

Raspberry PI

Basic Development

Brian Vesperman

Not an Arduino

Arduino

- Digital/Analog Pins
- PWM
- Serial Communication
- Little Storage
- needs to be flashed

Raspberry Pi

- Digital Pins
- HDMI/Audio/Video/Ethernet
- Serial Communication
- SD Card
- GUI for development

Operating Systems

- NOOBS - New Out Of Box Software
the OS that allows you to install
the actual OS
- Raspian - Windows 95/XP-ish command
line or GUI
- Pidora - Red Hat style OS
- RaspBMC, OpenELEC - XBMC Server,
media server

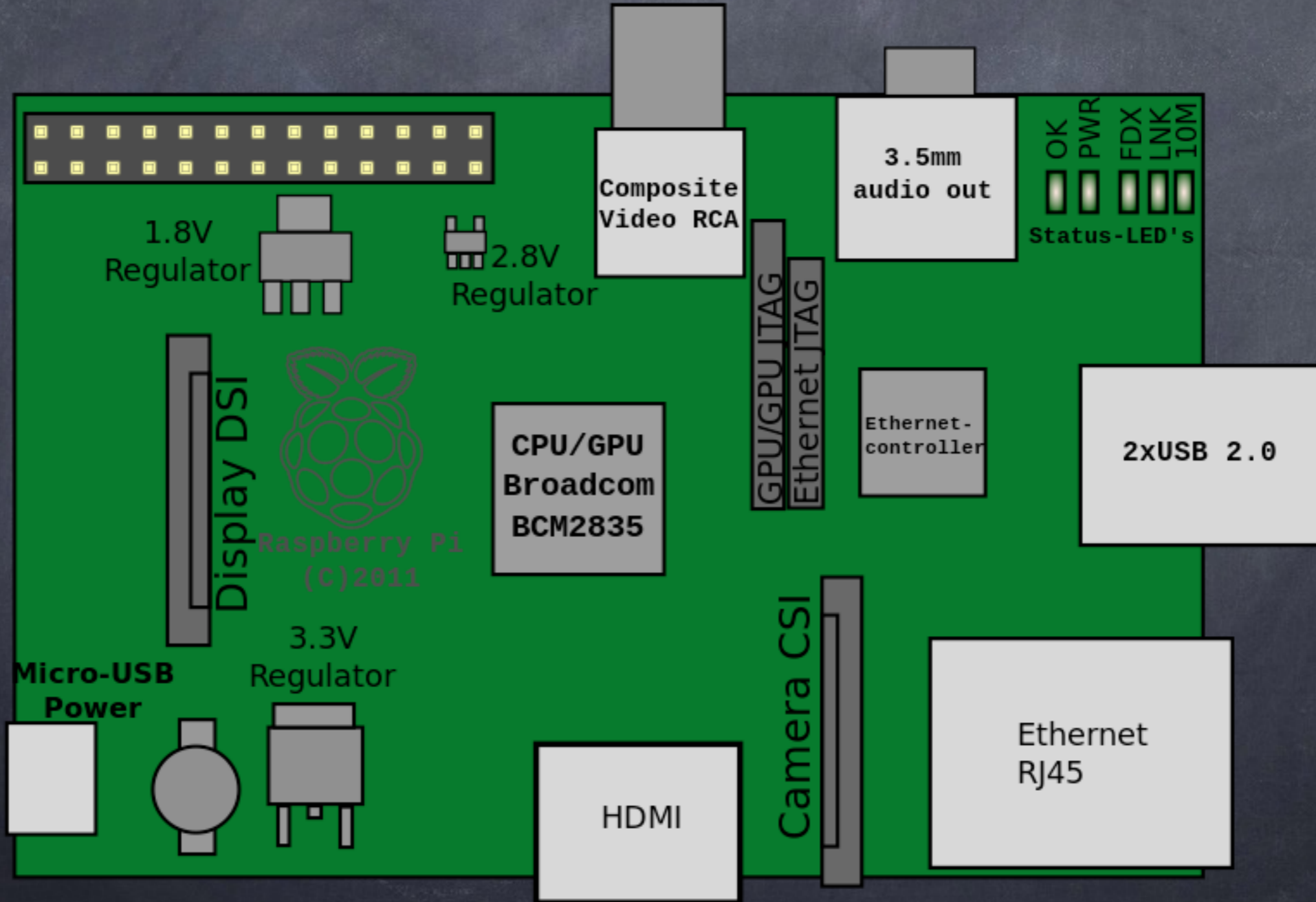
Raspberry Pi Variants

- A : 5V 300 mA (2.5 W)
- B : 5V 200 mA (1 W)
- A+ : 5V 700 mA (3.5 W)
- B+ : 5V 600 mA (3.0 W)
- Gen 2 B : 5V 900 mA (4.5 to 5.5 W)

