Co-funded by the
Erasmus+ Programme
of the European Union

WirelessUP!

UPraising VET skills for innovation in European electrotechnical sector

Project number: 2017-1-HR01-KA202-035434

WirelessUP! Toolkit

Module 3.2: Implementing Wireless Technologies in Automation Systems

Raspberry Pi project

Intellectual Output 3

February 2019

Content

1. Introduction to Raspberry PI platform

A Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. It plugs into a monitor and you can attach a keyboard, mouse and speakers.

The Raspberry Pi can be used for browsing the web, creating documents and spreadsheets, playing games, watching videos and lots more.

It also provides a great environment for learning programming and digital making. You can also connect up hardware to the Pi’s GPIO (general purpose input/output) pins and learn to program using electronics components.

A Raspberry Pi can also be built into custom projects such as IOT projects or home automation solutions.

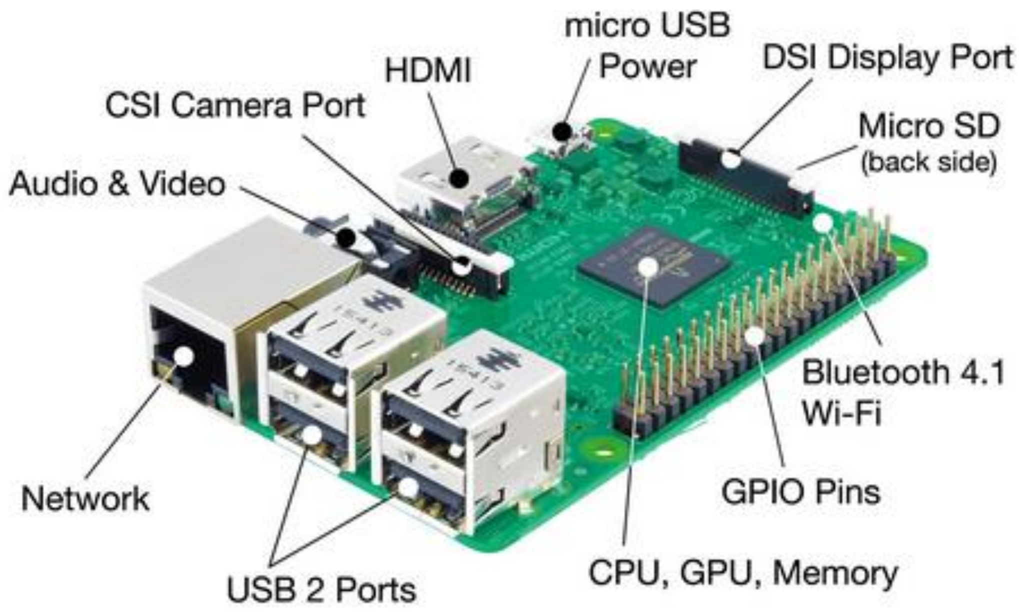


Figure 1. Raspberry PI platform

The Raspberry PI platform consist of the following parts:

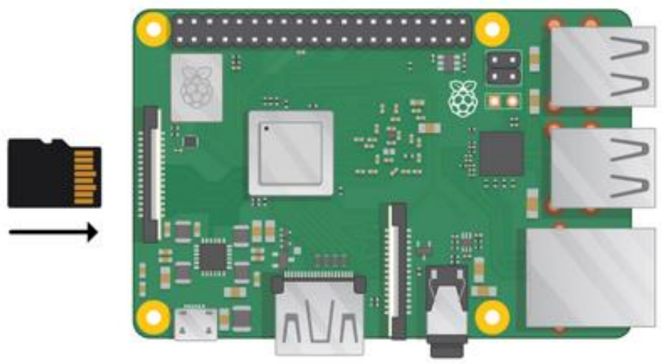
- **USB ports** — these are used to connect a mouse and keyboard. You can also connect other components, such as a USB drive.
- **SD card slot** — you can slot the SD card in here. This is where the operating system software and your files are stored.
- **Ethernet port** — this is used to connect the Raspberry Pi to a network with a cable. The Raspberry Pi can also connect to a network via wireless LAN.
- **Audio jack** — you can connect headphones or speakers here.
- **HDMI port** — this is where you connect the monitor (or projector) that you are using to display the output from the Raspberry Pi. If your monitor has speakers, you can also use them to hear sound.
- **Micro USB power connector** — this is where you connect a power supply. You should always

do this last, after you have connected all your other components.

- **GPIO ports** — these allow you to connect electronic components such as LEDs and buttons to the Raspberry Pi.

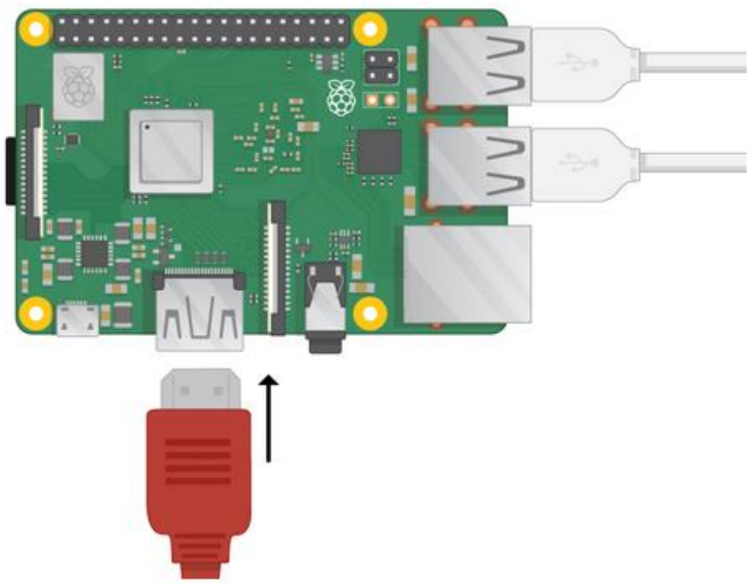
2. Connect your Raspberry Pi

Check whether your Raspberry Pi already has an SD card in the slot at the underside. If it doesn't, insert an SD card with the Raspbian operating system installed (via NOOBS).



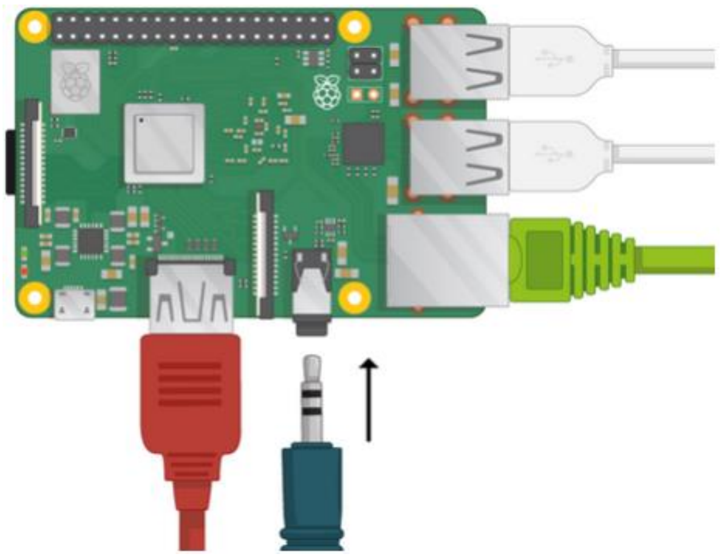
Lots of micro SD cards will come inside a larger adapter — you can slide the card out using the lip at the bottom.

Find the ***USB connector*** for your mouse and connect the mouse to one of the USB ports on the Raspberry Pi (it doesn't matter which one). Connect the ***keyboard*** in the same way. Look at the ***HDMI port*** on the Raspberry Pi — notice that it has a large flat side on top (Figure next page).

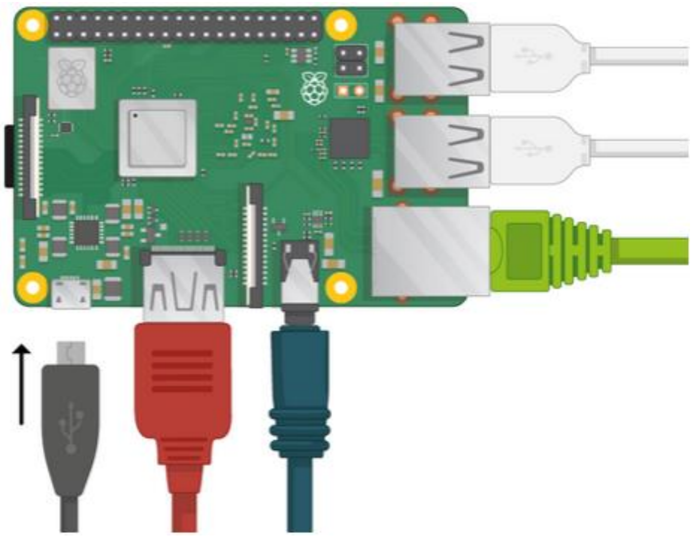


Make sure your monitor is plugged into a wall socket and turned on. Connect the monitor cable to the Pi’s HDMI port — use an adapter if necessary.

Nothing will display yet. If you want to connect the Pi to the internet via Ethernet, use an Ethernet cable to connect the Ethernet port on the Raspberry Pi to an Ethernet socket in the wall or on your internet router. You don’t need to do this if you’ll be using WiFi or if you don’t want to connect to the internet. Sound will come from your screen if it has speakers or you can connect headphones or speakers to the audio jack if you have them.

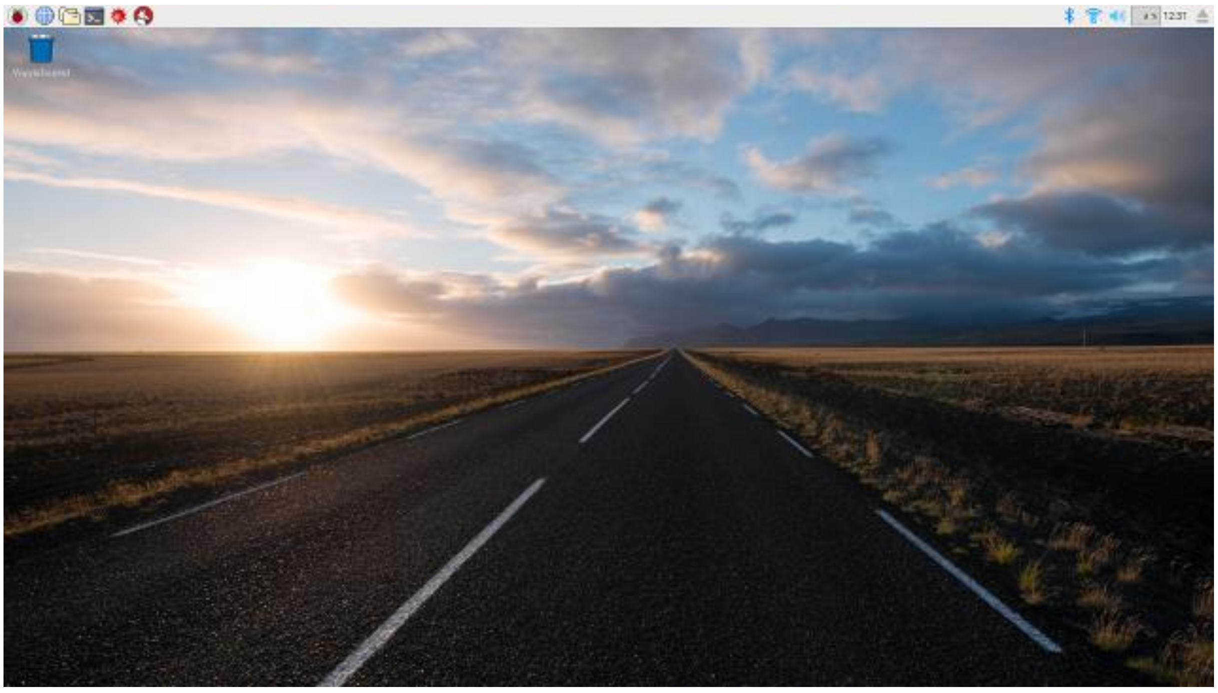


Plug the power supply into a socket and connect it to the micro USB power port.



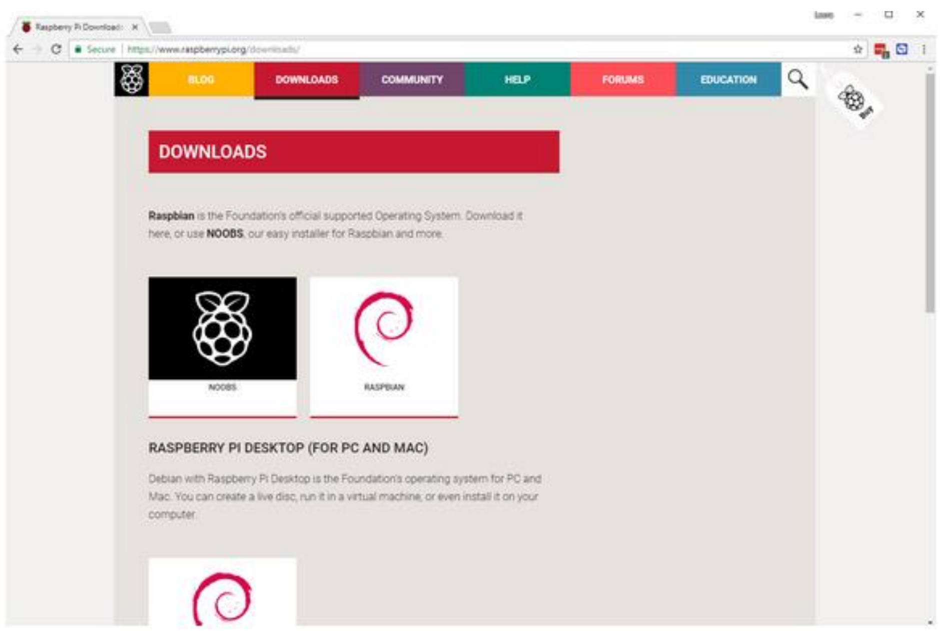
You should see a red light on the Raspberry Pi and raspberries on the monitor.

