Wi-Fi Tutorial

# Raspberry Pi as a Wi-Fi gateway (Access-Point)

## Hardware

Raspberry Pi 3 as an integrated Wi-Fi chip. So, it’s not necessary to add a Wi-Fi dongle to you Raspberry Pi. Wi-Fi will be used to provide the access for Arduino things and the wired Ethernet socket will be used to connect you Raspberry Pi to the Internet.

## Software

To turn your Raspberry Pi 3 into a Wi-Fi Access Point, you must install the following script and execute it.

git clone https://github.com/lavirott/wifi-ap-pi3.git

cd wifi-ap-pi3

chmod a+x install.sh

sudo ./install.sh

for more information about that script, you can read <https://github.com/lavirott/wifi-ap-pi3>.

## Node-RED configuration

All needed nodes are standard ones, so you don’t have to install anything.

Here is a basic example and the corresponding flow. Don’t forget to modify the IP address to correspond to your Arduino one.



[{"id":"5dad9352.5e1a1c","type":"tcp in","z":"6413de59.77d4a8","name":"","server":"client","host":"192.168.0.36","port":"1000","datamode":"stream","datatype":"utf8","newline":"\\n","topic":"","base64":false,"x":133,"y":57,"wires":[["7902051d.94cfb4"]]},{"id":"7902051d.94cfb4","type":"debug","z":"6413de59.77d4a8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":393,"y":57,"wires":[]}]

# Configure and Test with a Wi-Fi Bee module connected to your PC

## Hardware

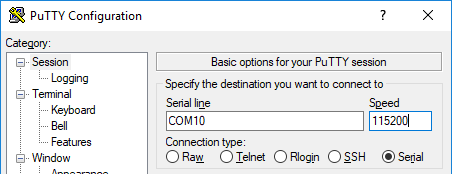
Before configuring the Arduino board with its own XBee socket, you can perform tests with a XBee module connected to your PC with a XBee USB Board. It could be the following platforms.



For this tutorial, we use the [Wi-Fi Bee module MT7681](https://www.gotronic.fr/art-module-wifi-bee-tel0107-26076.htm). This module allows to add a Wi-Fi interface to Arduino projects. You can connect it to a XBee socket and communicate via the UART interface. You can configure it using AT commands.



## Software

First, to test your Wi-Fi module and try to connect it to you Raspberry, you need to download and install [putty](https://www.putty.org/), if you don’t have it already installed on your PC. Configure your connection with the following parameters:

Connection type: Serial

Serial line: COM10 (put your own serial ports your device is connected to)

Speed: 115200

Then click on the “Open” button.

You now must configure your Wi-Fi Bee module with the following commands to indicate the SSID name your device has to connect to and the password to use. Here are the names use for the Raspberry Pi Access Point we just configured.

+++

AT+SSID=RaspberryPi

AT+Password=raspberry

AT+CONNECT

As responses, you should have the following log trace of this communication.

OK

OK

OK

==> Recovery Mode

<== RecoveryMode

(-)

df\_callback\_reg()

SM=0, Sub=0

SM=2, Sub=0

SM=3, Sub=0

Auth with:ssid = RaspberryPi, auth mode = 7,

SM=3, Sub=1

SM=4, Sub=0

SM=4, Sub=1

SM=5, Sub=0

SM=6, Sub=0

fd 0 uip\_aborted.9999. 0 128 0

Got IP:192.168.0.36

DNS wiki.dfrobot.com.cn, 182.254.133.61

Here, you can find all the [AT commands for Wi-Fi Bee module MT7681](https://github.com/DFRobot/WiFiBee-MT7681/blob/master/AT%20Command%20Manual/EN/WiFiBee-MT7681%20AT%20Command%20V1.1.pdf).

When your Wi-Fi module is configured to connect to a specific access point, it will reconnect to it automatically at each reboot of your module unless you modify this configuration.

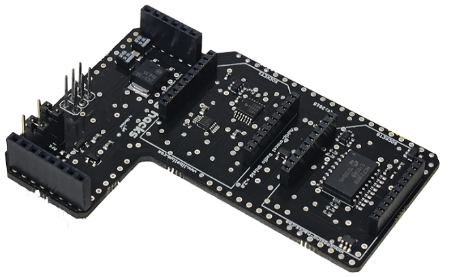
# Arduino as a Thing

You can configure the Arduino as a Wi-Fi client or as an Access-Point. In this tutorial, we will configure it as a client to connect to the Raspberry Pi access point.

## Hardware

To provide Wi-Fi capability to your Arduino for we use the Wi-Fi Bee MT7681. To connect it to Arduino board, we use “[Multiprotocol Radio Shield for Arduino](https://www.cooking-hacks.com/multiprotocol-radio-shield-board-for-arduino-rpi-intel-galileo)”. You can connect the Wi-Fi Bee module to **Socket 0** of 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).

## Program

Here is a simple example to connect to your Raspberry Pi access point configured just before.

#include <multiprotocolShield.h>

#include <MCP23008.h>

#include <Wire.h>

const char ssid[]={"AT+SSID=RaspberryPi"}; // WiFi SSID

const char passwd[]={"AT+PASSWORD=raspberry"}; // WiFi password

static char ip\_addr[15];

void setup()

{

socket0.ON();

delay(100);

socket0.setMUX();

wifi\_connect();

Serial.println(wifi\_wait\_ip());

}

void wifi\_connect()

{

Serial.begin(115200);

delay(100);

while(!Serial);

Serial.print("+++");

delay(1000);

Serial.println(ssid);

delay(100);

Serial.println(passwd);

delay(100);

Serial.println("AT+CONNECT");

delay(500);

}

char \* wifi\_wait\_ip()

{

String gotip = "Got IP:";

String line;

if (ip\_addr[0] == '\0') {

Serial.println("Debug: Wait init");

while(true) { // wait until got ip

line = Serial.readString();

if line.length

Serial.println(line);

if (gotip.equals(line.substring(0,7)))

break;

}

int i = 7;

while (line[i] != '\0') {

ip\_addr[i-7] = line[i];

i++;

}

Serial.println("Debug: End init");

return ip\_addr;

}

}

void loop()

{

Serial.println("Hi! I'm Arduino.");

delay(1000);

}