1. Introduction to Raspberry Pi 2

1.1 Raspberry Pi 2

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse (source: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/).

The Raspberry Pi 2 comes with a much more powerful processor (four Cortex -A7 cores with up to 900 MHz) and more memory (1GB RAM). The following is technical specification of Raspberry Pi 2 device:

- ARM 7 Quad Core CPU
- 1GB RAM
- 900MHz Board Clock Speed
- 40 GPIO Pins
- 4 x USB Ports
- 4 Pole Stereo Output
- 1x HDMI Port
- 1x 0/100 Ethernet
- 1x Micro SD Card slot

You can see Raspberry Pi 2 device with model B on the Figure below.

1.2 Getting Hardware

How to get Raspberry Pi 2 device?

Officially you can buy it from the official distributor

- RS, http://uk.rs-online.com/web/generalDisplay.html?id=raspberrypi
- Element14, http://www.element14.com/raspberrypi

You also buy Raspberry Pi peripheral devices for instance, keyboard, mouse, HDMI cable, SD card, USB hub, etc.

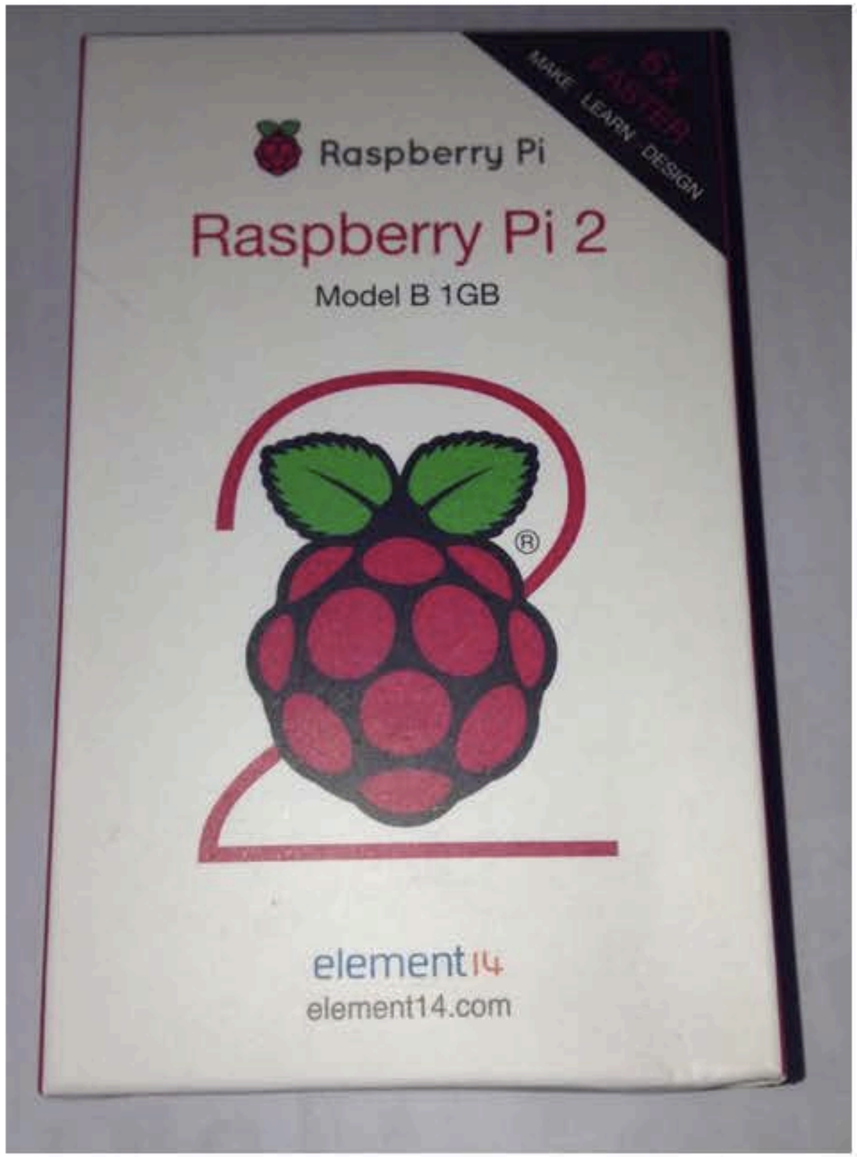
I tried to look for buying Raspberry Pi 2 device and found that there are another options to buy

- The Pi Hut, http://thepihut.com
- EXP-Tech, http://www.exp-tech.de/
- Cooking-hack, http://www.cooking-hacks.com/
- Amazon, http://www.amazon.com
- Ebay, http://www.ebay.com

You also can buy this board at your local electronics stores.

1.3 Unboxing

After bought Raspberry Pi 2, we get a small box as follows.



- Raspberry Pi 2 board
- Manual book

We open this box. We get the following items:


Raspberry Pi
QUICK START GUIDE
&
SAFETY INSTRUCTIONS
MANUAL
HDMI

2. Operating System

This chapter explains how to work with Operating System for Raspberry Pi 2.

2.1 Raspberry Pi 2 Operating System

Raspberry Pi provides some Operating Systems you can use and run on the top of Raspberry Pi. The following is the list of Raspbery Pi OS:

- Raspbian "wheezy"
- Arch Linux ARM
- Pidora
- RISC OS

If we have Raspberry Pi 2 board, we can use Snappy Ubuntu core and Windows 10 OS.

You can download these OS files on http://www.raspberrypi.org/downloads.

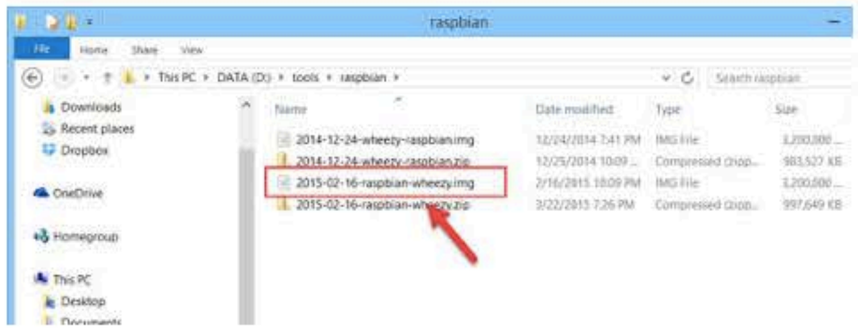
In this book, I use Raspbian "wheezy" OS for testing.

2.2 Preparation

Raspbian is an Operating system based on Debian Linux for the Raspberry Pi hardware. Officially you can download this OS image file on http://www.raspbian.org/RaspbianImages.

I recommend you to download OS image file on http://www.raspberrypi.org/downloads . For illustration, I use Raspbian OS.

After extracted this file, you will obtain *.img file, for instance, **2015-02-16-raspbian-wheezy.img** file.



2.2.1 Setup MicroSD Card

If we are working with Raspberry Pi 2 board, we need MicroSD card to extract this OS image file. I use MicroSD Card 8 GB.

Insert this card into your computer.

For Linux users:

You can mount it, for instance, /dev/sdd1

```
umount /dev/sdd1
```

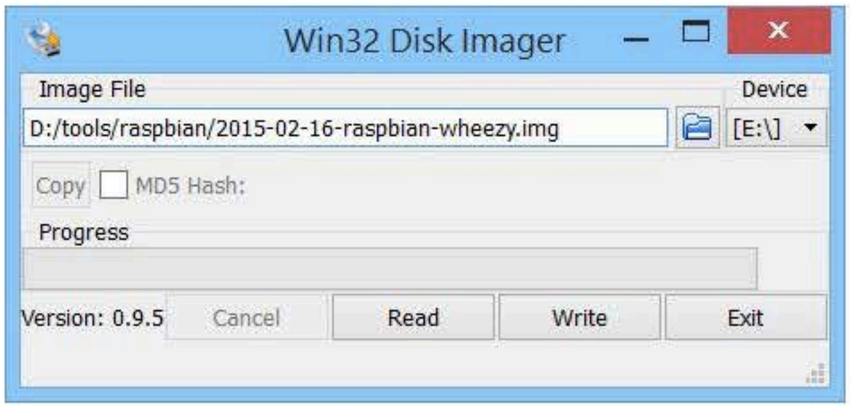
Then, you can copy all img file into MicroSD card.

```
dd bs=1M if=~/2015-02-16-raspbian-wheezy.img of=/dev/sdd1
```

For Windows users:

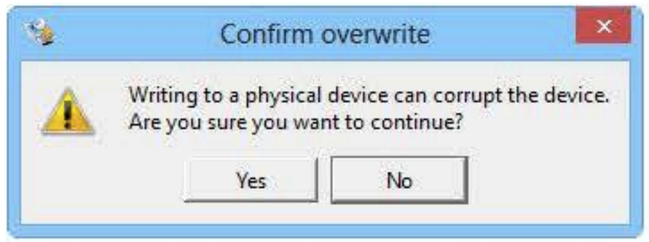
Download Win32DiskImager on https://launchpad.net/win32-image-writer/+download

Run Win32DiskImager and navigate Raspberry Pi image file.



Click **Write** button to start for copying files.

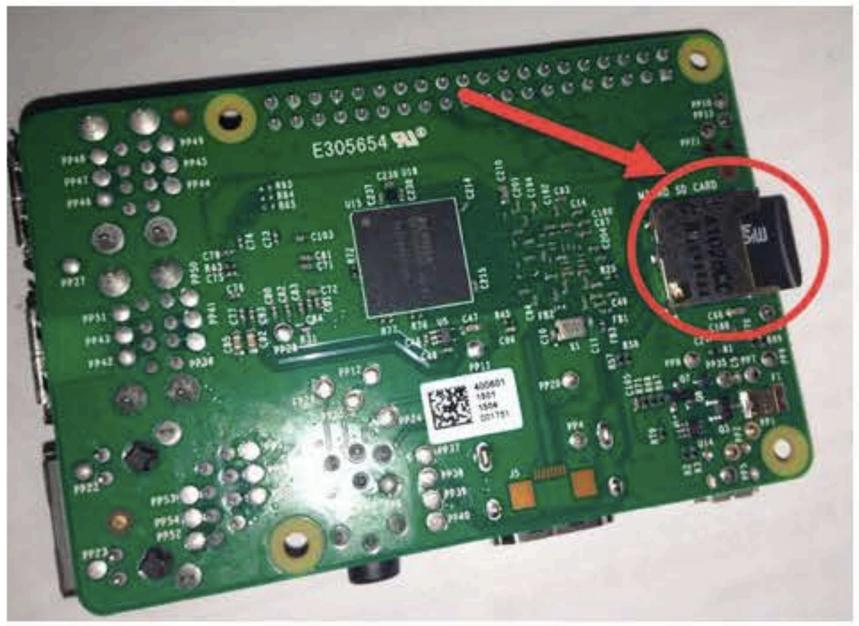
Click **Yes** to confirm overwrite files



Then, Win32DiskImager app will copy all files into Micro SD card.

If success, you can see all files in Micro SD card.

Plug out SD card from computer. Then, plug in it into Raspberry Pi



Now your Raspberry Pi 2 is ready to be deployed
OS.

3. Powering Up and Running

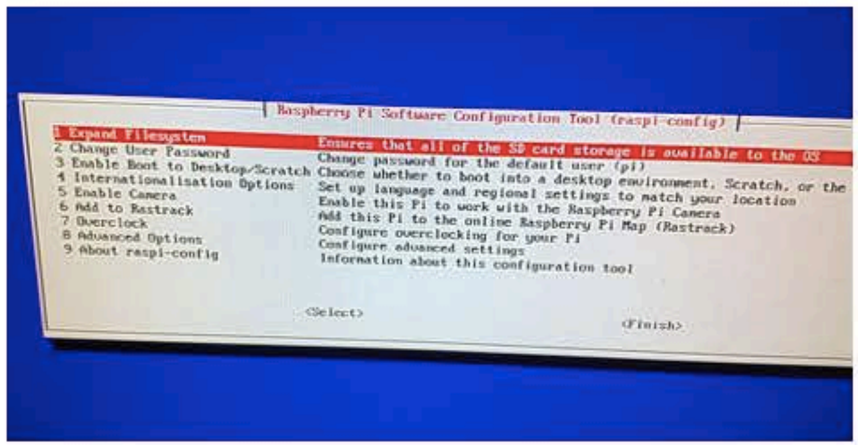
In this chapter we start to run and configure Raspberry Pi 2.

3.1 Put Them All!

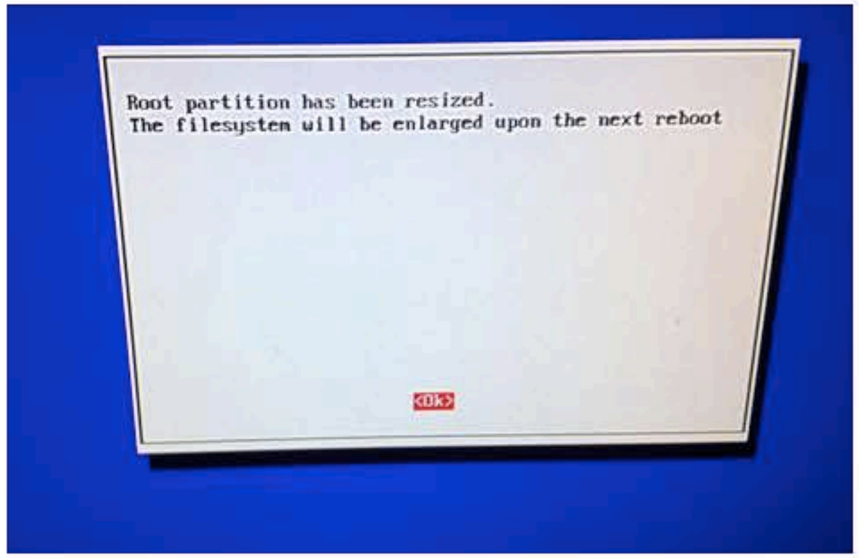
Now you are ready to boot your Raspberry Pi. Please plug in all devices, for instance, mouse, keyboard, power, and HDMI cable, into Raspberry Pi.

