

Trinity School Raspberry Pi Specification

Hardware

- 10 x Raspberry Pi (RPi) each with:
 - Power supply
 - Case preferably transparent
 - [USB memory stick]
 - 8GB SD card with operating system preconfigured (by PVT)
 - Operating system uses about 2-3 GB
 - x 15 to provide some spares
 - Ethernet network cable
- 1 x local control kit for problem determination, performance analysis, and multimedia applications
 - HD TV with HDMI input
 - Or monitor with HDMI input and either inbuilt sound or separate powered speakers
 - Basic wired USB keyboard
 - Basic wired USB mouse
 - SD card reader/writer?
- 1 x RPi network
 - 16 port Ethernet switch
 - Ethernet network cable to interconnect with school network
 - Dedicated backup device for student work on RPi?
- 1 x RPi power system
 - Power distribution of 12 x 13A sockets

Software on RPi

- Raspbian “wheezy” operating system - Optimised version for RPi of Debian GNU/Linux
- Python 3 programming language
- PyGame toolkit for Python - which adds useful classes for game making in particular
- VNC Server - to allow another computer to drive it remotely
- Adobe Chromium Browser

Software on [some] School Windows Computers

- Python 3 – ensure is exactly same version as on RPi computers, 3.2.3 at present
 - <http://www.python.org/download/>
 - e.g. Python 3.2.3 Windows x86 MSI Installer (python-3.2.3.msi)
 - Do not use 64 bit version as this is not compatible with PyGame
 - Beware there are two current versions of Python (2 and 3), we want latest Python 3.something
- PyGame (Windows) – ensure is exactly same version as on RPi computers
 - <http://www.pygame.org/download.shtml>
 - Be careful that filename has correct version of Python
e.g. pygame-1.9.2a0.win32-**py3.2**.msi 6.4MB
- SSH Client (emergency Telnet access to a RPi)
 - Download PuTTY from <http://www.putty.org/> (putty.exe) Runs from here, just needs shortcut
- VNC Client (normal xWindow remote control of RPi)
 - Download and install Tightvnc Viewer only
 - www.tightvnc.com version 2.5.2
- If SD cards to be “burned” on suitable reader/writer
 - Download Image Writer for MS Windows 0.6 (r46)
 - <https://launchpad.net/win32-image-writer>
 - Pre-req is QT for Windows SDK (a big install!) - <http://qt.nokia.com/downloads/>
 - Download sha1sum.exe to allow image checksum to be verified
 - http://downloads.raspberrypi.org/verifying_an_image.html

RPi Operating System Build

- Base = Raspbian “Wheezy” Linux (2012-07-15-wheezy-raspbian.img)
- **config.txt** Adjust for HDMI monitor/TV provided for maintenance
Note: Can be edited from Windows machine with SD card reader, but use either **Wordpad** or a specialised editor – **not** Notepad:

```
disable_overscan = 0
overscan_left = 16
overscan_right = 16
overscan_top = -4
overscan_bottom = -4
hdmi_drive = 2
```

This is for Mr V's HDMI TV

- **SSH** Secure Shell allows access to the RPi console using a secure channel.
 - Enabled on the RPi by default so can use PuTTY from Windows for initial setup and maintenance.
 - Start PuTTY and use DHCP-assigned IP address (which you have to find out) to connect to the RPi.
- **Static IP Address** Using SSH – if required
 - This is the default network configuration in the /etc/network/interfaces file which allow the RPi to search for a DHCP server to allocate it a temporary IP address. It can be used in this configuration, especially if later SAMBA configuration is allowed as it will replace the address with a symbolic NetBios name which can be used for SSH and VNC connections instead. If not, then this is where static IP addresses for the Raspberry Pi computers might make sense.

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp
```

- This is the procedure to change the RPi to use a static IP address:
 - The use of **sudo** in front of a command gives the user “pi” root (super user) authority.
 - Permission to use sudo by the user pi is separately authorised in the user setup.
 - # in the file means a comment line.

