

# Getting Started with Raspberry Pi

# 3



Agus Kurniawan

# **Copyright**

Getting Started with Raspberry Pi 3

Agus Kurniawan

1st Edition, 2016

Copyright © 2016 Agus Kurniawan

\* Raspberry Pi Logo is trademark and copyright from Raspberry Pi Foundation, <https://www.raspberrypi.org> .

# Table of Contents

[Copyright](#)

[Preface](#)

[1. Introduction to Raspberry Pi 3](#)

[1.1 Raspberry Pi 3](#)

[1.2 Getting Hardware](#)

[1.3 Unboxing](#)

[2. Operating System](#)

[2.1 Raspberry Pi 3 Operating System](#)

[2.2 Preparation](#)

[2.2.1 Setup MicroSD Card](#)

[3. Powering Up and Running](#)

[3.1 Put Them All!](#)

[3.2 Expanding File System](#)

[3.3 Configure Timezone](#)

[3.4 Configure Keyboard](#)

[3.5 Rebooting](#)

[3.6 Shutdown](#)

[3.7 Change Password](#)

[3.8 Configure All Settings](#)

[4. Connecting to a Network](#)

[4.1 Getting Started](#)

## 4.2 Attaching a Network Module

## 4.3 Connecting to a Network

### 4.3.1 Connecting a Network via Ethernet

### 4.3.2 Connecting a Network via WiFi

## 4.4 Auto Connect to WiFi

## 4.5 Configuring Static IP Address

## 4.6 Browsing Internet

## 4.7 SSH

## 4.8 Update Package Repository

## 4.9 Upgrading Package and Firmware

## 4.10 Remote Desktop

# 5. Raspberry Pi Programming

## 5.1 Getting Started

## 5.2 Python

## 5.3 C/C++

## 5.4 Node.js

## 5.5 Scratch

## 5.6 Wolfram Mathematica

## 5.7 Java

# 6. Working with Bluetooth and iBeacon

## 6.1 Getting Started

## 6.2 Setting up Bluetooth

## 6.3 Ping Bluetooth

## 6.4 Working with Blueman Bluetooth Manager

6.4.1 Setting up

6.4.2 Pairing

6.4.3 Transferring File

## 6.5 Building your own iBeacon

## 7. Deploying LAMP Stack

7.1 Getting Started

7.2 Installing Apache Server

7.3 Installing MySQL

7.4 Installing PHP and MySQL Driver for PHP

7.5 Testing PHP

7.6 Testing PHP and MySQL

## 8. Accessing GPIO

8.1 Introduction to GPIO

8.2 Accessing GPIO

8.3 Demo

8.4 Further Reading

## 9. Raspberry Pi 3 Serial Debugging

9.1 Preparation

9.2 Enabling Serial Debugging

9.3 Wiring

9.4 Testing

Contact



# **Preface**

This book was written to help anyone who wants to get started in Raspberry Pi 3. It describes all the basic elements of the Raspberry Pi 3 with step-by-step approach.

Agus Kurniawan

Berlin, March 2016

# 1. Introduction to Raspberry Pi 3

## 1.1 Raspberry Pi 3

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/>).

Built on the latest Broadcom 2837 ARMv8 64bit processor the new generation Raspberry Pi 3 Model B is faster and more powerful than its predecessors. It has improved power management to support more powerful external USB devices and further to customer feedback based development the new Raspberry Pi 3 now comes with built-in wireless and Bluetooth connectivity.

The following is technical specification of Raspberry Pi 3 device:

- Broadcom BCM2837 64bit ARMv8 Quad Core Processor powered Single Board Computer running at 1.2GHz
- 1GB RAM
- BCM43143 WiFi on board
- Bluetooth Low Energy (BLE) on board
- 40pin extended GPIO
- 4 x USB2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- MicroSD port for loading your operating system and storing data
- Upgraded switched Micro USB power source (now supports up to 2.4 Amps)
- Same form factor as the Raspberry Pi 2 Model B, however the LEDs will change position

You can see Raspberry Pi 3 model B device on the Figure below.



## 1.2 Getting Hardware

How to get Raspberry Pi 3 device?

Officially you can buy it from the official distributor  
on <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> .

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 3 device and found that there are another options to buy

- The Pi Hut, <http://thepihut.com>
- EXP-Tech, <http://www.exp-tech.de/>
- Sparkfun, <https://www.sparkfun.com>
- adafruit, <https://www.adafruit.com>
- Ebay, <http://www.ebay.com> (personal transaction)

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

## 1.3 Unboxing

After bought Raspberry Pi 3 from The Pi Hut (<http://thepihut.com>), I get the board as follows.



## **2. Operating System**

This chapter explains how to work with Operating System for Raspberry Pi 3.

## **2.1 Raspberry Pi 3 Operating System**

Raspberry Pi provides some Operating Systems you can use and run on the top of Raspberry Pi 3. You can download these OS files on <http://www.raspberrypi.org/downloads>.

In this book, I use Raspbian “Jessie” OS.

## 2.2 Preparation

Raspbian is an Operating system based on Debian Linux for the Raspberry Pi hardware. I recommend you to download OS image file on <http://www.raspberrypi.org/downloads> . For illustration, I use Raspbian Jessie OS.

After extracted this file, you will obtain \*.img file, for instance, 2016-02-26-raspbian-jessie.img file.

### 2.2.1 Setup MicroSD Card

If we are working with Raspberry Pi 3 board, we need MicroSD card to extract this OS image file. I use MicroSD Card 16 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 Mac user, you can use a script from this tool, <https://github.com/RayViljoen/Raspberry-PI-SD-Installer-OS-X> .

Navigate file install with Jessie image file.

```
$ ./install 2016-02-26-raspbian-jessie.img
```

Wait it until done.

```
[agusk$ sudo ./install 2016-02-26-raspbian-jessie.img

[=====  
-----  


|    | Filesystem    | Size  | Used  | Avail | Capacity | iused    | ifree   | %iused | Mounted on    |
|----|---------------|-------|-------|-------|----------|----------|---------|--------|---------------|
| 1) | /dev/disk1    | 233Gi | 208Gi | 25Gi  | 90%      | 54466806 | 6514440 | 89%    | /             |
| 2) | devfs         | 331Ki | 331Ki | 0Bi   | 100%     | 1144     | 0       | 100%   | /dev          |
| 3) | map -hosts    | 0Bi   | 0Bi   | 0Bi   | 100%     | 0        | 0       | 100%   | /net          |
| 4) | map auto_home | 0Bi   | 0Bi   | 0Bi   | 100%     | 0        | 0       | 100%   | /home         |
| 5) | /dev/disk2s1  | 7.4Gi | 2.5Mi | 7.4Gi | 1%       | 0        | 0       | 100%   | /Volumes/DATA |

  
-----  
  
Select the disk to use by entering the disk number.  
*** MAKE SURE YOU SELECT THE CORRECT DISK ***  
*** Refer to the Readme if uncertain ***  
  
Use disk [ 1, 2, 3, 4, 5 ] #
```

```
raspbian — bash — 80x28
2) devfs      331Ki  331Ki   0Bi  100%    1144     0  100% /dev
3) map -hosts   0Bi   0Bi   0Bi  100%      0     0  100% /net
4) map auto_home 0Bi   0Bi   0Bi  100%      0     0  100% /home
5) /dev/disk2s1 7.4Gi  2.5Mi  7.4Gi   1%      0     0  100% /Volumes/D
ATA
-----
Select the disk to use by entering the disk number.
*** MAKE SURE YOU SELECT THE CORRECT DISK ***
*** Refer to the Readme if uncertain ***

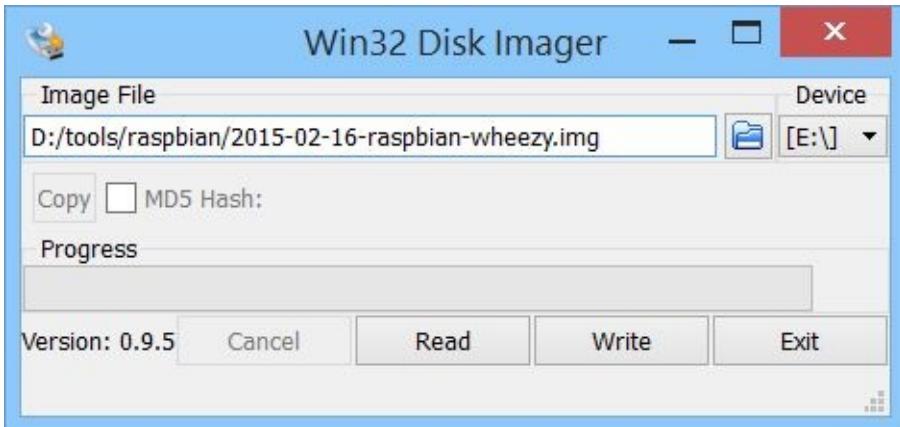
Use disk [ 1, 2, 3, 4, 5 ] #5
Unmounting Disk
Volume DATA on disk2s1 unmounted
Writing image
Ctrl+T to see progress..
load: 3.71 cmd: dd 2071 uninterruptible 0.00u 1.59s
1785+0 records in
1784+0 records out
1870659584 bytes transferred in 301.767229 secs (6199015 bytes/sec)
3843+0 records in
3843+0 records out
4029677568 bytes transferred in 655.138684 secs (6150877 bytes/sec)
Ejecting Disk
Disk /dev/rdisk2 ejected
All Done!
agusuk$
```

For Windows users:

Download Win32DiskImager on <https://sourceforge.net/projects/win32diskimager/> .

Run Win32DiskImager and navigate Raspberry Pi image file.

The following is a sample of Win32 Disk Imager with Raspbian Wheezy image.



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 3.



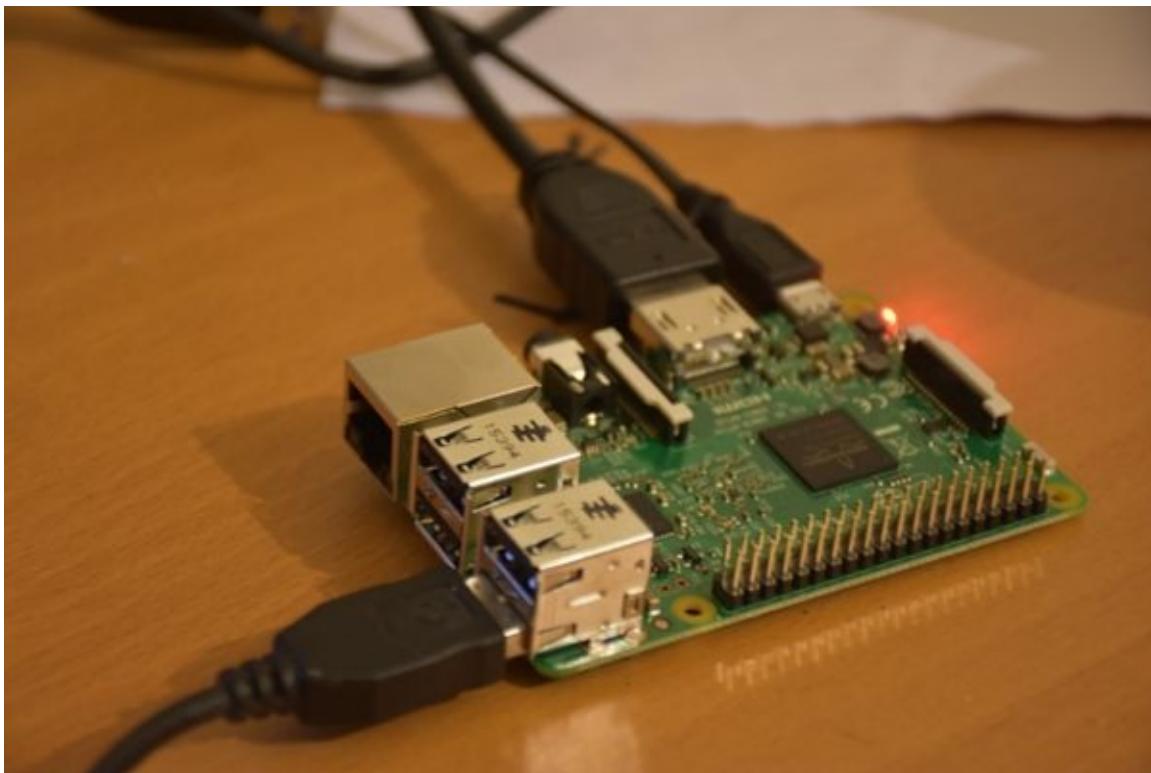
Now your Raspberry Pi 3 is ready to be deployed OS.

### **3. Powering Up and Running**

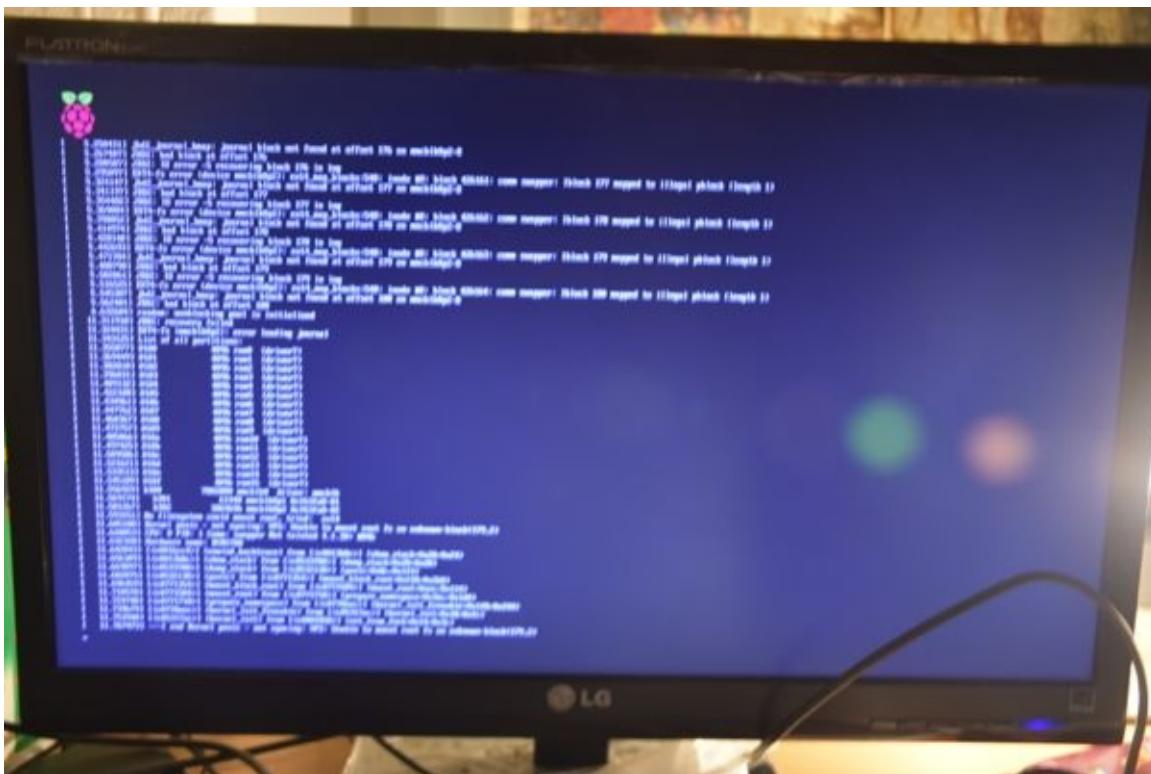
In this chapter we start to run and configure Raspberry Pi 3.

### 3.1 Put Them All!

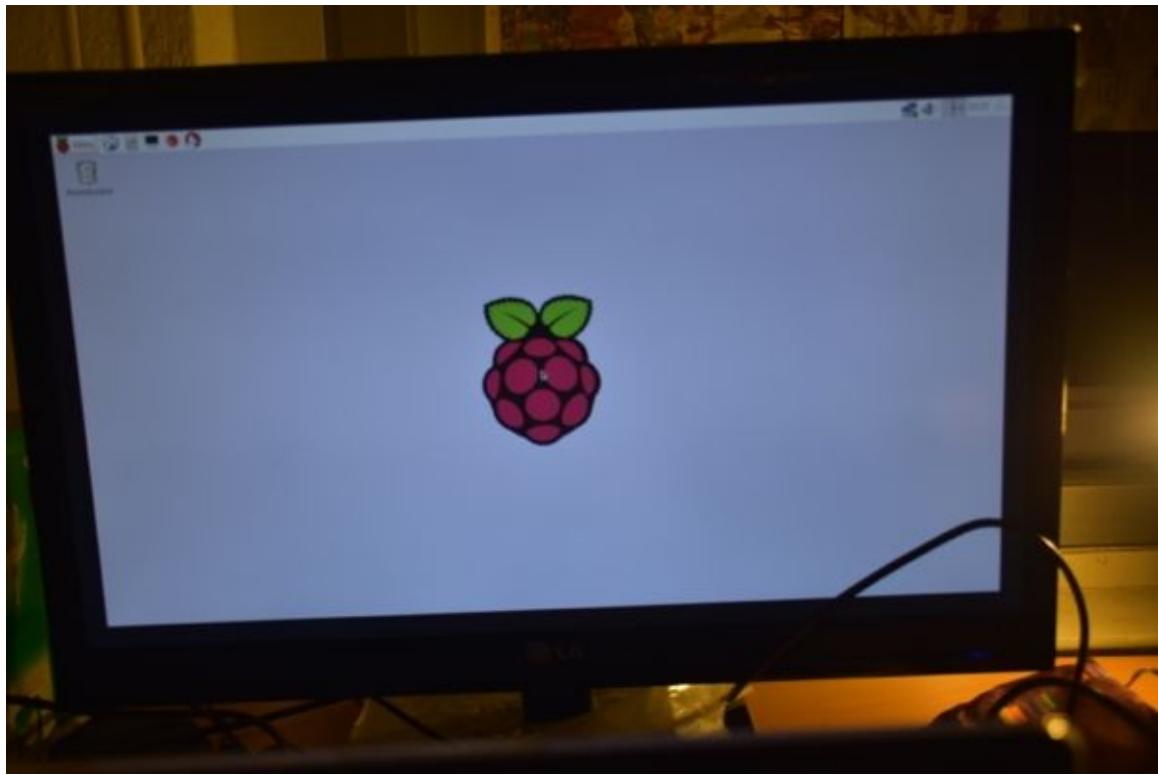
Now you are ready to boot your Raspberry Pi. Connect Monitor via HDMI, a keyboard via USB and power into Raspberry Pi.



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 Jessie desktop as below



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

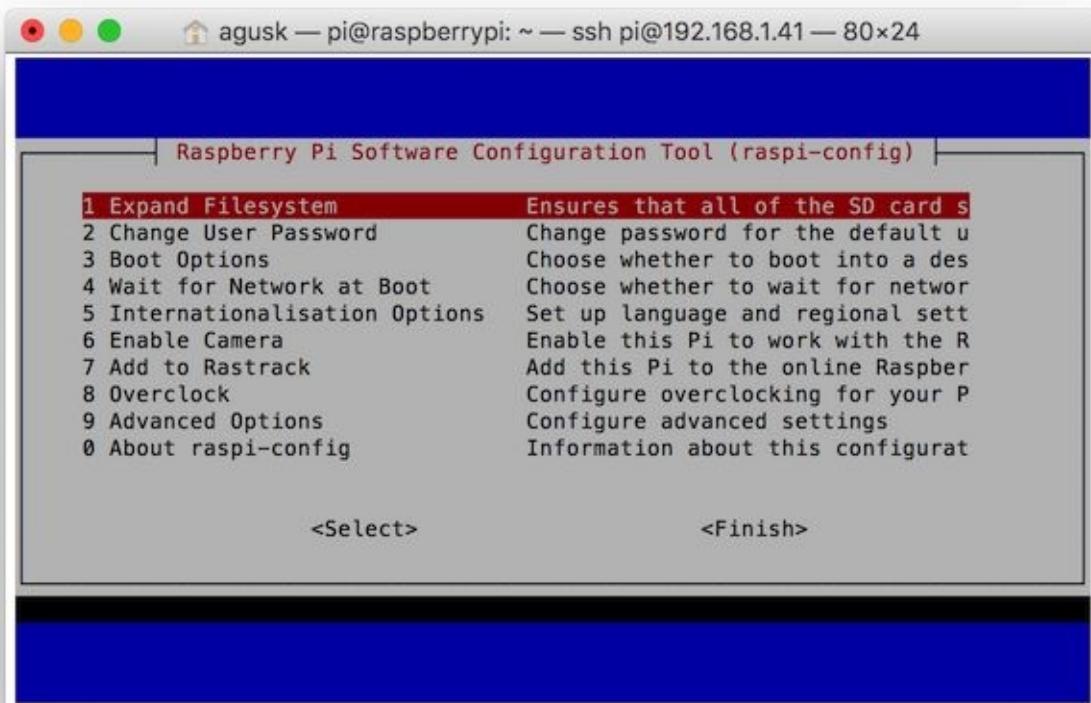


## 3.2 Expanding File System

After running Jessie OS, you can configure it by expanding Filesystem via raspi-config to optimize your micro SD storage. Type this command on Terminal.

```
sudo raspi-config
```

Then, you get the following form.



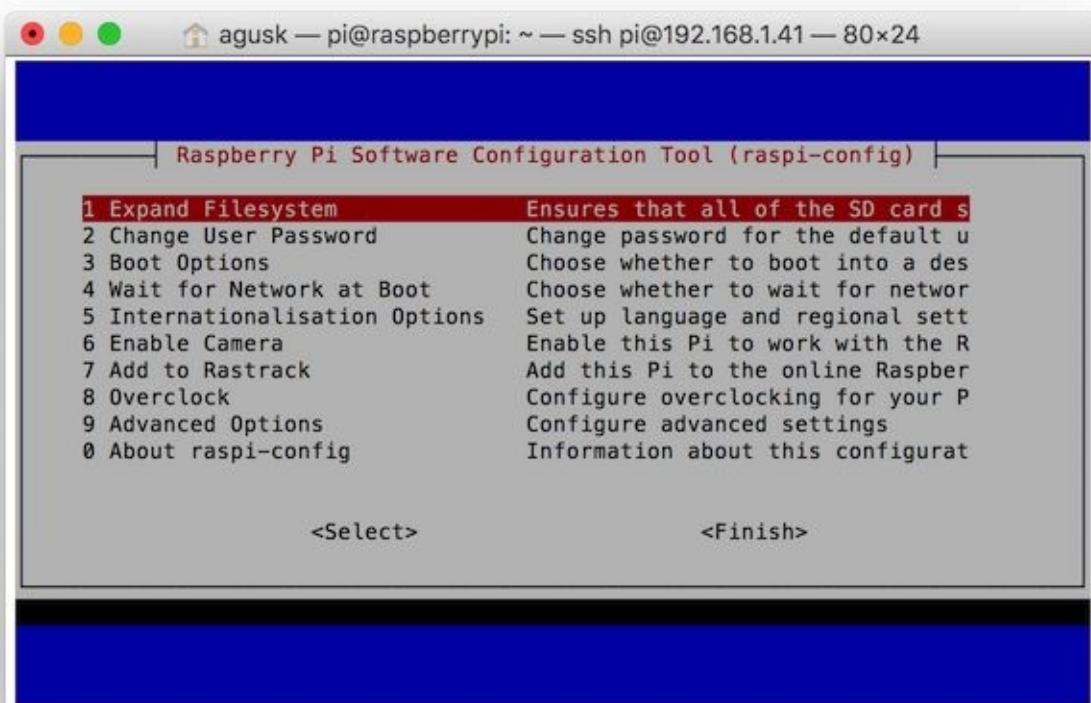
Select 1 Expand Filesystem. After that, you are required to restart Raspbian.

### 3.3 Configure Timezone

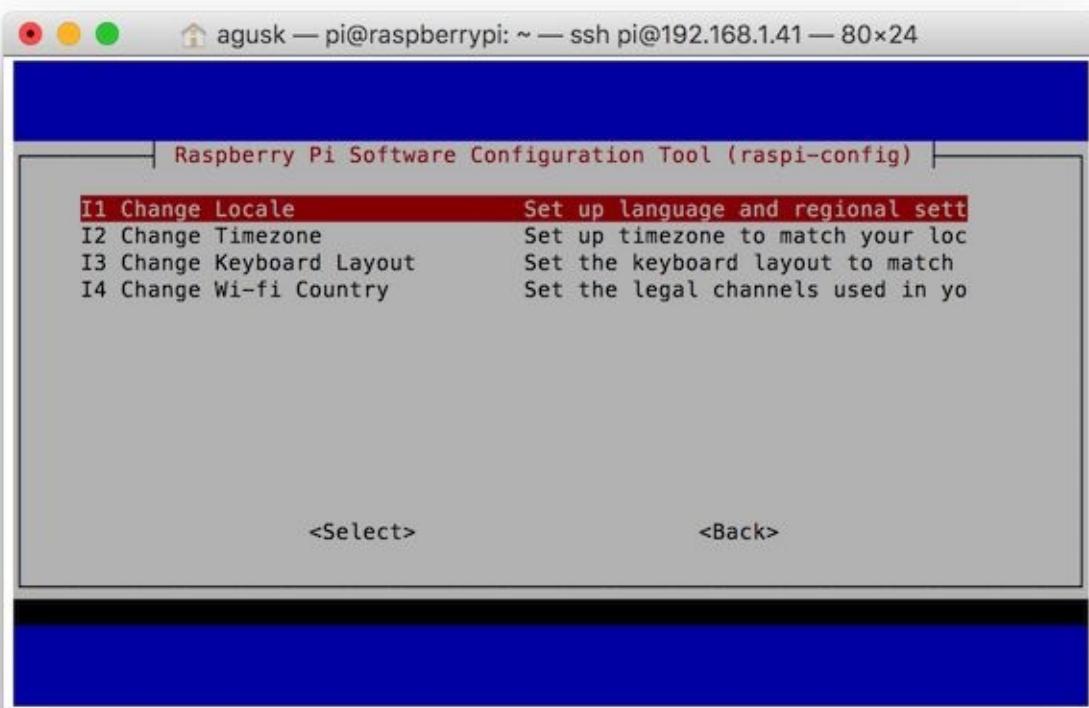
If you want to change timezone in Raspberry Pi 3, you can do it in console with typing

```
sudo raspi-config
```

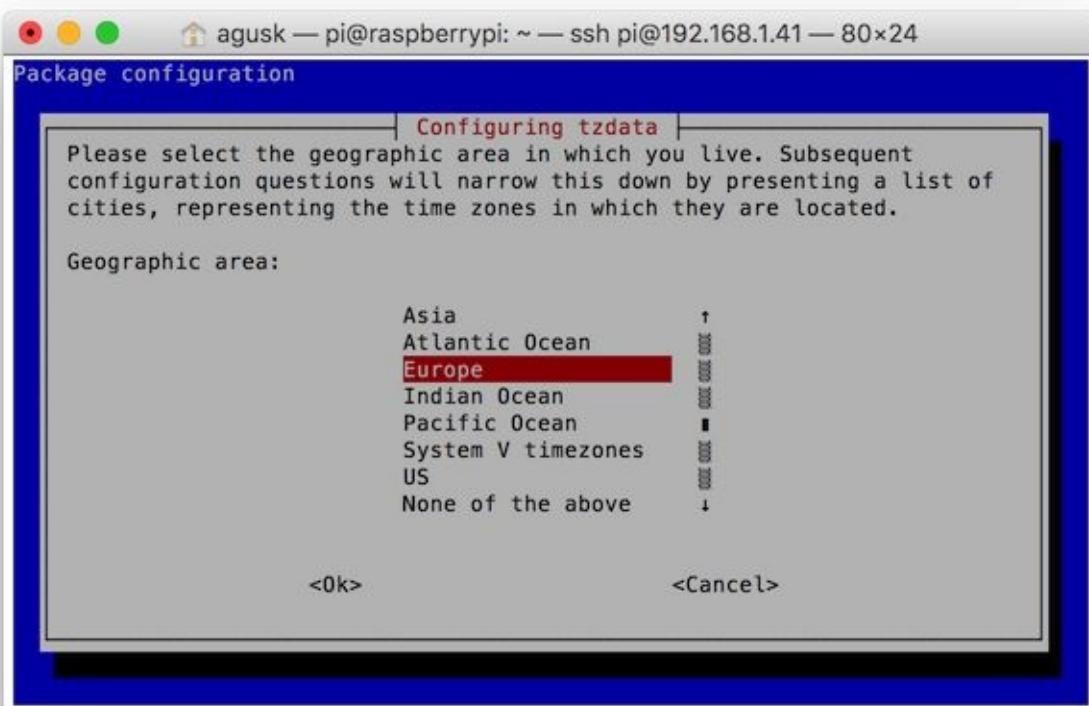
Then, you will get a dialog as below



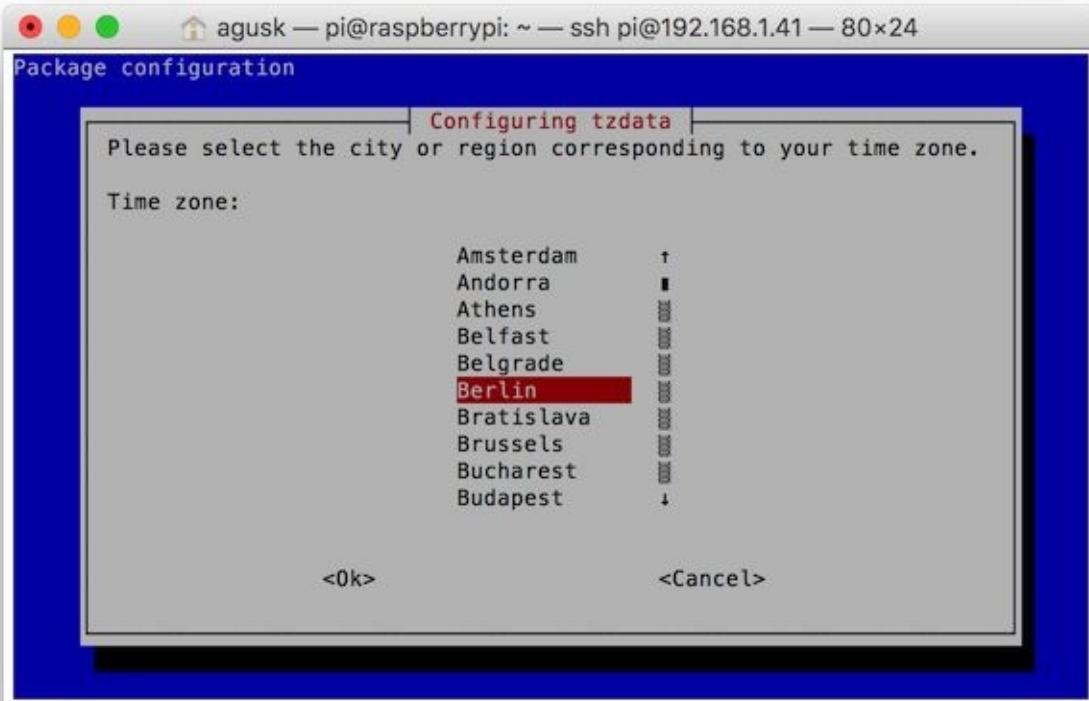
Select 5 Internationalisation Options.



Then, I2 Change Timezone.



Select your Geographic area.



Then, select your city or region.



After that, Raspbian will configure your Timezone.