Turn on the power for your Raspberry Pi. Raspbian OS will boot for the first time.

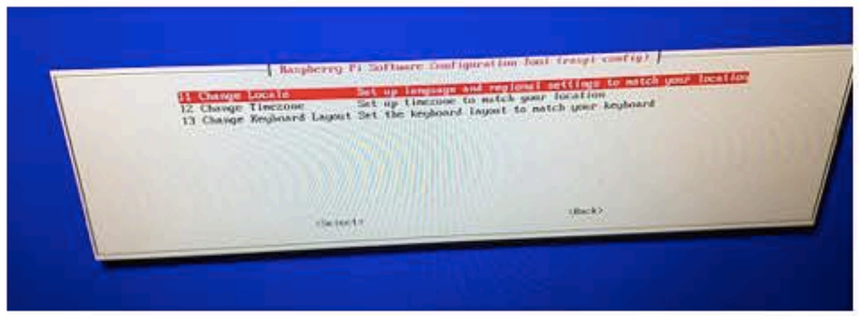
If success, you will get the first screen of Raspberry Pi as below



The first step we must deploy our File System. Select 1 Expand Filesystem . Then, it will do partition and configure. If done, you will get a notification.

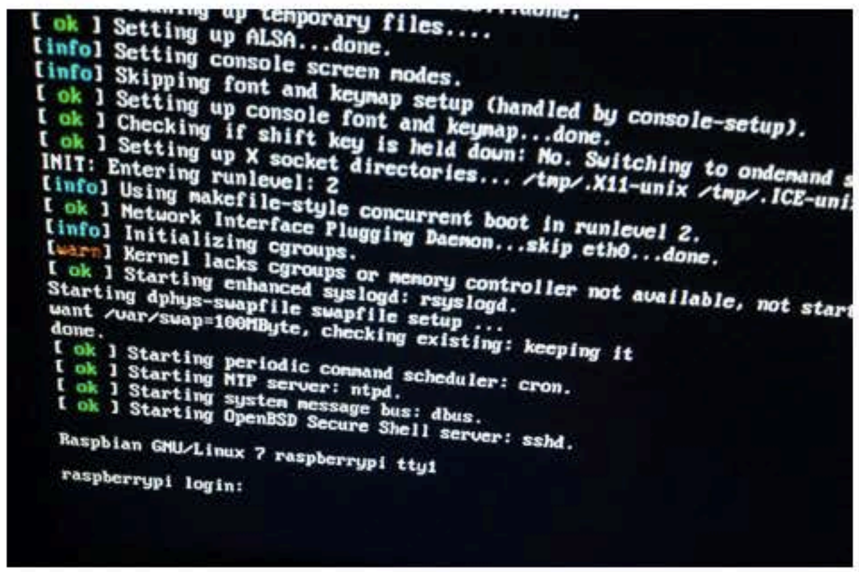


After that, you can configure any option, for instance, configure TimeZone.



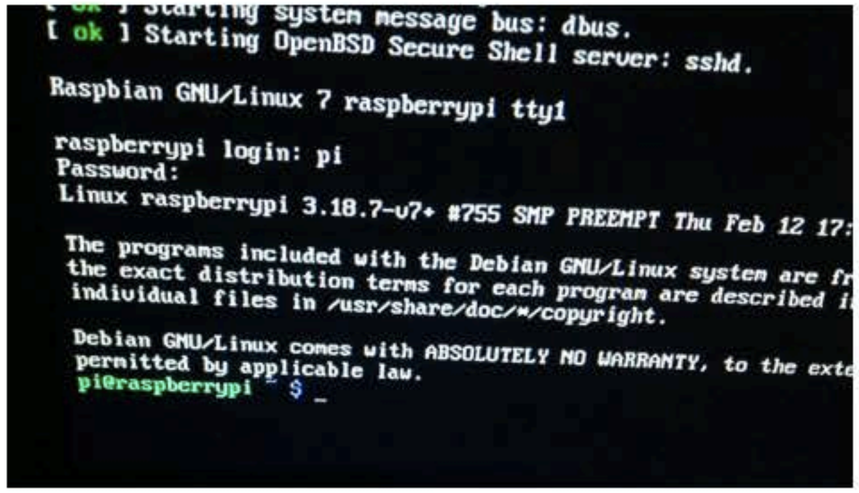
If done, click **Finish** menu.

Furthermore, Raspbian will restart. After restarted, we will get authentication form.



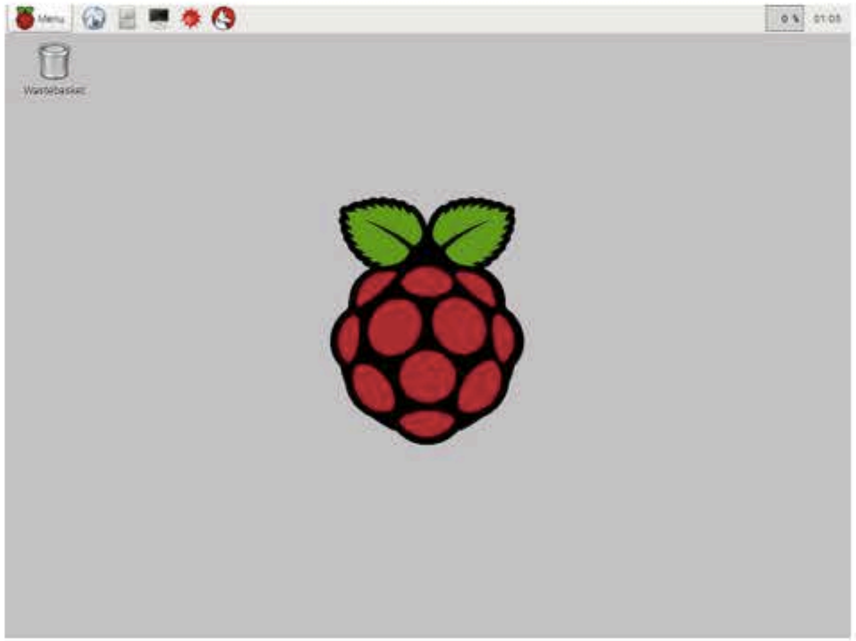
By default, Raspbian has user: **pi** and password: **raspberry**. Type this account.

If success, you will get Terminal on Raspbian.



To work with Desktop GUI, you can type

```
startx
```



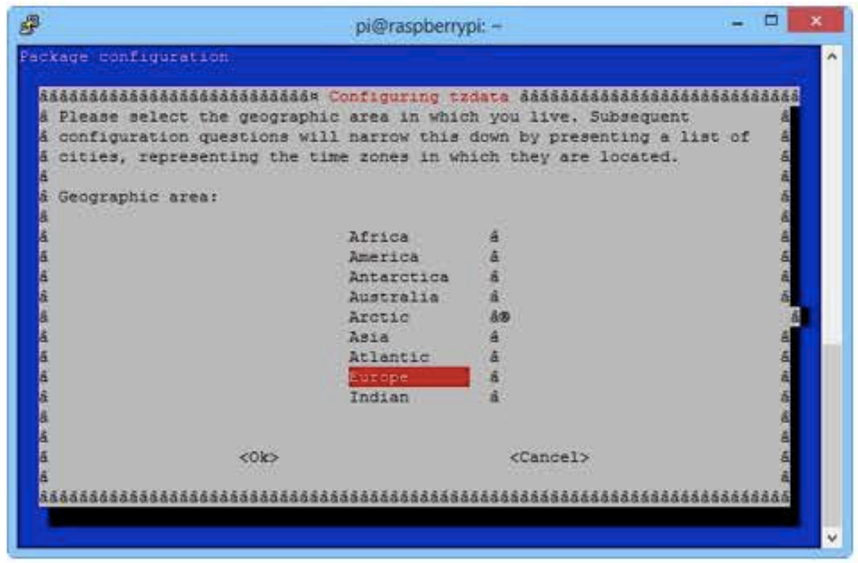
On desktop mode, if you want to work with Terminal, you can click black monitor icon, shown in Figure below.

3.2 Configure Timezone

If you want to change timezone in Raspberry Pi, you can do it in console with typing

```
sudo dpkg-reconfigure tzdata
```

Then, you will get a dialog as below



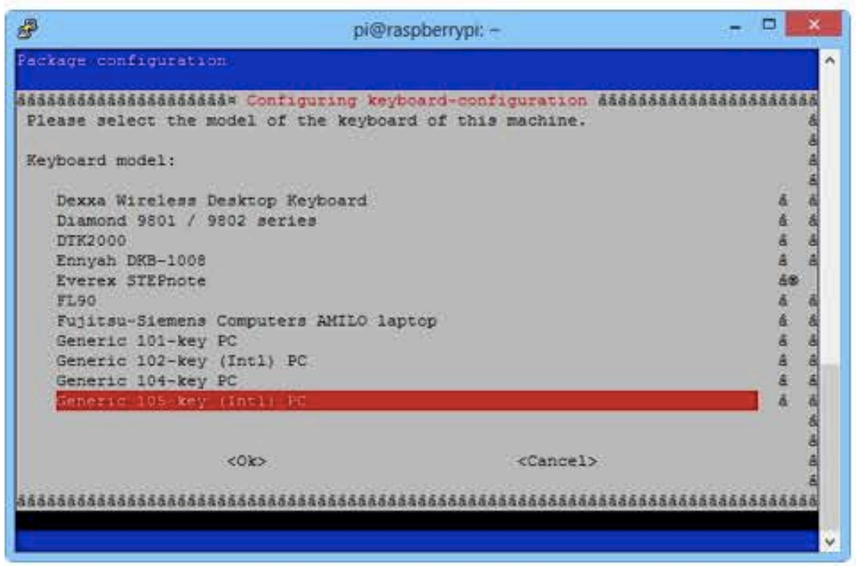
Choose your timezone.

3.3 Configure Keyboard

You may change your keyboard configuration. You can use dpkg-reconfigure command. Write this script

```
sudo dpkg-reconfigure keyboard-
configuration
```

Then you will get a dialog as below



choose your keyboard type and model.

3.4 Rebooting

If you want to reboot your Raspberry Pi, write this script

```
sudo shutdown -r now
```

You also can do it with writing this script

```
sudo reboot
```

3.5 Shutdown

It's better to shutdown your Raspberry Pi If you don't use it. Please don't turn off the power directly.

Write this script to shutdown and turn off your Raspberry Pi

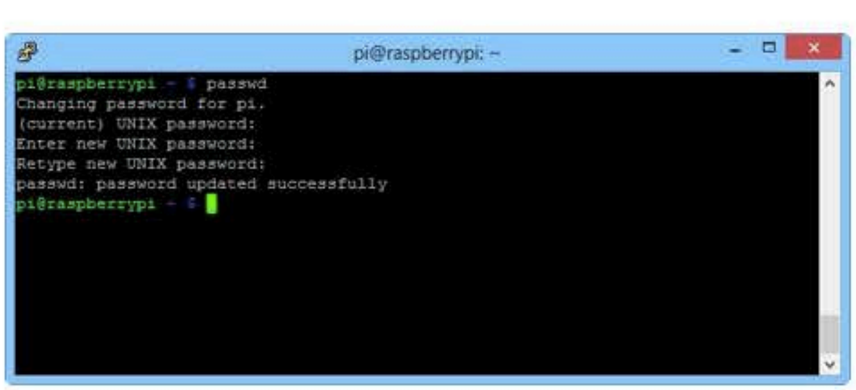
```
$ sudo shutdown -h -P now
```

3.6 Change Password

By default, Raspbian provides username: pi and password: raspberry. If you want to change password, you can do it by calling passwd

```
$ passwd
```

Then type the current password and new password



3.7 Configure All Settings

We can configure all settings on Raspbian by typing the following command.

```
$ raspi-config
```

Configure what you want to edit.

Further information about this command, you can read it on https://www.raspberrypi.org/documentation/configuration/raspi-config.md .

4. Connecting to a Network

This chapter explains how to work with networking in Raspberry Pi 2.

4.1 Connecting to Network

4.1.1 LAN

Raspberry Pi can connect to LAN easily. Just plug UTP cable into Raspberry Pi 2 board. Raspbian, by default, uses DHCP client to configure IP Address.

