

# UP-IQRF IoT Starter Kit – Part 2: Install your IQRF Gateway

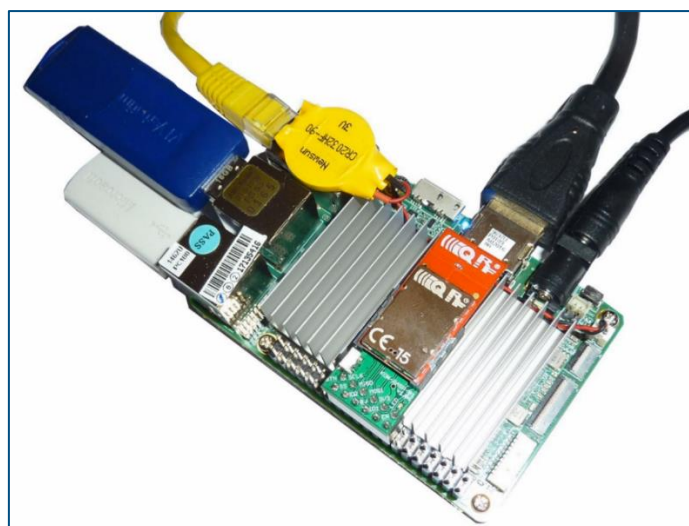
**Note:** If the PDF Guide is opened in a viewer mode, we strongly recommend downloading it and open on your computer locally to have hyperlinks functional and to be able to copy strings. The Download button you will find at the top of the page with a PDF preview.

This step-by-step guide is prepared for the UP board but 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. And last we will read data and control the development kits that are part of the UP-IQRF IoT Starter Kit.

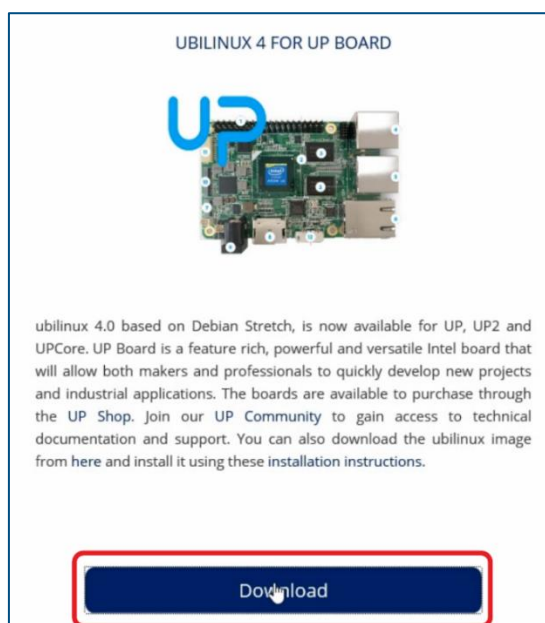
## 1 Operating system

### 1.1 Install Ubinlinux

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 the [Ubinlinux 4.0](#) for UP board and save it to your disk.

