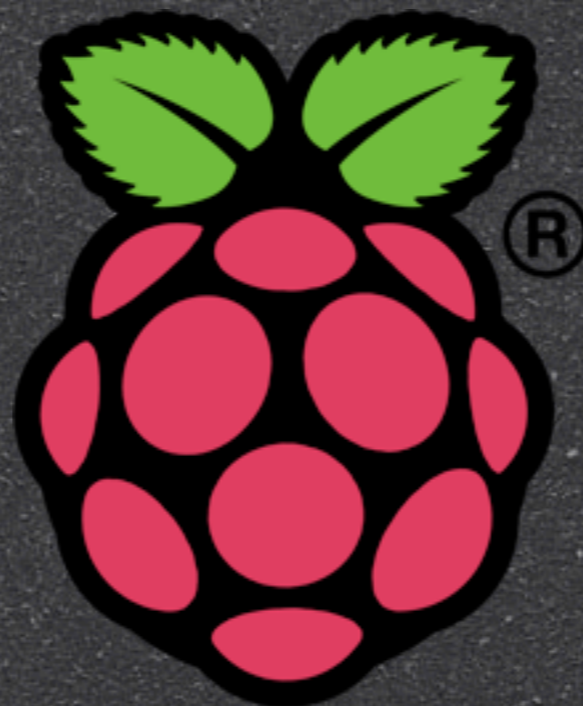


Getting Started with Raspberry Pi

Thomas Tongue - 1/12/13



The Plan

- Creating the Boot SD Card (Raspbian)
- Hardware Setup
- First Boot & Setup
- Navigating the Raspbian Distribution
- Adding New Software & Updates
- Remote Access / Headless Operation
- Introduction to Scratch
- GPIO Access

Creating the Boot SD Card

- Download the recommended OS (Raspbian)
Go to:

<http://downloads.raspberrypi.org/images/raspbian/>

- Select the most recent distribution directory
- Download the .zip file, and unzip.
You'll end up with a .img file.

Creating the Boot SD Card (Windows)

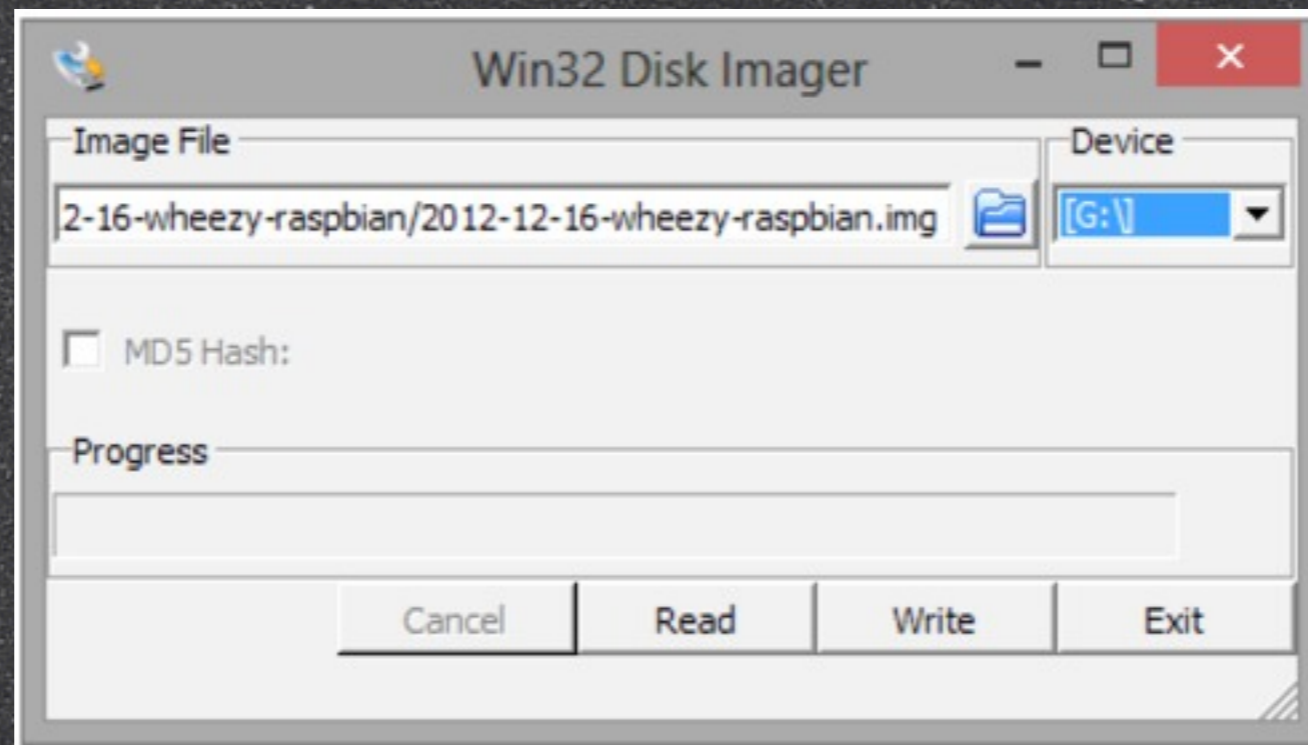
- Download win32diskimager-binary.zip (currently version 0.6) from:

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

- Unzip the downloaded file
- You now have a directory called win32diskimager-binary

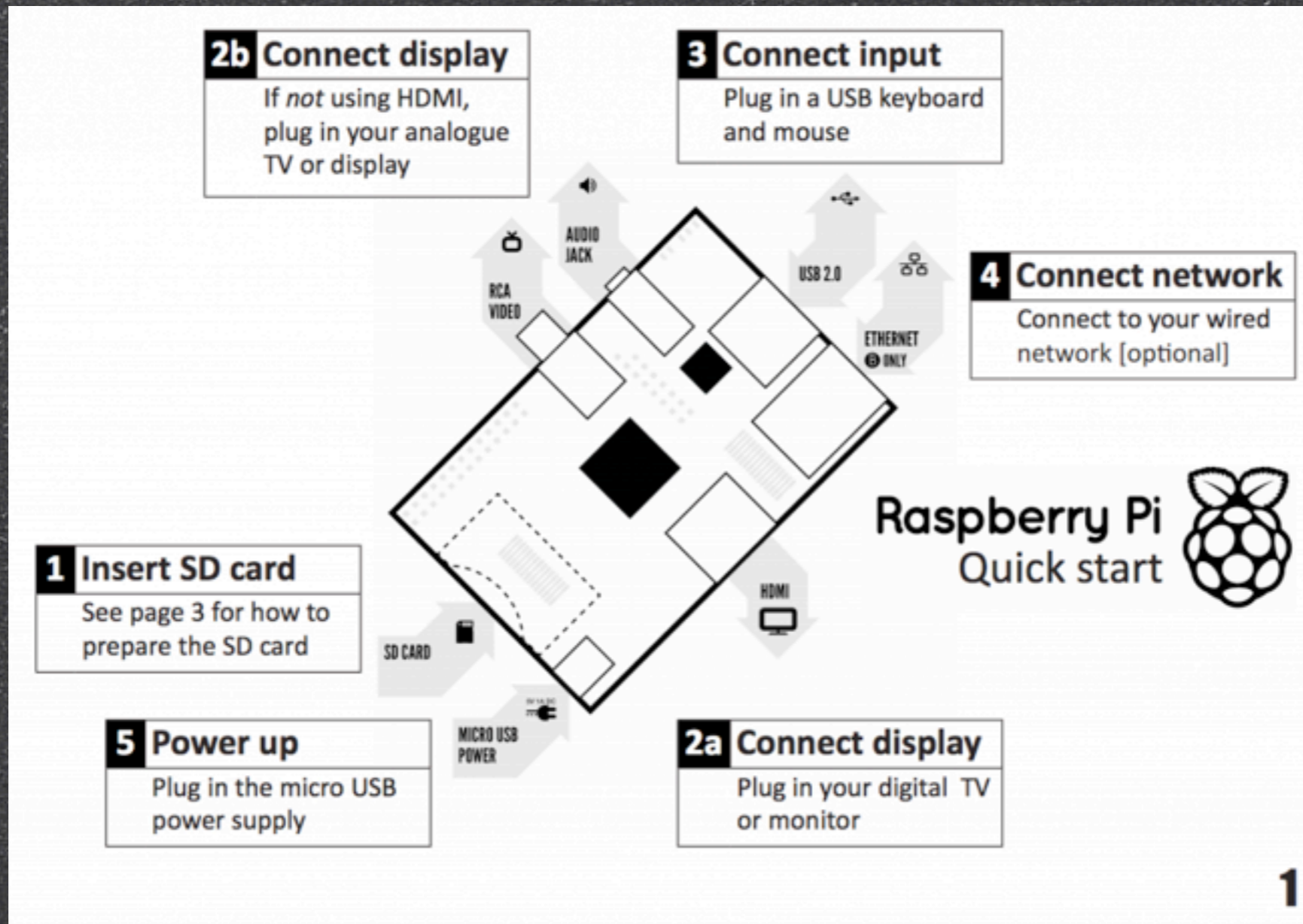
Creating the Boot SD Card (Windows)