4.1.2 WIFI

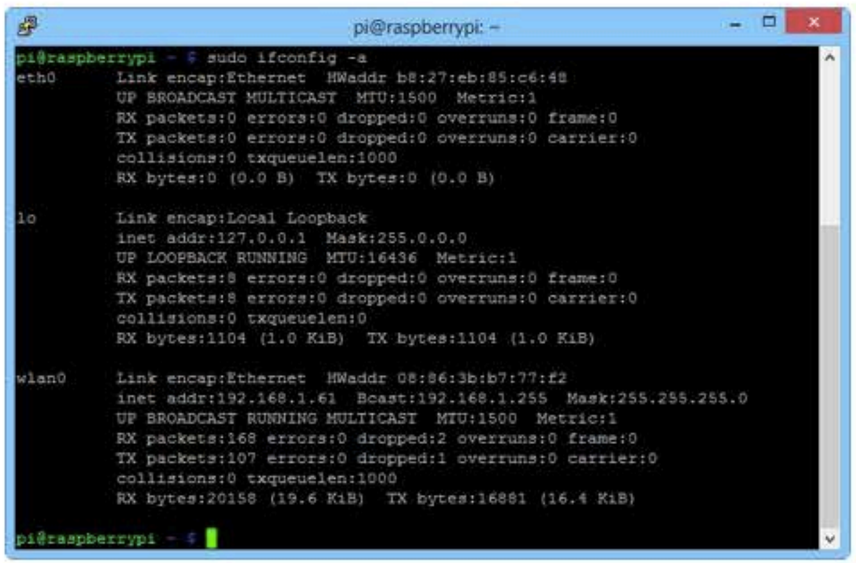
You also can connect your Raspberry Pi with Wifi USB. You can buy it from Raspberry Pi distributor. I also found a website that provides information WIFI dongles for Raspberry Pi. You can obtain it on electronics stores.

I use WIFI dongle from Belkin. My Raspberry Pi detected this WIFI adapter.

4.2 Configuring IP Address

You can check your current IP Address by writing this script

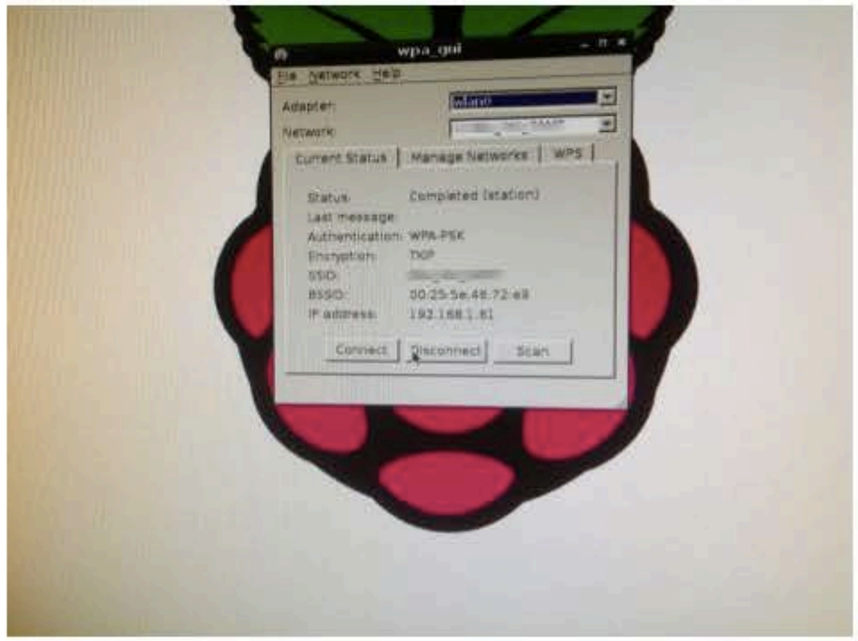
```
$ sudo ifconfig -a
```



You may want to install wireless-tool to configure wireless network.

```
$ sudo apt-get install wireless
-tools
```

You can see your IP Adress in WIFI GUI from Raspberry Pi desktop



4.3 Static IP Address

By default, Raspberry Pi configures IP address in DHCP mode. If you want to change to static IP Address, you edit interfaces file.

Type these commands on Terminal.

```
$ cd /etc/network
```

```
$ sudo nano interfaces
```

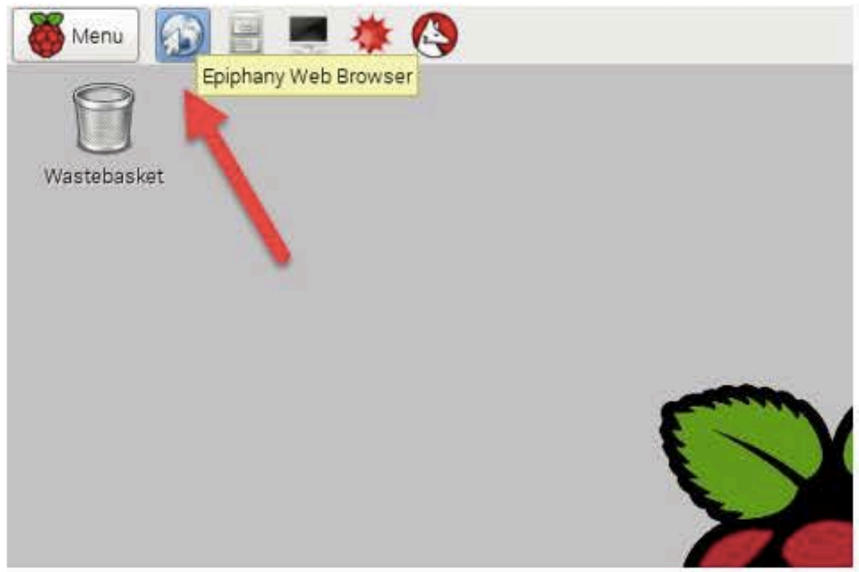
Then, you will see a content of file interface. Replace **iface etho inet dhcp** with

```
iface eth0 inet static
address 192.168.1.10
netmask 255.255.255.0
gateway 192.168.1.1
```

After that, you can verify your current IP Address now. You may reboot your Raspberry Pi.

4.4 Browsing Internet

If your Raspberry Pi already connected to Internet, you can browse the Internet. Raspberry Pi provides **Epiphany** as browser. Click its icon, shown in Figure below.



Furthermore, a browser is opened. Navigate to a specific URL. If success, it show the target URL. A sample output of browser can be seen in Figure below.


```
$ sudo /etc/init.d/ssh start
```

Sometimes, you want to run SSH service every booting. Try to update update-rc.d file

```
$ sudo update-rc.d ssh defaults
```

and then reboot your Raspberry Pi

```
$ sudo reboot
```

4.5 SSH

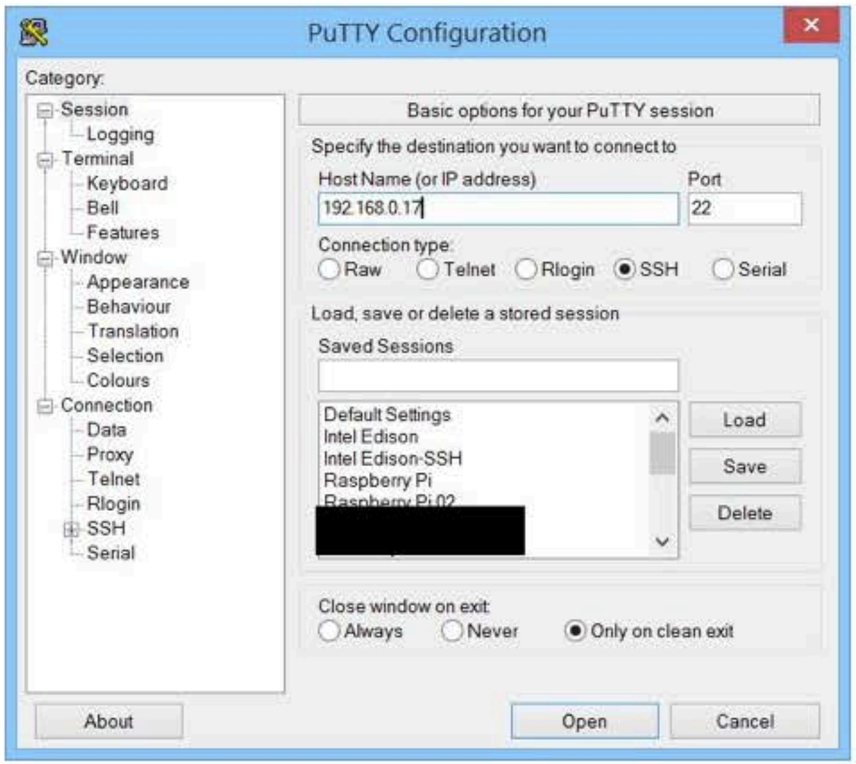
If you use Raspbian with February 2015 version, SSH have installed on OS image. If you don't install yet SSH in Raspberry Pi, you can write his script

```
$ sudo apt-get install ssh
```

To start a service, try to write this script

For testing, I used PuTTY application in Windows 8 to remote Raspberry Pi via SSH.

Fill IP Address of Raspberry Pi.



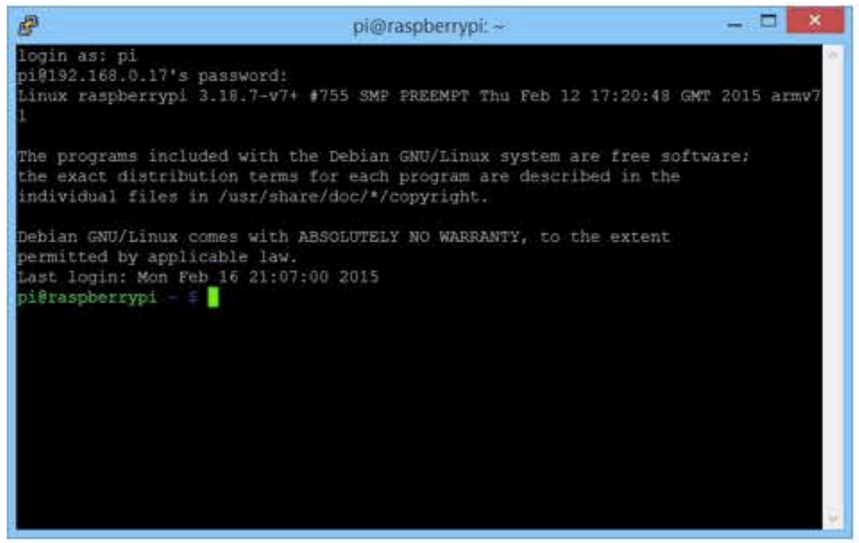
You can get IP address your Raspberry Pi 2 board by checking it on your router. For instance, my router detected my board MAC.

You also can fill Raspberry Pi hostname. By default, the Pi hostname is **raspberrypi**.

Then, click **Open** button. If connected, you will get a security alert.

Click **Yes** button.

Entry username and password. If success, you will get Raspberry Pi console.

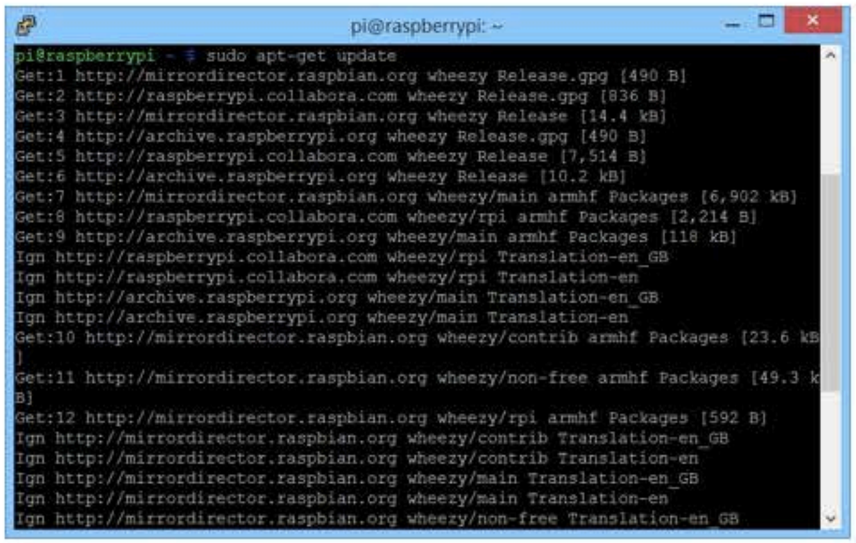


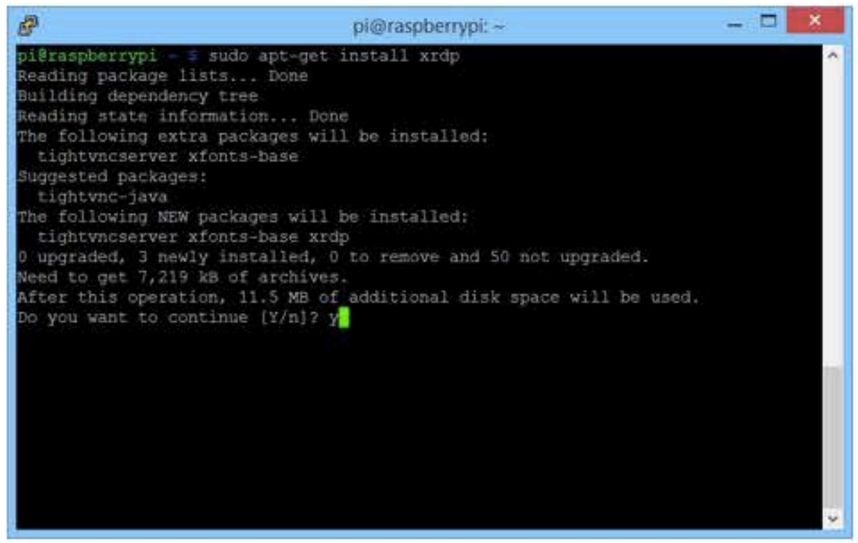
4.6 Update Package Repository

If you want to update your package repository, you can execute by writing this script

```
$ sudo apt-get update
```

Note: It needs the Internet connection.