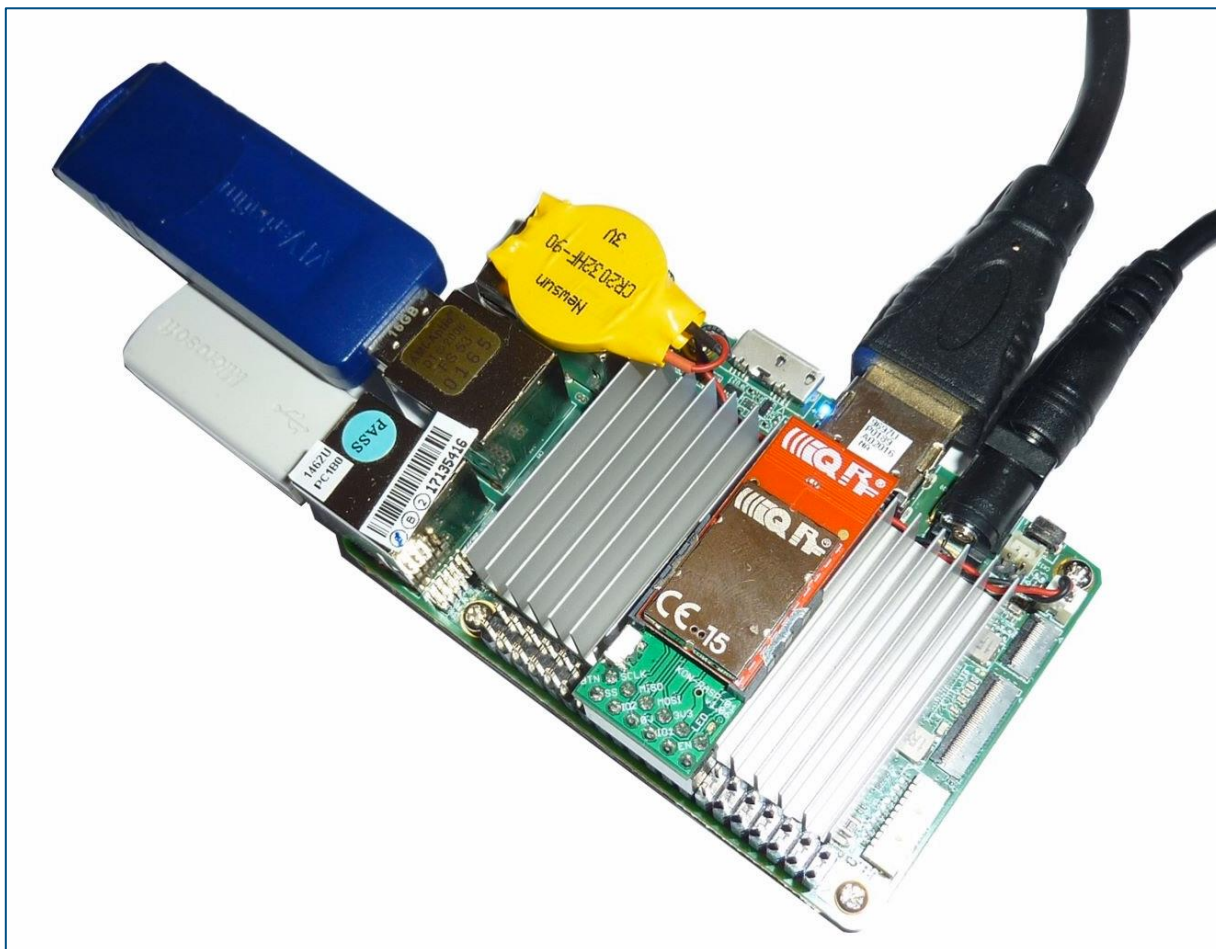


Then, download the [Etcher](#) software for burning the image of the operating system to a USB flash drive and install it.

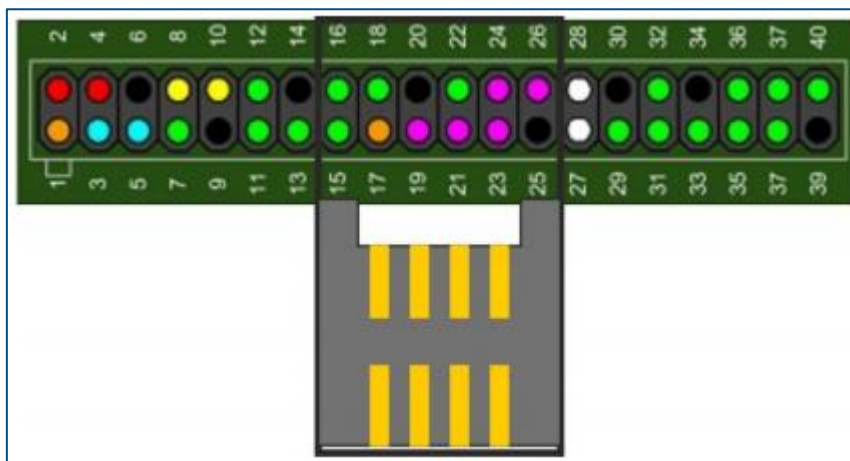
After starting Etcher, select the image of the operating system and choose your USB flash drive to burn it on.



