Sensor node/mote Connect2Server

Πίνακας περιεχομένων

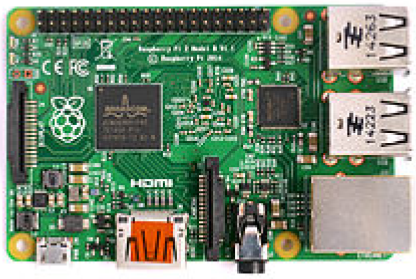
1. Create a mote with Raspberry Pi . . . 1
1.1. Install Raspberry Pi . . . 2
1.1.1. Step 1: Download Raspbian . . . 2
1.1.2. Step 2: Unzip the file . . . 3
1.1.3. Step 3: Write the disc image to your microSD card . . . 3
1.1.4. Step 4: Enabling SSH . . . 3
1.1.5. Step 5: Put the microSD card in your Pi and boot up . . . 4
1.1.6. Step 6: Access via SSH . . . 4
1.1.7. Step 7: Configure your Raspberry Pi. . . . 4
1.2. Arduino Uno Raspberry Pi Serial Communication . . . 5
1.2.1. Serial config on Raspi . . . 5
1.2.2. Reading in arduino . . . 6
1.2.3. Python . . . 6
1.2.4. PHP . . . 6
1.2.5. NodeJS . . . 7
1.3. Send data2server . . . 8
1.3.1. NodeJS . . . 8

1. Create a mote with Raspberry Pi

The Raspberry Pi is a low-cost credit-card sized single-board computer. The Raspberry Pi was created in the UK by the Raspberry Pi Foundation. The Raspberry Pi Foundation’s goal is to "advance the education of adults and children, particularly in the field of computers, computer science and related subjects."

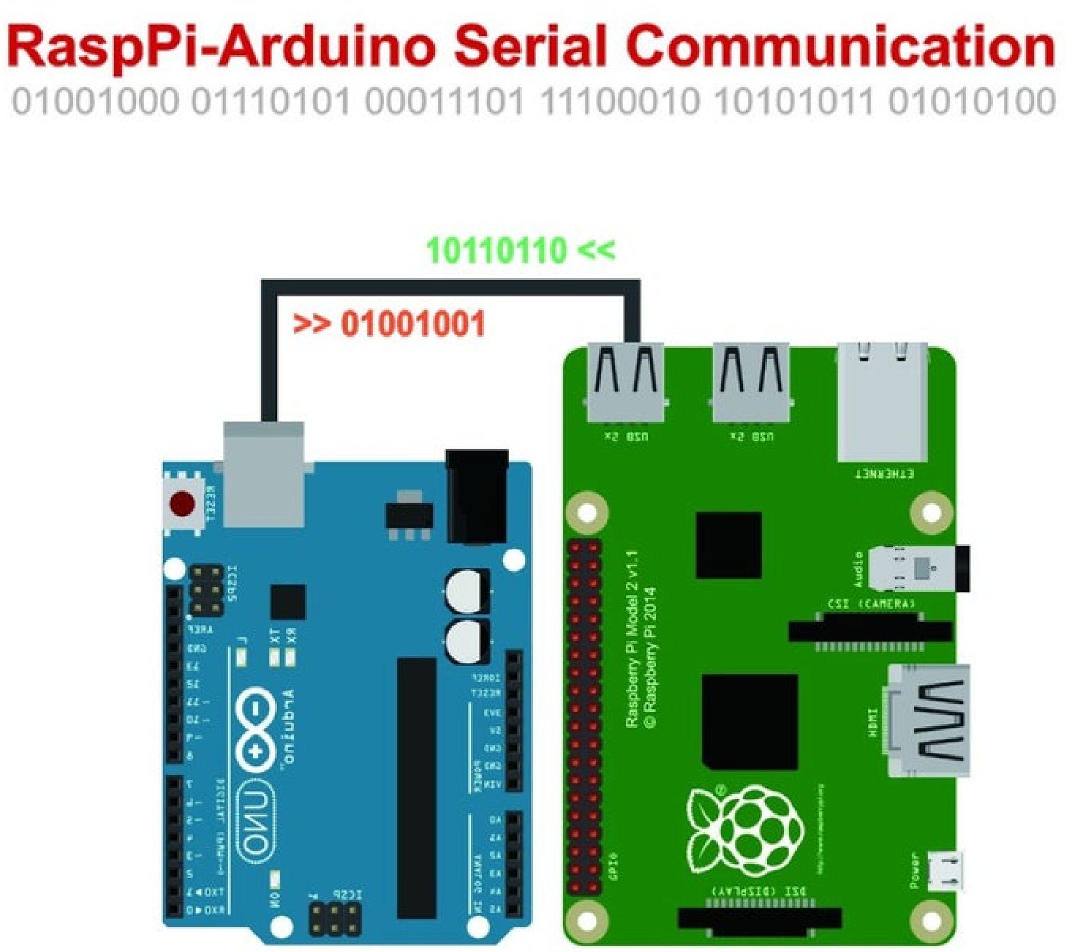
Wikipedia

Raspberry Pi 2 Model B



Remember

A mote is a node but a node is not always a mote!