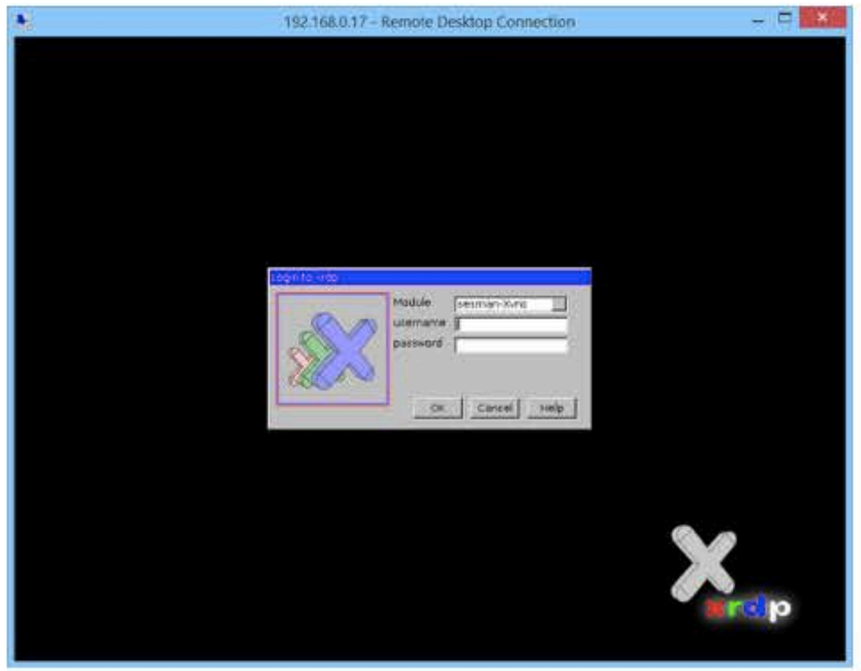
4.7 Remote Desktop

We can remote our Raspbian desktop using remote desktop (RDP). By default, it's not be installed yet so you can install **xrdp**. Type this command on Terminal.

```
$ sudo apt-get install xrdp
```

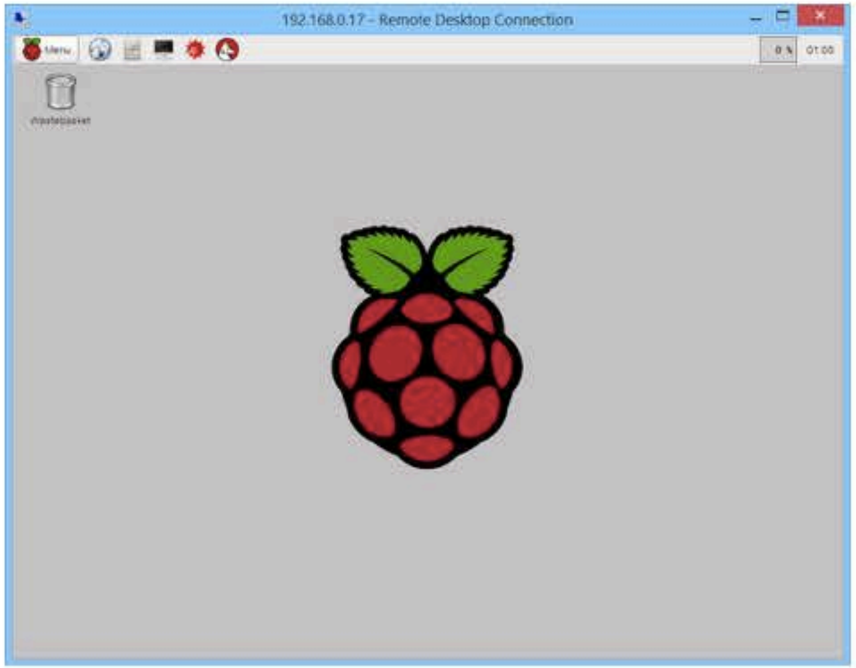
To test, I use Remote Desktop on Windows 8.1. Fill IP address of Raspberry Pi 2.

Click **Connect** button. If you will get a warning dialog. Click **Yes** button.



If you success, you will get xrdp dialog. Fill Raspberry Pi account.

If done, you will get Raspbian desktop.


192.168.0.17 - Remote Desktop Connection
Menu
01:00

5. Deploying LAMP Stack

This chapter explains how to deploy LAMP stack on Raspberry Pi 2 board.

5.1 Getting Started

In this section, we try to deploy LAMP on our Raspberry Pi. The following is a list of required component which must be installed:

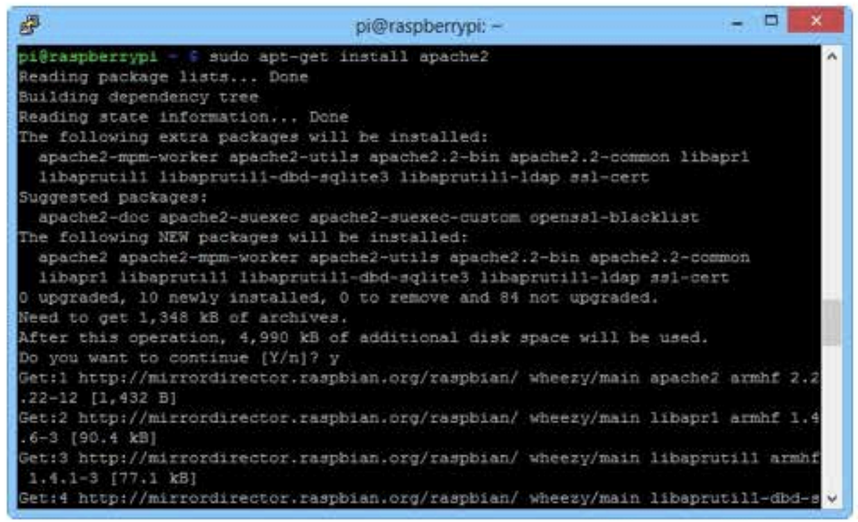
- Web Server, Apache
- Database, MySQL
- PHP
- MySQL Database driver for PHP

We will install these components on next section.

5.2 Installing Apache Server

Firstly, we install Apache Server.

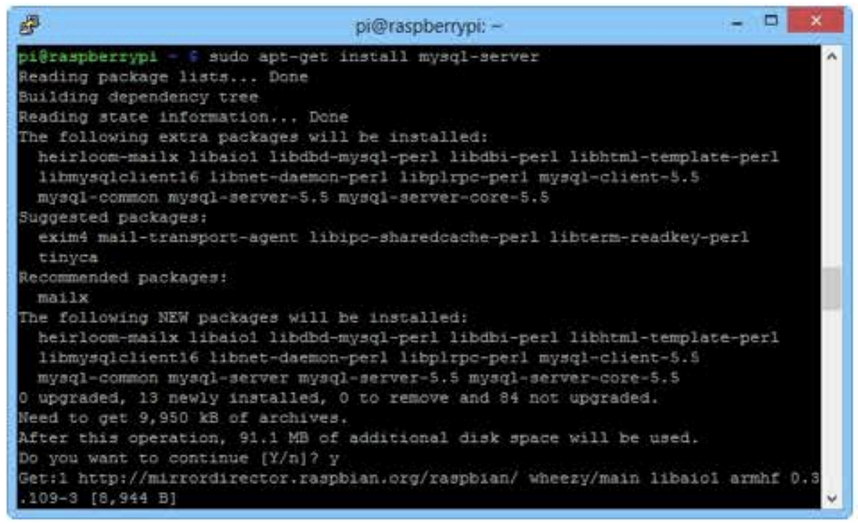
```
$ sudo apt-get install apache2
```



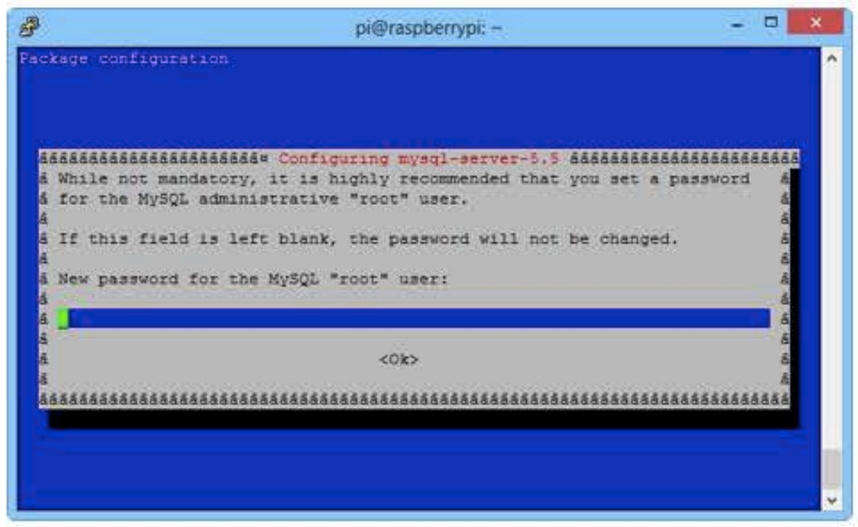
5.3 Installing MySQL

The second step is to install MySQL. Execute this command

```
$ sudo apt-get install mysql-se
rver
```



In the middle of installing process, you will be asked to fill root password for MySQL



If installation process is done, you can verify your MySQL by executing this command

```
$ mysql --version
```

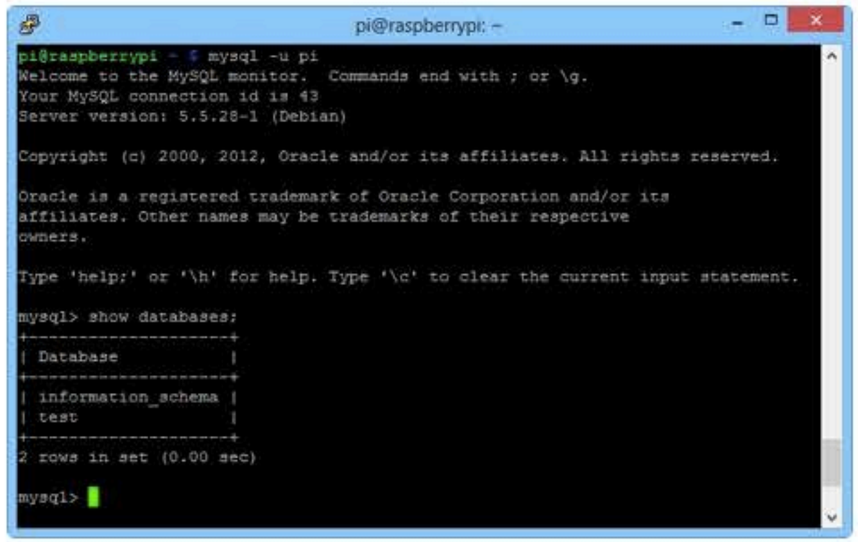

Now you can connect to MySQL server. Execute this command

```
$ mysql -u pi
```

Note: you may change MySQL user.

Type this command to retrieve databases in MySQL.

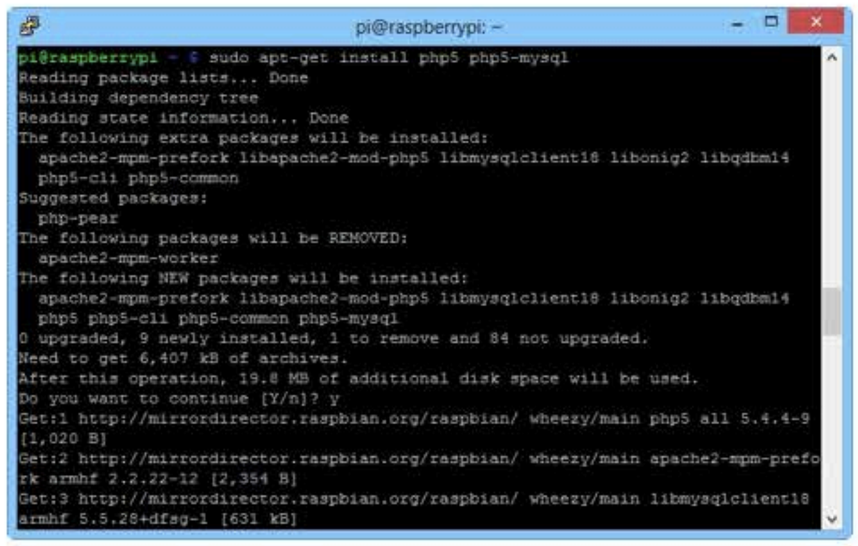
```
mysql> show databases;
```



5.4 Installing PHP and MySQL Driver for PHP

You need MySQL driver for PHP to access MySQL database. Execute this command to install PHP-MySQL driver

```
$ sudo apt-get install php5 php
5-mysql
```



5.5 Testing PHP

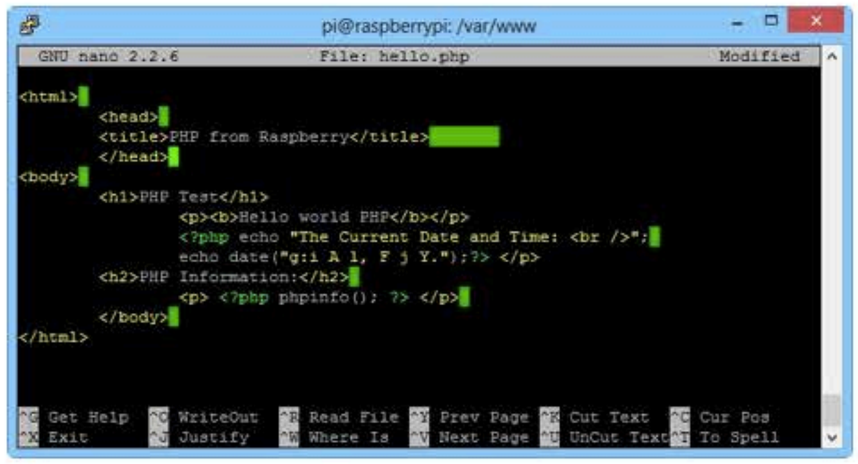
For testing, we are going to write "Hello world" PHP. Create a file, called **hello.php**, in /var/www/

```
$ sudo nano /var/www/hello.php
```

Then, write this code.

