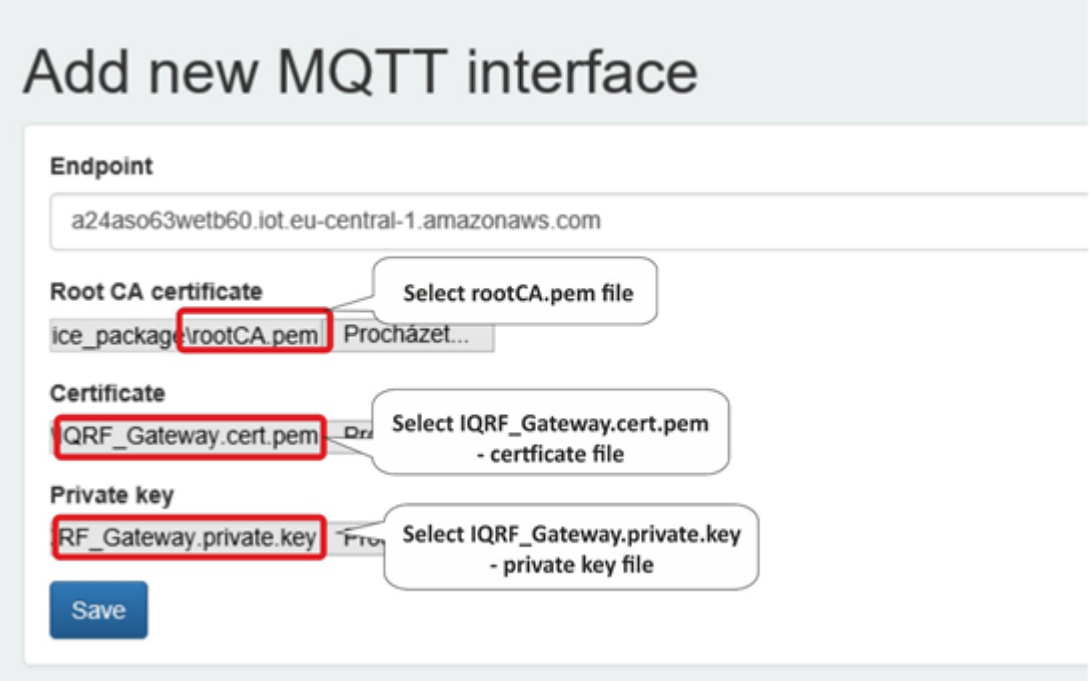
Files **rootCA.pem** (root certificate), **IQRF_Gateway.private.key** (private key file), and **IQRF_Gateway.cert.pem** (certificate file) should be already unzipped. We will transfer them to the UP board through the IQRF Gateway Daemon web application.

In the web browser on your computer, insert the IP address of your UP board, and login to it as *admin* with the password *iqrf*. Ask your network administrator how to find out your IP address or you can use common network tools. In the **IQRF Gateway Daemon web application,** click on the **Amazon AWS** item in the **Clouds** menu.



Enter the name of the **Endpoint** (find it in Settings of your AWS IoT). Select **rootCA.pem** as a Root CA certificate, **IQRF_Gateway.cert.pem** as a Certificate and **IQRF_Gateway.private.key** as a Private key file. Save the configuration.

