

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.

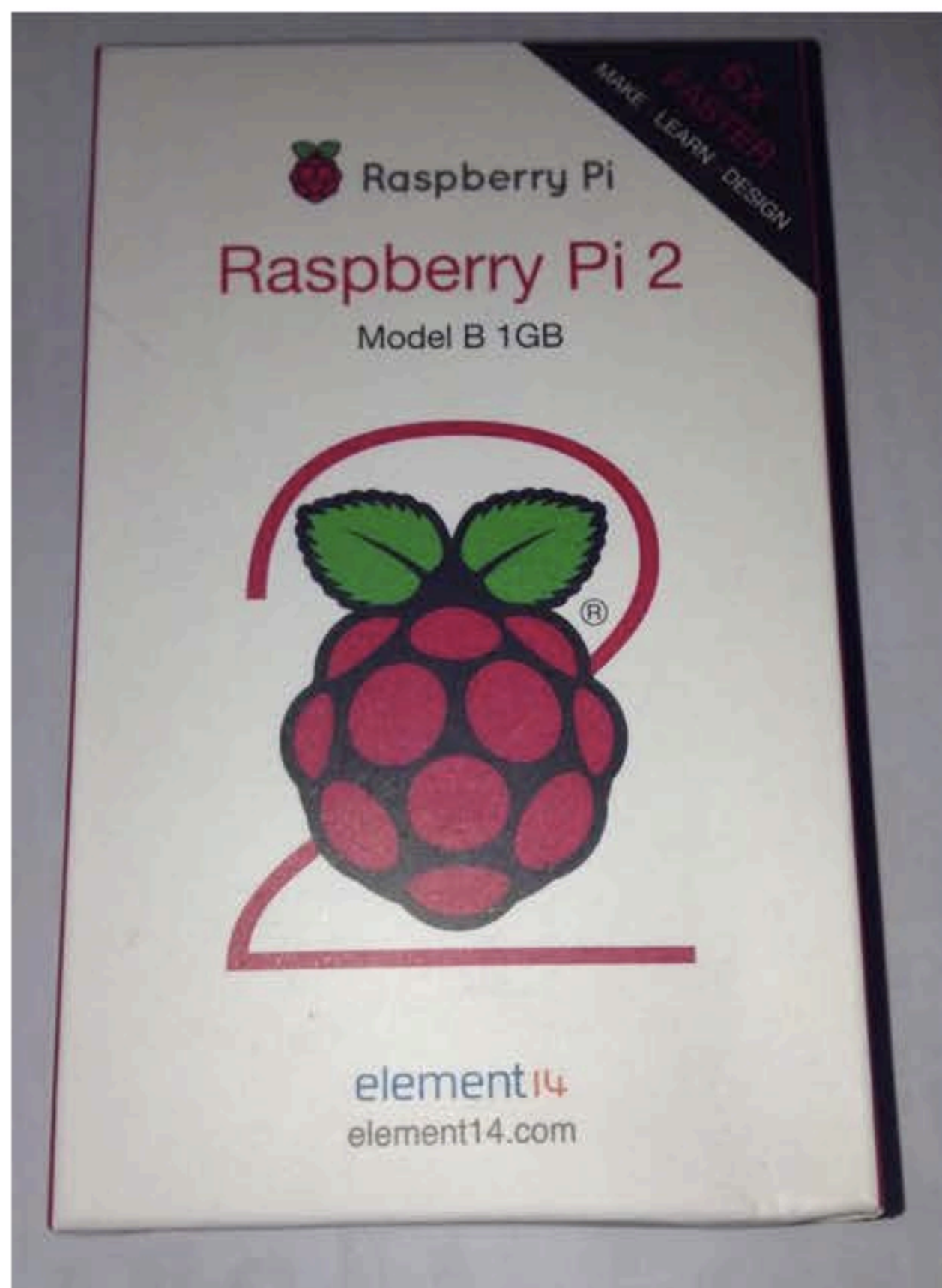
I tried to look for buying Raspberry Pi 2 device and found that there are another options to buy

- 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:



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 Raspberry 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>.

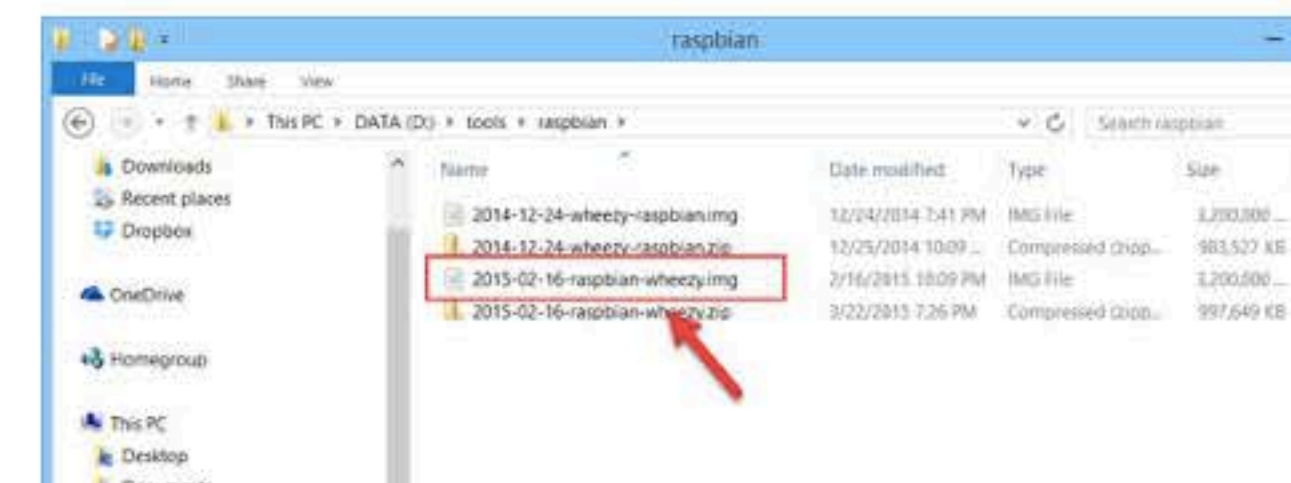
In this book, I use Raspbian "wheezy" OS for testing.

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
```

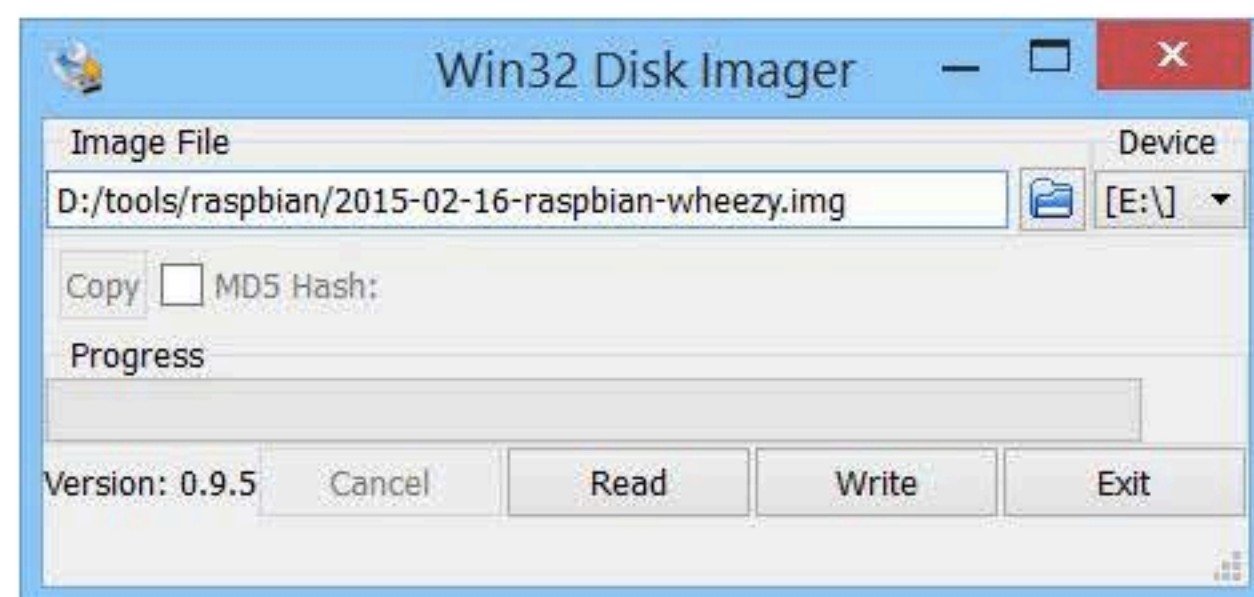

Then, you can copy all img file into MicroSD card.

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

For Windows users:

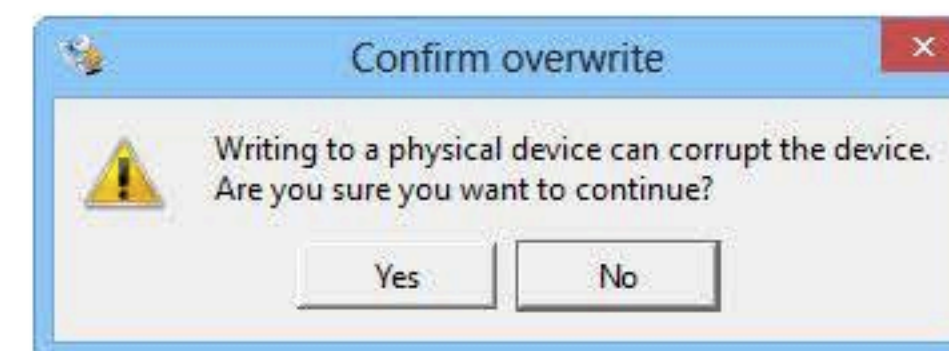
Download Win32DiskImager on
<https://launchpad.net/win32-image-writer/+download>

Run Win32DiskImager and navigate Raspberry Pi image file.



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

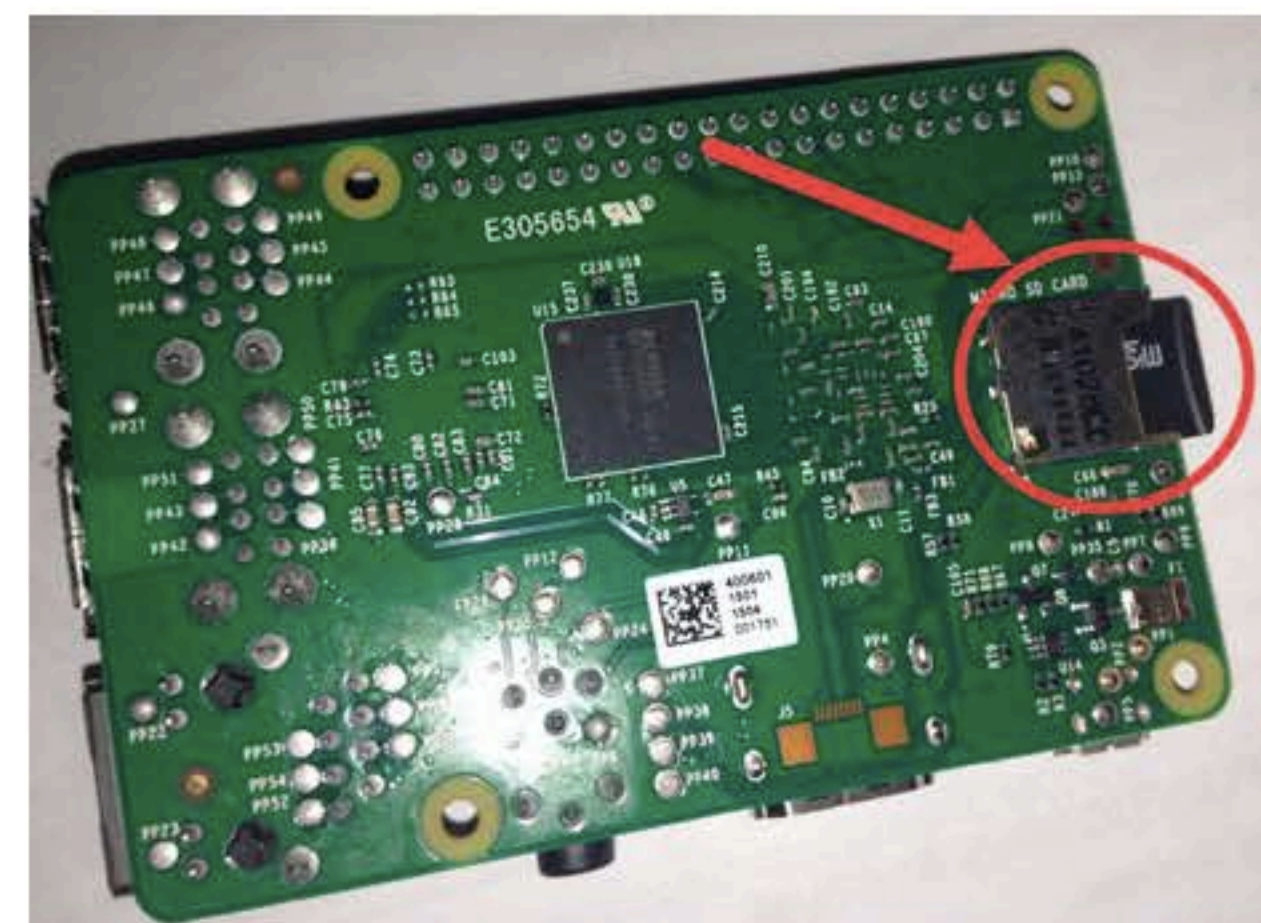
Click **Yes** to confirm overwrite files



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

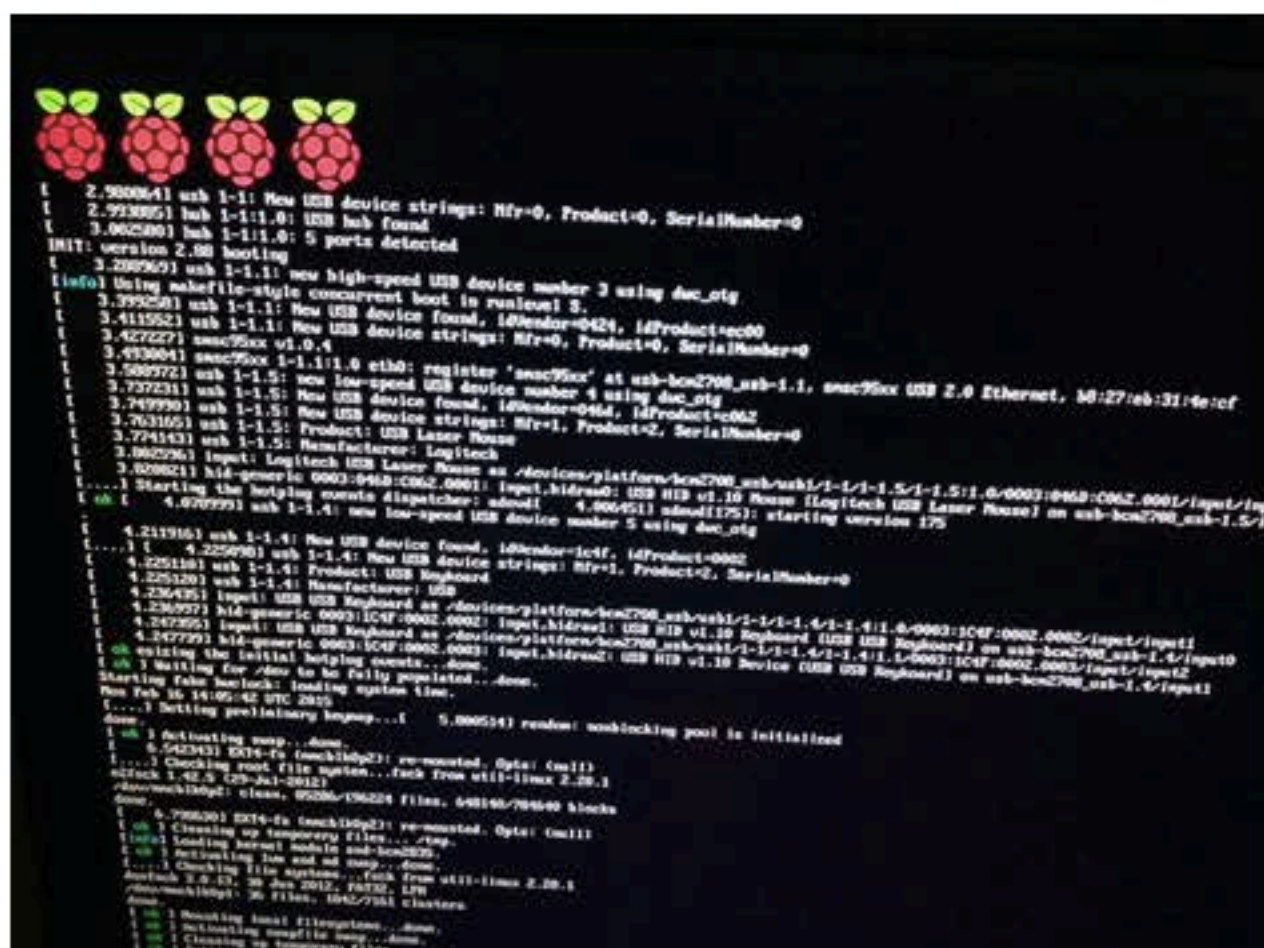
In this chapter we start to run and configure Raspberry Pi 2.

3.1 Put Them All!

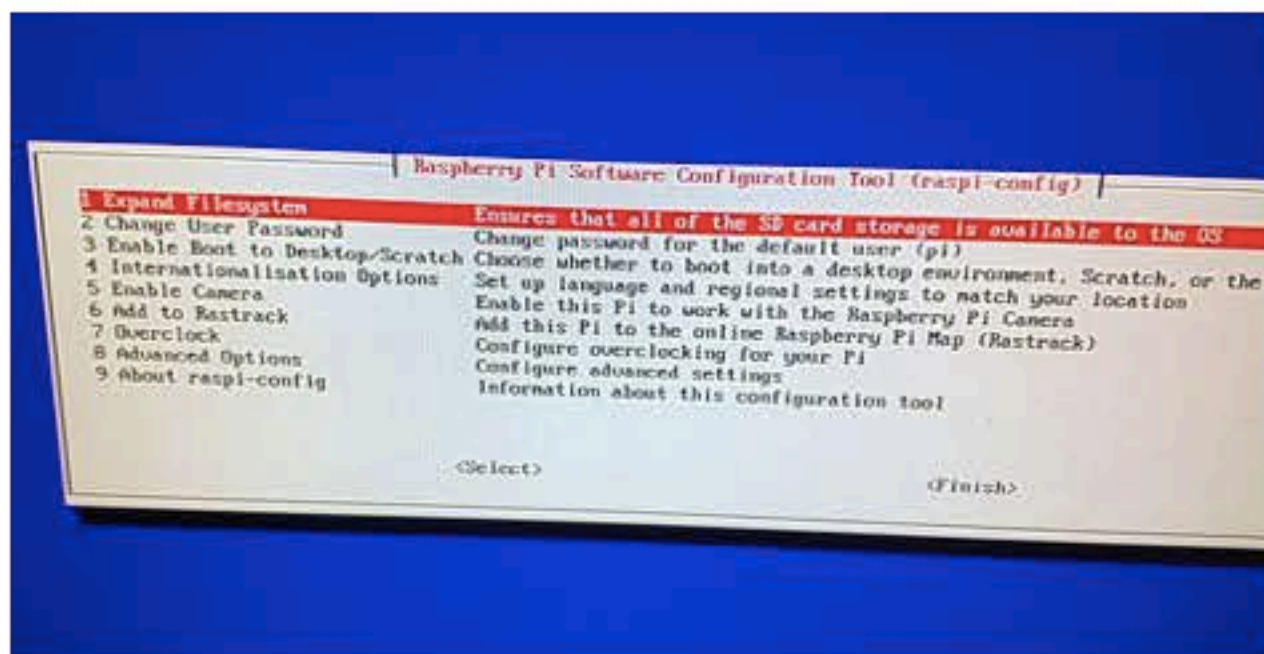
Now you are ready to boot your Raspberry Pi. Please plug in all devices, for instance, mouse, keyboard, power, and HDMI cable, into Raspberry Pi.



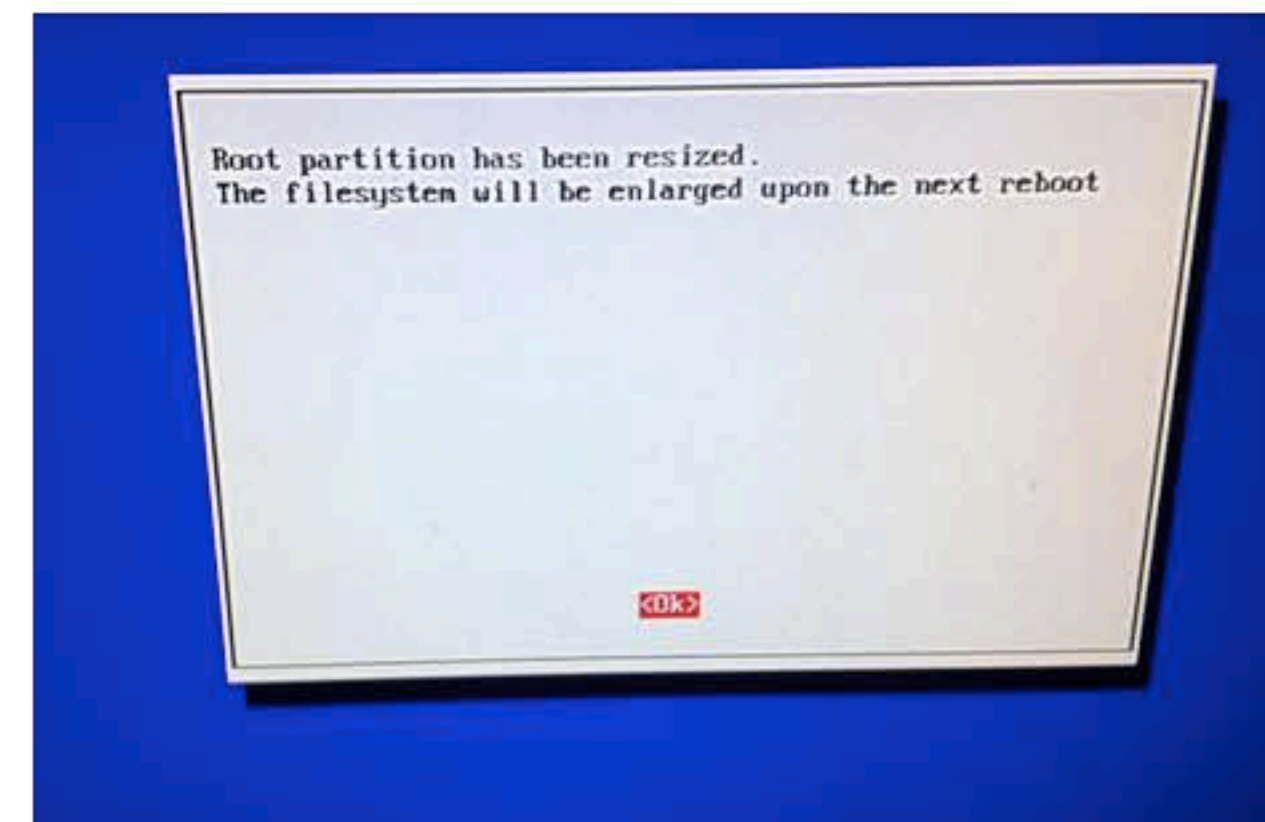
Turn on the power for your Raspberry Pi. Raspbian OS will boot for the first time.



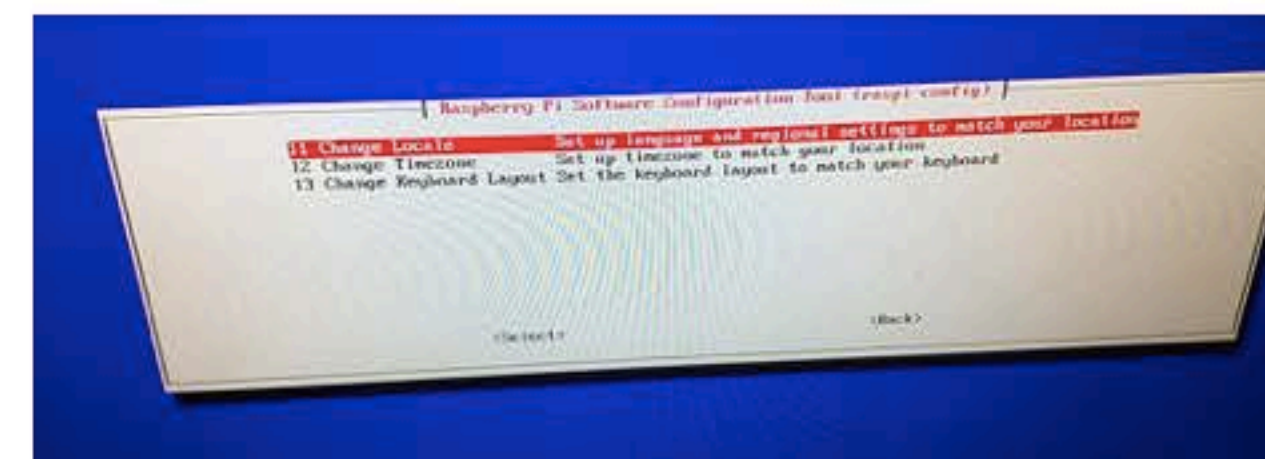
If success, you will get the first screen of Raspberry Pi as below



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.