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

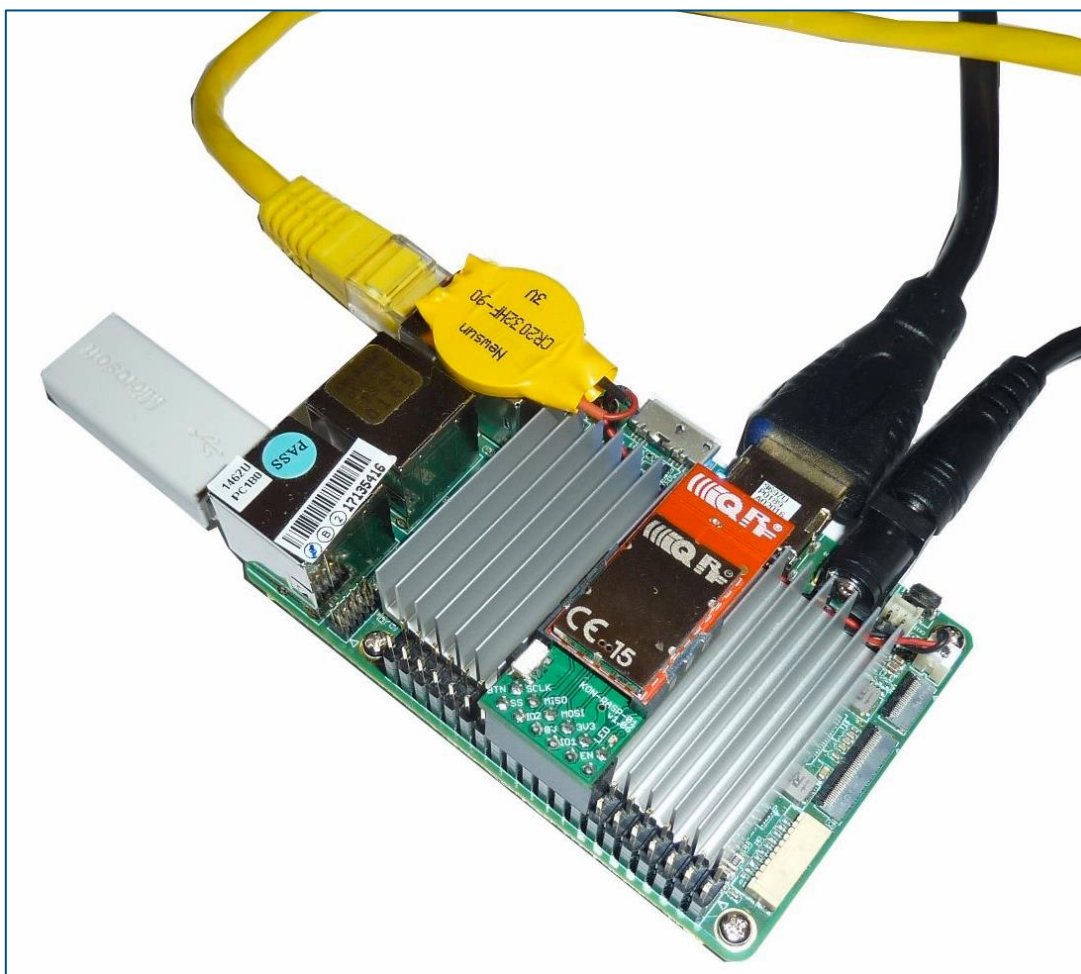


The installation continues automatically and doesn't last for more than 4 minutes.

```
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@ubinux4-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.



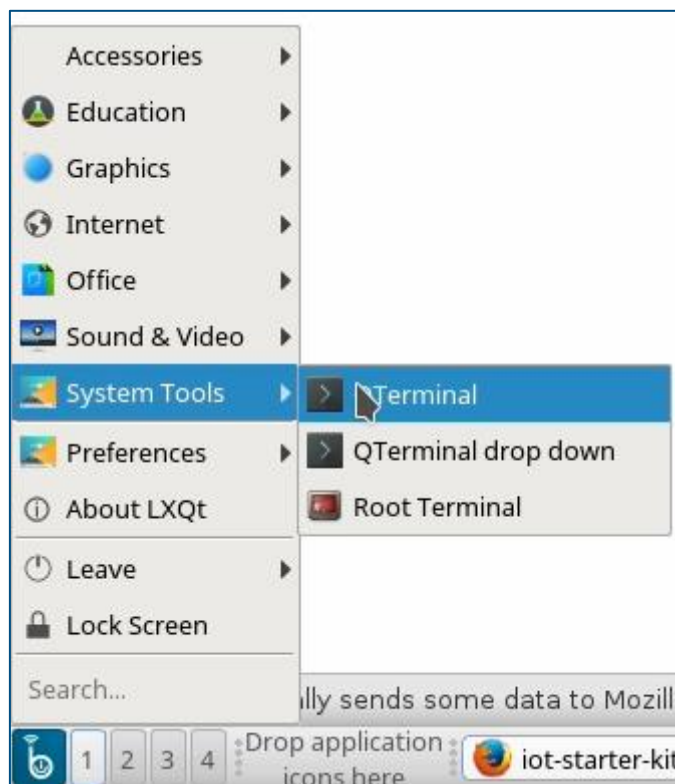
### 1.2 Update UbiLinux

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.

```
File Actions Edit View Help
ubilinux@ubilinux4: ~
ubilinux@ubilinux4:~$ sudo apt-get update && sudo apt-get full-upgrade -y
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for ubilinux: 
```

## 2 MQTT Broker

### 2.1 Install MQTT Broker

Install the MQTT Broker by using this command.

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

### 2.2 Confirm the MQTT Broker is running

Verify that the MQTT Broker is running.

```
systemctl status mosquitto.service
```

```
ubilinux@ubilinux4:~$ systemctl status mosquitto.service
● mosquitto.service - LSB: mosquitto MQTT v3.1 message broker
   Loaded: loaded (/etc/init.d/mosquitto; generated; vendor preset: enabled)
   Active: active (running) since Tue 2017-12-12 18:22:07 UTC; 13s ago
     Docs: man:systemd-sysv-generator(8)
   CGroup: /system.slice/mosquitto.service
           └─11771 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
```

## 3 IQRF Gateway Daemon

### 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
```

### 3.2 Confirm IQRF Gateway Daemon is running

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

**systemctl status iqrfd-daemon.service**

```
ubinux@ubinux4:~$ systemctl status iqrfd-daemon.service
● iqrfd-daemon.service - IQRF daemon iqrfd_startup
   Loaded: loaded (/lib/systemd/system/iqrfd-daemon.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2017-12-12 18:23:37 UTC; 16s ago
     Main PID: 13048 (iqrfd_startup)
        Tasks: 11 (limit: 4915)
      CGroup: /system.slice/iqrfd-daemon.service
              └─13048 /usr/bin/iqrfd_startup /etc/iqrfd-daemon/config.json
```