The Pi will boot up into a graphical desktop.



3. Installing Raspbian operating system with NOOBS

Using NOOBS is the easiest way to install Raspbian on your SD card. To get hold of a copy of NOOBS, visit www.raspberrypi.org/downloads/.



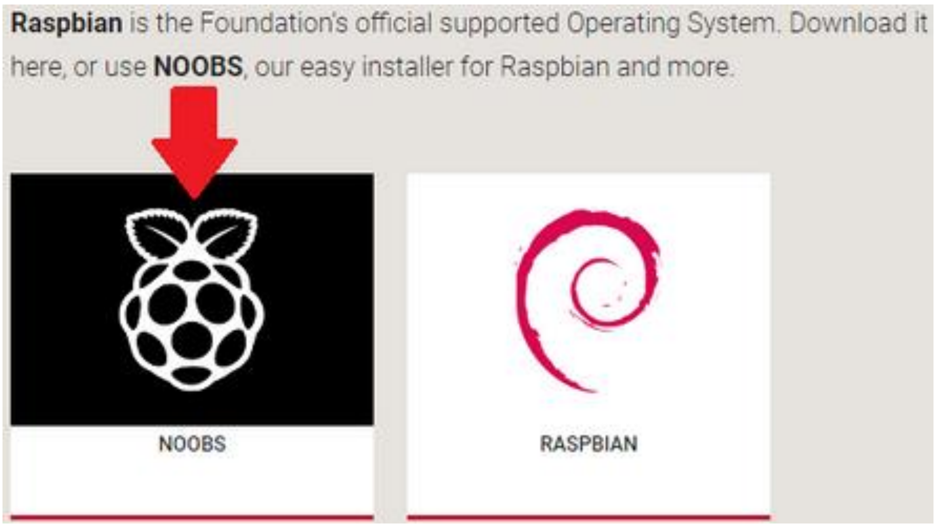
DOWNLOADS

Raspbian is the Foundation's official supported Operating System. Download it here, or use **NOOBS**, our easy installer for Raspbian and more.

RASPBERRY PI DESKTOP (FOR PC AND MAC)

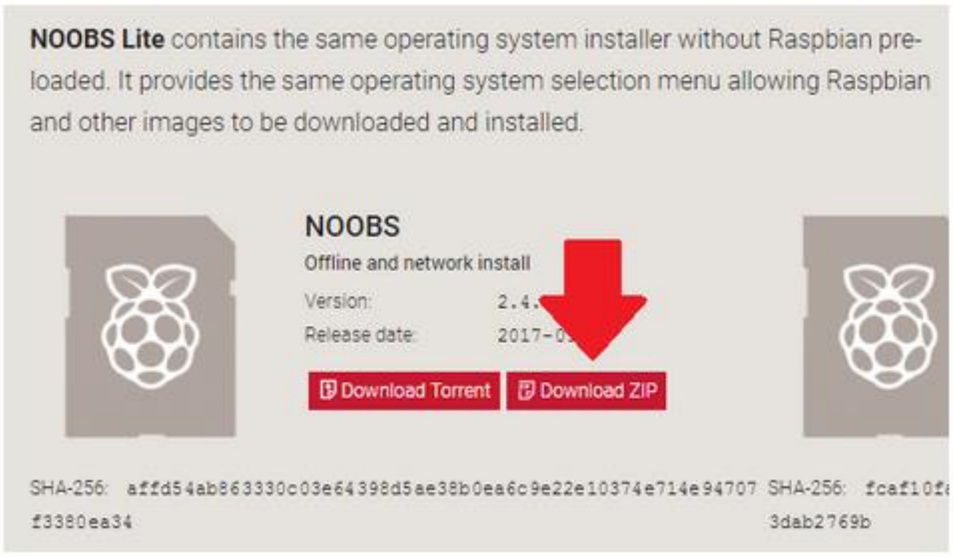
Debian with Raspberry Pi Desktop is the Foundation's operating system for PC and Mac. You can create a live disc, run it in a virtual machine, or even install it on your computer.

You should see a box with a link to the NOOBS files. Click on the link.



Raspbian is the Foundation's official supported Operating System. Download it here, or use **NOOBS**, our easy installer for Raspbian and more.

The simplest option is to download the zip archive of the files.


NOOBS Lite contains the same operating system installer without Raspbian pre-loaded. It provides the same operating system selection menu allowing Raspbian and other images to be downloaded and installed.

NOOBS
Offline and network install
Version: 2.4.
Release date: 2017-0

SHA-256: affd54ab863330c03e64398d5ae38b0ea6c9e22e10374e714e94707f3380ea34 SHA-256: fcaf10f 3dab2769b

3.1 Formatting the SD Card

If the SD card on which you wish to install Raspbian currently has an older version of Raspbian on it, you may wish to back up the files from the card first, as they will be overwritten during this process. Visit the SD Association’s website and download SD Formatter 4.0 for Windows or Mac.

Follow the instructions to install the software:

- Insert your SD card into the computer or laptop’s SD card reader and make a note of the drive letter allocated to it, e.g. **F:/**.
- In SD Formatter, select the drive letter for your SD card and format it.

3.2 Extracting NOOBS from the zip archive

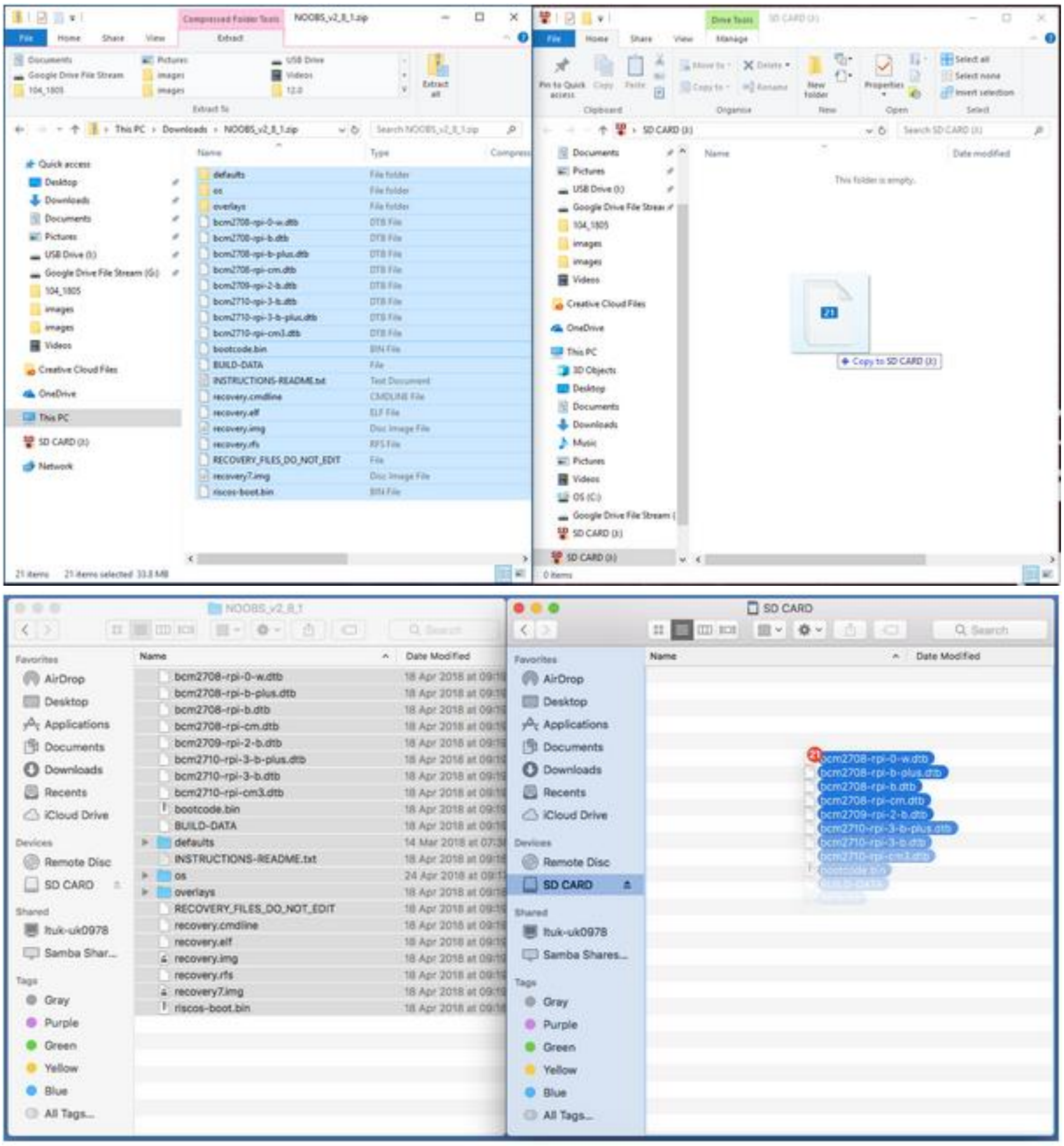
Next, you will need to extract the files from the NOOBS zip archive you downloaded from the Raspberry Pi website.

- Go to your Downloads folder and find the zip file you downloaded.
- Extract the files and keep the resulting Explorer/Finder window open.

3.3 Copying the files

Now open another Explorer/Finder window and navigate to the SD card. It’s best to position the two windows side by side.

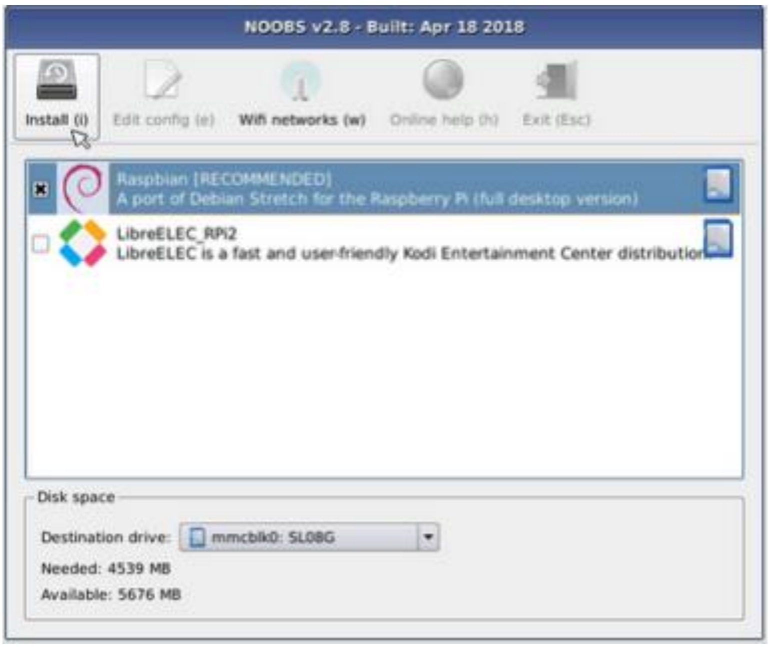
- Select all the files from the NOOBS folder and drag them onto the SD card.



- Eject the SD card.

3.4 Booting from NOOBS

Once the files have been copied over, insert the micro SD Card into your Raspberry Pi, and plug the Pi into a power source. You will be offered a choice when the installer has loaded. You should check the box for ***Raspbian***, and then click ***Install***.


NOOBS v2.8 - Built: Apr 18 2018

[✓ Raspbian [RECOMMENDED]] A port of Debian Stretch for the Raspberry Pi (full desktop version)

LibreELEC_RPi2 LibreELEC is a fast and user-friendly Kodi Entertainment Center distribution

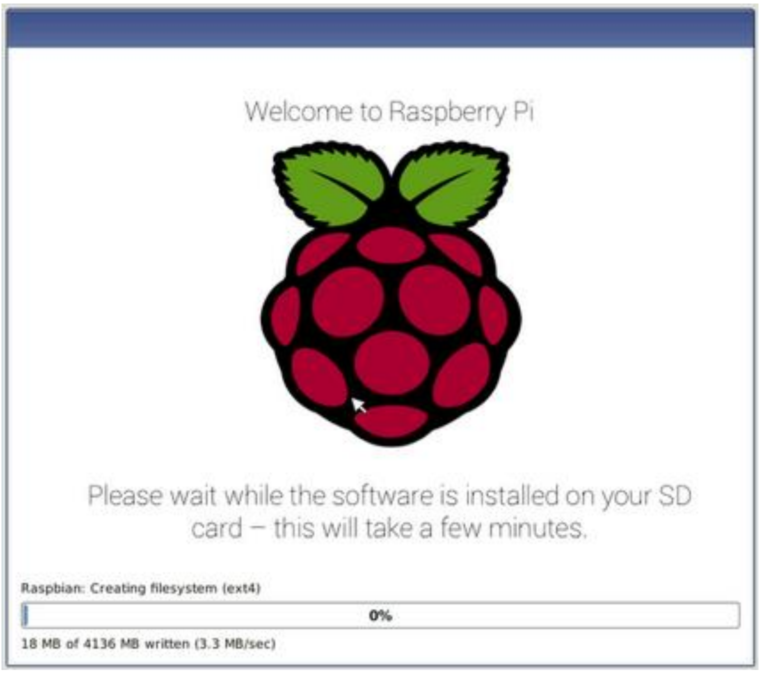
Disk space

Destination drive: mmcblk0: SL08G

Needed: 4539 MB

Available: 5676 MB

Click Yes at the warning dialog, and then sit back and relax. It will take a while but Raspbian will install.