```
[ ok ] Setting up temporary files....
[info] Setting up ALSA...done.
[info] Setting console screen nodes.
[ ok ] Skipping font and keymap setup (handled by console-setup).
[ ok ] Setting up console font and keymap...done.
[ ok ] Checking if shift key is held down: No. Switching to ondemand s
[ ok ] Setting up X socket directories... /tmp/.X11-unix /tmp/.ICE-un
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.
[ ok ] Network Interface Plugging Daemon...skip eth0...done.
[info] Initializing cgroups.
[warn] Kernel lacks cgroups or memory controller not available, not start
[ ok ] Starting enhanced syslogd: rsyslogd.
Starting dphys-swapfile swapfile setup ...
want /var/swap=100MByte, checking existing: keeping it
done.
[ ok ] Starting periodic command scheduler: cron.
[ ok ] Starting NTP server: ntpd.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenBSD Secure Shell server: sshd.

Raspbian GNU/Linux 7 raspberrypi tty1
raspberrypi login:
```

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

If success, you will get Terminal on Raspbian.

```
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenBSD Secure Shell server: sshd.

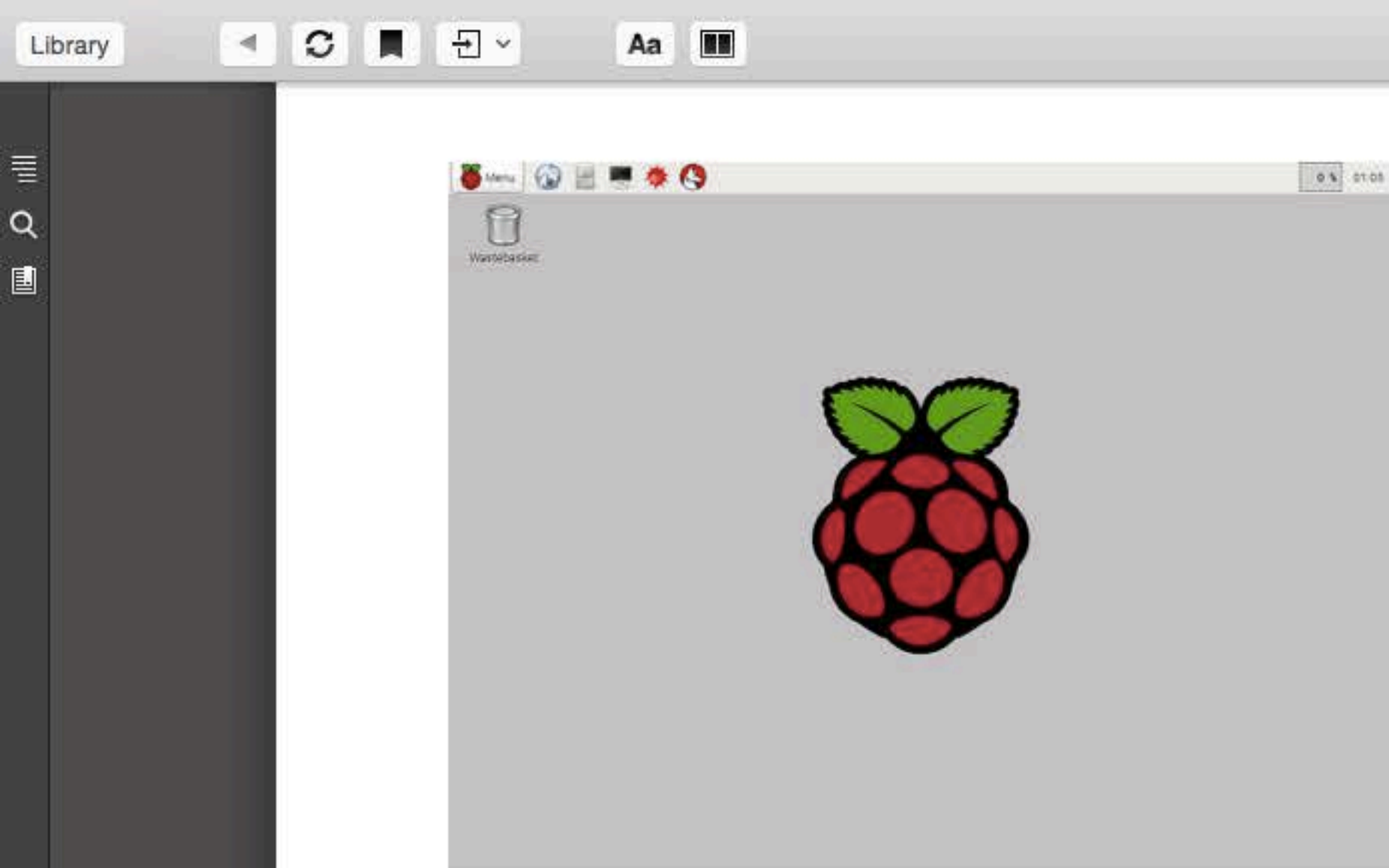
Raspbian GNU/Linux 7 raspberrypi tty1

raspberrypi login: pi
Password:
Linux raspberrypi 3.18.7-v7+ #755 SMP PREEMPT Thu Feb 12 17:
The programs included with the Debian GNU/Linux system are fr
the exact distribution terms for each program are described in
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the exte
permitted by applicable law.
pi@raspberrypi ~ $ _
```

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.

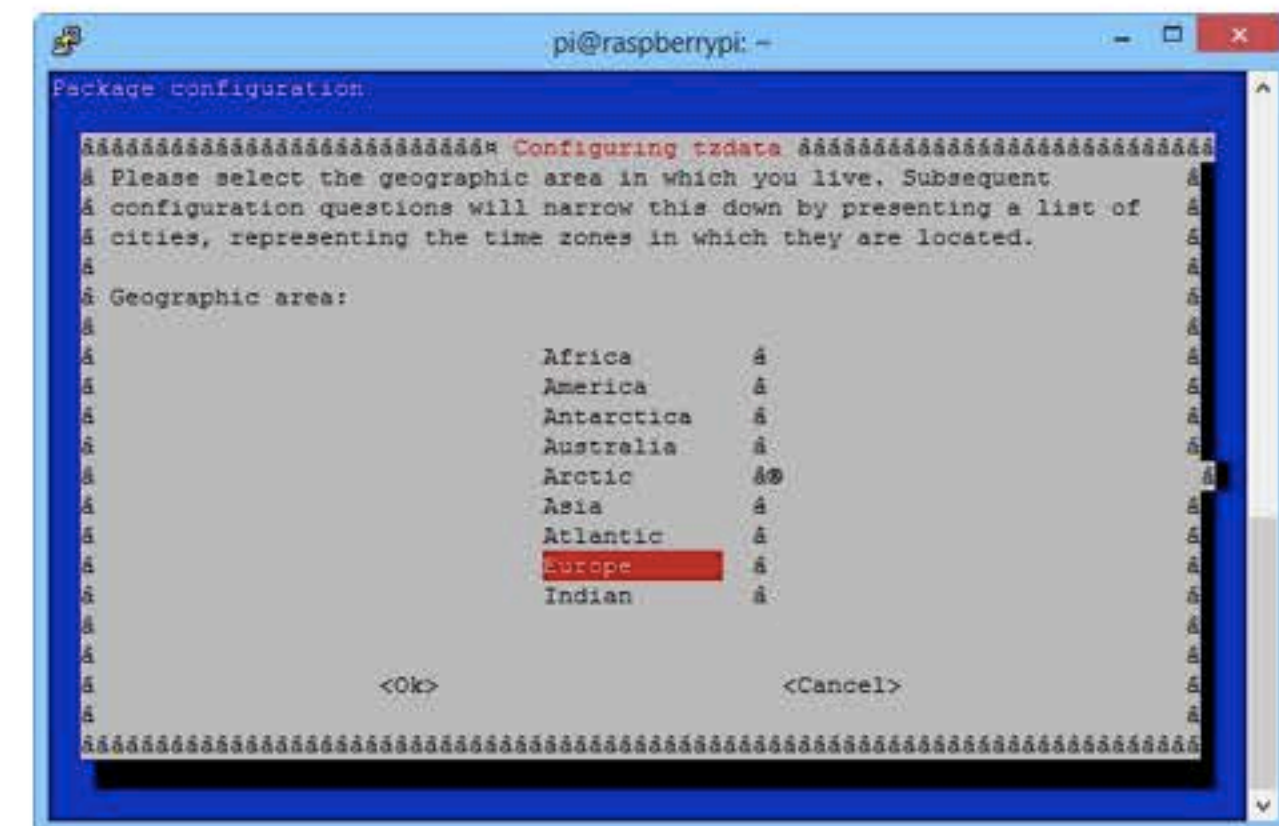


3.2 Configure Timezone

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



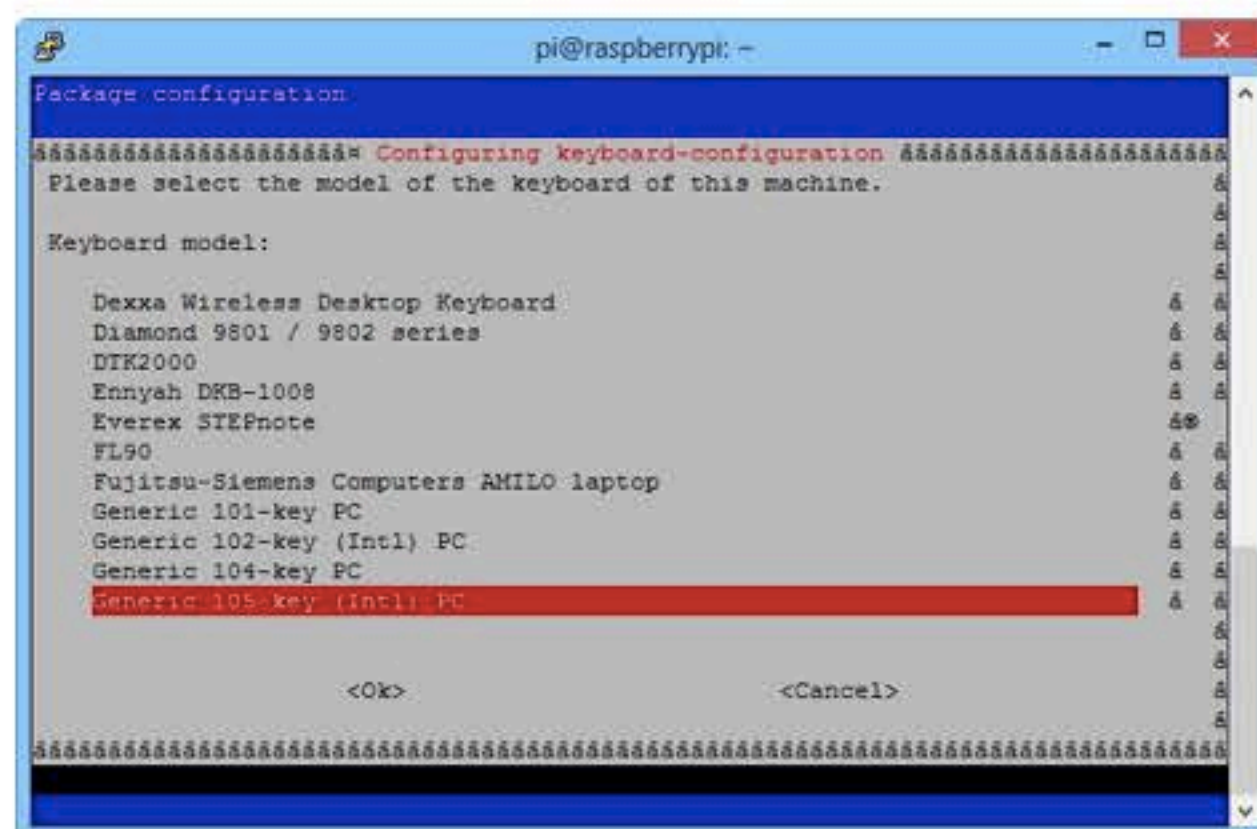
Choose your timezone.

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

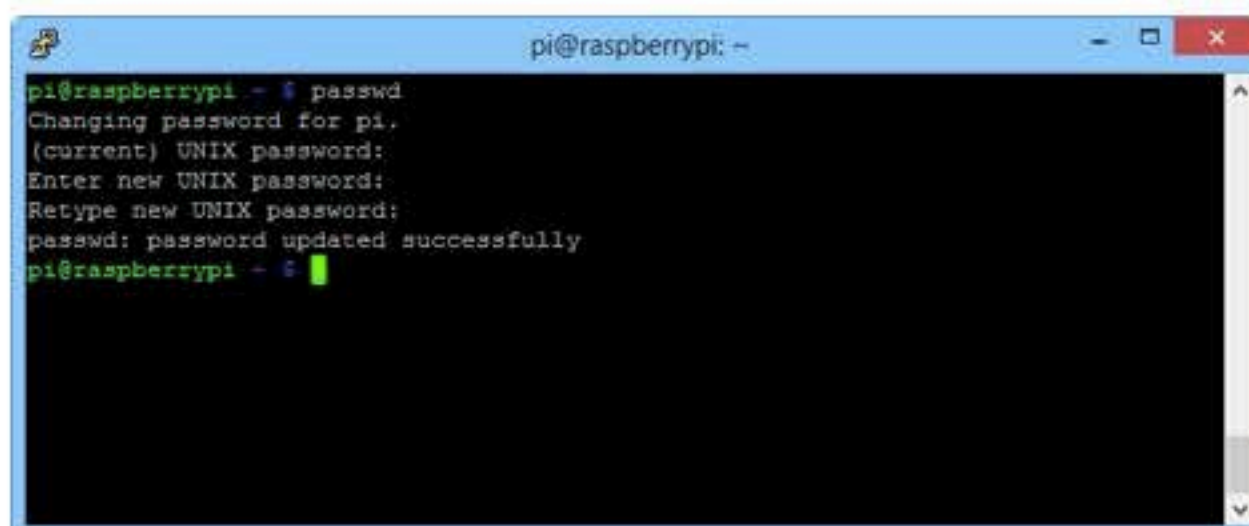
```
$ sudo shutdown -h -P now
```

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

A terminal window titled 'pi@raspberrypi: ~' with a blue title bar. The terminal shows the command 'passwd' being executed. The output is: 'Changing password for pi.', '(current) UNIX password:', 'Enter new UNIX password:', 'Retype new UNIX password:', 'passwd: password updated successfully', and 'pi@raspberrypi: ~' with a green cursor.