## 4 IQRF Gateway Daemon WebApp

### 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/ubinux**

**git clone https://github.com/iqrfsdk/iqrfd-daemon-webapp.git**

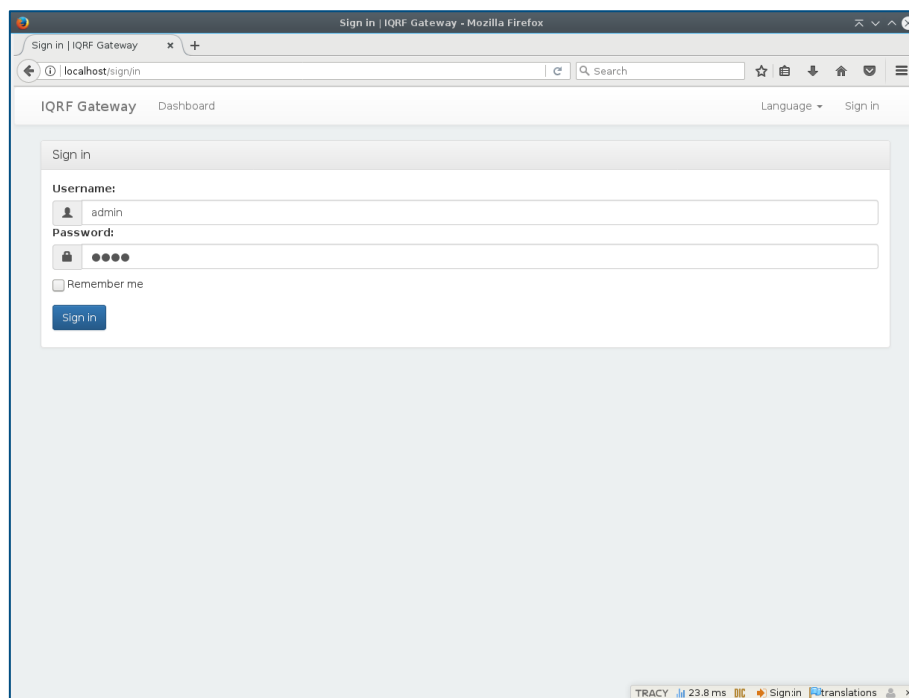
**cd iqrfd-daemon-webapp/install/**

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

### 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 **iqrfd**.

**http://localhost/en**

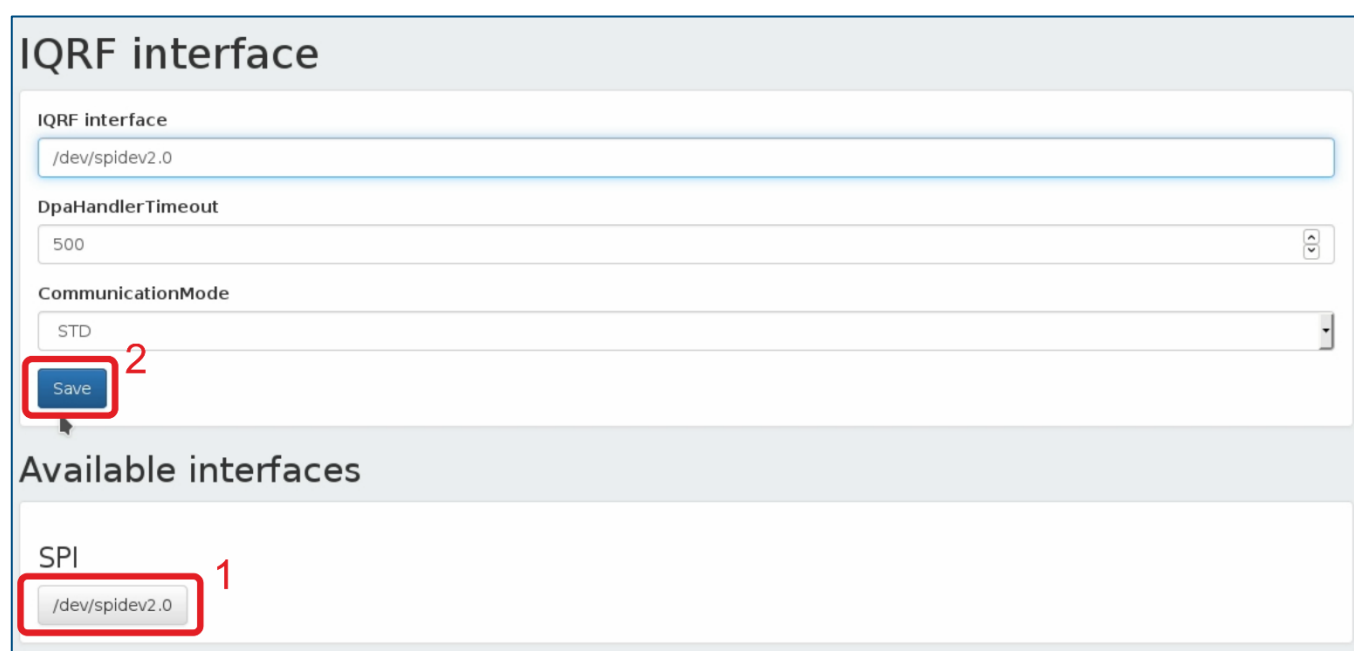
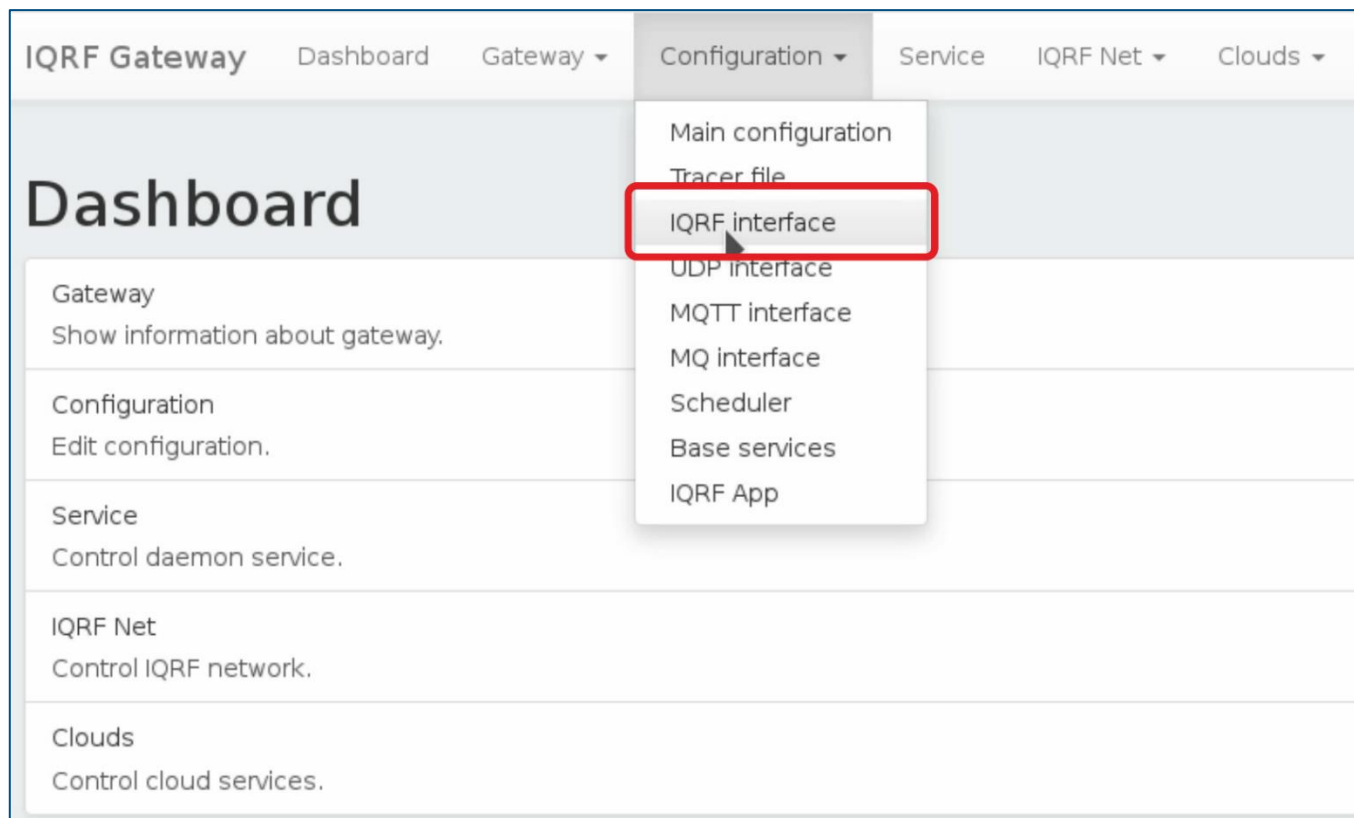


