

# **Programming on Raspberry Pi**

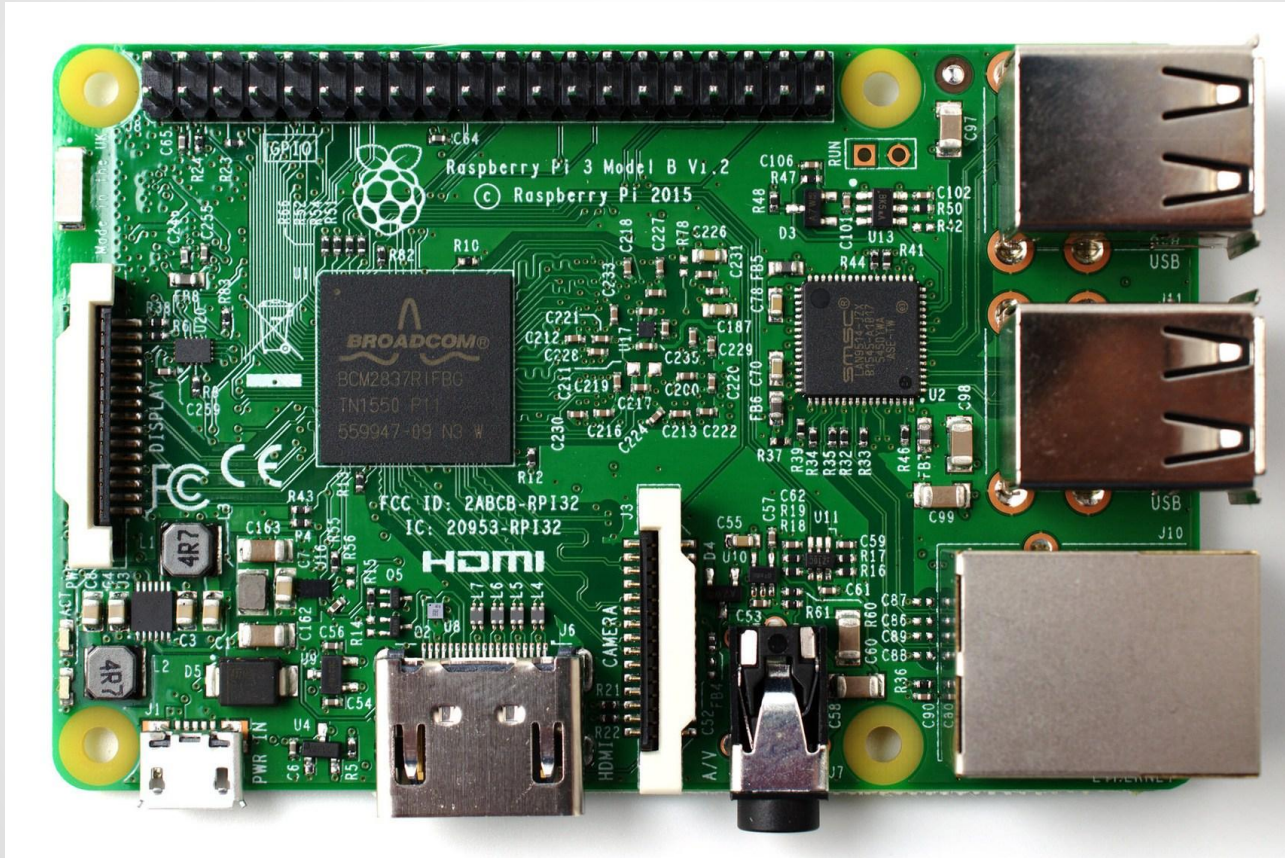
Capstone Project

**<https://goo.gl/abeHro>**

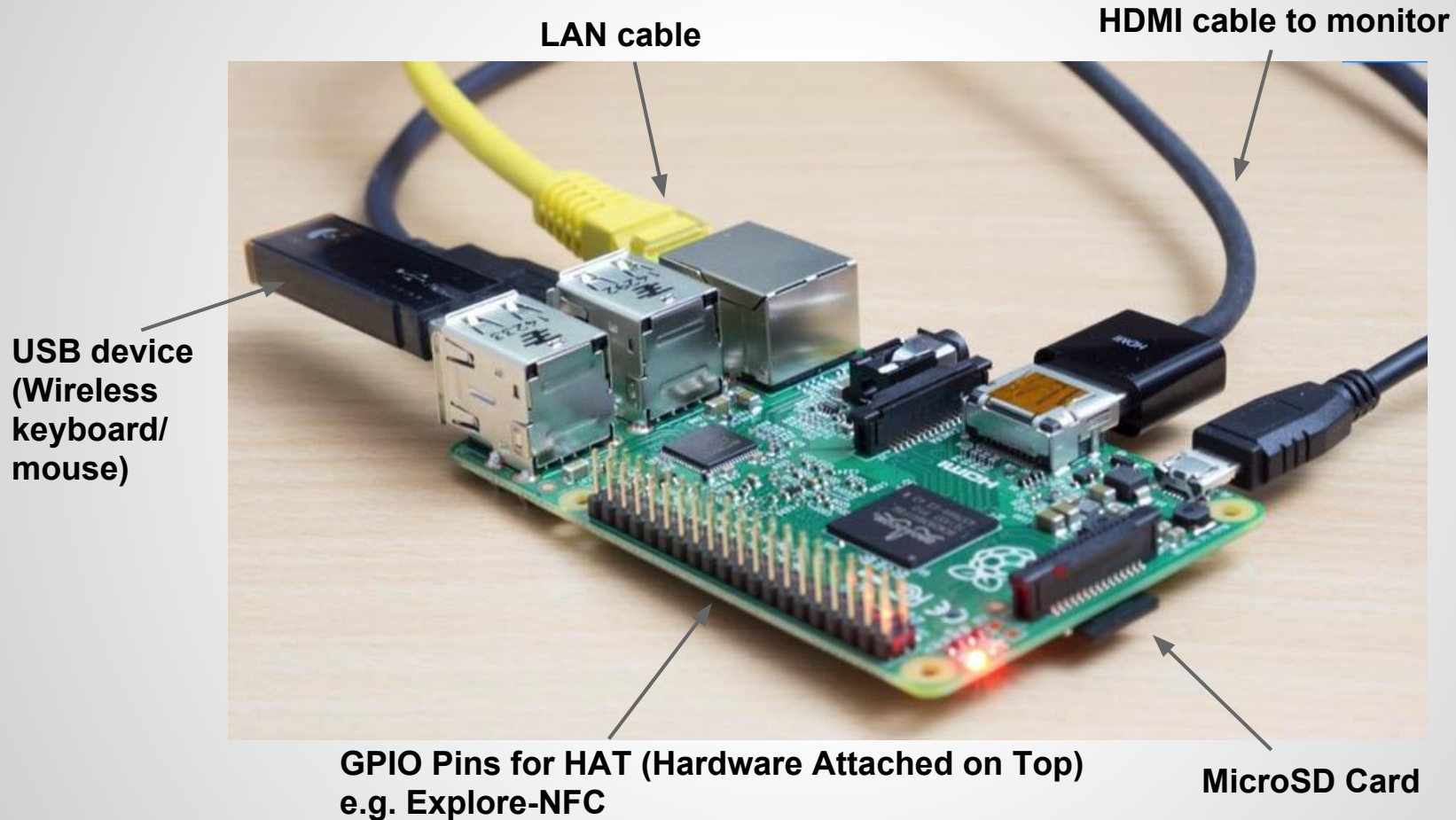
# Raspberry Pi (RPi) 3 Model B

- Single-board computer (SBC)
- 1.2GHz 64-bit quad-core Cortex-A53 ARMv8 processor
- 1GB RAM
- Linux as the main Operating System (OS)
- Persistent storage with microSD Card (for OS, applications)
- Python as the main programming language, Javascript with Node.js is also popular

# Raspberry Pi (RPi) 3 Model B



# Raspberry Pi 3 setup



# Raspberry Pi Power Supply

- MicroUSB connector is used to supply power
- No on/off switch
- 5V
- Recommended power supply current of 2A
- PC/Laptop USB 2.0 port is limited to 500mA
- Unpowered USB device plugged into Raspberry Pi can draw up to 600mA
- Underpowered RPi may still boot up but will fail unpredictably

# Raspberry Pi Display

- Connect the Raspberry Pi HDMI Output to monitor/TV
  - with HDMI input directly with HDMI cable
  - with VGA input via VGA-to-HDMI converter
  - with DVI input via DVI-to-HDMI converter
- Power up Raspberry Pi without HDMI connection to monitor/TV may disable the HDMI output until reset
- Without a display, user can secure shell (SSH) to the Raspberry Pi connected to the same network