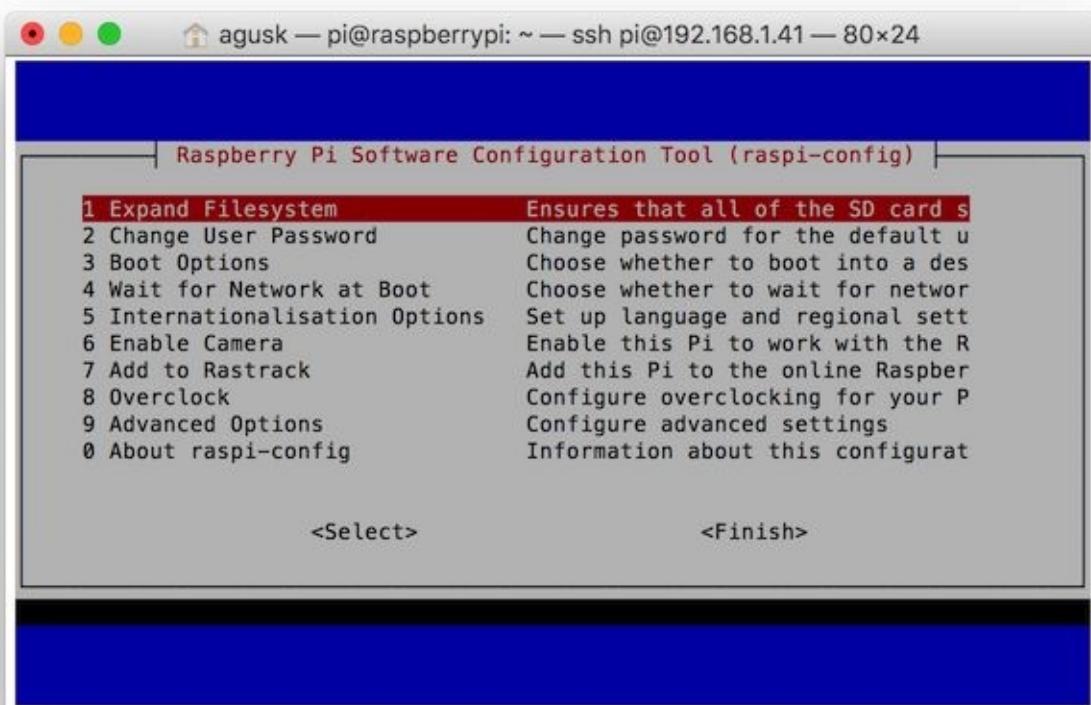


## 3.4 Configure Keyboard

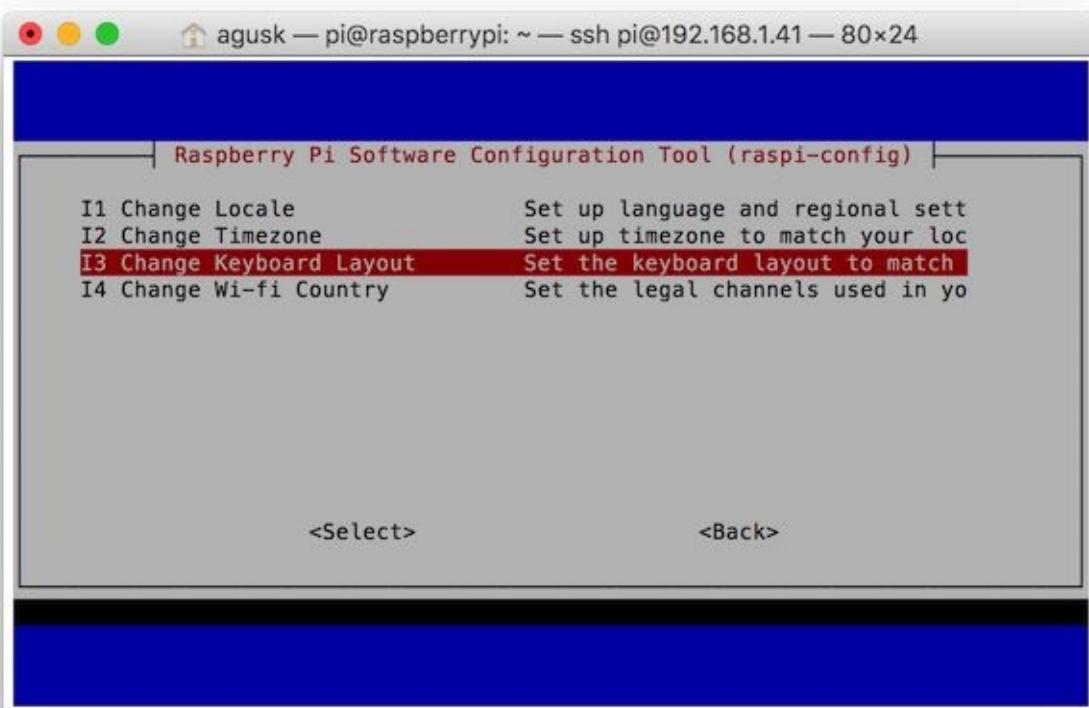
You may change your keyboard configuration. You can type this command into Terminal.

```
sudo raspi-config
```

Then you will get a dialog as below



Select 5 Internationalisation Options.



Select I3 Change Keyboard Layout. Choose your keyboard type and model.

## 3.5 Rebooting

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

```
sudo shutdown -r now
```

You also can do it with writing this script on Terminal.

```
sudo reboot
```

## 3.6 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

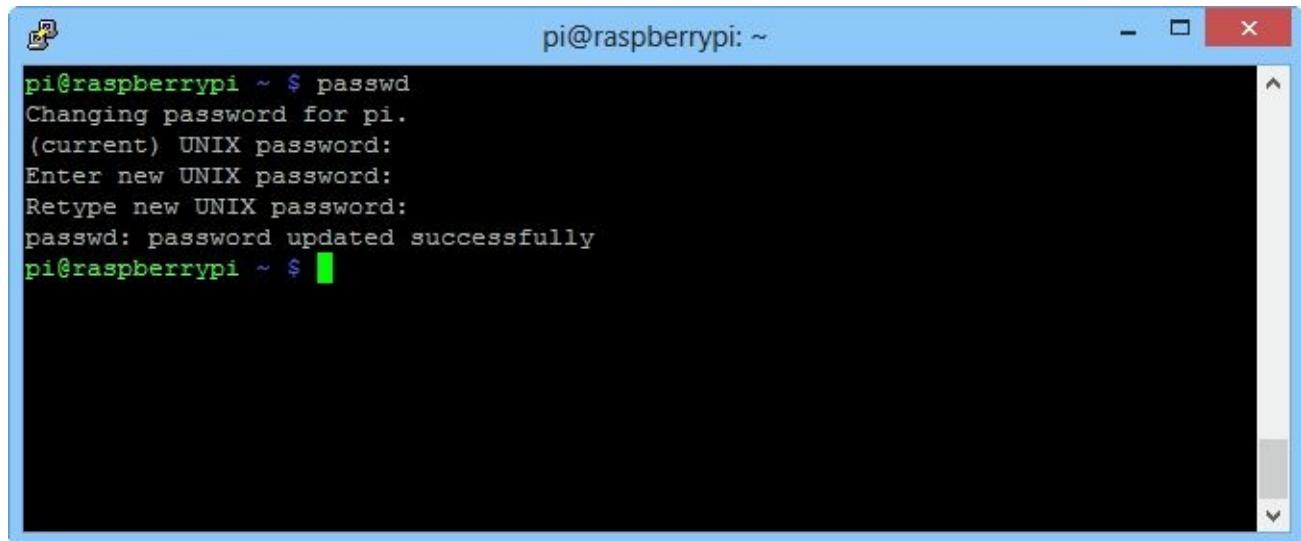
```
$ sudo shutdown -h -P now
```

## 3.7 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 screenshot of a terminal window titled "pi@raspberrypi: ~". The window contains the following text:

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

## 3.8 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 3.

## 4.1 Getting Started

In this chapter, we learn how to connect Raspberry Pi 3 to a network through built-in Ethernet and WiFi.

Let's start.

## 4.2 Attaching a Network Module

Raspberry Pi 3 has built-in Ethernet and WiFi so we can connect to a network directly. We don't need additional network module.

## 4.3 Connecting to a Network

### 4.3.1 Connecting a Network via Ethernet

Connect Ethernet USB to Raspberry Pi and a network.



Turn on Raspberry Pi 3. You can see the monitor and access it via Keyboard. You also access it via SSH, read section 4.7.

Now you can check your current IP Address by writing this script

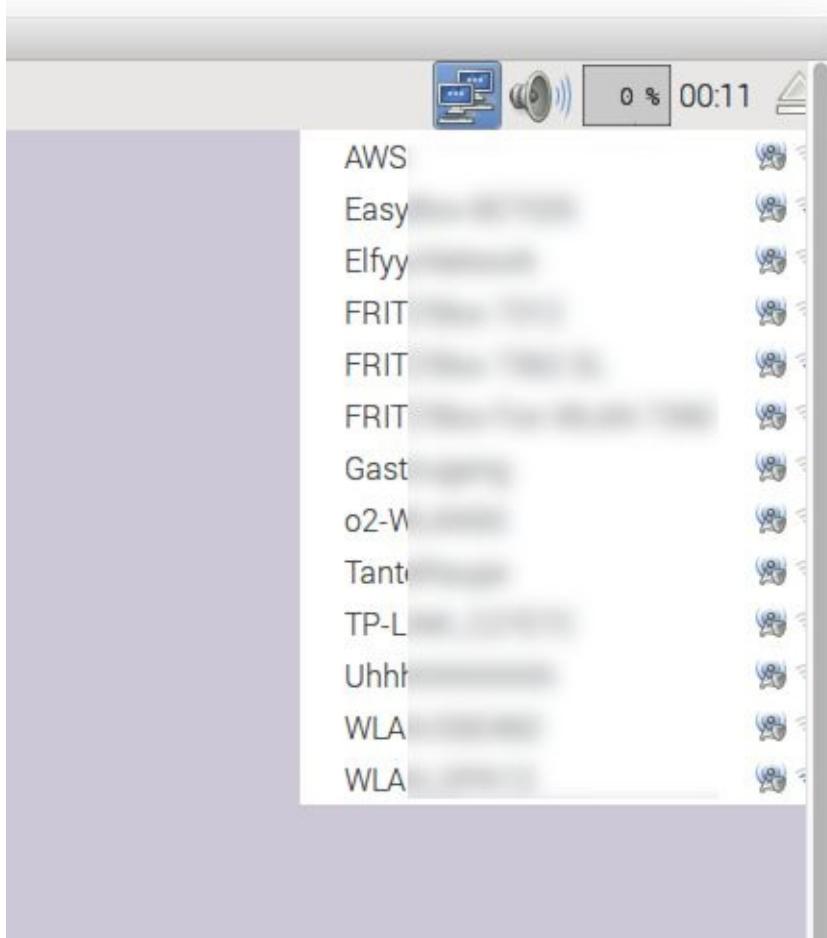
```
$ ifconfig -a
```

You should see your current IP address.

### 4.3.2 Connecting a Network via WiFi

Raspberry Pi 3 has built-in WiFi so you can connect it to a network. on Raspbian Jessie desktop, you can click on the right tool (Network icon) so you can see the list of WiFi Hotspot.

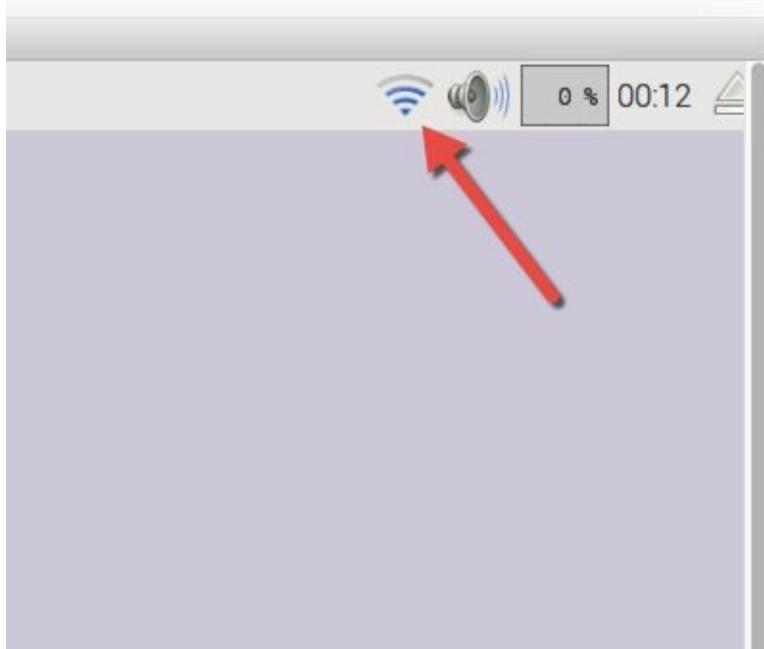
Select your SSID.



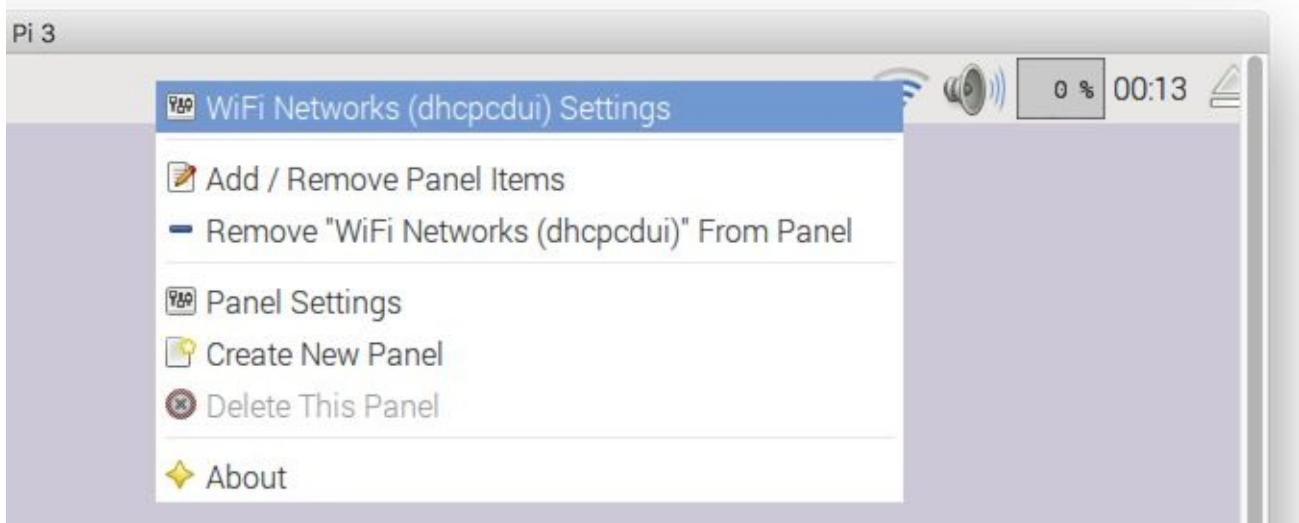
Fill SSID key if your WiFi Hotspot needs authentication key.



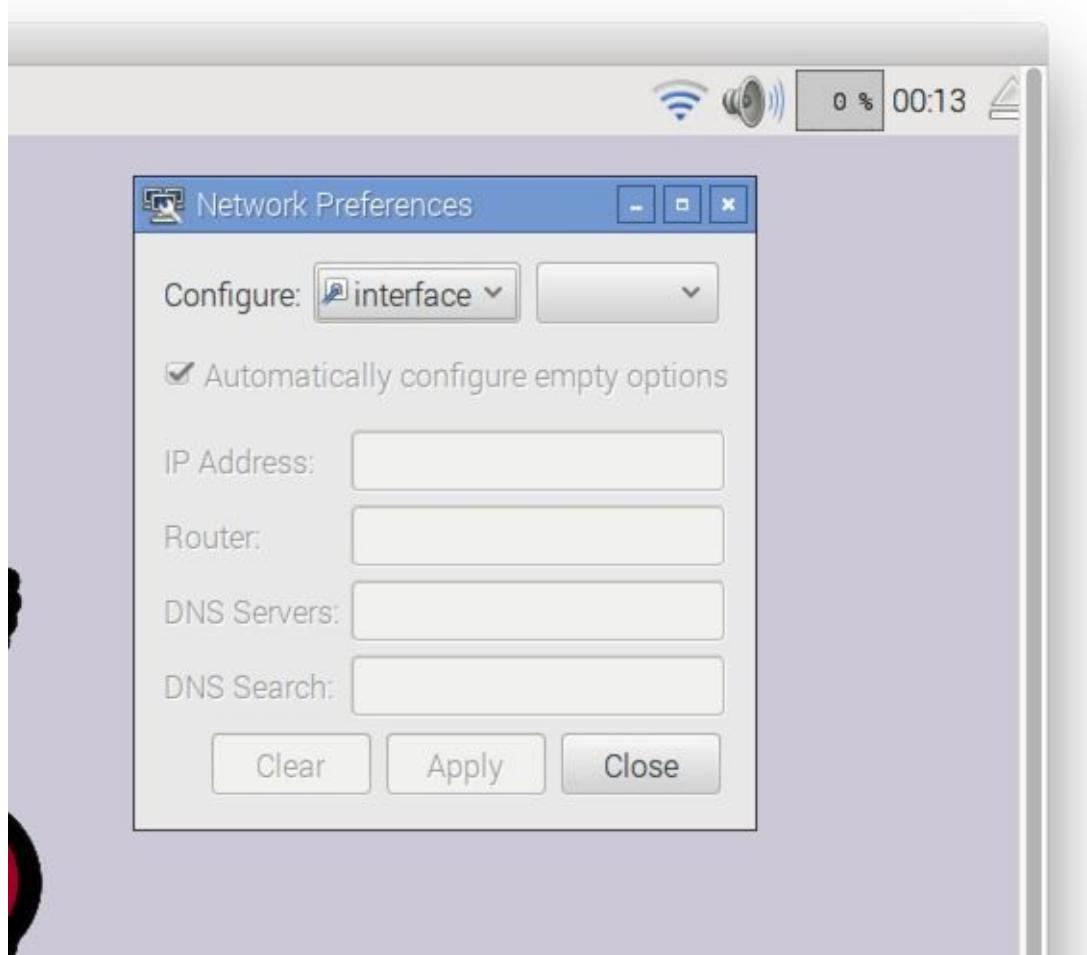
If succeed, WiFi icon will show on the right toolbar.



You can configure the WiFi setting by right-click on WiFi icon. Select WiFi Networks (dhcpcd) Settings.



Then, you should get Network dialog.



## 4.4 Auto Connect to WiFi

You may want to connect to WiFi automatically after rebooting. You can open /etc/network/interfaces file.

```
$ sudo nano /etc/network/interfaces
```

Then, on wlan0 you change from manual to dhcp.

The following is the final configuration on /etc/network/interfaces file.



```
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

After that, you can edit /etc/wpa\_supplicant/wpa\_supplicant.conf file to configure SSID included its key.

```
$ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Add these settings. Change ssid and psk values.

```
network={
    ssid="SSID_WIFI"
    psk="SSID_Key"
    key_mgmt=WPA-PSK
}
```

The screenshot shows a terminal window titled "agusk — pi@raspberrypi: ~ — ssh pi@192.168.1.59 — 80x22". The file being edited is "/etc/wpa\_supplicant/wpa\_supplicant.conf". The content of the file is as follows:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=DE

network={
    ssid="SSID_WIFI"
    psk="SSID_key"
    key_mgmt=WPA-PSK
}
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts:

^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 all settings. Now try to reboot Raspbian.

## 4.5 Configuring 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
```

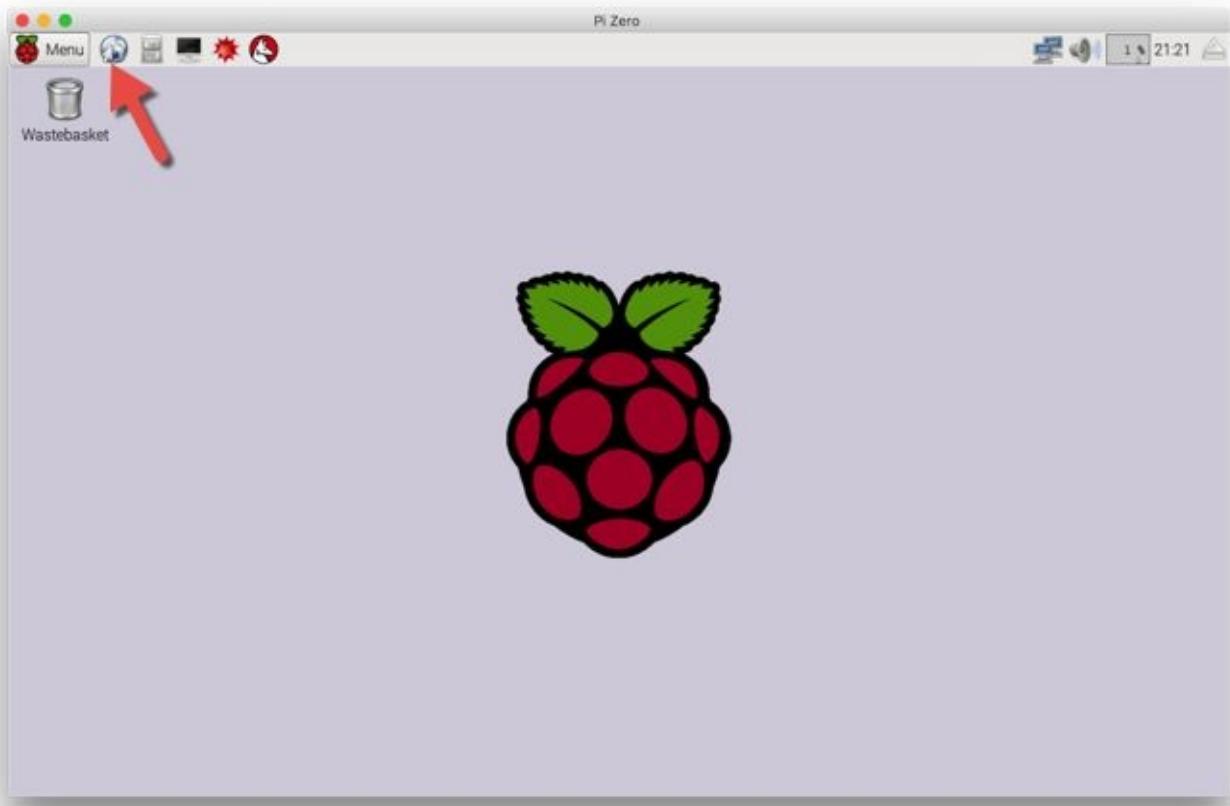
Then, you will see a content of file interface. Replace **iface eth0 inet dhcp** with

```
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.

## 4.6 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.

Pi Zero

Menu Agus Kurniawan | The Art of Computing 21:24

Wastebasket

Agus Kurniawan | The Art of Computing http://blog.aguskurniawan.net/ Home Professional Services Archive Contact Subscribe Filter by APML

**Arduino and Genuino 101 Development Workshop**  
By Agus Kurniawan 1. March 2016 15:15  
Intel has released Intel Curie which deployed on Arduino and Genuino 101. Then, I released a new book, Arduino and Genuino 101 Development Workshop. This book helps you to get started with Arduino and Genuino 101 development using Sketch. Table of Content 1. Setting up Development ... [More]  
\*\*\*\*\* Currently rated 5.0 by 1 people

Tags: Books | C/C++ | Embedded System  
Submit to DotNetKicks... 0 Comments

**Getting Started with Wolfram Language and Mathematica for Raspberry Pi**  
By Agus Kurniawan 12. February 2016 13:42  
Raspberry Pi with Raspbian OS has built-in Wolfram Language and Mathematica. I released a new book, Getting Started with Wolfram Language and Mathematica for Raspberry Pi. This book helps you to get started with Wolfram Mathematica programming in Raspberry Pi environment. Table of Content... [More]

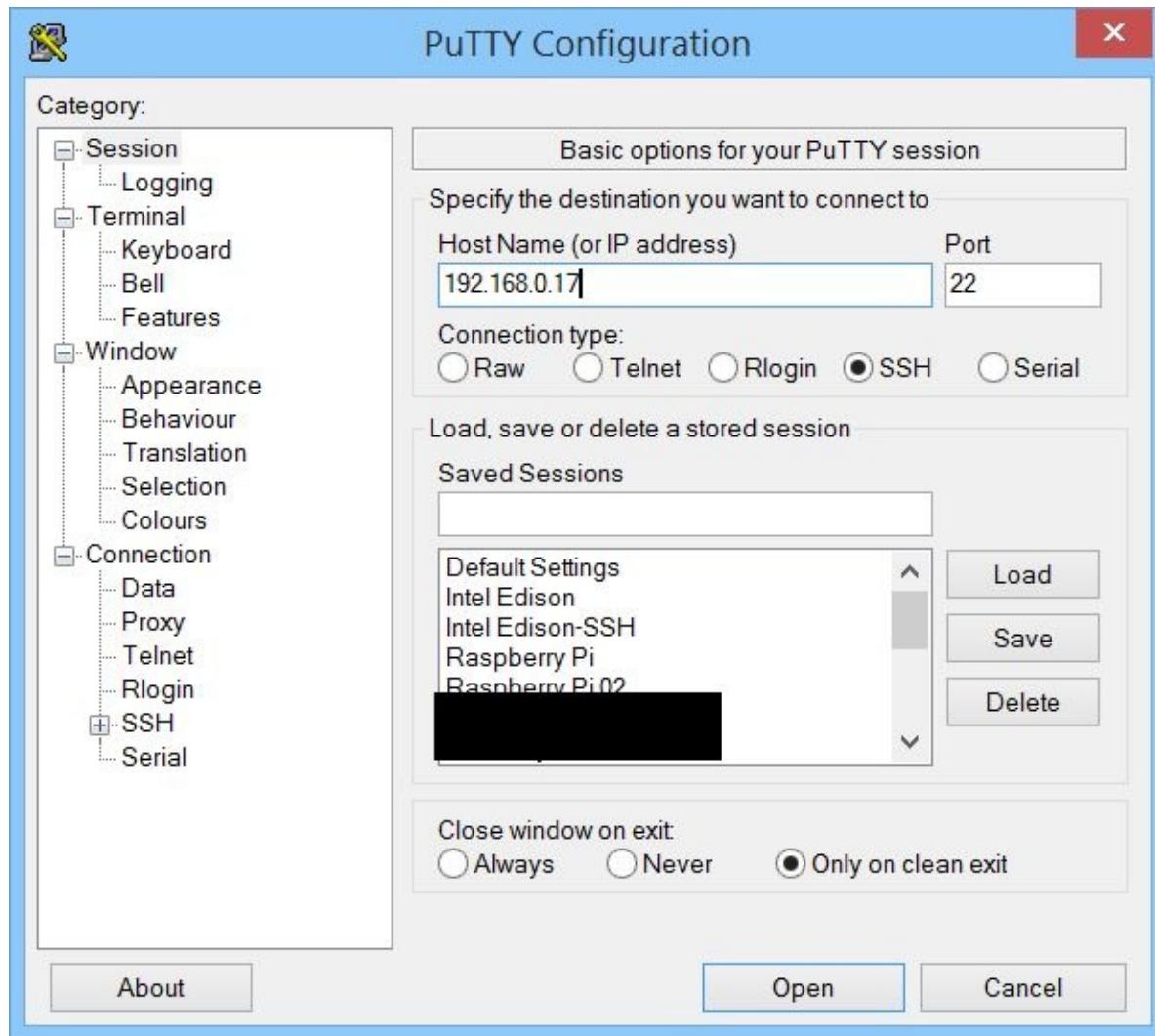
About Kurniawan  
  
For more click here  
Please e-mail my guess  
If you want to please go  
Go here services

## 4.7 SSH

By default, Raspbian Jessie has installed and enabled SSH service so we can use it directly.

For testing, I used PuTTY, <http://www.putty.org>, application in Windows platform to remote Raspberry Pi via SSH.

Fill IP Address of Raspberry Pi.



You can get IP address your Raspberry Pi 3 board by checking it on your router. For instance, my router detected my board MAC.

DHCP Server  Yes  No  
 Starting Local Address 192.168.0.10  
 Number of CPEs 119  
 Lease Time 3600  
 Apply Back

DHCP Client Lease Info				
MAC Address	IP Address	Subnet Mask	Duration	Expires
[REDACTED]	192.168.0.017	255.255.255.009	D:00 H:01 M:00 S:00	Thu Apr 09 23:36:04 2015
Current System Time: Thu Apr 09 22:36:38 2015				
<input type="button" value="Force Available"/>				

Raspberry Pi usually has MAC address with prefix B8:27:EB . 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.

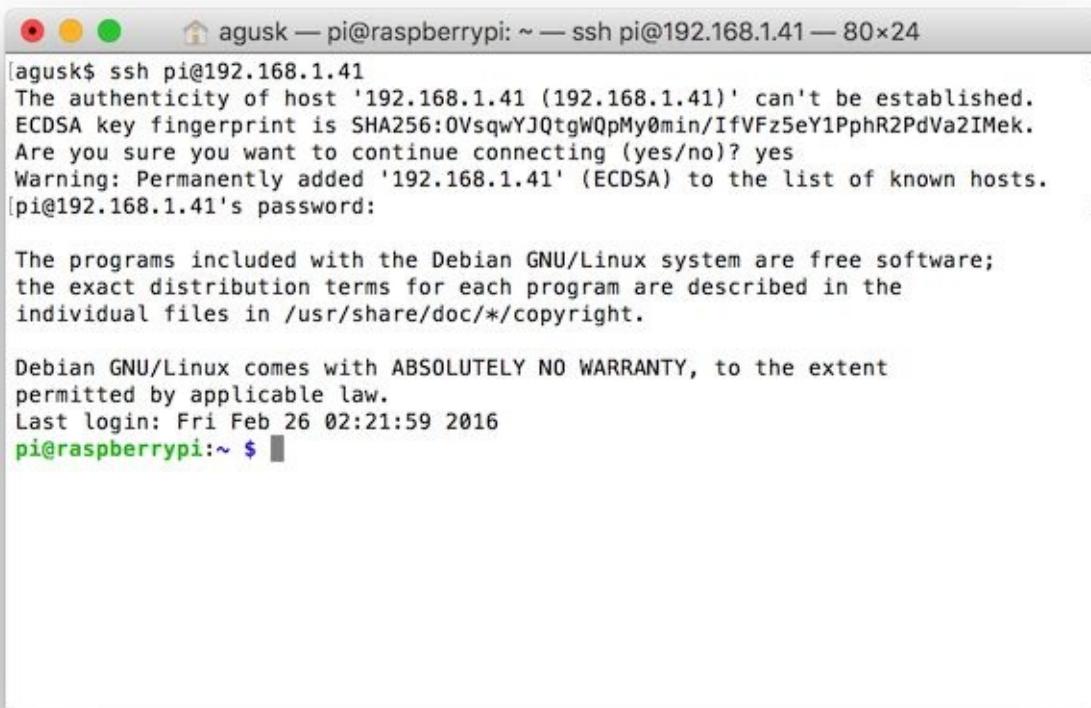


Click **Yes** button.

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

For Mac and Linux, you can use SSH on Terminal. For instance, IP address of Raspberry Pi is 192.168.1.41, you can type this command.

```
$ ssh pi@192.168.1.41
```



The screenshot shows a Mac OS X terminal window titled "agusk — pi@raspberrypi: ~ — ssh pi@192.168.1.41 — 80x24". The window contains the following text:

```
[agusk$ ssh pi@192.168.1.41
The authenticity of host '192.168.1.41 (192.168.1.41)' can't be established.
ECDSA key fingerprint is SHA256:OVsqwYJQtgWQpMy0min/IfVFz5eY1PphR2PdVa2IMek.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.41' (ECDSA) to the list of known hosts.
[pi@192.168.1.41's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

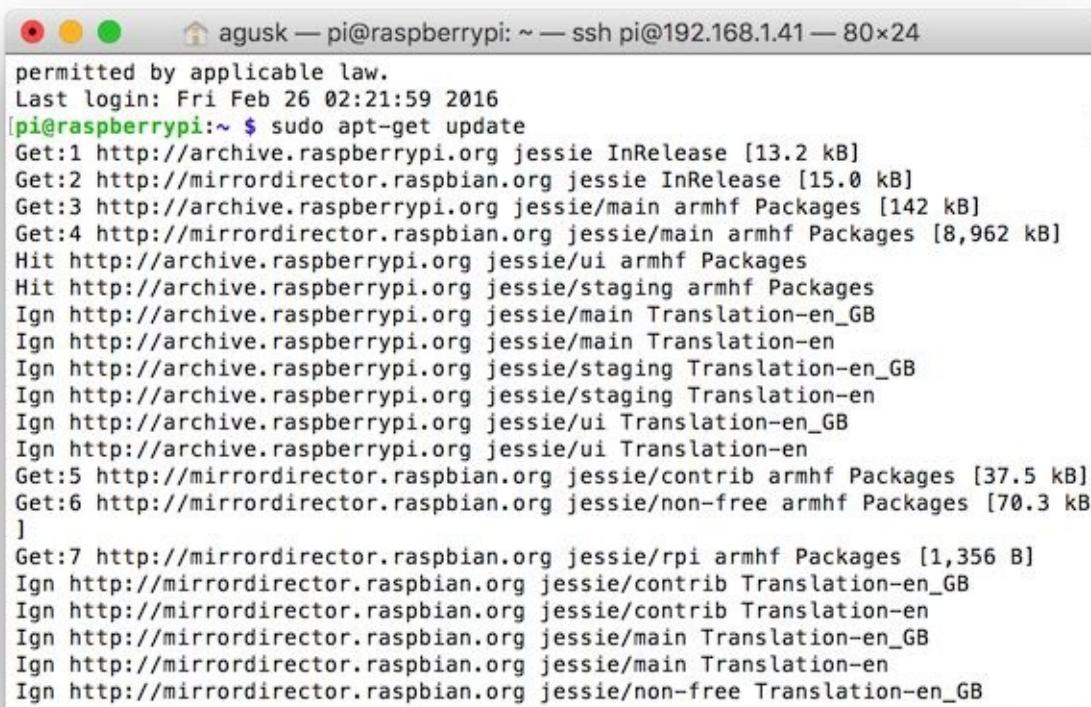
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb 26 02:21:59 2016
pi@raspberrypi:~ $ ]
```

## 4.8 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.



The screenshot shows a terminal window titled "agusk — pi@raspberrypi: ~ — ssh pi@192.168.1.41 — 80x24". The window contains the command "sudo apt-get update" followed by its output. The output shows the system checking for updates from various repositories, including "archive.raspberrypi.org" and "mirrordirector.raspbian.org", for packages like "jessie InRelease", "armhf Packages", and "Translation-en\_GB". The process involves multiple "Get:" and "Ign" commands, indicating both downloads and ignoring existing files. The total download size is approximately 100 kB.

```
permitted by applicable law.  
Last login: Fri Feb 26 02:21:59 2016  
[pi@raspberrypi:~ $ sudo apt-get update  
Get:1 http://archive.raspberrypi.org jessie InRelease [13.2 kB]  
Get:2 http://mirrordirector.raspbian.org jessie InRelease [15.0 kB]  
Get:3 http://archive.raspberrypi.org jessie/main armhf Packages [142 kB]  
Get:4 http://mirrordirector.raspbian.org jessie/main armhf Packages [8,962 kB]  
Hit http://archive.raspberrypi.org jessie/ui armhf Packages  
Hit http://archive.raspberrypi.org jessie/staging armhf Packages  
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB  
Ign http://archive.raspberrypi.org jessie/main Translation-en  
Ign http://archive.raspberrypi.org jessie/staging Translation-en_GB  
Ign http://archive.raspberrypi.org jessie/staging Translation-en  
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB  
Ign http://archive.raspberrypi.org jessie/ui Translation-en  
Get:5 http://mirrordirector.raspbian.org jessie/contrib armhf Packages [37.5 kB]  
Get:6 http://mirrordirector.raspbian.org jessie/non-free armhf Packages [70.3 kB]  
]  
Get:7 http://mirrordirector.raspbian.org jessie/rpi armhf Packages [1,356 B]  
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB  
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en  
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB  
Ign http://mirrordirector.raspbian.org jessie/main Translation-en  
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
```

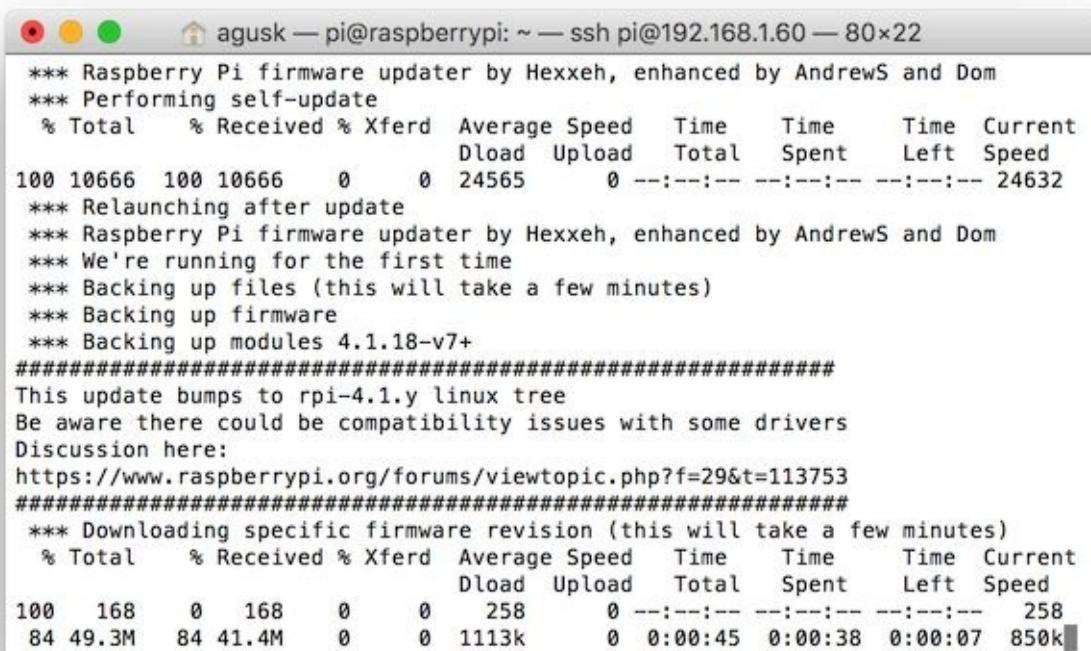
## 4.9 Upgrading Package and Firmware

You can update your current packages/libraries using this command.