## 5 SPI interface

### 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>

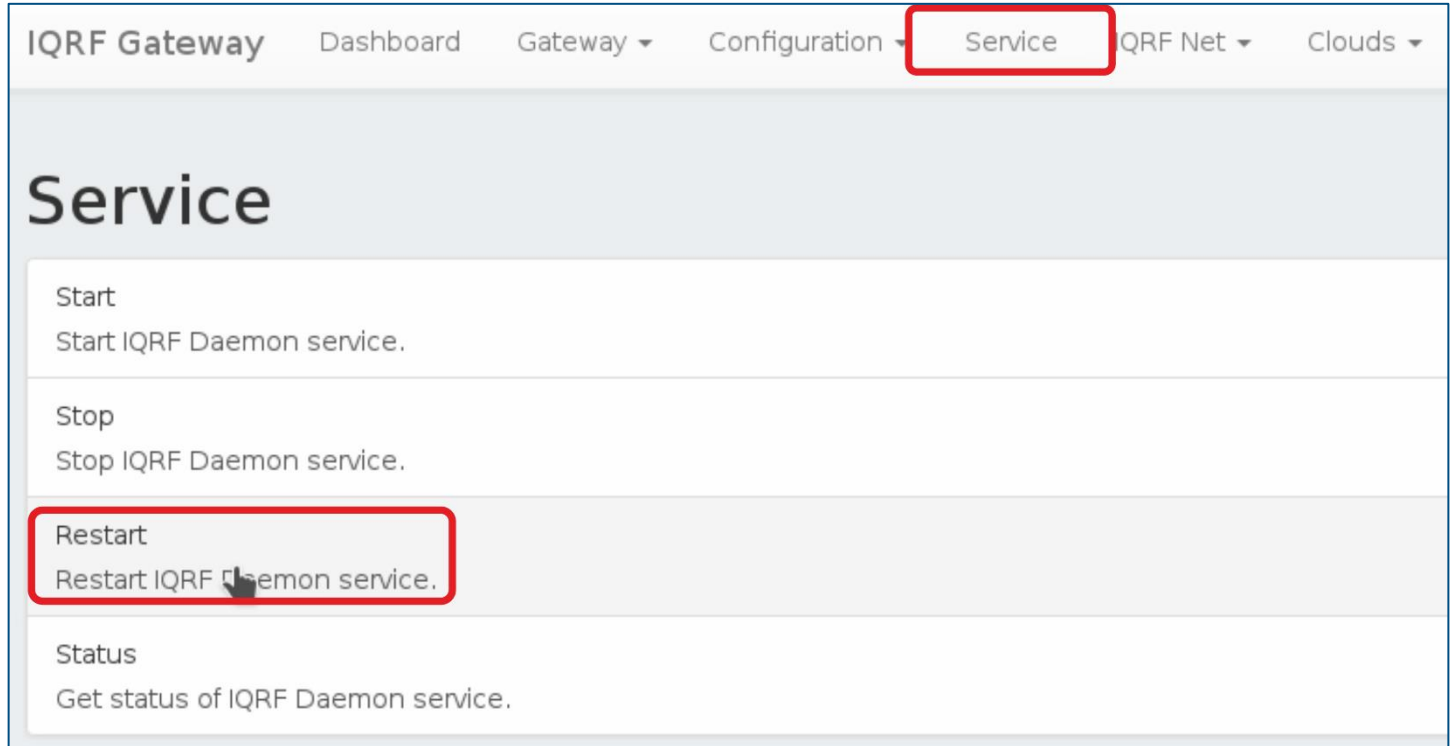




## 5.2 Restart IQRF Gateway Daemon

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>



## 6 Node.js

### 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/ubinux
```

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