Welcome to Raspberry Pi

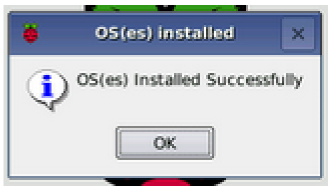
Please wait while the software is installed on your SD card – this will take a few minutes.

Raspbian: Creating filesystem (ext4)

0%

18 MB of 4136 MB written (3.3 MB/sec)

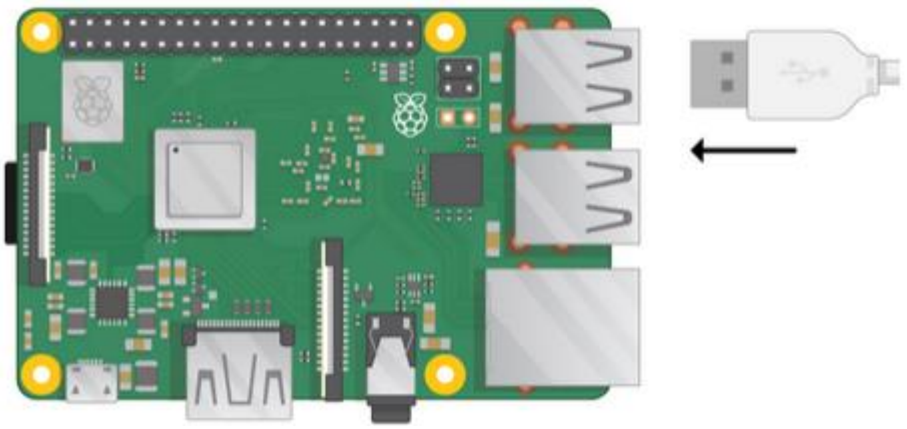
When Raspbian has been installed, click OK and your Raspberry Pi will restart and Raspbian will then boot up.


OS(es) installed

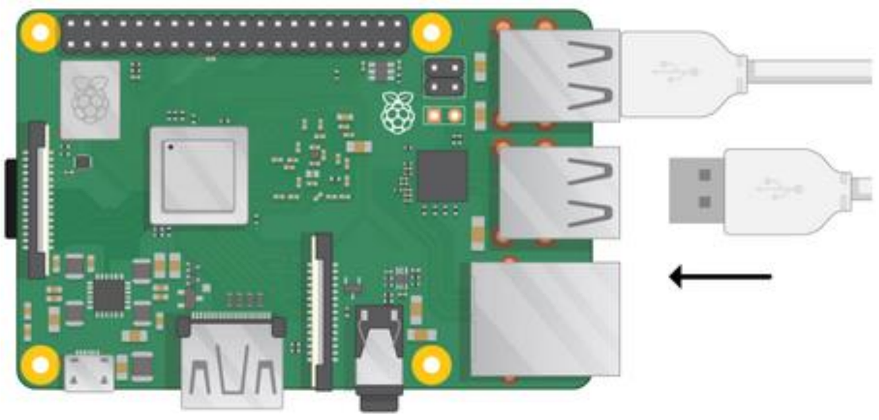
OS(es) Installed Successfully

4. Connecting peripheral units

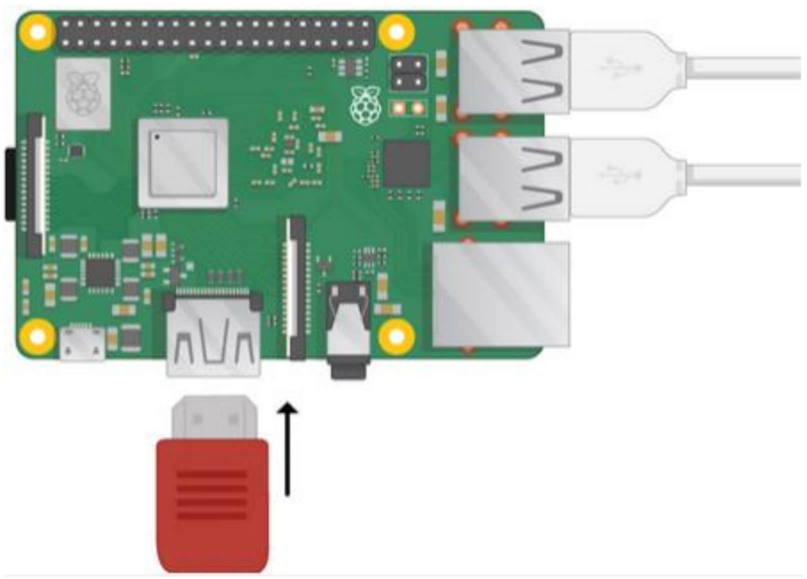
Find the USB connector for your mouse and connect the mouse to one of the USB ports on the Raspberry Pi (it doesn't matter which one).



Connect the keyboard in the same way.



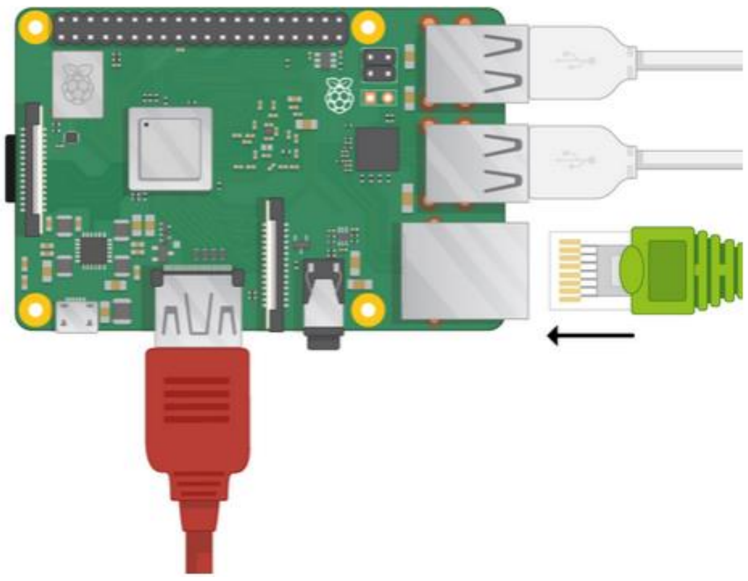
Look at the HDMI port on the Raspberry Pi — notice that it has a large flat side on top.



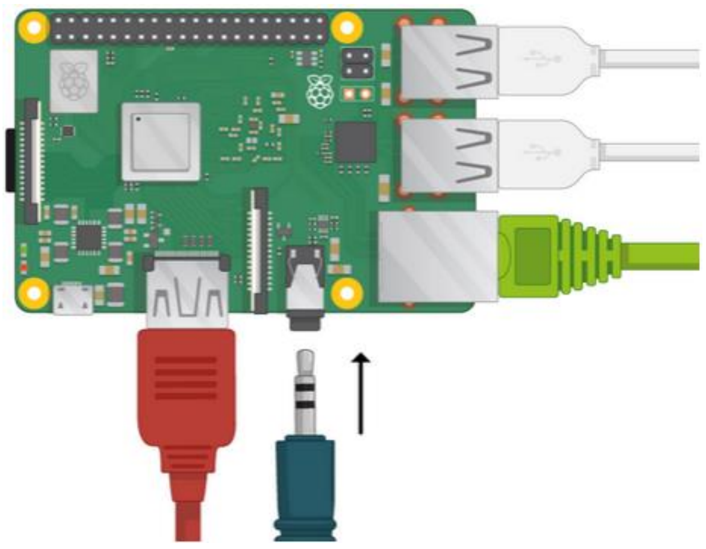
- Make sure your monitor is plugged into a wall socket and turned on.
- Connect the monitor cable to the Pi's HDMI port — use an adapter if necessary.

Nothing will display yet.

- If you want to connect the Pi to the internet via Ethernet, use an Ethernet cable to connect the Ethernet port on the Raspberry Pi to an Ethernet socket in the wall or on your internet router. You don't need to do this if you'll be using WiFi or if you don't want to connect to the internet.

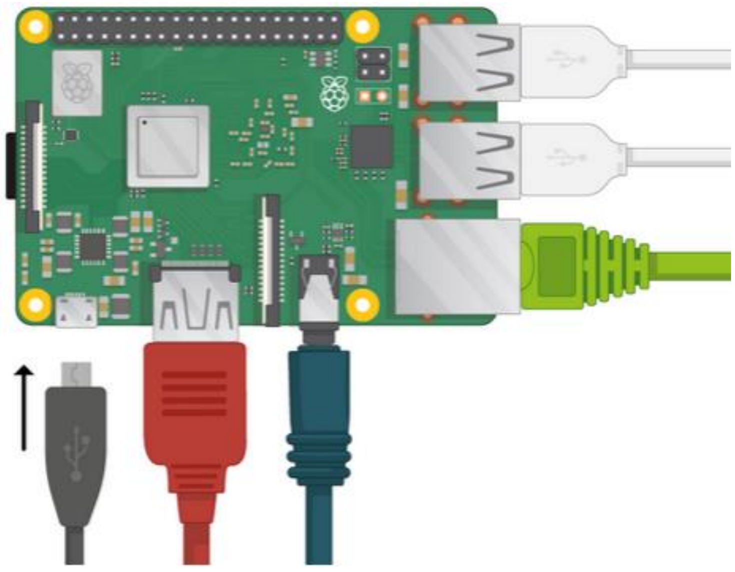


Sound will come from your screen if it has speakers or you can connect headphones or speakers to the audio jack if you have them.



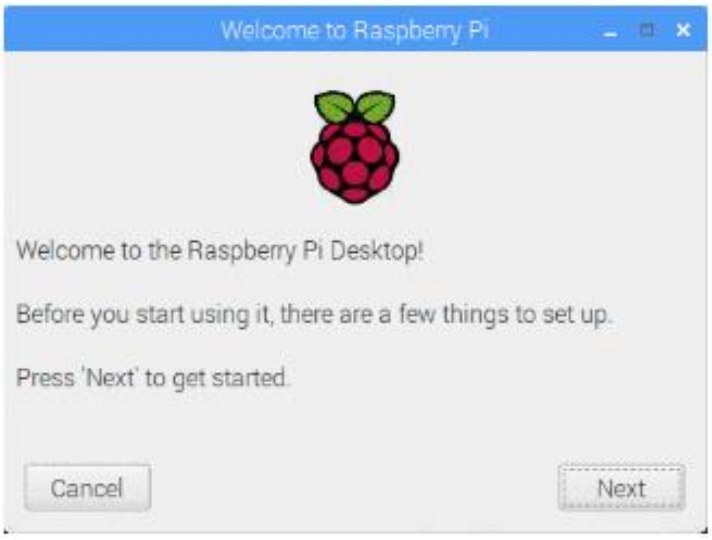
Notice that the micro USB power port has a longer flat side on top.

- Plug the power supply into a socket and connect it to the micro USB power port.



5. Finishing the setup

When you start your Raspberry Pi for the first time, the Welcome to Raspberry Pi application will pop up and guide you through the initial setup.

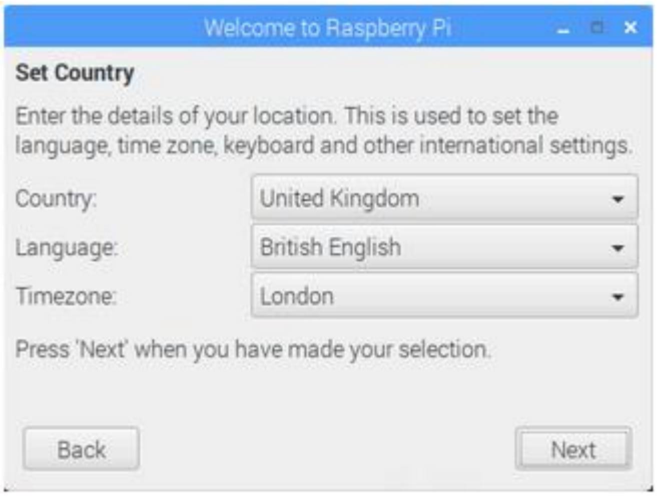


Welcome to the Raspberry Pi Desktop!

Before you start using it, there are a few things to set up.

Press 'Next' to get started.

Click **Next** to start the setup.


Welcome to Raspberry Pi

Set Country

Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

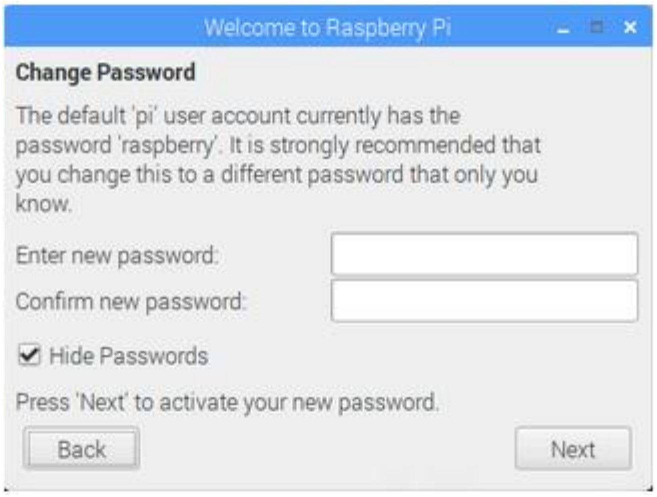
Country: United Kingdom

Language: British English

Timezone: London

Press 'Next' when you have made your selection.

Set your **Country**, **Language** and **Timezone**, then click Next again.


Welcome to Raspberry Pi

Change Password

The default 'pi' user account currently has the password 'raspberry'. It is strongly recommended that you change this to a different password that only you know.

