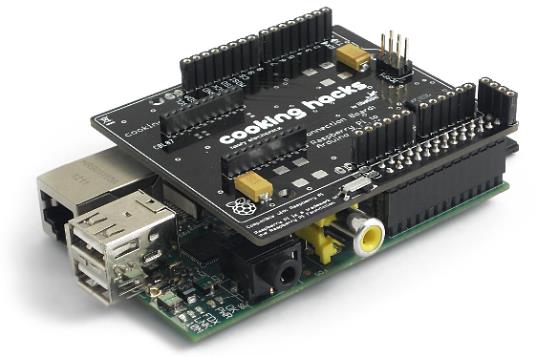
Raspberry Configuration Tutorial

# Raspberry Pi as a multi-protocols gateway

## Hardware

To add communication functionalities 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 different communication modules like XBee, LoRa, LoRaWAN, … You can find a [tutorial made by Cooking Hacks](https://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-to-arduino-shields-connection-bridge/) who sells the Shield bridge module. Normally, you won’t need to apply all the Cooking Hacks tutorial, as the aim is not to write C++ programs, but use Node-Red to manage functionalities implementation.



## Software configuration of the Shield

First, we must enable interfaces for the shield to communicate with the Raspberry Pi. You must edit the /boot/config.txt file.

sudo /boot/config.txt

You must add the following lines in the /boot/config.txt file to activate the UART, SPI And I2C interfaces:

#enable uart interface

enable\_uart=1

#enable spi interface

dtparam=spi=on

#enable i2c interface

dtparam=i2c\_arm=on

and if you have a Raspberry Pi also the following lines:

#map mini-UART to internal bluetooth an free-up main UART to handle module

dtoverlay=pi3-miniuart-bt

This will remap the Bluetooth (integrated in the Raspberry Pi 3) to the mini-UART interface (less efficient) to reserve the main UART interface for the XBee socket module.

If you a Raspberry Pi 3, you also must modify the /boot/cmdline.txt to disable the console on the UART interface and map it to tty1. So,

console=ttyAMA0,115200 must be replaced by console=tty1.

Then, you will have to upgrade to the last update release of all packages, then reboot the system.

sudo apt-get update

sudo apt-get upgrade

sudo reboot

Finally, to activate the socket on the shield, so to power the XBee module on, you must type the following commands:

sudo apt-get install wiringpi

gpio mode 4 OUT

gpio write 4 1

gpio readall

You can verify than GPIO 4 in configured as 1 and OUT.

## Node-RED installation

To install Node-RED, the recommendation on the website is not to use the Debian package but instead, a specific Shell script. So, as recommended on the [Node-RED website for Raspberry Pi](https://nodered.org/docs/hardware/raspberrypi), you just must type the following command to install Node-RED 10.13 (LTS) and the right npm release:

bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)

For Node-RED to automatically start when you power on the Raspberry Pi, you must activate Node-RED as a service with the following command:

sudo systemctl enable nodered.service

Then, to stop or start the service, you just must type the following commands:

node-red-stop

node-red-start

Launch a web browser and connect to the Raspberry by with its IP address (to the Raspberry IP address, use the ifconfig command). If you host is properly configured, you should be able to use the name of the Raspberry Pi instead of its IP address (raspberrypi.local).

http://IP:1880/

# Tests

If you want to quickly make some tests with the shield and different modules, you can download the following repository:

git clone https://github.com/lavirott/arduPi.git

To compile a program, go to the examples directory in arduPi and to a specific protocol and type

./cook.sh sample.cpp

It will compile all the needed libraries and then the program. To test it, just type

sudo ./sample.cpp\_exe

# Conclusion

Do not forget that if you activate Node-RED as a service, if you restart your Raspberry Pi and saved a flow, this flow will be instantiated and activated.