How to install the Raspberry Pi Gateway

A Gateway is a device which is used to receive data from a Node (in this example, it is an Arduino Node), the technology used is LoRaWAN. This Gateway is used for a project in the Østfold University College. This one gets data like temperature sensors and also light sensors. If you want to understand how we make this kind of device, you can follow this little guide.

First of all, to use this guide you must have these different components:

	Raspberry Pi 3 model B
	iC880-SPI Concentrator module
	Micro SD card 16 GB minimum
	USB Keyboard
	USB Mouse

	Screen
	HDMI wire
	Power supply (5V, 2A) micro USB
	Ethernet wire type RJ45
	Wires

The Gateway allows to gather data from the Arduino node in order to send it to the Ubuntu server where they'll be stocked.

Table of contents

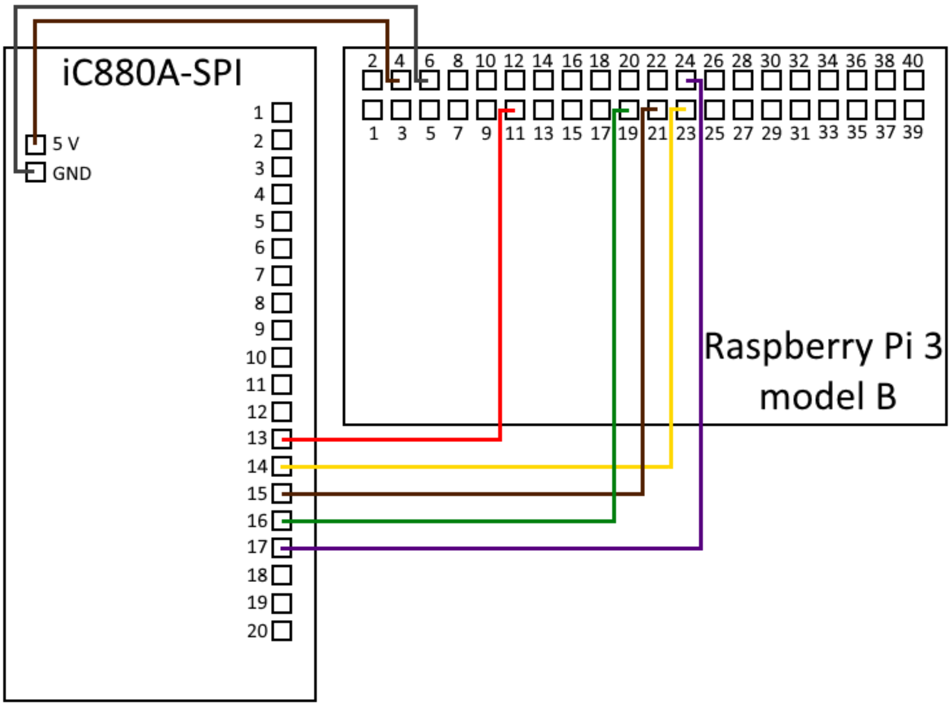
Step 1: Installing the Raspberry Pi

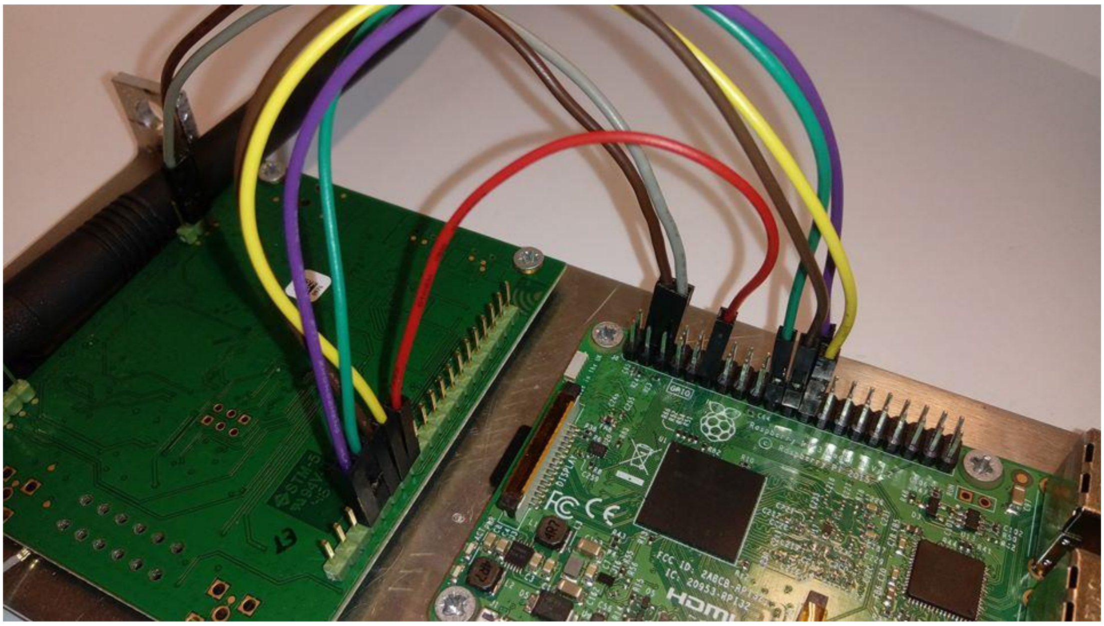
Configure the Raspberry Pi 3 model B by following the instructions using the video below:

Getting Started With The Raspberry Pi 3 by element 14

Step 2: Cabling

Now, you need to connect the iC880A-SPI to the Raspberry Pi 3 model B with the wires, just below, you can see how is cabled the module and the Raspberry Pi.





Step 3: Commands for configuration

Open the terminal window and close it just when it is written “Close the terminal window“ because it is important not to close it.
Now type these commands:

Little tip: If you want to copy these commands you can just select the commands with your cursor and do Ctrl + C and after if you want to paste them in the terminal window, you should press Ctrl + Shift + V.

You should download the folder called LoRa, extract it on the Raspberry Pi desktop and finally type the following commands in a terminal window:

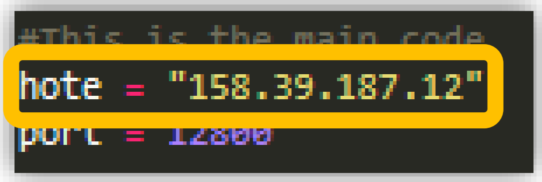
```
cd Desktop

mv LoRa /home/pi
```

In order to transmit data from the Gateway to the Server, you now need to indicate its IP address. To obtain it you need to write this command on the server’s terminal window and after you can close the terminal window:

```
hostname -I
```

Once you have obtained it, go in *LoRa/lora_gateway/lora_gateway/util_pkt_logger* and open the file called client.py and write the server’s IP address at the *hote* line, between the double quotes (it is at the end of the code):



Here

```
#This is the main code
hote = "158.39.187.12"
port = 12800
```

Now, you can save the file by pressing Ctrl + S and close it.

Step 5: Compile the codes

In order to use our programs, open a terminal window and type these commands:

Type that to go in the *LoRa* repertory :

```
cd LoRa/lora_gateway/lora_gateway
```

To compile our programs and make them functional, enter this on the terminal window:

```
make
```

You can now close the terminal window.

Step 6: Run the programs

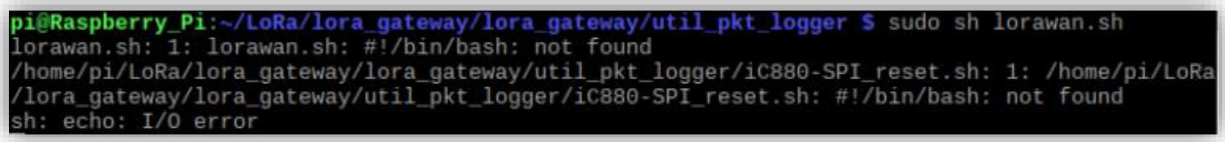
Open a terminal window and type the first command, it permits to locate the file we will use to run our program.

```
cd LoRa/lora_gateway/lora_gateway/util_pkt_logger
```

The second one executes the program to reset the pins of the Raspberry Pi:

```
sudo sh iC880-SPI_reset.sh
```

You should obtain this result:

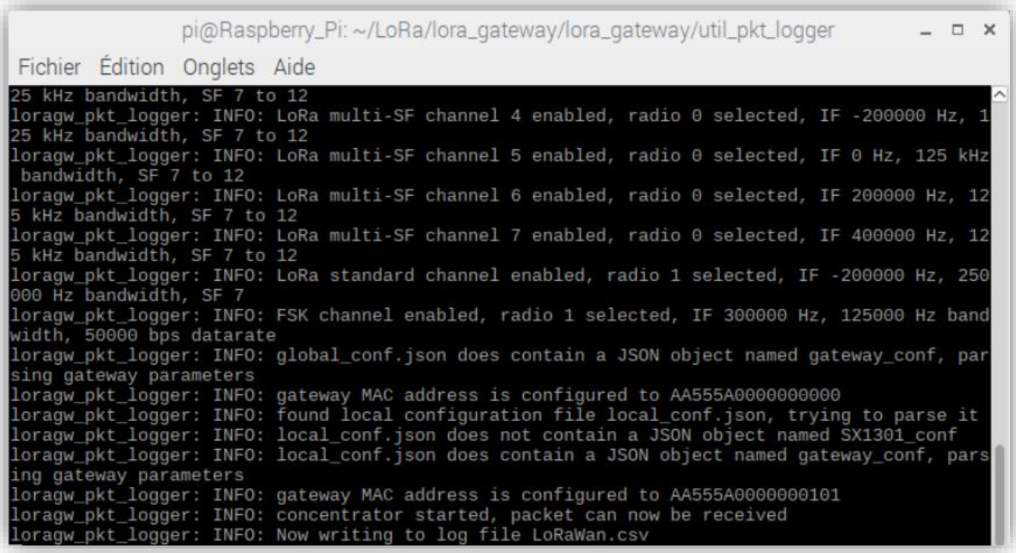


```
pi@Raspberry_Pi:~/LoRa/lora_gateway/lora_gateway/util_pkt_logger $ sudo sh lorawan.sh
lorawan.sh: 1: lorawan.sh: #!/bin/bash: not found
/home/pi/LoRa/lora_gateway/lora_gateway/util_pkt_logger/iC880-SPI_reset.sh: 1: /home/pi/LoRa
/lora_gateway/lora_gateway/util_pkt_logger/iC880-SPI_reset.sh: #!/bin/bash: not found
sh: echo: I/O error
```

Next, this command allows to get the data from the Arduino Uno Rev 3, and to stock them in the Raspberry SD card:

```
./util_pkt_logger
```

And after this you should obtain this result:



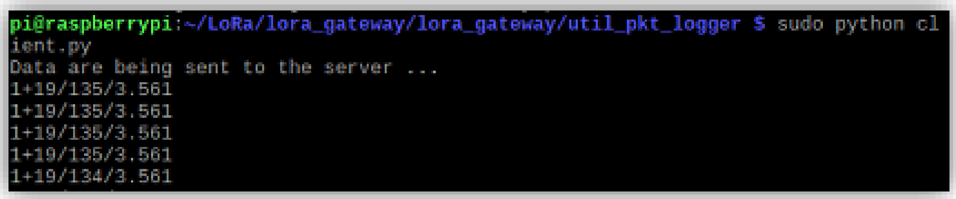
pi@Raspberry_Pi: ~/LoRa/lora_gateway/lora_gateway/util_pkt_logger
Fichier
Édition
Onglets
Aide

```
25 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 4 enabled, radio 0 selected, IF -200000 Hz, 1
25 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 5 enabled, radio 0 selected, IF 0 Hz, 125 kHz
 bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 6 enabled, radio 0 selected, IF 200000 Hz, 12
5 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 7 enabled, radio 0 selected, IF 400000 Hz, 12
5 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa standard channel enabled, radio 1 selected, IF -200000 Hz, 250
000 Hz bandwidth, SF 7
loragw_pkt_logger: INFO: FSK channel enabled, radio 1 selected, IF 300000 Hz, 125000 Hz band
width, 50000 bps datarate
loragw_pkt_logger: INFO: global_conf.json does contain a JSON object named gateway_conf, par
sing gateway parameters
loragw_pkt_logger: INFO: gateway MAC address is configured to AA555A0000000000
loragw_pkt_logger: INFO: found local configuration file local_conf.json, trying to parse it
loragw_pkt_logger: INFO: local_conf.json does not contain a JSON object named SX1301_conf
loragw_pkt_logger: INFO: local_conf.json does contain a JSON object named gateway_conf, pars
ing gateway parameters
loragw_pkt_logger: INFO: gateway MAC address is configured to AA555A0000000101
loragw_pkt_logger: INFO: concentrator started, packet can now be received
loragw_pkt_logger: INFO: Now writing to log file LoRaWan.csv
```

Open another terminal window to write this command. The command will execute a program which sends all the data to the server:

```
sudo python client.py
```

Finally, you should obtain this result:



```
pi@raspberrypi:~/LoRa/lora_gateway/lora_gateway/util_pkt_logger $ sudo python cl
ient.py
Data are being sent to the server ...
1+19/135/3.561
1+19/135/3.561
1+19/135/3.561
1+19/135/3.561
1+19/134/3.561
```

This tutorial is finished. You should now go read the tutorial about How to install the Apache2 Server and the Website.

Thank you for reading.

Aurora project members