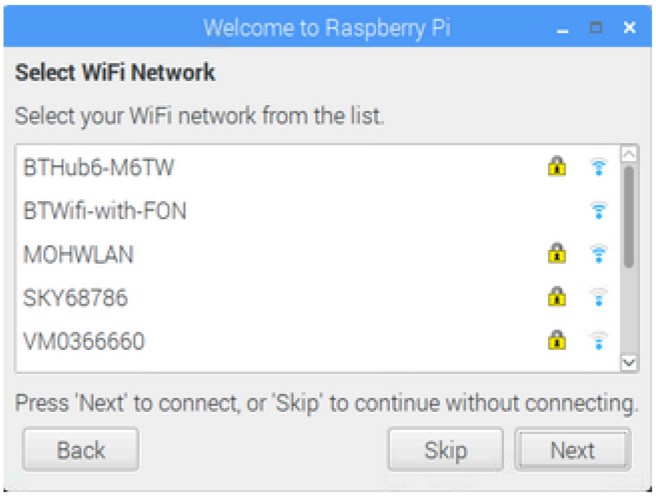
Enter new password:

Confirm new password:

☑ Hide Passwords

Press 'Next' to activate your new password.

Enter a new password for your Raspberry Pi and click Next.


Welcome to Raspberry Pi

Select WiFi Network

Select your WiFi network from the list.

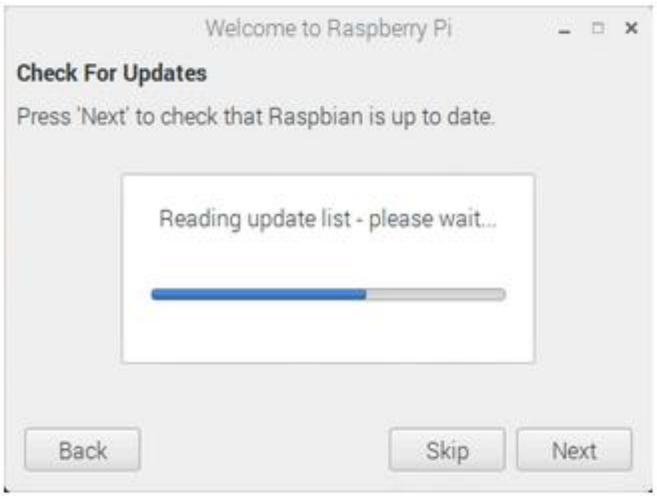
BTHub6-M6TW
BTWifi-with-FON
MOHWLAN
SKY68786
VM0366660

Press 'Next' to connect, or 'Skip' to continue without connecting.

Connect to your WiFi network by selecting its name, entering the password and clicking Next.

Note: if your Raspberry Pi model doesn't have wireless connectivity, you won't see this screen.

- Click Next and let the wizard check for updates to Raspbian and install them (this might take a little while).



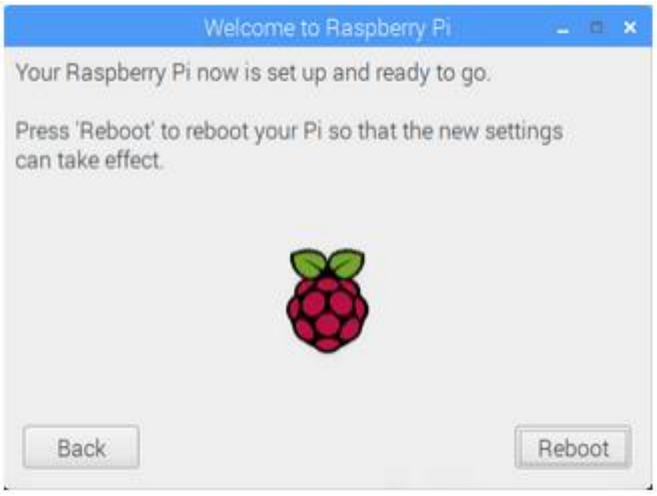
Check For Updates

Press 'Next' to check that Raspbian is up to date.

Reading update list - please wait...

Click Done or Reboot to finish the setup.

Note: you will only need to reboot if that's necessary to complete an update.

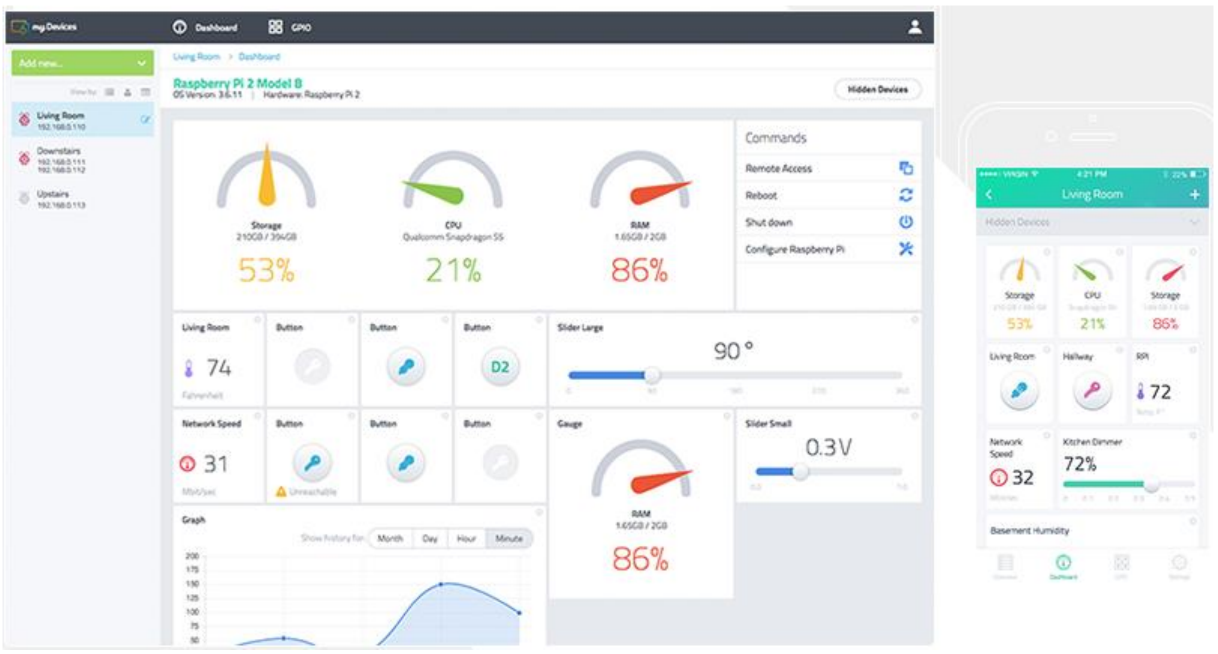


Your Raspberry Pi now is set up and ready to go.

Press 'Reboot' to reboot your Pi so that the new settings can take effect.

6. Cayenne – IOT cloud

Cayenne is the first of its kind. A drag and drop IoT project builder that empowers developers to quickly create and host their connected device projects. Cayenne was especially designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, analyze them and do many other cool things.



There are several major components in the platform:

- **Cayenne App** – setup and control your IoT projects with drag and drop widgets from an app.
- **Cayenne Online Dashboard** – Use a browser to setup and control your IoT projects.
- **Cayenne Cloud**- responsible for processing and the storage of the device, user and sensor data and for commands, actions, triggers and alerts.
- **Cayenne Agent** – enables communication with the server, agent and hardware for implementing incoming and outgoing commands, actions, triggers and alerts.

Every time you press a button from the Cayenne app or online dashboard, it travels to the Cayenne Cloud where it’s processed and finds its way to your hardware. It works the same in the opposite direction. You can use the Cayenne mobile app or online dashboard, it’s up to you. Any changes you make to hardware from the mobile app are reflected when viewing the online dashboard and vice versa.

6.1 Features

- Connection using Ethernet, Wi-Fi and cellular (mobile app only)
- Discover and setup Raspberry Pis on a network (Ethernet or Wi-Fi only)
- Customizable dashboard with drag and drop widgets
- Remotely access, reboot and shutdown a Pi
- Add and control sensors, actuators and extensions connected to Raspberry Pis
- Configure triggers for Pis, sensors and actuators
- Setup and receive threshold alerts via email and text messages
- Monitor device and sensor history data
- Remotely test and and configure hardware using GPIO
- Coming soon! Setup reoccurring actions and commands

6.2 Getting Started

This guide will help you get started using Cayenne in minutes. We will quickly cover the following:

- Creating your account
- Installing and setting up Cayenne using a terminal
- Configuring your first sensor (Temperature)
- Configure your first actuator (LED Switch)
- Setting up a trigger
- Preparing the Raspberry Pi platform

6.2.1 Power on the Raspberry Pi

Connect the power adapter to the Raspberry Pi and connect the Pi to the Internet using an Ethernet cable. Or, if you have a Wi-Fi dongle setup already, this works, too.

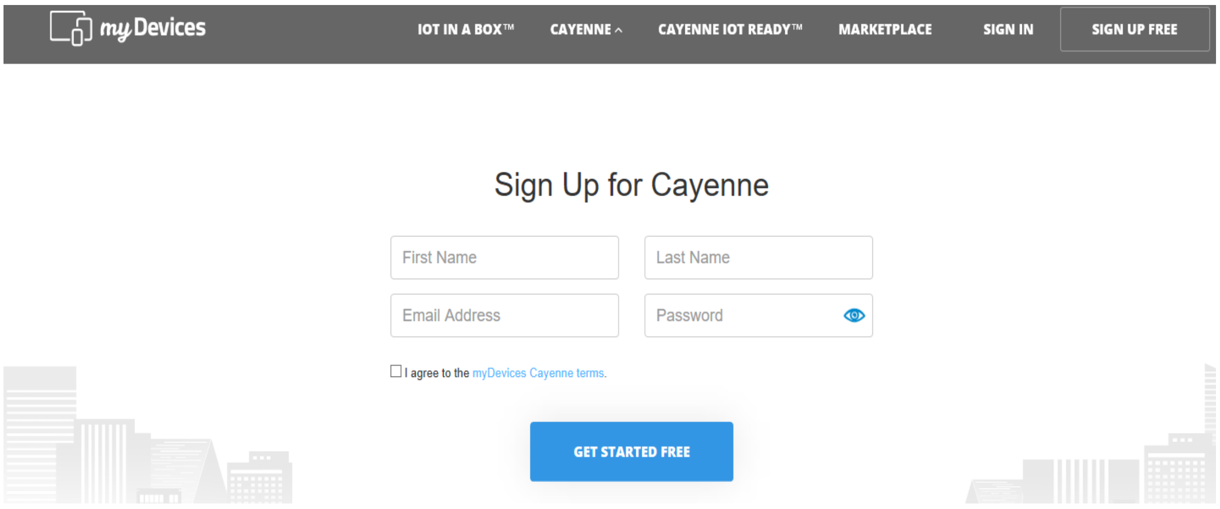
Make sure the Raspbian operating system is installed. Cayenne works with the Wheezyand Jessy OS versions of Raspbian. Please make sure that one of these is pre-installed.

Note: You do not need to make any configuration changes to the Pi device. Cayenne automatically handles setting up and configuring your Pi for use. However, it is recommended that you expand the filesystem on the flash card if necessary.

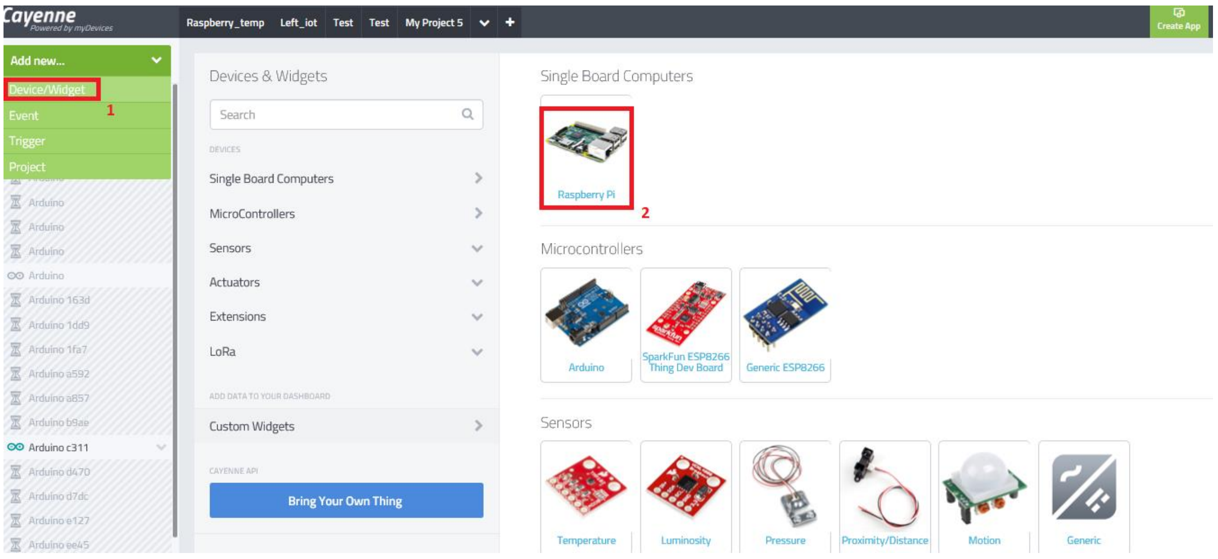
6.2.2 Creating the Cayenne IOT account

To use the online dashboard, you must first sign up for a free account. To begin creating your account, visit the Cayenne Sign Up page. On the “Sign Up” page, enter your Name, Email and create a Password.

Link for Cayenne – my Devices: https://mydevices.com/cayenne/signup/



After crating the account, you can now proceed with installing Cayenne onto your Raspberry Pi devices. First, add new device/widget from the dashboard and select Raspberry Pi.