# Raspbian

- Officially recommended Linux distribution for RPi
- Download from  
<https://www.raspberrypi.org/downloads/raspbian/>  
[https://www.nxp.com/lqfiles/updates/NFC/OM5577-PN7120S\\_Rpi\\_Linux\\_demo\\_v1.3.zip](https://www.nxp.com/lqfiles/updates/NFC/OM5577-PN7120S_Rpi_Linux_demo_v1.3.zip)
- The downloaded image file in zip format has to be extracted
- The image file (\*.img) has to be installed on a microSD Card (16GB recommended)

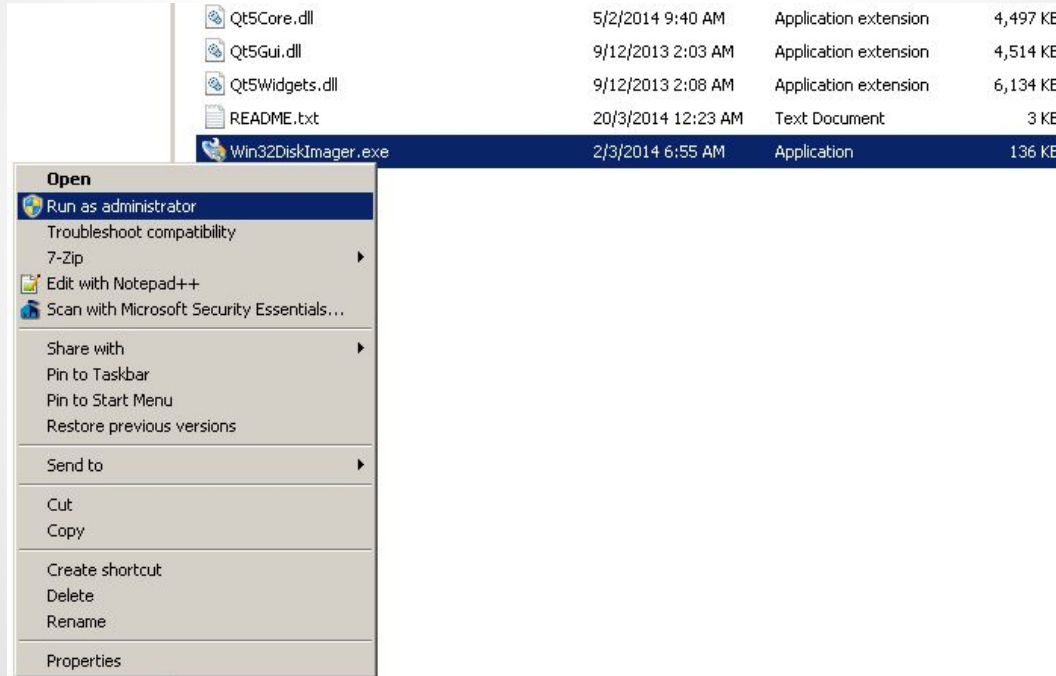
# Install Raspbian image on a microSD card

- Download Win32DiskImager binary (v1.0.0) from <http://sourceforge.net/projects/win32diskimager/files/Archive/>
- Unzip the zip file to a folder and browse into that folder
- Insert the microSD card into a microSD card reader and check the drive letter (e.g. F:)



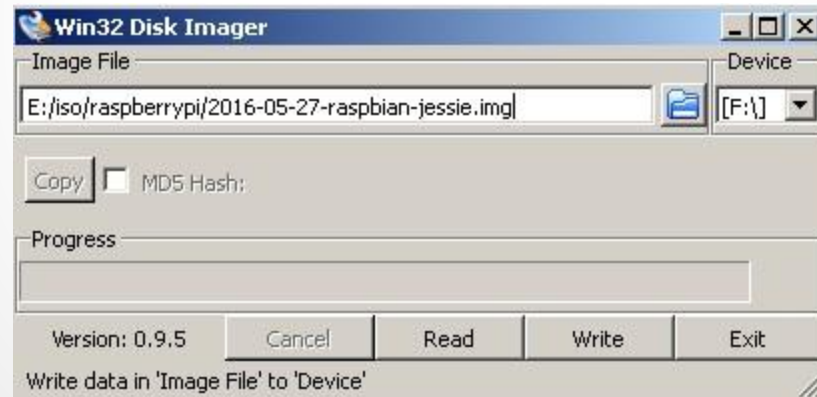
# Install Raspbian image on an SD card

- Right click on Win32DiskImager.exe and select “Run as administrator” and click Yes for User Account Control.