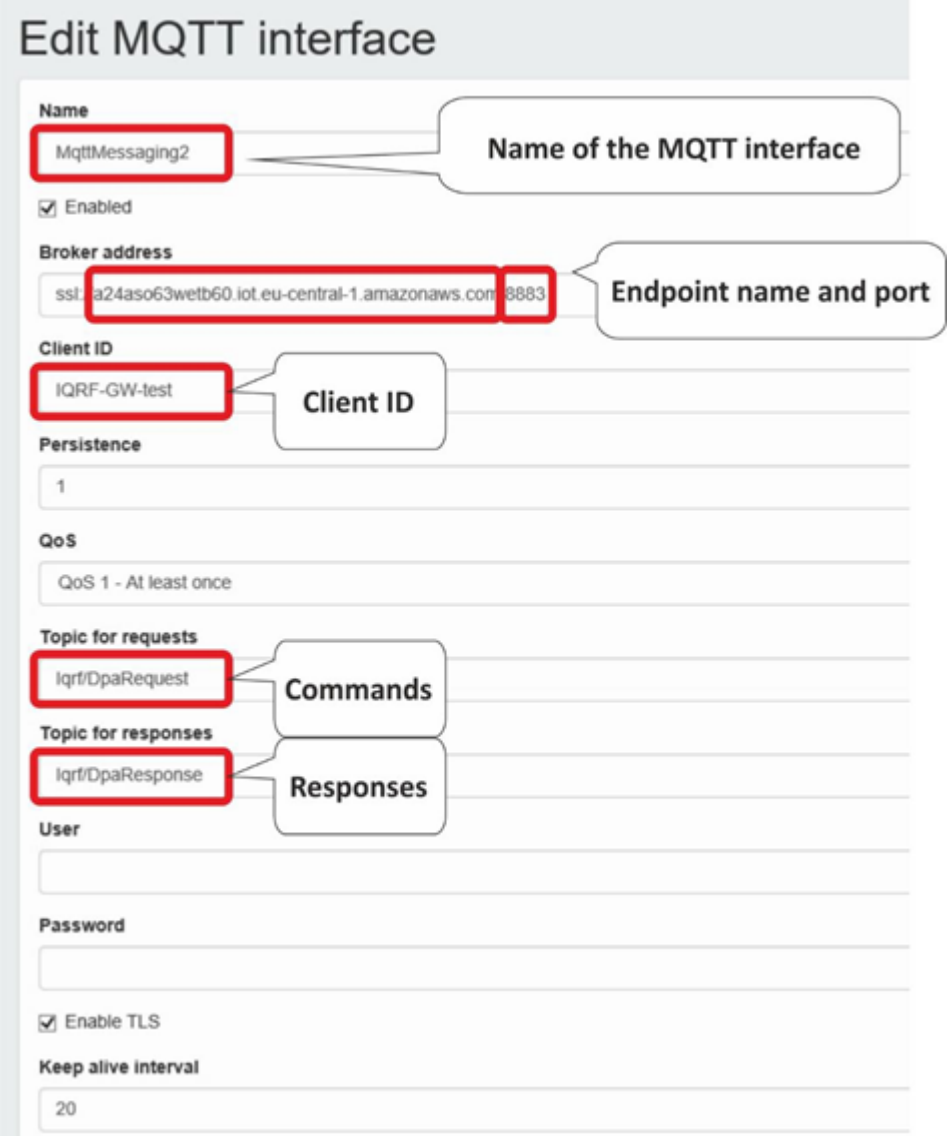
*Note*: If you named your virtual device in AWS with a different name, the names of files contain this name instead of
IQRF_Gateway.




Inspect the new MQTT interface for AWS.

The address of the **endpoint** goes after the **SSL** protocol and at the end of Broker address is the port number **8883**.

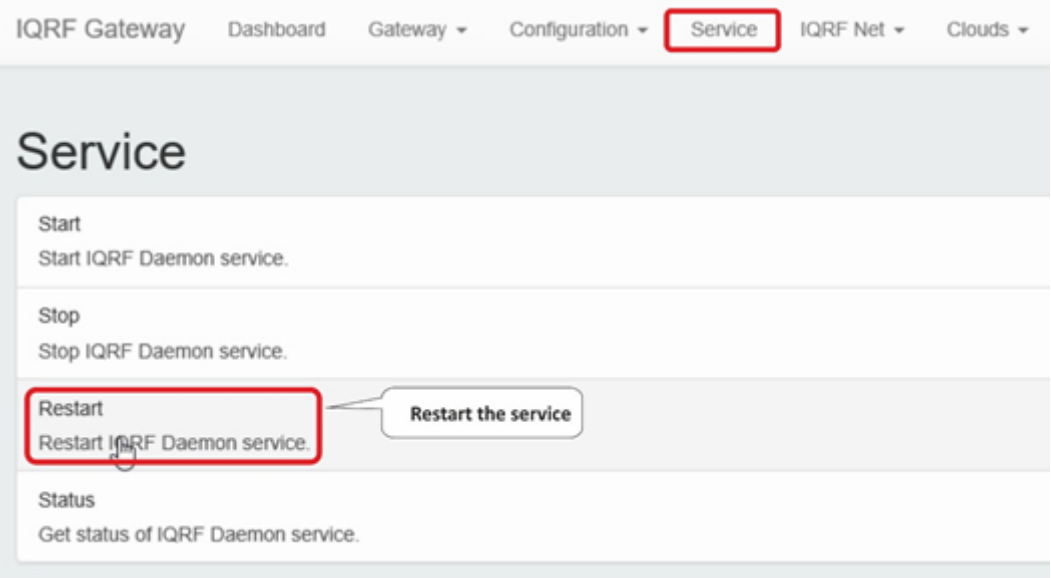
**Iqrf/DpaRequest** is set as the topic for commands and **Iqrf/DpaResponse** is set as the topic for responses.