6.2.3 Installing and setting up Cayene using a terminal

To download and install myDevices Cayenne on your Pi, use the Terminal on your Pi or SSH. Run the following commands:

```
wget https://cayenne.mydevices.com/dl/rpi_c9ufy9tgow.sh

sudo bash rpi_c9ufy9tgow.sh -v
```

Before you can manage your Raspberry Pi device using the online dashboard, you must install Cayenne onto the device. The instructions for installing Cayenne onto your device are also displayed in the Cayenne dashboard.

As soon as the installation process starts, the Installing screen automatically appears. From here you can check on the installation process as it completes.

Cayenne installs on your Raspberry Pi in 4 steps:

- Step 1: Installing libraries
- Step 2: Installing agent
- Step 3: Installing software
- Step 4: Installing drivers

This may take up to 10 minutes.

As soon as the installation process is completed, the online dashboard will automatically appear and Raspberry Pi will be ready to use!

6.2.4 Using drag and drop dashboard to control everything

After installing Cayenne, your Pi automatically shows up in the Cayenne drag and drop customizable dashboard. Within Cayenne, everything is a widget, which enables you to change the look and move things around for the look that works for you.

Things you can include:

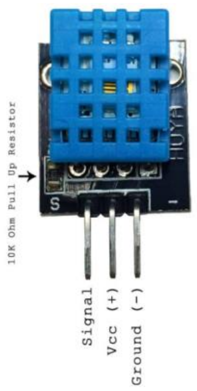
- View CPU, RAM and storage usage
- Remote access, restart and shutdown your Pi
- Configure hardware using GPIO controls
- Configure Raspberry Pi settings
- Customize widget type and look
- Add devices such as sensors, actuators (e.g. lights, motors) and extensions
- Configure triggers for Pis, sensors and actuators
- Schedule actions and commands

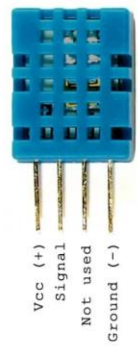
7. Cayenne IOT weather station project

Before starting the project assembly, make sure Raspberry Pi is powered off when connecting wires. When using a GPIO ribbon cable, make sure the power wire (it's a different color than the others) is connected to the corner of your Raspberry Pi and the top of your Pi cobbler. The weather station project will consist of Rasberry Pi platform and sensors for measuring temperature, humidity, CO2, light-level and relay 223/12V. Also, triggers for the desired scenarios will be created using mentioned sensors and actuator.

7.1 Connecting DHT11 or DHT 22 sensor

Sensors DHT 11 or DHT 22 are used for measuring temperature and humidity and can be connected with many platforms like Arduino and Raspberry Pi. There are 2 models of mentioned sensors, with three or four wire connections to Raspberry Pi.

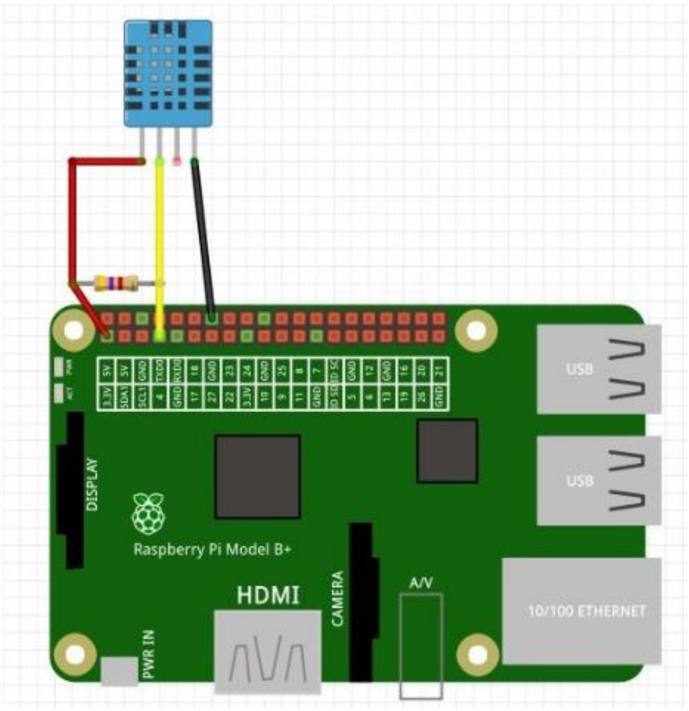




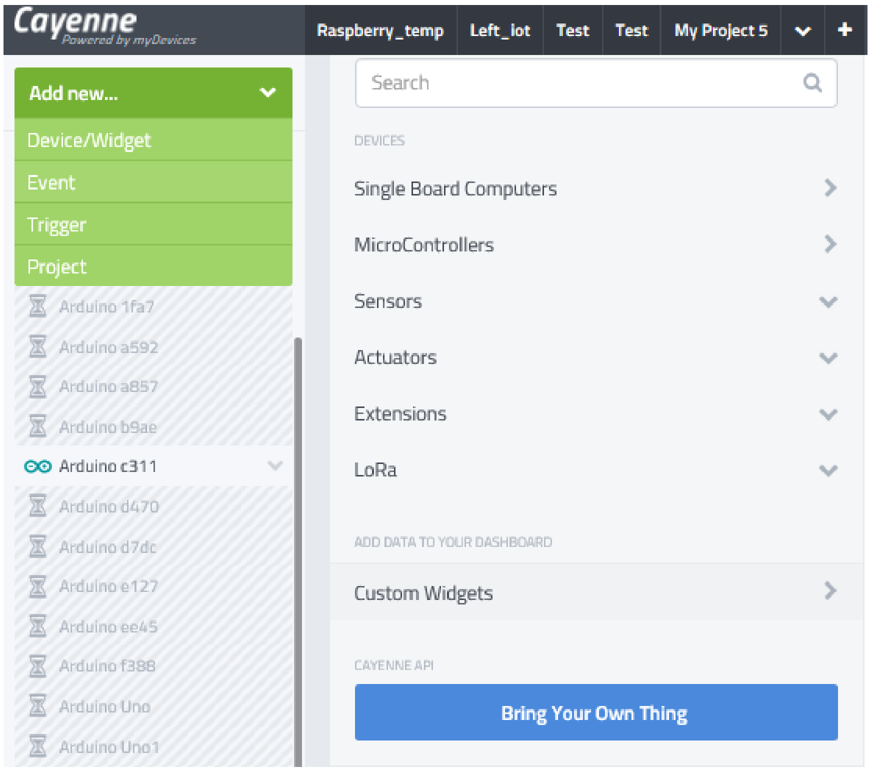
This project will read temperature and humidity from a DHT11 or DHT22 sensor and send the values to Cayenne using MQTT. The python MQTT client and Adafruit DHT sensor library will need to be installed for this script to run. Install the software prerequisites using the commands below and write it in the Raspberry Pi terminal:

```
sudo pip install paho-mqtt
sudo apt-get install build-essential python-dev python-openssl
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo python setup.py install
```

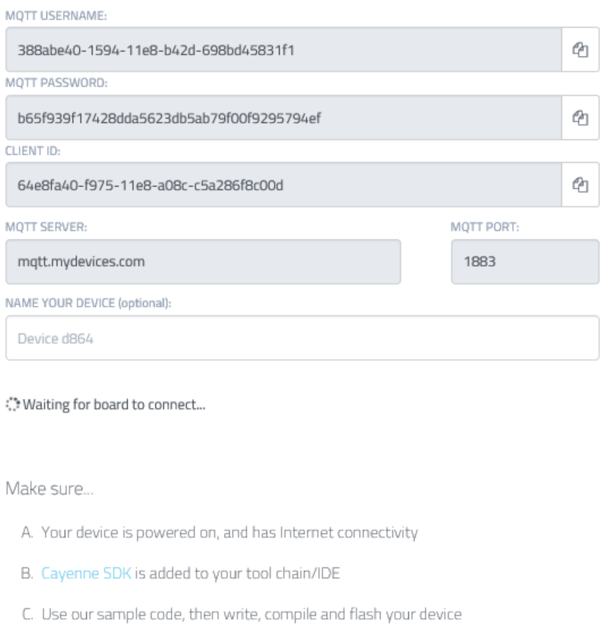
Wire the DHT11 as the picture below. The resistor is a 4.7k. The GPIO pin you connect it to does not matter. In the example code below, I used GPIO pin 17. For DHT with three connectors only 5V, ground and signal should be connected (without resistor).



On the dashboard, there are many sensors and actuators included but DHT 11 or DHT 22 sensors are not on the list so they have to be added as custom. From the Cayenne dashboard we have to add new Device/Widget and click on “Bring Your Own Thing” box.



After connecting the DHT 11 sensor, scripts needs to be uploaded to Raspberry Pi for communication and to get values through the server. Hence, the next thing is to write down the generated MQQT username, password and Client ID for the new device (DHT 11 sensor).


MQTT USERNAME:
388abe40-1594-11e8-b42d-698bd45831f1
MQTT PASSWORD:
b65f939f17428dda5623db5ab79f00f9295794ef
CLIENT ID:
64e8fa40-f975-11e8-a08c-c5a286f8c00d
MQTT SERVER:
mqtt.mydevices.com
MQTT PORT:
1883
NAME YOUR DEVICE (optional):
Device d864

Waiting for board to connect...

Make sure...

A. Your device is powered on, and has Internet connectivity

B. Cayenne SDK is added to your tool chain/IDE

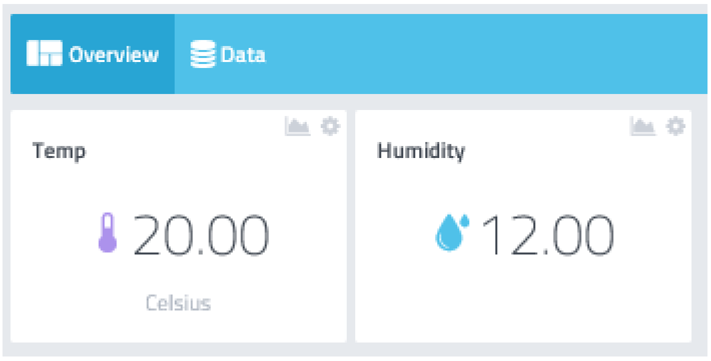
C. Use our sample code, then write, compile and flash your device

The next step is to upload the code for the DHT 11 sensor to the Raspberry Pi. On the Pi terminal, the next line of codes have to be writen and saved to /home/pi/python/tempsensor.py (path does not matter, just be sure to update the path to match yours throughout the next steps) and fill in username, password, and client id (lines 8, 9, 10) for the device. You can also delete the DHT11 or DHT22 lines based on which sensor you are using or add additional lines to check more than 1 sensor. The topic_dht lines will all need unique channels, so be sure the last digit in the string is unique "v1/username/things/clientid/data/ 1".

7.1.1 DHT sensor code

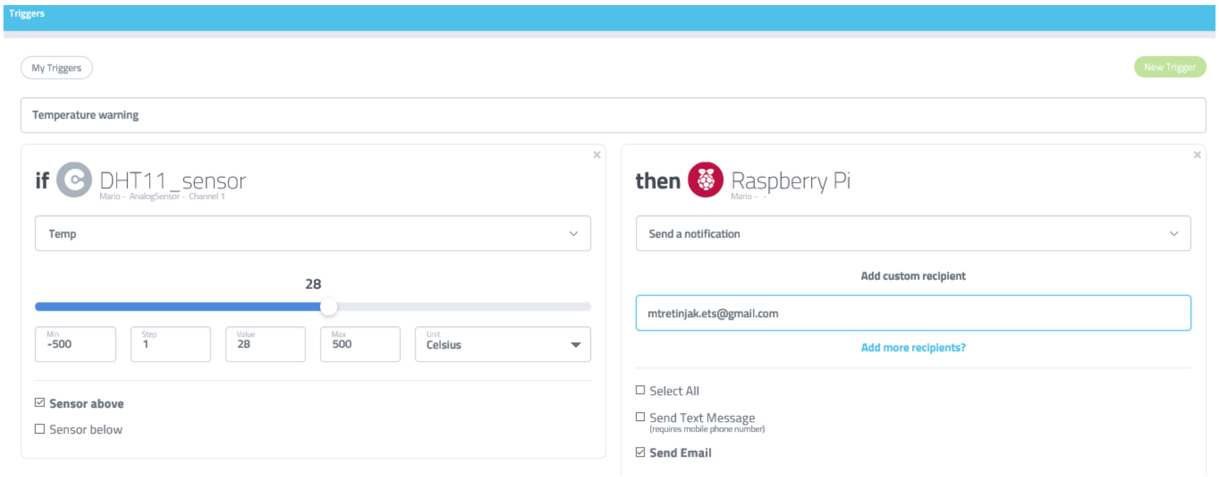
```
import paho.mqtt.client as mqtt
import time
import sys
import Adafruit_DHT

time.sleep(30) #Sleep to allow wireless to connect before starting MQTT

username = "MQTT Username From Dashboard"
password = "MQTT Passsword From Dashboard"
clientid = "MQTT Client ID From Dashboard"

mqttc = mqtt.Client(client_id=clientid)
mqttc.username_pw_set(username, password=password)
mqttc.connect("mqtt.mydevices.com", port=1883, keepalive=60)
mqttc.loop_start()

topic_dht11_temp = "v1/" + username + "/things/" + clientid + "/data/1"
topic_dht11_humidity = "v1/" + username + "/things/" + clientid + "/data/2"
topic_dht22_temp = "v1/" + username + "/things/" + clientid + "/data/3"
topic_dht22_humidity = "v1/" + username + "/things/" + clientid + "/data/4"

while True:
    try:
        humidity11, temp11 = Adafruit_DHT.read_retry(11, 17) #11 is the sensor type, 17 is the GPIO pin number (not physical pin number)
        humidity22, temp22 = Adafruit_DHT.read_retry(22, 18) #22 is the sensor type, 18 is the GPIO pin number (not physical pin number)

        if temp11 is not None:
            temp11 = "temp,c=" + str(temp11)
            mqttc.publish(topic_dht11_temp, payload=temp11, retain=True)
        if humidity11 is not None:
            humidity11 = "rel_hum,p=" + str(humidity11)
            mqttc.publish(topic_dht11_humidity, payload=humidity11, retain=True)
        if temp22 is not None:
            temp22 = "temp,c=" + str(temp22)
            mqttc.publish(topic_dht22_temp, payload=temp22, retain=True)
        if humidity22 is not None:
            humidity22 = "rel_hum,p=" + str(humidity22)
            mqttc.publish(topic_dht22_humidity, payload=humidity22, retain=True)
        time.sleep(5)
    except (EOFError, SystemExit, KeyboardInterrupt):
        mqttc.disconnect()
        sys.exit()
```

For testing, just run the file with python /home/pi/python/tempsensor.py. After it is working correctly, add the widget to the dashboard permanently by clicking the + in the upper right hand. Add the python file to your crontab with sudo crontab -e then type in @reboot python /home/pi/python/tempsensor.py and save. Reboot and you should see the values populate on your MQTT dashboard. To remove the script, use the command sudo crontab -e again and either put a # in front of the @reboot line you added, or delete it entirely.



