LoRaWAN Tutorial

# Overview

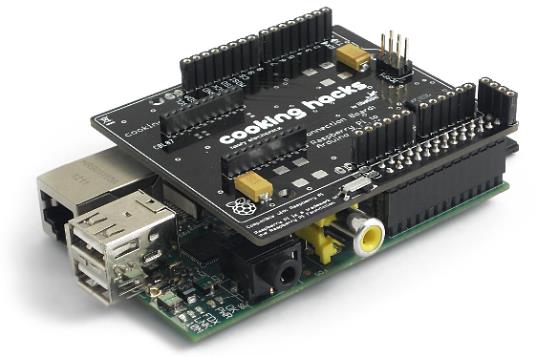
In this tutorial, we will interconnect an Arduino board with a Raspberry Pi one throw LoRaWAN modules. Here is an illustration of the infrastructure we will implement.

In the first step of this tutorial, we will configure a Raspberry Pi as a gateway to receive messages from multiple things.

# Raspberry Pi as a gateway

## Hardware

To add LoRaWAN functionality to Raspberry Pi 2 or 3, we use the following hardware: “[Raspberry Pi to Arduino Shields Connection Bridge](https://www.cooking-hacks.com/raspberry-pi-to-arduino-shield-connection-bridge)” and “[LoRaWAN modules](https://www.cooking-hacks.com/lorawan-module-for-arduino-raspberry-pi-868-mhz-xbee-socket)”). You can find a [tutorial made by Cooking Hacks](https://www.cooking-hacks.com/lorawan-module-for-arduino-raspberry-pi-868-mhz-xbee-socket) who sells the Shield bridge module. You can find 433MHz and 900MHz (for US) and 868MHz (for EU) modules. So, we a will use the last ones. You can also read the [datasheet of the RN2483A module](https://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf) (or the one dedicated to [RN2003 module](http://ww1.microchip.com/downloads/en/DeviceDoc/40001811A.pdf) which is fully compatible). Normally, you won’t need to apply all the Cooking Hacks tutorial, as we won’t write our own programs in C++, but use the Node-Red configuration.

## Software configurations

Read and apply all the necessary commands descripted in the “Raspberry Configuration Tutorial: Software configuration of the Shield”.

## Test communication with LoRaWAN module

To test if the LoRaWAN module is functional, you can use the minicom program to dialog with the XBee serial interface.

sudo apt-get install minicom

You can create a minicom profile with the right parameters to dialog the XBee module

sudo nano /etc/minicom/minirc.lorawan

Add the following lines to this file:

pu port /dev/ttyAMA0

pu baudrate 57600

pu bits 8

pu parity N

pu stopbits 1

pu rtscts No

pu localecho Yes

To communication with the LoRaWAN module, type the following command:

minicom -o lorawan

You can now use the commands described in the product datasheet to communicate with the LoRaWAN module. You don’t need to enter a specific command to configure the module. You just must type your command with parameters, press Return key (Carriage Return) and then press Ctrl-J (Line Feed). You will receive a response by the module.

For instance, type:

sys get ver

then type on the Return key, then Ctrl-J (to send [CR/LF](https://fr.wikipedia.org/wiki/Carriage_Return_Line_Feed)).

You should receive, as a response, the information on hardware platform, firmware version, release date.

RN2483 1.0.3 Mar 22 2017 06:00:42

Response: RN2483 X.Y.Z MMM DD YYYY HH:MM:SS, where X.Y.Z is firmware version, MMM is month, DD is day, HH:MM:SS is hour, minutes, seconds (format: [HW] [FW] [Date] [Time]). [Date] and [Time] refer to the release of the firmware.

To receive message with a LoRaWAN module, you must configure it to the right mode, frequency, power, … Here are the commands to type to configure you LoRaWAN module for P2P communication.

mac pause

radio set pwr 15

radio set freq 868100000

radio set sf sf12

radio set cr 4/5

radio set bw 125

radio set crc on

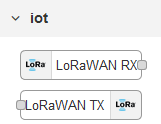
Then to receive a radio message, type:

radio rx 0

## Node-RED installation and configuration

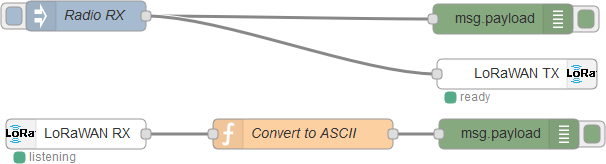
For the Node-Red installation on Raspberry Pi, you must read and apply commands descripted in the “Raspberry Configuration Tutorial / Node-RED installation”.

There was no node module to deal with RN2483 and RN2903 modules. So, I had to write code to deals with these modules.

Install node-red-contrib-lorawan module to communicate with LoRaWAN protocol, using the RN2483 or RN2903 radio modules. This will enable 2 new types of node: LoRaWAN RX and LoRaWAN TX respectively “Receive” and “Transmit” data throw LoRaWAN radio module (and the underlying serial socket).

You must then configure the LoRaWAN serial connexion. You must set all the needed parameters to configure the P2P LoRaWAN communication.

Here is a basic example and the corresponding flow. Don’t forget to configure the LoRaWAN node with you own parameters.



Here is the code of this flow.