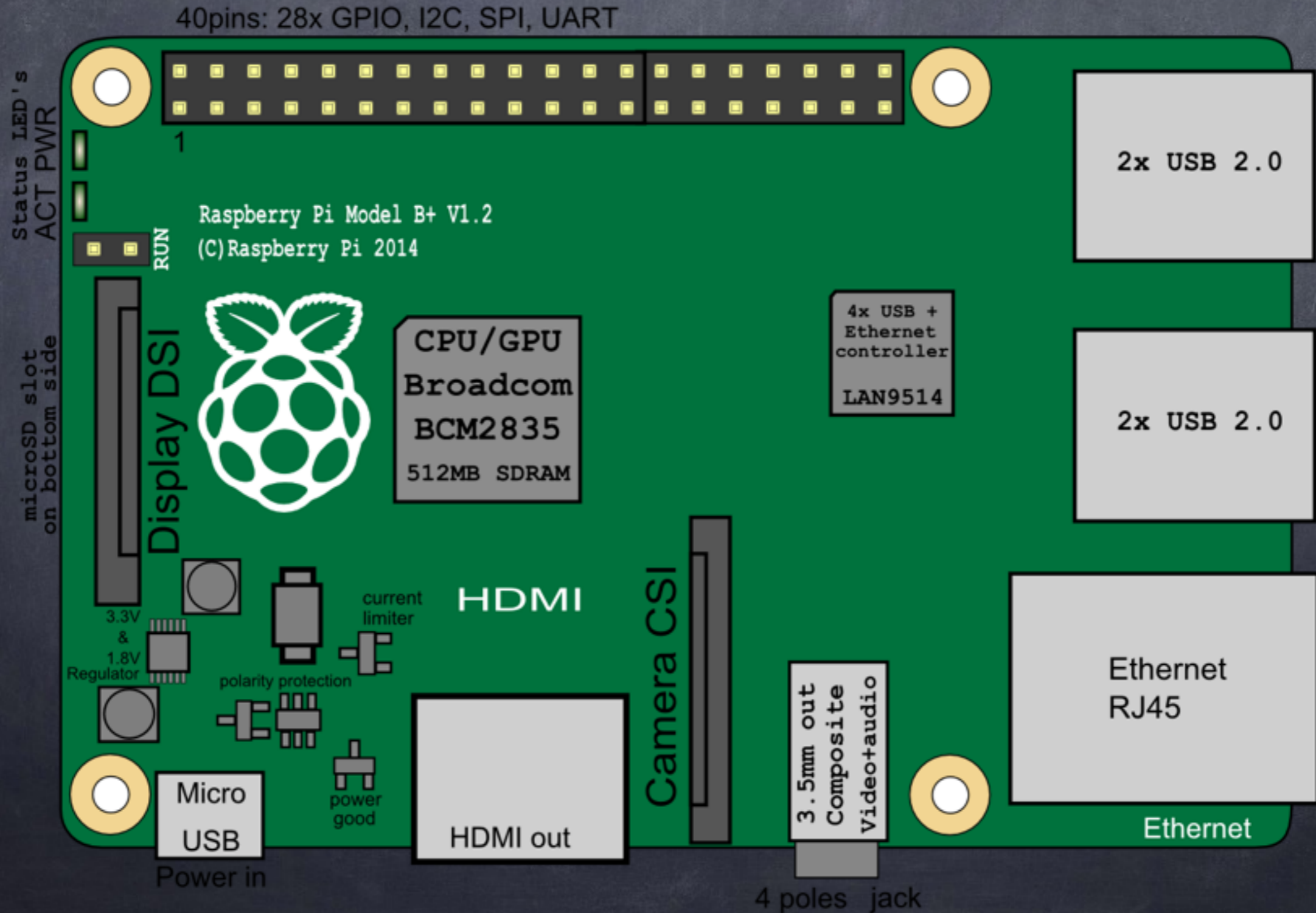
Raspberry Pi Variants

- A : 1 - 700 MHz cpu , 256 MB RAM
- B : 1 - 700 MHz cpu , 512 MB RAM
- A+ : 1 - 700 MHz cpu , 256 MB RAM
- B+ : 1 - 700 MHz cpu , 512 MB RAM
- Gen 2 B : 4 - 900 MHz cpu , 1 GB RAM