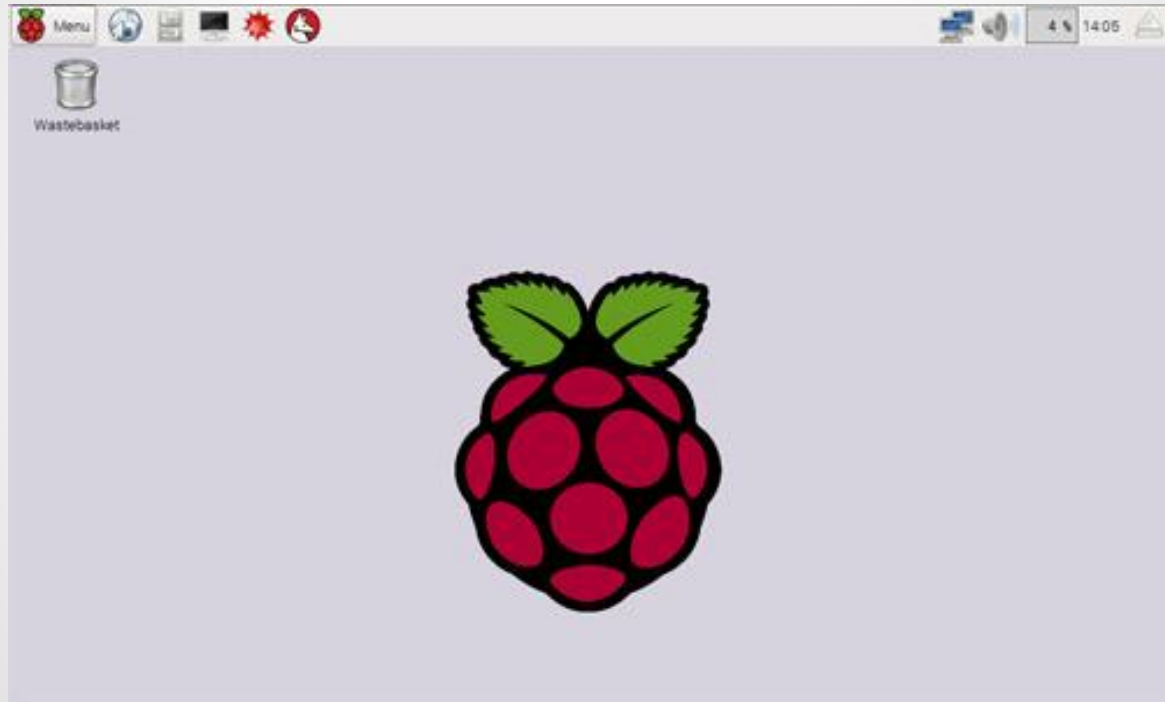
# Install Raspbian image on an SD card

- Select the image (e.g. 2016-05-27-raspbian-jessie.img)
- Select the drive (e.g. F:\)
- Click Write