```
pi@raspberrypi ~$ passwd
Changing password for pi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi ~$
```

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
```

```
pi@raspberrypi ~$ sudo ifconfig -a
eth0      Link encap:Ethernet  HWaddr b8:27:eb:85:c6:48
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1104 (1.0 KiB)  TX bytes:1104 (1.0 KiB)

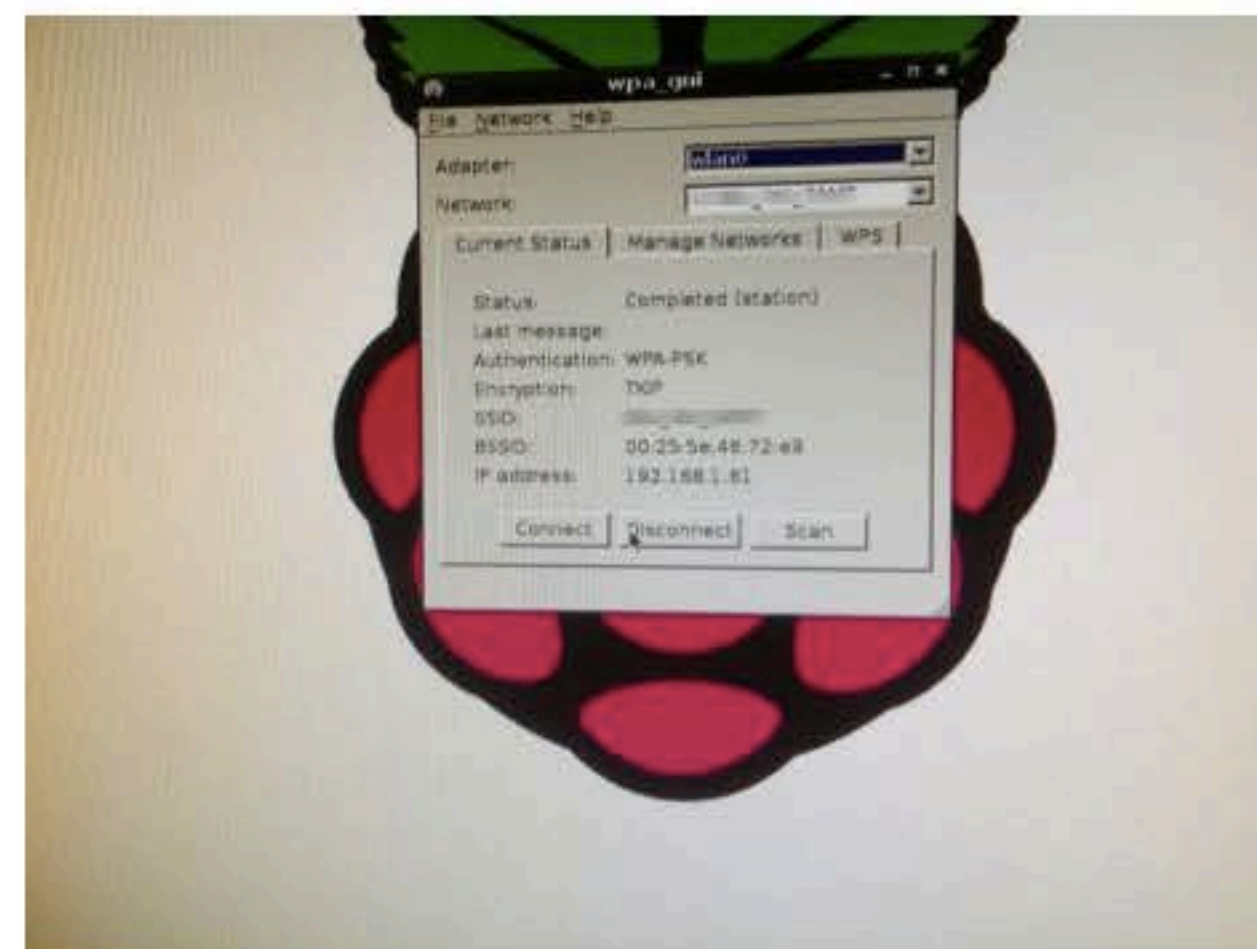
wlan0     Link encap:Ethernet  HWaddr 08:86:3b:b7:77:f2
          inet addr:192.168.1.61  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:168 errors:0 dropped:2 overruns:0 frame:0
          TX packets:107 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20158 (19.6 KiB)  TX bytes:16881 (16.4 KiB)

pi@raspberrypi ~$
```

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

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

You can see your IP Address 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.



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

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

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

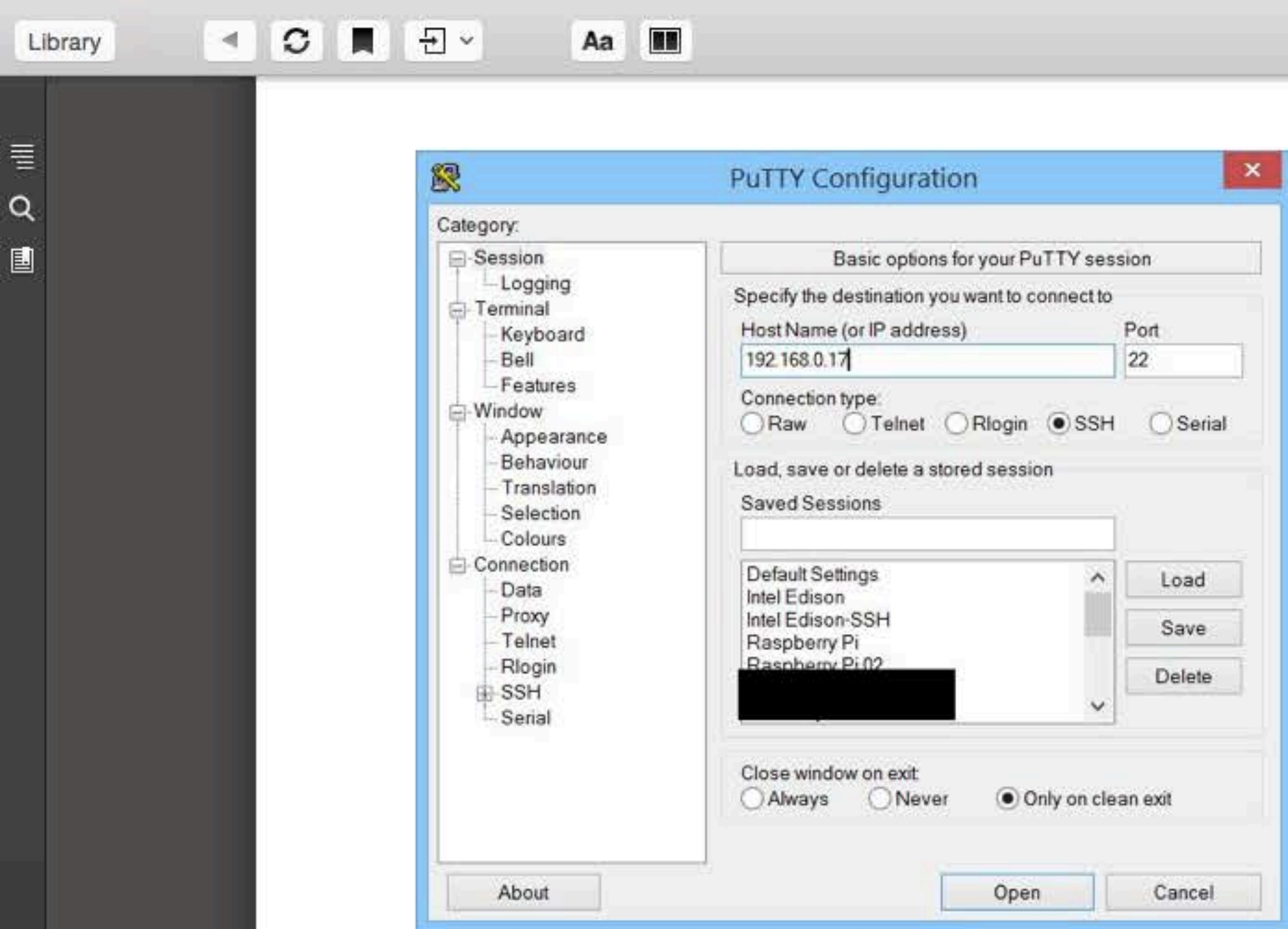
```
$ sudo update-rc.d ssh defaults
```

and then reboot your Raspberry Pi

```
$ sudo reboot
```

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**.

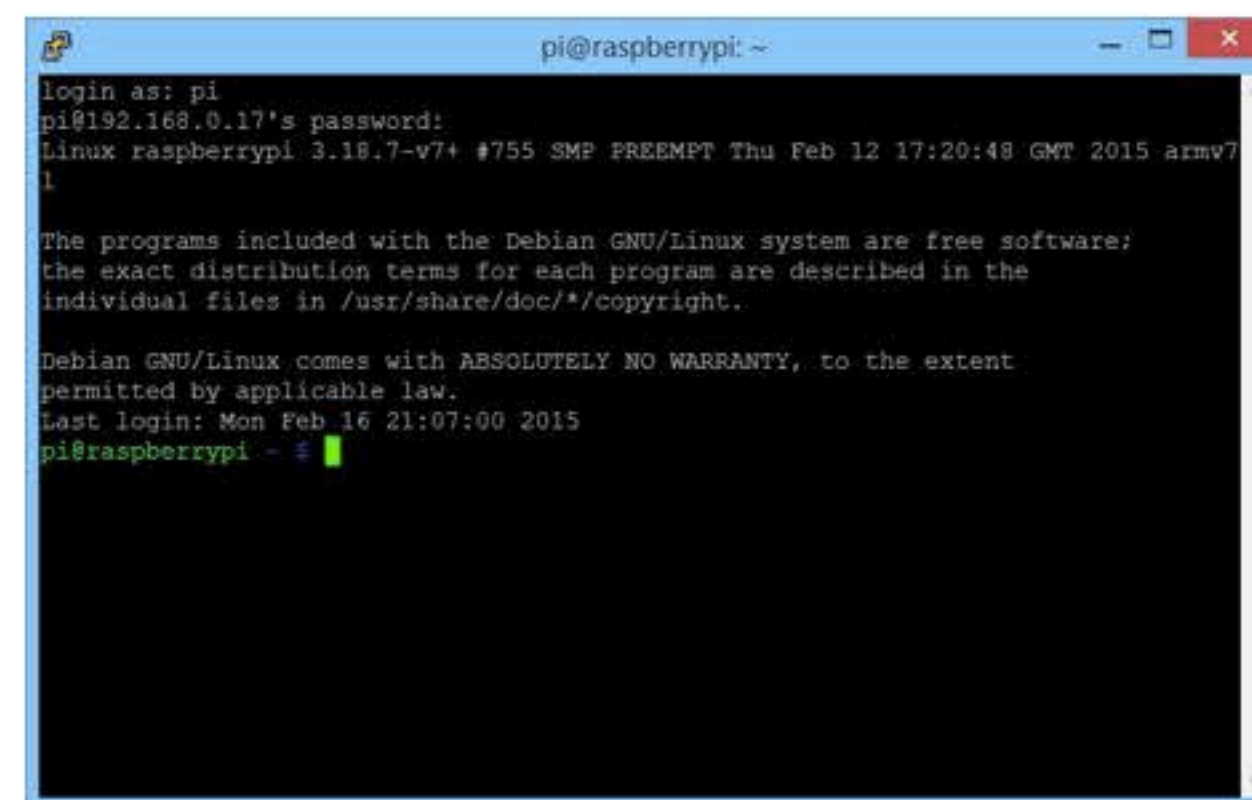
Then, click **Open** button. If connected, you will get a security alert.





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.


```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo apt-get update  
Get:1 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]  
Get:2 http://raspberrypi.collabora.com wheezy Release.gpg [836 B]  
Get:3 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]  
Get:4 http://archive.raspberrypi.org wheezy Release.gpg [490 B]  
Get:5 http://raspberrypi.collabora.com wheezy Release [7,514 B]  
Get:6 http://archive.raspberrypi.org wheezy Release [10.2 kB]  
Get:7 http://mirrordirector.raspbian.org wheezy/main armhf Packages [6,902 kB]  
Get:8 http://raspberrypi.collabora.com wheezy/rpi armhf Packages [2,214 B]  
Get:9 http://archive.raspberrypi.org wheezy/main armhf Packages [118 kB]  
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB  
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en  
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB  
Ign http://archive.raspberrypi.org wheezy/main Translation-en  
Get:10 http://mirrordirector.raspbian.org wheezy/contrib armhf Packages [23.6 kB]  
Get:11 http://mirrordirector.raspbian.org wheezy/non-free armhf Packages [49.3 kB]  
Get:12 http://mirrordirector.raspbian.org wheezy/rpi armhf Packages [592 B]  
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB  
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en  
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB  
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en  
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
```

```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo apt-get install xrdp  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  tightvncserver xfonts-base  
Suggested packages:  
  tightvnc-java  
The following NEW packages will be installed:  
  tightvncserver xfonts-base xrdp  
0 upgraded, 3 newly installed, 0 to remove and 50 not upgraded.  
Need to get 7,219 kB of archives.  
After this operation, 11.5 MB of additional disk space will be used.  
Do you want to continue [Y/n]? y
```