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

You also can update Raspberry Pi firmware using this command.

```
$ sudo raspi-update
```



```
*** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
*** Performing self-update
  % Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
               Dload  Upload   Total   Spent    Left  Speed
100 10666  100 10666     0      0  24565      0 --::-- --::-- --::-- 24632
*** Relaunching after update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
*** We're running for the first time
*** Backing up files (this will take a few minutes)
*** Backing up firmware
*** Backing up modules 4.1.18-v7+
#####
This update bumps to rpi-4.1.y linux tree
Be aware there could be compatibility issues with some drivers
Discussion here:
https://www.raspberrypi.org/forums/viewtopic.php?f=29&t=113753
#####
*** Downloading specific firmware revision (this will take a few minutes)
  % Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
               Dload  Upload   Total   Spent    Left  Speed
100   168     0    168     0      0    258      0 --::-- --::-- --::--    258
  84 49.3M   84 41.4M     0      0  1113k      0  0:00:45  0:00:38  0:00:07  850k
```

## 4.10 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
```

```
agus — pi@raspberrypi: ~ — ssh pi@192.168.1.41 — 80x24
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
Fetched 9,112 kB in 45s (199 kB/s)
Reading package lists... Done
[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:
  vnc4server x11-apps x11-session-utils xbase-clients xbitmaps xfonts-base
Suggested packages:
  vnc-java mesa-utils x11-xfs-utils
The following NEW packages will be installed:
  vnc4server x11-apps x11-session-utils xbase-clients xbitmaps xfonts-base
  xrdp
0 upgraded, 7 newly installed, 0 to remove and 12 not upgraded.
Need to get 8,468 kB of archives.
After this operation, 17.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] ]
```

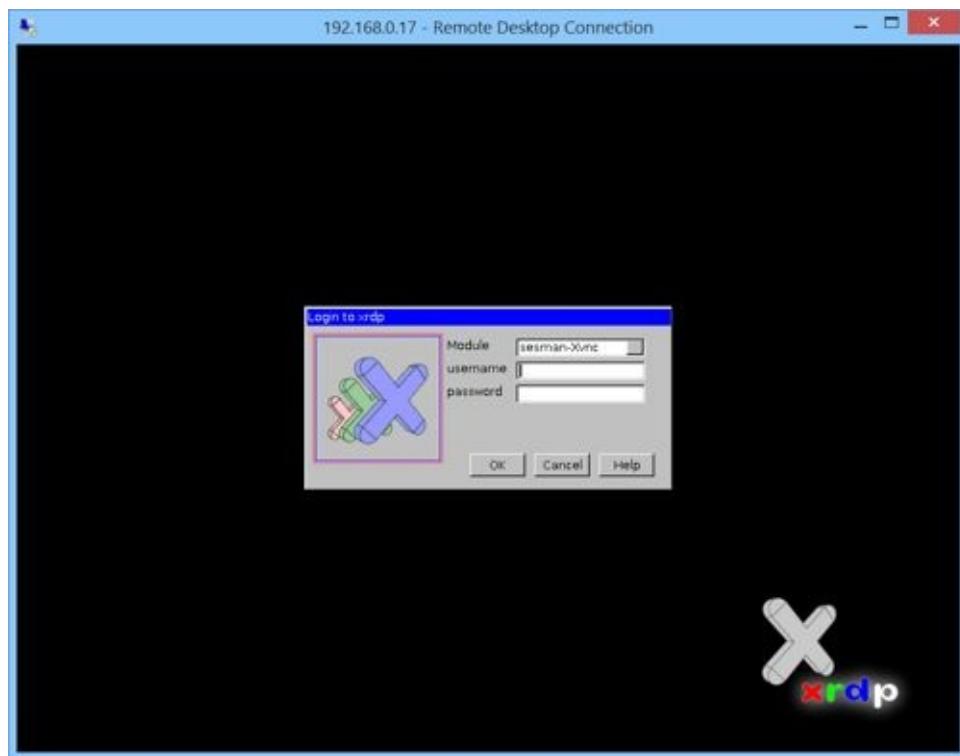
To test, I use Remote Desktop on Windows platform. Fill IP address of Raspberry Pi 3.



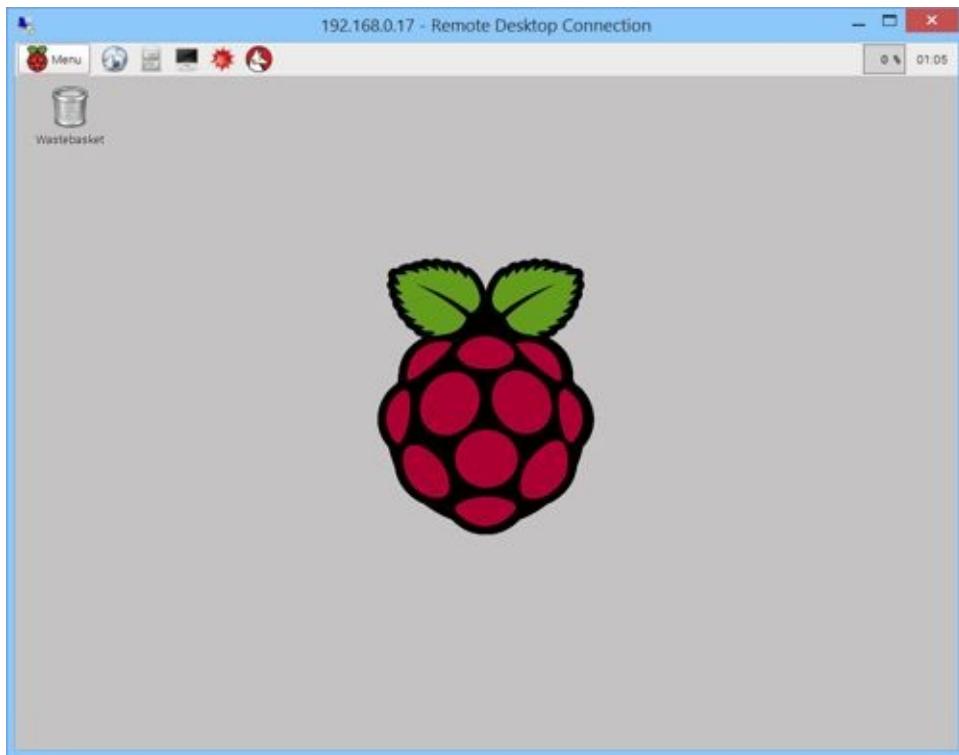
Click **Connect** button. If you will get a warning dialog. Click **Yes** button.



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



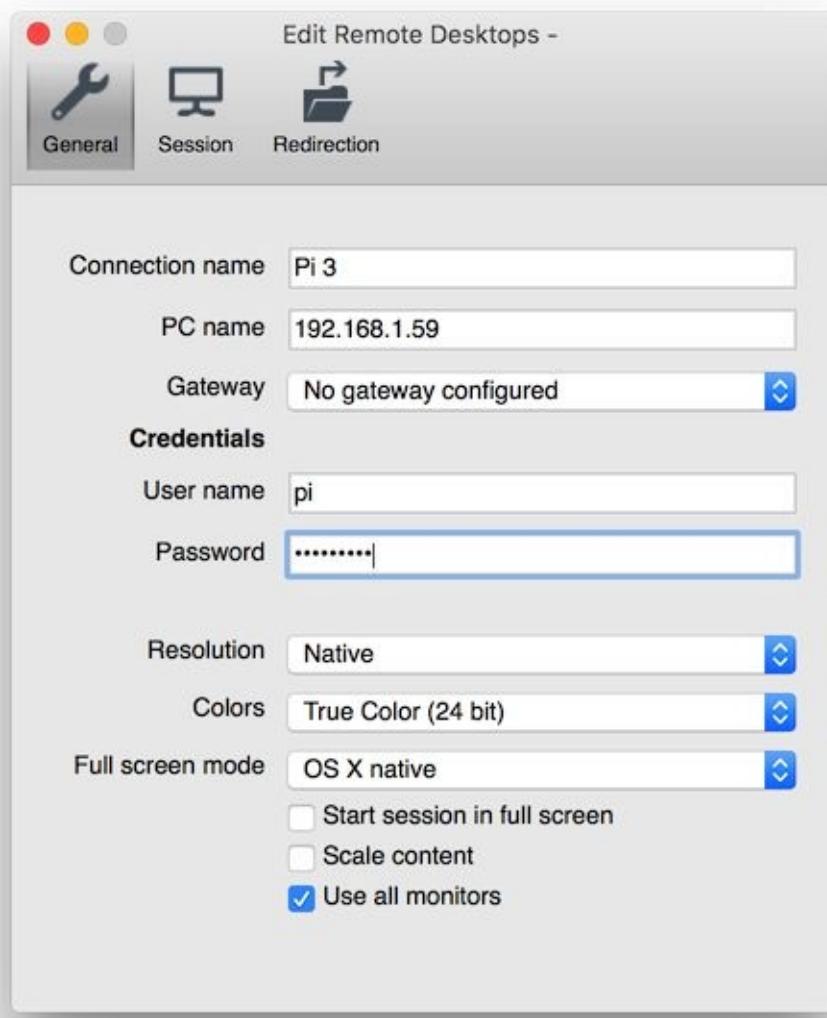
If done, you will get Raspbian desktop.



On Mac, I use Microsoft Remote Desktop

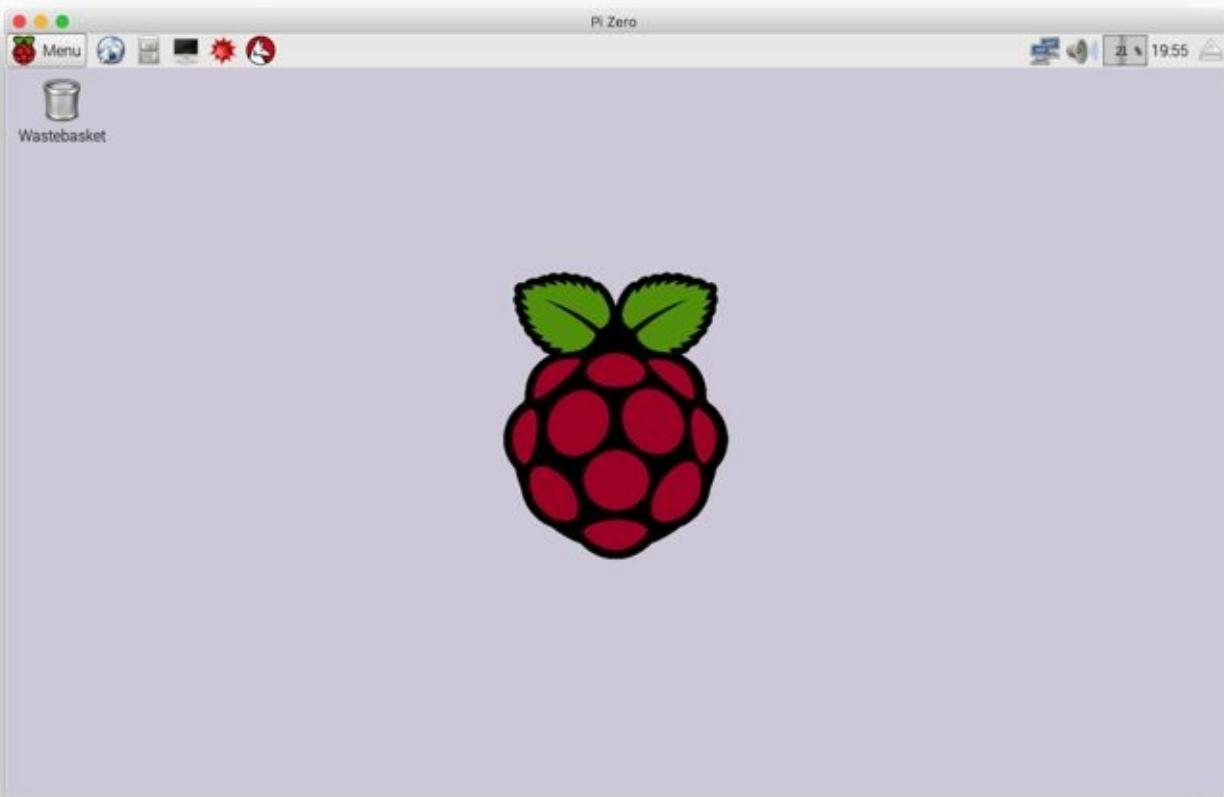


Fill IP address for your Raspberry Pi 3.



Then, connect it.

If succeed, you should see Jessie desktop.

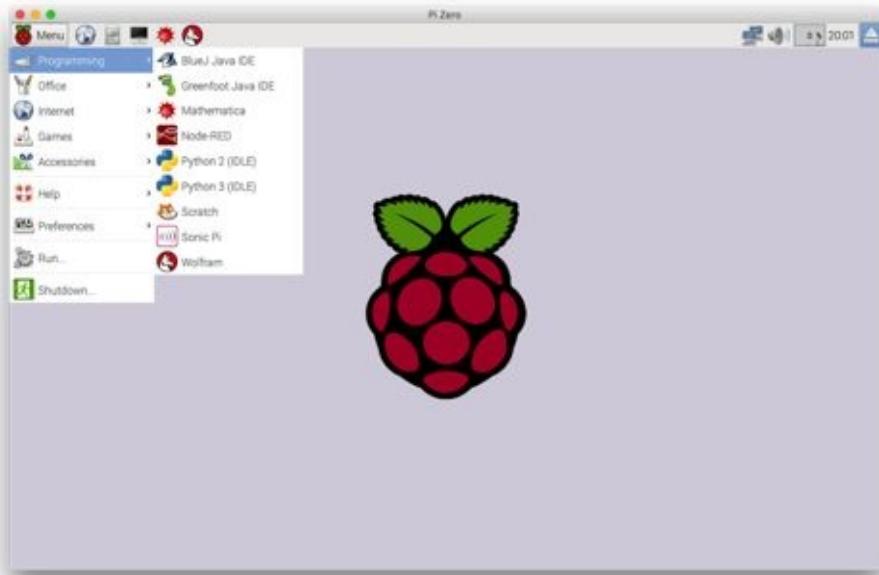


## **5. Raspberry Pi Programming**

This chapter explains how to work with development environment on Raspberry Pi 3.

## 5.1 Getting Started

In this section, we learn several programming on Jessie OS on Raspberry Pi 3.

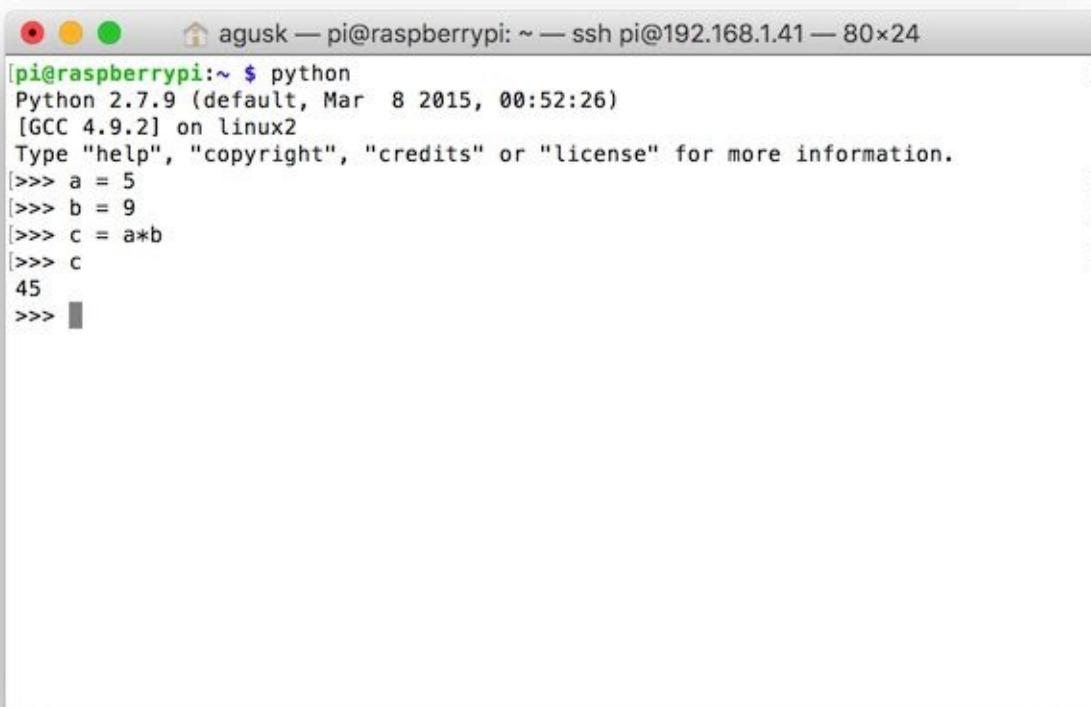


## 5.2 Python

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

```
$ python
```

After that, you run Python command.

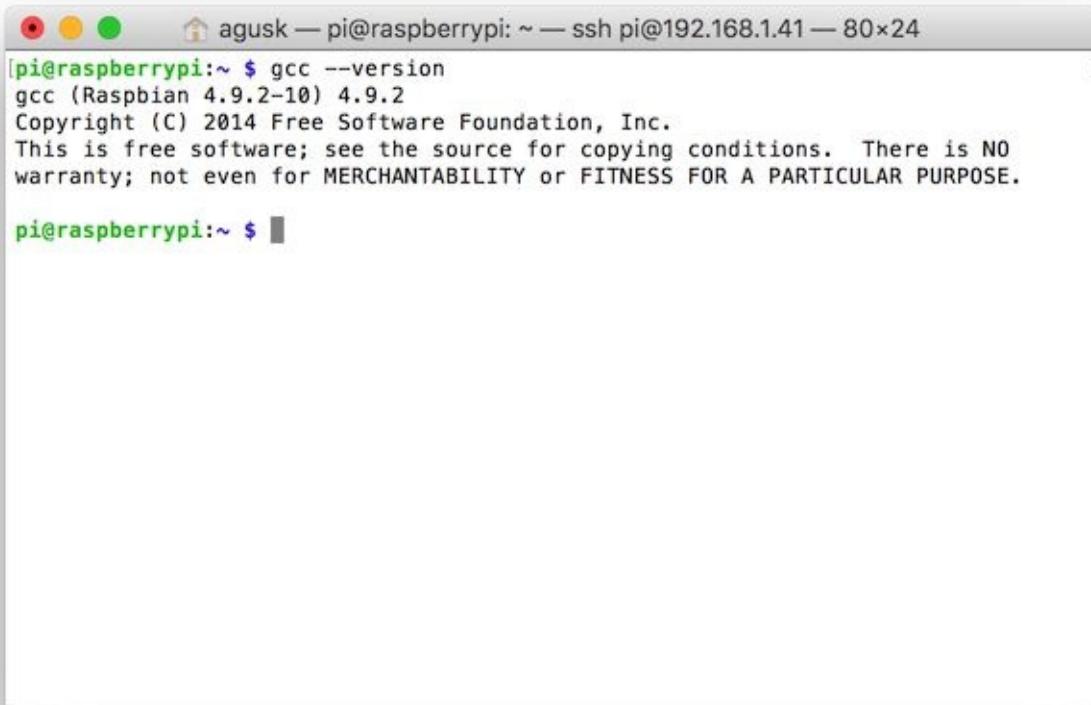


```
[pi@raspberrypi:~ $ python
Python 2.7.9 (default, Mar  8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
[>>> a = 5
[>>> b = 9
[>>> c = a*b
[>>> c
45
>>>
```

## 5.3 C/C++

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

```
$ gcc --version
```



The screenshot shows a terminal window with a light gray background and a dark gray title bar. The title bar displays the session information: agusk — pi@raspberrypi: ~ — ssh pi@192.168.1.41 — 80x24. The main area of the terminal shows the command \$ gcc --version followed by its output. The output includes the version number (gcc (Raspbian 4.9.2-10) 4.9.2), copyright information (Copyright (C) 2014 Free Software Foundation, Inc.), and a note about the software being free and having no warranty.

```
[pi@raspberrypi:~ $ gcc --version
gcc (Raspbian 4.9.2-10) 4.9.2
Copyright (C) 2014 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");
}
```

```
#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
```

```
agusk — pi@raspberrypi: ~/Documents — ssh pi@192.168.1.41 — 80x24
[pi@raspberrypi:~/Documents $ nano hello.c
[pi@raspberrypi:~/Documents $ ls
BlueJ Projects      hello.c          Wolfram Mathematica
Greenfoot Projects  Scratch Projects
[pi@raspberrypi:~/Documents $ gcc hello.c -o hello
[pi@raspberrypi:~/Documents $ ls
BlueJ Projects      hello           Scratch Projects
Greenfoot Projects  hello.c        Wolfram Mathematica
[pi@raspberrypi:~/Documents $ ./hello
Hello Worldpi@raspberrypi:~/Documents $ ]
```

## 5.4 Node.js

If you are node.js lovers, Raspbian Jessie has installed for you.

Try to check the node.js version

```
$ node -v
```



The screenshot shows a terminal window with a light gray background and a dark gray title bar. The title bar contains the text 'agusk — pi@raspberrypi: ~/Documents — ssh pi@192.168.1.41 — 80x24'. The main area of the terminal shows the command 'node --version' being run, followed by its output 'v0.10.29'. The terminal window has a rounded rectangular shape with a slight shadow at the bottom.

```
[pi@raspberrypi:~/Documents $ node --version
v0.10.29
pi@raspberrypi:~/Documents $ ]
```

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');
```

agusk — pi@raspberrypi: ~/Documents — ssh pi@192.168.1.41 — 80x24

GNU nano 2.2.6 File: mynode.js Modified

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

^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 execute **mynode.js** file using node.js

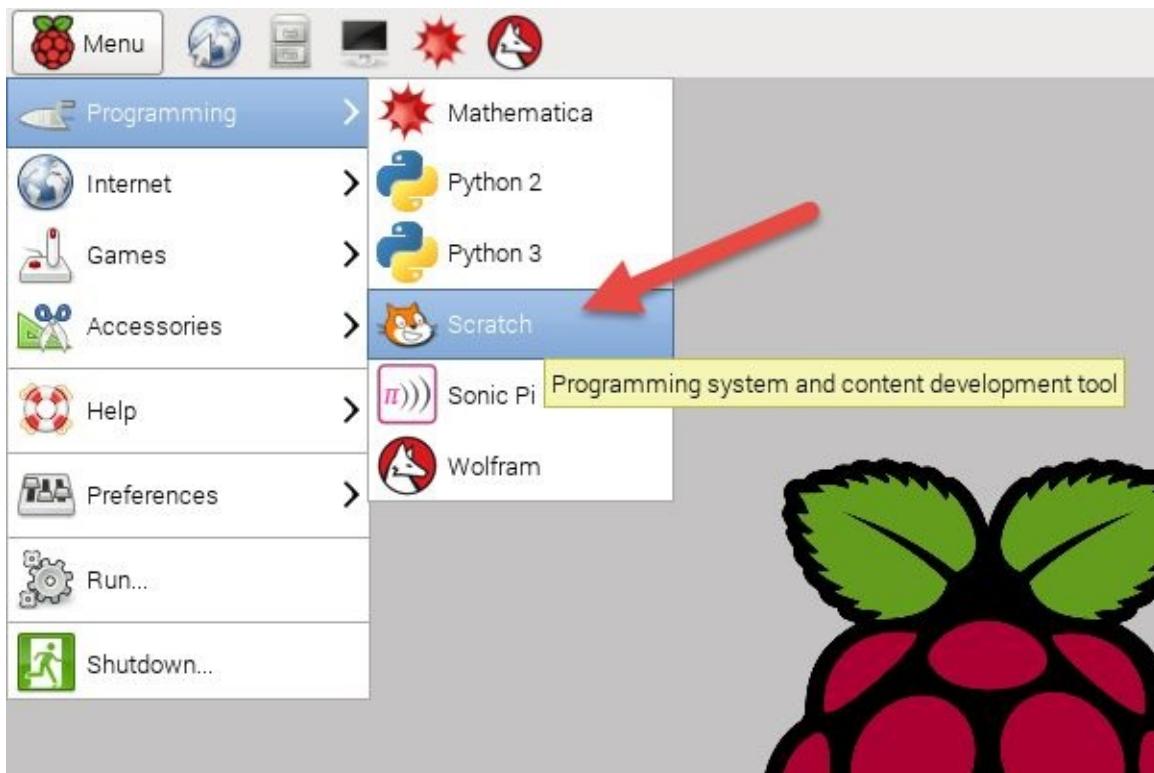
```
$ node mynode.js
```

```
agusk — pi@raspberrypi: ~/Documents — ssh pi@192.168.1.41 — 80x24
[pi@raspberrypi:~/Documents $ node -v
v0.10.29
[pi@raspberrypi:~/Documents $ nano mynode.js
[pi@raspberrypi:~/Documents $ ls
BlueJ Projects    hello      mynode.js      Wolfram Mathematica
Greenfoot Projects hello.c   Scratch Projects
[pi@raspberrypi:~/Documents $ node mynode.js
hello node.js
[pi@raspberrypi:~/Documents $ ]
```

## 5.5 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).

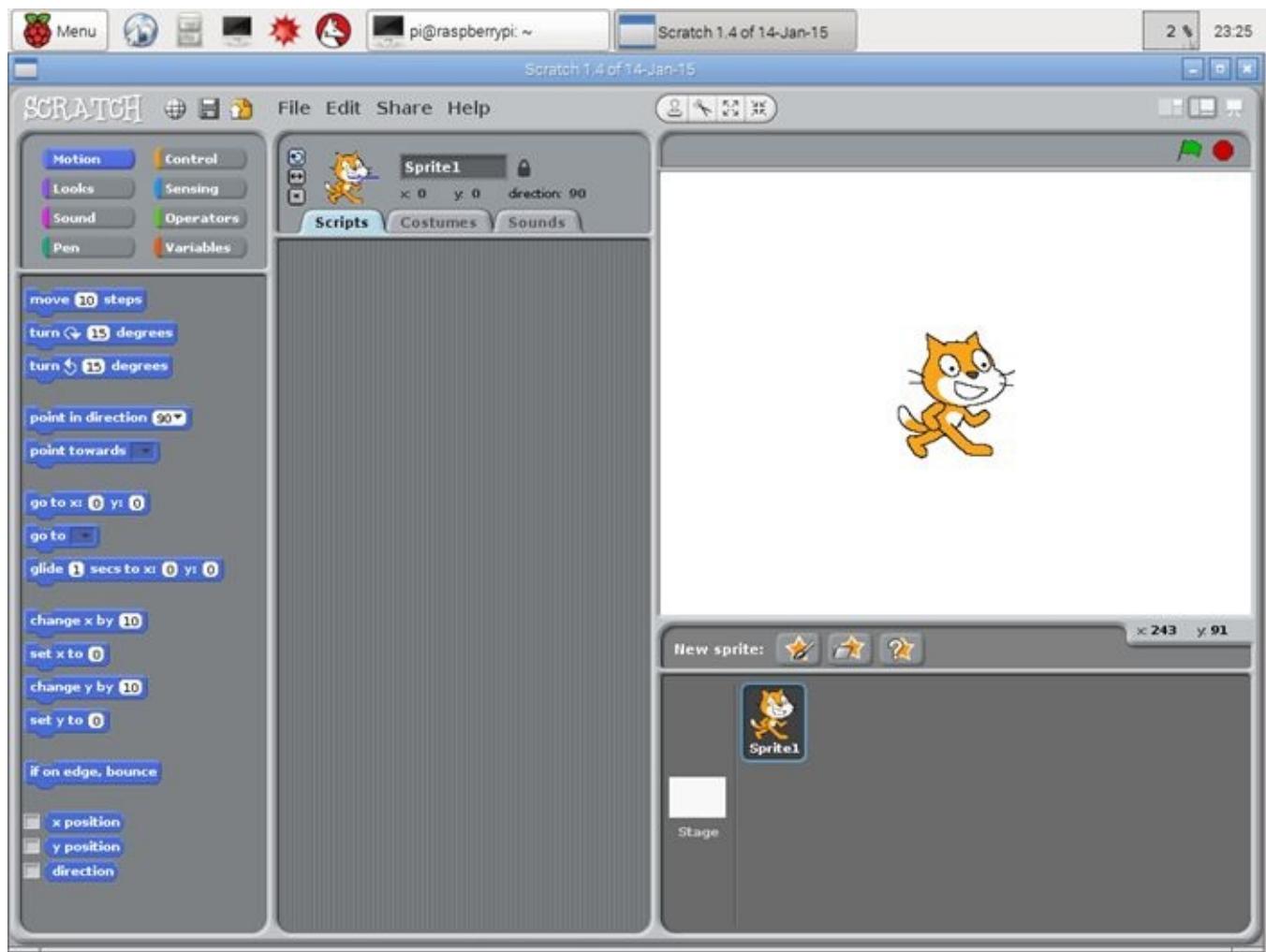


After that, you will get a Scratch application.

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

```
$ sudo scratch
```

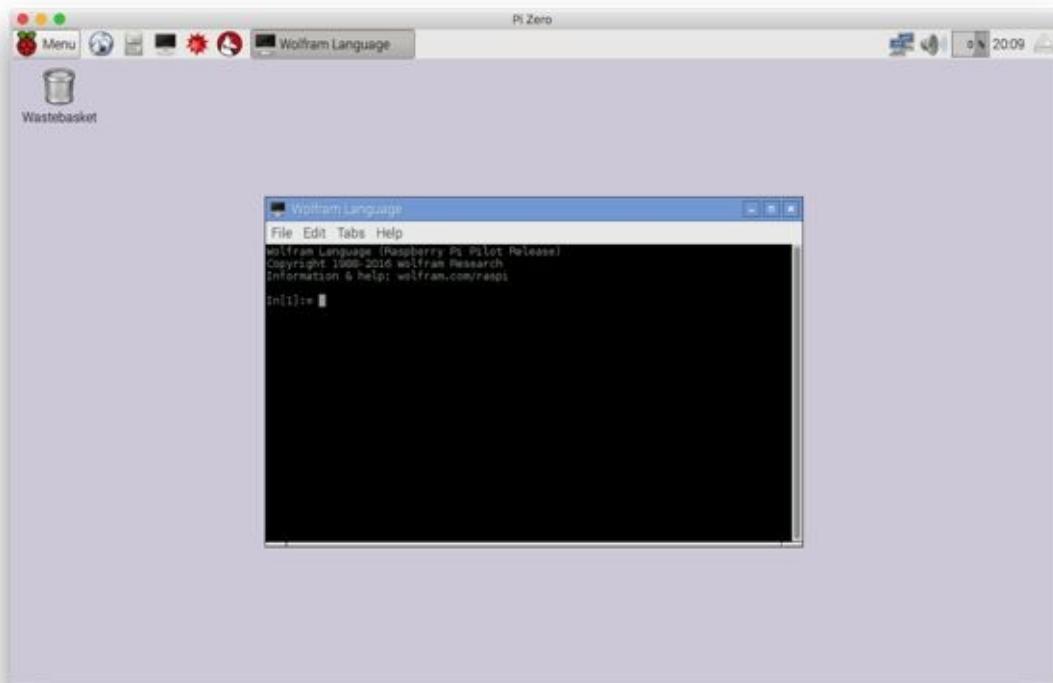
A sample output of Scratch application can be seen in Figure below.