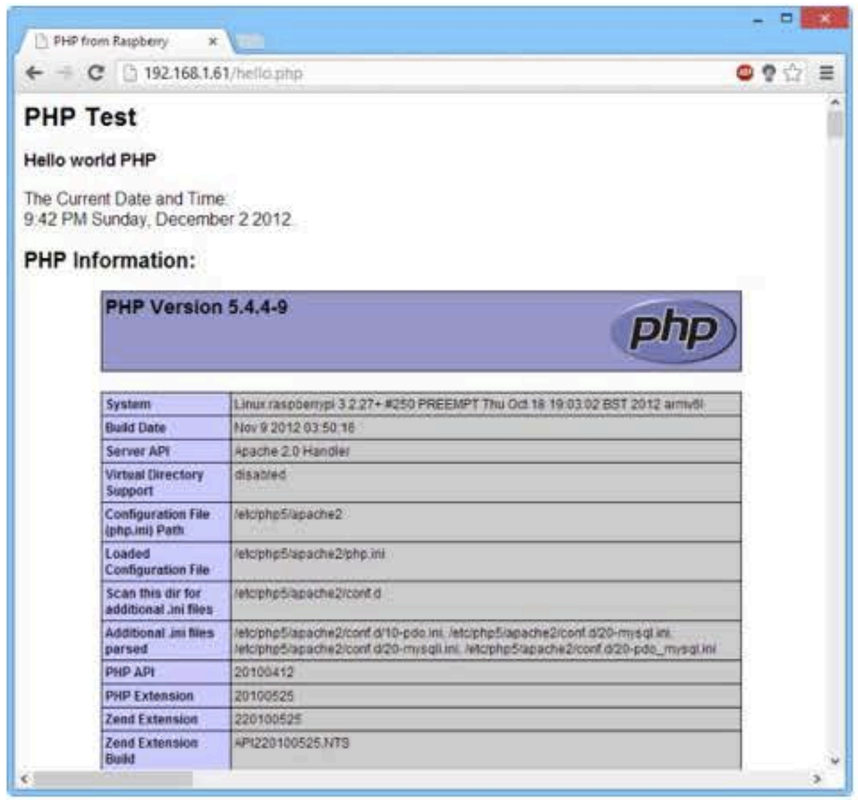
```
<html>
        <head>
```

```
        <title>PHP from Raspber
ry</title>
        </head>
<body>
        <h1>PHP Test</h1>
                <p><b>Hello wor
ld PHP</b></p>
                <?php echo "The
 Current Date and Time: <br />"
;
                echo date("g:i
A l, F j Y.");?> </p>
        <h2>PHP Information:</h
2>
                <p> <?php phpin
fo(); ?> </p>
        </body>
</html>
```



Save it.

You test it now. Open your browser and navigate to URL where **hello.php** file located.



5.6 Testing PHP and MySQL

In this section, we will create a PHP and MySQL application. We create a file, **hellodb.php**.

```
$ sudo nano /var/www/hellodb.ph
p
```

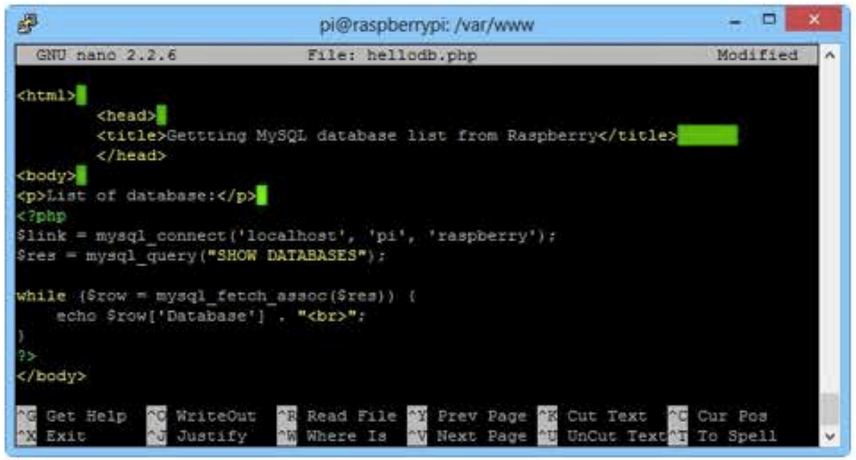
Write all code below.

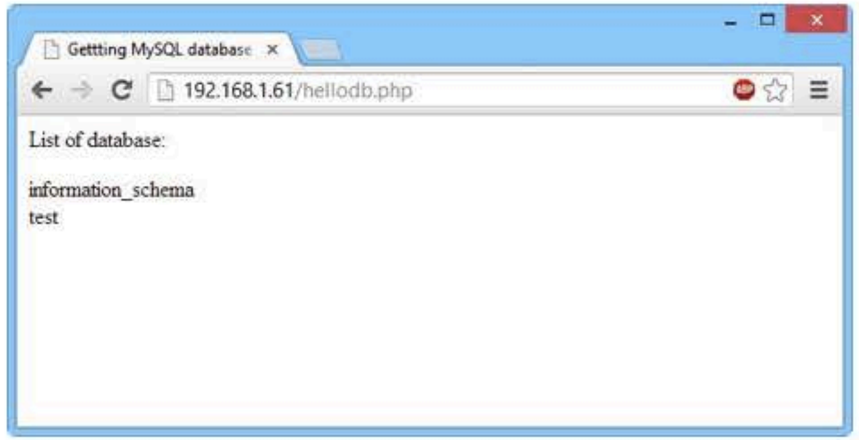
```
<html>
        <head>
        <title>Gettting MySQL d
atabase list from Raspberry</ti
tle>
        </head>
<body>
<p>List of database:</p>
<?php
$link = mysql_connect('localhos
t', 'pi', 'raspberry');
$res = mysql_query("SHOW DATABA
SES");

while ($row = mysql_fetch_assoc
($res)) {
    echo $row['Database'] . "<b
r>";
}
?>
</body>
</html>
```

Note: change username and password for

MySQL.

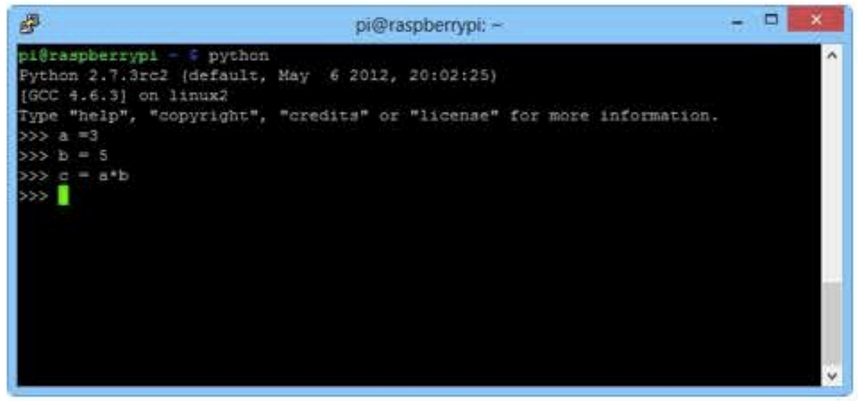




Save it.

You test it now. Open your browser and navigate to URL where **hellodb.php** file is located.

6. Raspberry Pi Programming



This chapter explains how to work with development environment on Raspberry Pi 2.

6.1 Python

Raspberry Pi Wheezy provides Python for development by default so you can execute Python code inside Raspberry Pi console.

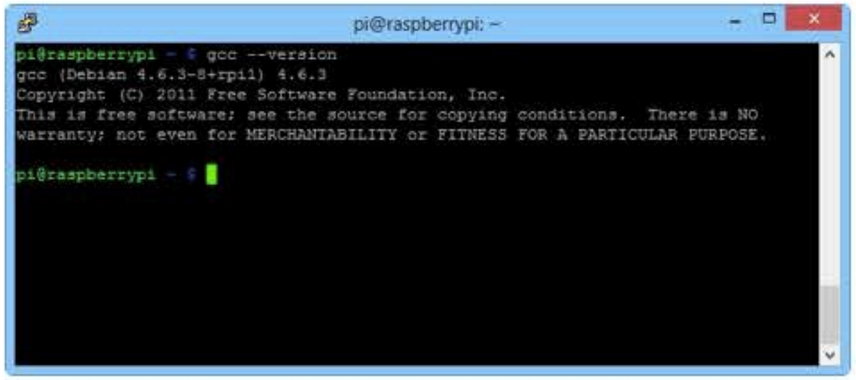
```
$ python
```

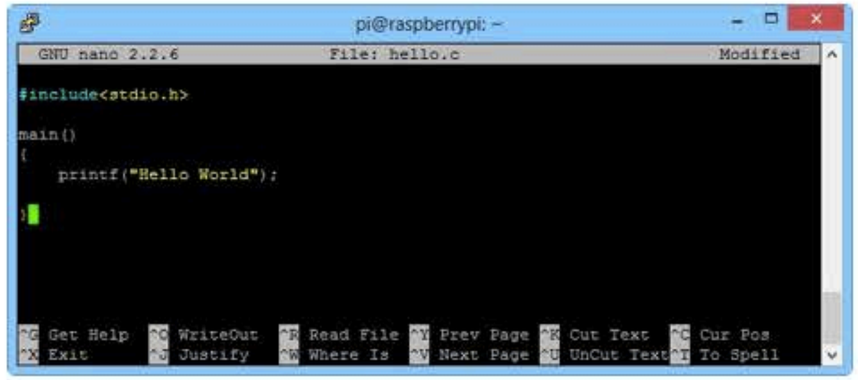
After that, you run Python command.

6.2 C/C++

Raspberry Pi also provides GCC inside package distribution. You can check your current GCC version by typing this command.

```
$ gcc --version
```





For illustration, we create a simple code, hello world. Create a file

```
$ nano hello.c
```

Write this code.

```
#include<stdio.h>

main()
{
    printf("Hello World");

}
```

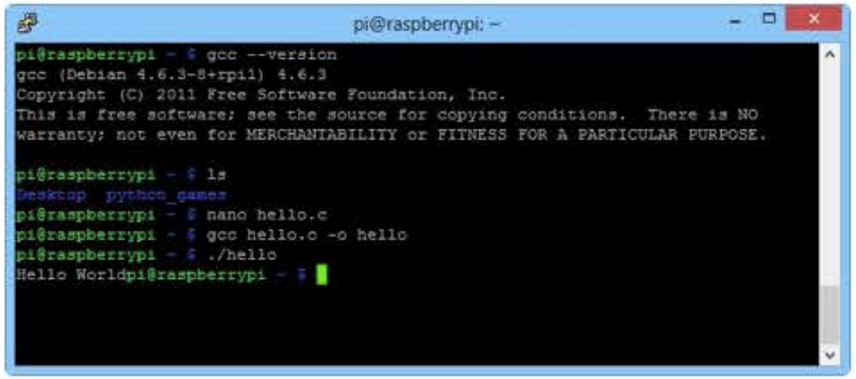
Save it.