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
```

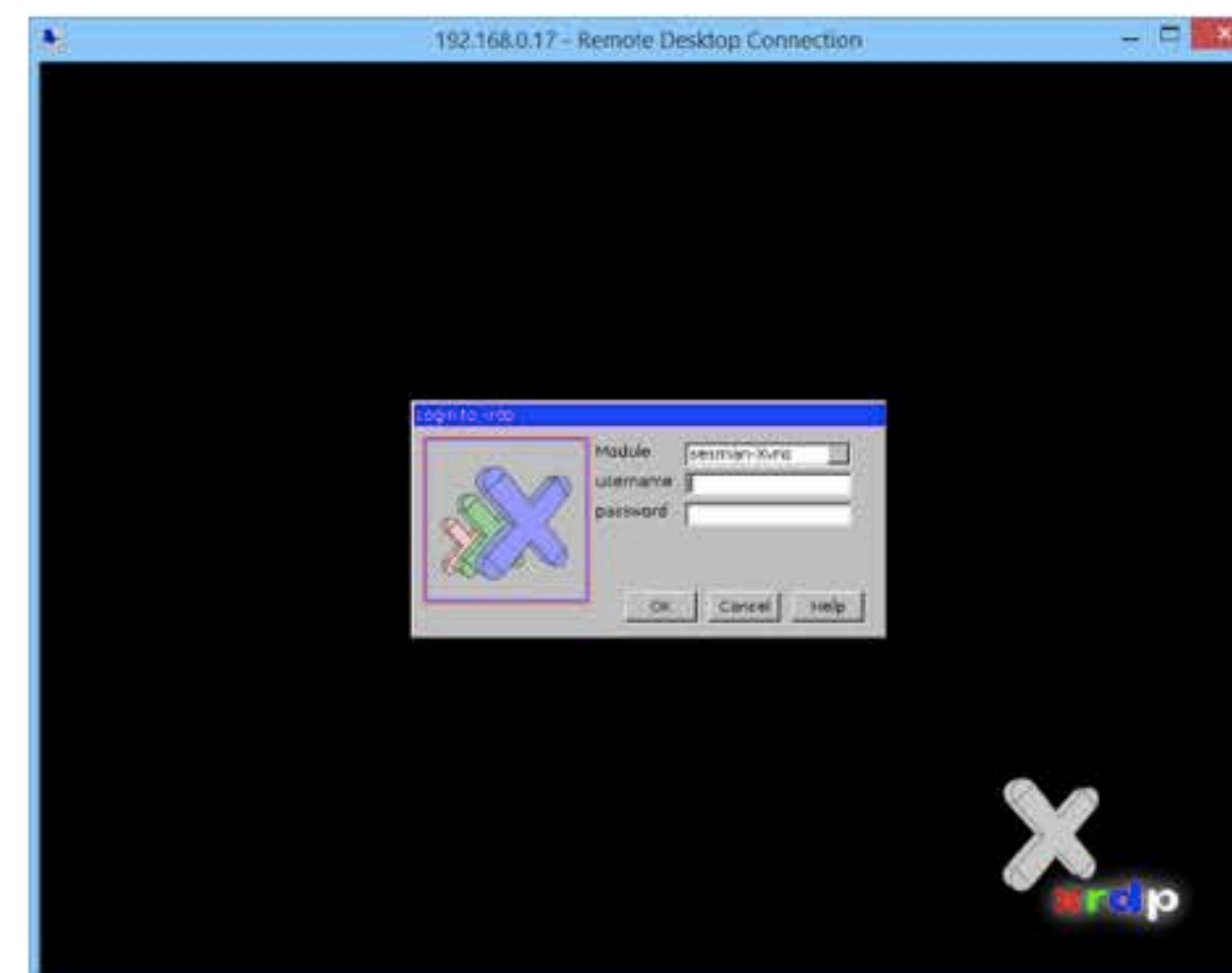
To test, I use Remote Desktop on Windows 8.1. Fill IP address of Raspberry Pi 2.



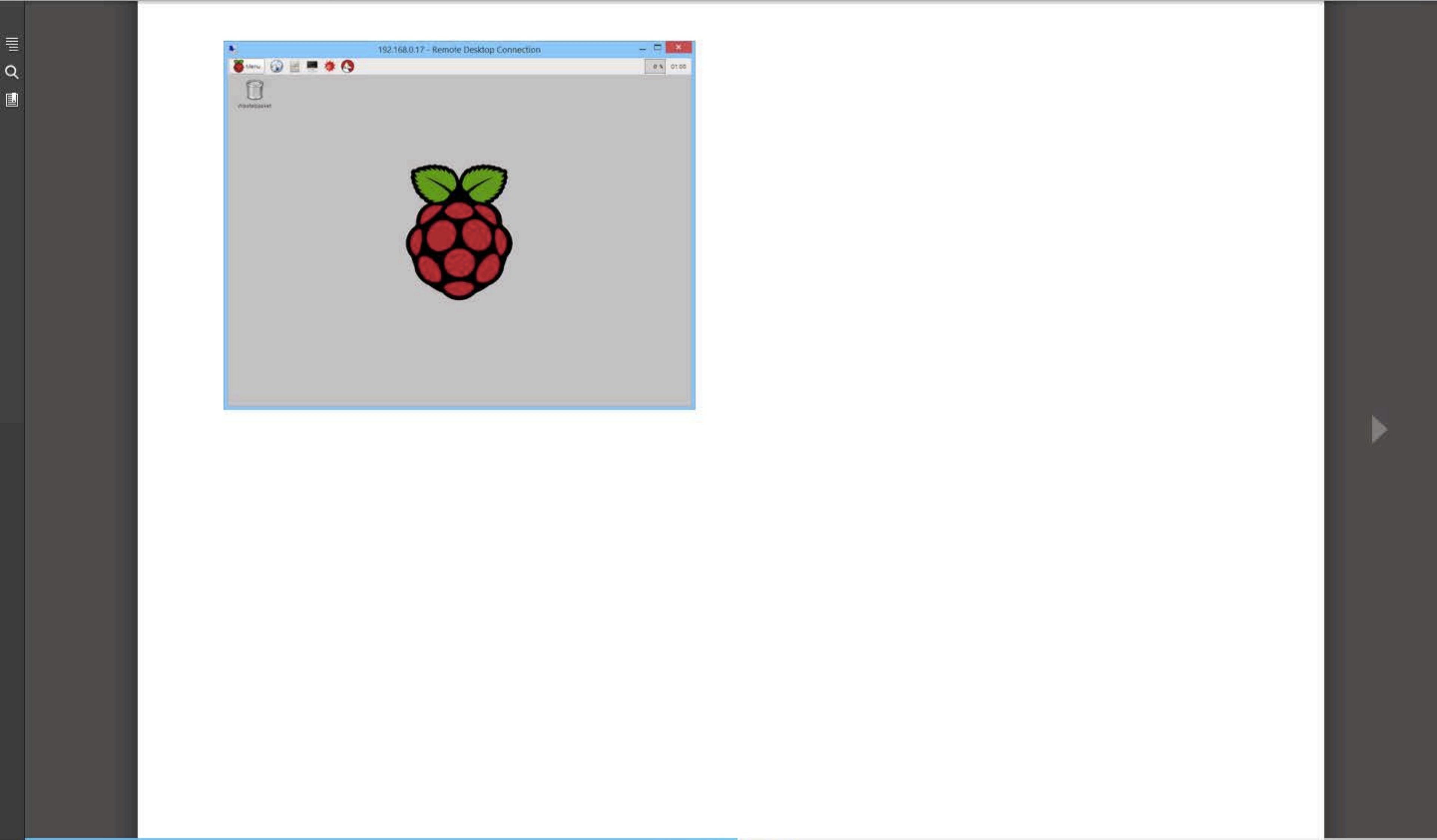
Click **Connect** button. If you will get a warning dialog. Click **Yes** button.



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



If done, you will get Raspbian desktop.



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
```

```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo apt-get install apache2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1  
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert  
Suggested packages:  
  apache2-doc apache2-suexec apache2-suexec-custom openssl-blacklist  
The following NEW packages will be installed:  
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common  
  libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert  
0 upgraded, 10 newly installed, 0 to remove and 84 not upgraded.  
Need to get 1,348 kB of archives.  
After this operation, 4,990 kB of additional disk space will be used.  
Do you want to continue [Y/n]? y  
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main apache2 armhf 2.2  
  .22-12 [1,432 B]  
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libapr1 armhf 1.4  
  .6-3 [90.4 kB]  
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libaprutil1 armhf  
  1.4.1-3 [77.1 kB]  
Get:4 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libaprutil1-dbd-s
```

5.3 Installing MySQL

The second step is to install MySQL. Execute this command

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



```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo apt-get install mysql-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  heirloom-mailx libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl  
  libmysqlclient16 libnet-daemon-perl libplrpc-perl mysql-client-5.5  
  mysql-common mysql-server-5.5 mysql-server-core-5.5  
Suggested packages:  
  exim4 mail-transport-agent libipc-sharedcache-perl libterm-readkey-perl  
  tinyca  
Recommended packages:  
  mailx  
The following NEW packages will be installed:  
  heirloom-mailx libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl  
  libmysqlclient16 libnet-daemon-perl libplrpc-perl mysql-client-5.5  
  mysql-common mysql-server-5.5 mysql-server-core-5.5  
0 upgraded, 13 newly installed, 0 to remove and 84 not upgraded.  
Need to get 9,950 kB of archives.  
After this operation, 91.1 MB of additional disk space will be used.  
Do you want to continue [Y/n]? y  
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libaio1 armhf 0.3  
.109-3 [8,944 B]
```

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

```
pi@raspberrypi: ~  
Package configuration  
  
##### Configuring mysql-server-5.5 #####  
# While not mandatory, it is highly recommended that you set a password  
# for the MySQL administrative "root" user.  
#  
# If this field is left blank, the password will not be changed.  
#  
# New password for the MySQL "root" user:  
#  
# _____  
#  
# <Ok>  
#####
```

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

```
$ mysql --version
```

```
pi@raspberrypi: ~  
pi@raspberrypi ~$ mysql --version  
mysql Ver 14.14 Distrib 5.5.28, for debian-linux-gnu (armv7l) using readline 6.  
2  
pi@raspberrypi ~$
```

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;
```

```
pi@raspberrypi: ~  
pi@raspberrypi ~$ mysql -u pi  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 43  
Server version: 5.5.28-1 (Debian)  
  
Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| test |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```

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 php5-mysql
```

```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo apt-get install php5 php5-mysql  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  apache2-mpm-prefork libapache2-mod-php5 libmysqlclient18 libonig2 libqdbm14  
  php5-cli php5-common  
Suggested packages:  
  php-pear  
The following packages will be REMOVED:  
  apache2-mpm-worker  
The following NEW packages will be installed:  
  apache2-mpm-prefork libapache2-mod-php5 libmysqlclient18 libonig2 libqdbm14  
  php5 php5-cli php5-common php5-mysql  
0 upgraded, 9 newly installed, 1 to remove and 84 not upgraded.  
Need to get 6,407 kB of archives.  
After this operation, 19.8 MB of additional disk space will be used.  
Do you want to continue [Y/n]? y  
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main php5 all 5.4.4-9  
[1,020 B]  
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main apache2-mpm-prefo  
rk armhf 2.2.22-12 [2,354 B]  
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libmysqlclient18  
armhf 5.5.28+dfsg-1 [631 kB]
```

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
```

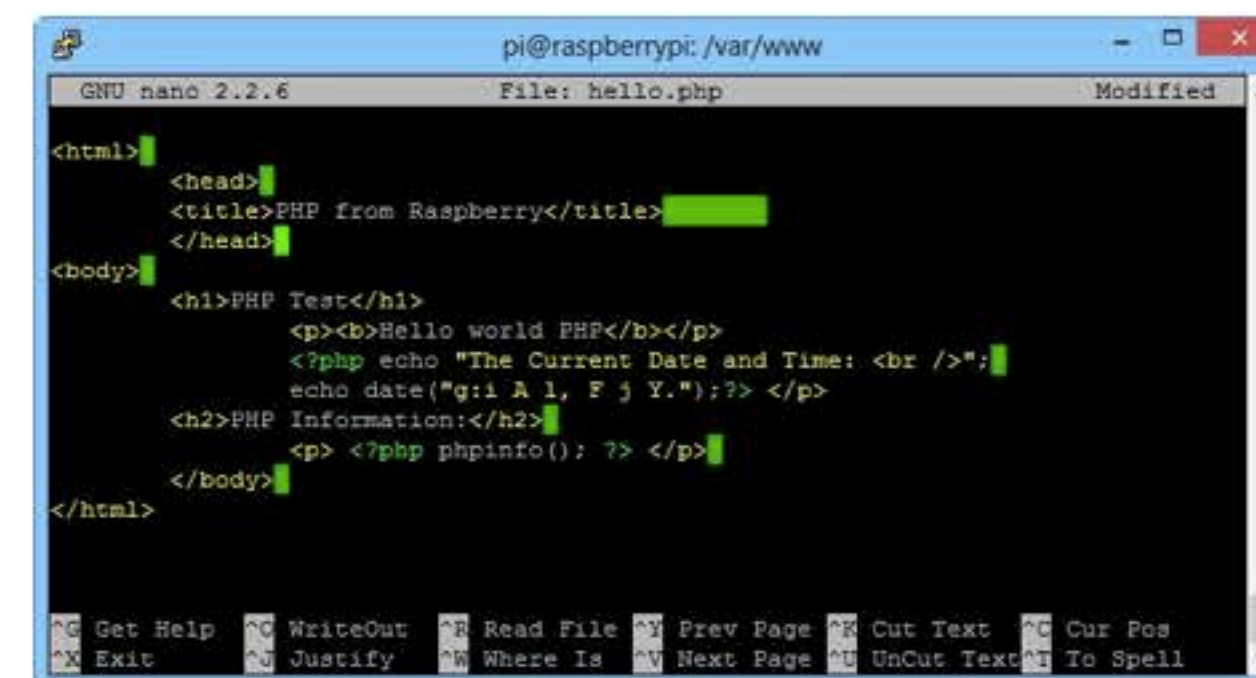
Then, write this code.

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



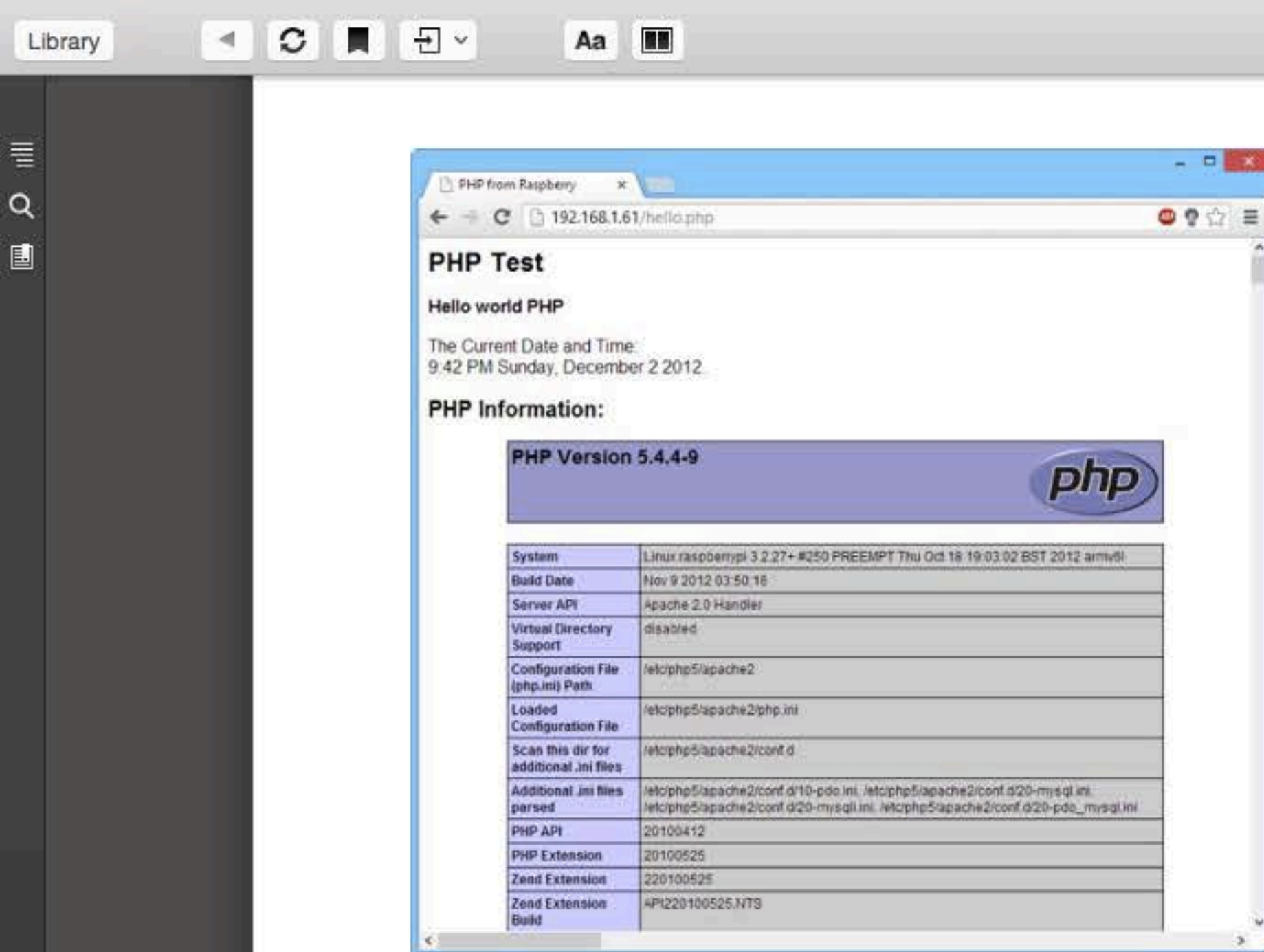
```
Library  ◀  ↻  📖  📄  Aa  🖨️

<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 1, F j Y.");?> </p>
    <h2>PHP Information:</h
2>
        <p> <?php phpinfo
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.php
```

Write all code below.

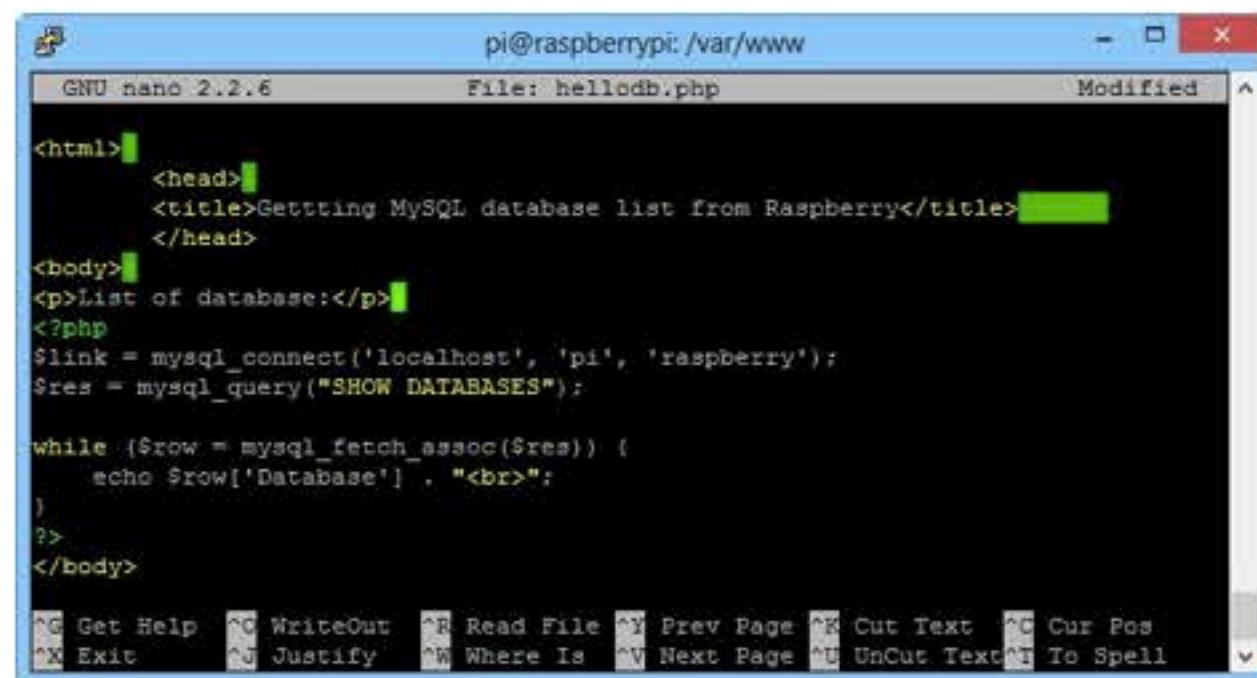
```
<html>
    <head>
        <title>Gettting MySQL d
atabase list from Raspberry</ti
tle>
    </head>

    <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>
```