```
cd iot-starter-kit/install
```

Enter ubinux as a password

```
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-ubinux /etc/lsb-release
```

## 7 Node-RED

### 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
```

### 7.2 Start Node-RED

Run Node-RED with these two commands.

```
cd /home/ubinux
```

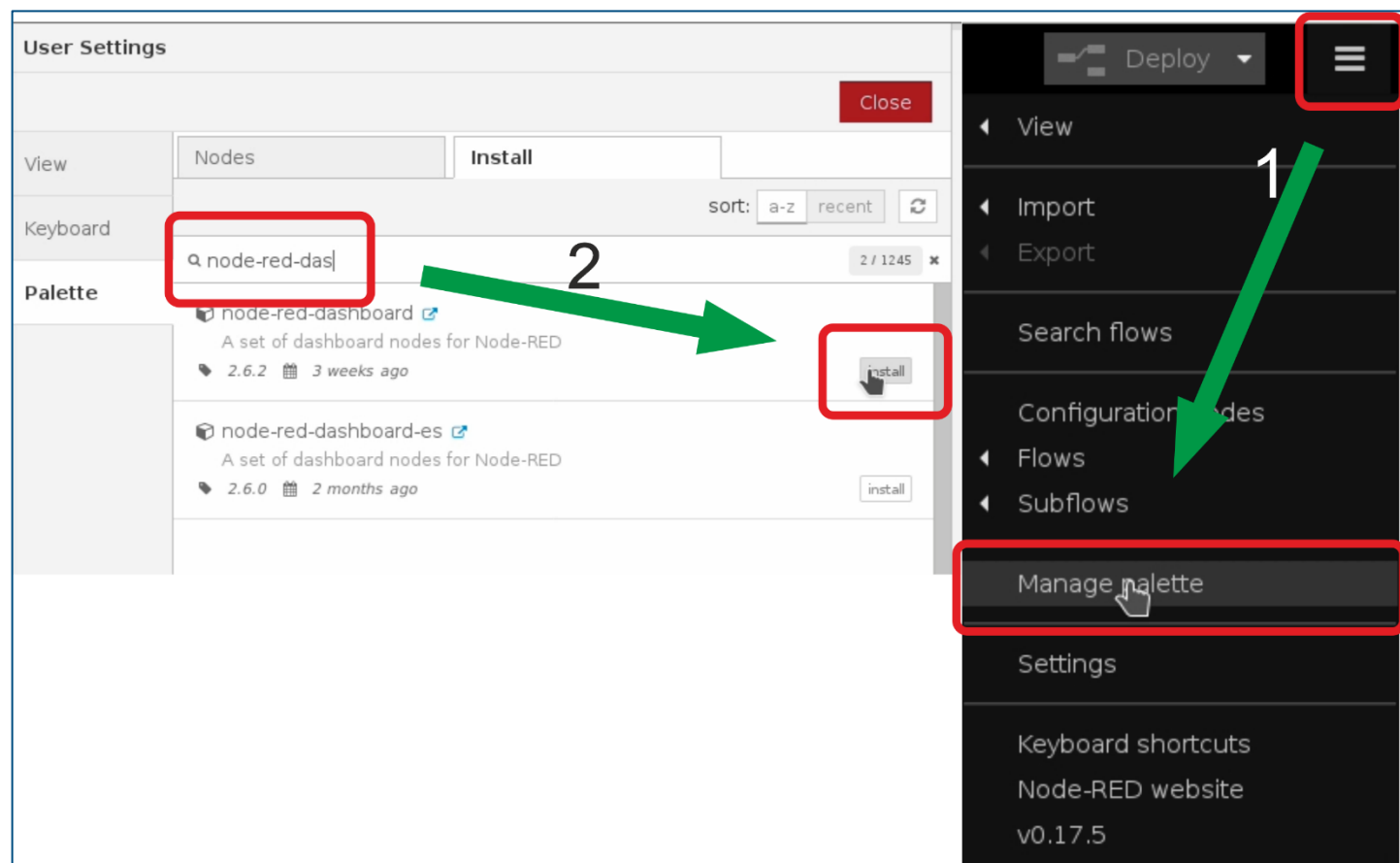
```
pm2 start /usr/bin/node-red
```

### 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>



### 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/ubinux/iot-starter-kit/install
```

```
cp up-board/node-red/* /home/ubinux/.node-red
```

```
pm2 restart node-red
```

### 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.

```
pm2 save
```

```
pm2 startup
```

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

### 7.6 Confirm Node-RED is running

Verify that Node-RED is running.