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

## 5.6 Wolfram Mathematica

Now you can get Wolfram Mathematica for Raspberry Pi for free license. You can find Wolfram Mathematica on desktop.



You can read how to use Wolfram Mathematica on this site, <http://www.wolfram.com/raspberry-pi/>.

I also wrote a book, **Getting Started with Wolfram Language and Mathematica for**

**Raspberry Pi.** You can reach it on <http://blog.aguskurniawan.net/post/pimath.aspx> .



Getting Started with  
**Wolfram Language &  
Mathematica**  
for **Raspberry Pi**

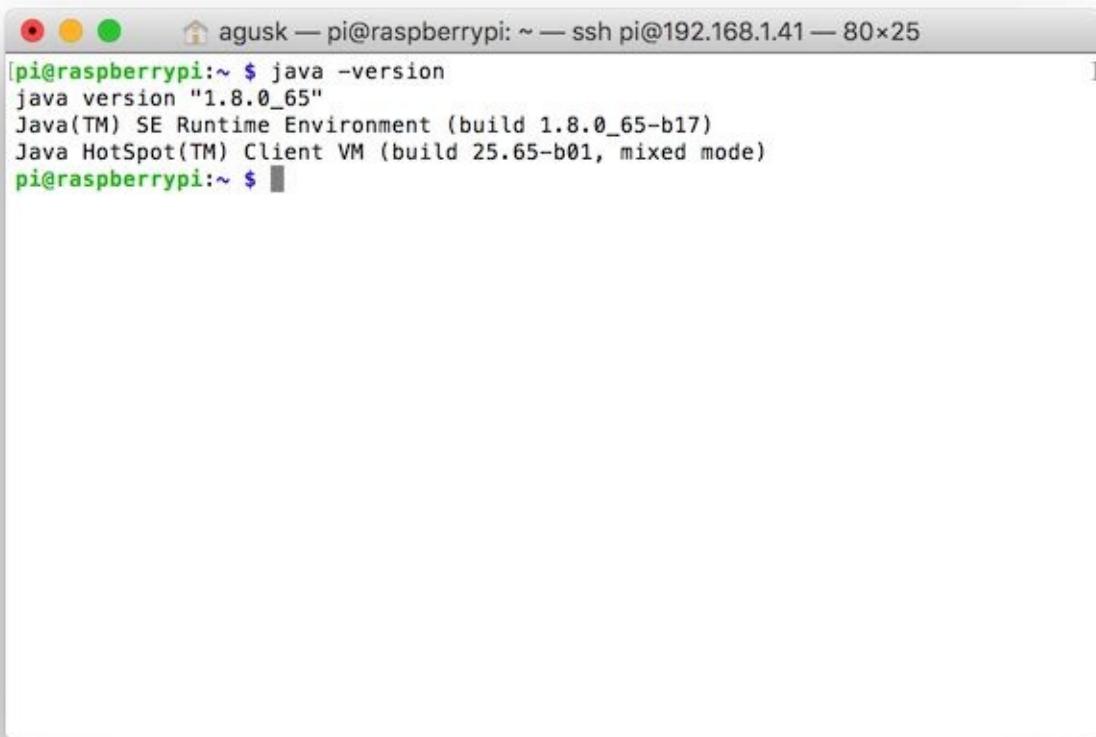
Agus Kurniawan

## 5.7 Java

Java also has installed on Raspbian Jessie. You can verify it typing this command on Terminal.

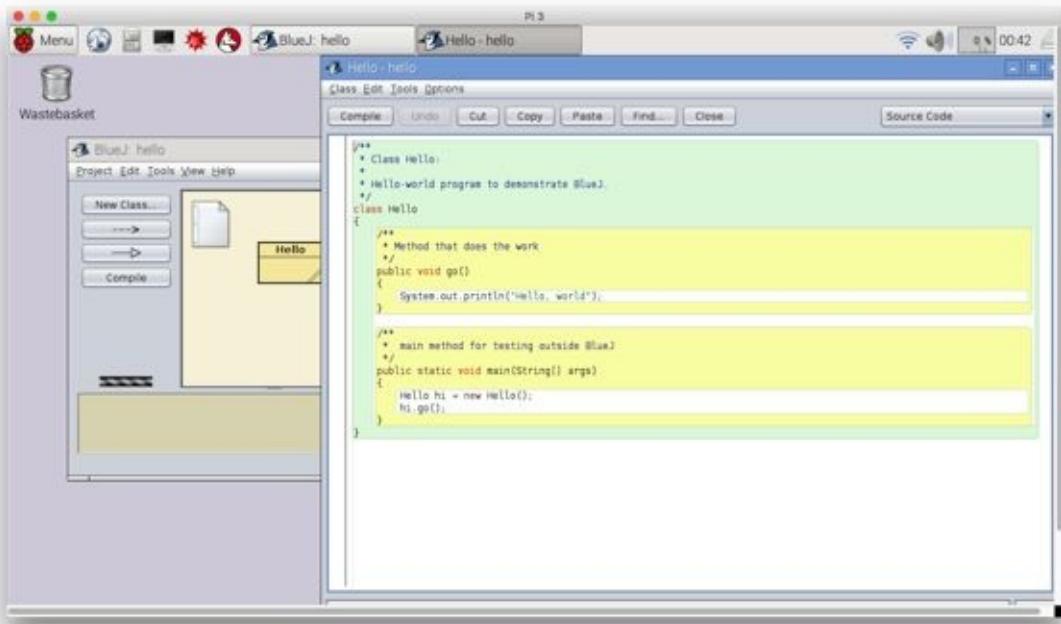
```
$ java -version
```

You should see Java version which installed on Raspberry Pi 3.



```
agusk — pi@raspberrypi: ~ — ssh pi@192.168.1.41 — 80x25
[pi@raspberrypi:~ $ java -version
java version "1.8.0_65"
Java(TM) SE Runtime Environment (build 1.8.0_65-b17)
Java HotSpot(TM) Client VM (build 25.65-b01, mixed mode)
pi@raspberrypi:~ $ ]
```

Raspbian Jessie has BlueJ for Java IDE too.



```
/**  
 * Class Hello  
 *  
 * Hello-world program to demonstrate BlueJ.  
 */  
class Hello  
{  
    /**  
     * Method that does the work  
     */  
    public void go()  
    {  
        System.out.println("Hello, world");  
    }  
  
    /**  
     * main method for testing outside BlueJ  
     */  
    public static void main(String[] args)  
    {  
        Hello hi = new Hello();  
        hi.go();  
    }  
}
```

For testing, you can write these codes on text editor such as nano or vi.

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

The screenshot shows a terminal window titled "agusk — pi@raspberrypi: ~/Documents — ssh pi@192.168.1.41 — 80x25". The title bar also displays "GNU nano 2.2.6", "File: HelloWorld.java", and "Modified". The main area of the terminal contains the following Java code:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

At the bottom of the terminal window, there is a menu of keyboard shortcuts:

```
^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 these codes into a file, called HelloWorld.java. Now you can compile and run it on Terminal.

```
$ javac HelloWorld.java  
$ java HelloWorld
```

Program output:

```
agusk — pi@raspberrypi: ~/Documents — ssh pi@192.168.1.41 — 80x25
[pi@raspberrypi:~ $ java -version
java version "1.8.0_65"
Java(TM) SE Runtime Environment (build 1.8.0_65-b17)
Java HotSpot(TM) Client VM (build 25.65-b01, mixed mode)
[pi@raspberrypi:~ $ ls
Desktop  Downloads  Pictures  python_games  Videos
Documents  Music  Public  Templates
[pi@raspberrypi:~ $ cd Documents/
[pi@raspberrypi:~/Documents $ nano HelloWorld.java
[pi@raspberrypi:~/Documents $ ls
BlueJ Projects      HelloWorld.java  Scratch Projects
Greenfoot Projects  leddemo.py
[pi@raspberrypi:~/Documents $ javac HelloWorld.java
[pi@raspberrypi:~/Documents $ java HelloWorld
Hello, World
pi@raspberrypi:~/Documents $ ]
```

## **6. Working with Bluetooth and iBeacon**

This chapter explains how to work with built-in Bluetooth on Raspberry Pi 3.

## **6.1 Getting Started**

In this chapter, we learn how to use built-in Bluetooth on Raspberry Pi 3 board.

Let's start.

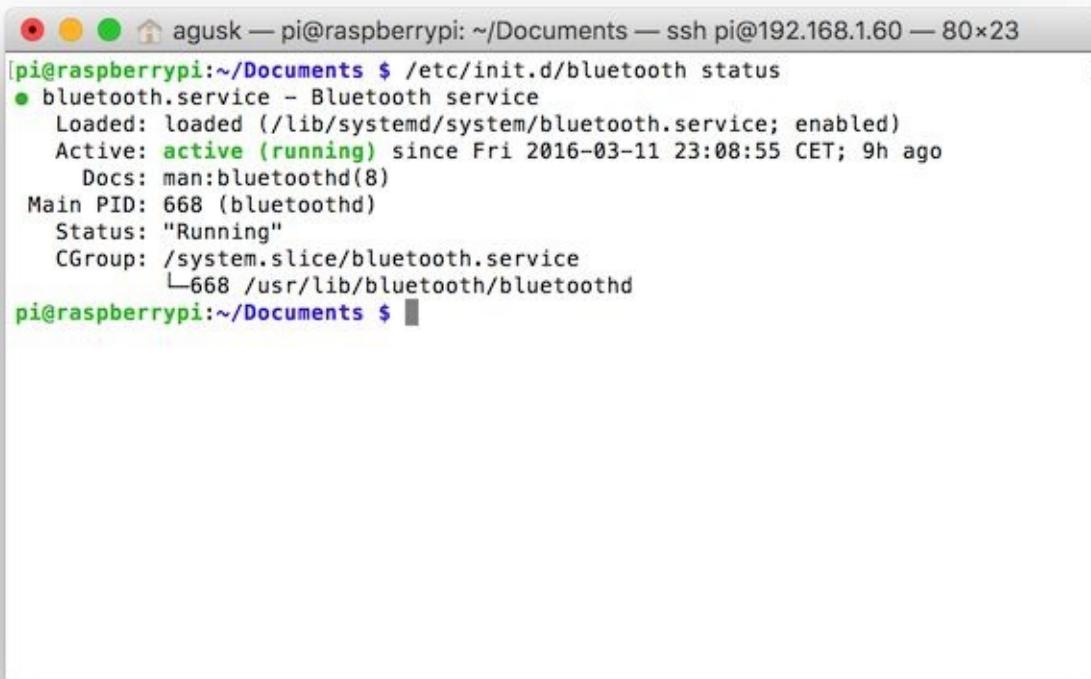
## 6.2 Setting up Bluetooth

Raspbian Jessie already installed Bluetooth library so we can use it.

Now we check status of Bluetooth device using this command.

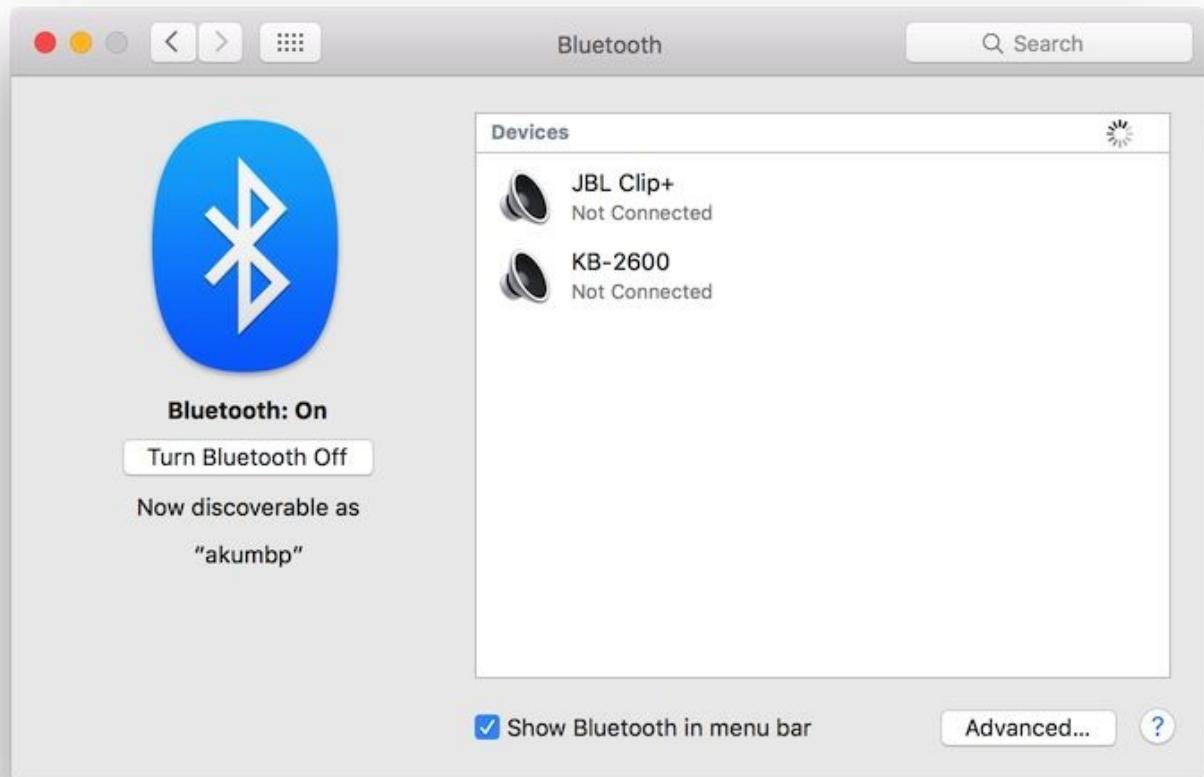
```
$ /etc/init.d/bluetooth status
```

By default, Bluetooth device is enabled and running.



```
[pi@raspberrypi:~/Documents $ /etc/init.d/bluetooth status
● bluetooth.service - Bluetooth service
  Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled)
  Active: active (running) since Fri 2016-03-11 23:08:55 CET; 9h ago
    Docs: man:bluetoothd(8)
   Main PID: 668 (bluetoothd)
     Status: "Running"
       CGrou[...]
```

I use Macbook to verify existence of Raspberry Pi Bluetooth. I don't see Raspberry Pi Bluetooth.



This means Raspberry Pi Bluetooth is not discoverable.

We can set it using this command.

```
$ sudo hciconfig hci0 piscan
```

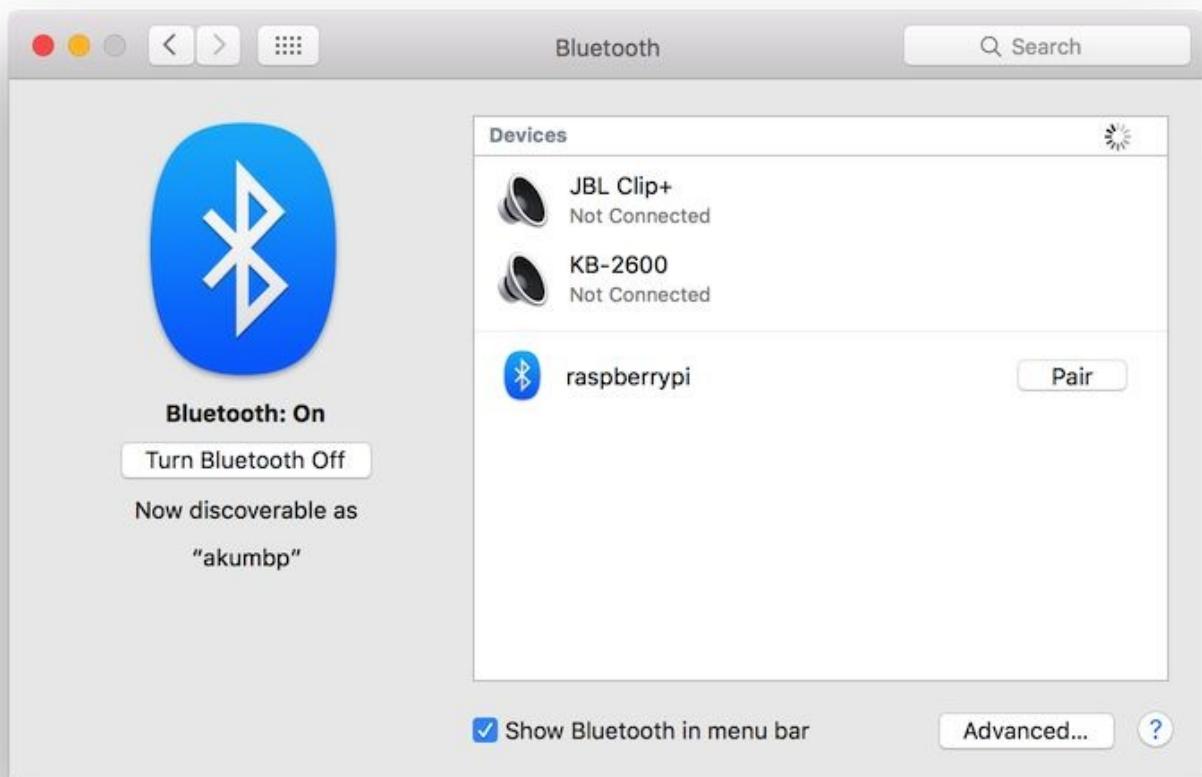
Then, we can use our Raspberry Pi Bluetooth using hciconfig.

```
$ sudo hciconfig -a
```

```
agusk — pi@raspberrypi: ~/Documents — ssh pi@192.168.1.60 — 80x23
[pi@raspberrypi:~/Documents $ sudo hciconfig hci0 piscan
[pi@raspberrypi:~/Documents $ sudo hciconfig -a
hci0:  Type: BR/EDR  Bus: UART
        BD Address: B8:27:EB:B5:B2:3B  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING PSCAN ISCAN
        RX bytes:774 acl:0 sco:0 events:51 errors:0
        TX bytes:1614 acl:0 sco:0 commands:49 errors:0
        Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH SNIFF
        Link mode: SLAVE ACCEPT
        Name: 'raspberrypi'
        Class: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Version: 4.1 (0x7)  Revision: 0xb6
        LMP Version: 4.1 (0x7)  Subversion: 0x2209
        Manufacturer: Broadcom Corporation (15)

pi@raspberrypi:~/Documents $
```

Now my Macbook detected Raspberry Pi Bluetooth.





## 6.3 Ping Bluetooth

In this section, we want to pair my Macbook Bluetooth to Raspberry Pi Bluetooth.

Firstly, I scan my Macbook Bluetooth using hcitool to retrieve MAC address of Macbook Bluetooth.

```
$ sudo hcitool scan
```

To ping, you can use l2ping with passing target MAC address.

```
$ sudo l2ping -c 1 MAC_ADD_BLUETOOTH
```

If succeed, you should get a ping report.

A sample of pinging on Macbook Bluetooth.



```
agusk — pi@raspberrypi: ~/Documents — ssh pi@192.168.1.60 — 80x23
[pi@raspberrypi:~/Documents $ sudo hcitool scan
Scanning ...
D0:A6:37:[REDACTED] akumbp
[pi@raspberrypi:~/Documents $ sudo l2ping -c 1 D0:A6:37:[REDACTED]
Ping: D0:A6:37:[REDACTED] from B8:27:EB:[REDACTED] (data size 44) ...
44 bytes from D0:A6:37:[REDACTED] d 0 time 8.16ms
1 sent, 1 received, 0% loss
pi@raspberrypi:~/Documents $ ]
```

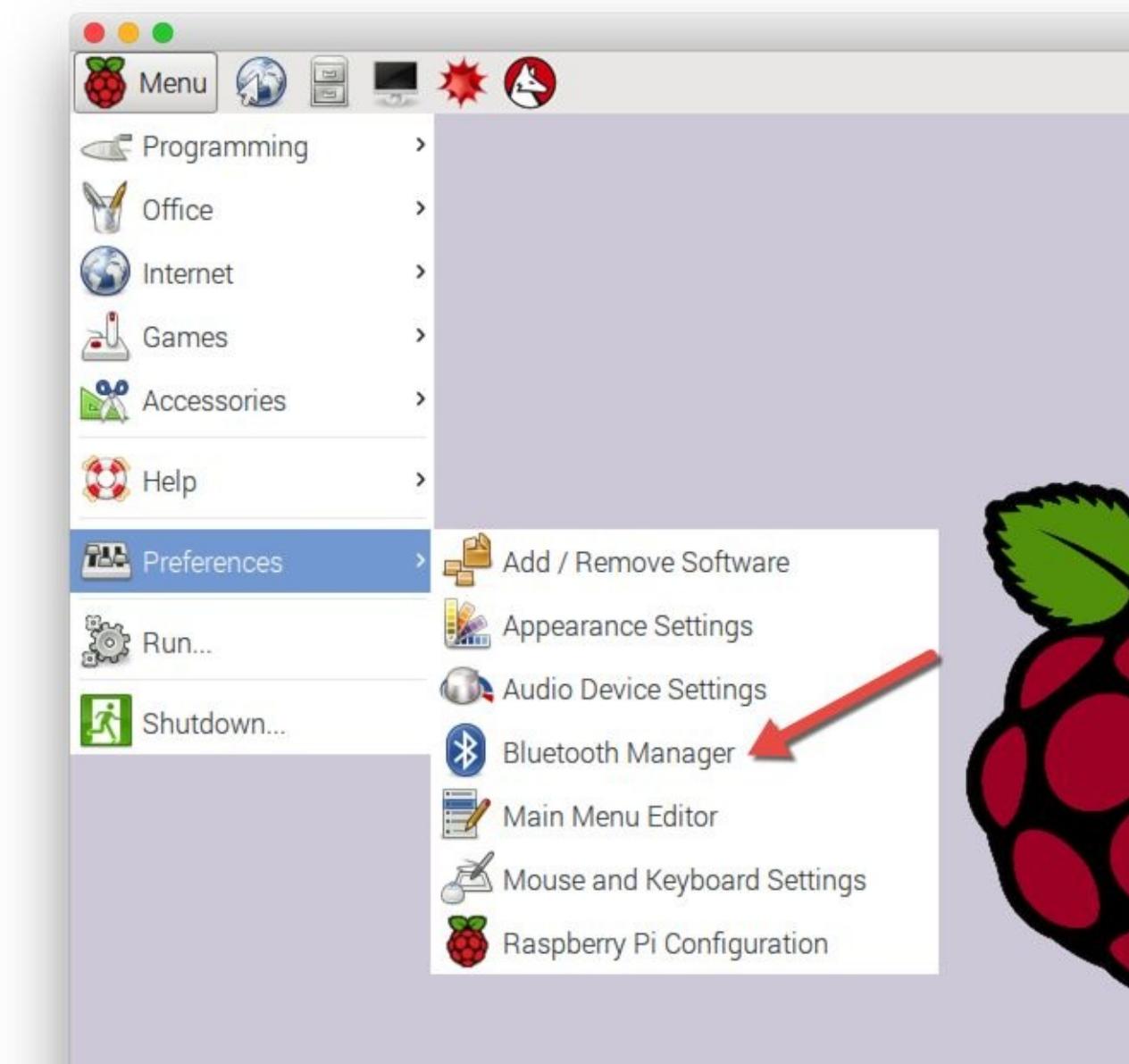
## 6.4 Working with Blueman Bluetooth Manager

In this section, we learn to manage Raspberry Pi 3 Bluetooth using GUI, Bluetooth Manager.

Let's start.

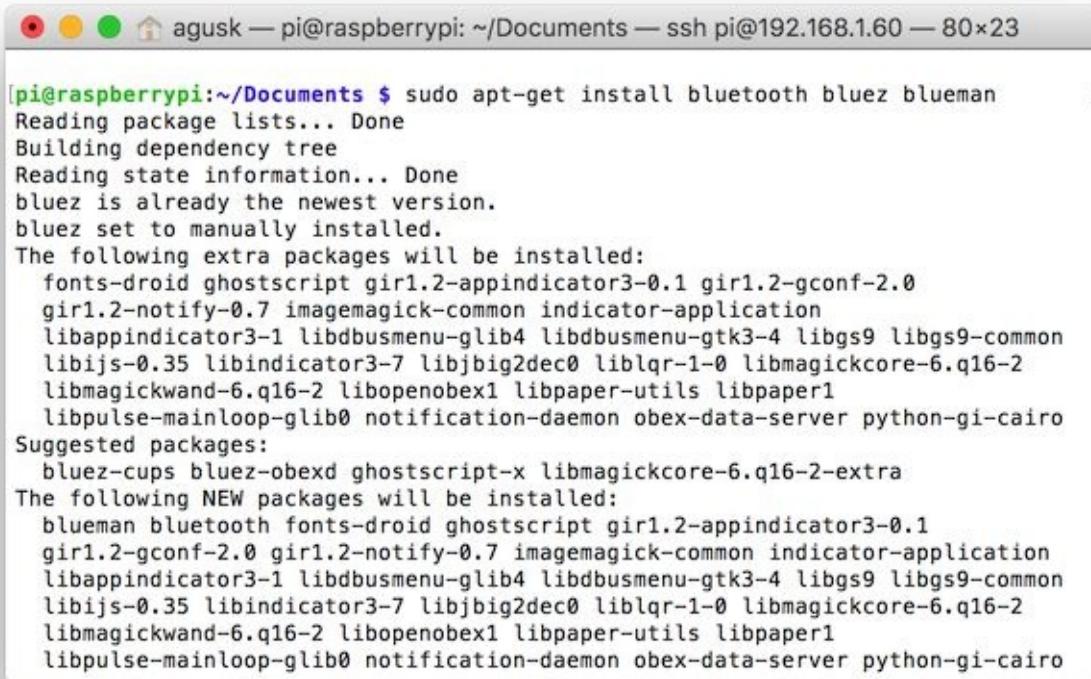
### 6.4.1 Setting up

Firstly, you run Raspbian Jessie in desktop mode. You should see Bluetooth Manager on Preferences menu, shown in Figure below.



If you don't see it, Raspbian Jessie may not install blueman. You can install it using this command on Raspberry Pi Terminal.

```
$ sudo apt-get install bluetooth bluez blueman
```



```
[pi@raspberrypi:~/Documents $ sudo apt-get install bluetooth bluez blueman
Reading package lists... Done
Building dependency tree
Reading state information... Done
bluez is already the newest version.
bluez set to manually installed.
The following extra packages will be installed:
  fonts-droid ghostscript gir1.2-appindicator3-0.1 gir1.2-gconf-2.0
  gir1.2-notify-0.7 imagemagick-common indicator-application
  libappindicator3-1 libdbusmenu-glib4 libdbusmenu-gtk3-4 libgs9 libgs9-common
  libijs-0.35 libindicator3-7 libjbig2dec0 liblqr-1-0 libmagickcore-6.q16-2
  libmagickwand-6.q16-2 libopenobex1 libpaper-utils libpaper1
  libpulse-mainloop-glib0 notification-daemon obex-data-server python-gi-cairo
Suggested packages:
  bluez-cups bluez-obexd ghostscript-x libmagickcore-6.q16-2-extra
The following NEW packages will be installed:
  blueman bluetooth fonts-droid ghostscript gir1.2-appindicator3-0.1
  gir1.2-gconf-2.0 gir1.2-notify-0.7 imagemagick-common indicator-application
  libappindicator3-1 libdbusmenu-glib4 libdbusmenu-gtk3-4 libgs9 libgs9-common
  libijs-0.35 libindicator3-7 libjbig2dec0 liblqr-1-0 libmagickcore-6.q16-2
  libmagickwand-6.q16-2 libopenobex1 libpaper-utils libpaper1
  libpulse-mainloop-glib0 notification-daemon obex-data-server python-gi-cairo
```

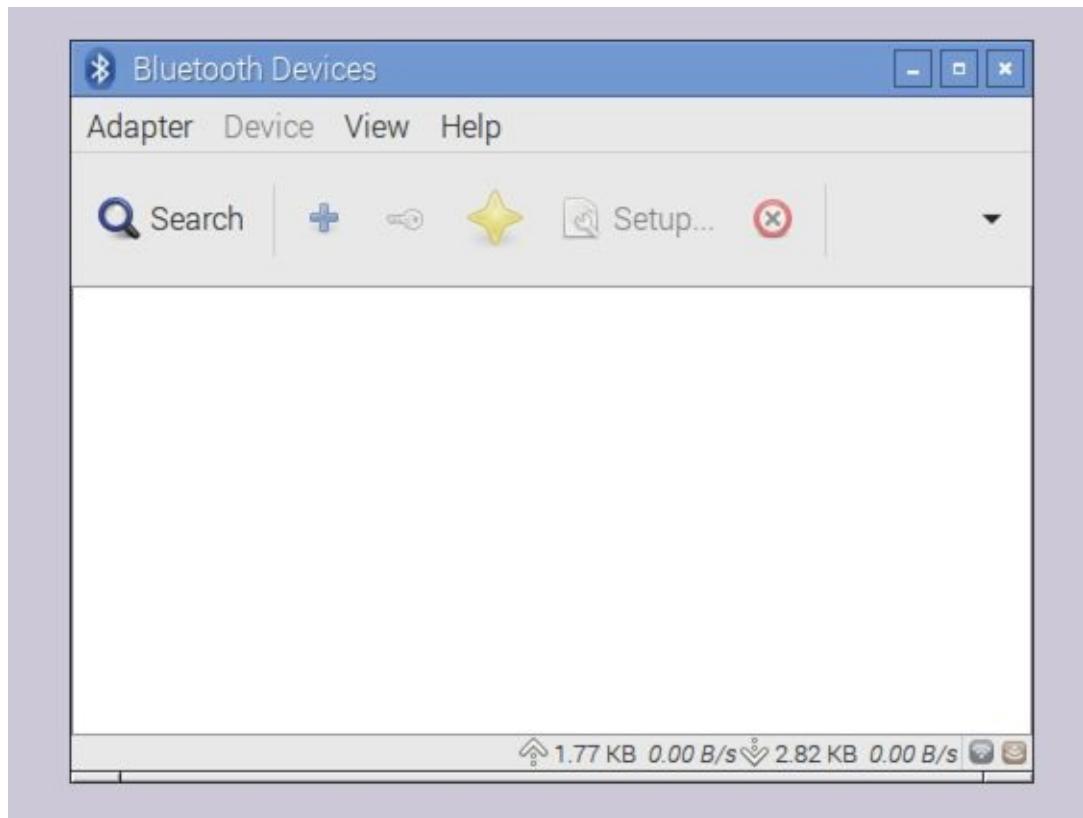
Then, add your account into member of bluetooth.