***Note**: your files and name of the endpoint may differ from the names shown in the pictures.*

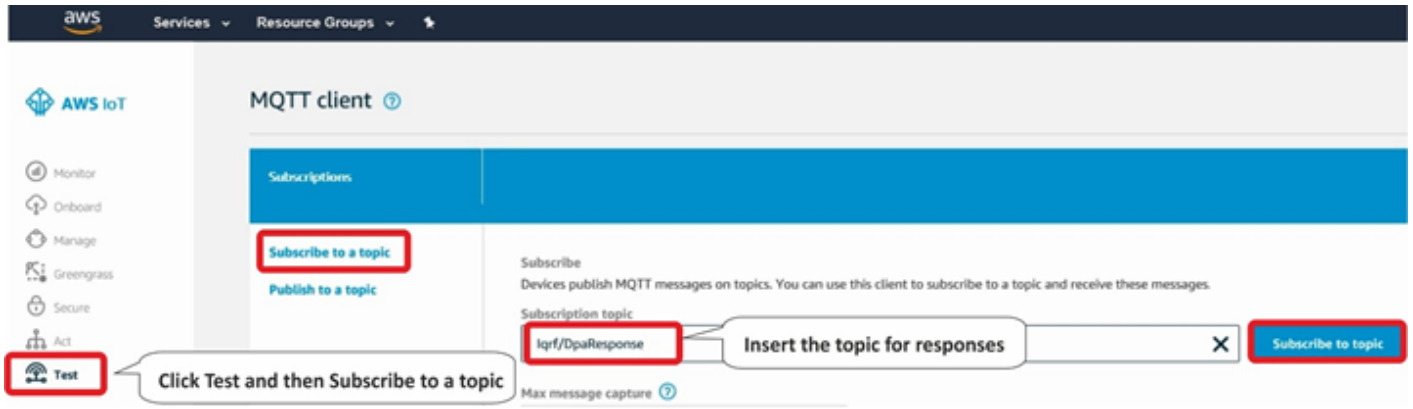
There are the **timeout**, the **minimum,** and **maximum** connections set, and the path to the uploaded files that set up a secure connection between the gateway and the cloud. Check the **Enable server certificate authentication** item.


Restart IQRF Gateway Daemon. After restarting, check the at the status of the UP board if the selected services are running.




## 3.4 Test the connection

In the web browser on your computer, in AWS IoT, click **Test**. Enter the **Iqrf/DpaResponse** to the Response topic to retrieve the gateway responses and click on **Subscribe to topic**.




To send commands from the cloud to the gateway, set the **Iqrf/DpaRequest** as the topic for


requests. The gateway will expect commands in this topic.

Insert a DPA packet in the JSON format into the text box and click on **Publish to topic**. In our example, we sent a command to turn on the red LED on the coordinator.


```
{
"
c
t
y
p
e
"
:

"
d
p
a
"
,

"
t
y
p
e
"
:

"
r
a
w
"
,
"msgid":
"1510754
980",
"request"
:
"00.00.06
.01.FF.FF"
,
"request_
ts":       "",
```

"confirma
tion": "",
"confirma
tion_ts":
"",
"response
": "",
"response_ts": ""
}