7.1.2 Adding temperature trigger – email notification

Cayenne server also provides triggers for connected devices. We will add a temperature trigger and set it for e-mail notification on the desired e-mail when the temperature gets over 28°C. From the Cayenne dashboard, add a new Trigger and name it Temperature warning.


Triggers

My Triggers

New Trigger

Temperature warning

if DHT11_sensor
Mario - AnalogSensor - Channel 1

Temp

28

Min -500
Step 1
Value 28
Max 500
Unit Celsius

☑ Sensor above
☐ Sensor below

then Raspberry Pi
Mario -

Send a notification

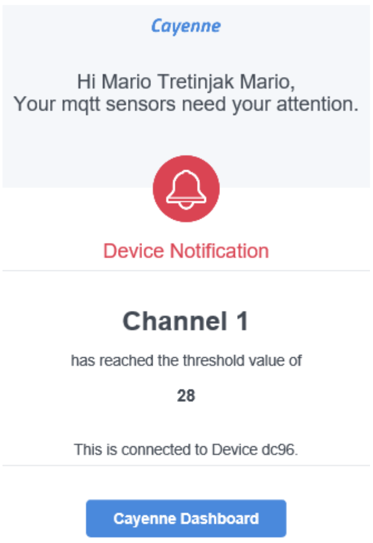
Add custom recipient

mtretinjak.ets@gmail.com

Add more recipients?

☐ Select All
☐ Send Text Message
(requires mobile phone number)
☑ Send Email

Next thing is to drag and drop our DHT11_sensor device under "if" statement and set it to the desired temperature warning (above 28°C). Under "then" statement drag the Raspberry Pi device and set custom recipient e-mail adress like on the picture above. Finish the trigger setting by clicking "save" on the right bottom corner of the screen. The Trigger is set and when the temperature gets over the defined value, Cayenne will sent a notification to the recipient   e-mail adress.


Cayenne

Hi Mario Tretinjak Mario,
Your mqtt sensors need your attention.

Device Notification

Channel 1

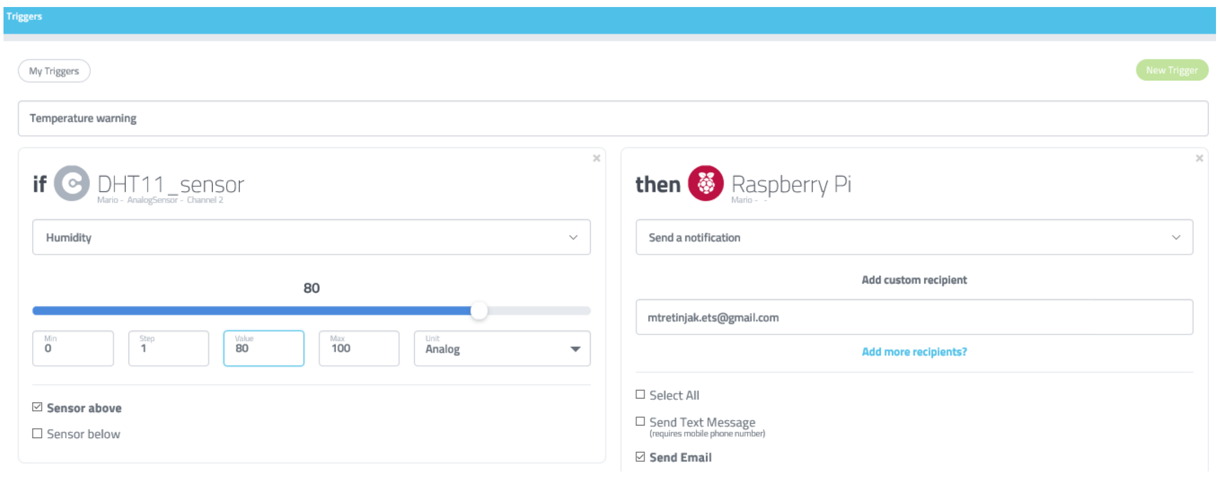
has reached the threshold value of

28

This is connected to Device dc96.

Cayenne Dashboard

A Trigger with humidity can be also added to the project. If the humidity value reaches a value of 80, Cayenne will send a notification to the recipient e-mail adress.


Triggers

My Triggers

New Trigger

Temperature warning

if DHT11_sensor
Mario - AnalogSensor - Channel 2

Humidity

80

Min 0 | Step 1 | Value 80 | Max 100 | Unit Analog

☑ Sensor above
☐ Sensor below

then Raspberry Pi
Mario -

Send a notification

Add custom recipient

mtretinjak.ets@gmail.com

Add more recipients?

☐ Select All
☐ Send Text Message (requires mobile phone number)
☑ Send Email

7.2 Connecting light-level sensor

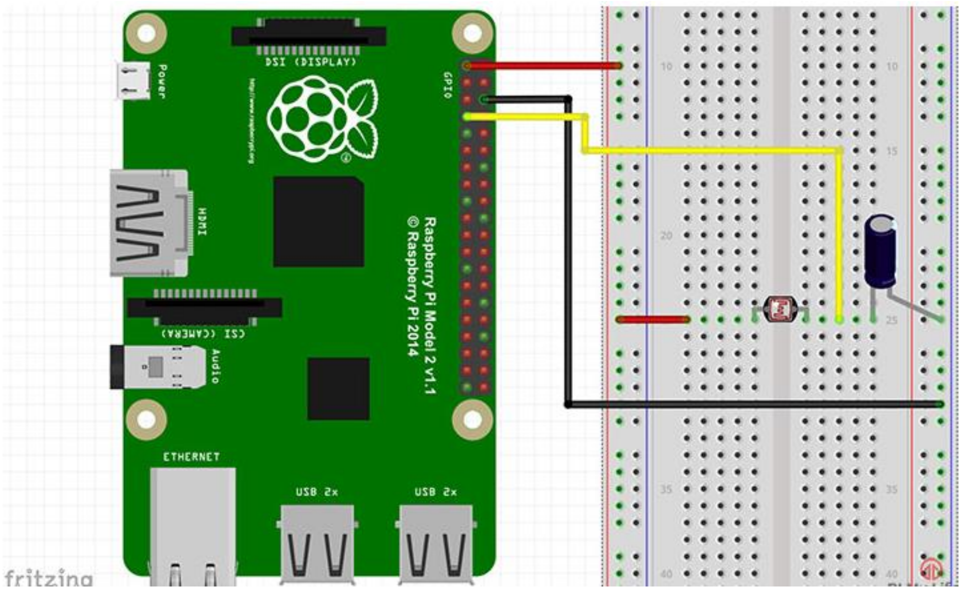
The light dependent resistor or also known as the LDR sensor is the most important piece of equipment in our circuit. Without it, we wouldn’t be able to detect whether it is dark or light. In the light, this sensor will have a resistance of only a few hundred ohms while in the dark, it can have a resistance of several megohms.

The capacitor in our circuit is there, so we’re able to measure the resistance of the LDR sensor. A capacitor essentially acts like a battery, charging up while receiving power and then discharging when no longer receiving power. Using this in series with the LDR, we can work out how much resistance the LDR is giving out and thus whether it is light or dark.

To get the light sensor circuit built correctly, follow the steps below or check out the circuit diagram right underneath the steps. Following steps are referring to the physical numbers of the pins (Logical order).

1. First, connect pin #1 (3v3) to the positive rail on the breadboard
2. Next, connect pin #6 (ground) to the ground rail on the breadboard
3. Now place the LDR sensor onto the board and have a wire go from one end to the positive rail
4. On the other side of the LDR sensor place a wire leading back to the Raspberry Pi. Hook this to pin #7
5. Finally, place the capacitor from the wire to the negative rail on the breadboard. Make sure you have the negative pin of the capacitor in the negative rail

We’re now ready to move onto the Python code. If you have any trouble with the circuit, refer to the diagram below.

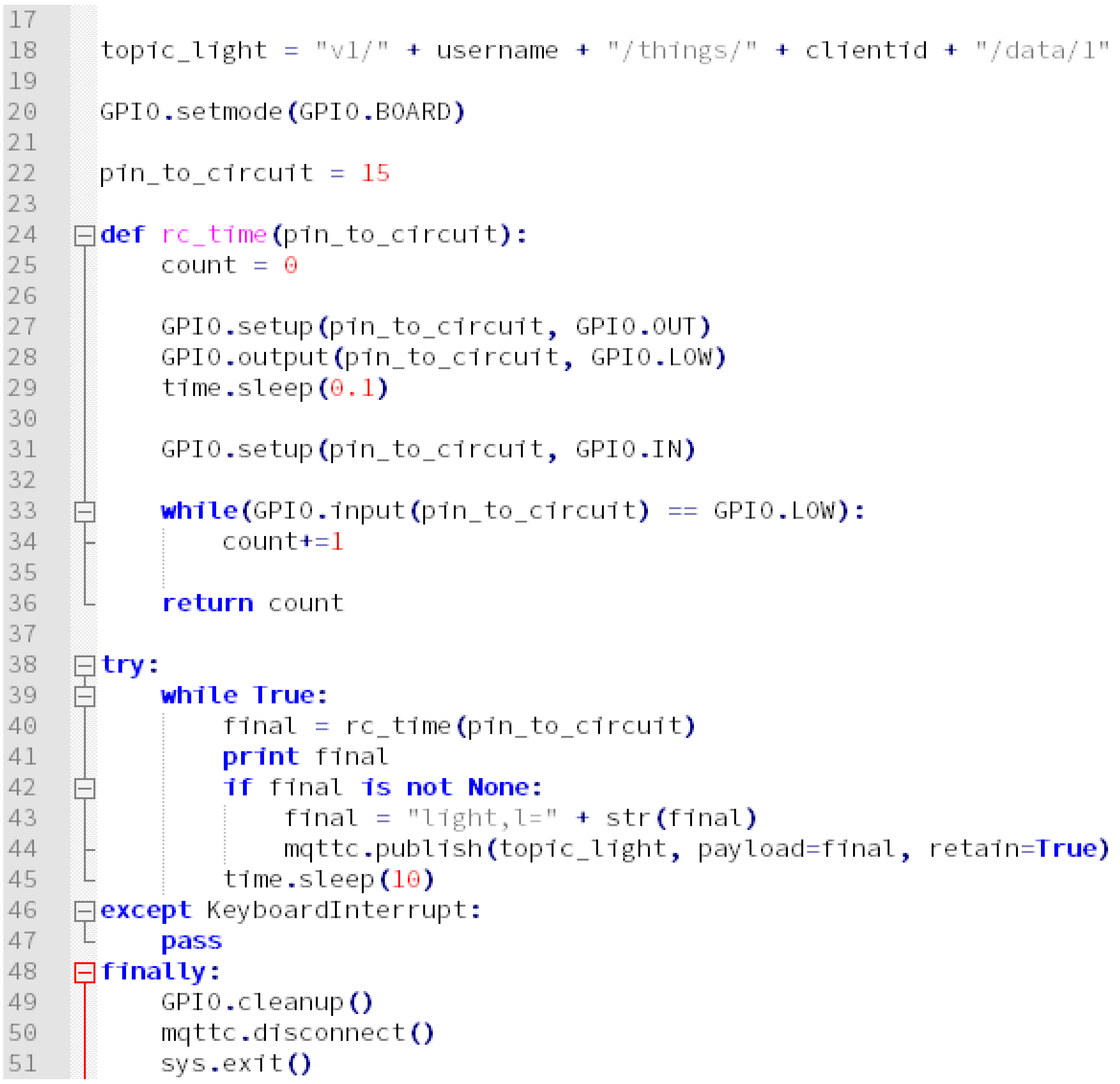


7.2.1 Python code

The code for this project is writen to roughly determine whether it is light, shady or completely dark.