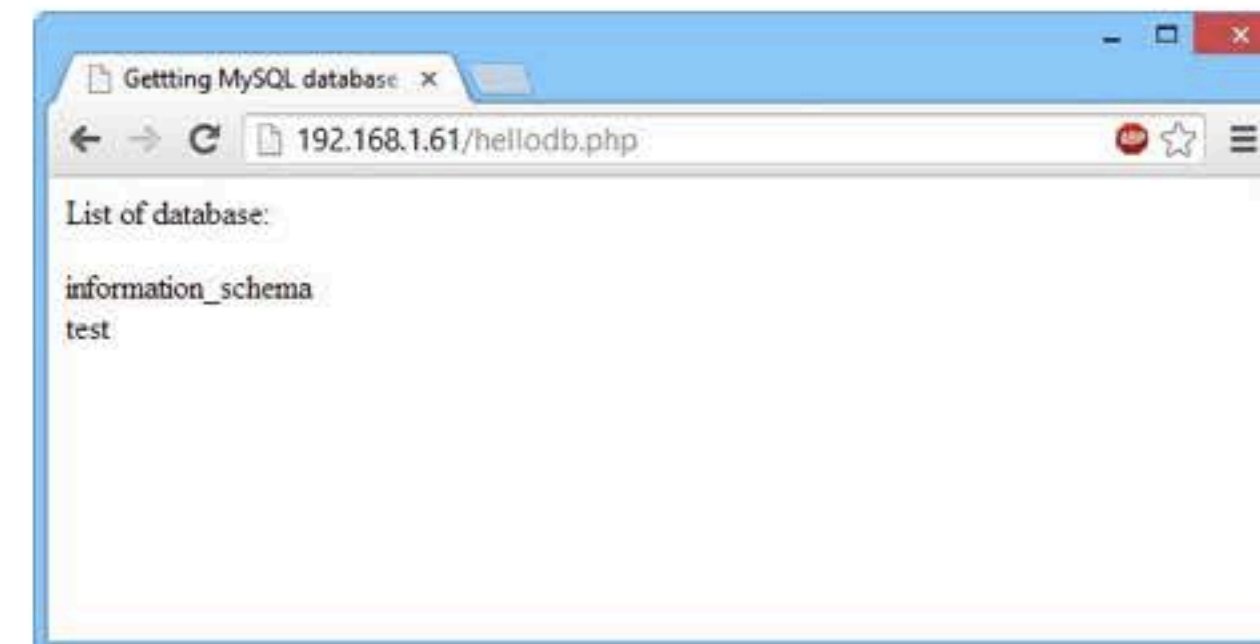
Note: change username and password for

MySQL.



```
pi@raspberrypi: /var/www
GNU nano 2.2.6 File: hellodb.php Modified
<html>
  <head>
    <title>Gettting MySQL database list from Raspberry</title>
  </head>
  <body>
    <p>List of database:</p>
    <?php
$link = mysql_connect('localhost', 'pi', 'raspberry');
$res = mysql_query("SHOW DATABASES");

while ($row = mysql_fetch_assoc($res)) {
    echo $row['Database'] . "<br>";
}
?>
  </body>
</html>
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```



Save it.

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

6. Raspberry Pi Programming

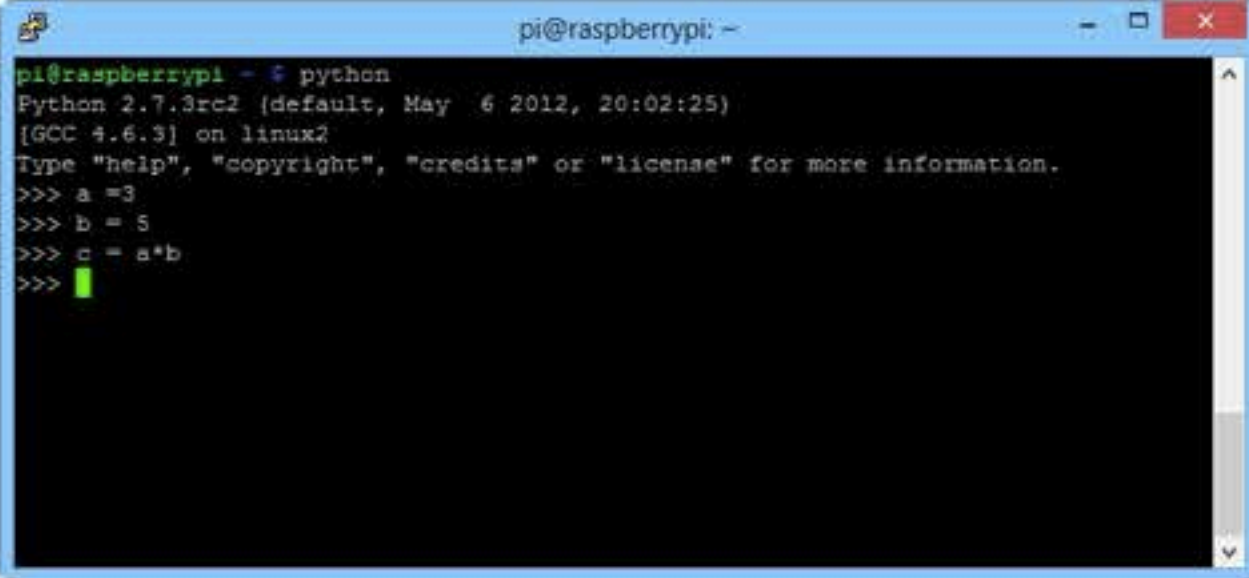
This chapter explains how to work with development environment on Raspberry Pi 2.

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.

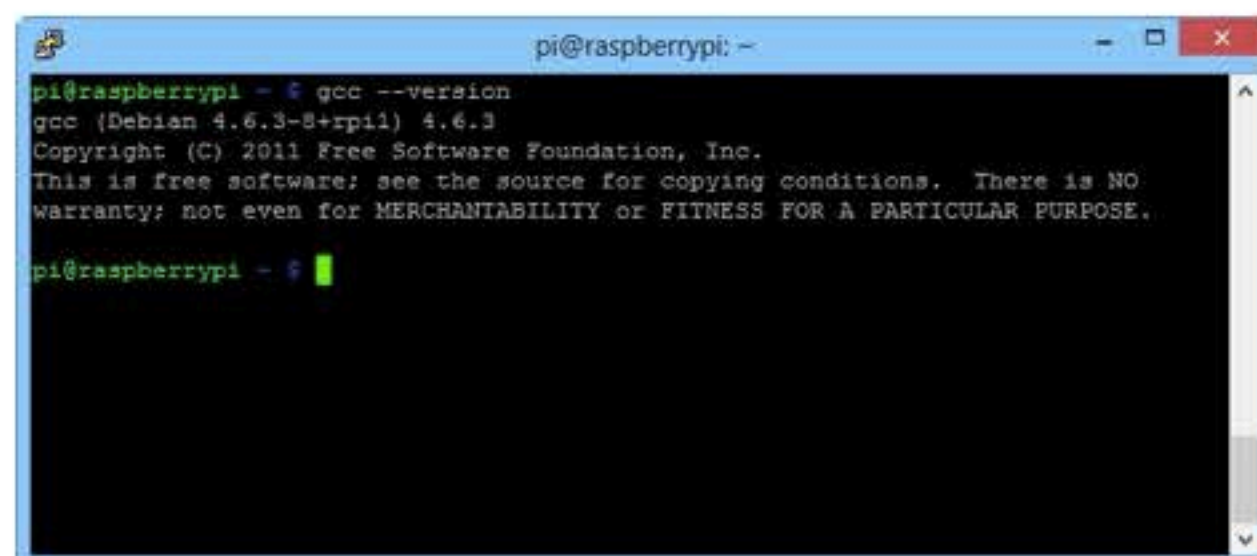


```
pi@raspberrypi: ~  
pi@raspberrypi ~$ python  
Python 2.7.3rc2 (default, May  6 2012, 20:02:25)  
[GCC 4.6.3] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> a = 3  
>>> b = 5  
>>> c = a*b  
>>>
```

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
```


A terminal window titled 'pi@raspberrypi: ~' showing the output of the command 'gcc --version'. The output displays the GCC version as 4.6.3-8+rp11 and includes copyright information for the Free Software Foundation.

```
pi@raspberrypi ~$ gcc --version
gcc (Debian 4.6.3-8+rp11) 4.6.3
Copyright (C) 2011 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

pi@raspberrypi ~$
```

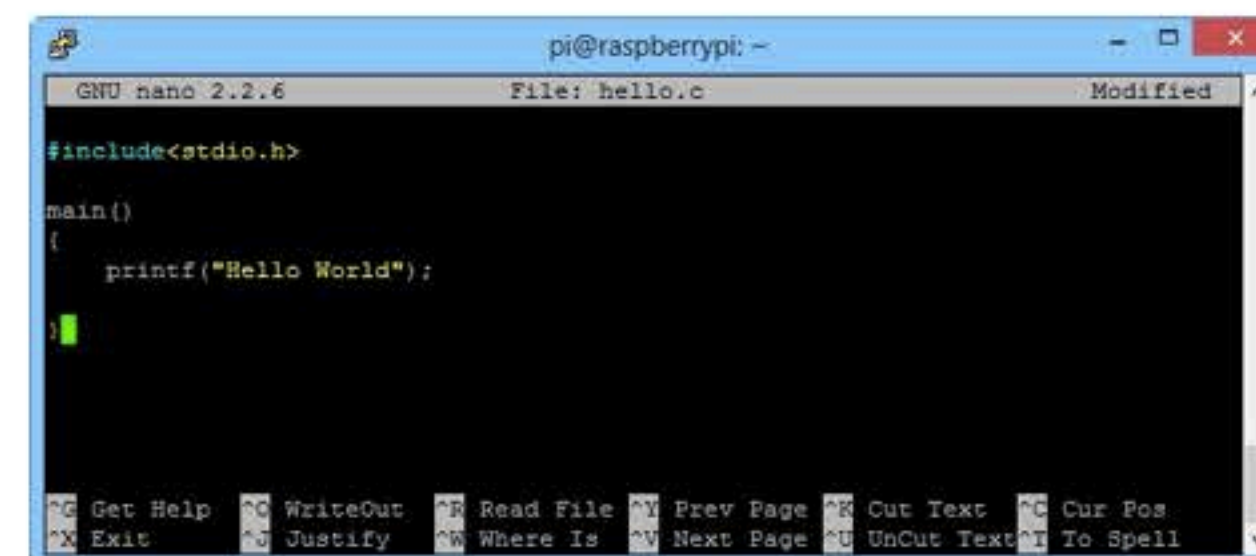
For illustration, we create a simple code, hello world. Create a file

```
$ nano hello.c
```

Write this code.

```
#include<stdio.h>

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

A nano editor window titled 'pi@raspberrypi: ~' editing 'File: hello.c'. The code inside is a simple C program that prints 'Hello World'. The bottom status bar shows various keyboard shortcuts like ^G Get Help, ^O WriteOut, etc.

```
GNU nano 2.2.6 File: hello.c Modified
#include<stdio.h>

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

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Save it.

Now you can compile C code using GCC.

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

To run, you type this script.

```
$ ./hello
```

```
pi@raspberrypi ~  
pi@raspberrypi ~$ gcc --version  
gcc (Debian 4.6.3-8+rpi1) 4.6.3  
Copyright (C) 2011 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
  
pi@raspberrypi ~$ ls  
Desktop  python_games  
pi@raspberrypi ~$ nano hello.c  
pi@raspberrypi ~$ gcc hello.c -o hello  
pi@raspberrypi ~$ ./hello  
Hello Worldpi@raspberrypi ~$
```

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
```

```
pi@raspberrypi ~$ sudo apt-get install nodejs  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  libc-ares2 libev4 libv8-3.8.9.20  
The following NEW packages will be installed:  
  libc-ares2 libev4 libv8-3.8.9.20 nodejs  
0 upgraded, 4 newly installed, 0 to remove and 84 not upgraded.  
Need to get 2,123 kB of archives.  
After this operation, 5,650 kB of additional disk space will be used.  
Do you want to continue [Y/n]? y  
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libc-ares2 armhf  
1.9.1-3 [69.0 kB]  
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libv8-3.8.9.20 ar  
mhf 3.8.9.20-2+rpi1 [1,256 kB]  
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libev4 armhf 1:4.  
11-1 [35.3 kB]  
Get:4 http://mirrordirector.raspbian.org/raspbian/ wheezy/main nodejs armhf 0.6.  
19-dfsg1-2 [763 kB]  
Fetched 2,123 kB in 4s (512 kB/s)  
Selecting previously unselected package libc-ares2:armhf.  
(Reading database ... 58945 files and directories currently installed.)  
Unpacking libc-ares2:armhf (from .../libc-ares2_1.9.1-3_armhf.deb) ...
```

Try to check the node.js version

```
$ node -v
```

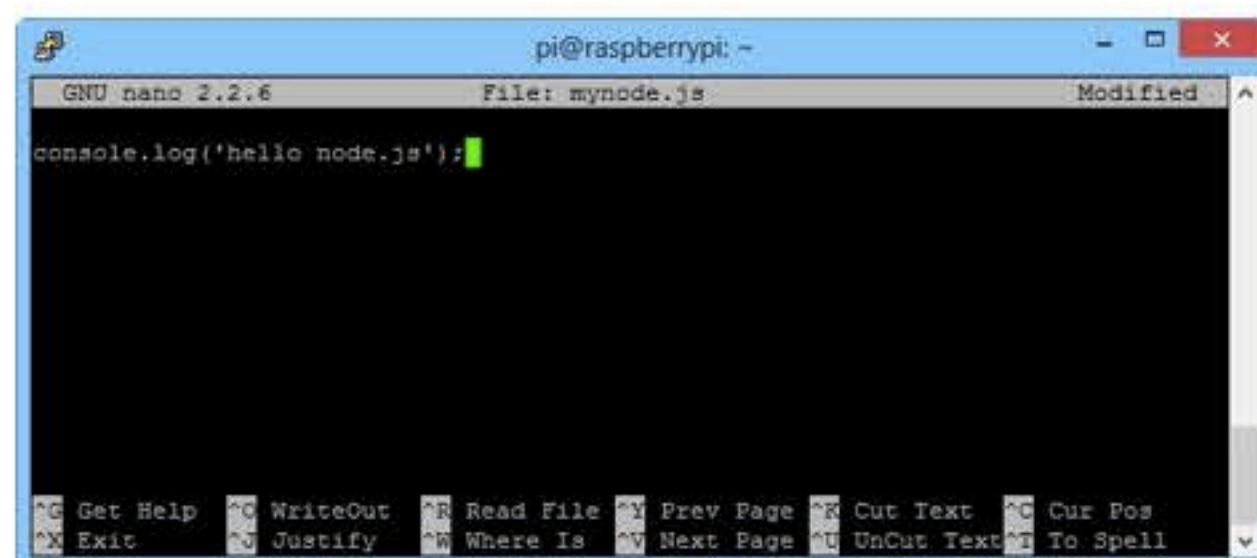
```
pi@raspberrypi ~$ node -v  
v0.6.19  
pi@raspberrypi ~$
```


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