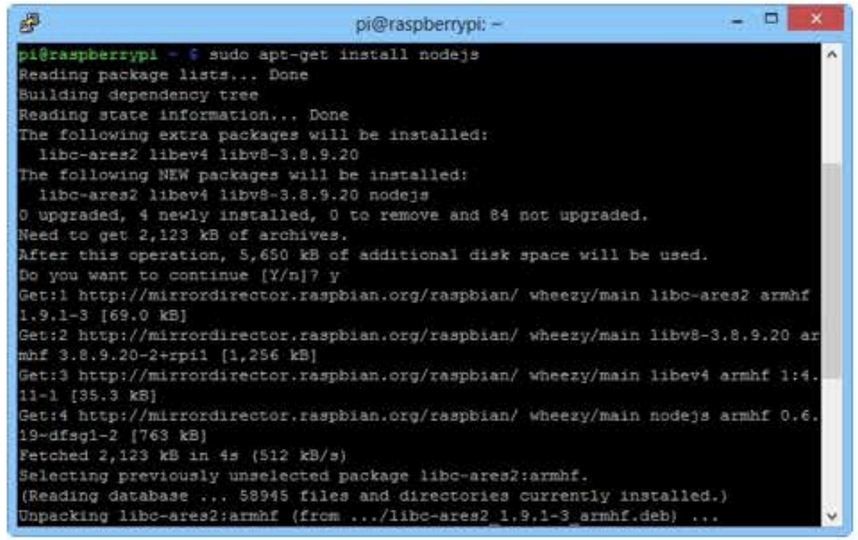
Now you can compile C code using GCC.

```
$ gcc hello.c -o hello
```

To run, you type this script.

```
$ ./hello
```





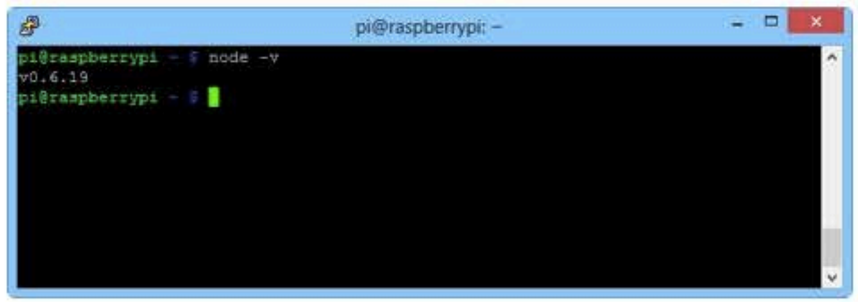
6.3 Node.js

If you are node.js lovers, you can install it into Raspberry Pi. To install execute this command

```
$ sudo apt-get install nodejs
```

Try to check the node.js version

```
$ node -v
```

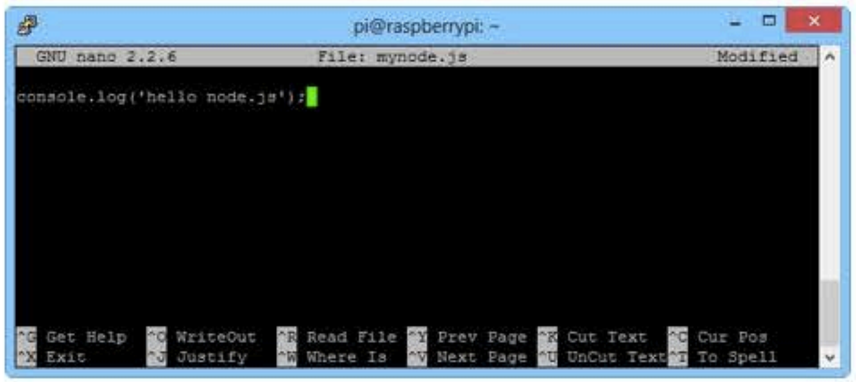


How to get started? Ok, we can create a file, called **mynode.js.**

```
$ nano mynode.js
```

and write this code

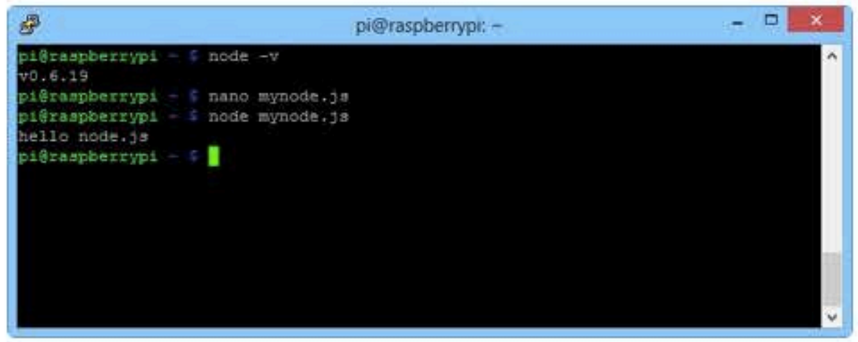
```
console.log('hello node.js');
```



Save it.

Now you can execute **mynode.js** file using node.js

```
$ node mynode.js
```



6.4 Scratch

Scratch is a free desktop and online multimedia authoring tool that can be used by students, scholars, teachers, and parents to easily create games and provide a stepping stone to the more advanced world of computer programming or even be used for a range of educational and entertainment constructivist purposes from math and science projects, including simulations and visualizations of experiments, recording lectures with animated presentations, to social sciences animated stories, and interactive art and music.

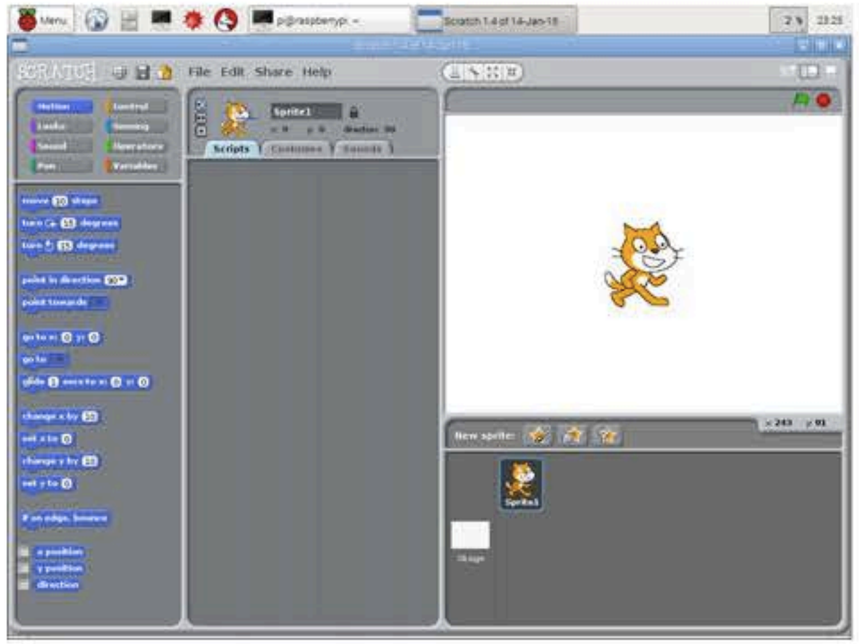
Raspbian already installed it for you. You can run Scratch by clicking scratch logo (see it in Figure below).

After that, you will get a Scratch application.

If you don't see Scratch application, you can type this command on Terminal.

```
$ sudo scratch
```

A sample output of Scratch application can be seen in Figure below.



Further information about Scratch, you can read and learn it on https://scratch.mit.edu/ .

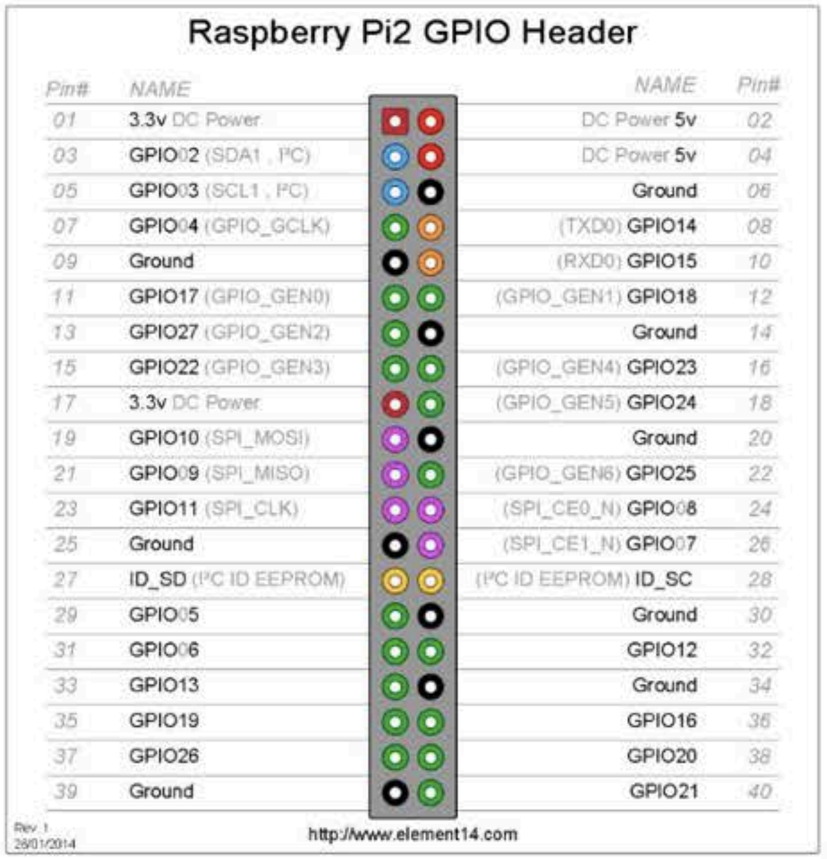
7. Accessing GPIO

This chapter explains how to work with GPIO on Raspberry Pi 2.

7.1 Introduction to GPIO

General-purpose input/output (GPIO) is a generic pin on an integrated circuit whose behavior, including whether it is an input or output pin, can be controlled by the user at run time. GPIO pins have no special purpose defined, and go unused by default.

To understand GPIO on Raspberry Pi 2 board, you can see it in Figure below.



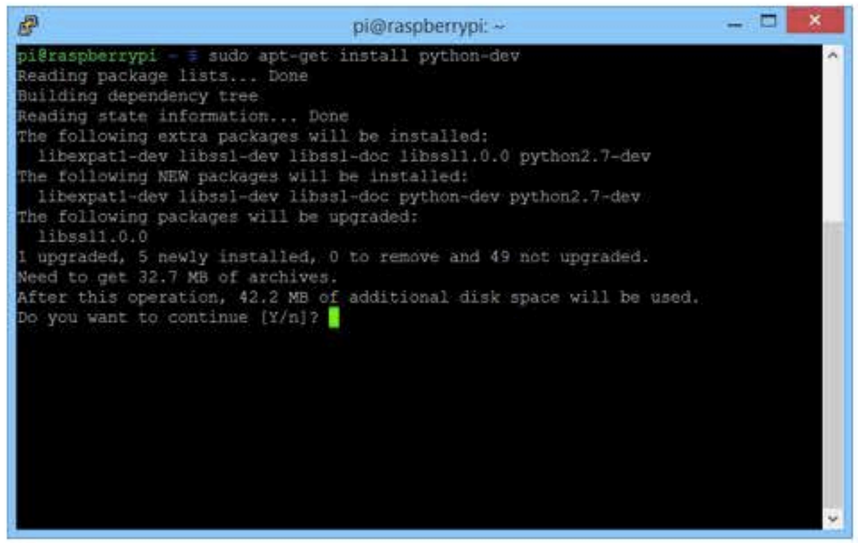
(source: http://www.element14.com/community/docs/DOC-73950/l/raspberry-pi-2-model-b-gpio-40-pin-block-pinout)

7.2 Accessing GPIO

In this section, we will focus on Raspberry Pi GPIO programming using Python. There are many Raspberry Pi GPIO libraries you can choose. I used RPi.GPIO, http://pypi.python.org/pypi/RPi.GPIO . How to install?

For illustration, we will install RPi.GPIO 0.5.11. Firstly, we need Python development library. Type the following command.

```
$ sudo apt-get install python-dev
```



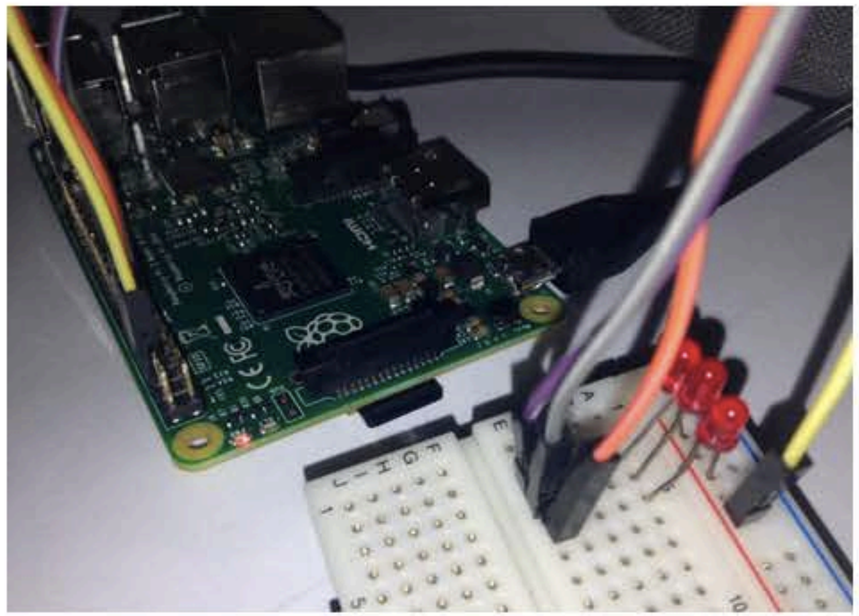
Make sure your Raspberry Pi already connected to Internet network.