The biggest problem is the fact that the Pi doesn't have any analogue pins. They're all digital, so it is not possible to accurately measure the variance in resistance on our input. This lack of analogue pins wasn't a problem with the sensors that require 2 states but for analog values we will measure the time it takes for the capacitor to charge and send the pin high.

```python
#!/usr/local/bin/python

import RPi.GPIO as GPIO
import time
import paho.mqtt.client as mqtt
import sys
time.sleep(10)

username = "388abe40-1594-11e8-b42d-698bd45831f1"
password = "b65f939f17428dda5623db5ab79f00f9295794ef"
clientid = "e0306ef0-fc94-11e8-a056-c5cffe7f75f9"

mqttc = mqtt.Client(client_id=clientid)
mqttc.username_pw_set(username, password=password)
mqttc.connect("mqtt.mydevices.com", port=1883, keepalive=60)
mqttc.loop_start()
```



```python

topic_light = "v1/" + username + "/things/" + clientid + "/data/1"

GPIO.setmode(GPIO.BOARD)

pin_to_circuit = 15

def rc_time(pin_to_circuit):
    count = 0

    GPIO.setup(pin_to_circuit, GPIO.OUT)
    GPIO.output(pin_to_circuit, GPIO.LOW)
    time.sleep(0.1)

    GPIO.setup(pin_to_circuit, GPIO.IN)

    while(GPIO.input(pin_to_circuit) == GPIO.LOW):
        count+=1

    return count

try:
    while True:
        final = rc_time(pin_to_circuit)
        print final
        if final is not None:
            final = "light,l=" + str(final)
            mqttc.publish(topic_light, payload=final, retain=True)
        time.sleep(10)
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
    mqttc.disconnect()
    sys.exit()
```

To begin, we import the GPIO package that we will need in order to communicate with the GPIO pins. Time pacage also has to be imported, so we're able to put the script to sleep for when we need to. Setting GPIO mode to GPIO.BOARD means that all the numbering used in this script will refer to the physical numbering of the pins. For the connected device we have to write down the ussername, password and client from the Cayenne Dashboard (Cayenne API - Bring Your Own Thing).

Since we only have one input/output pin, we only need to set one variable. Set this variable to the number of the pin you have acting as the input/output pin (pin 15). Next, we have a function called rc_time that requires one parameter which is the pin number to the circuit. In this function we initialize a variable called count. We will return this value once the pin goes to high. We then set our pin to act as an output and then set it to low. We then have the script sleep for 10ms.

After this, we set the pin to become the input and then we enter a while loop. We stay in this loop until the pin goes to high. This is when the capacitor charges to about 3/4. Once it goes high, we return the count value to the main function. You can use this value to turn on and off an LED, activate something else or log the data and keep statistics on any variance in light.

The code can be written or downloaded from Github.com:

```
git clone https://github.com/pimylifeup/Light_Sensor/
cd ./Light_Sensor
```

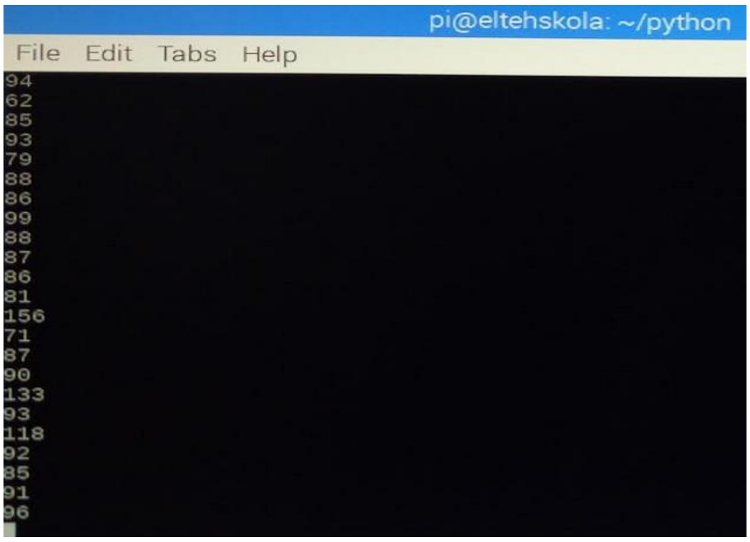
Alternatively, you can copy and paste the code. Just make sure the file is a **python script**.

```
sudo nano light_sensor.py
```

Once you're done in the file, simply use ctrl x then y to save and exit.

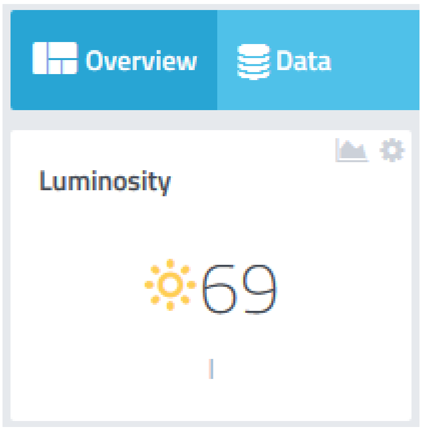
Finally, run the code by using the following command:

```
sudo python light_sensor.py
```



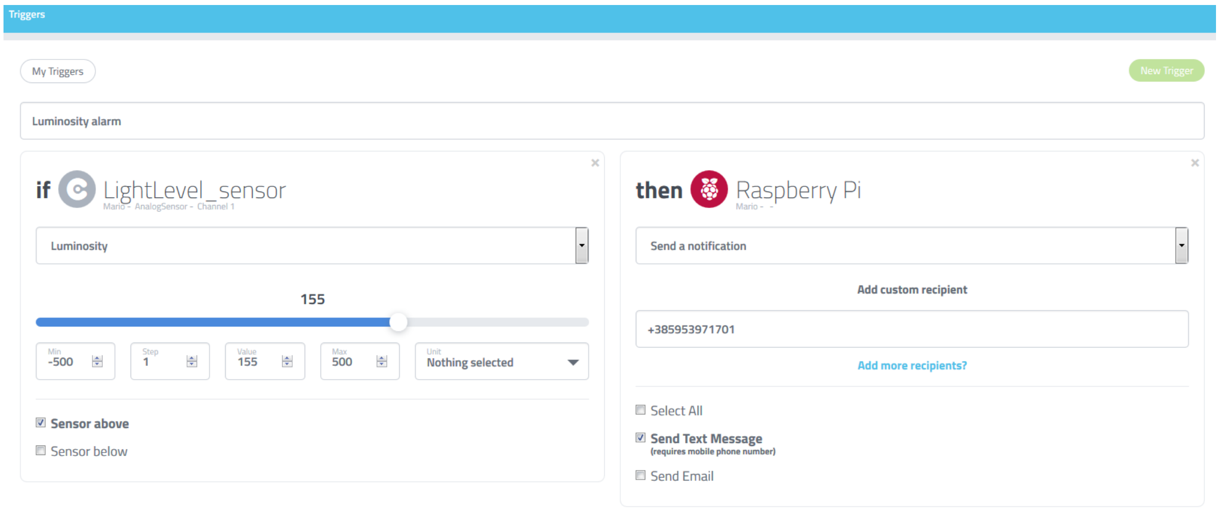
```
94
62
85
93
79
88
86
99
88
87
86
81
156
71
87
90
133
93
118
92
85
91
96
```

After a few seconds, the icon should appear on the Cayenne Dashboard, too, with the light level indication.



The light level indication is between 0 (high britness) and 5000 (dark). The information about light level can be used for many projects, such as a light activated alarm.

Light Activated Alarm – using light dependent resistor (LDR) to detect when it starts to get light so you can sound an alarm to wake up. The alarm can slowly get louder as it gets lighter.



Triggers

My Triggers

New Trigger

Luminosity alarm

if LightLevel_sensor
Mario - AnalogSensor - Channel 1

Luminosity

155

Min -500 | Step 1 | Value 155 | Max 500 | Unit Nothing selected

☑ Sensor above

☐ Sensor below

then Raspberry Pi
Mario - -

Send a notification

Add custom recipient

+385953971701

Add more recipients?

☐ Select All

☑ Send Text Message
(requires mobile phone number)

☐ Send Email

7.3 Conecting a CO2 sensor

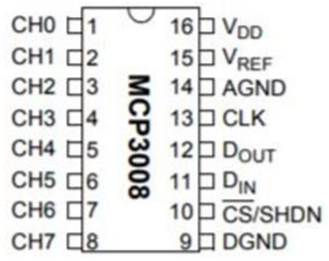
All MQ-X sensors return analogue signals which cannot be easily read with the Raspberry Pi. One possibility would be to use an Arduino or analog-to-digital converter (ADC) for Raspberry Pi, which can be read out via the I2C bus. In addition, a logic level converter for connecting 3.3V to 5V is also needed. For the realization of the project with the Raspberry Pi platform, we will need the following equipent:

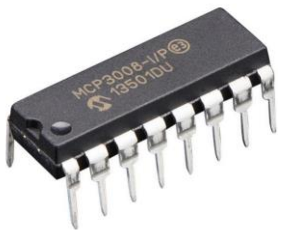
- Analog-Digital Converter (8 Ports) - MCP3008
- Logic Level Converter 5V to 3.3V
- Co2 sensor
- Jumper wire

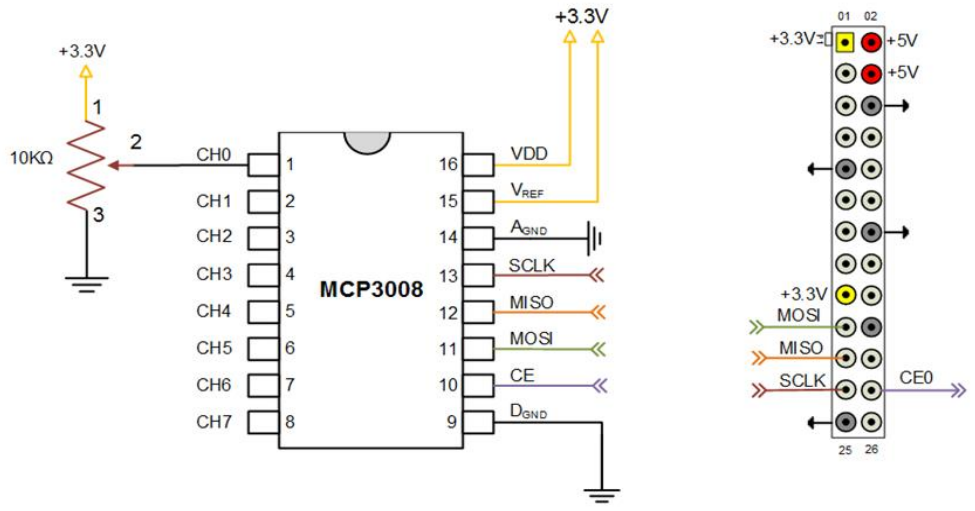
▪ *Analog-Digital Converter (8 Ports) - MCP3008*

The MCP3008 10-bit Analog-to-Digital Converter (ADC) combines high performance and low power consumption in a small package, making it ideal for embedded control applications. The MCP3008 features a successive approximation register (SAR) architecture and an industry-standard SPI serial interface, allowing 10-bit ADC capability to be added to any PIC® microcontroller.

The MCP3008 features 200k samples/second, 8 input channels, low power consumption (5nA typical standby, 425µA typical active), and is available in 16-pin PDIP and SOIC packages. Applications for the MCP3008 include data acquisition, instrumentation and measurement, multi-channel data loggers, industrial PCs, motor control, robotics, industrial automation, smart sensors, portable instrumentation and home medical appliances.







▪ *MCP3008 Pin specifications*

- Digital Ground (DGND):

Digital ground connection to internal digital circuitry.

- Analog Ground (AGND):

Analog ground connection to internal analog circuitry.

- Analog inputs (CH0 – CH7):

Analog inputs for channels 0 – 7, respectively, for the multiplexed inputs. Each pair of channels can be programmed to be used as two independent channels in single-ended mode or as a single pseudo-differential input where one channel is IN+ and one channel is IN.

- Serial Clock (CLK):

The SPI clock pin is used to initiate a conversion and clock out each bit of the conversion as it takes place.

- Serial Data Input (DIN):

The SPI port serial data input pin is used to load channel configuration data into the device.

Serial Data Output (DOUT):

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

- **Chip Select/Shutdown (CS/SHDN):**

The CS/SHDN pin is used to initiate communication with the device when pulled low. When pulled high, it will end a conversion and put the device in low-power standby. The CS/SHDN pin must be pulled high between conversions.

- **Analog Inputs:**

The MCP3008 devices offer the choice of using the analog input channels configured as single-ended inputs or pseudo-differential pairs. The MCP3008 can be configured to provide four pseudo-differential input pairs or eight single-ended inputs. Configuration is done as part of the serial command before each conversion begins.

When used in the pseudodifferential mode, each channel pair (i.e., CH0 and CH1, CH2 and CH3 etc.) are programmed as the IN+ and IN- inputs as part of the command string transmitted to the device. The IN+ input can range from IN- to (VREF + IN-). The IN- input is limited to ±100 mV from the VSS rail. The IN- input can be used to cancel small signal common-mode noise, which is present on both the IN+ and IN- inputs. When operating in the pseudo-differential mode, if the voltage level of IN+ is equal to or less than IN-, the resultant code will be 000h.

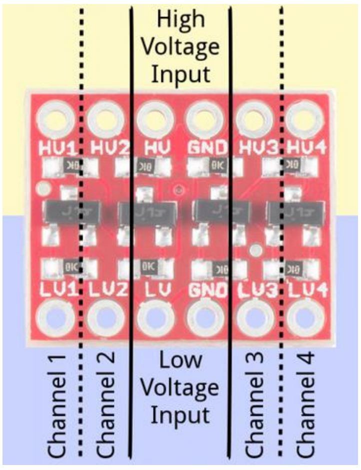
If the voltage at IN+ is equal to or greater than {[VREF + (IN-)] – 1 LSB}, then the output code will be 3FFh. If the voltage level at IN is more than 1 LSB below VSS, the voltage level at the IN+ input will have to go below VSS to see the 000h output code. Conversely, if IN- is more than 1 LSB above VSS, the 3FFh code will not be seen unless the IN+ input level goes above VREF level. For the A/D converter to meet specification, the charge holding capacitor (CSAMPLE) must be given enough time to acquire a 10-bit accurate voltage level during the 1.5 clock cycle sampling period.