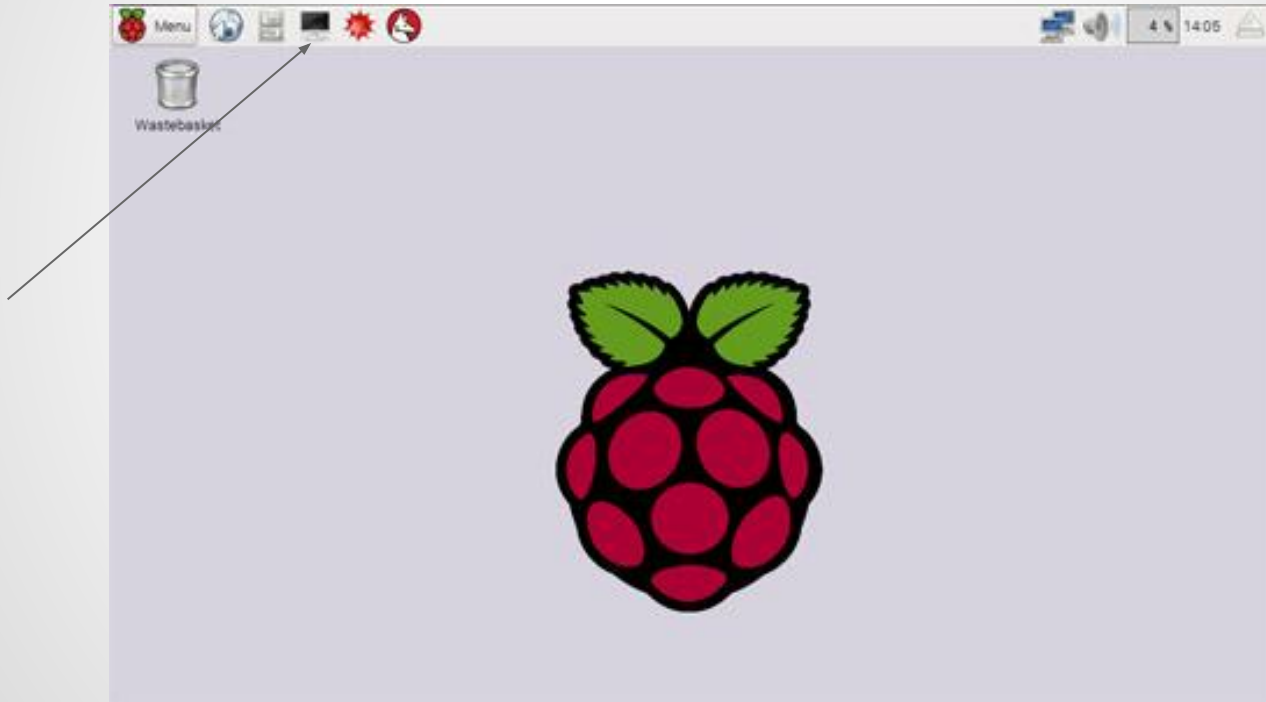
# Raspbian

- Raspbian Jessie desktop



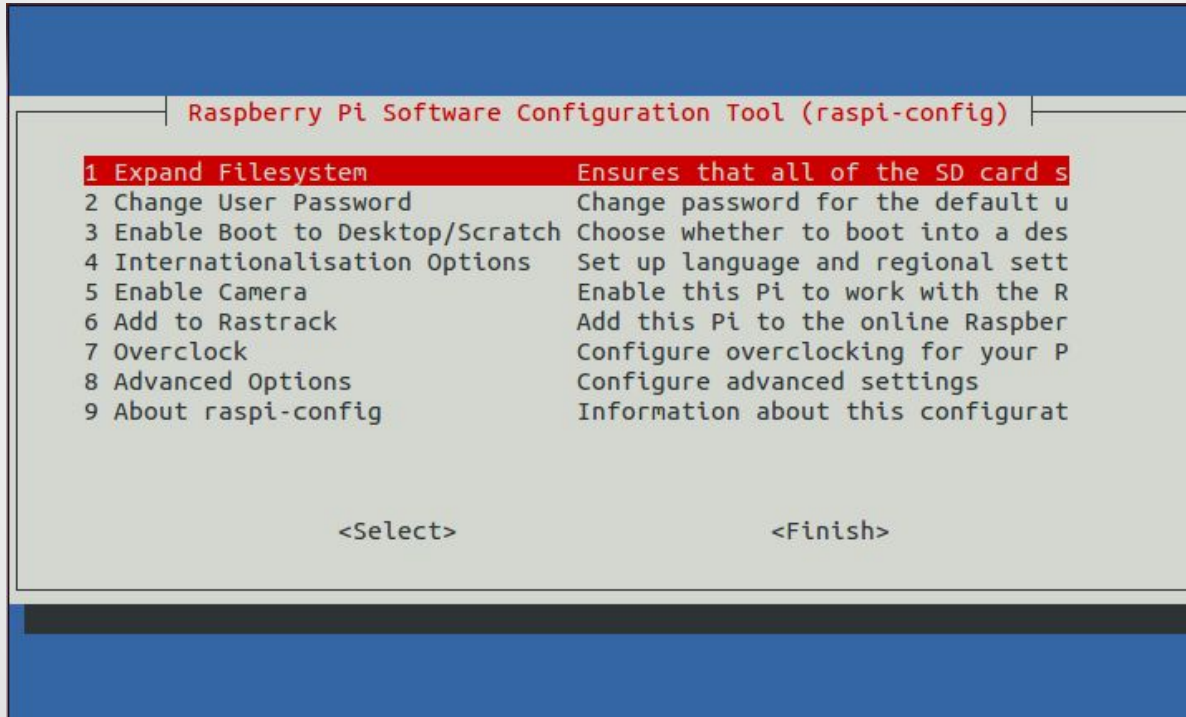
# Raspbian Terminal

- Start the **Terminal** (command prompt)