```
$ nano mynode.js
```

and write this code

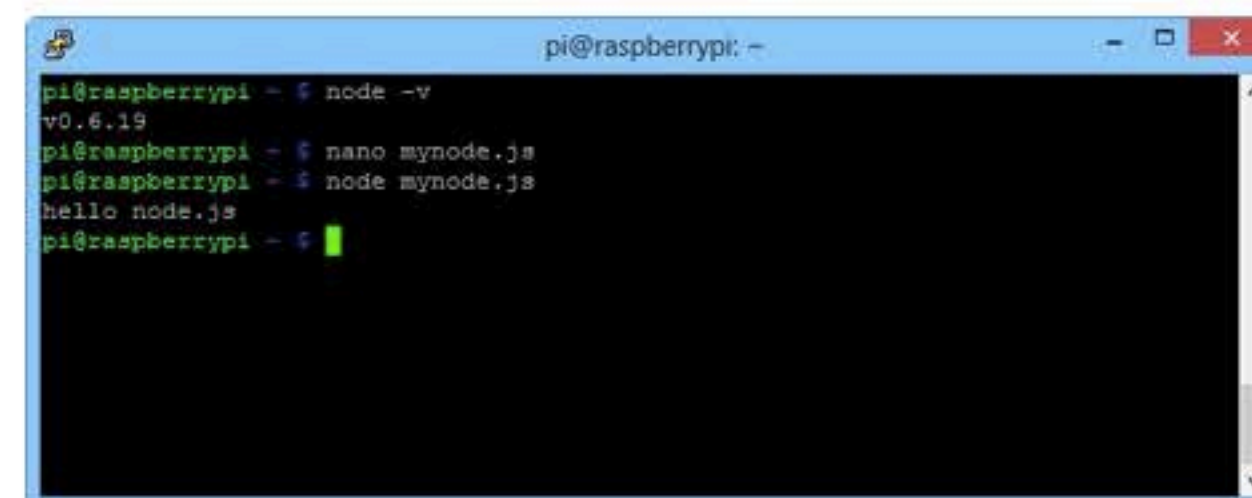
```
console.log('hello node.js');
```



Save it.

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

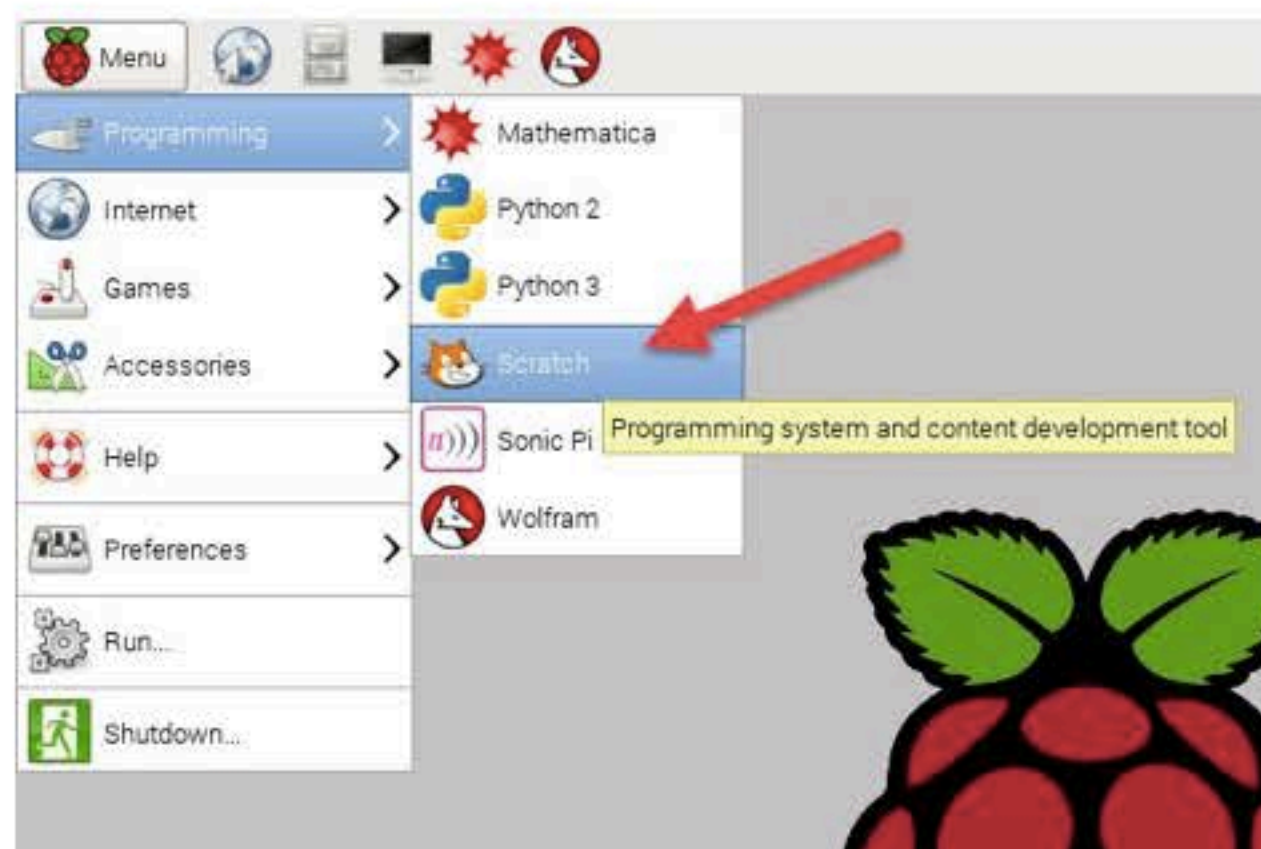
```
$ node mynode.js
```



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).



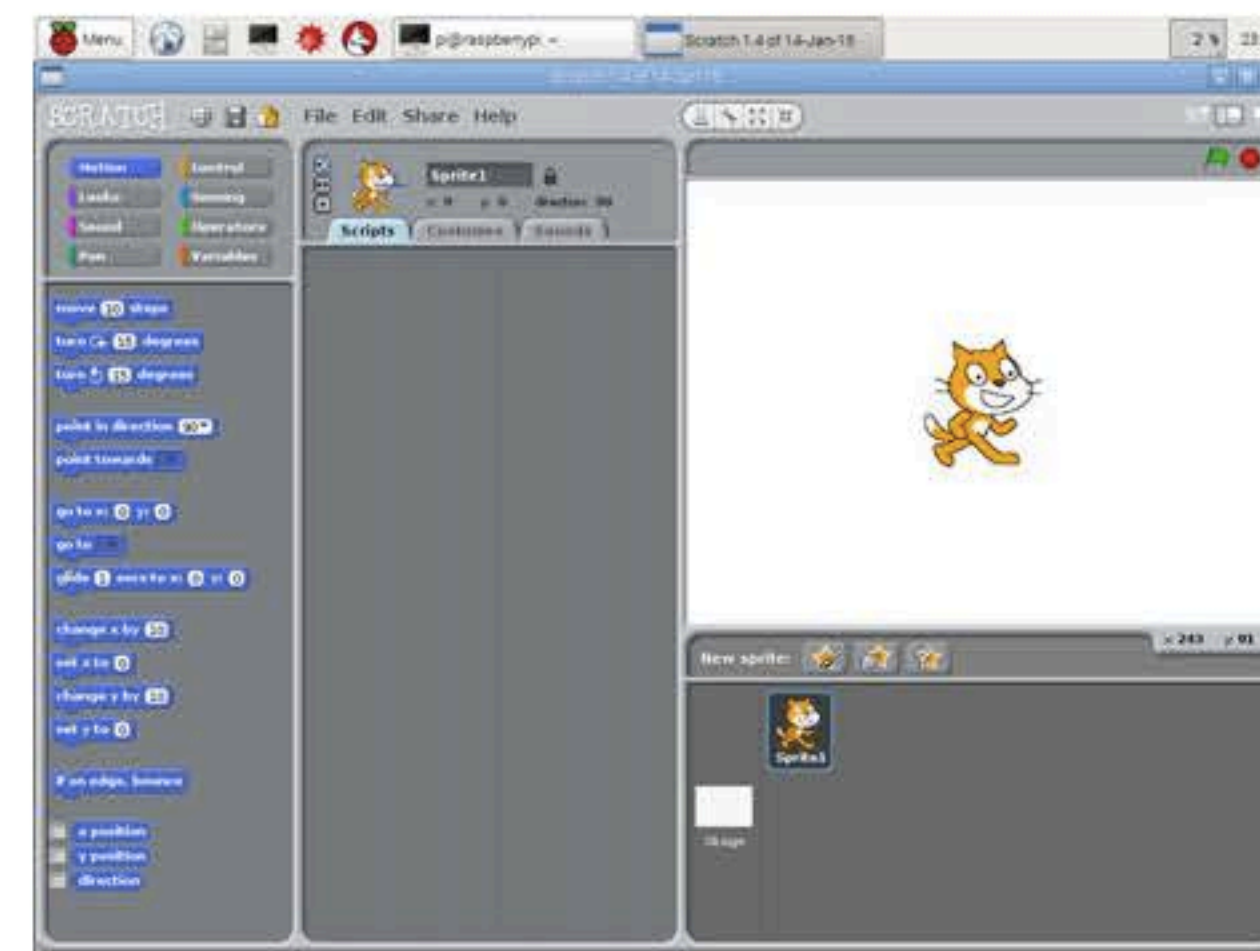
After that, you will get a Scratch application.

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.

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev 1
29/01/2014

<http://www.element14.com>

(source:
<http://www.element14.com/community/docs/DOC-73950/1/raspberry-pi-2-model-b-gpio-40-pin-block-pinout>)

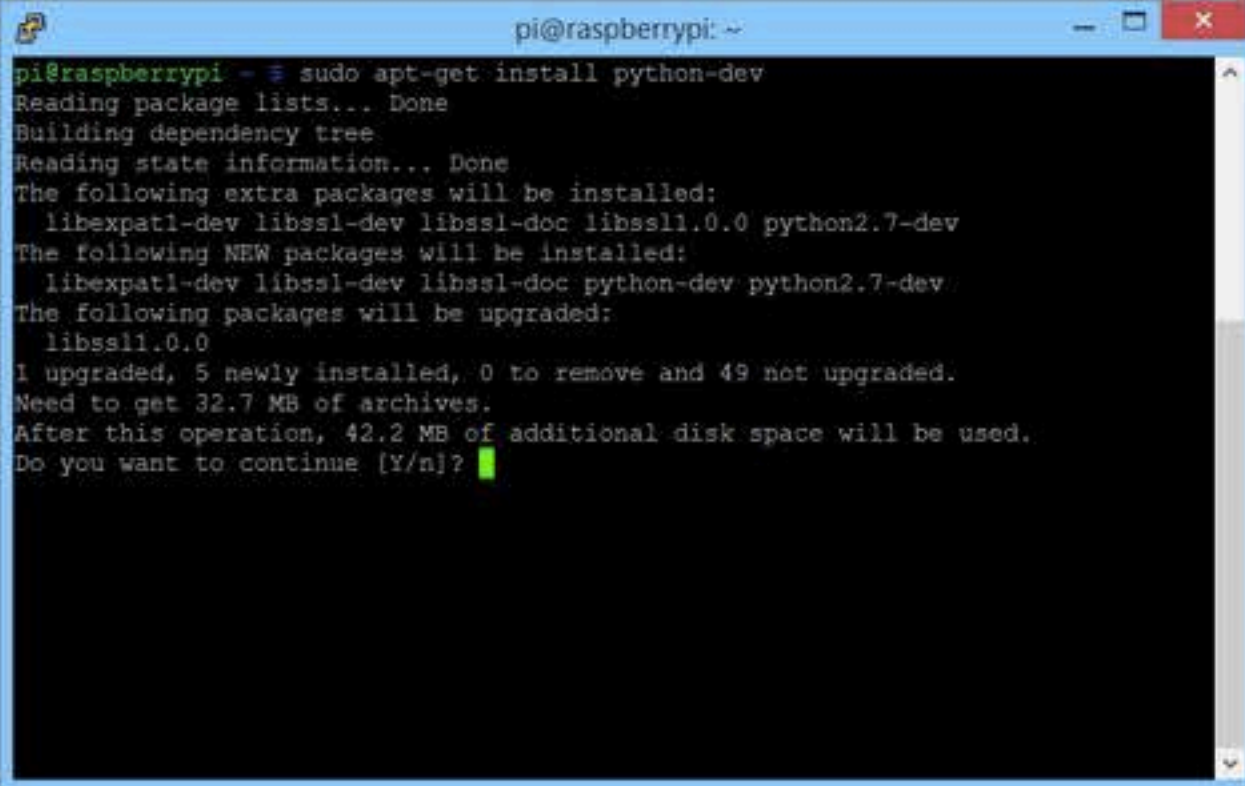
7.2 Accessing GPIO

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.



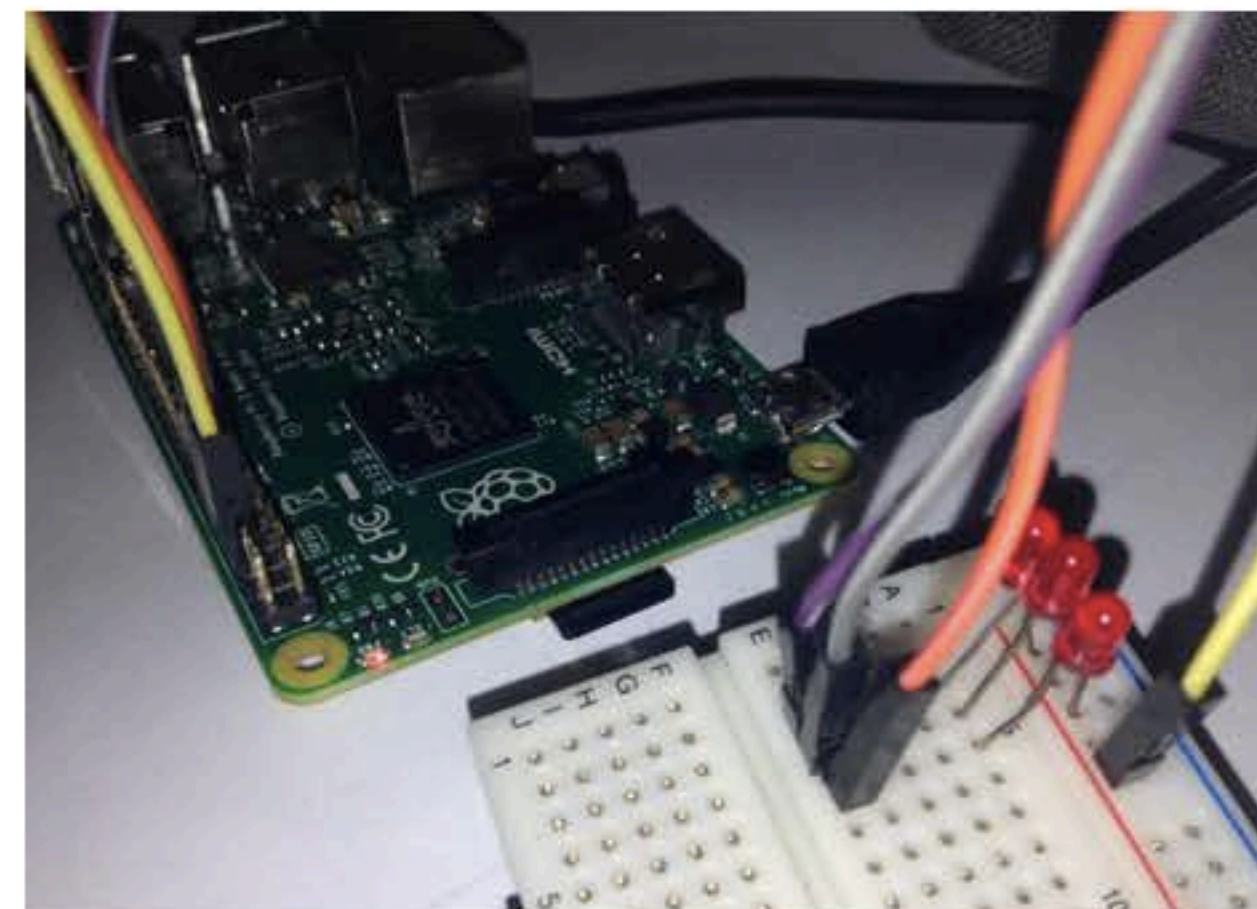
```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo apt-get install python-dev  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  libexpat1-dev libssl-dev libssl-doc libssl1.0.0 python2.7-dev  
The following NEW packages will be installed:  
  libexpat1-dev libssl-dev libssl-doc python-dev python2.7-dev  
The following packages will be upgraded:  
  libssl1.0.0  
1 upgraded, 5 newly installed, 0 to remove and 49 not upgraded.  
Need to get 32.7 MB of archives.  
After this operation, 42.2 MB of additional disk space will be used.  
Do you want to continue [Y/n]?
```

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)

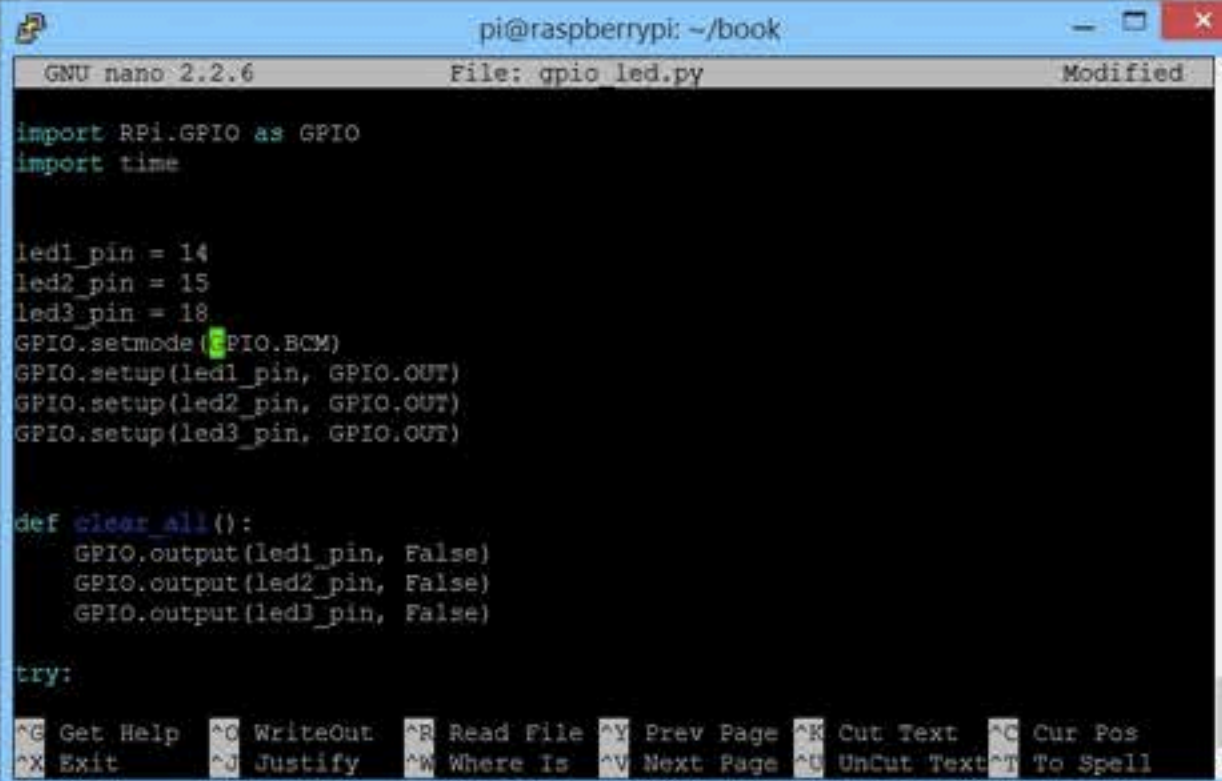
        time.sleep(2)
        clear_all()
```

```
        print("turn 3 on")
        GPIO.output(led3_pin, T
rue)

        time.sleep(2)
        clear_all()

except KeyboardInterrupt:
    GPIO.cleanup()

print("done")
```



```
pi@raspberrypi: ~/book
GNU nano 2.2.6 File: gpio_led.py Modified
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, True)

        time.sleep(2)
        clear_all()
        print("turn 2 on")
        GPIO.output(led2_pin, True)

        time.sleep(2)
        clear_all()
        print("turn 3 on")
        GPIO.output(led3_pin, True)

        time.sleep(2)
        clear_all()

except KeyboardInterrupt:
    GPIO.cleanup()

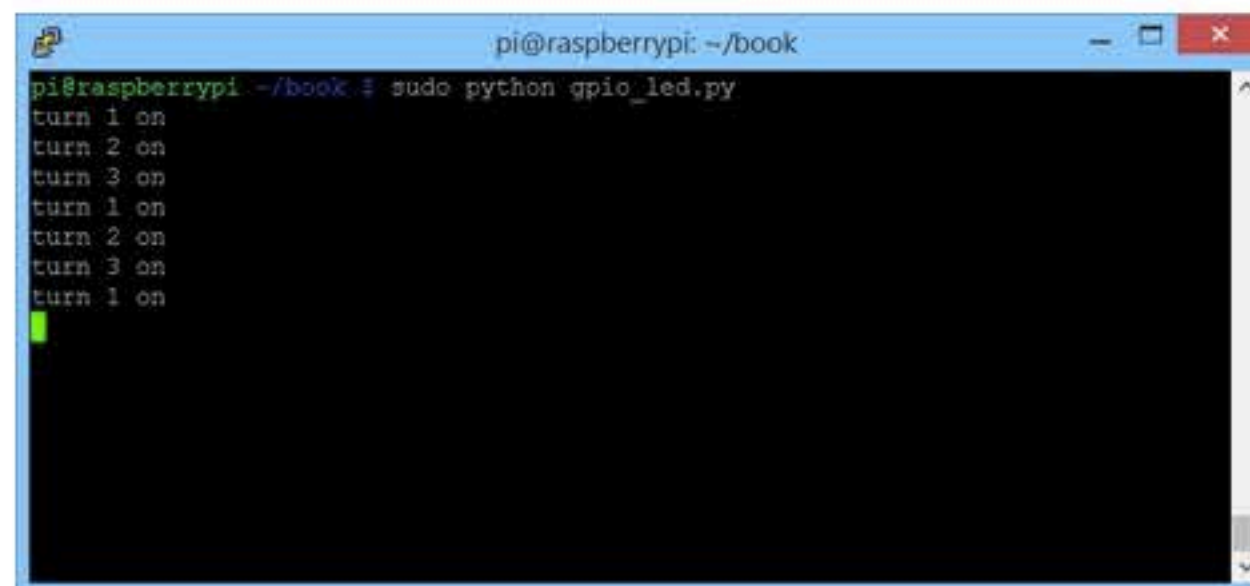
print("done")
```

Save this code.

You can run the application using the following command.

