

One way to setup Raspberry Pi 2. Links to guides that I used.

Content

1. Goal
2. Setup Guide
3. Resistor colors and values
4. General tips
5. Useful commands

1. Goal

I try to use Raspberry Pi and Funduino UNO with Node.js to create setup which measures temperature, and sends messages (when temperature gets too high or low) over Internet to clients.

2. Setup Guide:

Step 1:

Get Raspberry Pi 2

<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

Step 2:

Setup NOOBS to SD-card and install Raspbian

<https://www.raspberrypi.org/downloads/noobs/>

Step 3:

Setup Raspberry Pi with monitor, keyboard, mouse

Raspberry Pi Software Guide

<https://www.raspberrypi.org/learning/software-guide/>

Step 4:

How to configure Raspbian (how to setup Wi-fi, which is not included in Raspberry Pi 2, only 3)

How to Install updates to Raspbian. **(this can take a lot of time!)**

How to setup Raspbian for Windows Remote Desktop Connection

How to create share folder and network mapping for it.

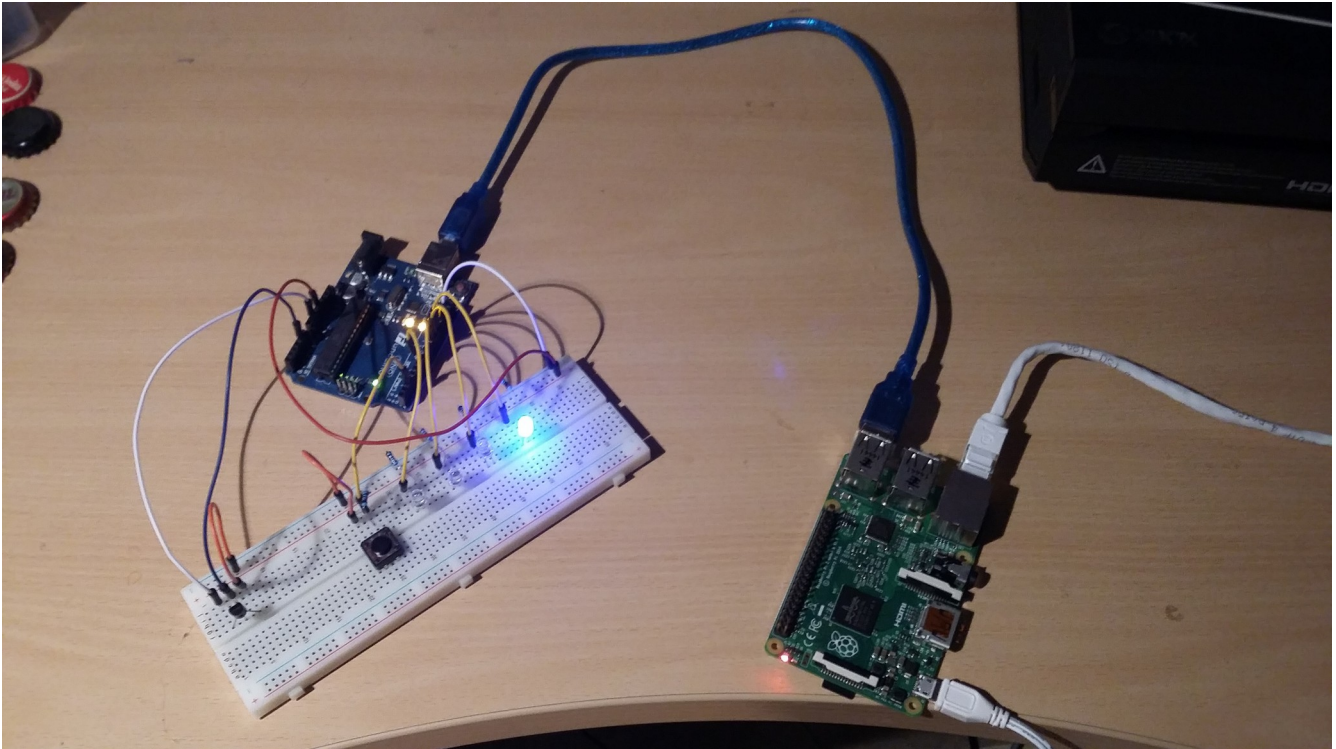
How to install Node.js

<http://thisdavej.com/beginners-guide-to-installing-node-js-on-a-raspberry-pi/>

Step 5: Arduino test with Python 2

I did own setup with Arduino. 4 leds, 1 button and 1 thermistor. Thermistor is for temperature measurement, and button and leds for led changing with button press.