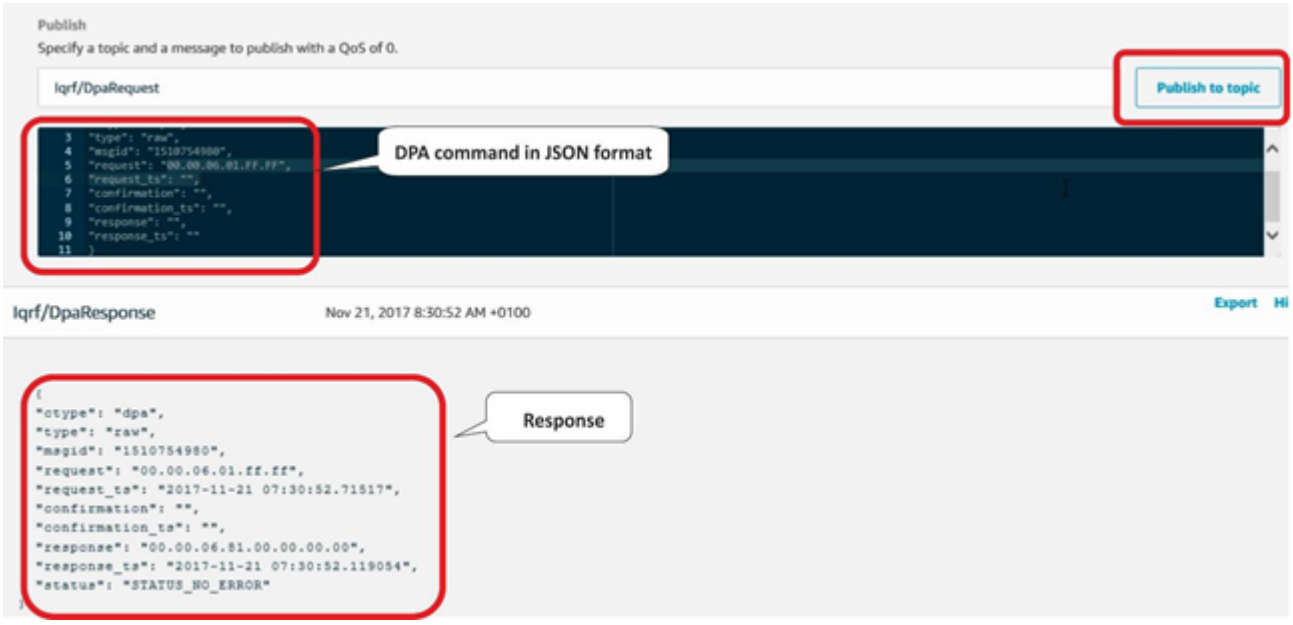
In the "*request*" item, you can insert other DPA commands for the network control and monitoring. You can find these commands in macros for the IoT Starter Kit or you can set them up in the Terminal window in the IQRF IDE.

Examples:

- Collecting all sensoric data from the Node #1 with the connected DDS-SE kit:
  {"ctype": "dpa","type": "raw","msgid": "1510754980","request": "01.00.5E.01.FF.FF.FF.FF.FF.FF","request_ts": "","confirmation": "","confirmation_ts": "","response": "","response_ts": ""}
- Turning on both relays on the Node #2 with connected DDC-RE kit:
  {"ctype": "dpa","type": "raw","msgid": "1510754980","request": "02.00.4B.00.FF.FF.0C.00.00.00.01.01","request_ts": "","confirmation": "","confirmation_ts": "","response": "","response_ts": ""}
- Getting temperature from Node #3:
  {"ctype": "dpa","type": "raw","msgid": "1510754980","request": "03.00.0A.00.FF.FF","request_ts": "","confirmation": "","confirmation_ts": "","response": "","response_ts": ""}

For more information about macros and the IQRF network read the IoT Starter Kit – Part 1: Build your IQRF network.

We can see that the gateway picked up and executed the command and sent a confirmation with "No Error" into the **Iqrf/DpaResponse** topic.




We can visually double check the result of this command. The red LED turned on.

## 3.5 Summary

The bidirectional communication between the IQRF network and the Amazon Web Services is up and running. Now it's just up to you to use it for your own IoT solution. In the next parts, we will show you how to add other sensors and actuators of our industrial partners ($CO_2$ sensor, wirelessly controlled power socket etc.).

From factory, IQRF transceivers have the following default settings: TX power: 7, RX filter: 0, RF channel A: 52. With these settings (TX power, RX filter), an area of 500 m radius in open space can be covered with the wireless IQRF signal.