Version B, A has only 1 USB and no Ethernet



Version B+, A+ has only 1 USB and no Ethernet



R-Pi GPIO



P1-25 P1-26
bottom top

R1: Revision 1
R2: Revision 2

right

Get The OS and Install

- wheezy-raspbian.zip
- Windows: win32diskimager
- Mac: RPi-sd card builder

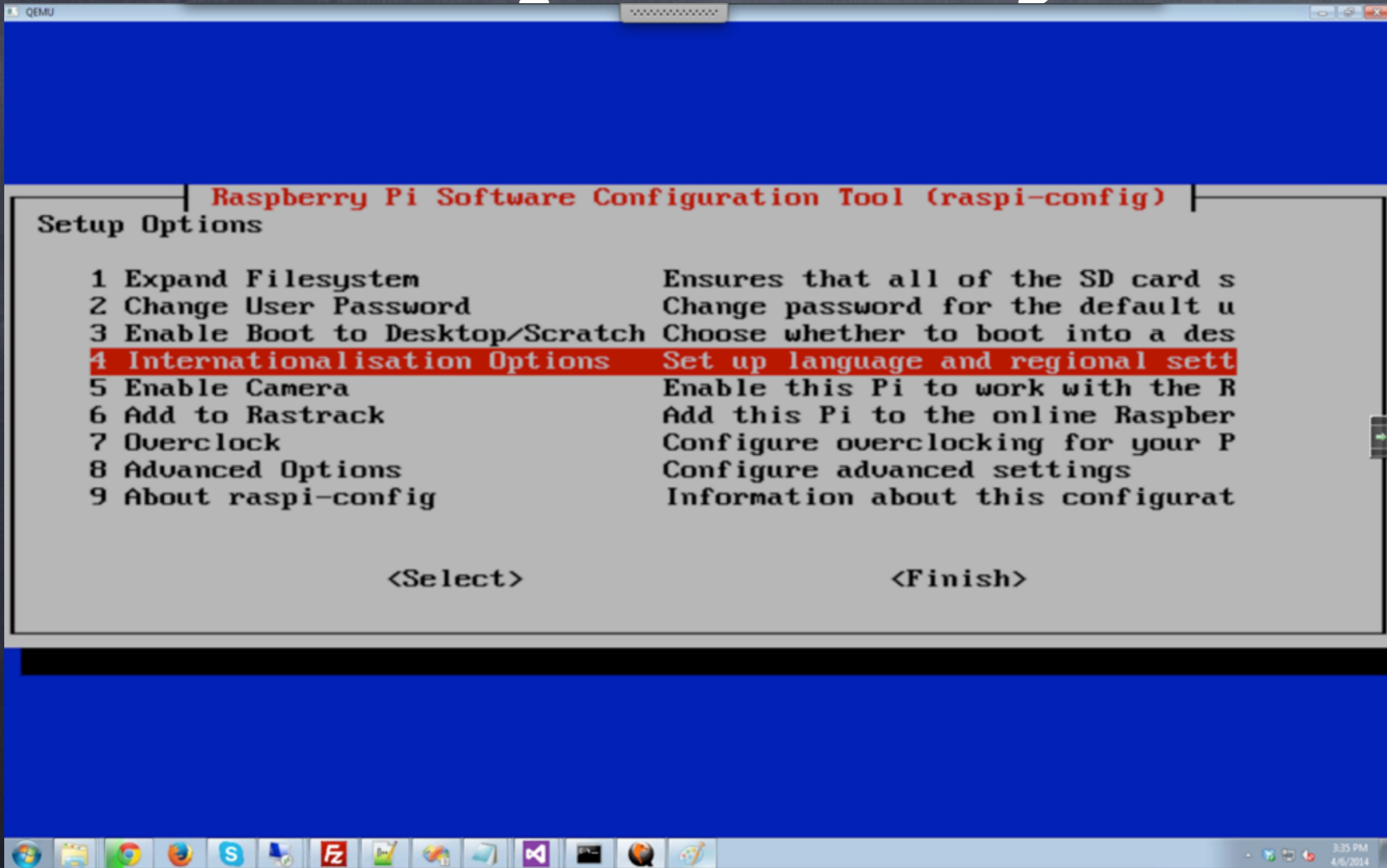
linux things

- ls = dir
- cd / = cd\
- sudo = run as admin (superuser do)
- nano = edit
- chattr = attrib
- cp = copy
- rm = del
- sudo reboot
- bash file = batch file

Boot and Login

- username: pi
- password: raspberry
- Do updates:
sudo apt-get update
sudo apt-get upgrade
sudo raspi-config
- load GUI: startx

Raspi-Config



Setup Options

- | | |
|---------------------------------------|--|
| 1 Expand Filesystem | Ensures that all of the SD card s |
| 2 Change User Password | Change password for the default u |
| 3 Enable Boot to Desktop/Scratch | Choose whether to boot into a des |
| 4 Internationalisation Options | Set up language and regional sett |
| 5 Enable Camera | Enable this Pi to work with the R |
| 6 Add to Rastrack | Add this Pi to the online Raspber |
| 7 Overclock | Configure overclocking for your P |
| 8 Advanced Options | Configure advanced settings |
| 9 About raspi-config | Information about this configurat |

<Select>

<Finish>

Raspberry Pi Software Configuration Tool (raspi-config)

Internationalisation Options

- | | |
|---------------------------|--|
| I1 Change Locale | Set up language and regional sett |
| I2 Change Timezone | Set up timezone to match your loc |