```
$ sudo usermod -G bluetooth -a pi
```

After installed, please reboot bluetooth service.

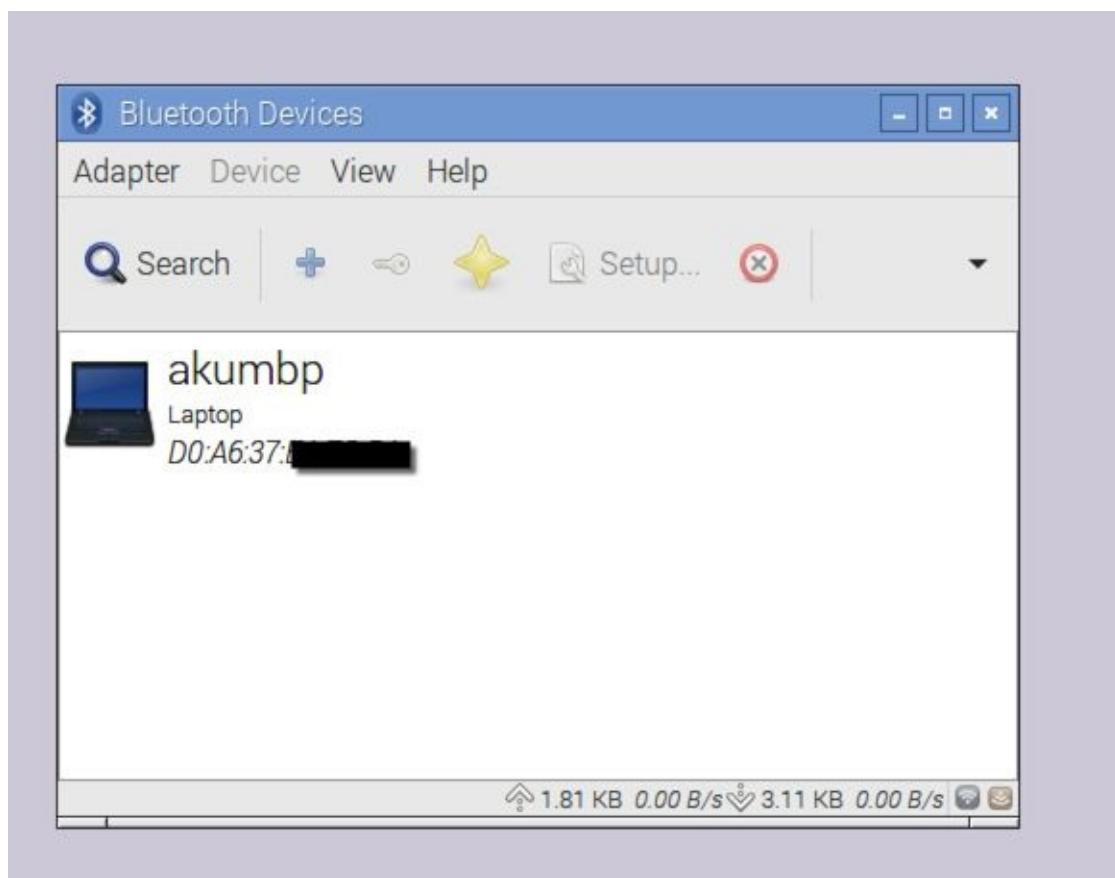
```
$ sudo service bluetooth restart
```

Now you should Bluetooth Manager now. You can open it and the get the following dialog.



You can click Search icon to scan Bluetooth beacons.

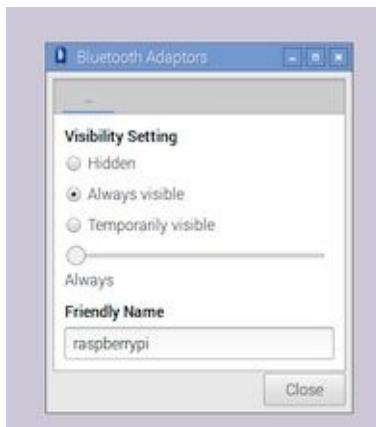
For example, Bluetooth Manager detected my Macbook Bluetooth.



We can configure Raspberry Pi Bluetooth in always visible mode. You can click Bluetooth icon. Then, a menu is shown.



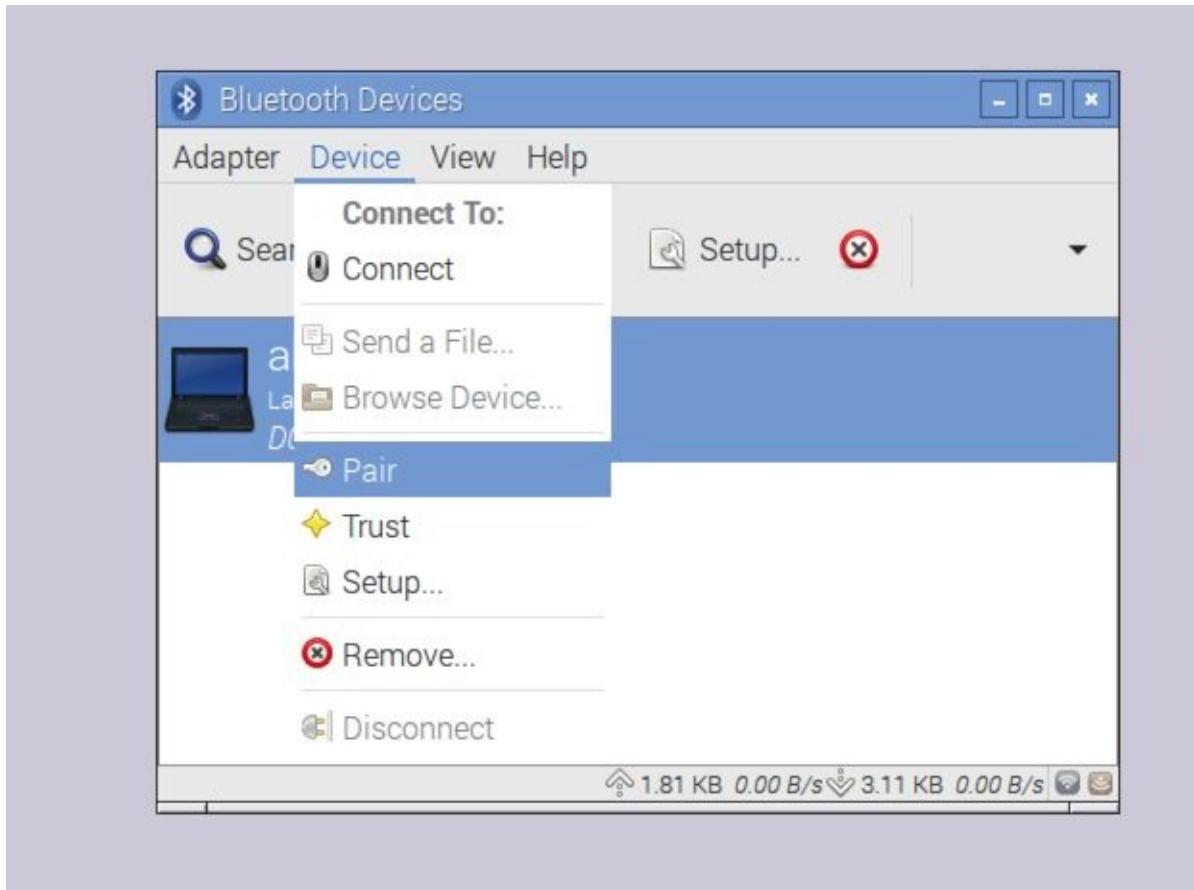
Select Adapters. Next, you can select Always visible.



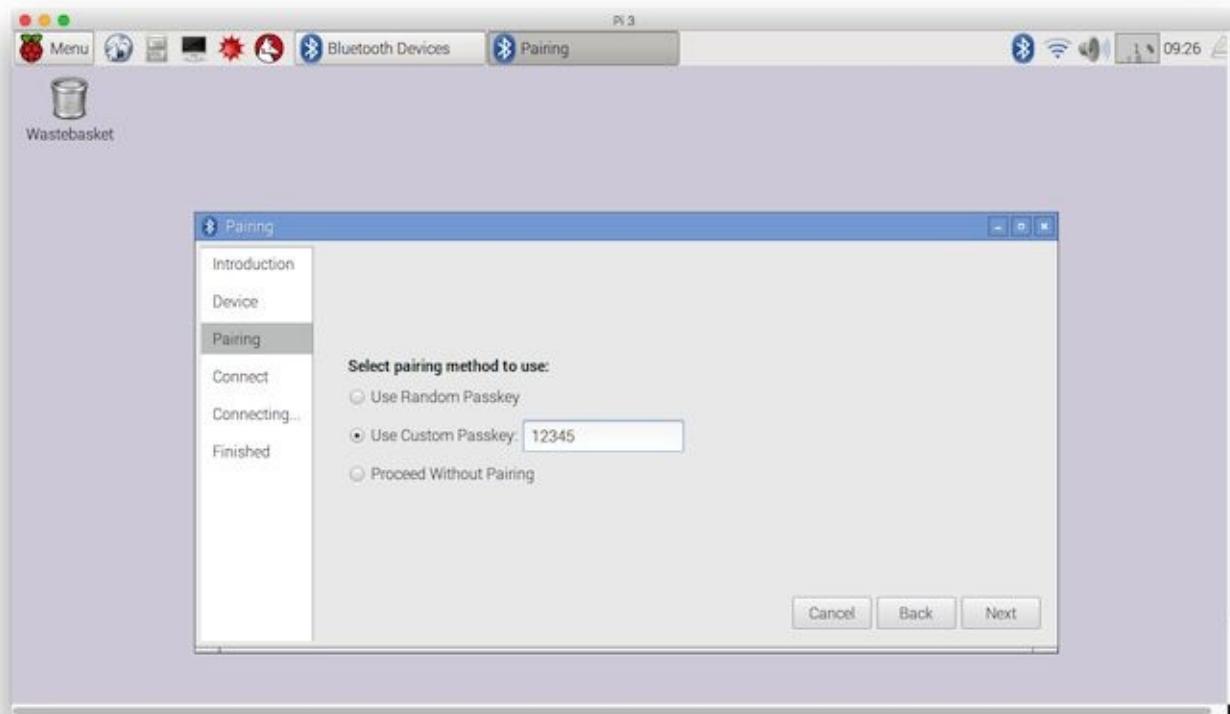
## 6.4.2 Pairing

In this section, we learn how to pair Raspberry Pi Bluetooth to other Bluetooth device.  
How to pair?

Open Bluetooth Manager. Search and find Bluetooth device which you want to pair. After found, select menu Device -> Pair.

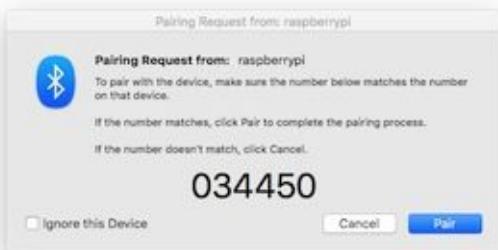


Then, you should get a dialog as follows.

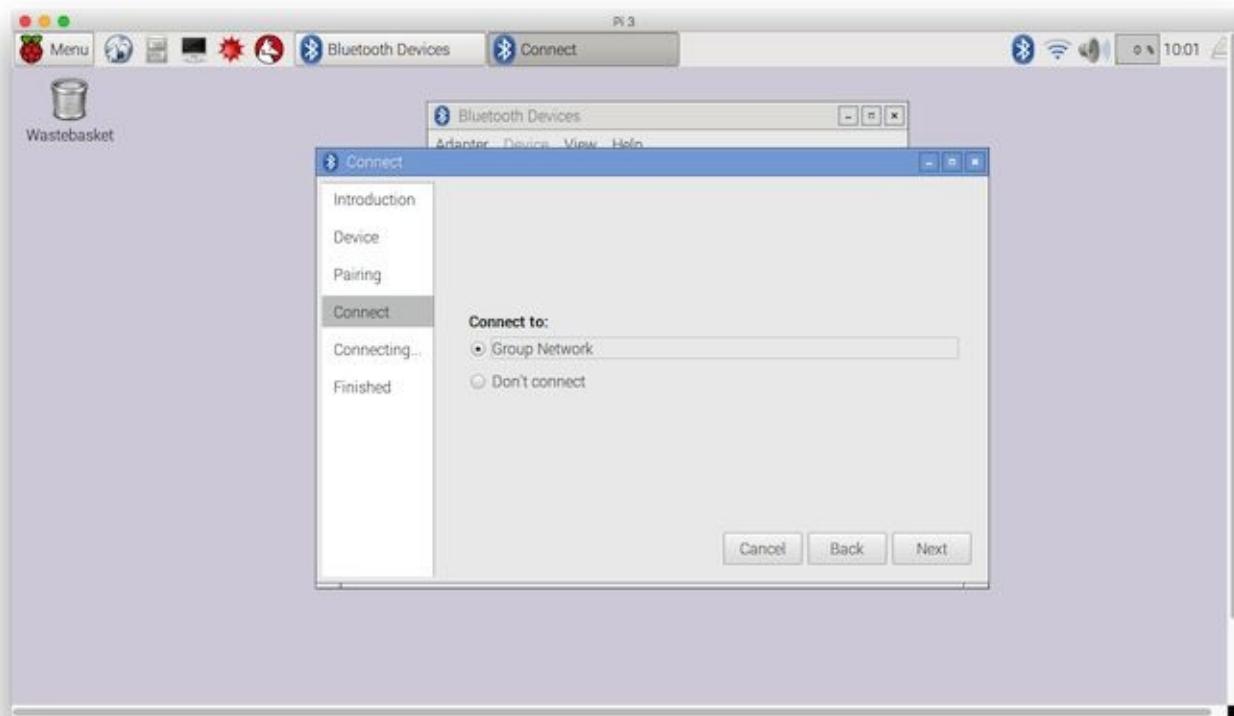


You can select Random Passkey or Custom Passkey.

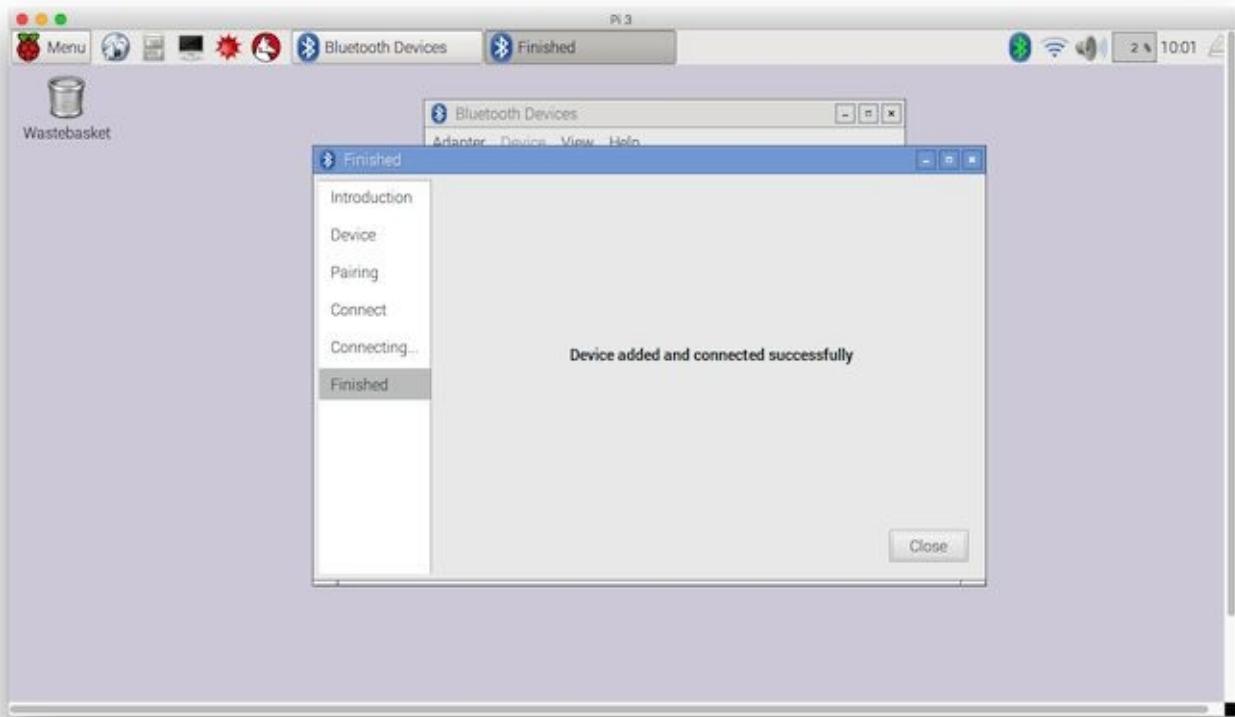
On target Bluetooth machine, it should get notification for pairing.



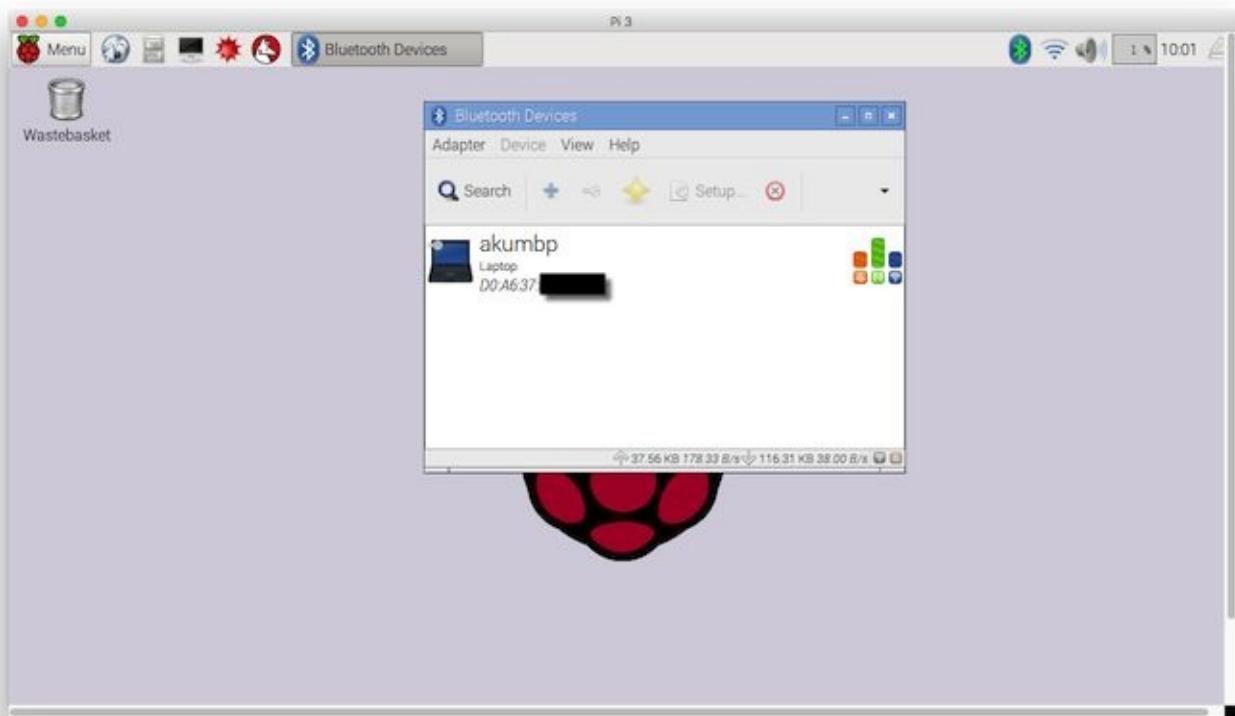
After confirmed for pairing, we try to connect to Group Network



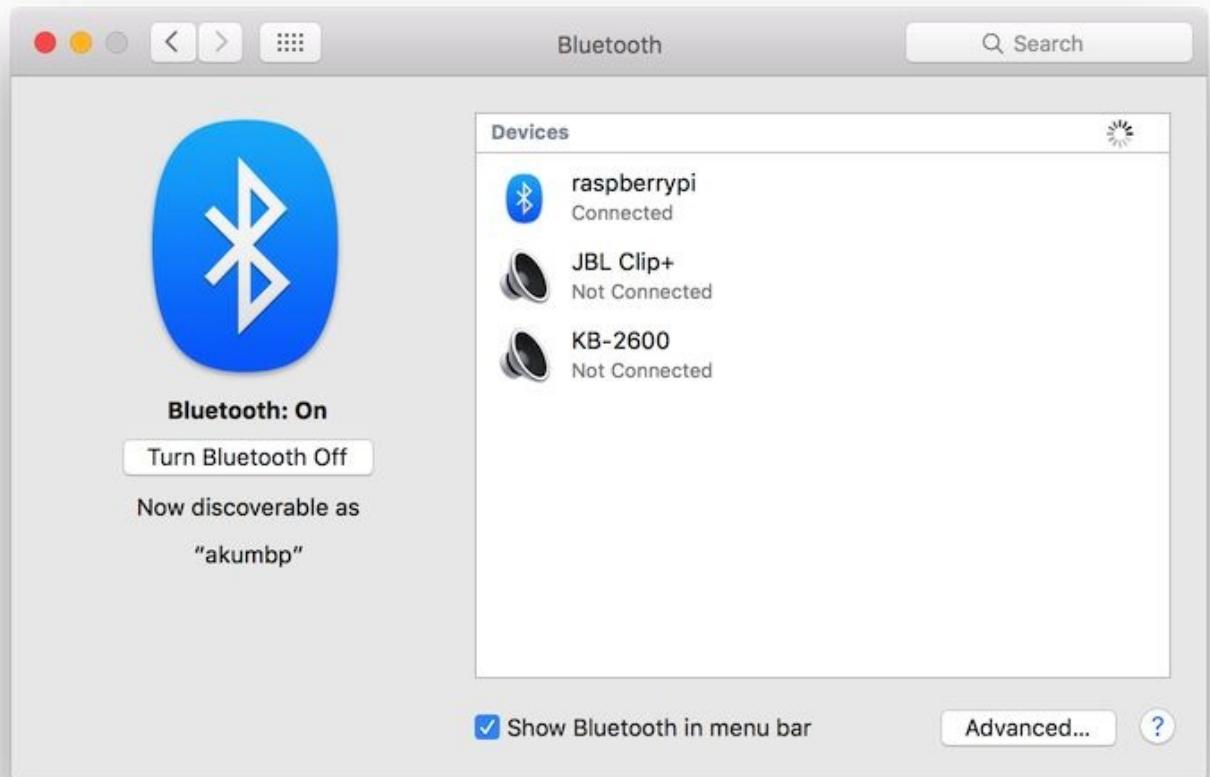
Click Close if done.



Now you can see paired Bluetooth on Bluetooth Manager.

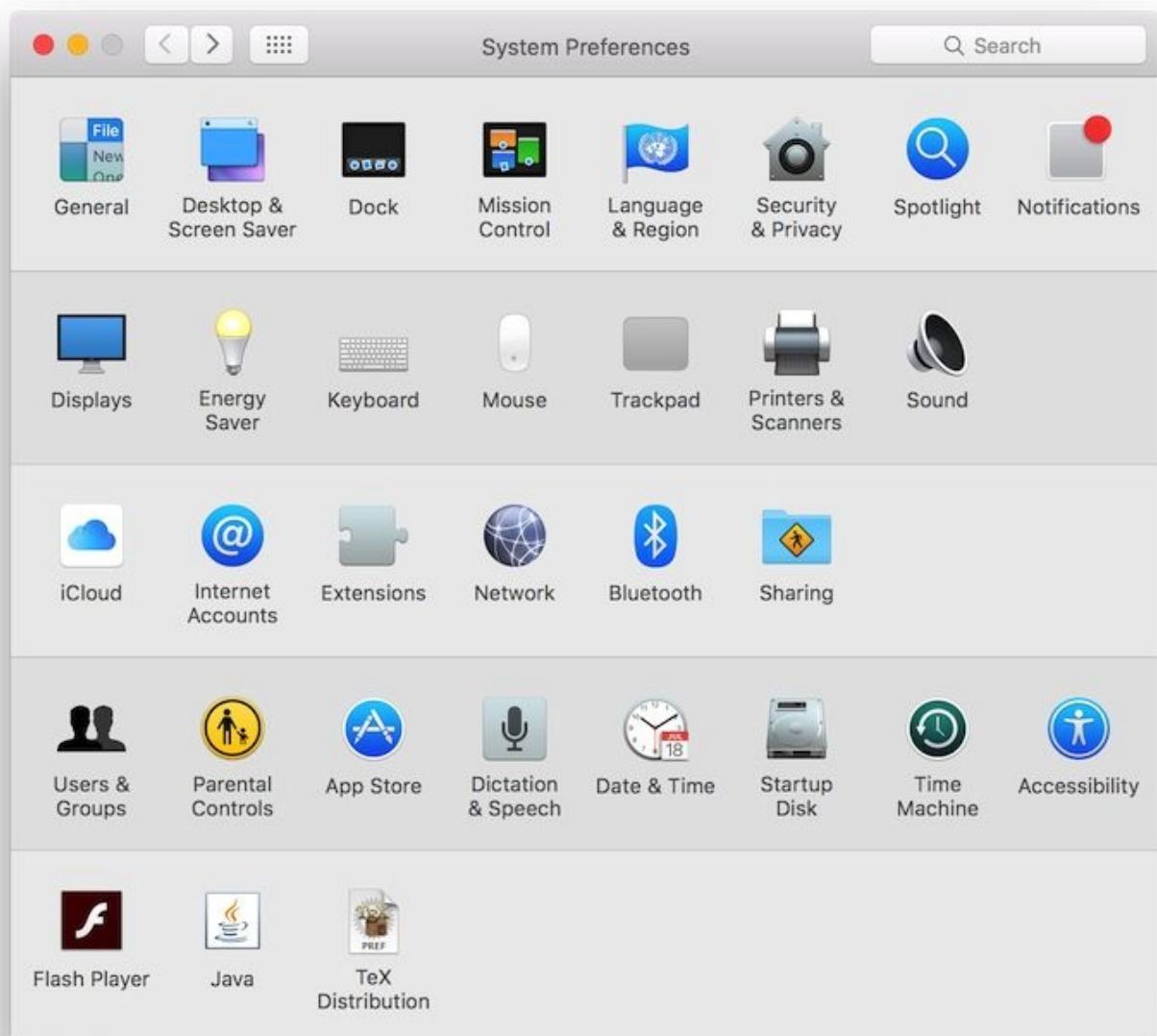


On target Bluetooth, for instance, my Mackbook shown paired with Raspberry Pi Bluetooth.

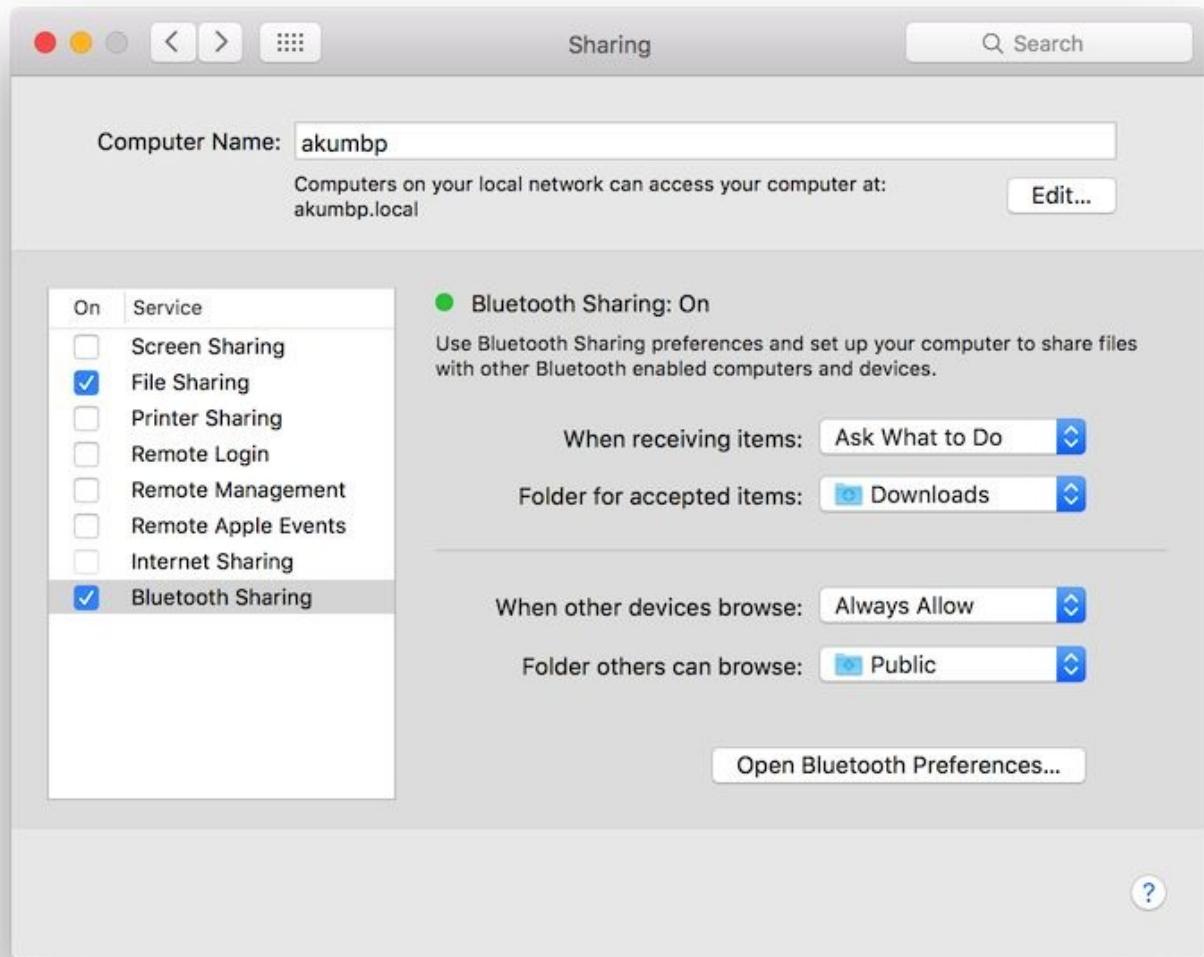


### 6.4.3 Transferring File

After paired with other Bluetooth device, we can transfer a file. In this scenario, I use Macbook. I should configure my Bluetooth can receive a file.

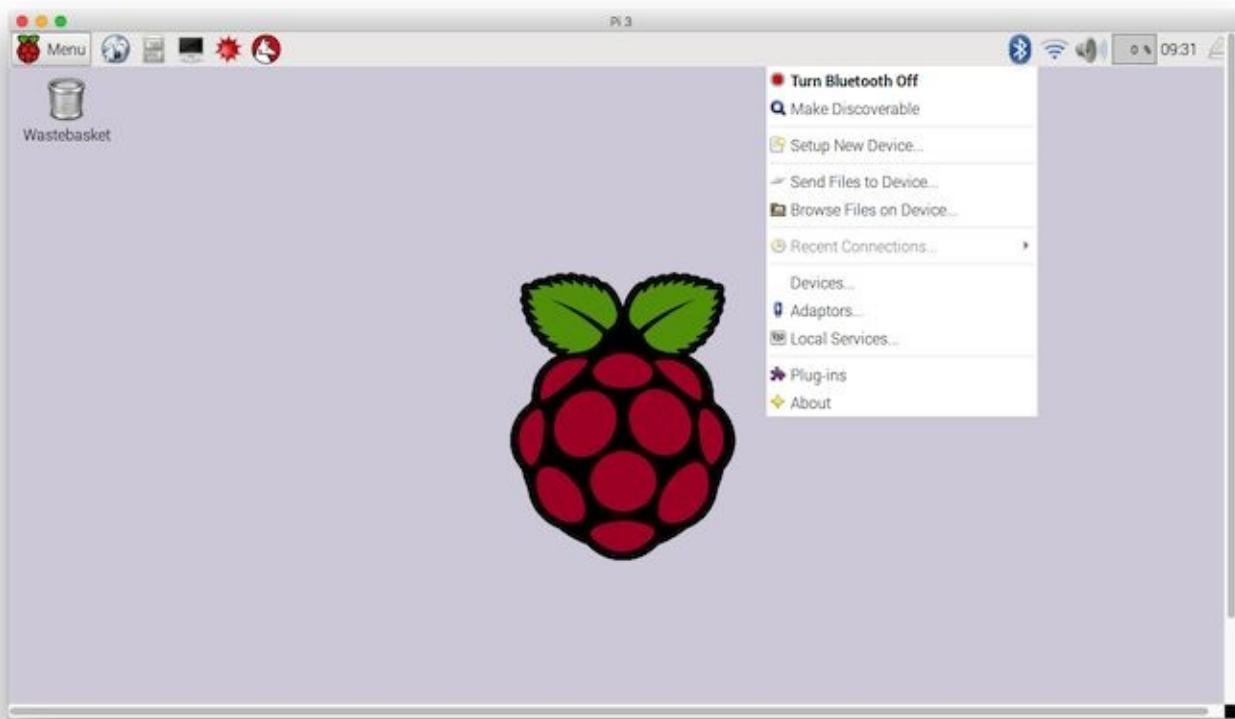


On System Preferences, select Sharing.

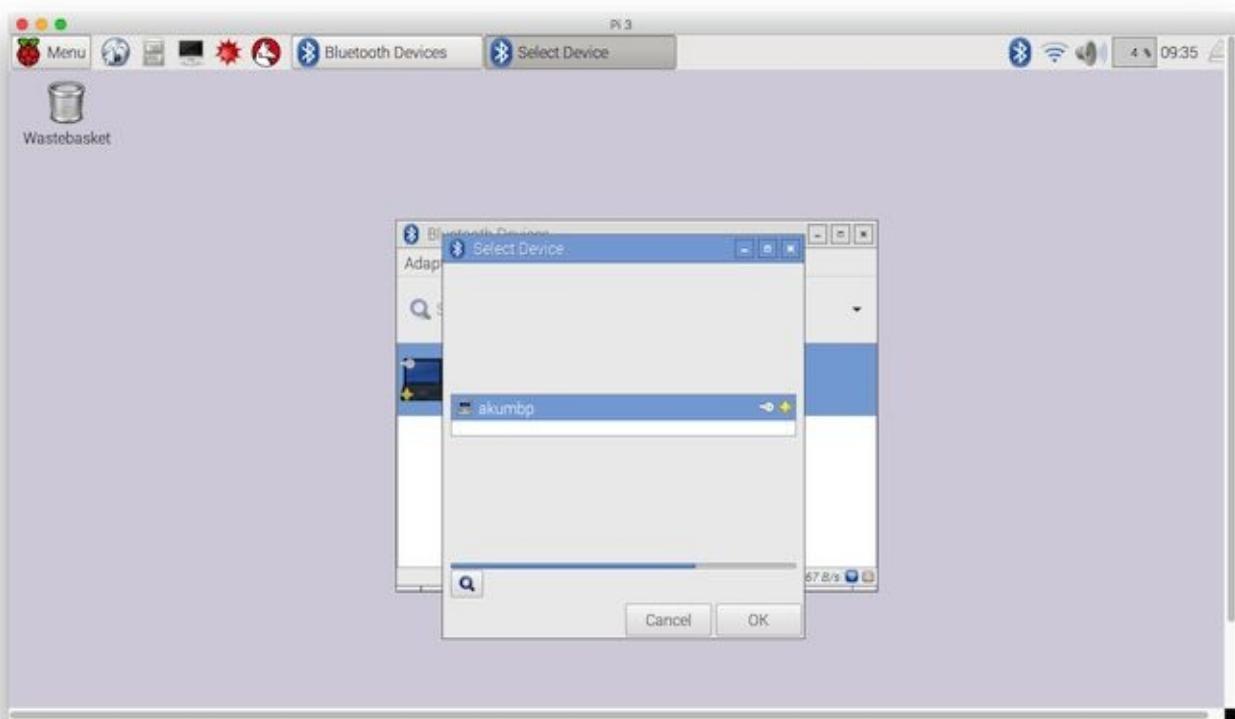


Checked Bluetooth Sharing. Select Ask What to Do on When receiving items.

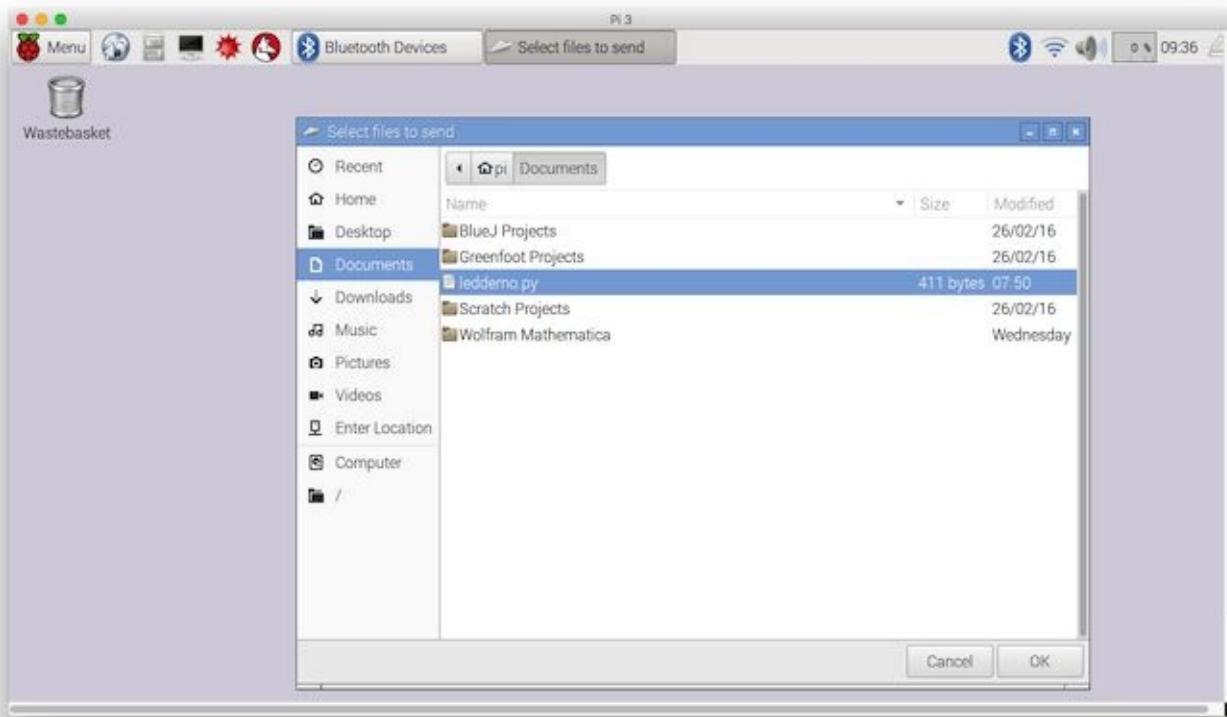
Now we send a file from Raspberry Pi. Select Bluetooth icon, select Send Files to Device.



You should get a dialog. Click OK for targeted Bluetooth.

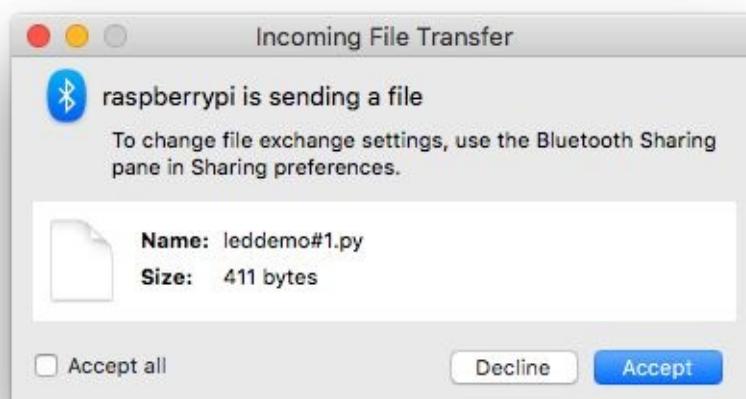


Choose a file.



This will send our file.

On target machine, Macbook, I got a confirmation, shown in Figure below.



Click Accept so you will get a file.

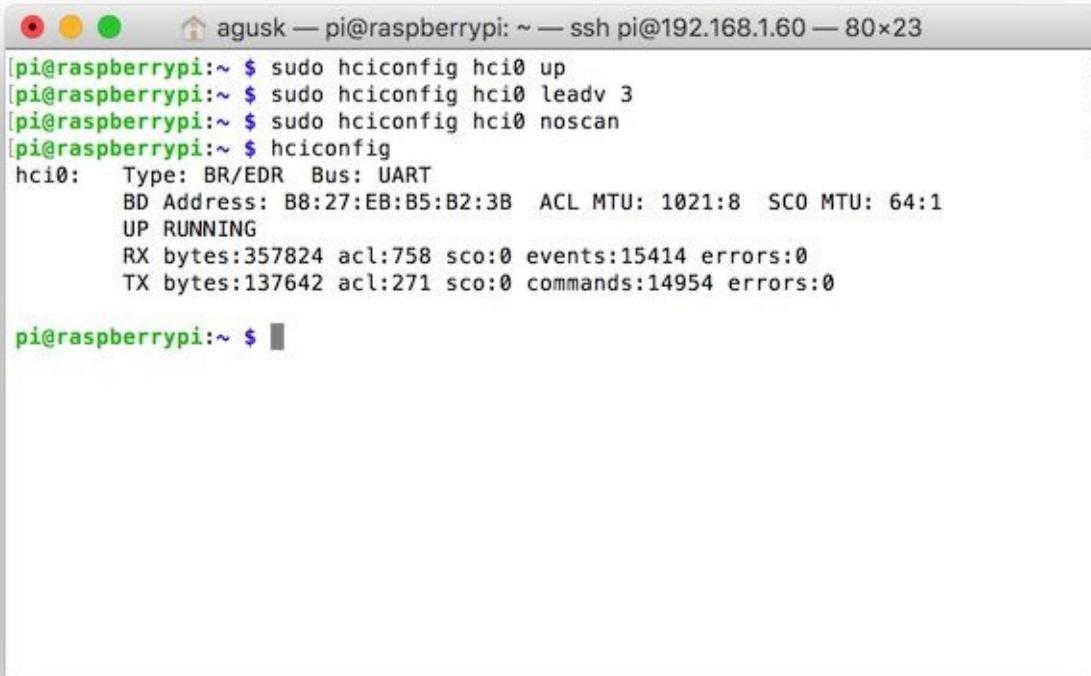
## 6.5 Building your own iBeacon

The last section, we try to build iBeacon using hcitool. Firstly, we configure Bluetooth for iBeacon mode. Type these commands.