1.1. Install Raspberry Pi

1.1.1. Step 1: Download Raspbian

Download the Raspbian disc image - Choose Raspbian Lite

Why Raspbian Lite?

Because it is a lightweight version of the Raspbian and it doesn’t have a graphical user interface installed.

This means that it doesn’t have any unnecessary software installed that we don’t need for our projects, so this makes it the perfect solution for future automation projects.

1.1.2. Step 2: Unzip the file

- Windows users, you'll want 7-Zip.
- Linux users will use the appropriately named Unzip.

1.1.3. Step 3: Write the disc image to your microSD card

Next, pop your microSD card into your computer and write the disc image to it. You'll need a specific program to do this:

- Windows users, your answer is Win32 Disk Imager.
- Linux people, Etcher – which also works on Windows – is what the Raspberry Pi Foundation recommends.

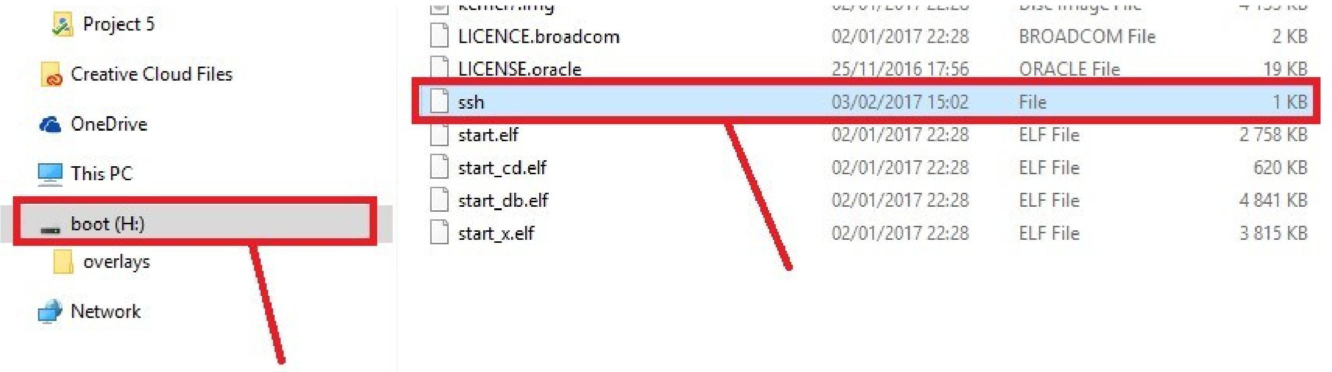
The process of actually writing the image will be slightly different across these programs, but it's pretty self-explanatory no matter what you're using.

- Each of these programs will have you select the destination (make sure you've picked your microSD card!) and the disc image (the unzipped Raspbian file).
- Choose, double-check, and then hit the button to write.

1.1.4. Step 4: Enabling SSH

- Windows users

Create ssh file (no extension)



LICENCE.broadcom	02/01/2017 22:28	BROADCOM File	2 KB
LICENSE.oracle	25/11/2016 17:56	ORACLE File	19 KB
ssh	03/02/2017 15:02	File	1 KB
start.elf	02/01/2017 22:28	ELF File	2 758 KB
start_cd.elf	02/01/2017 22:28	ELF File	620 KB
start_db.elf	02/01/2017 22:28	ELF File	4 841 KB
start_x.elf	02/01/2017 22:28	ELF File	3 815 KB

- Linux Users

Create ssh file

```
sudo fdisk -l
# find dev and Boot partition
sudo mkdir /mnt/sdcardP1
sudo mount /dev/device_partion_boot /mnt/sdcardP1 -rw
cd /mnt/sdcardP1
sudo touch ssh
```

1.1.5. Step 5: Put the microSD card in your Pi and boot up

Your default credentials are username **pi** and password **raspberry**

1.1.6. Step 6: Access via SSH

- The boot protocol for the ethernet interface is set to DHCP by default

You can find the open SSH ports on your network using the nmap utility:

find ports on Network

```
nmap -p 22 --open -sV 192.168.1.0/24
```

You should find your pi listed in the output along with the IP assigned to the pi.

- You can change the boot protocol to static and define a static IP address for the pi by editing the ifcfg-eth0 file:

static IP address

```
sudo fdisk -l
# find dev and Boot partition
sudo mkdir /mnt/sdcardP1
sudo mount /dev/device_partion_ext /mnt/sdcardP1 -rw
cd /mnt/sdcardP1
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Then edit the file to suit your needs

static IP address

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
NETWORK=192.168.1.0
NETMASK=255.255.255.0
IPADDR=192.168.1.200
GATEWAY=192.168.1.1
```