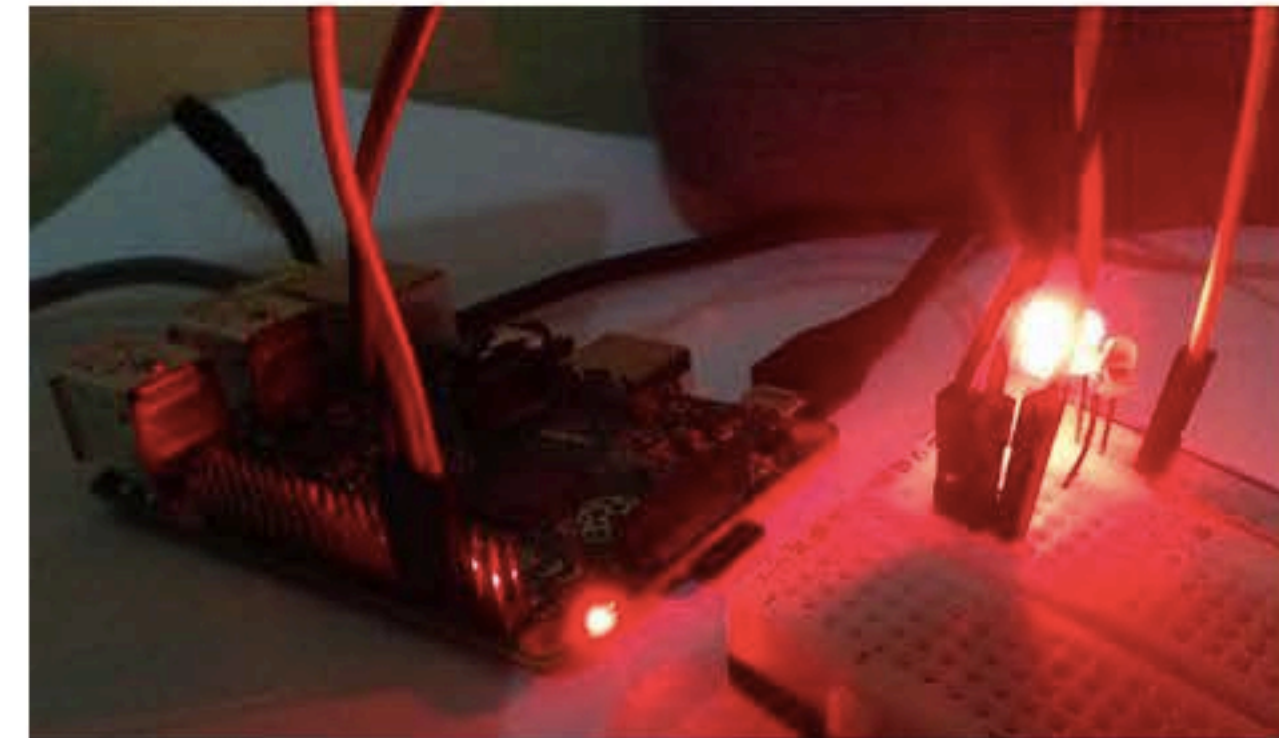
```
$ sudo python gpio_led.py
```

If success, you can see the output as below.



```
pi@raspberrypi: ~/book
pi@raspberrypi ~/book $ sudo python gpio_led.py
turn 1 on
turn 2 on
turn 3 on
turn 1 on
turn 2 on
turn 3 on
turn 1 on
```

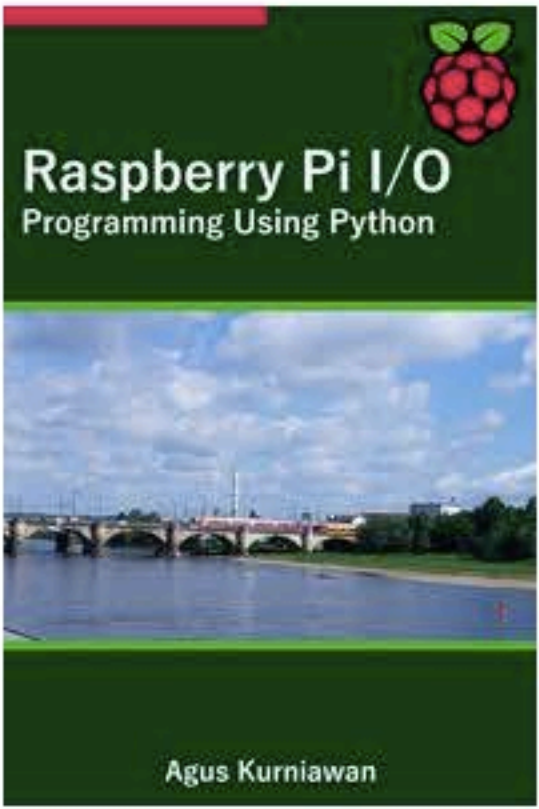
Three LEDs also are blinking.



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> .



8. Raspberry Pi 2 Serial Debugging

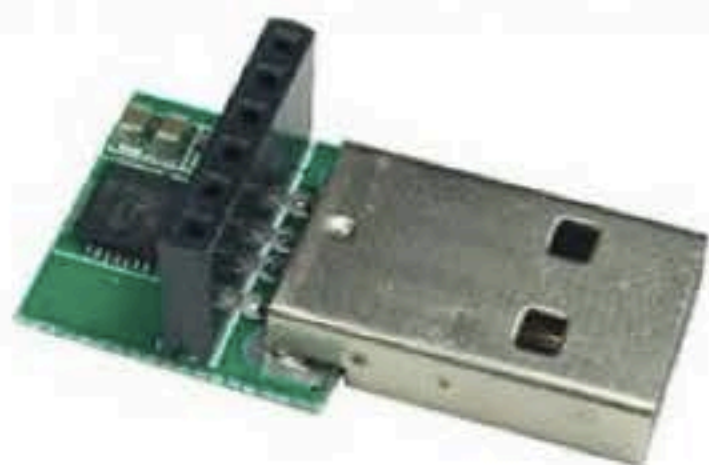
This chapter explains how to access Raspberry Pi 2 via Serial Port.

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>.

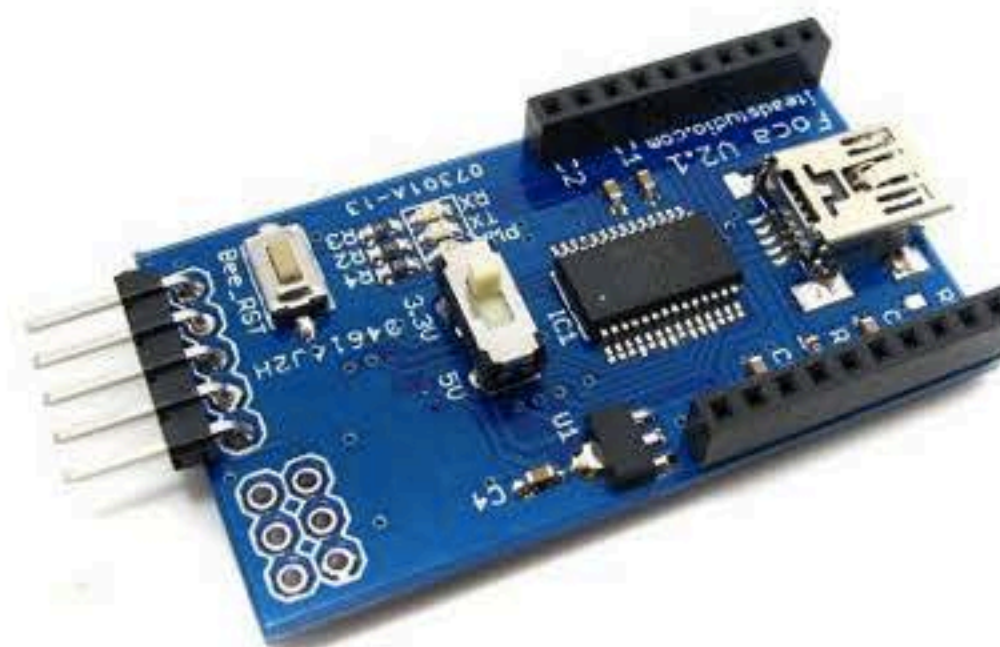


Another device, you can buy USB to TTL on Cooking-Hacks, <http://www.cooking-hacks.com/index.php/usb-to-ttl-converter-cp210.html>.



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>

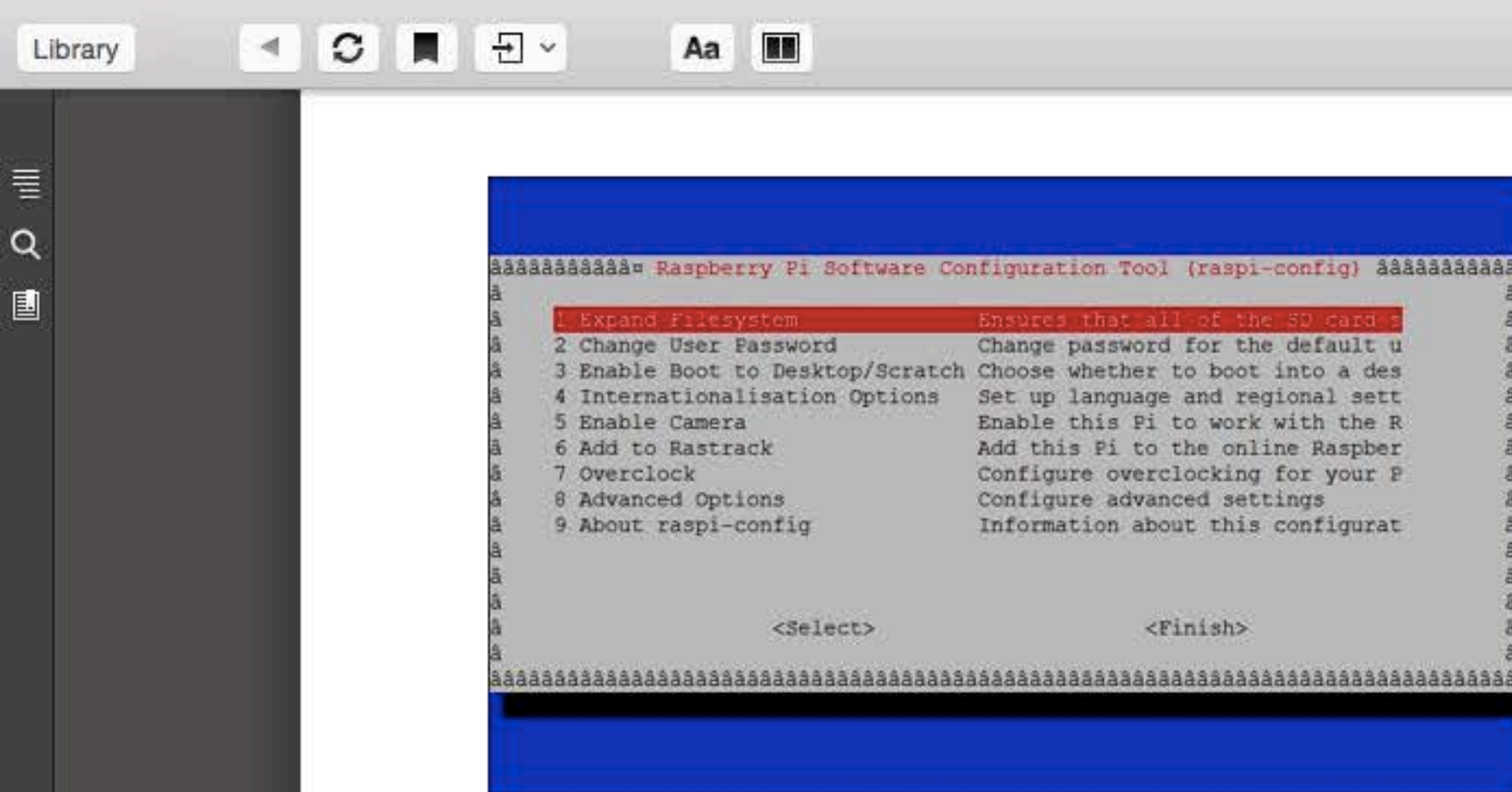


8.2 Enabling Serial Debugging

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

```
$ sudo raspi-config
```

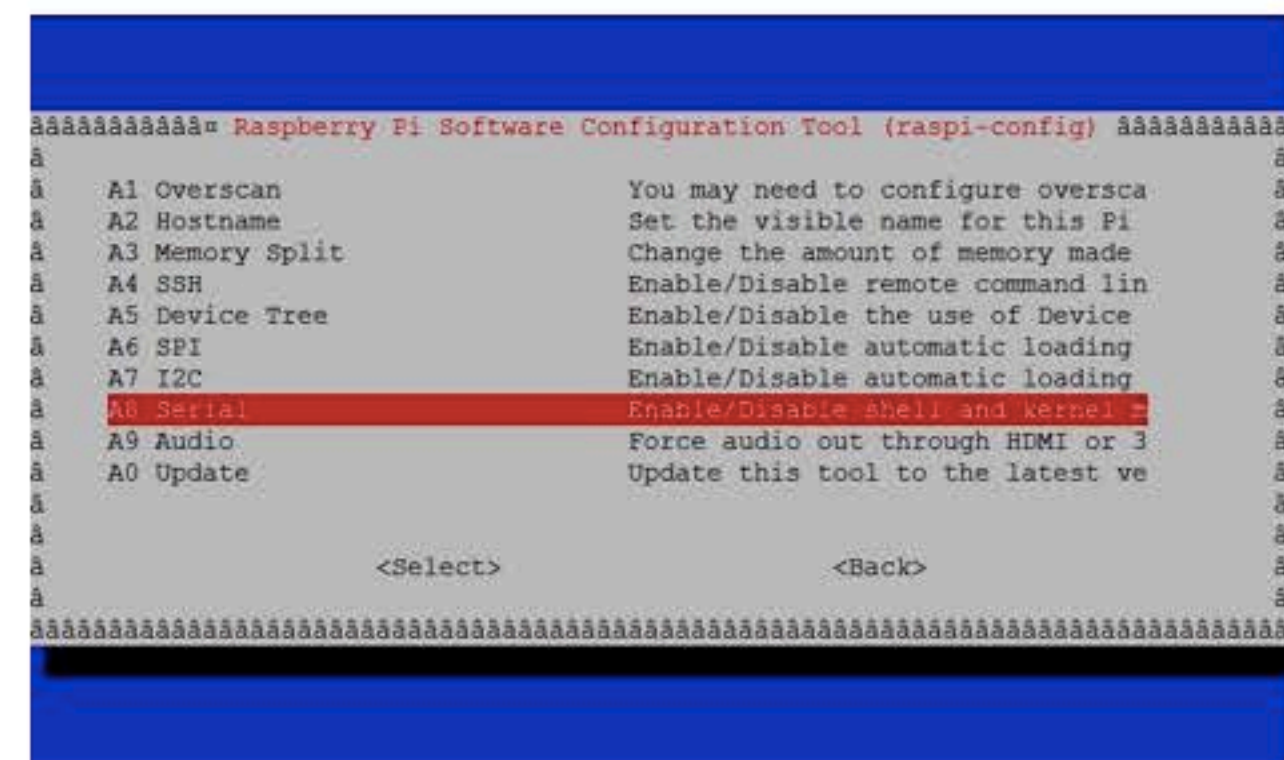
You will get a dialog. Select **8 Advanced Options** menu.