```
$ sudo hciconfig hci0 up  
$ sudo hciconfig hci0 leadv 3  
$ sudo hciconfig hci0 noscan
```

Then, verify it using hciconfig.

```
$ hciconfig
```



A screenshot of a terminal window titled "agusk — pi@raspberrypi: ~ — ssh pi@192.168.1.60 — 80x23". The window shows the command history and the output of the hciconfig command. The output details the configuration of the hci0 interface, including its type (BR/EDR), bus (UART), BD Address (B8:27:EB:B5:B2:3B), ACL MTU (1021:8), SCO MTU (64:1), and various statistics for RX and TX bytes.

```
[pi@raspberrypi:~ $ sudo hciconfig hci0 up  
[pi@raspberrypi:~ $ sudo hciconfig hci0 leadv 3  
[pi@raspberrypi:~ $ sudo hciconfig hci0 noscan  
[pi@raspberrypi:~ $ hciconfig  
hci0:  Type: BR/EDR  Bus: UART  
        BD Address: B8:27:EB:B5:B2:3B  ACL MTU: 1021:8  SCO MTU: 64:1  
        UP RUNNING  
        RX bytes:357824 acl:758 sco:0 events:15414 errors:0  
        TX bytes:137642 acl:271 sco:0 commands:14954 errors:0  
pi@raspberrypi:~ $
```

For testing, I use Beacon Scan tool to listen iBeacon packet.

The screenshot shows the Mac App Store interface. At the top, there are navigation links: Featured, Top Charts, Categories, Purchased, and Updates. A search bar on the right contains the text "ibeacon". Below the search bar, the title "Beacon Scan" is displayed next to a circular icon showing signal strength. A "Get" button is visible. To the right of the main content area, there's a sidebar with links: Twocanoes Software, Inc. Web Site, Beacon Scan Support, and Privacy Policy. The main content area includes a "Feature Set:" section with bullet points: "See nearby beacons!", "Supports both iBeacons and Physical Web Beacons!", "Copy hard to type UUID and other beacon identifiers!", and "Works great with the best beacons on the market, Bleu Beacons from Twocanoes Labs...". A "...More" link is at the bottom of this list. On the left, there's a preview window showing the application's interface with a table of beacon data.

Type	UUID / URL	Major	Minor	Power (dBm)	RSSI (dB)	Last Seen
iBeacon	50003A2C-3F71-4714-BD22-63B98274B314	1	68	-48	-77	Sunday, M...
iBeacon	E2C56D88-4E8B-8D86-DF1A71096ED0	1	2	-40	-74	Sunday, M...

To send iBeacon, you can type this command.

```
$ sudo hcitool -i hci0 cmd 0x08 0x0008 1E 02 01 1A 1A FF 4C 00 02 15 63
```

It will generate iBeacon data.

```
[pi@raspberrypi:~ $ sudo hcitool -i hci0 cmd 0x08 0x0008 1E 02 01 1A 1A FF 4C 00 ]  
02 15 63 6F 3F 8F 64 91 4B EE 95 F7 D8 CC 64 A8 63 B5 00 00 00 00 C8  
< HCI Command: ogf 0x08, ocf 0x0008, plen 31  
 1E 02 01 1A 1A FF 4C 00 02 15 63 6F 3F 8F 64 91 4B EE 95 F7  
  D8 CC 64 A8 63 B5 00 00 00 00 C8  
> HCI Event: 0x0e plen 4  
  01 08 20 00  
pi@raspberrypi:~ $
```

On Beacon Scan application, it shows iBeacon data.

Beacon Scan

Get Beacons

Clear

Type	UUID / URL	Major	Minor	Power Calibration (dB)	RSSI (dB)	Last...
Beacon	636F3F8F-6491-4BEE-95F7-D8CC64A863B5	0	0	-56	-43	Sat...

## **7. Deploying LAMP Stack**

This chapter explains how to deploy LAMP stack on Raspberry Pi 3 board.

## 7.1 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:

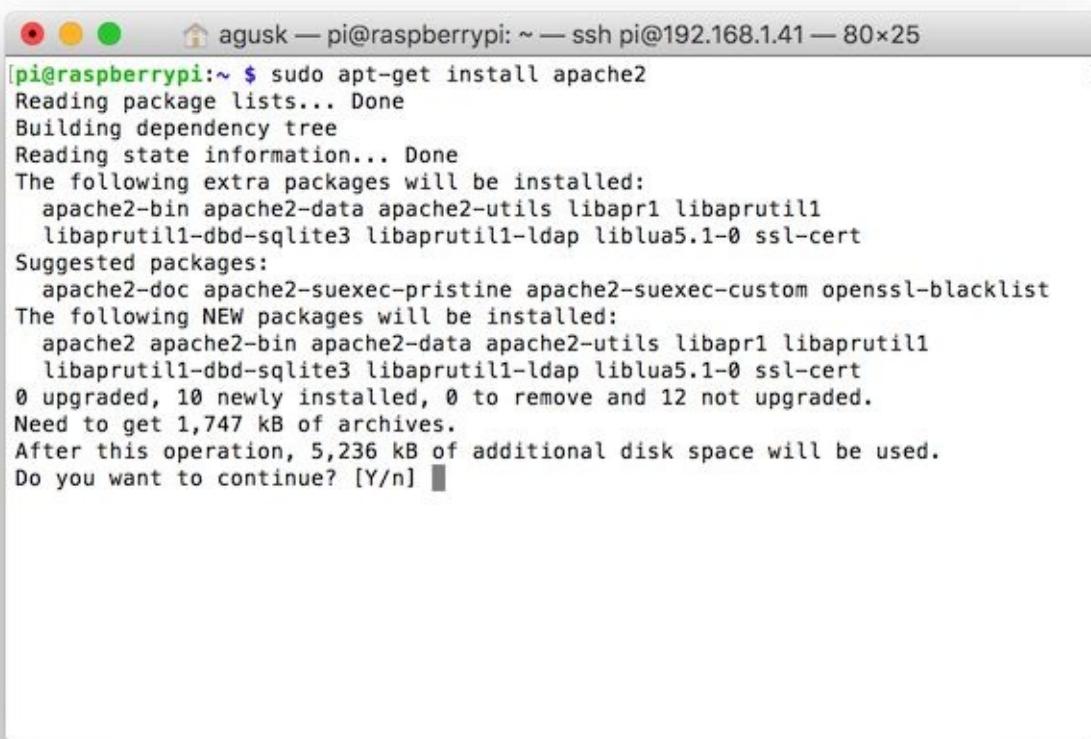
- Web Server, Apache
- Database, MySQL
- PHP
- MySQL Database driver for PHP

We will install these components on next section.

## 7.2 Installing Apache Server

Firstly, we install Apache Server.

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

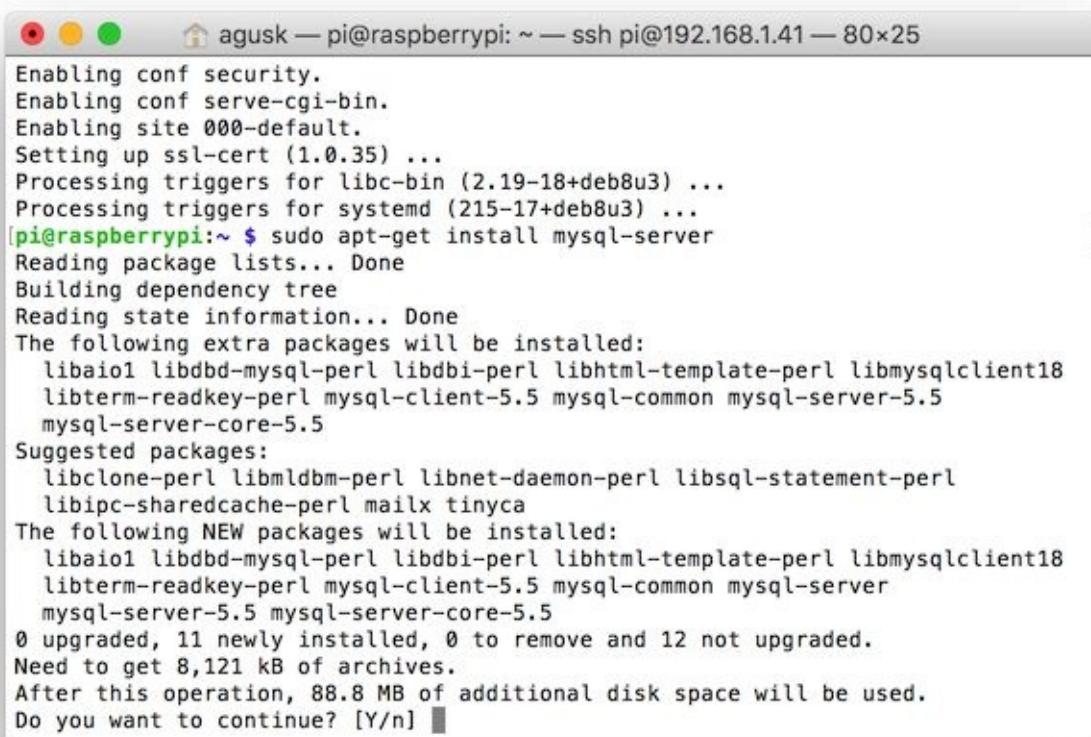


```
[pi@raspberrypi:~ $ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 12 not upgraded.
Need to get 1,747 kB of archives.
After this operation, 5,236 kB of additional disk space will be used.
Do you want to continue? [Y/n] ]
```

## 7.3 Installing MySQL

The second step is to install MySQL. Execute this command

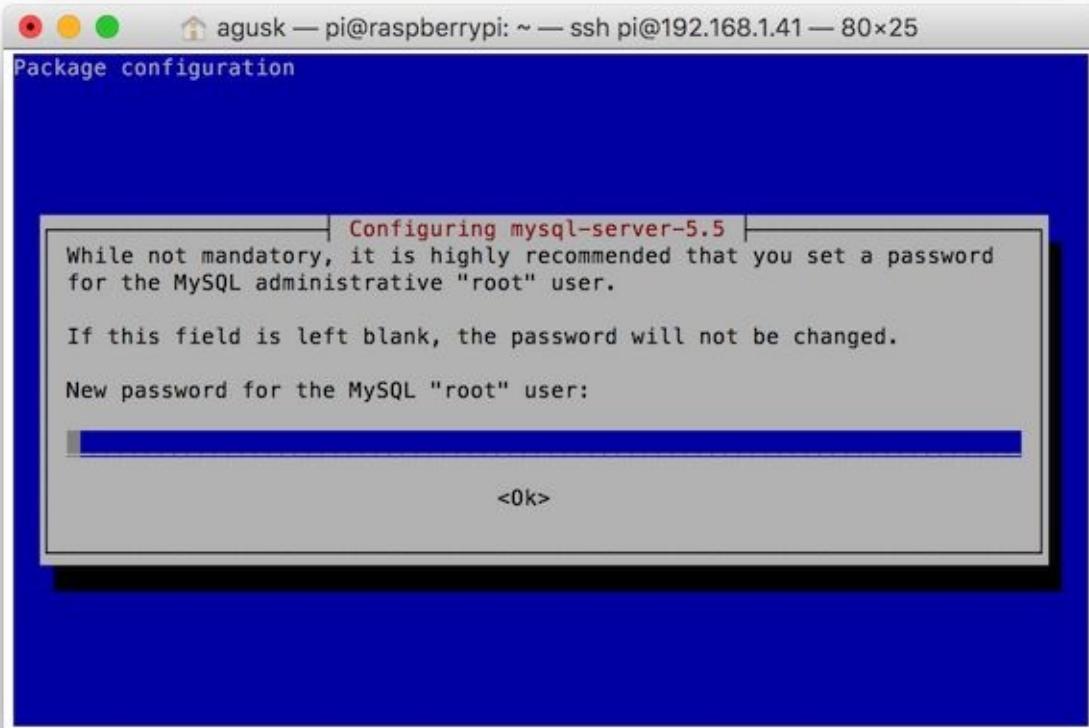
```
$ sudo apt-get install mysql-server
```



A screenshot of a terminal window titled "agusk — pi@raspberrypi: ~ — ssh pi@192.168.1.41 — 80x25". The window shows the output of the command "sudo apt-get install mysql-server". The output includes messages about enabling configuration files, processing triggers for various packages, and listing extra packages to be installed. It also shows suggested packages and lists of NEW and UPGRADING packages. Finally, it asks if the user wants to continue with the operation.

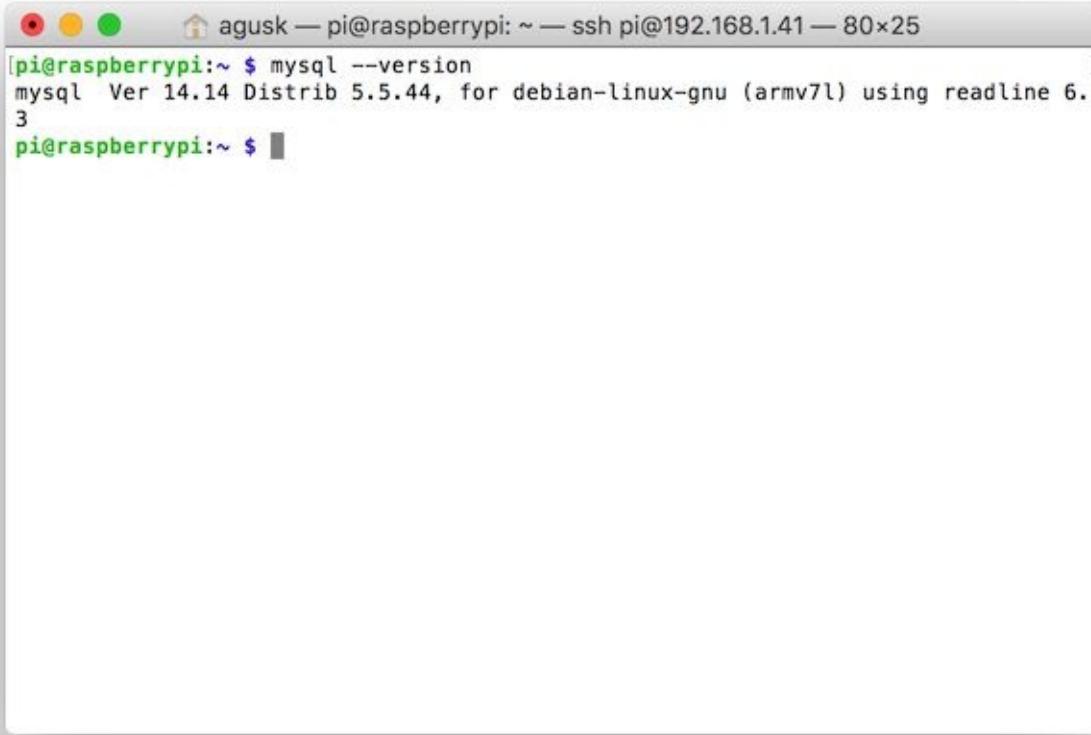
```
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Setting up ssl-cert (1.0.35) ...
Processing triggers for libc-bin (2.19-18+deb8u3) ...
Processing triggers for systemd (215-17+deb8u3) ...
[pi@raspberrypi:~ $ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
  libterm-readkey-perl mysql-client-5.5 mysql-common mysql-server-5.5
  mysql-server-core-5.5
Suggested packages:
  libclone-perl libmldbmm-perl libnet-daemon-perl libsql-statement-perl
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
  libterm-readkey-perl mysql-client-5.5 mysql-common mysql-server
  mysql-server-5.5 mysql-server-core-5.5
0 upgraded, 11 newly installed, 0 to remove and 12 not upgraded.
Need to get 8,121 kB of archives.
After this operation, 88.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] ]
```

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
```



```
agusk — pi@raspberrypi: ~ — ssh pi@192.168.1.41 — 80x25
[pi@raspberrypi:~ $ mysql --version
mysql Ver 14.14 Distrib 5.5.44, for debian-linux-gnu (armv7l) using readline 6.
3
pi@raspberrypi:~ $ ]
```

Now you can connect to MySQL server. Execute this command

```
$ mysql -u root -p
```

Note: you may change MySQL user and password.

Type this command to retrieve databases in MySQL.

```
mysql> show databases;
```

agus — pi@raspberrypi: ~ — ssh pi@192.168.1.41 — 80x25

```
[pi@raspberrypi:~ $ mysql -u root -p
[Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 46
Server version: 5.5.44-0+deb8u1 (Raspbian)

Copyright (c) 2000, 2015, 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.

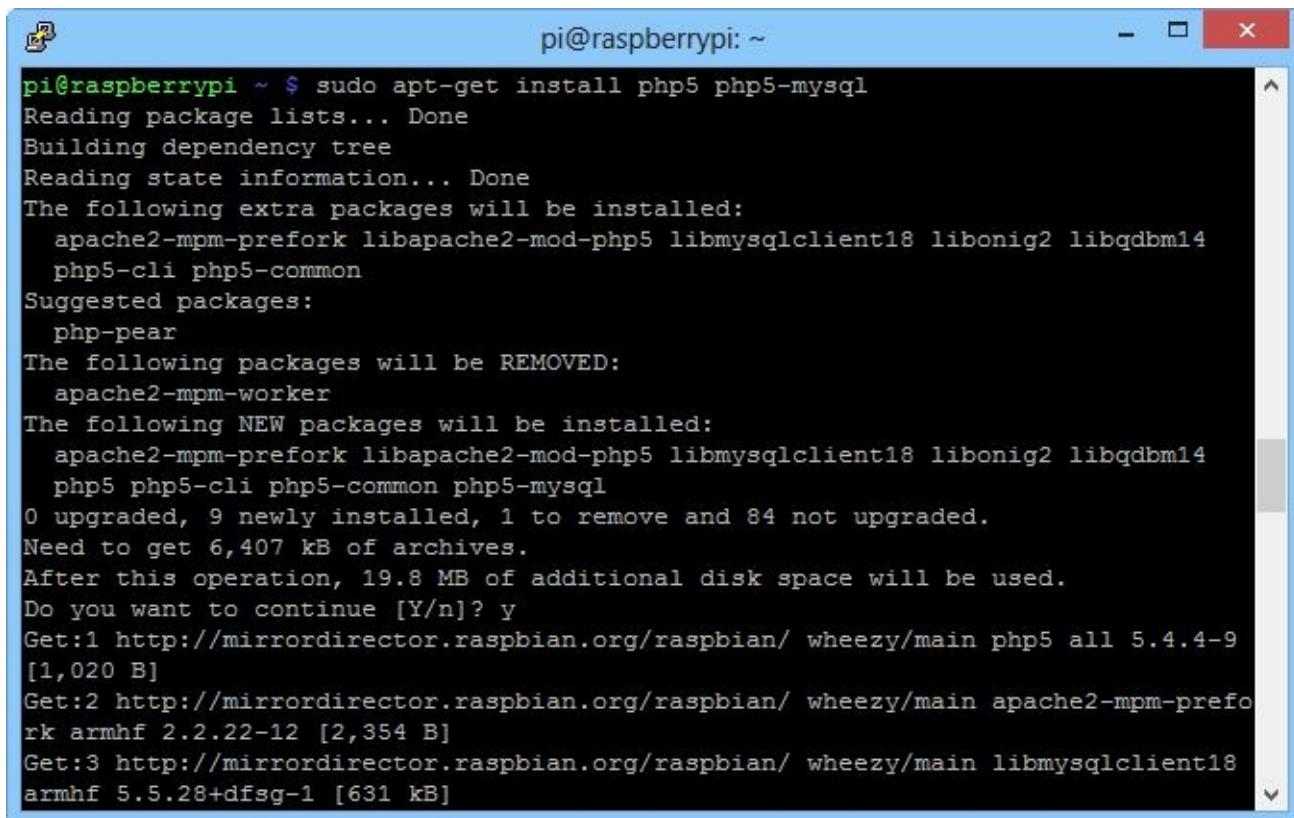
[mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql> ]
```

## 7.4 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
```



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window shows the command \$ sudo apt-get install php5 php5-mysql being run and its output. The output includes package lists, dependency building, state information, extra packages to be installed (apache2-mpm-prefork, libapache2-mod-php5, libmysqlclient18, libonig2, libqdbm14, php5-cli, php5-common), suggested packages (php-pear), packages to be removed (apache2-mpm-worker), new packages to be installed (apache2-mpm-prefork, libapache2-mod-php5, libmysqlclient18, libonig2, libqdbm14, php5, php5-cli, php5-common, php5-mysql), upgrade counts, disk space requirements, and download details for three files from the mirror director.

```
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]
```

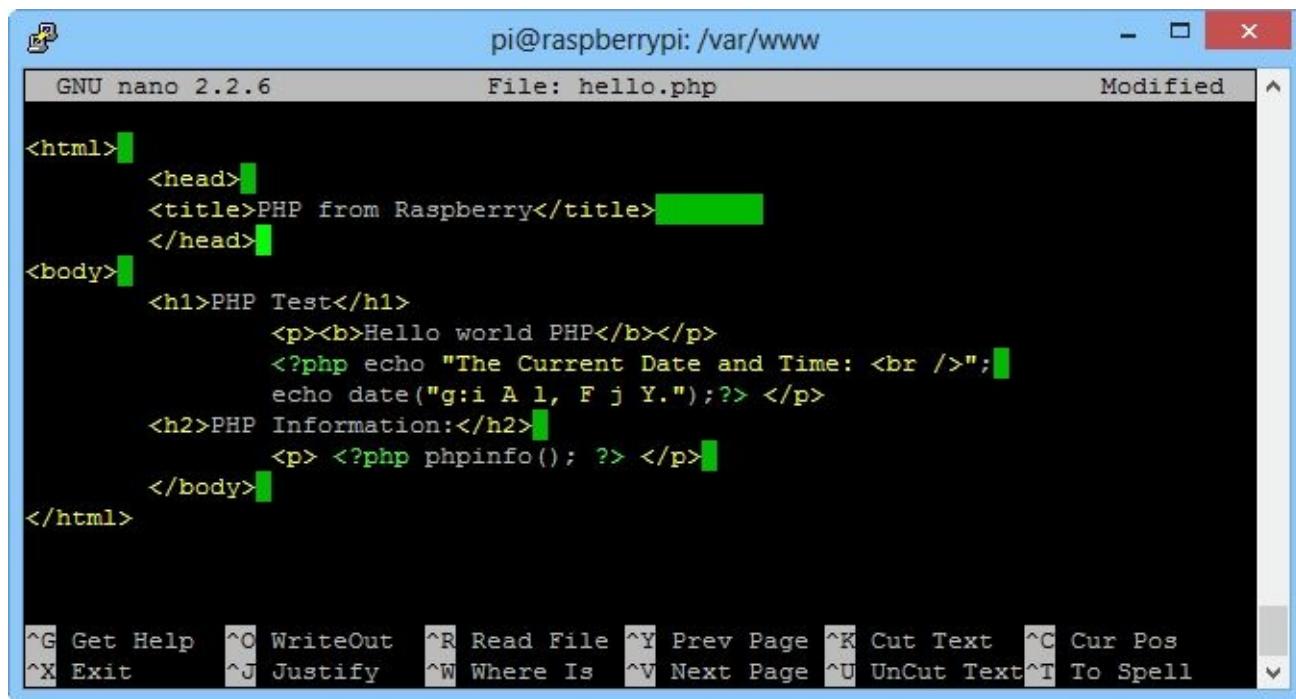
## 7.5 Testing PHP

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 Raspberry</title>
    </head>
<body>
    <h1>PHP Test</h1>
        <p><b>Hello world PHP</b></p>
        <?php echo "The Current Date and Time: <br />" ;
        echo date("g:i A l, F j Y.");?> </p>
    <h2>PHP Information:</h2>
        <p> <?php phpinfo(); ?> </p>
    </body>
</html>
```



The screenshot shows a terminal window titled "pi@raspberrypi: /var/www". The window title bar also displays "File: hello.php". The main area of the terminal shows the content of the "hello.php" file being edited in the nano text editor. The code is identical to the one shown in the previous code block. The bottom of the terminal window shows a menu of keyboard shortcuts for the nano editor.

```
GNU nano 2.2.6          File: hello.php          Modified
<html>
    <head>
        <title>PHP from Raspberry</title>
    </head>
<body>
    <h1>PHP Test</h1>
        <p><b>Hello world PHP</b></p>
        <?php echo "The Current Date and Time: <br />" ;
        echo date("g:i A l, F j Y.");?> </p>
    <h2>PHP Information:</h2>
        <p> <?php phpinfo(); ?> </p>
    </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 **hello.php** file located.

PHP from Raspberry

192.168.1.61/hello.php

## PHP Test

### Hello world PHP

The Current Date and Time:  
9:42 PM Sunday, December 2 2012.

### PHP Information:

**PHP Version 5.4.4-9**

PHP Version 5.4.4-9	
System	Linux raspberrypi 3.2.27+ #250 PREEMPT Thu Oct 18 19:03:02 BST 2012 armv6l
Build Date	Nov 9 2012 03:50:16
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API20100525,NTS

## 7.6 Testing PHP and MySQL

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

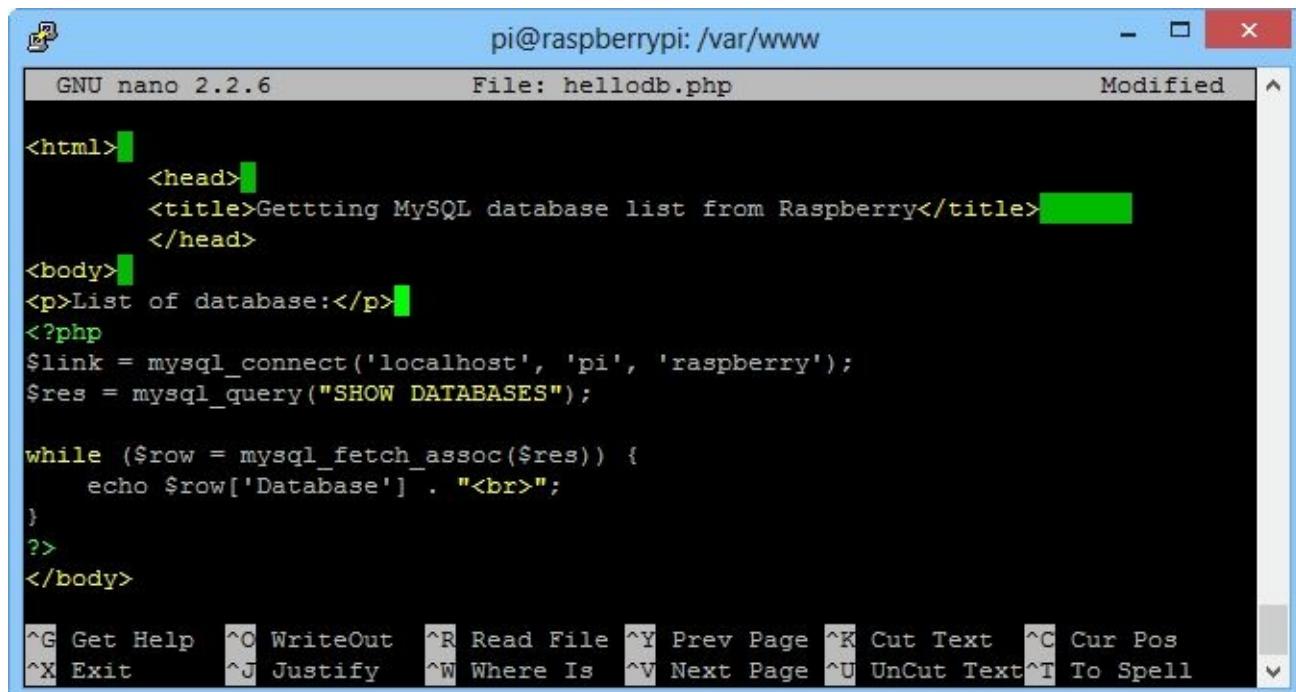
```
$ sudo nano /var/www/hellodb.php
```

Write all code below.