| I3 Change Keyboard Layout | Set the keyboard layout to match |

<Select>

<Back>

Package configuration

Configuring locales

Locales are a framework to switch between multiple languages and allow users to use their language, country, characters, collation order, etc.

Please choose which locales to generate. UTF-8 locales should be chosen by default, particularly for new installations. Other character sets may be useful for backwards compatibility with older systems and software.

Locales to be generated:

```
[ ] en_NG UTF-8
[ ] en_NZ ISO-8859-1
[ ] en_NZ.UTF-8 UTF-8
[ ] en_PH ISO-8859-1
[ ] en_PH.UTF-8 UTF-8
[ ] en_SG ISO-8859-1
[ ] en_SG.UTF-8 UTF-8
[*] en_US ISO-8859-1
[*] en_US.ISO-8859-15 ISO-8859-15
[*] en_US.UTF-8 UTF-8
[ ] en_ZA ISO-8859-1
```

<Ok>

<Cancel>

Package configuration

Configuring keyboard-configuration

Please select the model of the keyboard of this machine.

Keyboard model:

- Dell Latitude series laptop
- Dell Precision M65
- Dell SK-8125
- Dell SK-8135
- Dell USB Multimedia Keyboard
- Dexxa Wireless Desktop Keyboard
- Diamond 9801 / 9802 series
- DTK2000
- Ennyah DKB-1008
- Everex STEPnote
- FL90
- Fujitsu-Siemens Computers AMILO laptop
- Generic 101-key PC
- Generic 102-key (Intl) PC
- Generic 104-key PC
- Generic 105-key (Intl) PC**



<Ok>

<Cancel>



Package configuration

Configuring keyboard-configuration

The layout of keyboards varies per country, with some countries having multiple common layouts. Please select the country of origin for the keyboard of this computer.