```
sudo cp /etc/network/interfaces /etc/network/interfaces.sav    Make backup first!
sudo nano /etc/network/interfaces                               Edit interfaces file
```

```
# The loopback interface
auto lo
iface lo inet loopback
# Ethernet port
#iface eth0 inet dhcp
auto eth0
iface eth0 inet static
address 192.168.0.100
# Router
gateway 192.168.0.1
netmask 255.255.255.0
# VINNTEC network
network 192.168.0.0
broadcast 192.168.0.255
```

This is for Mr V's network
- will need to be changed
for school use

- Check **VERY CAREFULLY** before saving – mistakes can leave the RPi unable to communicate on SSH!

| | |
|--|---|
| Ctrl-Z | To exit the editor |
| Y | “Save modified buffer” (or “N” to discard) |
| ENTER | “File Name to Write:” to keep existing name |
| sudo /etc/init.d/networking restart | Start using the changes |

- **RASPI-CONFIG** Make basic configuration decisions from SSH.

| | |
|--------------------------------|---|
| <code>sudo raspi-config</code> | Edit using simple editor |
| Info | Information about the tool |
| Expand_rootfs | Expand root partition to fill SD disk |
| Overscan | Enabled |
| Configure_keyboard | Layout of keyboard provided [only if connected] |
| Change_pass | Leave password for pi as default raspberry! |
| Change_locale | GB by default but hard to work out from these menus! |
| Change_timezone | Europe – London |
| Memory_split | 192 192 MIB for ARM (RAM), 64 MIB for VideoCore (video) |
| SSH | Enable to allow console access remotely for maintenance |
| Boot_behaviour | No to start desktop on boot |
| Update | Try to upgrade RASPI-CONFIG - should usually be done <u>first</u> |

- **INSTALLED PACKAGES**

| | |
|-----------------------------------|---|
| <code>sudo apt-get update</code> | Resynchronise distribution |
| <code>sudo apt-get upgrade</code> | Upgrade distribution with any <u>released</u> changes |

- **FIRMWARE** Used to boot the RPi – only use the **rpi-update** command when directed!

```
sudo uname -a
sudo /opt/vc/bin/vcgencmd version
sudo wget http://goo.gl/1BOfJ -O /usr/bin/rpi-update
sudo chmod +x /usr/bin/rpi-update
```

FIRST TIME ONLY

- First response should be similar to **Linux raspberrypi 3.1.9+ ...**
- Second response should be similar to **July 14 2012 Version 325444**
- The last two commands should only be done the first time as they install the rpi-update command

- **VNC Server** – see <http://www.penguintutor.com/linux/tightvnc> for details. Virtual Network Computing (VNC) allows remote access to the RPi's graphical desktop.

| | |
|--|--|
| <code>sudo apt-get update</code> | Resync distribution |
| <code>sudo apt-get install tightvncserver</code> | Install VNX Server |
| <code>/usr/bin/tightvncserver</code> | Set remote login password <u>for user pi</u> |

- Use the `sudo vncpasswd` command at any time to change this password later.
- Make the VNC configuration file using the simple Nano editor:

```
sudo nano /etc/init.d/tightvncserver
```

```
#!/bin/sh
# /etc/init.d/tightvncserver
# Start tightvncserver at boot time.

VNCUSER='pi'

case "$1" in
  start)
    su - $VNCUSER -c '/usr/bin/tightvncserver :1 -geometry 1024x768 -depth 24'
    echo "Starting TightVNC Server for $VNCUSER "
    ;;
  stop)
    pkill Xtightvnc
    echo "Stopping TightVNC Server"
    ;;
  *)
    echo "Usage: /etc/init.d/tightvncserver {start|stop}"
    exit 1
    ;;
esac

exit 0
```



- **NOTE:** `su - $VNCUSER..` will give you the same environment as `pi`. If the first dash is missed out then it will still work but you will see `root`'s environment (but with `pi`'s authority).

| | |
|---|---------------------------------------|
| <code>sudo chmod 755 /etc/init.d/tightvncserver</code> | Make procedure above executable |
| <code>sudo update-rc.d /etc/init.d/tightvncserver defaults</code> | Enable dependency based boot sequence |
| <code>sudo update-rc.d tightvncserver defaults</code> | If previous command gives errors |
| <code>sudo shutdown -r 0</code> | Reboot to fire it up |

- Try and make a connection from a school computer with VNC Client installed.

- **SAMBA** – advertise the RPI's SD card to school network as a disk
 - Install Samba