```
<html>
    <head>
        <title>Getting 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>
```

Note: change username and password for MySQL.



The screenshot shows a terminal window titled "pi@raspberrypi: /var/www". The window contains the code for the hellodb.php file, which is being edited with the nano text editor version 2.2.6. The code is identical to the one provided in the previous text block, including the PHP code and the HTML structure. The terminal window also displays the standard nano key bindings at the bottom.

```
pi@raspberrypi: /var/www
GNU nano 2.2.6          File: hellodb.php          Modified
<html>
    <head>
        <title>Getting 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.



## **8. Accessing GPIO**

This chapter explains how to work with GPIO on Raspberry Pi 3.

## 8.1 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 3 board, you can see it in Figure below.

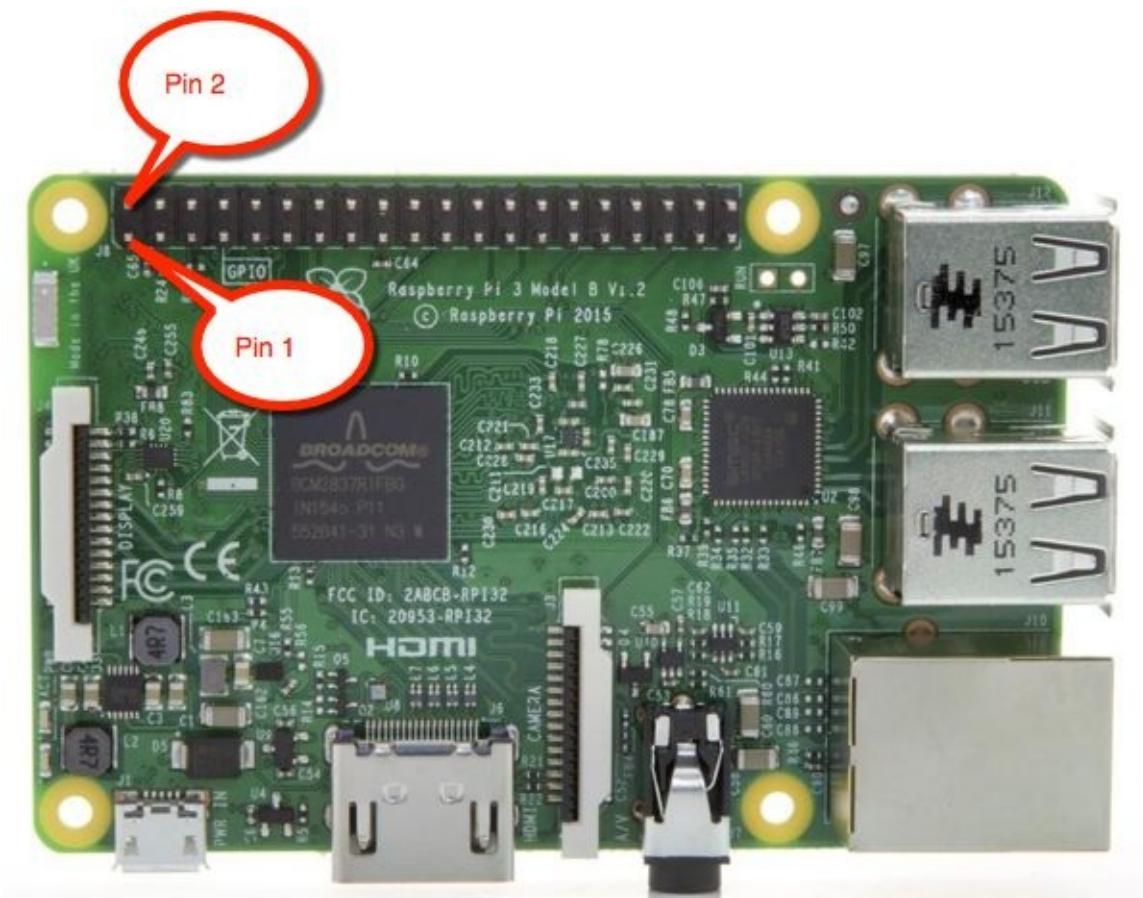
Raspberry Pi 3 GPIO Header			
Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)	DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	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 (I <sup>2</sup> C ID EEPROM)	(I <sup>2</sup> C 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. 2  
29/02/2016

[www.element14.com/RaspberryPi](http://www.element14.com/RaspberryPi)

(source: <https://www.element14.com/community/docs/DOC-73950/l/raspberry-pi-3-model-b-gpio-40-pin-block-pinout> )

See the pin 1 and pin 2 on physical board on the following Figure.



## 8.2 Accessing GPIO

In this section, we will focus on Raspberry Pi GPIO programming using Python. Python and RPi.GPIO library, <http://pypi.python.org/pypi/RPi.GPIO>, have installed for you so you can use them directly to access Raspberry Pi GPIO.

Now you read all GPIO configuration using gpio command. Type this command on Terminal.

```
$ gpio readall
```

The terminal window shows the output of the `gpio readall` command. The title bar indicates the session is for user agusk on a Raspberry Pi with IP 192.168.1.60. The command prompt is [pi@raspberrypi:~ \$]. The output is a table showing the pin mapping for the Raspberry Pi 3 Model B. The columns are labeled: BCM, wPi, Name, Mode, V, Physical, V, Mode, Name, wPi, and BCM. The table lists 40 pins, each with its name, mode (e.g., IN, OUT), value (e.g., 0, 1), and corresponding BCM and wPi numbers. For example, pin 3 (BCM) is 3.3v, pin 2 (wPi) is SDA.1, and pin 14 (BCM) is 5v.

Pi 3											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	IN	1	3	4		5V			
3	9	SCL.1	IN	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	1	ALT5	TxD	15	
		0v			9	10	1	ALT5	RxD	16	
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	
27	2	GPIO. 2	IN	0	13	14			0v		
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	
		3.3v			17	18	0	IN	GPIO. 5	5	
10	12	MOSI	IN	0	19	20			0v		
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	
		0v			25	26	1	IN	CE1	11	
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	
5	21	GPIO.21	IN	1	29	30			0v		
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	
13	23	GPIO.23	IN	0	33	34			0v		
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	
		0v			39	40	0	IN	GPIO.29	29	
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	

pi@raspberrypi:~ \$

You can see pin mapping for Raspberry Pi 3 GPIO.

You also can install RPi.GPIO library manually. How to install?

For illustration, we will install RPi.GPIO 0.6.2. 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. For instance, I install RPi.GPIO 0.6.2.

```
$ wget $ https://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.6  
$ tar -xvzf RPi.GPIO-0.6.2.tar.gz  
$ cd RPi.GPIO-0.6.2/  
$ sudo python setup.py install
```

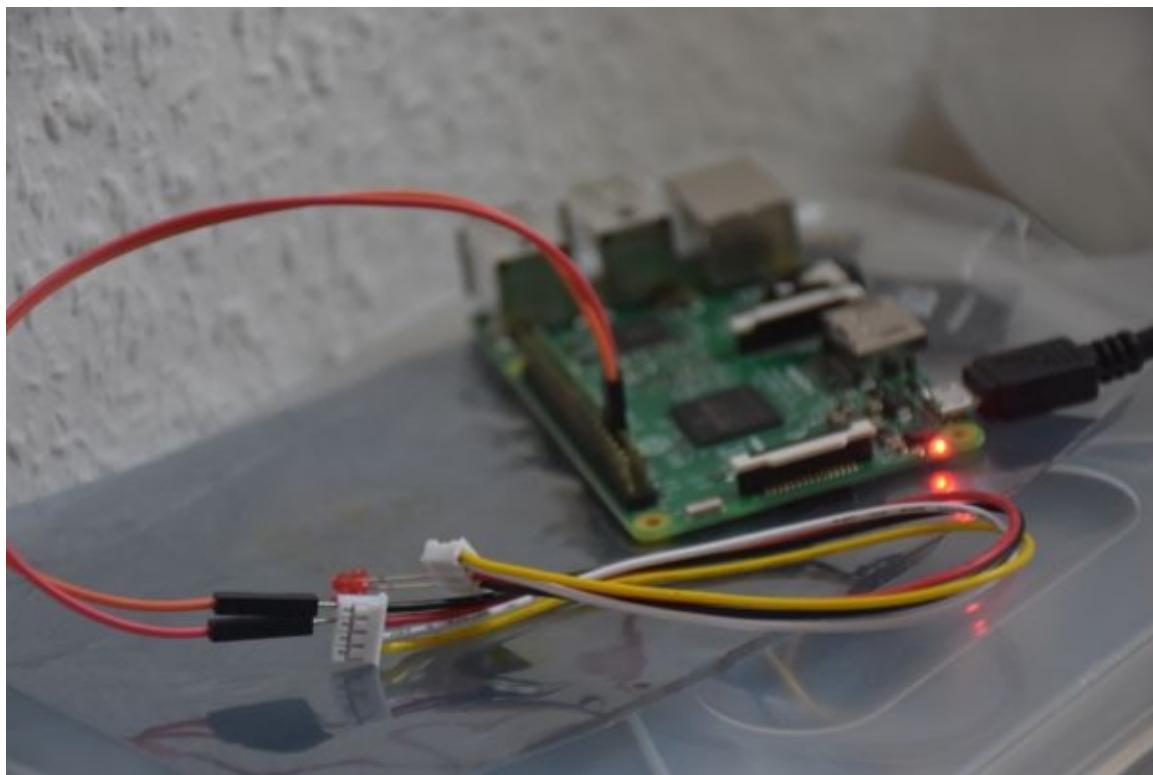
Make sure your Raspberry Pi already connected to Internet network.

## 8.3 Demo

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

The LED is connected to GPIO pin 11 (GPIO17). LED ground pin is connected to GPIO GND.

The following is my wiring.



Now we create Python application to write data on GPIO. We can use `GPIO.output()` to write data, `GPIO.HIGH` and `GPIO.LOW`.

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

```
import RPi.GPIO as GPIO
import time

led_pin = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)

try:
    while 1:
        print("turn on led")
        GPIO.output(led_pin, GPIO.HIGH)
        time.sleep(2)
        print("turn off led")
        GPIO.output(led_pin, GPIO.LOW)
```

```

time.sleep(2)

except KeyboardInterrupt:
    GPIO.output(led_pin, GPIO.LOW)
    GPIO.cleanup()

print("done")

```

The screenshot shows a terminal window titled "GNU nano 2.2.6" with the file "leddemo.py" open. The code is identical to the one shown above. At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts.

```

import RPi.GPIO as GPIO
import time

led_pin = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)

try:
    while 1:
        print("turn on led")
        GPIO.output(led_pin, GPIO.HIGH)
        time.sleep(2)
        print("turn off led")
        GPIO.output(led_pin, GPIO.LOW)
        time.sleep(2)

except KeyboardInterrupt:
    GPIO.output(led_pin, GPIO.LOW)
    GPIO.cleanup()

^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 ^I To Spell

```

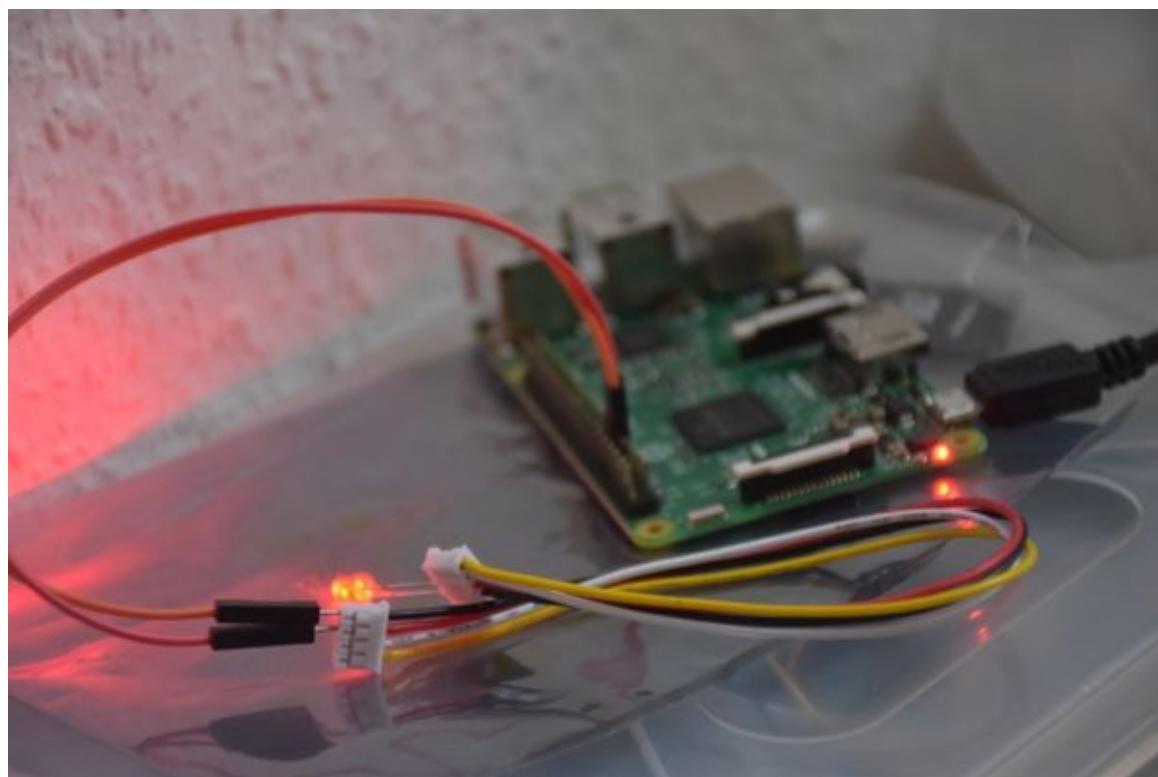
Save this code.

You can run the application using the following command.

```
$ sudo python leddemo.py
```

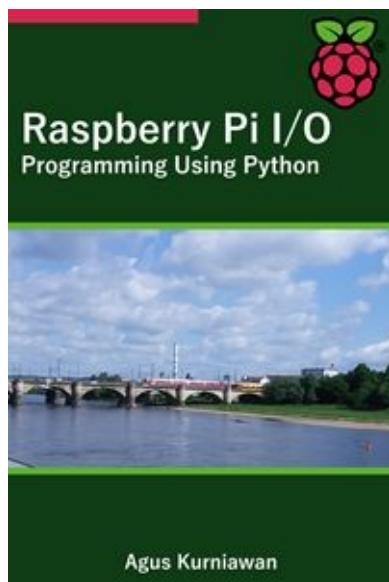
If success, you can see the output as below.

The LED also is blinking.



## 8.4 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>.



## **9. Raspberry Pi 3 Serial Debugging**

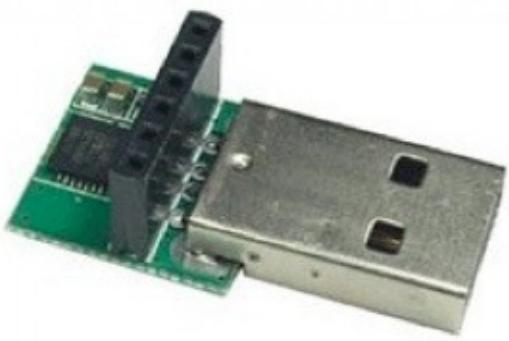
This chapter explains how to access Raspberry Pi 3 via Serial Port.

## 9.1 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> .



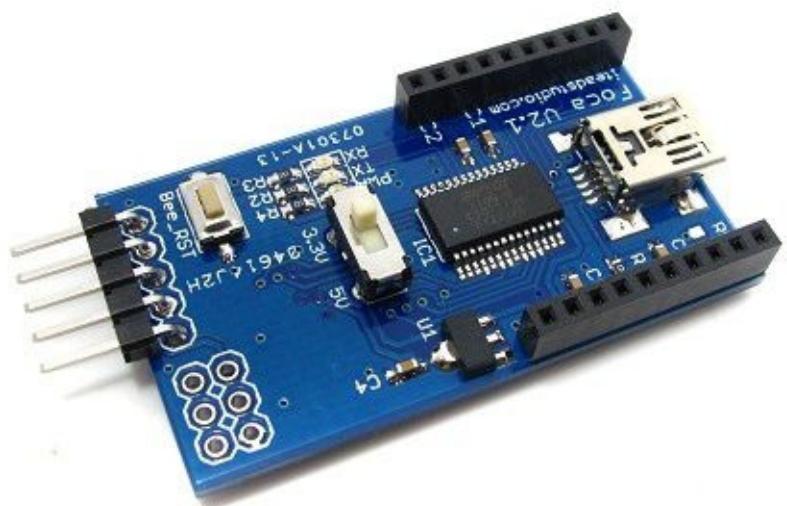
Another device, you can buy USB to TTL on Cooking-Hacks, <http://www.cooking-hacks.com/index.php/usb-to-ttl-converter-cp210.html> .



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](#)

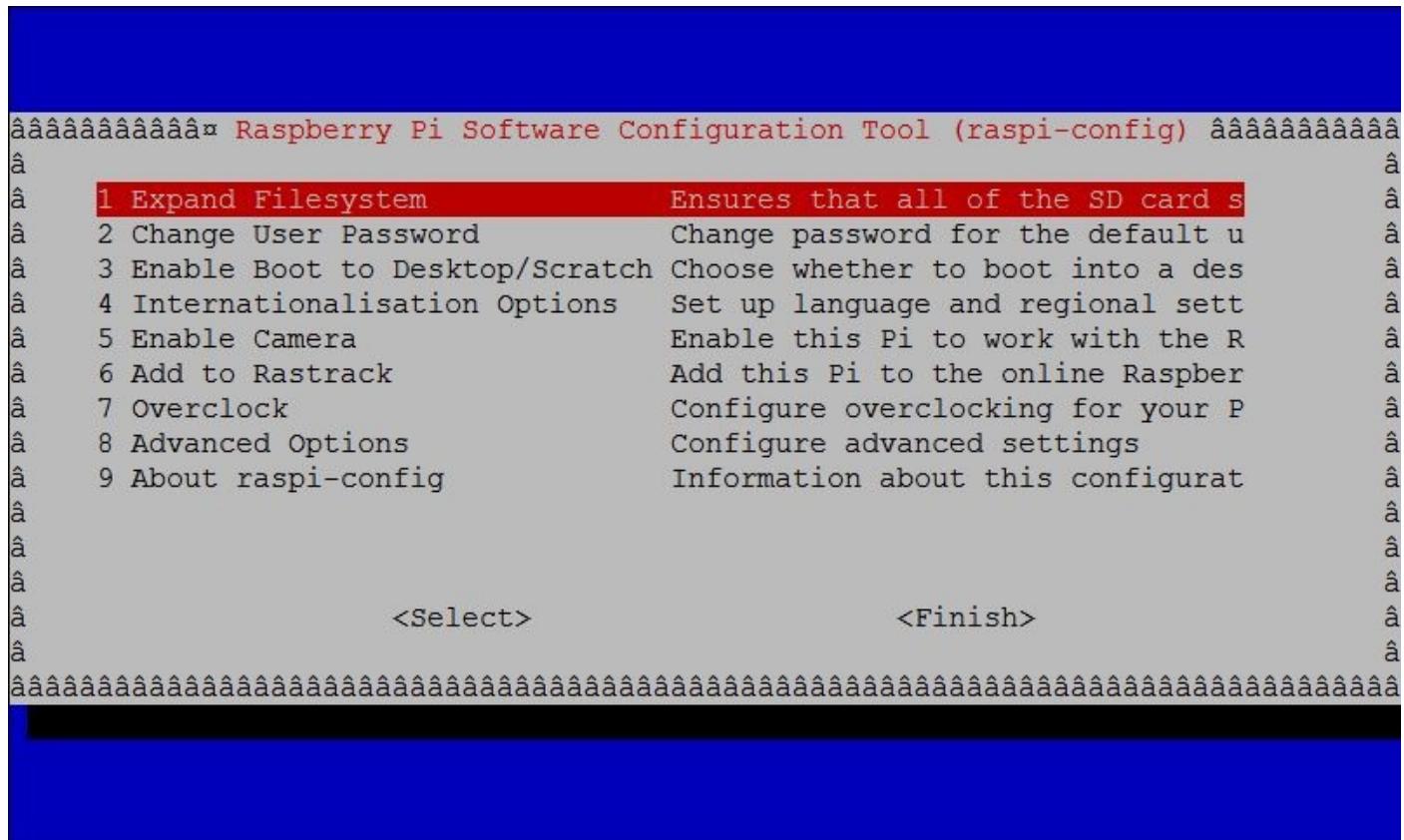


## 9.2 Enabling Serial Debugging

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.



Then, select **A8 Serial** menu.

```
Raspberry Pi Software Configuration Tool (raspi-config) 

A1 Overscan           You may need to configure overscan
A2 Hostname            Set the visible name for this Pi
A3 Memory Split        Change the amount of memory made
A4 SSH                 Enable/Disable remote command line
A5 Device Tree         Enable/Disable the use of Device Tree
A6 SPI                 Enable/Disable automatic loading
A7 I2C                Enable/Disable automatic loading
A8 Serial              Enable/Disable shell and kernel module loading
A9 Audio               Force audio out through HDMI or 3.5mm jack
A0 Update              Update this tool to the latest version

<Select>          <Back>
```

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

```
Would you like a login shell to be accessible over serial?
```

**<Yes>**      **<No>**

If success, you will get a success confirmation.



Close this config dialog.

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

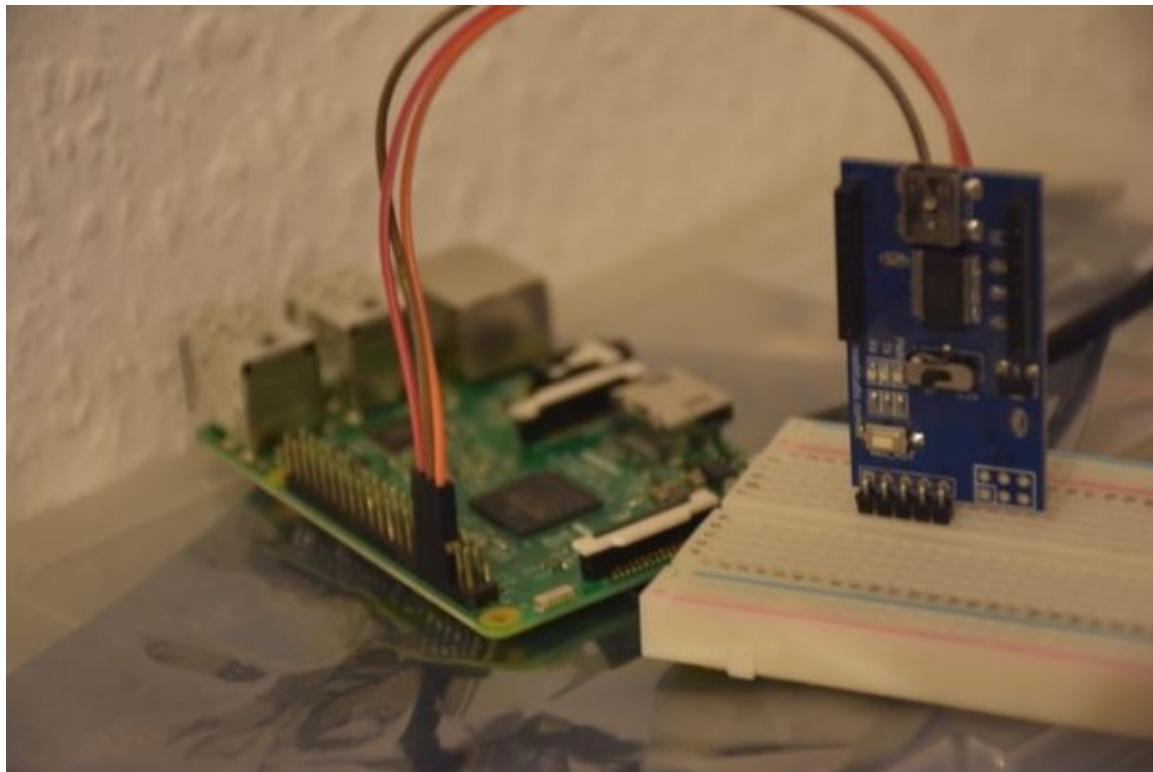
```
$ sudo reboot
```

### 9.3 Wiring

How to implement?

It's easy. You can just connect Tx from USB TTL to Raspberry Pi UART0\_TXD and USB TTL RX to Raspberry Pi UART0\_RXD. Some USB TTL must change them. It means USB TTL TX should be connected to Raspberry Pi UART0\_RXD and USB TTL RX to Raspberry Pi UART0\_TXD. (Optional) You can connect GND from USB TTL to GND of Raspberry Pi board.

Here is a sample of connected hardware.

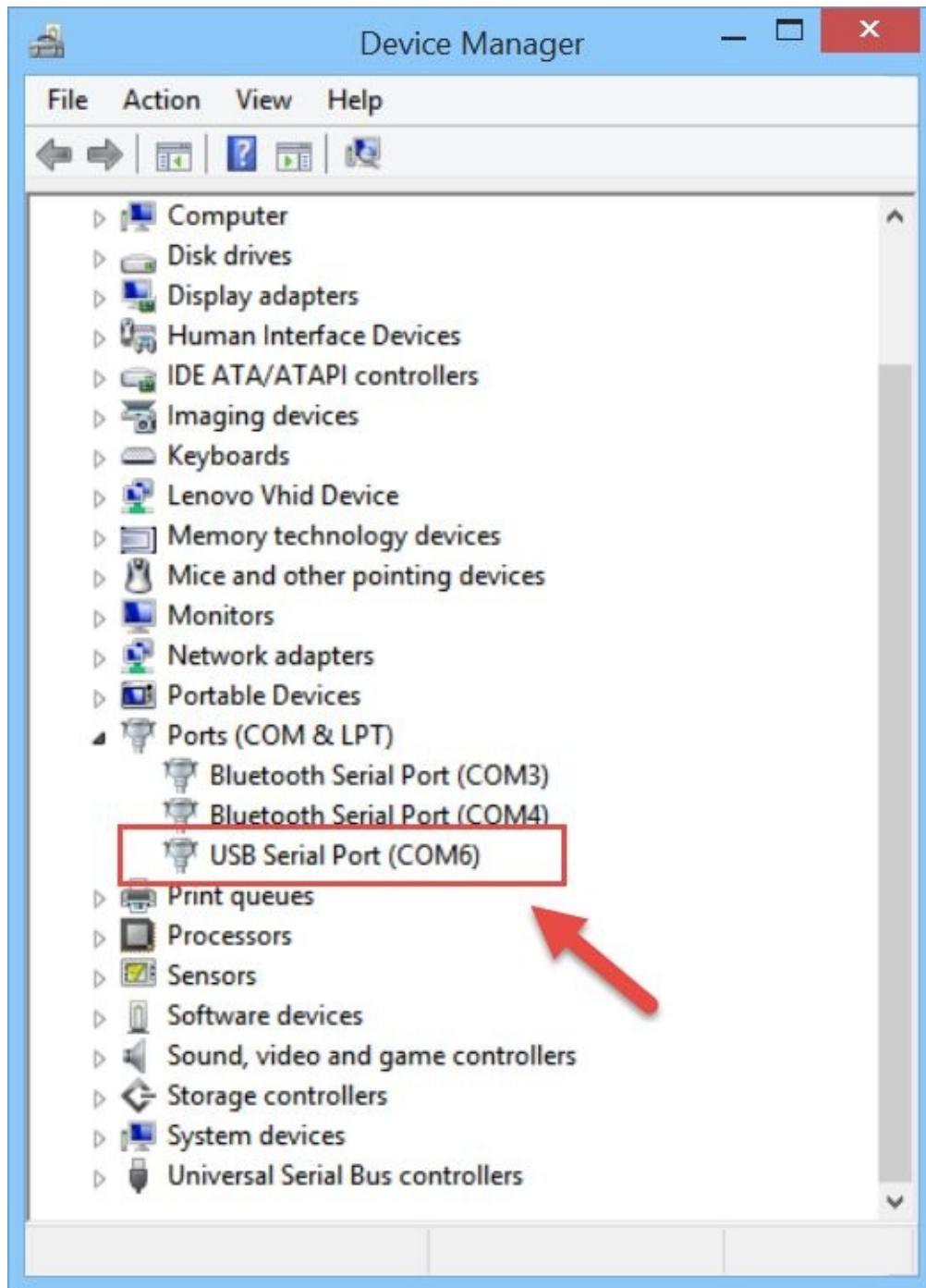


Now your USB cable of USB TTL device is be connected to your computer. You can use any serial application to execute.