Country of origin for the keyboard:

- Catalan
- Chinese
- Croatian
- Czech
- Danish
- Dhivehi
- Dutch
- Dzongkha
- English (Cameroon)
- English (Ghana)
- English (Nigeria)
- English (South Africa)
- English (UK)
- English (US)**



<Ok>

<Cancel>

Package configuration

Configuring keyboard-configuration

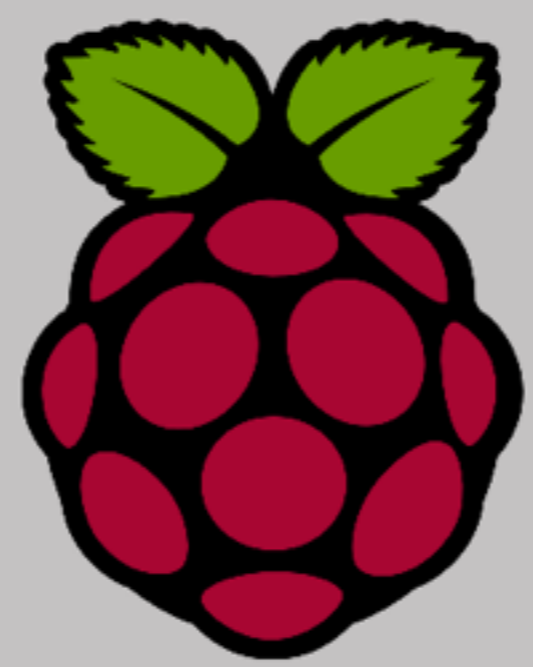
Please select the layout matching the keyboard for this machine.

Keyboard layout:

- English (US)**
- English (US) - Cherokee
- English (US) - English (classic Dvorak)
- English (US) - English (Colemak)
- English (US) - English (Dvorak)
- English (US) - English (Dvorak alternative international no dead keys)
- English (US) - English (Dvorak international with dead keys)
- English (US) - English (international AltGr dead keys)
- English (US) - English (layout toggle on multiply/divide key)
- English (US) - English (left handed Dvorak)
- English (US) - English (Macintosh)
- English (US) - English (programmer Dvorak)
- English (US) - English (right handed Dvorak)
- English (US) - English (US, alternative international)
- English (US) - English (US, international with dead keys)
- English (US) - English (US, with euro on 5)

<Ok>

<Cancel>



Networking setup

- Menu
- Preferences
- WiFi Configuration
- Hit scan - Find s67-guest, double click
For PSK(password):WelcomeToSector67
- Close out test internet in web browser

Install code IDE

- `sudo apt-get install geany`
- `git clone https://github.com/bvesperman/PiClass.git`



client.py - /home/pi/CodeExamples - Geany

File Edit Search View Document Project Build Tools Help

helloworld.py bashexample.sh CplusplusExample.cpp GPIO blink.py server.py client.py

```

6 # Copyright 2015 <pi@raspberrypi>
7 #
8 # This program is free software; you can redistribute it and/or modify
9 # it under the terms of the GNU General Public License as published by
10 # the Free Software Foundation; either version 2 of the License, or
11 # (at your option) any later version.
12 #
13 # This program is distributed in the hope that it will be useful,
14 # but WITHOUT ANY WARRANTY; without even the implied warranty of
15 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 # GNU General Public License for more details.
17 #
18 # You should have received a copy of the GNU General Public License
19 # along with this program; if not, write to the Free Software
20 # Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
21 # MA 02110-1301, USA.
22 #
23 #
24
25 import socket
26
27 s = socket.socket()
28 host = socket.gethostname()
29 port = 12345
30
31 s.connect((host, port))
32 print s.recv(1024)
33 s.send('Set UI asdlkjasd')
34 s.close()
35
36 s = socket.socket()
37 s.connect((host, port))
38 print s.recv(1024)
39 s.send('Get UI')
40 print s.recv(1024)
41 s.close()
42

```

Status 01:56:51: This is Geany 1.22.

01:56:52: File /home/pi/CodeExamples/helloworld.py opened(1).

Compiler 01:56:53: File /home/pi/CodeExamples/bashexample.sh opened(2).

01:56:53: File /home/pi/CodeExamples/CplusplusExample.cpp opened(3).

Messages 01:56:53: File /home/pi/CodeExamples/GPIO blink.py opened(4).

01:56:53: File /home/pi/CodeExamples/server.py opened(5).

Scribble 01:56:53: File /home/pi/CodeExamples/client.py opened(6).