1.1.7. Step 7: Configure your Raspberry Pi.

raspi-config is the Raspberry Pi configuration tool

config Pi

```
sudo raspi-config
```

It has the following options available:

config options

```
 ┌───────────────────────────────────────┤ Raspberry Pi Software Configuration Tool
(raspi-config) ├──────────────────────────────────────────┐
 │
 │
 │        1 Change User Password Change password for the current user
 │
 │        2 Network Options      Configure network settings
 │
 │        3 Boot Options         Configure options for start-up
 │
 │        4 Localisation Options Set up language and regional settings to match your
location        │
 │        5 Interfacing Options  Configure connections to peripherals
 │
 │        6 Overclock            Configure overclocking for your Pi
 │
 │        7 Advanced Options     Configure advanced settings
 │
 │        8 Update               Update this tool to the latest version
 │
 │        9 About raspi-config   Information about this configuration tool
 │
 │
 │
 │
 │
 │
 │
 │                            <Select>                           <Finish>
 │
 │
 │
 └─────────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────────────────────────────────────────────
───────────────────────────┘
```

1.2. Arduino Uno Raspberry Pi Serial Communication

1.2.1. Serial config on Raspi

config 1 (recommended)

```
whoami
sudo usermod -a -G dialout pi
reboot
```

This gives read/write permission for all users to the Raspberry Pi (potentially unsafe):

config 2

```
sudo chmod 777 /dev/ttyACM0
```

This provides some configuration for the Arduino serial connection:

configuration for the Arduino serial connection

```
sudo stty -F /dev/ttyACM0 cs8 9600 ignbrk -brkint -icrnl -imaxbel -opost -onlcr -isig
-icanon -iexten -echo -echoe -echok -echoctl -echoke noflsh -ixon -crtscts
```

1.2.2. Reading in arduino

C code in the arduino

```
void loop() {
    meas = analogRead(a);
    if (Serial.available())
    {
        if (Serial.read() == '1')
        {
            Serial.println(meas);
        }
    }
}
```

1.2.3. Python

Python code in Raspberry Pi

```
import serial
from datetime import datetime
from time import sleep
now = datetime.now()

ser = serial.Serial('/dev/ttyACM0', 9600)
ser.write("1".encode())
sleep(0.05);
s = ser.readline()

file = open("dataset", "a")
file.write(now.strftime("%Y-%m-%d %H:%M") + " Sensor Value:" + str(s)+ "\n")
file.close()
```

1.2.4. PHP

ℹ *PHP Class*

Download Class

PHP code in Raspberry Pi - read

```
<?php

include "php_serial.class.php";

$serial = new phpSerial();
$serial->deviceSet("/dev/ttyACM0");
$serial->confBaudRate(9600);
$serial->confParity("none");
$serial->confCharacterLength(8);
$serial->confStopBits(1);
$serial->confFlowControl("none");

$serial->deviceOpen();
$read = $serial->readPort();
$serial->deviceClose();

echo $read
```

Sends a string to the Arduino.

PHP code in Raspberry Pi - send

```
<?php

error_reporting(E_ALL);
ini_set('display_errors', '1');
include "php_serial.class.php";

$serial = new phpSerial;
$serial->deviceSet("/dev/ttyAMA0");
$serial->confBaudRate(115200);
$serial->confParity("none");
$serial->confCharacterLength(8);
$serial->confStopBits(1);
$serial->deviceOpen();
$serial->sendMessage("Hello from my PHP script, say hi back!");

$serial->deviceClose();

echo "I've sended a message! \n\r";
```

1.2.5. NodeJS

Read the writing carefully on your Raspberry Pi circuit board to confirm it indicates something like “Raspberry Pi 4 Model B” or “Raspberry Pi 2 Model B”. If in doubt, run the following command in the terminal:

```
$ uname -m
```

If the result returned starts with **“armv6”**, you are running a Raspberry Pi based on the older ARMv6 chipset and the next Node.js installation step **will not work**; otherwise, you are ready for the next step.

Install NodeJS

```
curl -sL https://deb.nodesource.com/setup_13.x | sudo -E bash -
sudo apt install -y nodejs
npm install raspi-serial
```

Install NodeJS - armv6

```
cd ~
wget http://nodejs.org/dist/v6.2.1/node-v6.2.1-linux-armv6l.tar.gz
tar -xzf node-v6.2.1-linux-armv6l.tar.gz
cd node-v6.2.1-linux-armv6l/
sudo cp -R * /usr/local/
export PATH=$PATH:/usr/local/bin
npm install raspi-serial
```

NodeJS code in Raspberry Pi - read

```
mport { init } from 'raspi';
import { Serial } from 'raspi-serial';

init(() => {
  var serial = new Serial();
  serial.open(() => {
    serial.on('data', (data) => {
      process.stdout.write(data);
    });
    serial.write('Hello from raspi-serial');
  });
});
```

1.3. Send data2server

1.3.1. NodeJS

NodeJS code in Raspberry Pi - send

```
...
var serverIOT=IP_SERVER
    const socket = require('socket.io-client')('https://'+serverIOT+':9080');
    socket.on('connect', function () {
        socket.emit('subscribe', log);
        var obj = new Object();
    obj.room = log;
    obj.message  = data;
    var text = JSON.stringify(obj);
        var text1 = Buffer.from(text);
        var text5 = text1.toString('base64');
        socket.emit('log', text5, log )
        //console.log(util.inspect(text5, false, null, true /* enable colors */))

    res.json({
        'message':"ok"
    });
    });

...
```

You can find more information here and here