```
sudo apt-get install samba
sudo apt-get install samba-common-bin
```

- Edit the samba configuration file using the Nano editor

```
sudo nano /etc/samba/smb.conf
```

```
[global]
workgroup = VINNTEC
netbios name = RASPP1
security = share
domain master = yes
local master = yes
preferred master = yes
os level = 65

[public]
comment = Data
path = /
read only = No
force user = pi
force group = users
guest ok = Yes

[boot]
comment = Emergency access to config.txt
path = /boot
read only = No
force user = pi
force group = users
guest ok = Yes
```

This is what it will advertise itself on the network as

This is for Mr V's workgroup network - will need to be changed for school use

- Define password for accessing the disk

```
sudo smbpasswd -a pi
New SMB Password: raspberry
Retype new SMB password: raspberry
```

- Restart Samba – give it a few minutes and see if the SD card is visible to the school computers

```
sudo /etc/init.d/samba restart
```

- **PYTHON 3**
 - Version 3.2.3 pre-installed. First ensure it is up to date

```
sudo apt-get install python3
```

- **PYGAME** (check latest instructions at <http://www.pygame.org/wiki/compileUbuntu?parent=index>)

```
sudo apt-get install python3-dev libsdl-image1.2-dev libsdl-mixer1.2-dev
sudo apt-get install libsdl-ttf2.0-dev libsdl1.2-dev libsmpeg-dev
sudo apt-get install python-numpy subversion libportmidi-dev ffmpeg
sudo apt-get install libswscale-dev libavformat-dev libavcodec-dev
hg clone https://bitbucket.org/pygame/pygame
cd pygame
python3 setup.py build
sudo python3 setup.py install
```

- **Make Python 2 invisible** as we will only be using Python 3
 - Delete the **IDLE** and **Python Games** icons from the desktop (right-click then delete)
 - Remove **IDLE** from the menu system

```
cd /usr/share/applications
sudo mv idle.desktop /usr/share/idle.desktop
sudo mv idle-python2.7.desktop /usr/share/idle-python2.7.desktop
sudo mv python2.7.desktop /usr/share/python2.7.desktop
```

- **Adobe Chromium Web Browser** A better browser than the default ones but still limited

```
sudo apt-get install chromium-browser
```

- **OMXPLAYER** Simple multimedia player able to play full motion 1080p HD videos
 - Cannot be used across the network
 - Command below assumes HDMI vision + sound being used
 - Very little control once it is started
 - Run from console or xwindow or add command to MP4 files in file manager

```
omxplayer -o hdmi "filename"    (if playing sound up HDMI cable)
```

- Might need to be reinstalled if it doesn't work