# Raspi-config

- Run “sudo raspi-config” from terminal



# Raspi-config

- Expand the filesystem to fully use the whole SD card
- In Internationalisation Options
  - Change the Keyboard Layout to a US-style layout from the default UK-style layout
  - Change the Timezone to Asia/Kuala\_Lumpur
- Change the behaviour whether to boot to desktop or not
  - Select “Console” to only boot to command line
  - Can still run **startx** to boot to desktop
  - In desktop, start **Terminal** to get a terminal

# Raspbian

- Default login: username pi, password raspberry

```
[ ok ] Setting up console font and keymap...done.  
[ ok ] Setting kernel variables ...done.  
INIT: Entering runlevel: 2  
[info] Using makefile-style concurrent boot in runlevel 2.  
[ ok ] Network Interface Plugging Daemon...skip eth0...done.  
[ ok ] Starting enhanced syslogd: rsyslogd.  
[ ok ] Starting periodic command scheduler: cron.  
[ ok ] Starting system message bus: dbus.  
Starting dphys-swapfile swapfile setup ...  
want /var/swap=100MByte, checking existing: keeping it  
done.  
[ ok ] Starting NTP server: ntpd.  
[ ok ] Starting OpenBSD Secure Shell server: sshd.  
My IP address is 10.0.2.15  
  
Debian GNU/Linux wheezy/sid raspberrypi tty1  
  
raspberrypi login: pi  
Password: _
```

# Useful Linux Commands

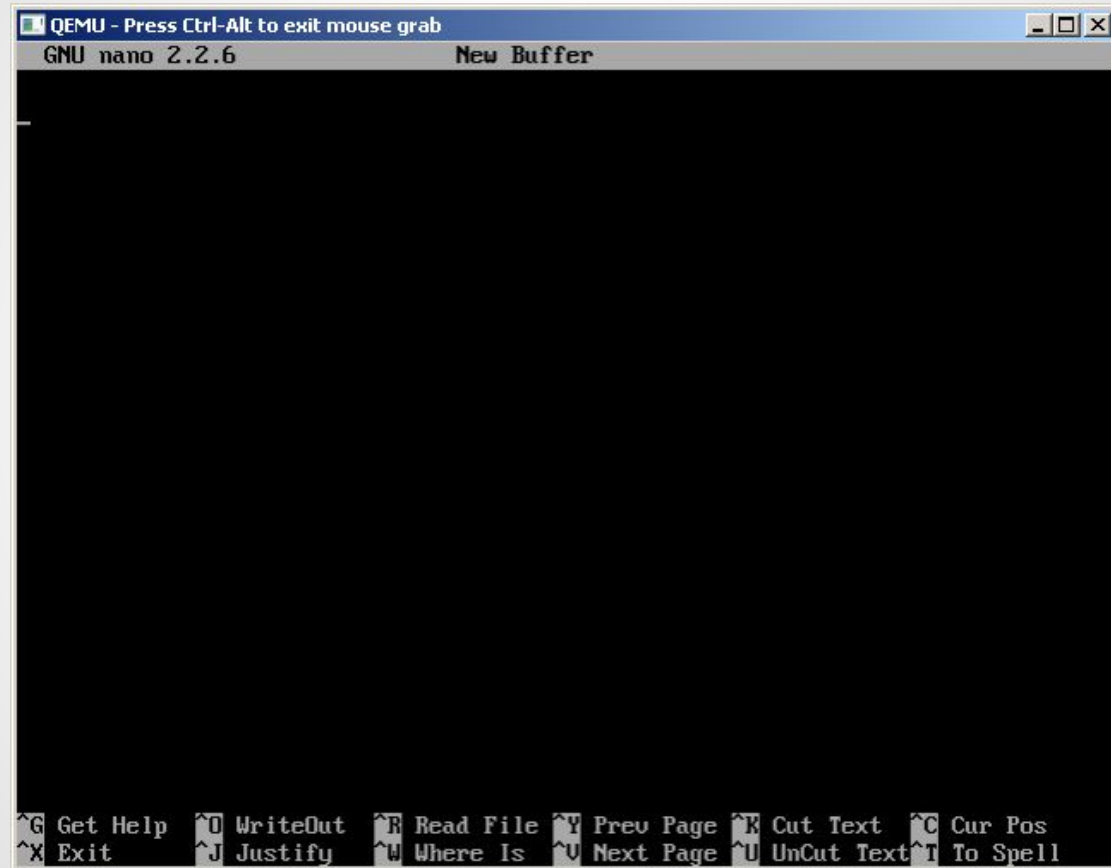
- ls - list directory contents
- mkdir - make directories
- cd - change directory
- rm - remove files or directories
- nano - start Nano text editor
- ping google.com - test Internet connection
- wget <URL> - download a file at <URL>
- man <command> - manual page for a <command> e.g.  
man rm