```
systemctl status pm2-ubinux
```

```
ubinux@ubinux4:~/iot-starter-kit/install$ systemctl status pm2-ubinux
● pm2-ubinux.service - PM2 process manager
   Loaded: loaded (/etc/systemd/system/pm2-ubinux.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2017-12-12 18:36:38 UTC; 12s ago
     Docs: https://pm2.keymetrics.io/
  Main PID: 25184 (PM2 v2.8.0: God)
    CGroup: /system.slice/pm2-ubinux.service
            └─ 25184 PM2 v2.8.0: God Daemon (/home/ubinux/.pm2)
```

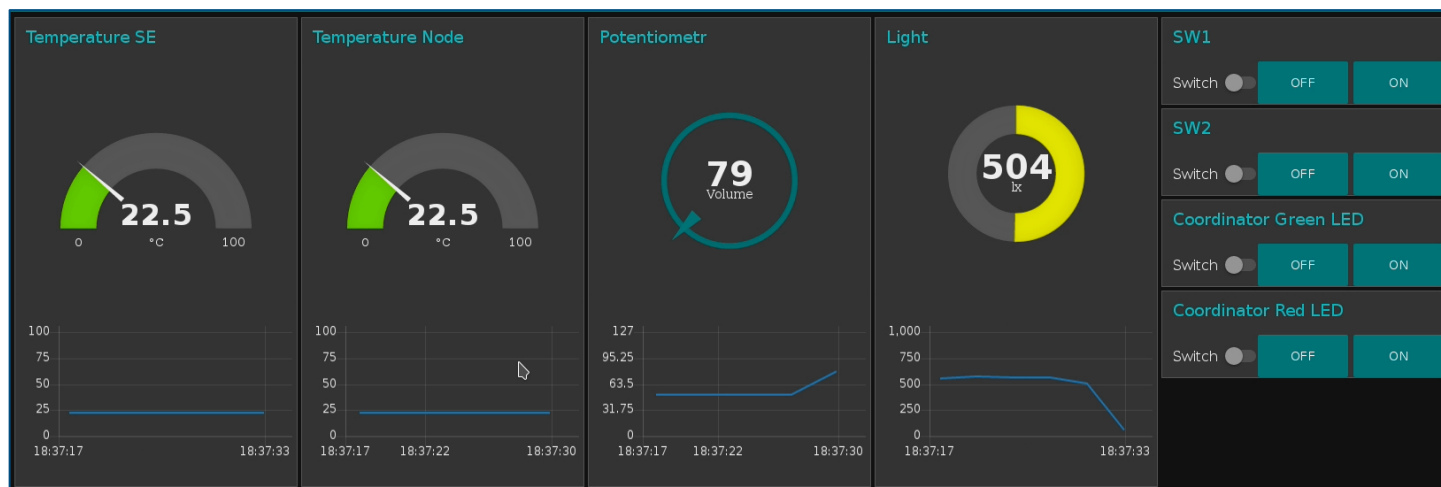
### 7.7 Check Node-RED dashboard

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 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.

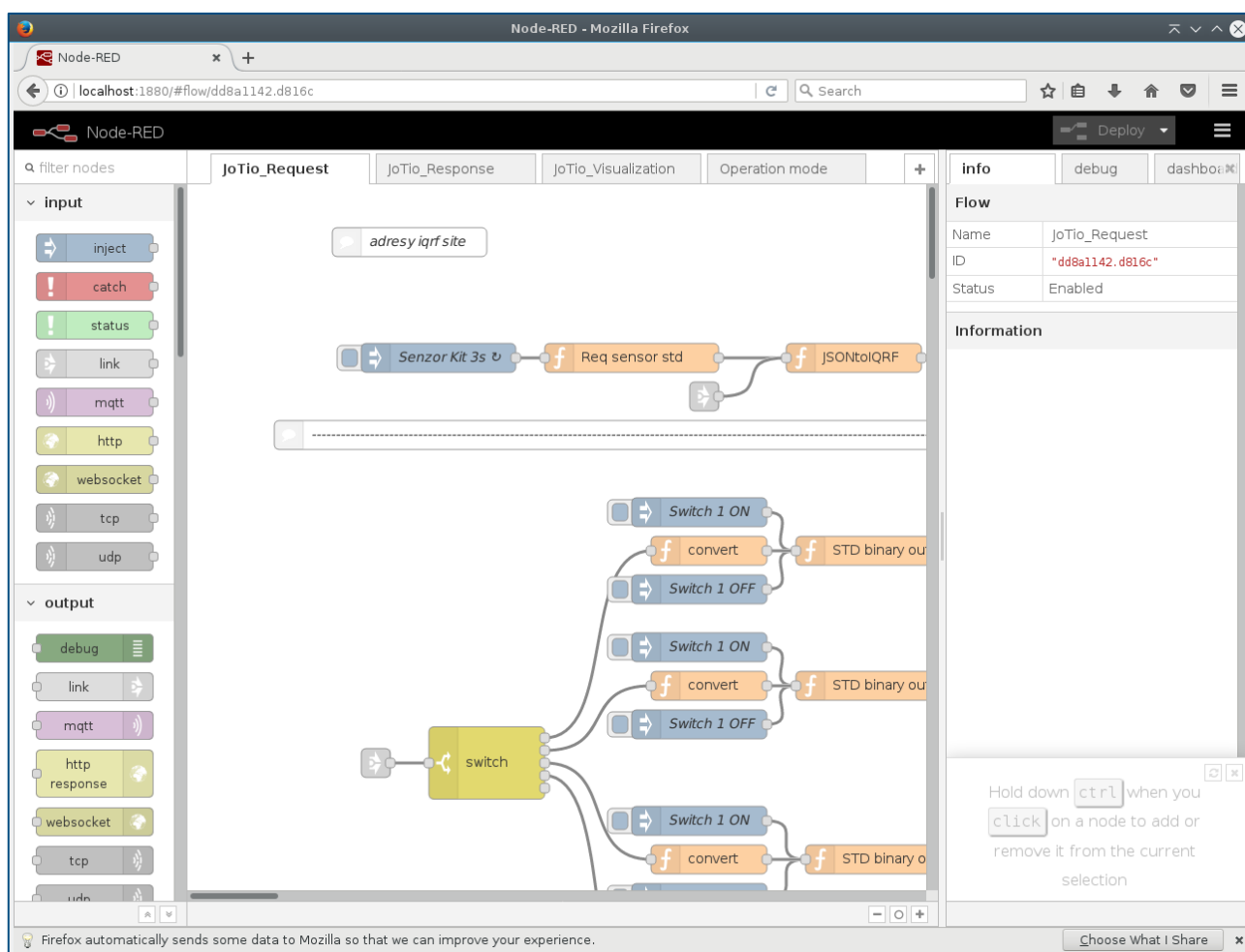
<http://localhost:1880/ui>