```
apt-get install omxplayer
```

- **USB Memory Stick**

- Should mount automatically when in desktop environment
- Outside desktop, if required:

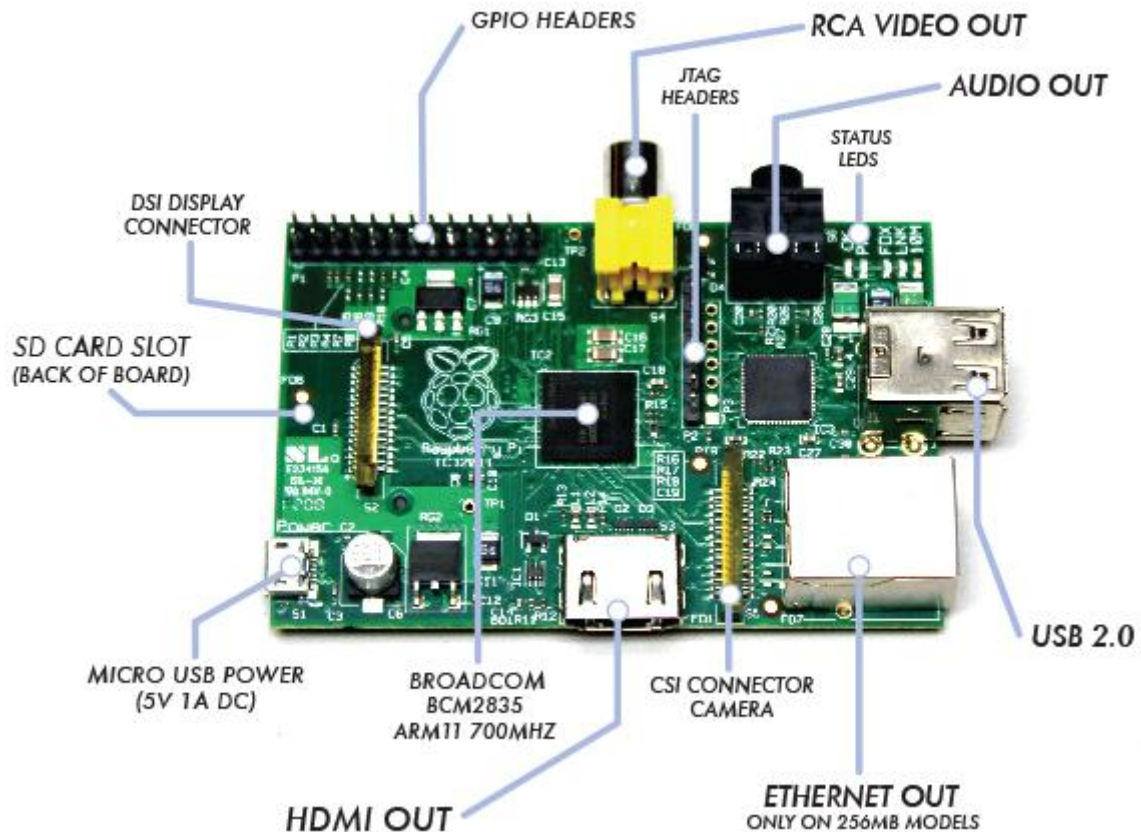
| | |
|--|--|
| <code>sudo fdisk -l</code> | List drives, look for /dev/sdxn (sda1 usually) |
| <code>sudo mkdir /media/external-drive</code> | Make mount point |
| <code>sudo chgrp -R users /media/external-drive</code> | Change ownership to be for any users |
| <code>sudo chmod -R g+w /storage</code> | Give read write access |

Raspberry Pi Reference

Raspberry Pi Status LEDs

Looking left to right:

- **10M** Ethernet running at 100 Mbps Continually on
- **LNK** Ethernet being used Flashes
- **FDX** Ethernet running full duplex Continually on
- **PWR** 5V power supply present on mini-USB port Continually on
- **OK** SD card being used Should normally be off



Specification

- **Broadcom BCM2835 700MHz ARM1176JZFS** processor with FPU and **Videocore 4 GPU** which provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. The GPU is capable of 1G pixel/s, 1.5 Gtexel/s or 24 GFLOPs with texture filtering and DMA infrastructure.
- **256MB RAM** (not upgradeable)
- **SD card socket** – the system boots from an SD card, running a version of the Linux operating system
 - Operating system takes up about 2GB
 - 8GB cards are probably most appropriate for the school
- **10/100 BaseT Ethernet** socket
- **Monitor** connections consisting of HDMI video socket and RCA composite video socket
 - Only one can be used
 - HDMI is best option as carries video + sound.
- **3.5mm audio out jack**
- **2 x USB 2.0** sockets
- Powered by a **5V supply** into a microUSB socket