Now you can download RPi.GPIO and install it.

```
$ wget $ https://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.5.11.tar.gz
$ tar -xvzf RPi.GPIO-0.5.11.tar.gz
$ cd RPi.GPIO-0.5.11/
$ sudo python setup.py install
```

Make sure your Raspberry Pi already connected to Internet network.

```
pi@raspberrypi: ~/lib/RPi.GPIO-0.5.11
RPi.GPIO-0.5.11/test/test.py
RPi.GPIO-0.5.11/test/btc.py
RPi.GPIO-0.5.11/test/test3.py
RPi.GPIO-0.5.11/CHANGELOG.txt
pi@raspberrypi ~/lib $ cd RPi.GPIO-0.5.11/
pi@raspberrypi ~/lib/RPi.GPIO-0.5.11 $ sudo python setup.py install
running install
running build
running build_py
creating build
creating build/lib.linux-armv7l-2.7
creating build/lib.linux-armv7l-2.7/RPi
copying RPi/__init__.py -> build/lib.linux-armv7l-2.7/RPi
running build_ext
building 'RPi.GPIO' extension
creating build/temp.linux-armv7l-2.7
creating build/temp.linux-armv7l-2.7/source
gcc -pthread -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototy
pes -fPIC -I/usr/include/python2.7 -c source/py_gpio.c -o build/temp.linux-armv7
l-2.7/source/py_gpio.o
gcc -pthread -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototy
pes -fPIC -I/usr/include/python2.7 -c source/c_gpio.c -o build/temp.linux-armv7l
-2.7/source/c_gpio.o
```



7.3 Demo

In this section, we learn how to write data using GPIO on Raspberry Pi. We can use 3 LEDs to illustrate our case.

Our LEDs are connected to GPIO pins: 14, 15, and 18. LED ground pin is connected to GPIO GND.

Now we create Python application to write data on GPIO. We can use GPIO.output() to write data, TRUE and FALSE.

Create a file, called **gpio_led.py**, and write the following code.

```
import RPi.GPIO as GPIO
import time


led1_pin = 14
led2_pin = 15
```

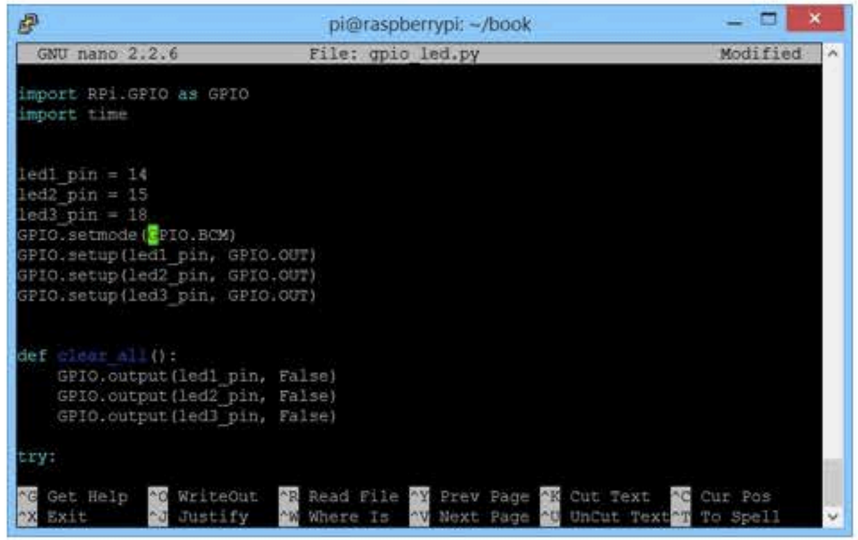
```
led3_pin = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(led1_pin, GPIO.OUT)
GPIO.setup(led2_pin, GPIO.OUT)
GPIO.setup(led3_pin, GPIO.OUT)


def clear_all():
    GPIO.output(led1_pin, False
)
    GPIO.output(led2_pin, False
)
    GPIO.output(led3_pin, False
)

try:
    while 1:
        clear_all()
        print("turn 1 on")
        GPIO.output(led1_pin, T
rue)
        time.sleep(2)
        clear_all()
        print("turn 2 on")
        GPIO.output(led2_pin, T
rue)
        time.sleep(2)
        clear_all()
        print("turn 3 on")
        GPIO.output(led3_pin, T
rue)
        time.sleep(2)
        clear_all()

except KeyboardInterrupt:
    GPIO.cleanup()

print("done")
```

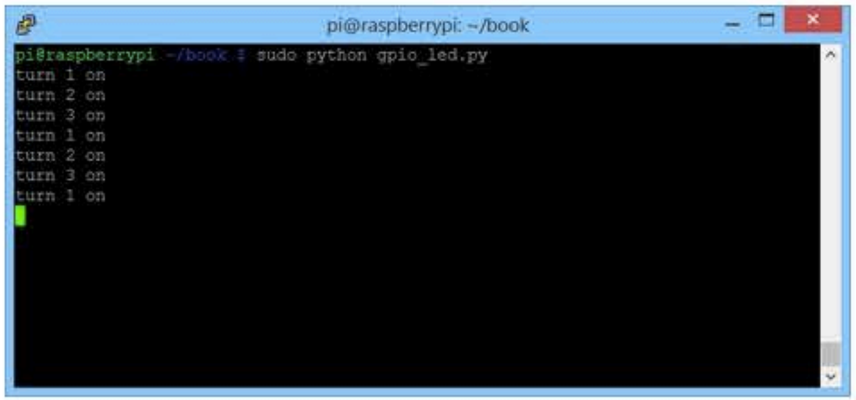


Save this code.

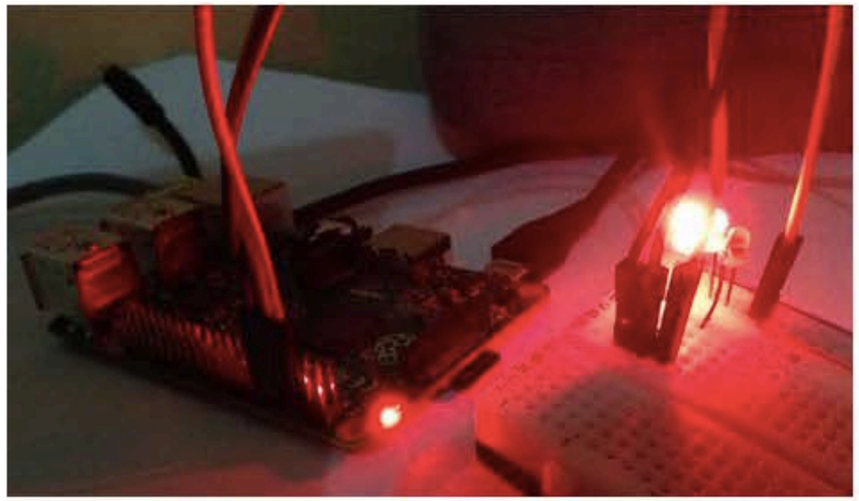
You can run the application using the following command.

```
$ sudo python gpio_led.py
```

If success, you can see the output as below.

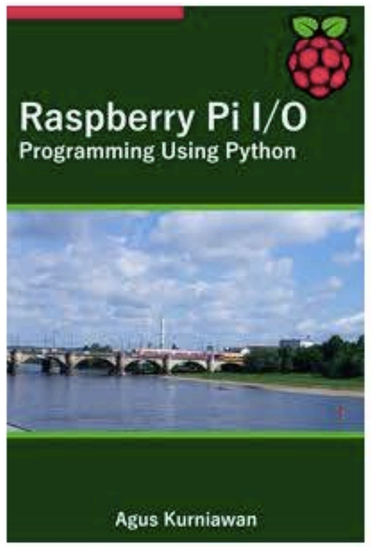


Three LEDs also are blinking.



7.4 Further Reading

You can learn Raspberry Pi GPIO by trying more practices. I have written a book, Raspberry Pi I/O Programming Using Python. This book helps you to get started Raspberry Pi I/O programming. This book uses Raspberry Pi 1 but it's compatible with Raspberry Pi 2. Further information about this book, you can visit to my blog, http://blog.aguskurniawan.net/post/Raspberry-Pi-IO-Programming-using-Python.aspx .


Raspberry Pi I/O
Programming Using Python
Agus Kurniawan

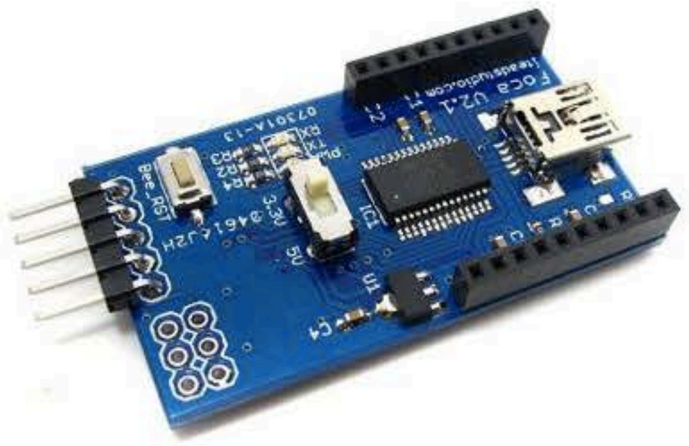
8. Raspberry Pi 2 Serial Debugging

This chapter explains how to access Raspberry Pi 2 via Serial Port.

8.1 Preparation

To debug Raspberry Pi using GPIO serial through computer, we need USB TTL device. There are a lot of USB TTL device, for instance, USB to TTL Serial Cable - Debug / Console Cable for Raspberry Pi from Adafruit, http://www.adafruit.com/products/954 .

Another device, you can buy USB to TTL on Cooking-Hacks, http://www.cooking-hacks.com/index.php/usb-to-ttl-converter-cp210.html .



In this section, I used a Foca V2.1 FT232RL Tiny Breakout USB to Serial UART Interface from iteadstudio. I bought it on

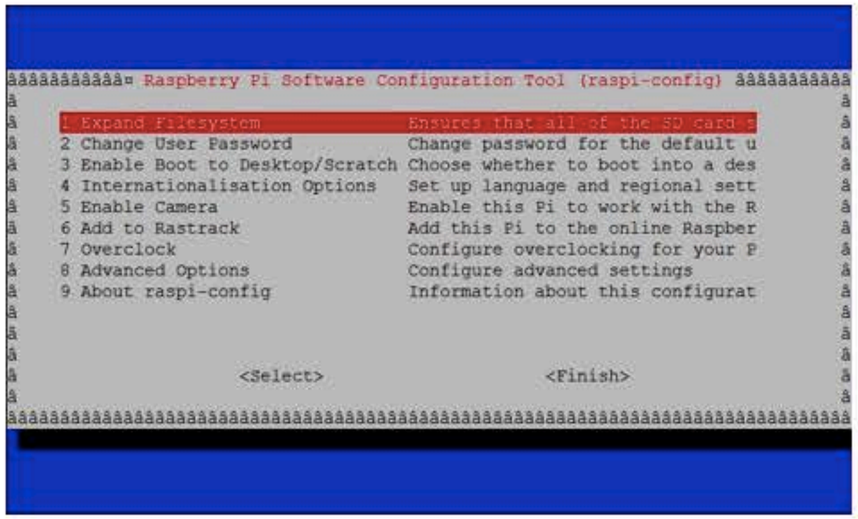
http://www.exp-tech.de/Shields/Foca-V2-1-FT232RL-Tiny-Breakout-USB-to-Serial-UART-Interface.html

8.2 Enabling Serial Debugging

By default, Raspbian disables serial debugging so we need to enable this feature. On Terminal type the following command.

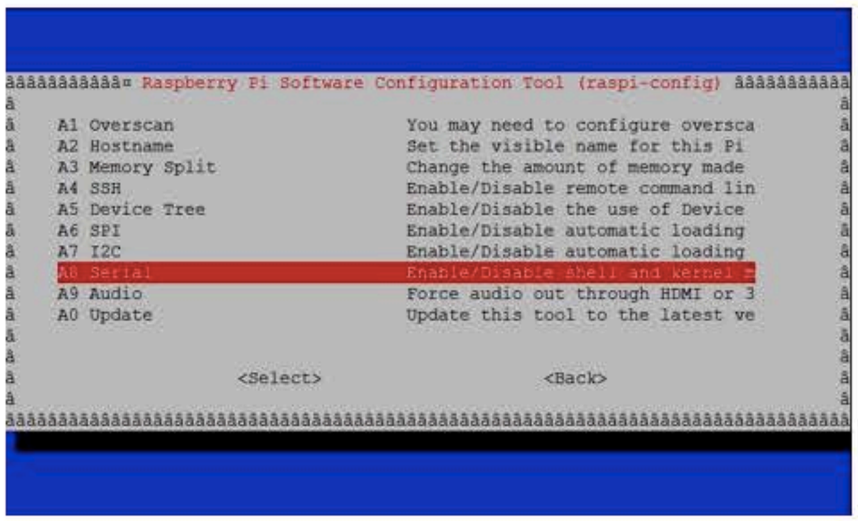
```
$ sudo raspi-config
```

You will get a dialog. Select **8 Advanced Options** menu.



Furthermore, you will get a confirmation. Please select **<Yes>** to enable serial debugging feature.

Then, select **A8 Serial** menu.



If success, you will get a success confirmation.

Close this config dialog.

Now you can reboot your Raspbian by typing the command on Terminal.