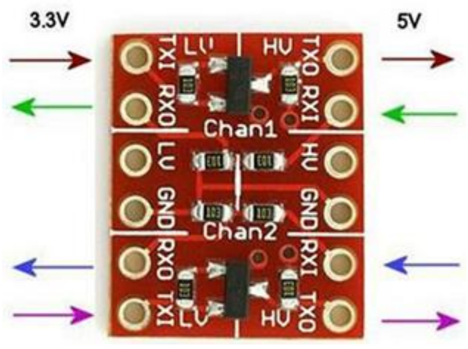
▪ *Logic Level Converter 5V to 3.3V*

The bi-directional Level Converter is used to interconnect two sections of an I2C-bus system, each section with a different supply voltage and different logic levels. The devices of each section have I/O’s with supply voltage related logic

input levels and an open drain output configuration.

There are 12 total pins on the Logic Level Converter with two parallel rows of six headers. One row contains all of the high voltage (e.g. 5V) inputs and outputs, the other row contains of low voltage pins (e.g. 3.3V).





The pins are labeled on both the bottom and top sides of the board and organized into groups. There are Voltage inputs and data channel pin groups.

- **Voltage Inputs**

The pins labeled HV, LV and two GND's provide high and low voltage references to the board. Supplying a steady, regulated voltage to both of these inputs is required.

The voltage supplied to the HV and GND inputs should be higher than that supplied to the LV side. For example, if you're interfacing from 5V to 3.3V, the voltage on the HV pin should be 5V and the voltage on LV sould be 3.3V.

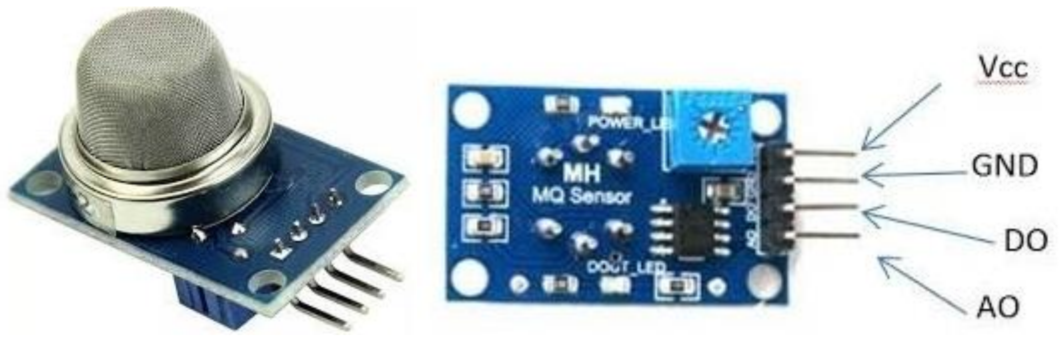
- **Data Channels**

There are four separate data channels on the BD-LLC, each capable of shifting data to and from high and low voltages. These pins are labeled HV1, LV1, HV2, LV2, HV3, LV3, HV4, and LV4. The number at the end of each label designates the channel of the pin and the HV or LV prefix determines whether it's on the high or low side of the channel.

A low-voltage signal sent in to LV1, for example, will be shifted up to the higher voltage and sent out HV1. Something sent in HV3 will be shifted down and sent out of LV3. These level shifters are purely digital. They can't map an analog voltage from one max voltage to another.

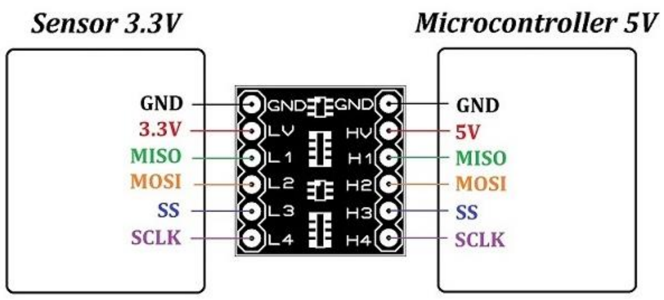
- ***CO2 sensor***

The MQ-x series of smoke detectors work well with Arduino and Raspberry Pi. This sensor, the MQ-7 FC-22, has analog and digital outputs. The digital output pin labeled d0 on the sensor provides warning if CO (i.e., smoke) is detected. Alternatively, the analog output could be used to measure the level of CO, but this will need an analog/digital converter like MCP3008. The smoke level which raises an alarm can be adjusted with the analog potentiometer on the backside of the smoke sensor (the tiny blue rectangle at the bottom of the picture).



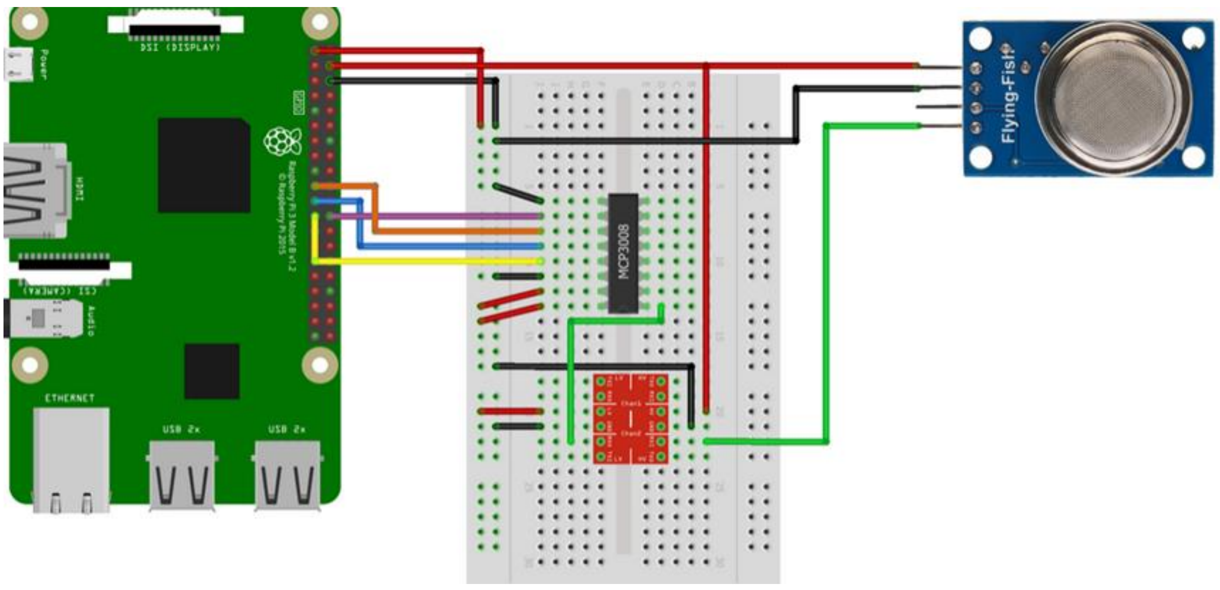
7.3.1 Connection between MQ-2 and Raspberry Pi

The output voltage needed for this project is 5V. Raspberry Pi GPIO (general purposes input/output pins) can take only 3.3V, which is why we have to use a logic level converter (TTL) that cuts down the voltage from 5V to 3.3V.



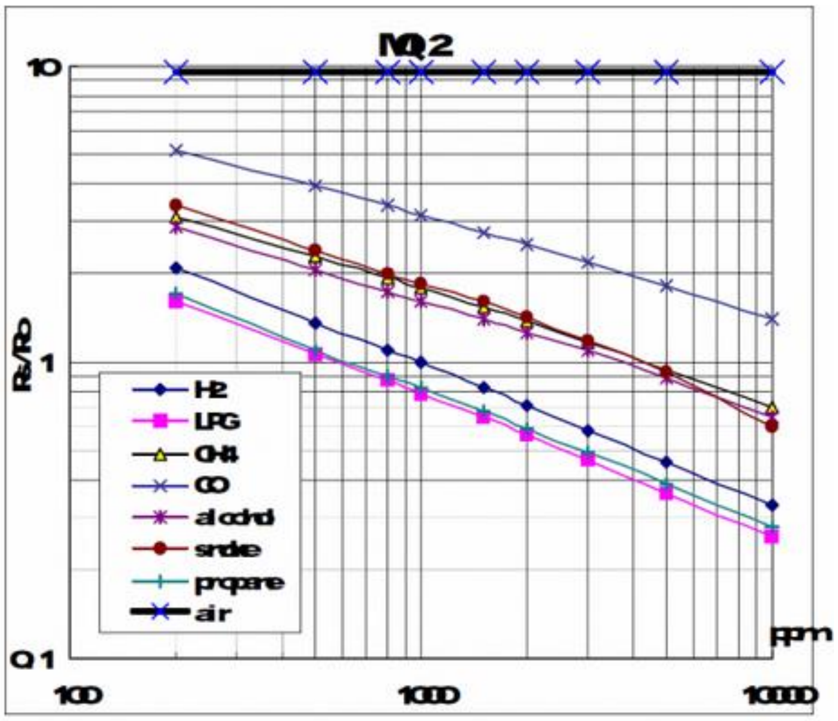
After the MCP3008 is correctly connected, we use port 0 and connect it to RX0 of the TTL. On the opposite side is RX1, which is connected to the analog pin (A0) of the MQ2 sensor. Also, connect 3.3V from the Raspberry Pi (LV) and 5V (HV) to the TTL, the 5V to the VCC pin of the gas sensor and GND from the Raspberry Pi comes to GND on the LV and HV side of the TTL, as well as to GND of the MQ2.

Schematically, connections between Raspberry Pi and Co2 gas sensor:



7.4 Configuration of the Raspberry Pi Gas Sensor

The concentration of a gas is given in PPM (parts per million). One difficulty of the MQ-2 is that a single analog value is given by which the gas content in the air has to be calculated for the various supported gases. However, the sensor must be configured for this purpose.



The scaling of the values is not linear but logarithmic to the base 10 (log). So, the first stroke on the X axis is 200, then 300, etc. The first stroke after 1000 is 2000, etc. The distance between is linear. The idea behind this script for calibration and reading is to create a straight line and calculate the amount of gas (in ppm). To do this, we need two points to calculate the slope.

We therefore take the point P1 (x = 200, y = ~ 1.62) and P2 (x = 10000, y = ~ 0.26). To calculate the "real" values, we apply the ten logarithm. Using the two-point form, we can calculate the slope, which in our case is -0.47 (link to the calculation). With the slope and the calculated logarithm from the left point (x = 2.3, y = 0.21), we can now determine the straight line.

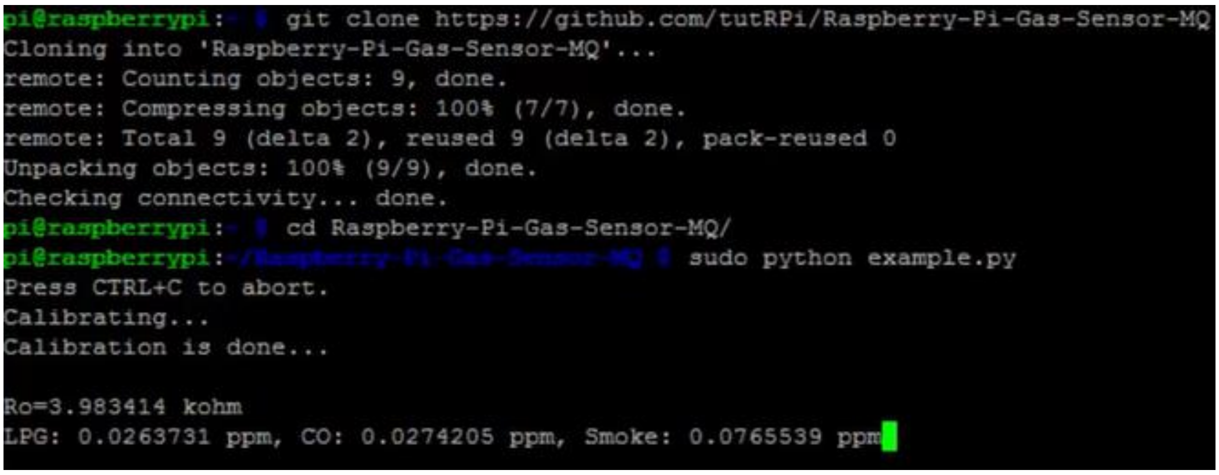
7.4.1 Calibration of the Raspberry Pi Gas Sensor – Code

The customized code can be found in a GitHub repository. A class is also included for reading the MCP3008. First, we have to clone the directory:

```
git clone https://github.com/tutRPi/Raspberry-Pi-Gas-Sensor-MQ
```

Then, change the path to the directory and run the existing Python test file.

```
cd Raspberry-Pi-Gas-Sensor-MQ
sudo python example.py
```



```
pi@raspberrypi:~ $ git clone https://github.com/tutRPi/Raspberry-Pi-Gas-Sensor-MQ
Cloning into 'Raspberry-Pi-Gas-Sensor-MQ'...
remote: Counting objects: 9, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 2), reused 9 (delta 2), pack-reused 0
Unpacking objects: 100% (9/9), done.
Checking connectivity... done.
pi@raspberrypi:~ $ cd Raspberry-Pi-Gas-Sensor-MQ/
pi@raspberrypi:~/Raspberry-Pi-Gas-Sensor-MQ $ sudo python example.py
Press CTRL+C to abort.
Calibrating...
Calibration is done...

Ro=3.983414 kohm
LPG: 0.0263731 ppm, CO: 0.0274205 ppm, Smoke: 0.0765539 ppm
```

The calibration is started automatically during initialization. It is important that the sensor is in good fresh air as other gases would falsify the calibration. The process takes a few seconds, but the gas content can already be measured. Some sensors are quite hot, but this should not be a cause for concern.