Arduino setup:



Arduino code:

See appendix on last page.

Python 2 code:

```
import serial
ser = serial.Serial('/dev/ttyACM0', 9600)
while 1 :
    ser.readline()
```

Serial should write out current temperature in Celsius and Fahrenheit.

<https://oscarliang.com/connect-raspberry-pi-and-arduino-usb-cable/>

Step 6: Install MongoDB (and test Node.js as Service)

Installed MongoDB, but could not make it work as Service yet.

<http://yannickloriot.com/2016/04/install-mongodb-and-node-js-on-a-raspberry-pi/>

Step 7: Node-serialport test, and Johnny-Five

Tried node-serialport, Johnny-Five and Raspi IO

<https://github.com/nebrius/raspi-io/wiki/Getting-a-Raspberry-Pi-ready-for-NodeBots>

Upload StandardFirmata to Arduino with Arduino IDE.

Then Johnny-five handles the rest in node.js!

<https://github.com/rwaldron/johnny-five/wiki/Getting-Started#trouble-shooting>

More on Johnny-Five

<http://johnny-five.io/>

3. Resistor colors and values

I have three kinds of resistor, all banded with 5 colors. These are as reminder for myself.

Resistor 1:

Red – Red – Black – Black – Brown 220 Ohms 1 % tolerance

Resistor 2:

Brown – Black – Black – Brown – Brown 1k Ohms 1 % tolerance

Resistor 3:

Brown – Black – Black – Red – Brown 10k Ohms 1 % tolerance

<http://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-5-band>

4. General tips:

Remember to use “poweroff” before unplugging Raspberry Pi. Otherwise it may brick itself.

If you have to format SD-card in Windows, use SDFormatter, because only RECOVERY-partition may be visible.

https://www.sdcard.org/downloads/formatter_4/

Keep backup of whole SD-card after install, so all work is not lost. (use Win32 Disk Imager)

<http://sourceforge.net/projects/win32diskimager/files/latest/download>

<https://thepihut.com/blogs/raspberry-pi-tutorials/17789160-backing-up-and-restoring-your-raspberry-pis-sd-card>

5. Useful commands

<code>sudo apt update</code>	for updating raspberry pi
<code>sudo apt full-upgrade</code>	for updating raspberry pi
<code>sudo poweroff</code>	shut down
<code>sudo reboot</code>	restart
<code>ifconfig</code>	ip-address and other info
<code>ls /dev/tty*</code>	used to check which port Arduino is connected to
<code>npm install johnny-five</code>	install johnny-five node-package

Appendix:

```
char buffer[18];

int redLed = 6;
int yellowLed = 10;
int blueLed1 = 11;

int blueLed2 = 12;
int button = 2; // button pin
unsigned long changeTime; // time since button pressed

int val;
int tempPin = 1;

void setup() {
  Serial.begin(9600);
  Serial.flush();

  pinMode(redLed, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(blueLed1, OUTPUT);
  pinMode(blueLed2, OUTPUT);
  pinMode(button, INPUT); // button on pin 2
  digitalWrite(blueLed2, HIGH);
}

void loop() {
  val = analogRead(tempPin);
  float mv = (val/1024.0)*5000;
  float cel = mv/10;
  float farh = (cel*9)/5 + 32;

  // print temperature in celsius and fahrenheit
  Serial.print("Temperature: ");
  Serial.print(cel);
  Serial.print("*C ");
  Serial.print(farh);
  Serial.print("*F");
  Serial.println();

  int state = digitalRead(button);
  /* check if button is pressed and it is over 3 seconds since last button press */
  if (state == HIGH && (millis() - changeTime) > 3000) {
    changeLights();
  }
}
```

```
    delay(1000);

}

void changeLights() {
    digitalWrite(blueLed2, LOW);

    // Red
    digitalWrite(redLed, HIGH);
    digitalWrite(yellowLed, LOW);
    digitalWrite(blueLed1, LOW);
    Serial.println("Red");
    delay(1000);

    // Yellow
    digitalWrite(redLed, LOW);
    digitalWrite(yellowLed, HIGH);
    digitalWrite(blueLed1, LOW);
    Serial.println("Yellow");
    delay(1000);

    // Blue 1
    digitalWrite(redLed, LOW);
    digitalWrite(yellowLed, LOW);
    digitalWrite(blueLed1, HIGH);
    Serial.println("Blue");
    delay(1000);

    digitalWrite(blueLed1, LOW);
    digitalWrite(blueLed2, HIGH);
    Serial.println("Waiting");

    // record the time since last change of lights
    changeTime = millis();
    // then return to the main program loop
}
```