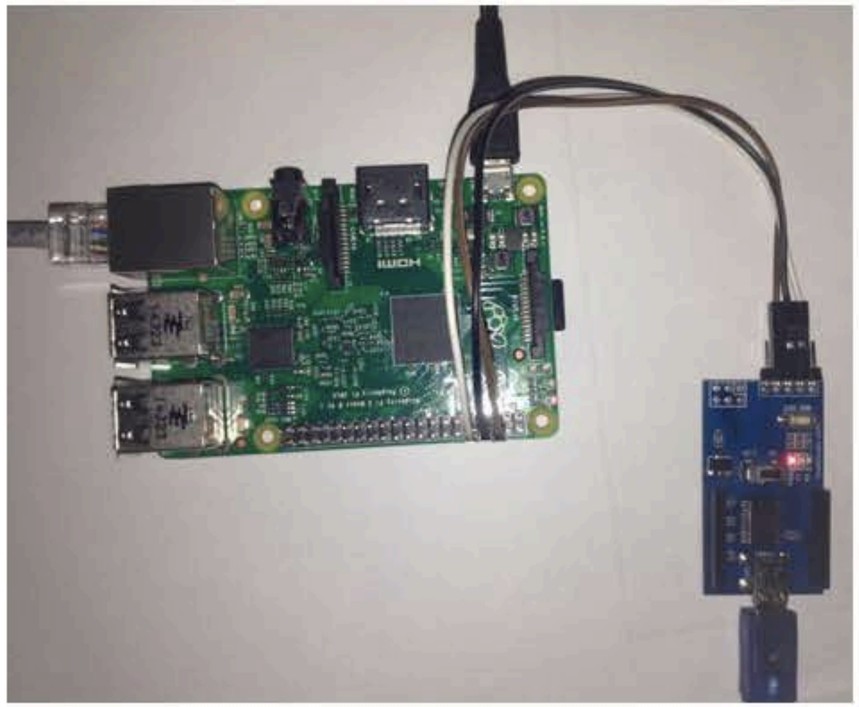
```
$ sudo reboot
```

8.3 Wiring

How to implement?

It's easy. You can just connect Tx from USB TTL to Raspberry Pi UARTo_TXD and USB TTL RX to Raspberry Pi UARTo_RXD. Some USB TTL must change them. It means USB TTL TX should be connected to Raspberry Pi UARTo_RXD and USB TTL RX to Raspberry Pi UARTo_TXD. (Optional) You can connect GND from USB TTL to GND of Raspberry Pi board.

Here is a sample of connected hardware.

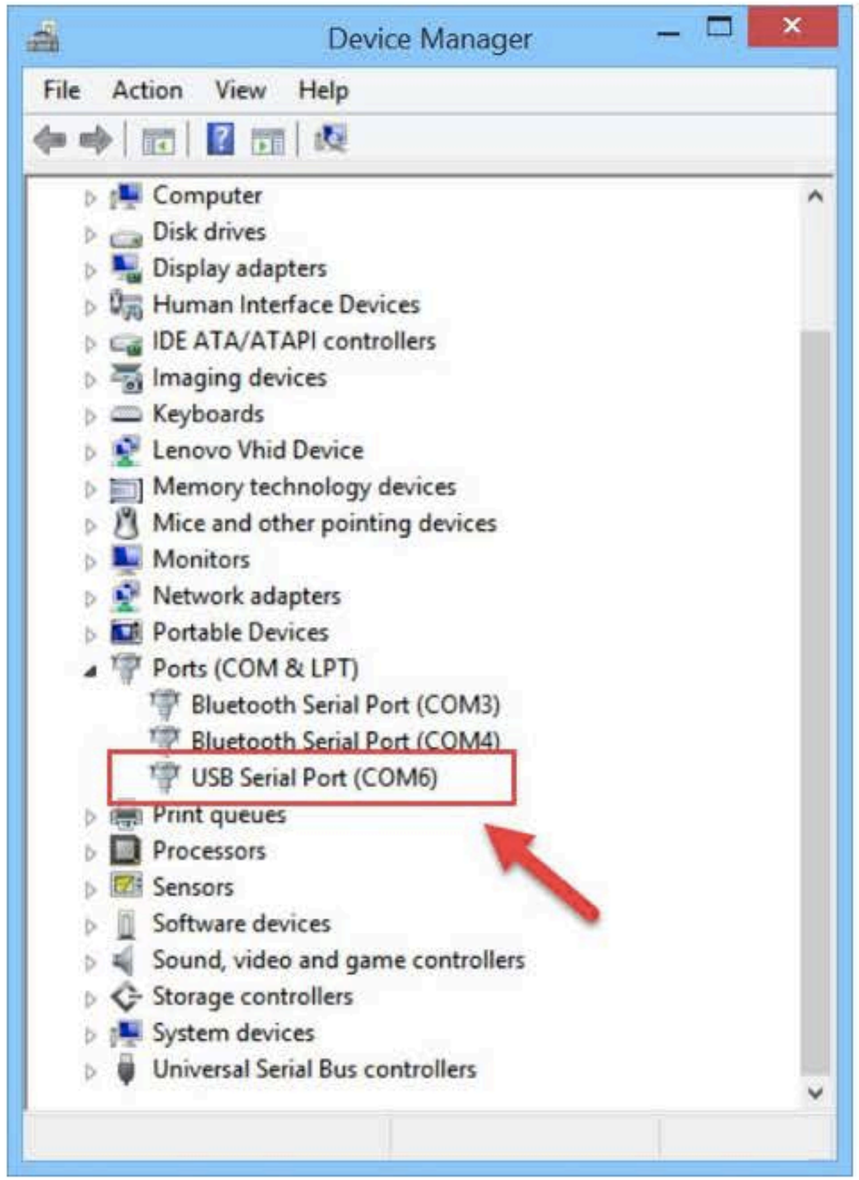


Now your USB cable of USB TTL device is be connected to your computer. You can use any

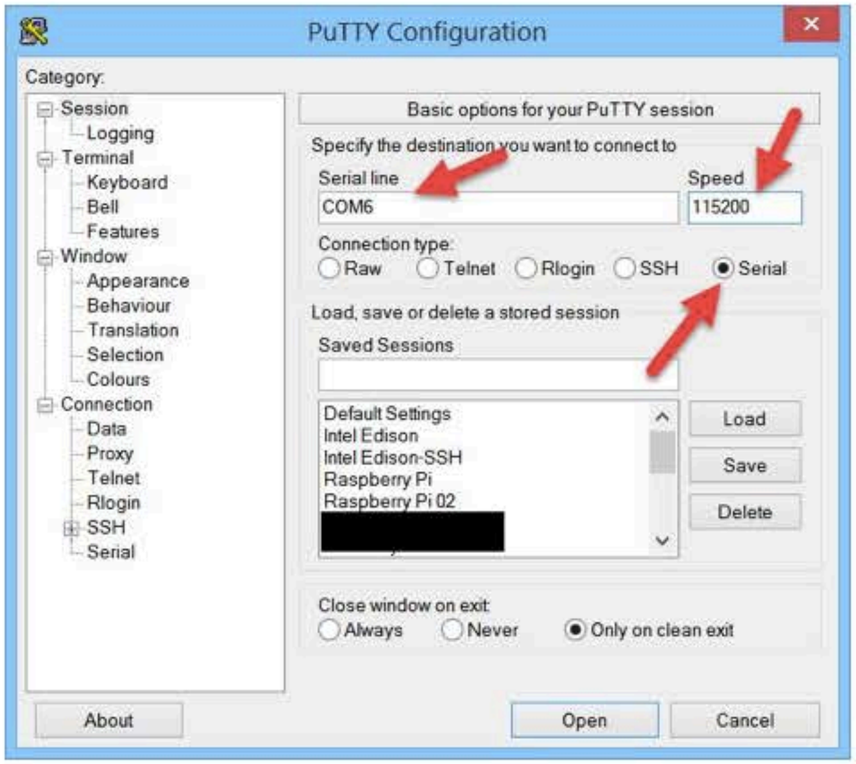
serial application to execute.

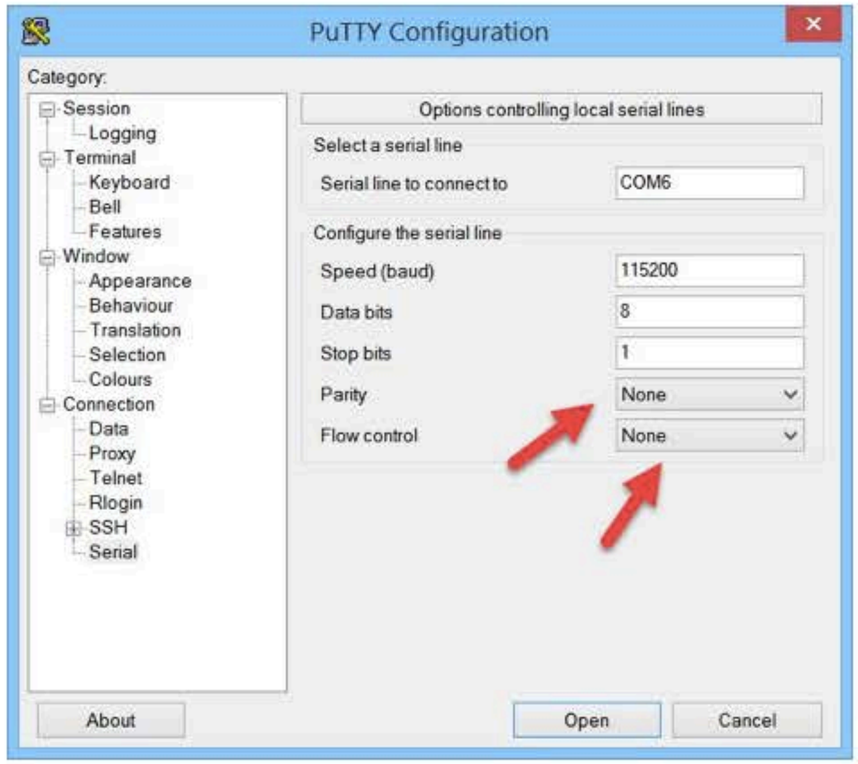
In this book, I used PuTTY, http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html, and run it on my Windows OS.

Run PuTTY and choose Serial for connection type. Fill Serial line name, for instance, my Windows detected it on COM6 as below.



Set 115200 for speed serial.



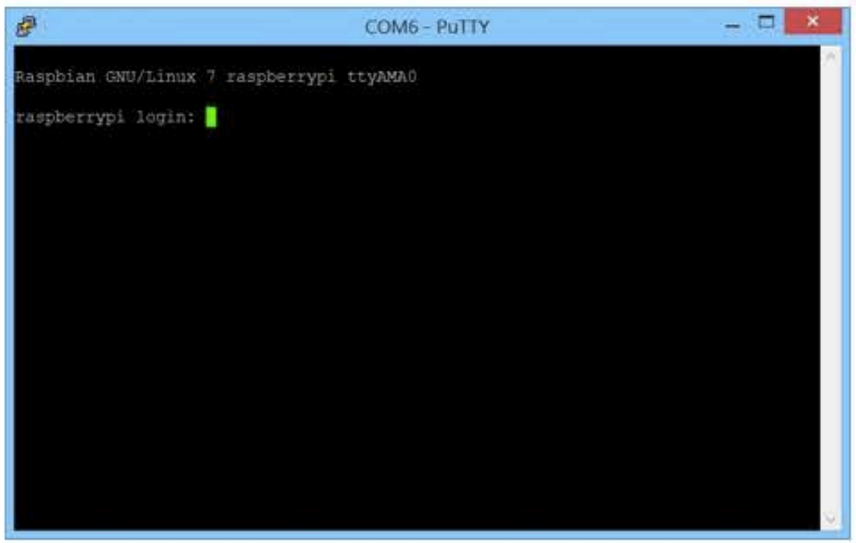


Click **Serial** on side menu and choose **None** for Parity and Flow control.

8.4 Testing

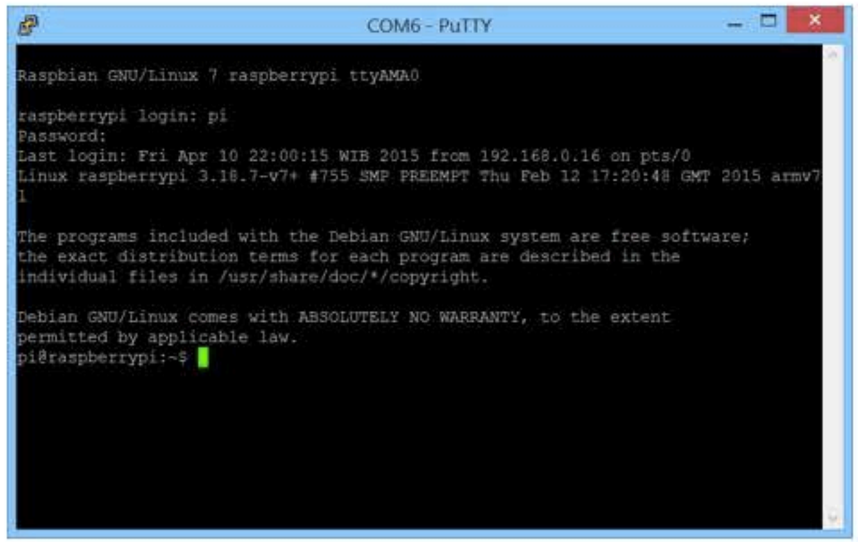
If you're ready, you can click **Open** button. You may press Enter on keyboard when you see blank screen.

If success, you will get the authentication form.



Then, try to logon to Raspberry Pi.

Here is a sample of serial debugging output.



Contact

If you have question related to this book, please contact me at aguskur@hotmail.com . My blog: http://blog.aguskurniawan.net