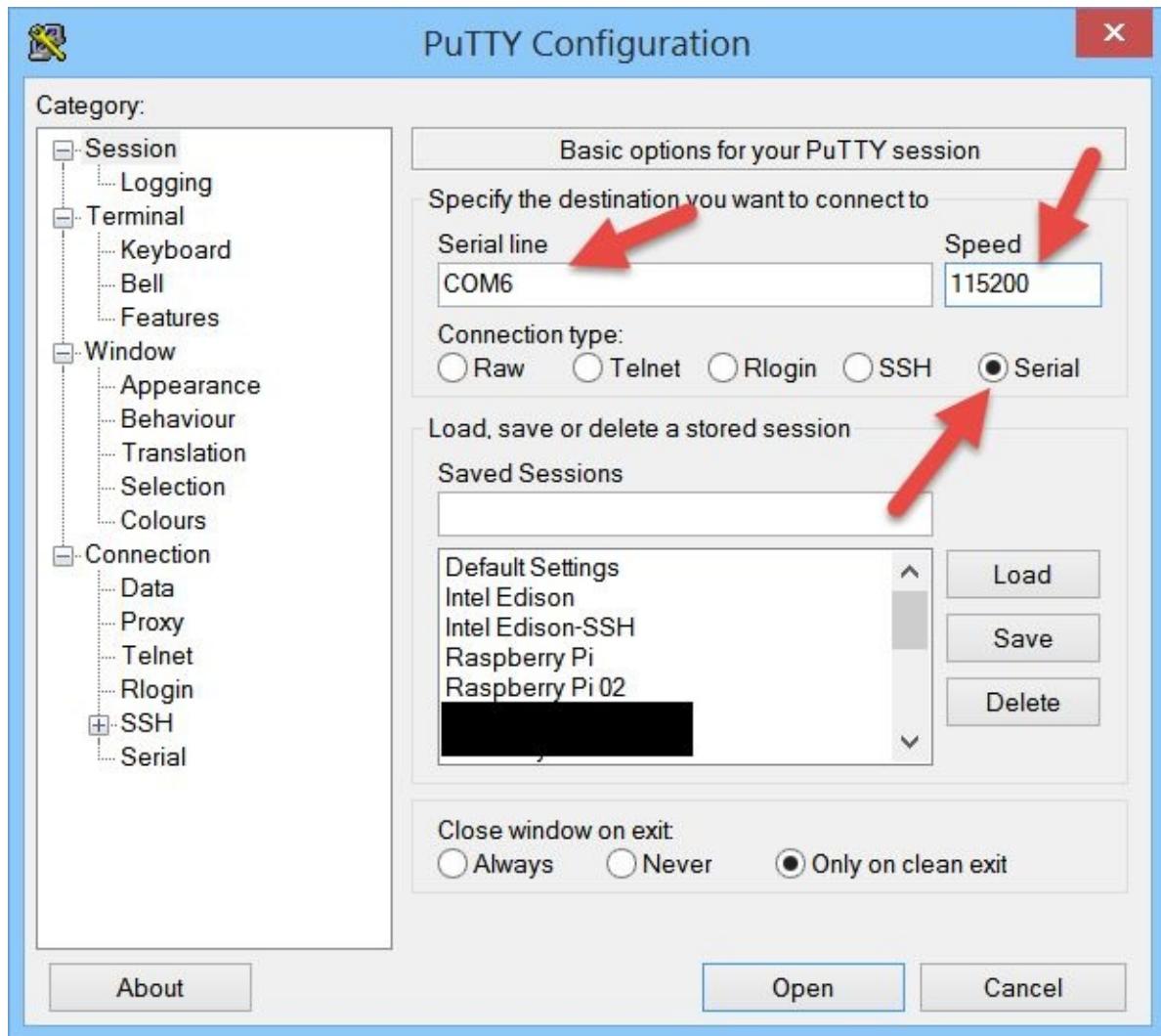
In this book, I used PuTTY,

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

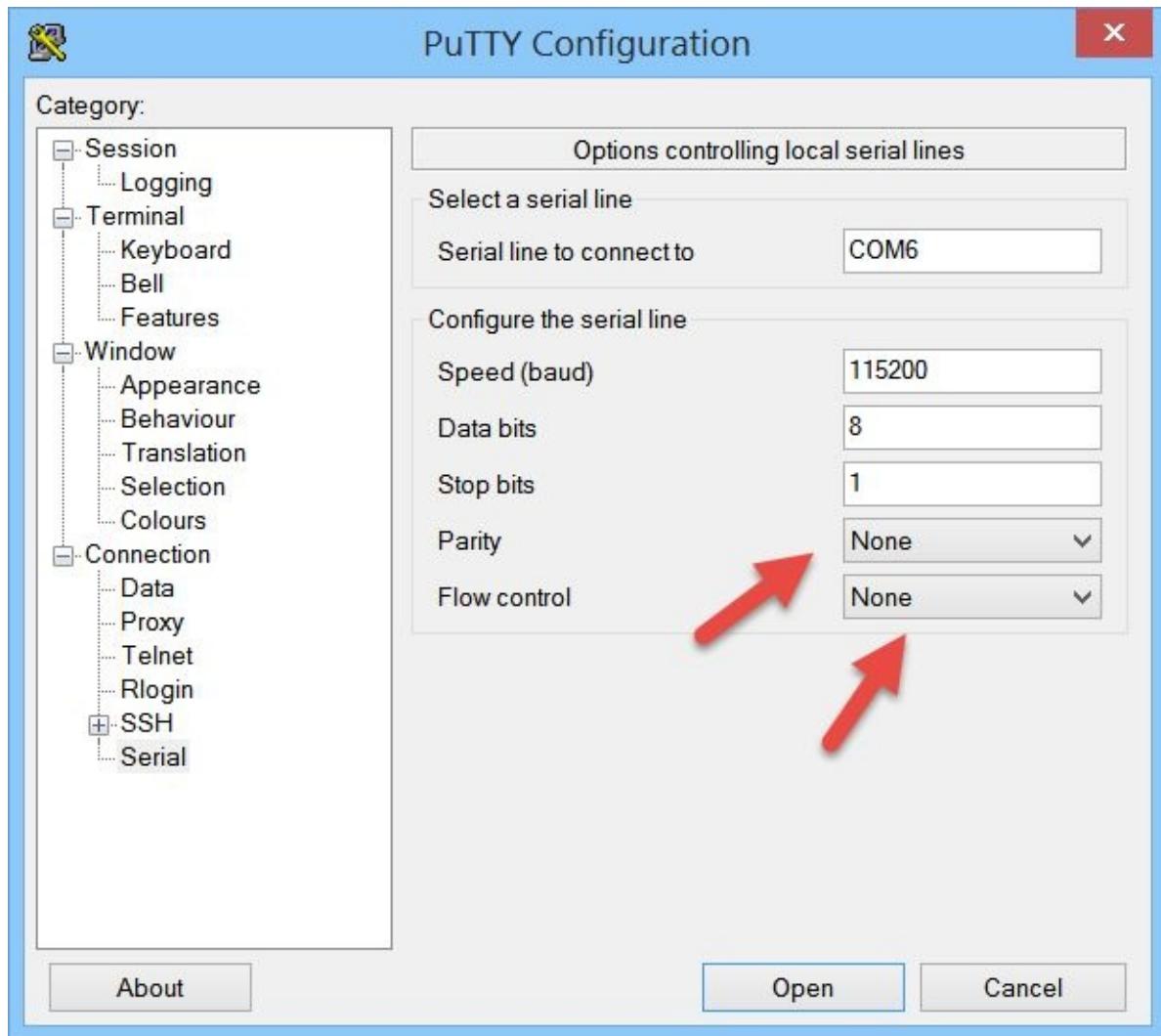
Run PuTTY and choose Serial for connection type. Fill Serial line name, for instance, my Windows detected it on COM6 as below.



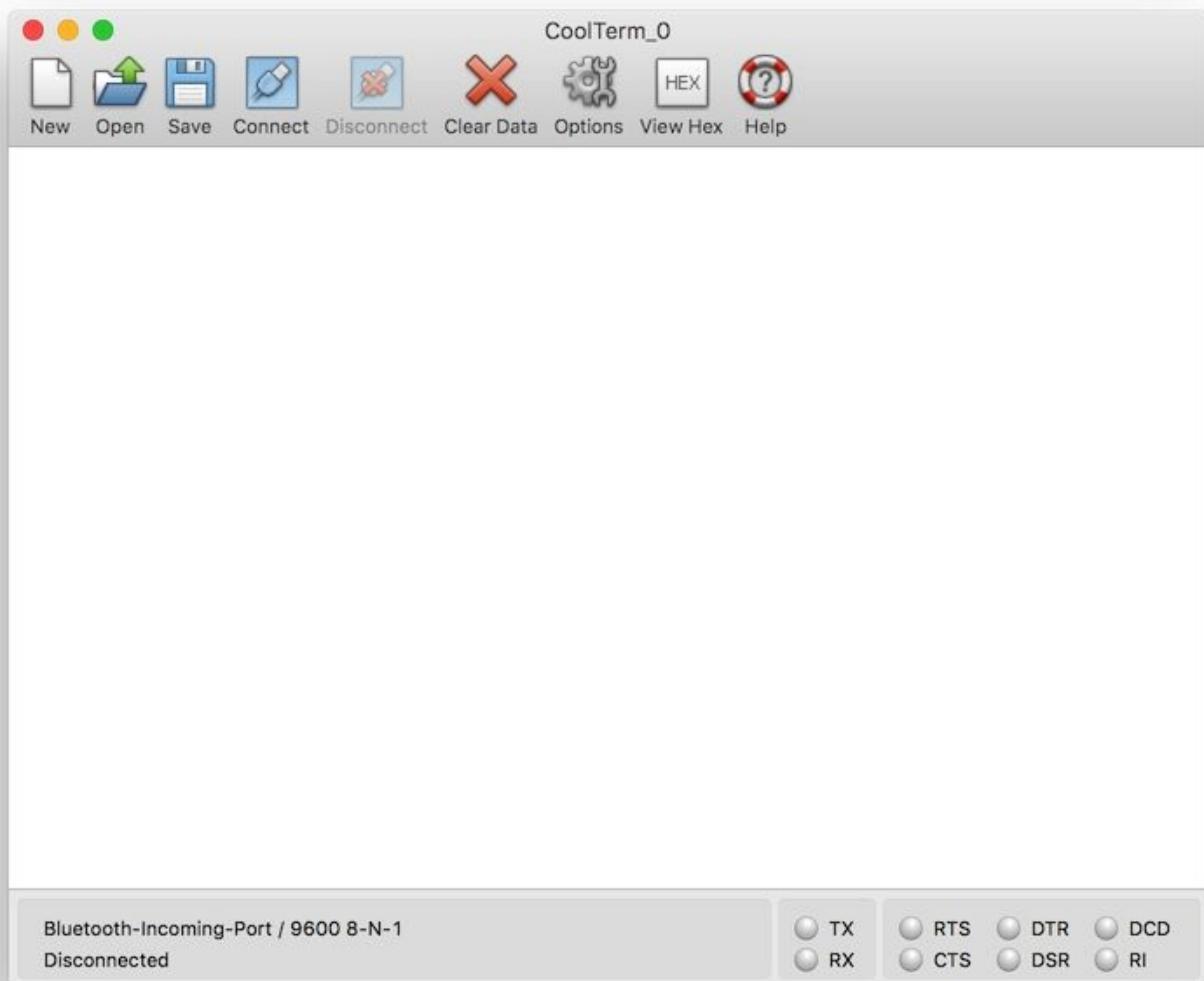
Set 115200 for speed serial.



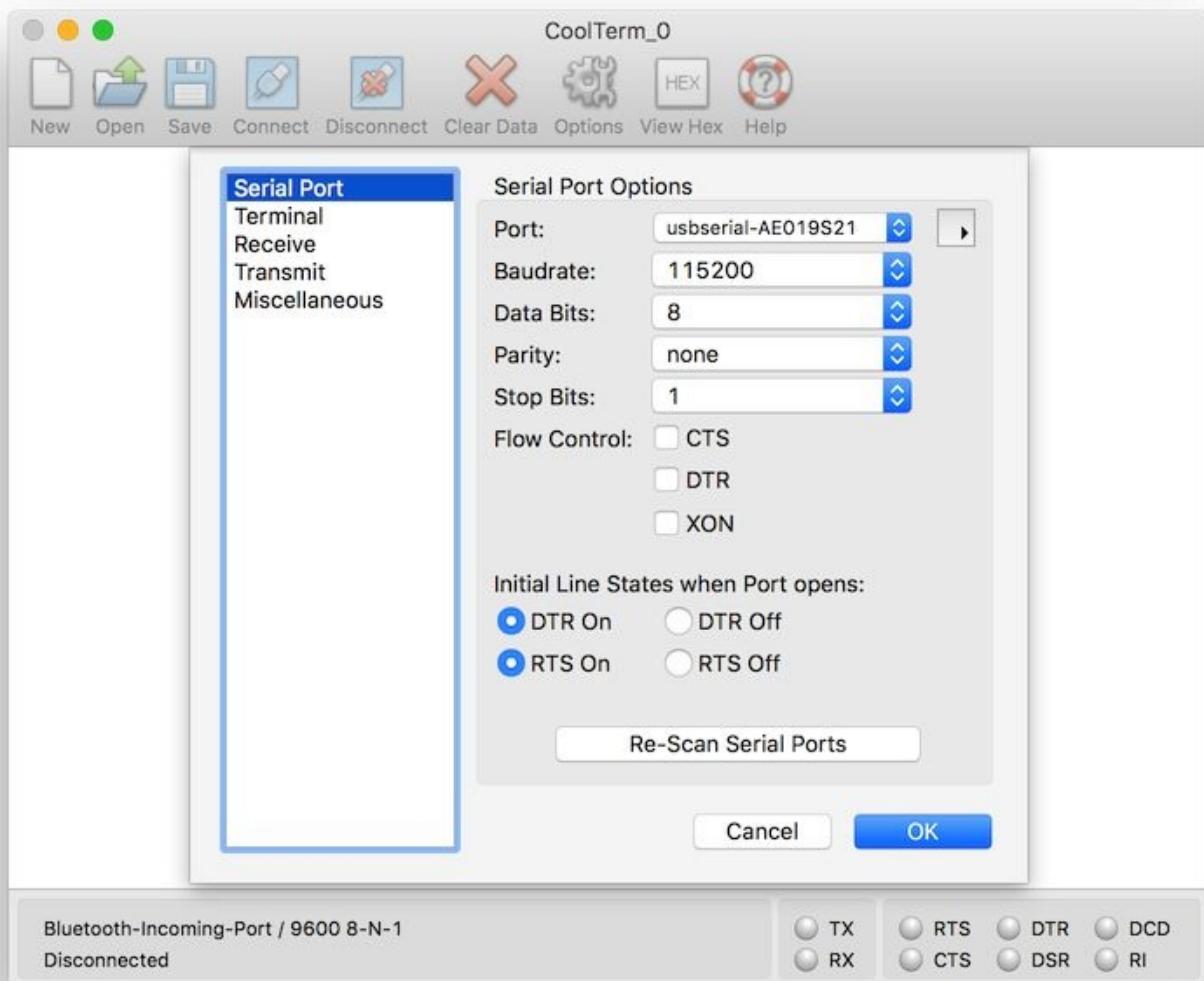
Click **Serial** on side menu and choose **None** for Parity and Flow control.



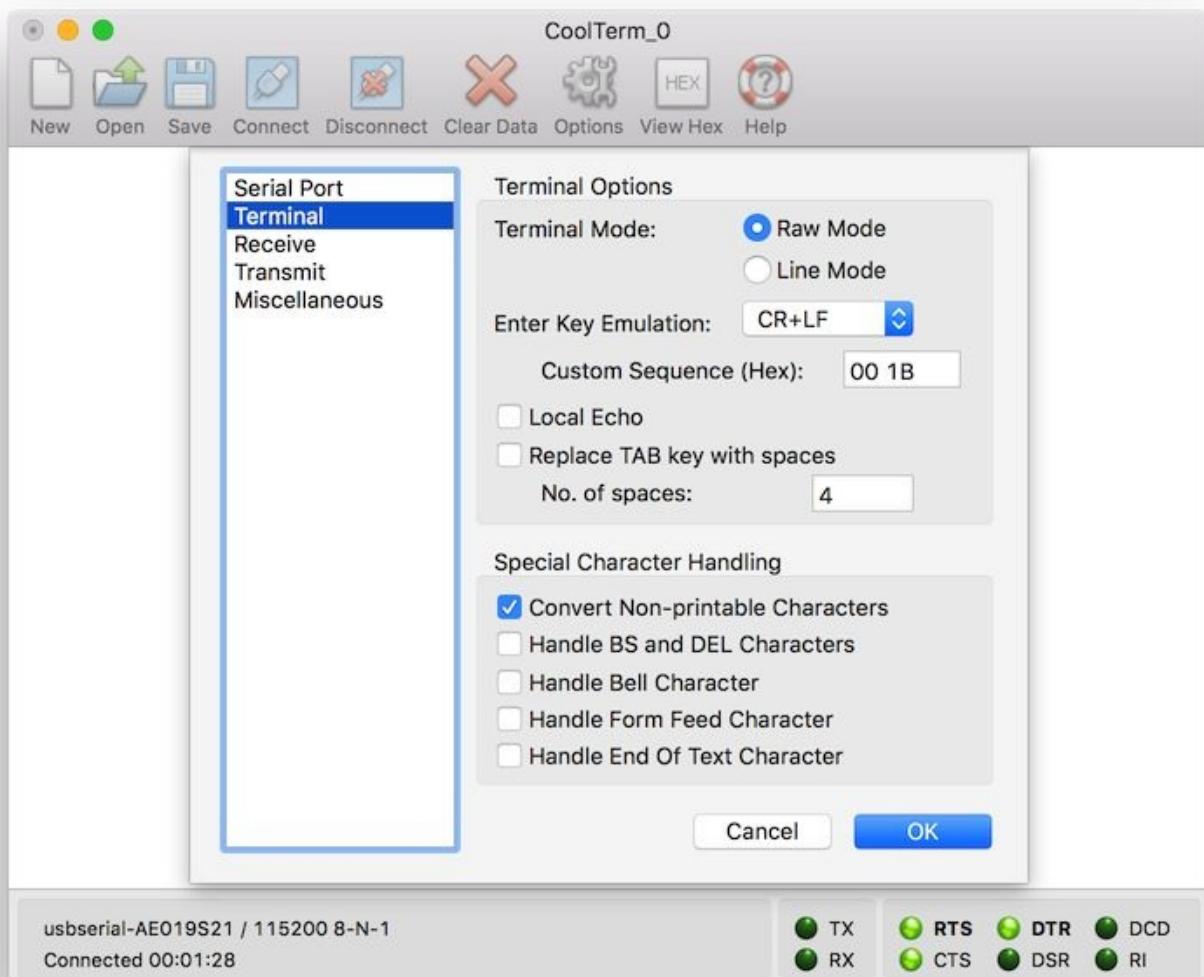
You also can use CoolTerm to view serial data. You can download and install it on <http://freeware.the-meiers.org>.



To use it, you can open Options menu and select Port and Baudrate on 115200.



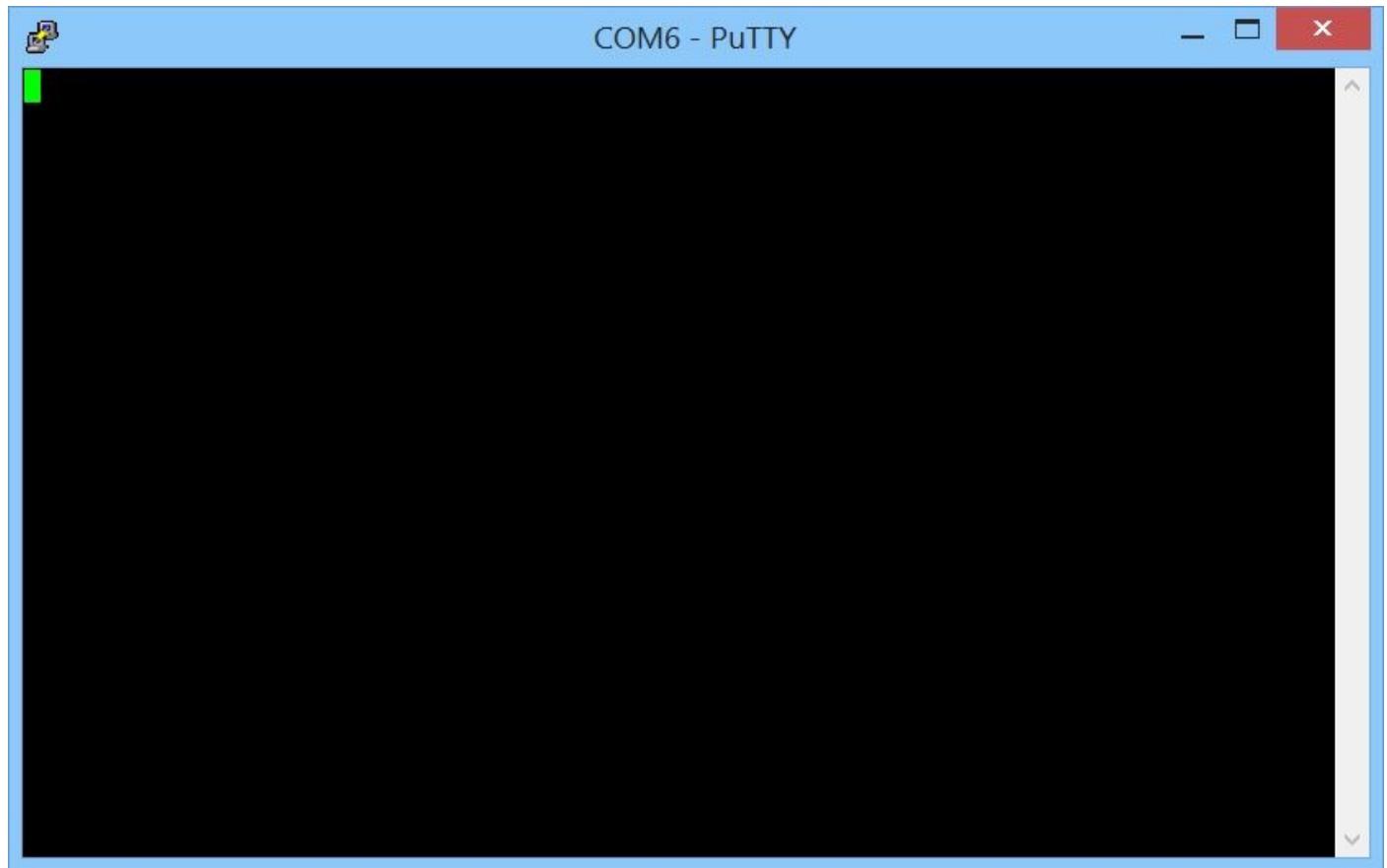
On Terminal tab, you don't configure anything.



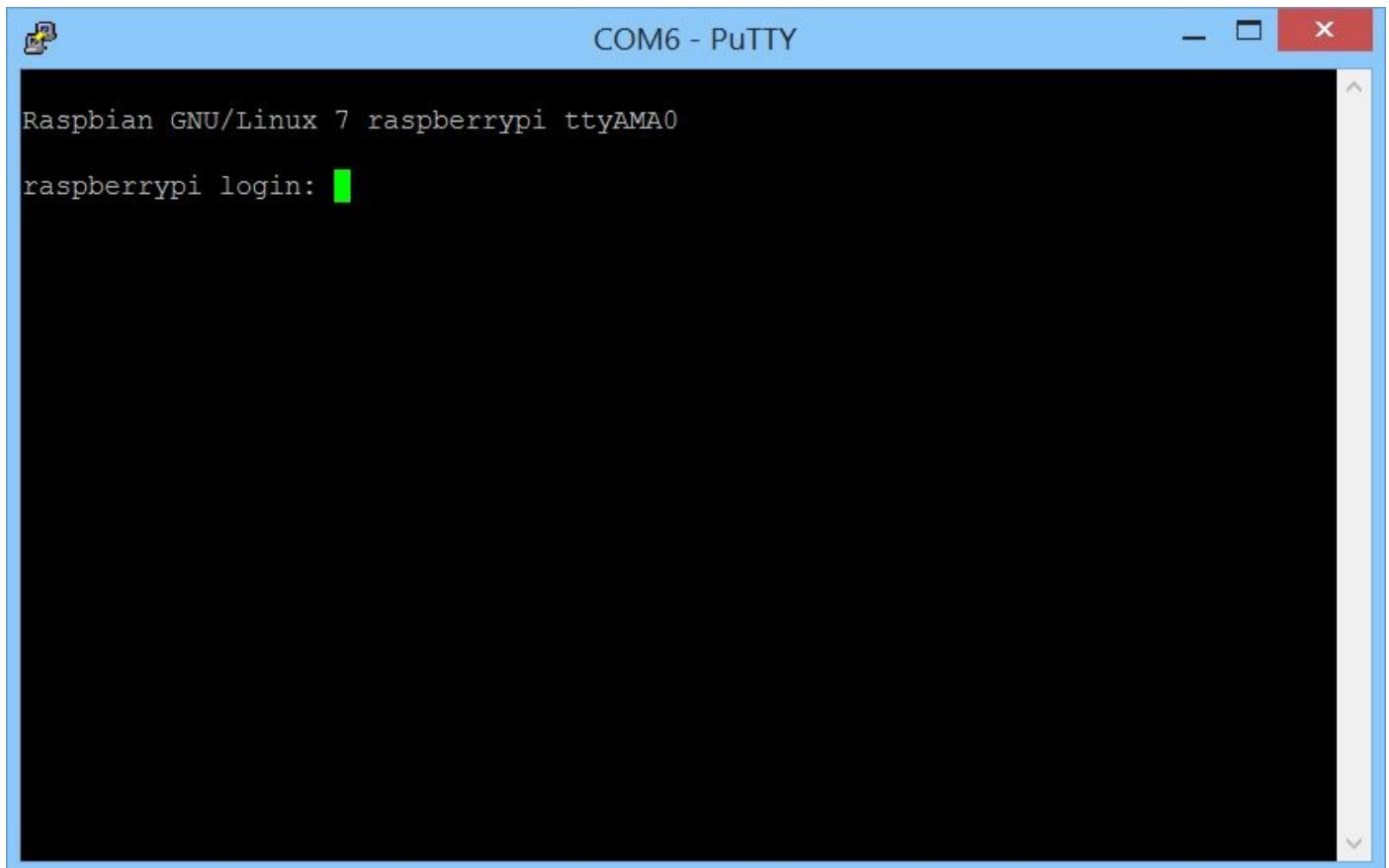
Click OK button to save the configuration.

## 9.4 Testing

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



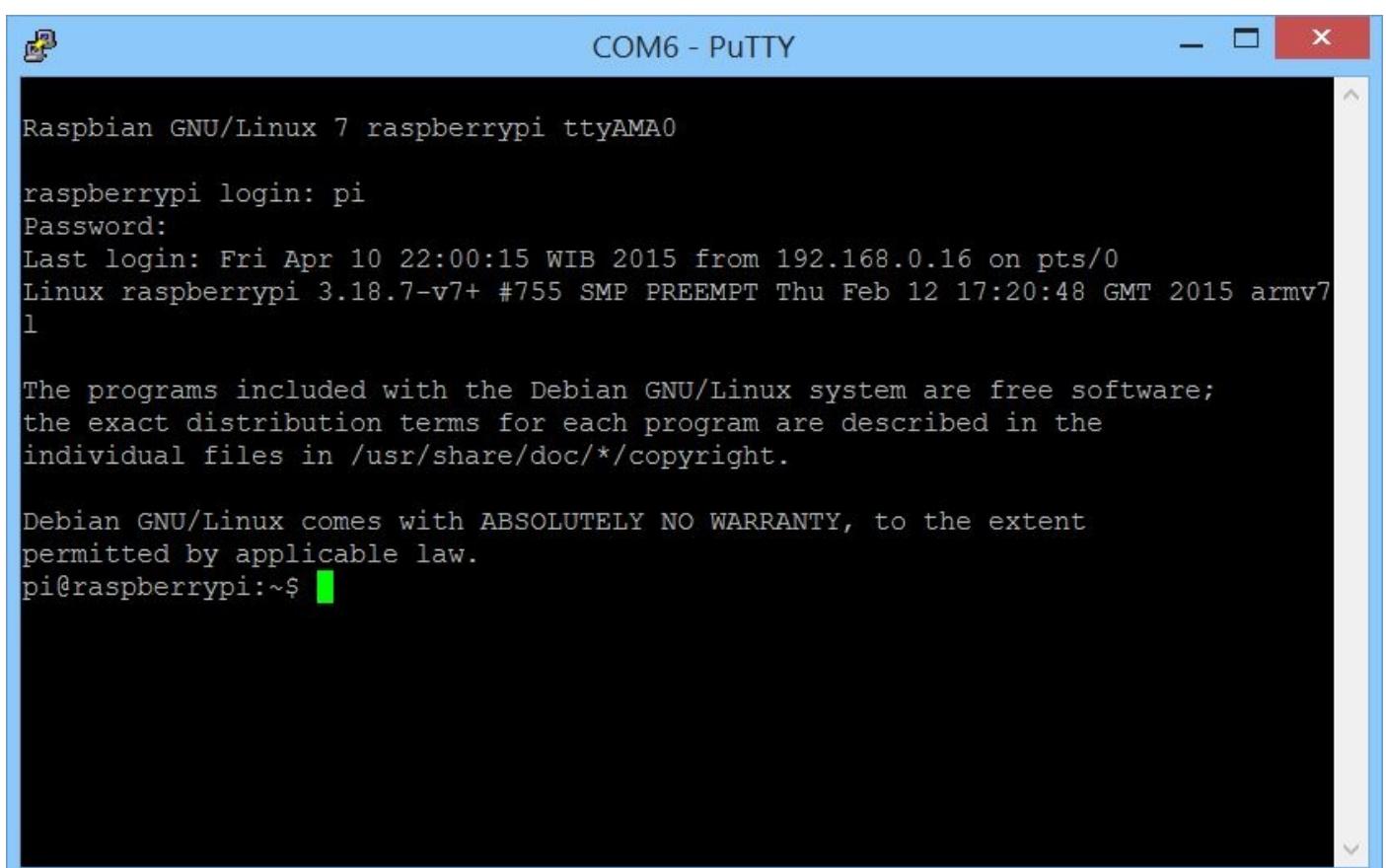
If success, you will get the authentication form.



Raspbian GNU/Linux 7 raspberrypi ttyAMA0  
raspberrypi login: █

Then, try to logon to Raspberry Pi.

Here is a sample of serial debugging output.

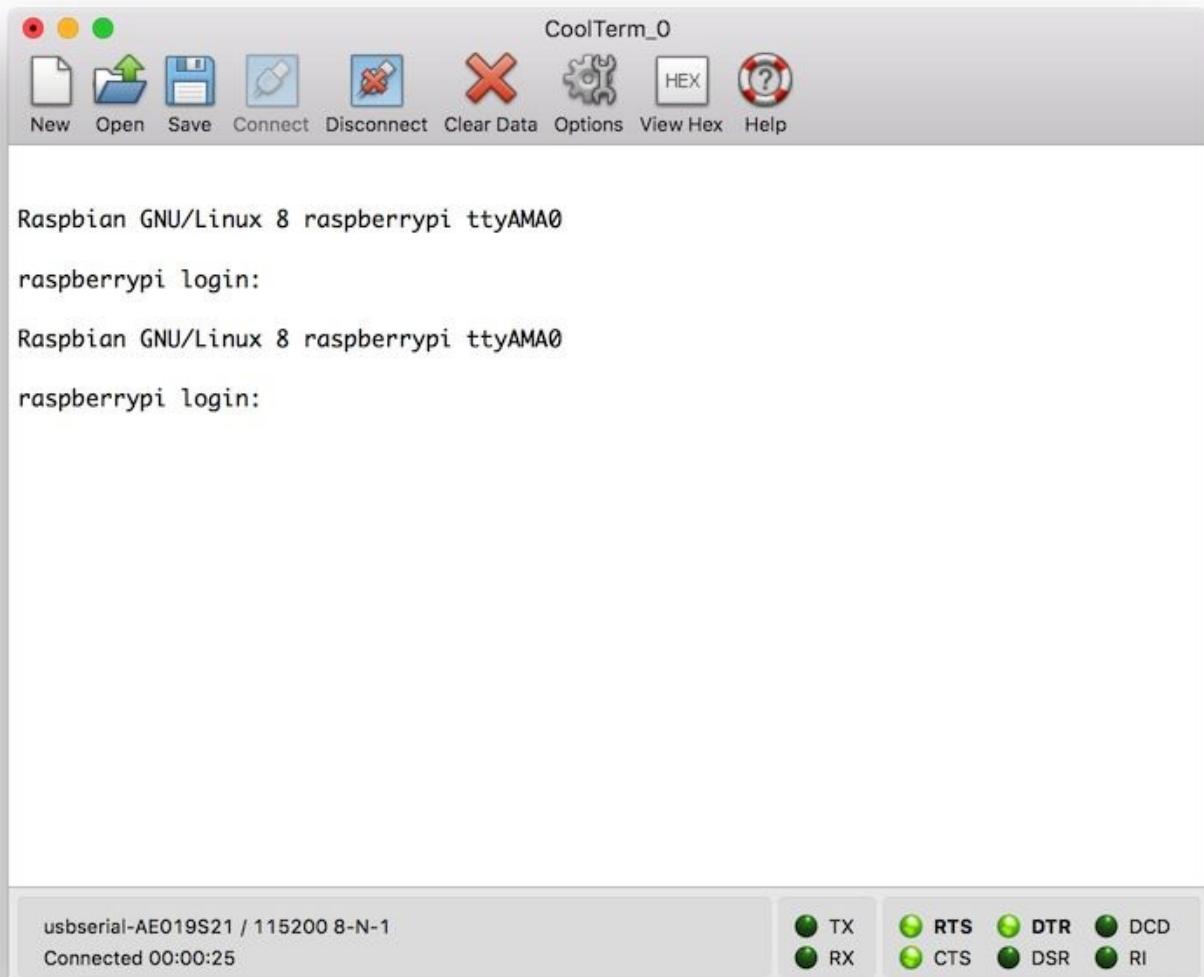


```
Raspbian GNU/Linux 7 raspberrypi ttyAMA0
raspberrypi login: pi
Password:
Last login: Fri Apr 10 22:00:15 WIB 2015 from 192.168.0.16 on pts/0
Linux raspberrypi 3.18.7-v7+ #755 SMP PREEMPT Thu Feb 12 17:20:48 GMT 2015 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

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

If you use CoolTerm, click Connect menu. After that, you can see Login Terminal.



Type username and password.

The screenshot shows the CoolTerm software interface. The title bar reads "CoolTerm\_0". The menu bar includes "New", "Open", "Save", "Connect", "Disconnect", "Clear Data", "Options", "View Hex", and "Help". The main window displays a terminal session with the following text:

```
raspberrypi login:  
Raspbian GNU/Linux 8 raspberrypi ttyAMA0  
raspberrypi login: pi  
Password:  
Last login: Fri Mar  4 16:59:30 UTC 2016 on ttyAMA0  
Linux raspberrypi 4.1.18+ #846 Thu Feb 25 14:11:56 GMT 2016 armv6l  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
[ 972.732857] EXT4-fs error (device mmcblk0p2): ext4_lookup:1588: inode #1849: comm  
bash: deleted inode referenced: 78532  
[ 974.032434] EXT4-fs error (device mmcblk0p2): ext4_lookup:1588: inode #1849: comm  
bash: deleted inode referenced: 78532  
[ 974.049399] EXT4-fs error (device mmcblk0p2): ext4_lookup:1588: inode #1849: comm  
bash: deleted inode referenced: 78532  
pi@raspberrypi:~$  
pi@raspberrypi:~$
```

At the bottom left, it says "usbserial-AE019S21 / 115200 8-N-1" and "Connected 00:00:59". At the bottom right, there are status indicators for TX, RX, RTS, CTS, DTR, DSR, DCD, and RI.

You also can execute common commands such as ls.

**CoolTerm\_0**

New Open Save Connect Disconnect Clear Data Options View Hex Help

```
raspberrypi login: pi
Password:
Last login: Fri Mar  4 16:59:30 UTC 2016 on ttymA0
Linux raspberrypi 4.1.18+ #846 Thu Feb 25 14:11:56 GMT 2016 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[ 972.732857] EXT4-fs error (device mmcblk0p2): ext4_lookup:1588: inode #1849: comm
bash: deleted inode referenced: 78532
[ 974.032434] EXT4-fs error (device mmcblk0p2): ext4_lookup:1588: inode #1849: comm
bash: deleted inode referenced: 78532
[ 974.049399] EXT4-fs error (device mmcblk0p2): ext4_lookup:1588: inode #1849: comm
bash: deleted inode referenced: 78532
pi@raspberrypi:~$ ls
Desktop  Downloads  Pictures  python_games  Videos
Documents  Music      Public    Templates
pi@raspberrypi:~$
```

usbserial-AE019S21 / 115200 8-N-1  
Connected 00:01:08

TX RX RTS CTS DTR DSR DCD RI

## Contact

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