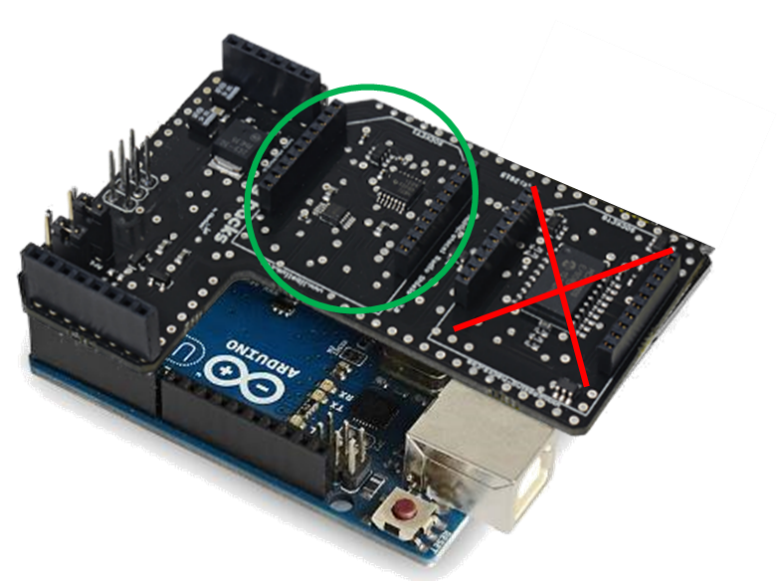
[{"id":"4df971fb.917258","type":"debug","z":"6413de59.77d4a8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":555,"y":57,"wires":[]},{"id":"468654e5.1989e4","type":"inject","z":"6413de59.77d4a8","name":"Radio RX","topic":"","payload":"radio rx 0","payloadType":"str","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":138,"y":54,"wires":[["4df971fb.917258","ccc9b28.9fea2d"]]},{"id":"67e30445.08bdec","type":"debug","z":"6413de59.77d4a8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":561,"y":172,"wires":[]},{"id":"ccc9b28.9fea2d","type":"lorawan-tx","z":"6413de59.77d4a8","name":"","loraWAN":"bc9b094a.2501d","x":569,"y":112,"wires":[]},{"id":"8781b419.e487e","type":"lorawan-rx","z":"6413de59.77d4a8","name":"","loraWAN":"bc9b094a.2501d","x":128,"y":172,"wires":[["1c916ea1.4ea971"]]},{"id":"1c916ea1.4ea971","type":"function","z":"6413de59.77d4a8","name":"Convert to ASCII","func":"msg.payload = msg.payload.toString('ascii');\nreturn msg;","outputs":1,"noerr":0,"x":355,"y":172,"wires":[["67e30445.08bdec"]]},{"id":"bc9b094a.2501d","type":"lorawan-config","z":"","rawFrames":false,"mode":"2","modeP2P\_radioPwr":"15","modeP2P\_radioFreq":"868100000","modeP2P\_radioSf":"sf12","modeP2P\_radioCr":"4/5","modeP2P\_radioBw":"125","modeP2P\_radioCRC":"on","serialPort":"/dev/ttyAMA0","lock":true,"baudRate":"57600","dataBits":"8","stopBits":"1","parity":"none"}]

# Arduino as a Thing

## Hardware

Now, we will configure an Arduino to communicate throw an LoRaWAN module. We use the following hardware: “[Multiprotocol Radio Shield for Arduino](https://www.cooking-hacks.com/multiprotocol-radio-shield-board-for-arduino-rpi-intel-galileo)” and of course a LoRaWAN module. You can find a [tutorial made by Cooking Hacks](https://www.cooking-hacks.com/documentation/tutorials/multiprotocol-shield-connect-two-xbee-connector-arduino-raspberry-pi-galileo/) who sells the multiprotocol shield for Arduino.

Connect the XBee module to **Socket 1** or the multiprotocol radio shield for Arduino.



## Software configurations

First, you need to download the Arduino IDE for your operating system. We won’t detail this in this tutorial. Then you must [download the library to manage the multiprotocol shield](http://www.cooking-hacks.com/media/cooking/images/documentation/multiprotocol_radio_shield/multiprotocol_shield_library.zip). Copy the two folders MCP23008 and Multiprotocol\_Shield in your Arduino libraries folder (on Windows, it’s in Documents\Arduino\libraries).

Download and install the [LoRaWAN library for Arduino](http://www.cooking-hacks.com/media/cooking/images/documentation/tutorial_kit_lorawan/LoRaWAN_library_arduino_v1_3.zip) in the Arduino IDE libraries folder.

Your Arduino IDE is no ready to compile a program and send it to the Arduino board.

## Program

With the installed library, you have some samples. To test communication with the Raspberry Pi, you can load the arduinoLoRaWAN/LoRaWAN\_P2P\_02\_send\_packets, going to the menu File / Examples.

This program will send the message “0102030405060708090A0B0C0D0E0F” to the LoRaWAN module if correctly configured.

In your Node-RED debug console, you should read this message sent appearing time to time.

# Conclusion

With this tutorial, you learned how to make LoRaWAN communications from Arduino (with your own program) to a Raspberry Pi (using node-RED). We hope you enjoyed!