Terminal

First Bash Script

```
# create a folder on the desktop  
# called {code}
```

```
#remember to make the file executable  
echo "Hello world"  
ls
```

First C++ Program

```
//File -> New (with template) ->  
main.cxx
```

```
#include <iostream>  
using namespace std;
```

```
int main(int argc, char **argv)  
{  
    cout << "Hello World" << endl;  
    return 0;  
}
```

First Python Program

```
print('Hello World')
```


Python Keyboard Input

```
i = int(raw_input("Enter a number:"))  
print('Hello World {0}'.format(i))
```

Python Basic Logic

```
while True:  
    i = int(raw_input("Enter a number:"))  
    if i == 1:  
        print('Hello World {0}'.format(i))  
    else:  
        print('No way')
```

Exceptions in Python

```
while True:
    try:
        i = int(raw_input("Enter a number:"))
        if i == 1:
            print('Hello World {0}'.format(i))
        else:
            print('No way')
    except ValueError:
        print 'not a number'
```

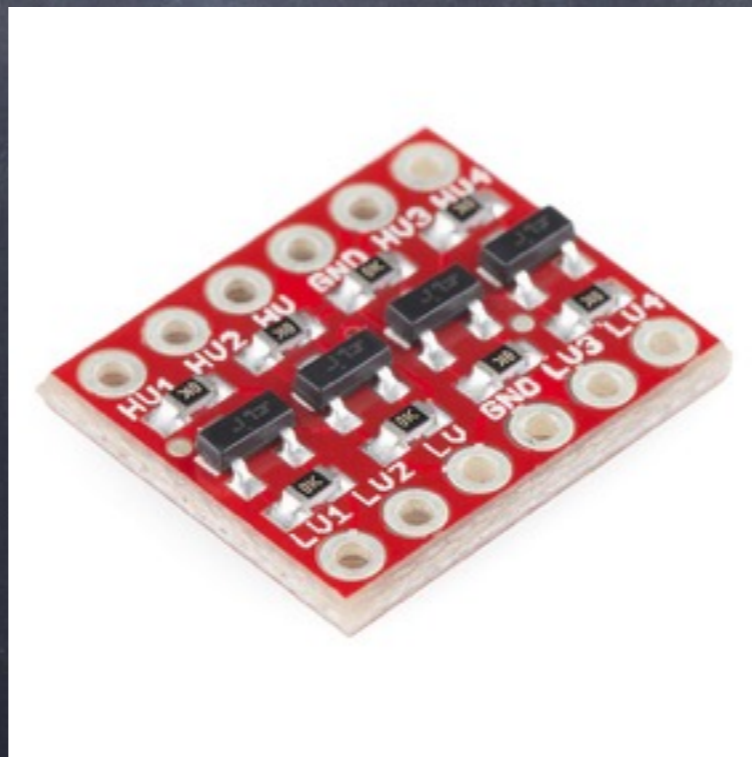
Python GPIO

```
#GPIO.py
import RPi.GPIO as io
import time
io.setmode(io.BCM)
io.setup(17, io.OUT)
while True:
    io.output(17, True)
    time.sleep(5)
    io.output(17, False)
    time.sleep(5)

#sudo python GPIO.py
```

Electronics

- GPIO runs at 3.3 V If interfacing with 5 V components (arduino) you need a level logic shifter
- <https://www.sparkfun.com/products/12009>



Running on boot

- `sudo nano /etc/rc.local`
 - similar to `autoexec.bat`
- `crontab -e`
 - similar to task manager

Additional - Webserver

- `sudo apt-get install apache2 -y`
- `sudo apt-get install php5
libapache2-mod-php5 -y`
- `/var/www/index.html`
- `http://localhost/` , `hostname -I`

Node JS

- `http://weworkweplay.com/play/raspberry-pi-nodejs/`
- `wget http://node-arm.herokuapp.com/node_latest_armhf.deb`
- `sudo dpkg -i node_latest_armhf.deb`
- `su pi -c 'node /home/pi/server.js </dev/null &'`

Additional – NAS

- <http://www.tinkernut.com/portfolio/make-raspberry-pi-nas-network-attached-storage/>