# Useful Linux Commands

- `sudo <command>` - execute a command as superuser
- `reboot` - reboot the system, requires superuser  
e.g. `sudo reboot`
- `shutdown` - shutdown the system, requires superuser  
e.g. `sudo shutdown -h now`
- `apt-get update`, `apt-get upgrade` and `apt-get install`, requires superuser  
e.g. `sudo apt-get update`  
`sudo apt-get upgrade`  
`sudo apt-get install mplayer2`

# Nano Text Editor



# Nano Text Editor

- Move around with keyboard direction keys
- To quit and save
  - Ctrl-x
  - Enter “y” to answer Yes
  - Enter to confirm the file name

# Secure Shell (SSH)

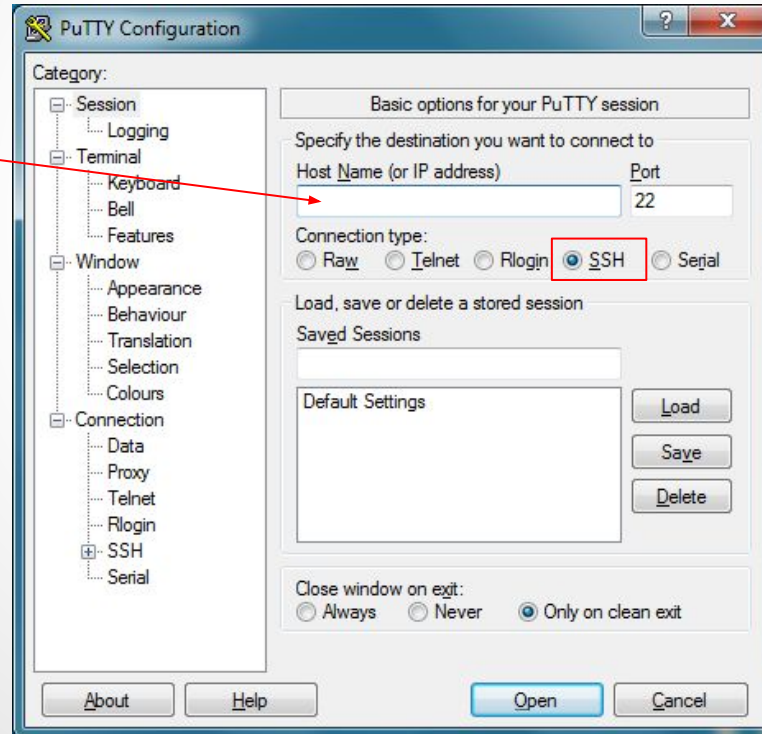
- Access the command line of a Raspberry Pi remotely from another computer connected to the same network (Ethernet or Wi-Fi)
- Obtain the IP Address of the Raspberry Pi
  - Run ifconfig on Raspberry Pi with a display
  - Check the client list of the router that the Raspberry Pi connected to
  - Run network tool such as (Fing, Android App at [https://play.google.com/store/apps/details?id=com.o  
verlook.android.fing](https://play.google.com/store/apps/details?id=com.verlook.android.fing))