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
```

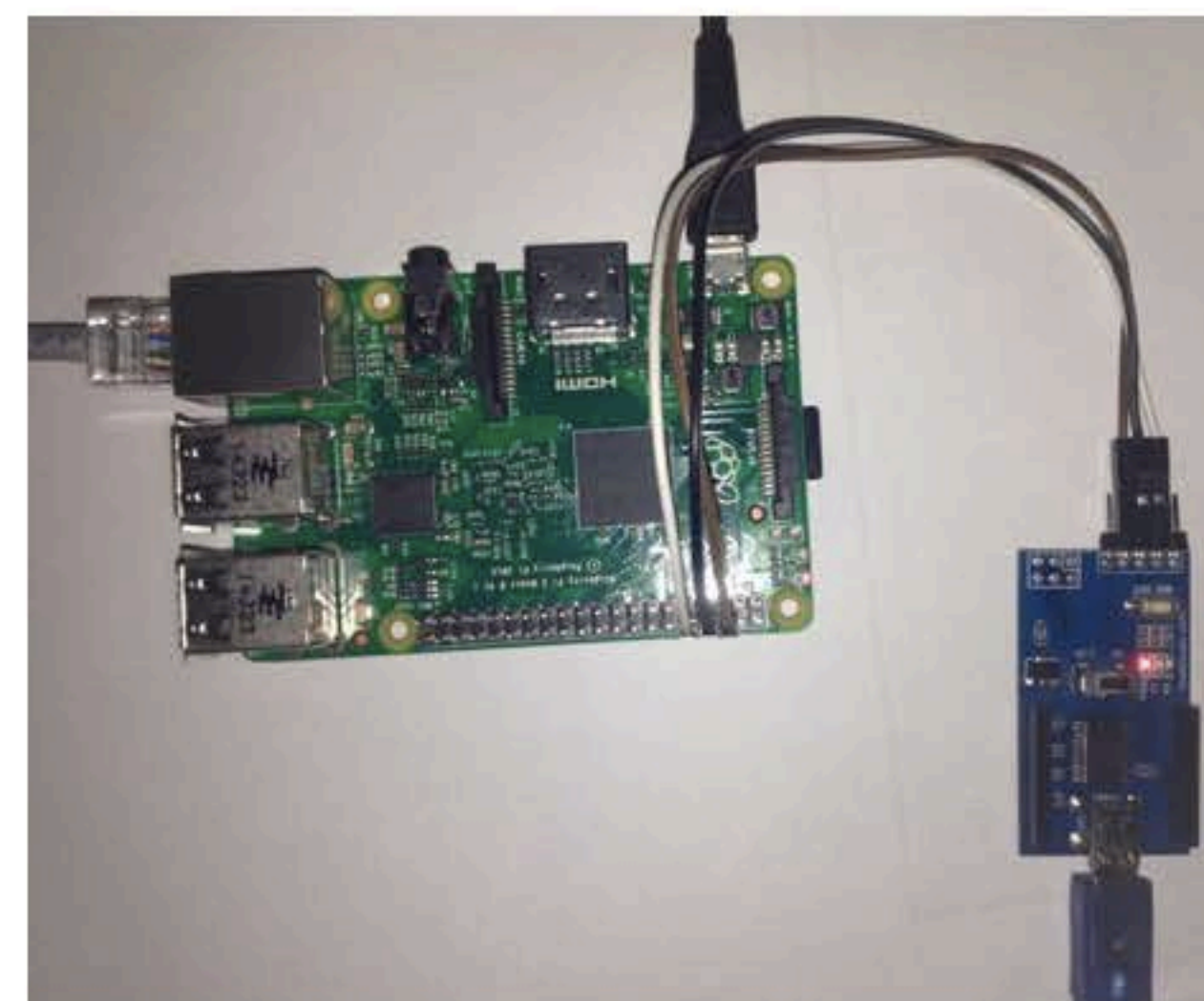
8.3 Wiring

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.

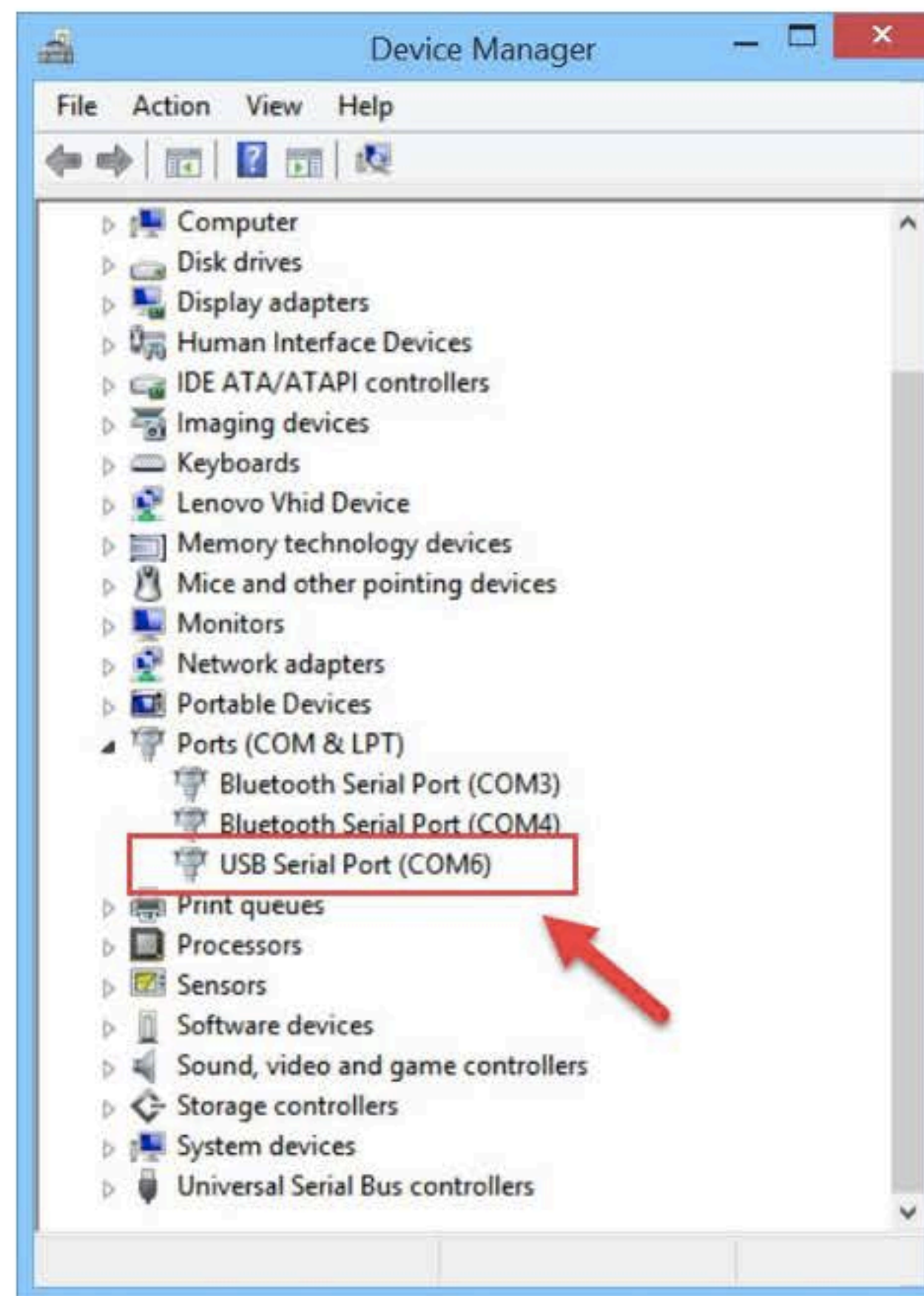


Now your USB cable of USB TTL device is be connected to your computer. You can use any

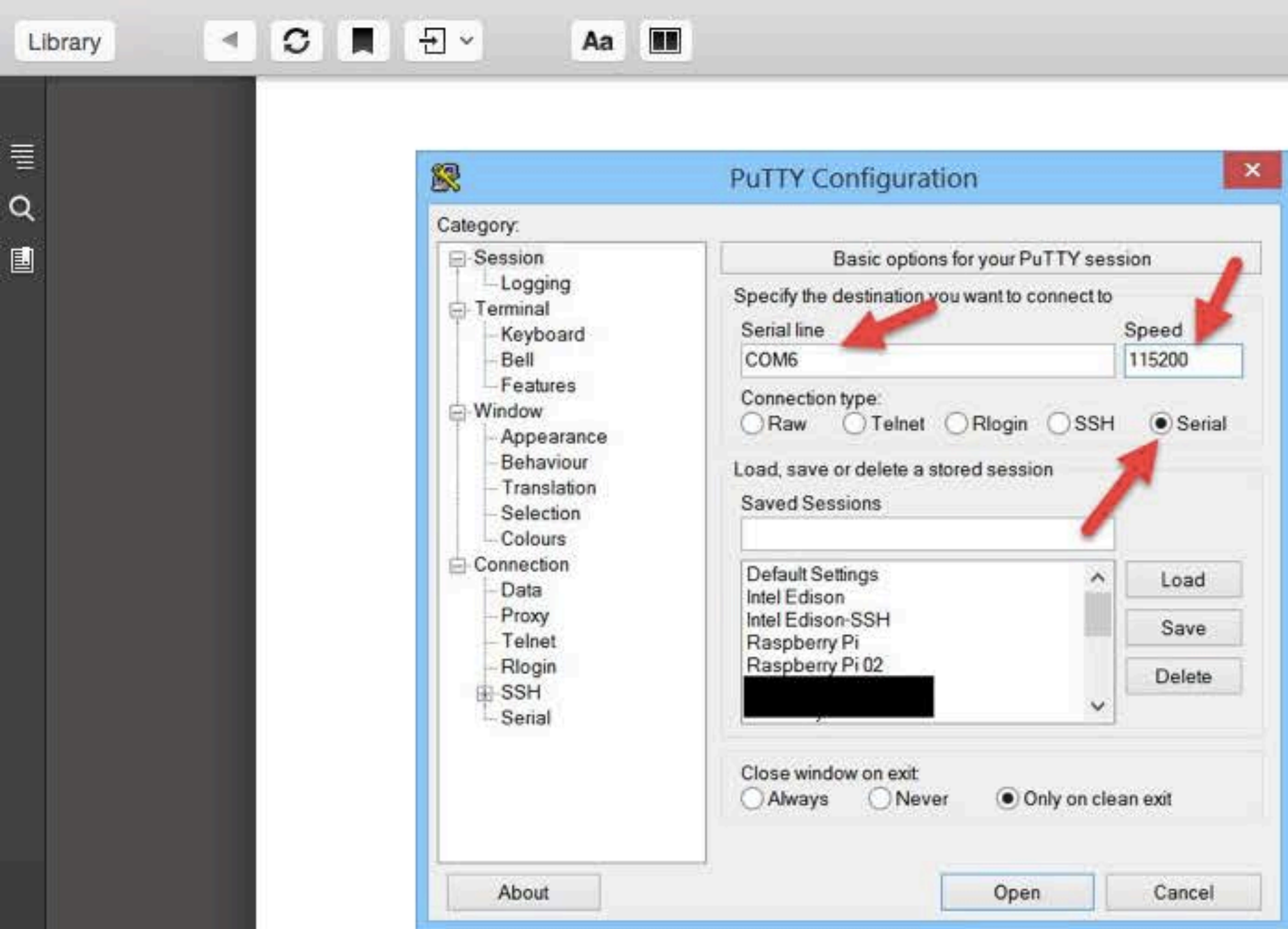
serial application to execute.

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

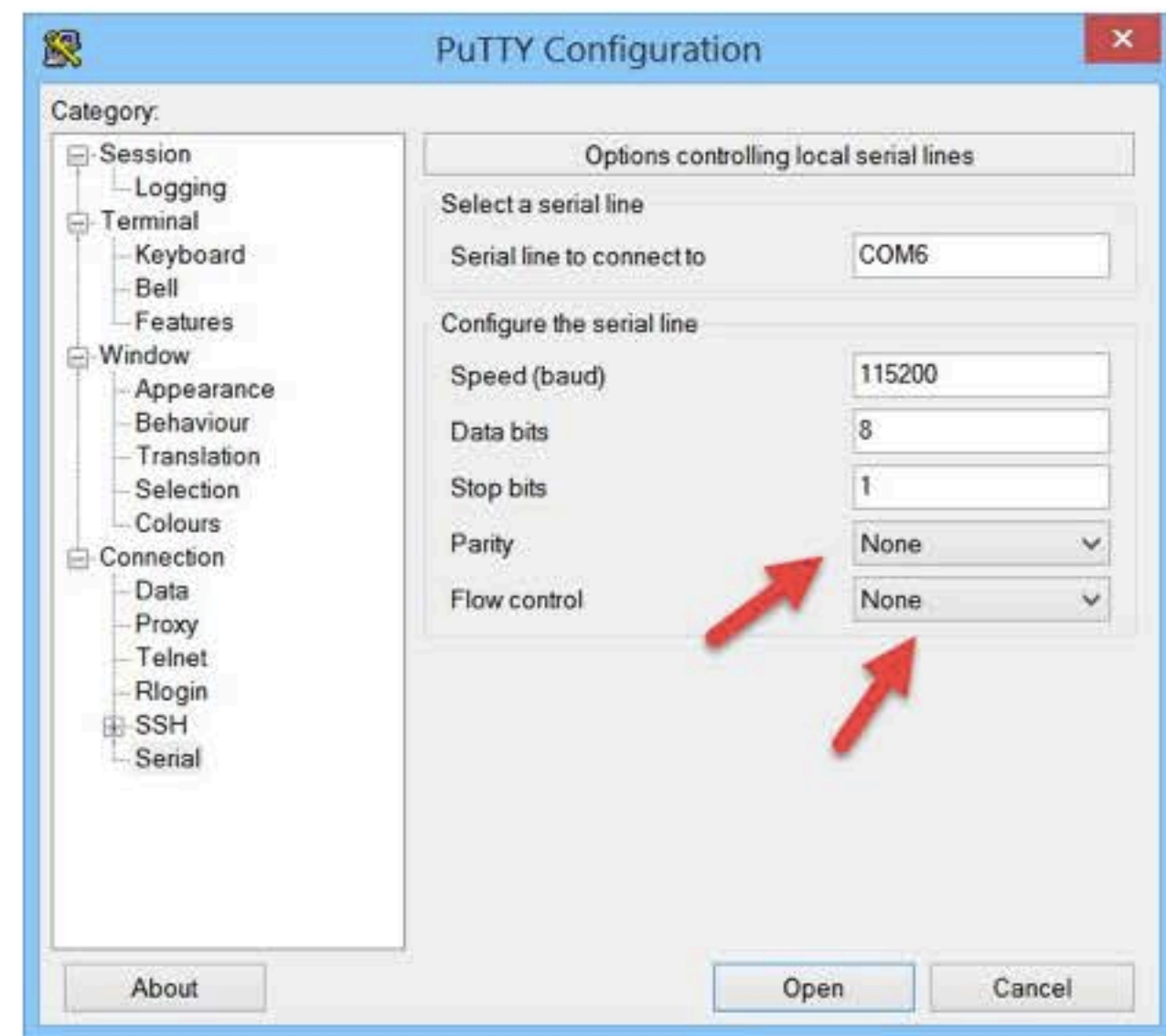
Run PuTTY and choose Serial for connection type. Fill Serial line name, for instance, my Windows detected it on COM6 as below.



Set 115200 for speed serial.

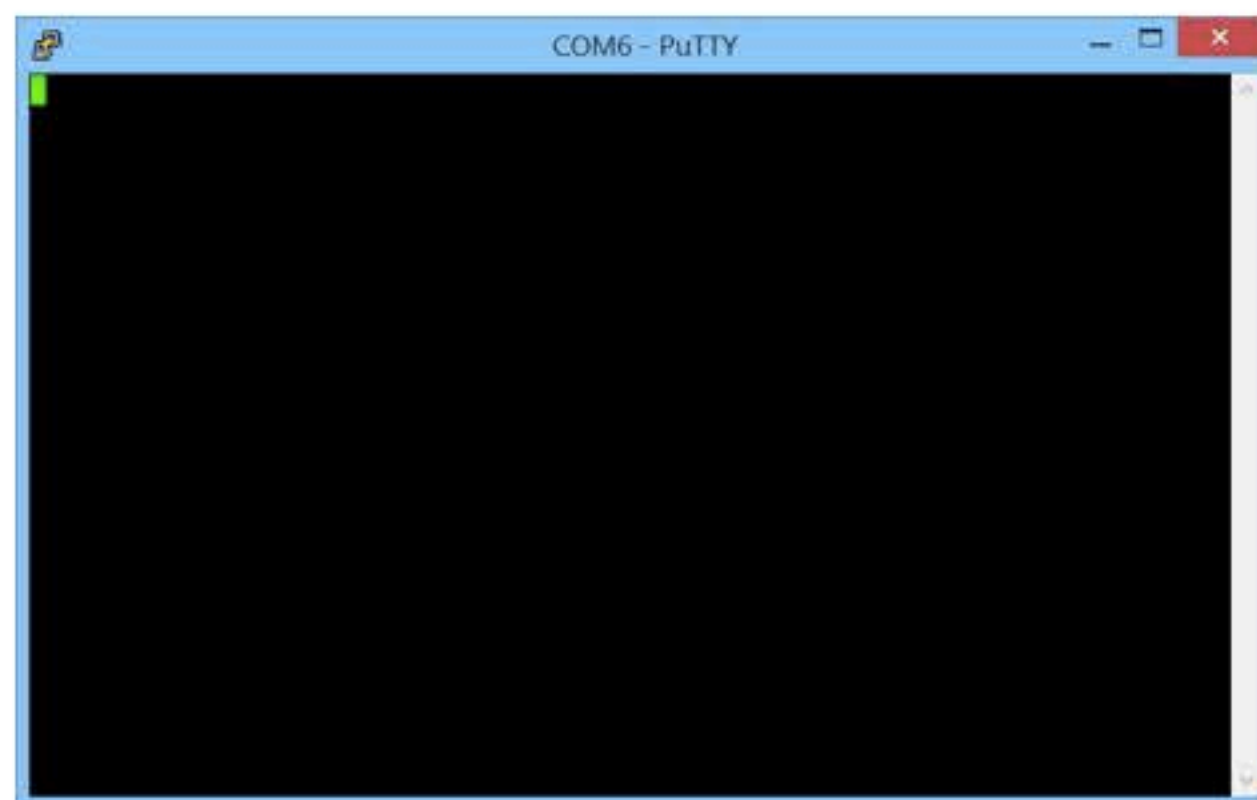


Click **Serial** on side menu and choose **None** for Parity and Flow control.



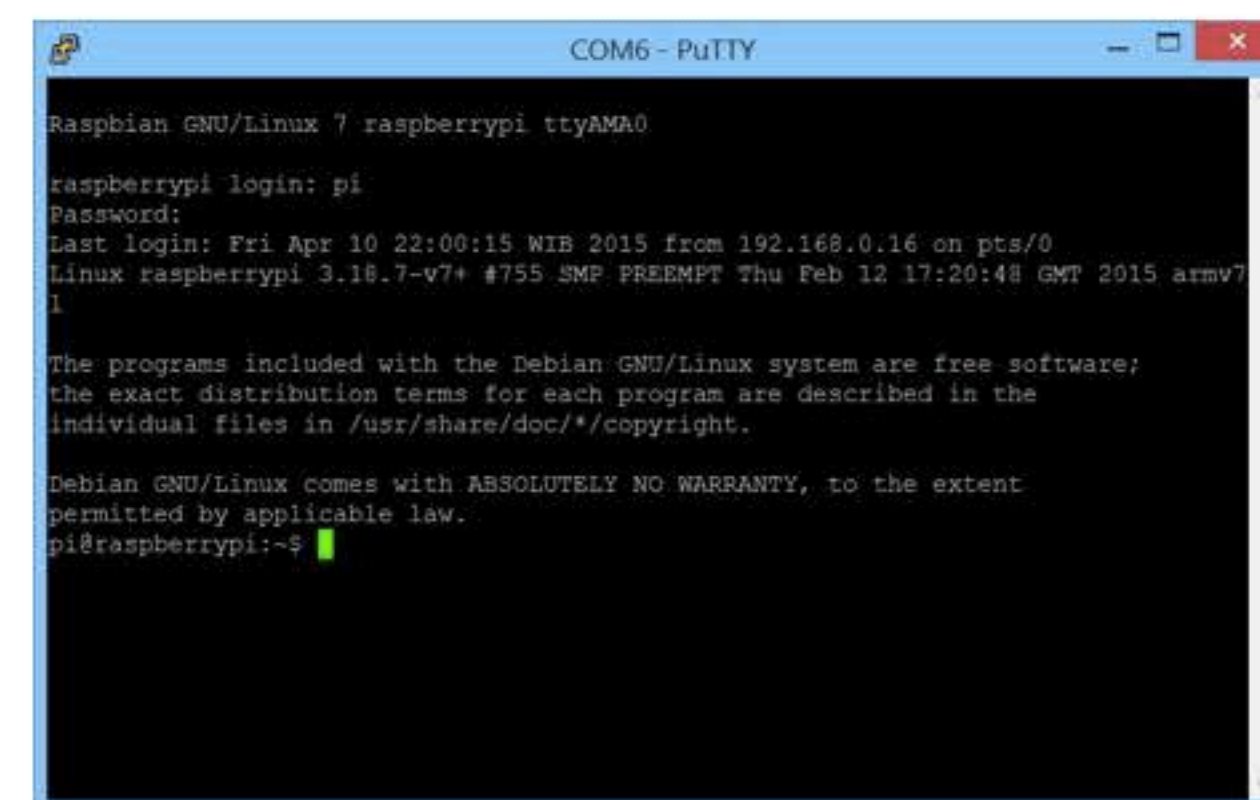
8.4 Testing

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

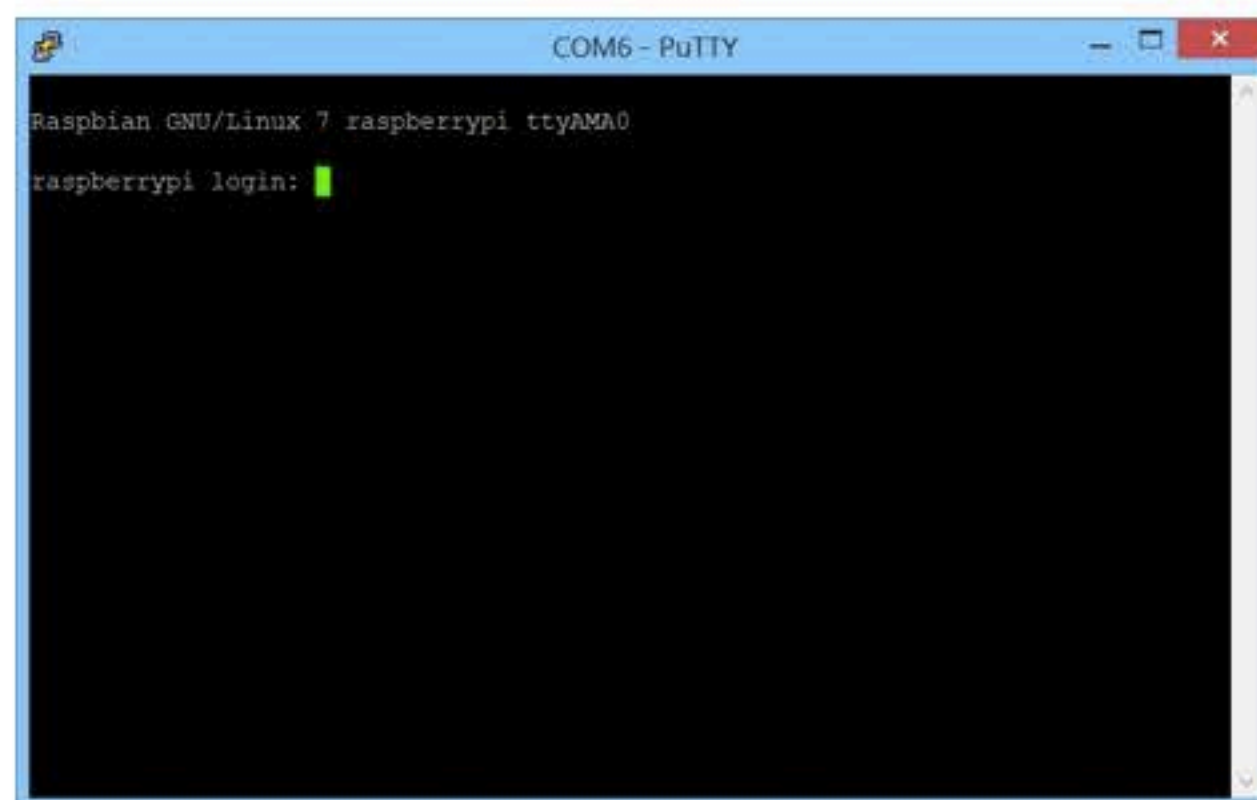


Then, try to logon to Raspberry Pi.

Here is a sample of serial debugging output.



If success, you will get the authentication form.



Contact

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