### 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>

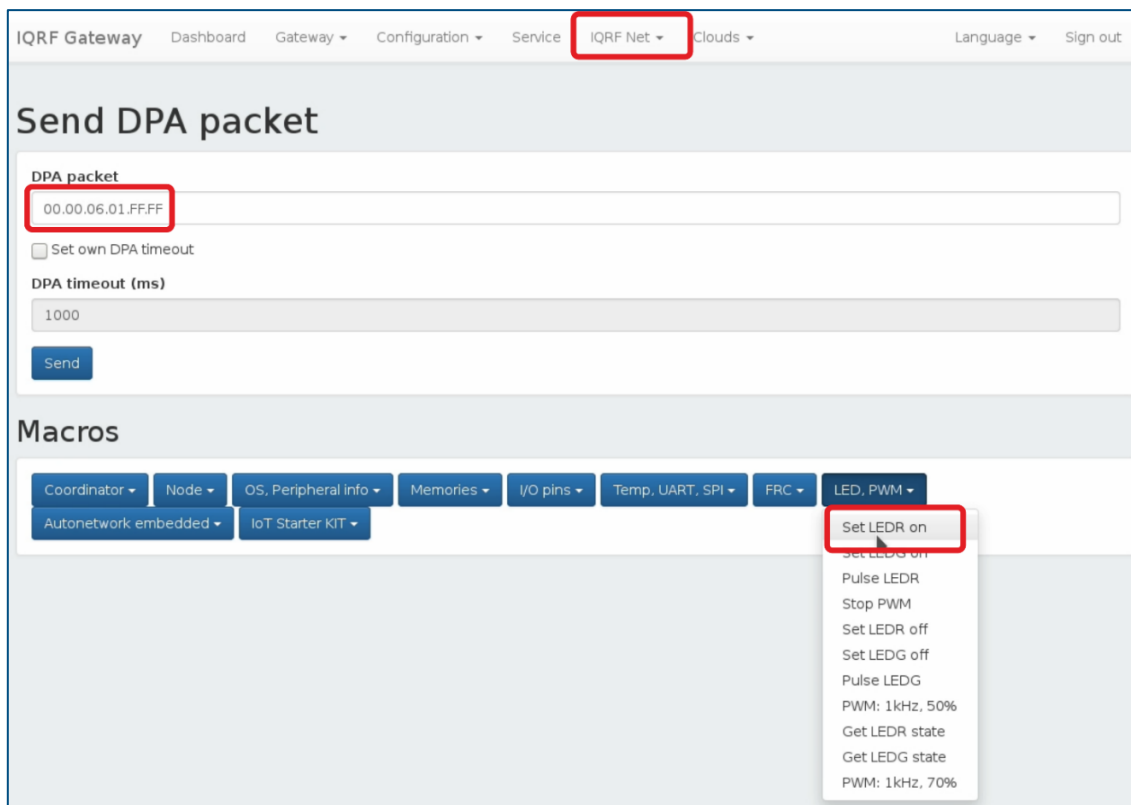


## 8 Test the functionality

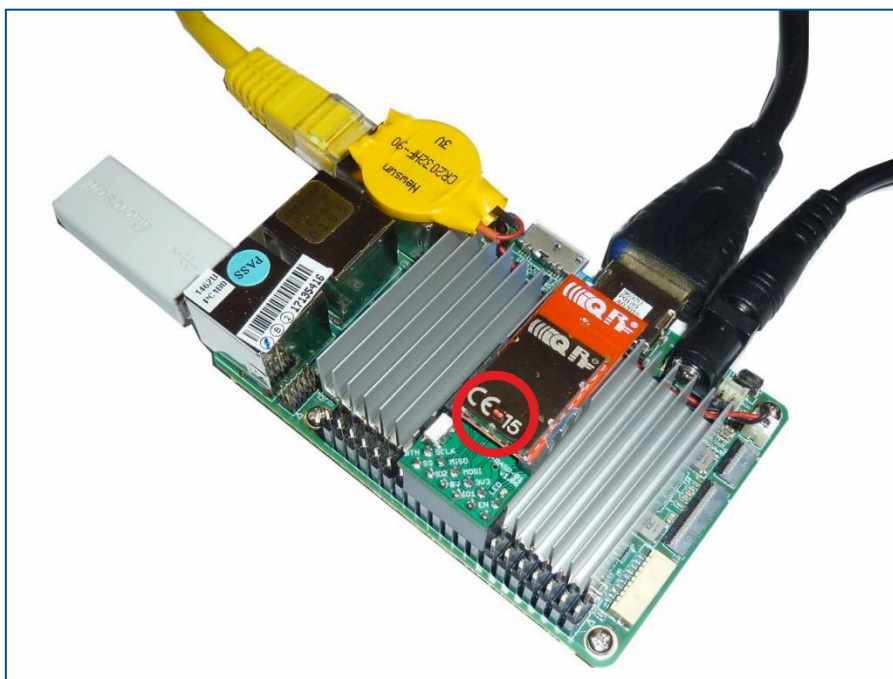
### 8.1 Send DPA Packet

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.



### 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.

```
ubinux@ubinux4:~/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 }
```

## 9 Check more examples

```
cd /home/ubinux
```

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

```
cd iqrf-daemon-examples
```

## 10 Summary

We made an IQRF Gateway from UP board. Be sure, you can control your IQRF network from the UP board – see the [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 it, it's the topic of the following part.