# Secure Shell (SSH)

- Using Putty for Windows

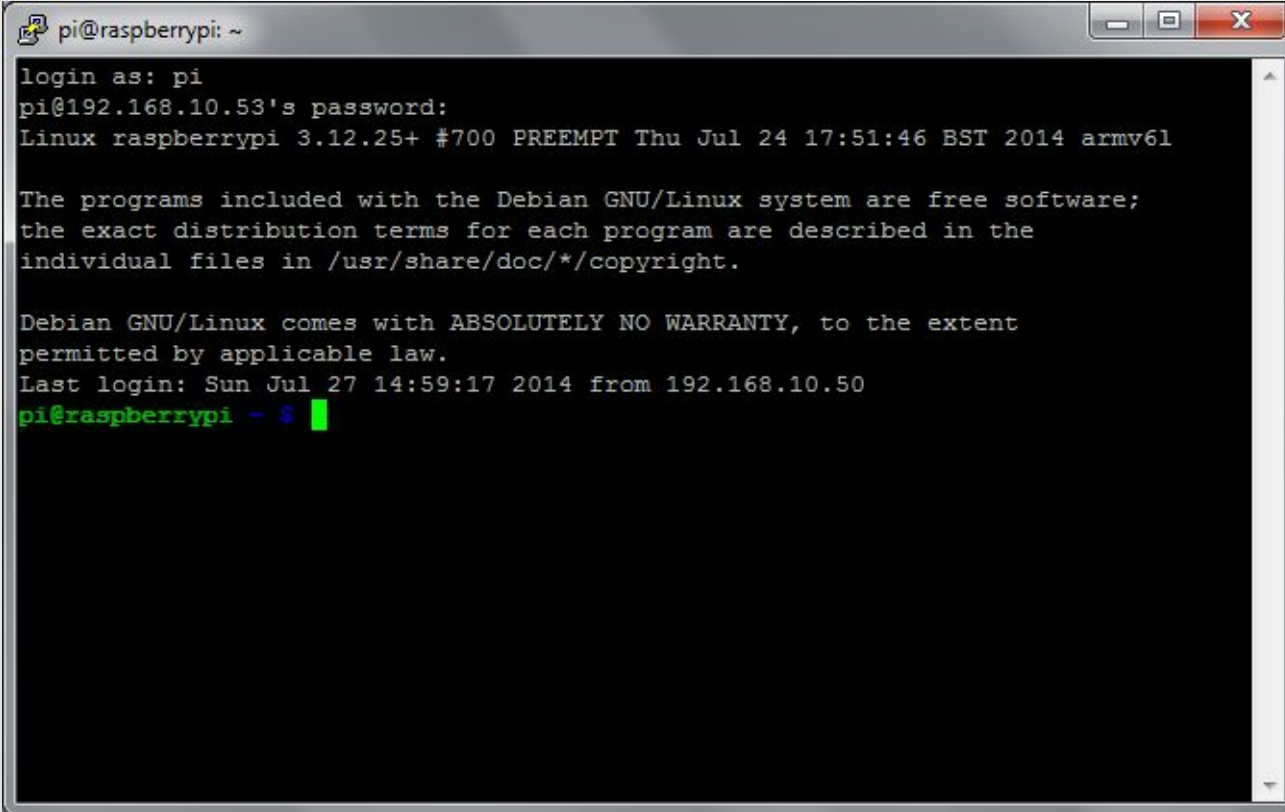
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

IP Address



# Secure Shell (SSH)

- Login to SSH with username pi, password raspberry



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.10.53's password:  
Linux raspberrypi 3.12.25+ #700 PREEMPT Thu Jul 24 17:51:46 BST 2014 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.  
Last login: Sun Jul 27 14:59:17 2014 from 192.168.10.50  
pi@raspberrypi ~ $
```

# Python

- An interpreted language, execute directly without compiling
- Python interpreter is pre-installed in Raspbian
- Run **nano hello.py** to create a Python script
- Enter

```
print "Hello World"
```

- Quit and save the file (CTRL-x → y → Enter)
- Run **python hello.py**

# Python

- **Whitespace** determines the structure of a Python script, not using curly braces/brackets as in C/C++
- For example:

```
n = 0
while True:
    n = n + 1
    if n > 10:
        break
    print n
```



# Python Modules

- A Python module is a Python file that has only definitions of variables, functions and classes that can be imported into other Python scripts
- For example:

```
import math  
print math.sqrt(2)
```

- Without the import:  
NameError: name 'math' is not defined

# Node.js

- An open source runtime environment for server-side applications written in **Javascript**
- Node.js provides an **event-driven** architecture and a non-blocking I/O API used for real-time web applications
- The version of Node.js pre-installed in Raspbian is too old (check with `node --version`)
- Installation:

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo bash -  
sudo apt-get install nodejs
```

# Node.js

- Run **nano hello.js** to create a Node.js script
- Enter

```
console.log("Hello World");
```
- Quit and save the file
- Run **node hello.js**

# Node.js

- For example:

```
n = 0;
while(true) {
    n++;
    if(n > 10) {
        break;
    }
    console.log(n);
}
```

**<https://goo.gl/abeHro>**

Thank you