- Insert your SD card
- Run Win32DiskImager.exe as Administrator (right click)



- Select the image & device, click Write. Wait....

Hardware Setup



First Boot

- Rasp-config:
 - Expand the root partition to fill SD card
 - Configure Keyboard - (English US)
 - Configure Locale en_US.UTF-8
 - Set timezone
 - Turn on SSH
 - Set the boot preference to start X Windows
 - Optional if you're going to use VNC
 - Required if you're using a Kbd / Mouse / Monitor
 - Finish & Reboot when prompted.
 - System will restart and then it will resize the root partition. Wait...

Getting Networked

- 📌 Plug in the network cable
- 📌 On the console, login and find your IP address using `ifconfig`
- 📌 Write it down!

Installing VNC

- Remote desktop access needs VNC
- Installation: SSH or login to the Pi and run:

```
sudo su
```

```
apt-get install tightvncserver
```

```
wget http://thomastongue.com/Pi/vncboot
```

```
mv vncboot /etc/init.d/vncboot
```

```
chmod 755 /etc/init.d/vncboot
```

```
update-rc.d vncboot defaults
```

```
exit
```


/etc/init.d/vncboot

```
### BEGIN INIT INFO
# Provides: vncboot
# Required-Start: $remote_fs $syslog
# Required-Stop: $remote_fs $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start VNC Server at boot time
# Description: Start VNC Server at boot time.
### END INIT INFO

#!/bin/sh
# /etc/init.d/vncboot

VNCUSER='pi'
eval cd ~$VNCUSER

case "$1" in
  start)
    echo "Starting VNC Server for user $VNCUSER"
    #Insert your favoured settings for a VNC session
    su $VNCUSER -c '/usr/bin/vncserver :0 -geometry 1280x800 -depth 16 -pixelformat rgb565'
    ;;

  stop)
    echo "Stopping VNC Server"
    pkill Xtightvnc
    ;;

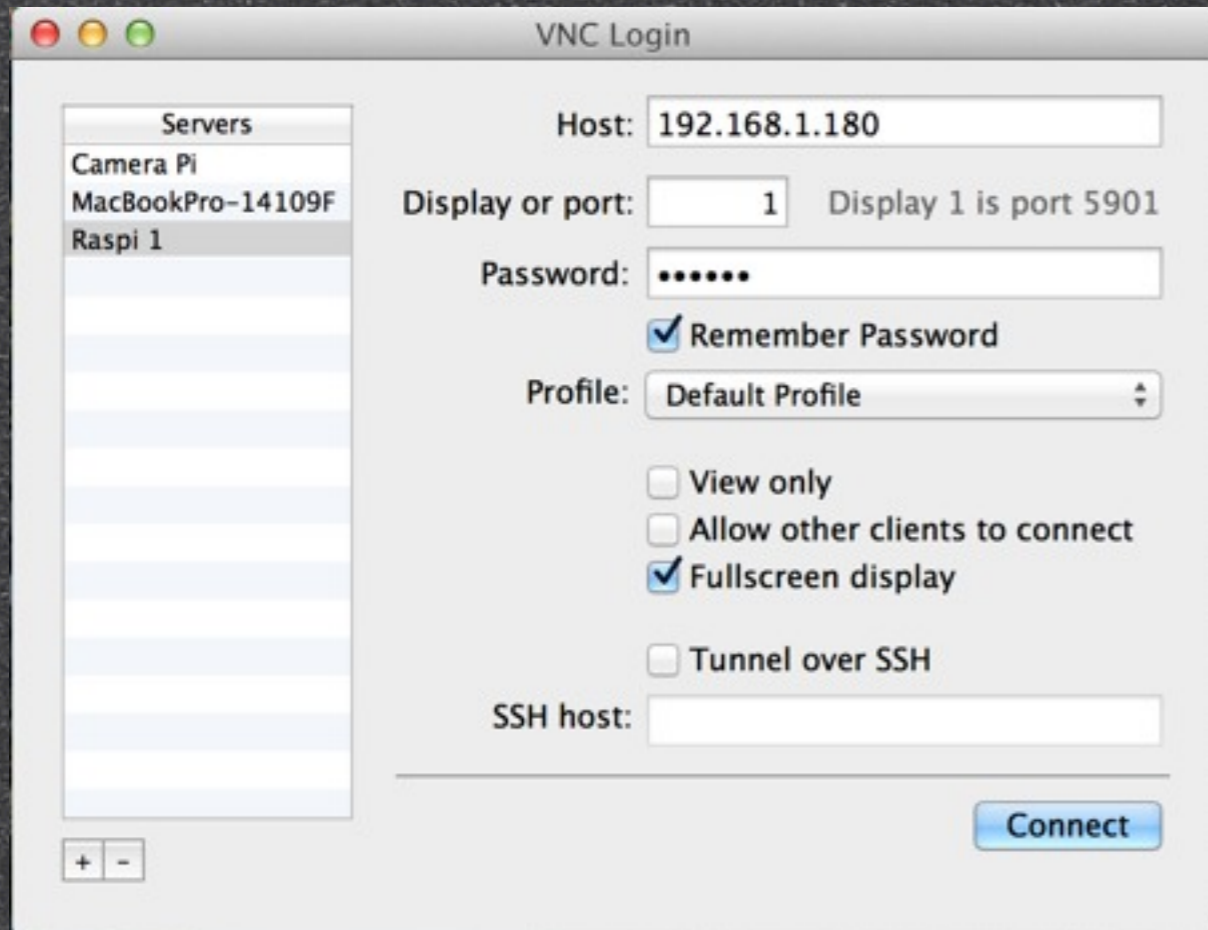
  *)
    echo "Usage: /etc/init.d/vncboot {start|stop}"
    exit 1
    ;;
esac

exit 0
```


Connecting with VNC

Chicken of the VNC
(Mac)

TightVNC (PC)



Navigating Raspbian

- Desktop similar to Windows 7
- Desktop icons run programs directly
- “Start” Menu in lower left
- Virtual desktops
- CPU Meter
- Power / Shutdown / Logout lower right
- File Manager
- Network / WiFi Setup

Adding New Software & Updates

- Open LXTerminal, use apt-get:

- To update the available package list:

- `sudo apt-get update`

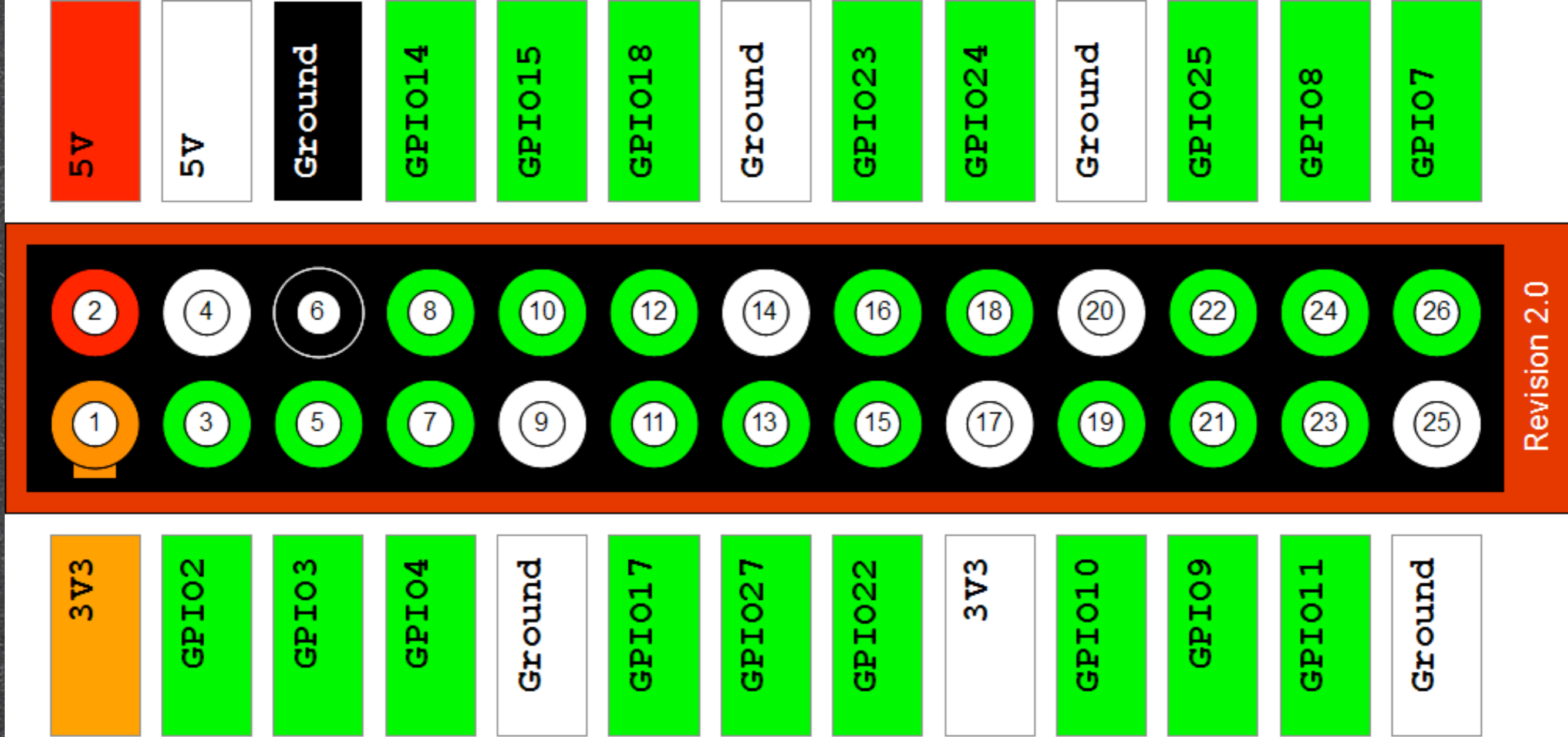
- To upgrade all existing packages:

- `sudo apt-get upgrade`

- To install new packages:

- `sudo apt-get install [package list]`

GPIO



Pin Number	Pin Name Rev1	Pin Name Rev2	Hardware Notes	Alt 0 Function	Other Alternative Functions
P1-02	5V0	5V0	Supply through input poly fuse		
P1-04	5V0	5V0	Supply through input poly fuse		
P1-06	GND	GND			
P1-08	GPIO 14	GPIO 14	Boot to Alt 0 ->	UART0_TXD	ALT5 = UART1_TXD
P1-10	GPIO 15	GPIO 15	Boot to Alt 0 ->	UART0_RXD	ALT5 = UART1_RXD
P1-12	GPIO 18	GPIO 18			ALT4 SPI1_CE0_N ALT5 = PWM0
P1-14	GND	GND			
P1-16	GPIO23	GPIO23			ALT3 = SD1_CMD ALT4 = ARM_RTCK
P1-18	GPIO24	GPIO24			ALT3 = SD1_DATA0 ALT4 = ARM_TDO
P1-20	GND	GND			
P1-22	GPIO25	GPIO25			ALT4 = ARM_TCK
P1-24	GPIO08	GPIO08		SPI0_CE0_N	
P1-26	GPIO07	GPIO07		SPI0_CE1_N	

Header Pinout, bottom row:

Pin Number	Pin Name Rev1	Pin Name Rev2	Hardware Notes	Alt 0 Function	Other Alternative Functions
P1-01	3.3 V	3.3 V	50 mA max (01 & 17)		
P1-03	GPIO 0	GPIO 2	1K8 pull up resistor	I2C0_SDA	I2C0_SDA / I2C1_SDA
P1-05	GPIO 1	GPIO 3	1K8 pull up resistor	I2C0_SCL	I2C0_SCL / I2C1_SCL
P1-07	GPIO 4	GPIO 4			GPCLK0
P1-09	GND	GND			
P1-11	GPIO17	GPIO17			ALT3 = UART0_RTS, ALT5 = UART1_RTS
P1-13	GPIO21	GPIO27		PCM_DIN	ALT5 = GPCLK1
P1-15	GPIO22	GPIO22			ALT3 = SD1_CLK ALT4 = ARM_TRST
P1-17	3.3 V	3.3 V	50 mA max (01 & 17)		
P1-19	GPIO10	GPIO10		SPI0_MOSI	
P1-21	GPIO9	GPIO9		SPI0_MISO	
P1-23	GPIO11	GPIO11		SPI0_SCLK	
P1-25	GND	GND			

Wiring the GPIO demo

- Output demo:

- Wire GPIO 25 to long lead of LED

- Wire Short lead of LED to GND

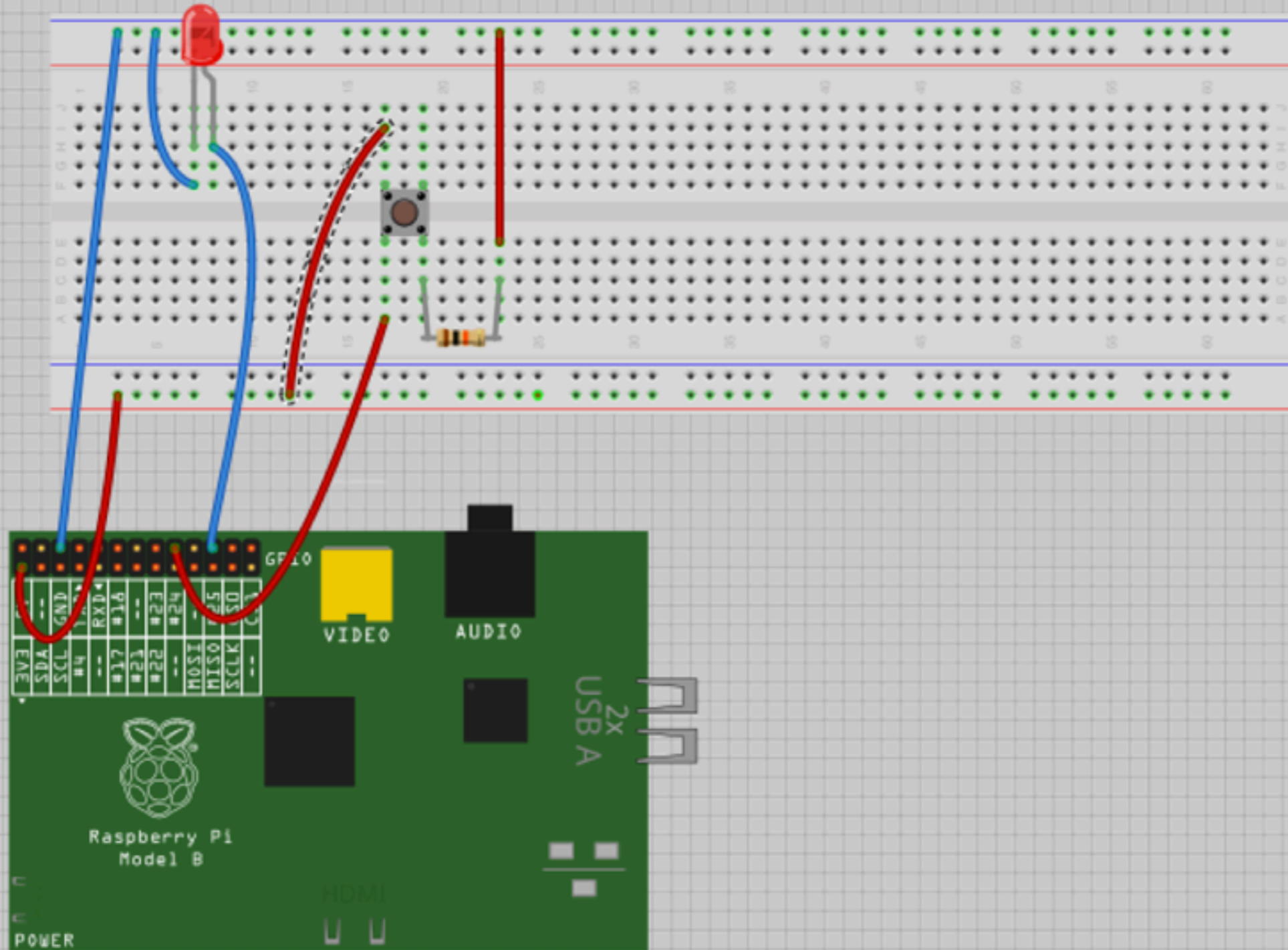
- Input demo:

- Wire GPIO 24 to one side of a switch

- Wire GPIO 24 to 10k resistor to ground

- Wire 3.3V from Pi to other side of switch

Wiring the GPIO Demo



GPIO Output from shell

- From the BASH command line:
 - `sudo su`
 - `echo 25 > /sys/class/gpio/export`
 - `cd /sys/class/gpio/gpio25`
 - `echo out > direction`
 - `echo 1 > value`
 - `echo 0 > value`

GPIO Input from shell

- 📌 From the BASH command line:
 - 📌 `sudo su`
 - 📌 `echo 24 > /sys/class/gpio/export`
 - 📌 `cd /sys/class/gpio/gpio24`
 - 📌 `echo in > direction`
 - 📌 `cat value`

GPIO in Python

 Install `python-rpi.gpio`:

▶ `sudo apt-get install python-rpi.gpio`

Blinking LED (Python)

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(25, GPIO.OUT)

while True:
    GPIO.output(25, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(25, GPIO.LOW)
    time.sleep(1)
```

Steps:

- Create a file called 'blink.py' with the text above (example: nano blink.py)
- Run the file using: `sudo python blink.py`
(the GPIO pins usually need super-user access)

Reading a button (Python)

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(24, GPIO.IN)

count=0

while True:
    inputValue=GPIO.input(24)
    if (inputValue==True):
        count=count+1
        print("Button pressed " + str(count) + " times.")
    time.sleep(0.01)
```


Improved button reading (Python)

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(24, GPIO.IN)

count=0

while True:
    inputValue=GPIO.input(24)
    if (inputValue==True):
        count=count+1
        print("Button pressed " + str(count) + " times.")
        while (inputValue == True):
            inputValue=GPIO.input(24)
            time.sleep(0.01)
        time.sleep(0.01)
```


GPIO in Scratch

📌 Reference Web Site:

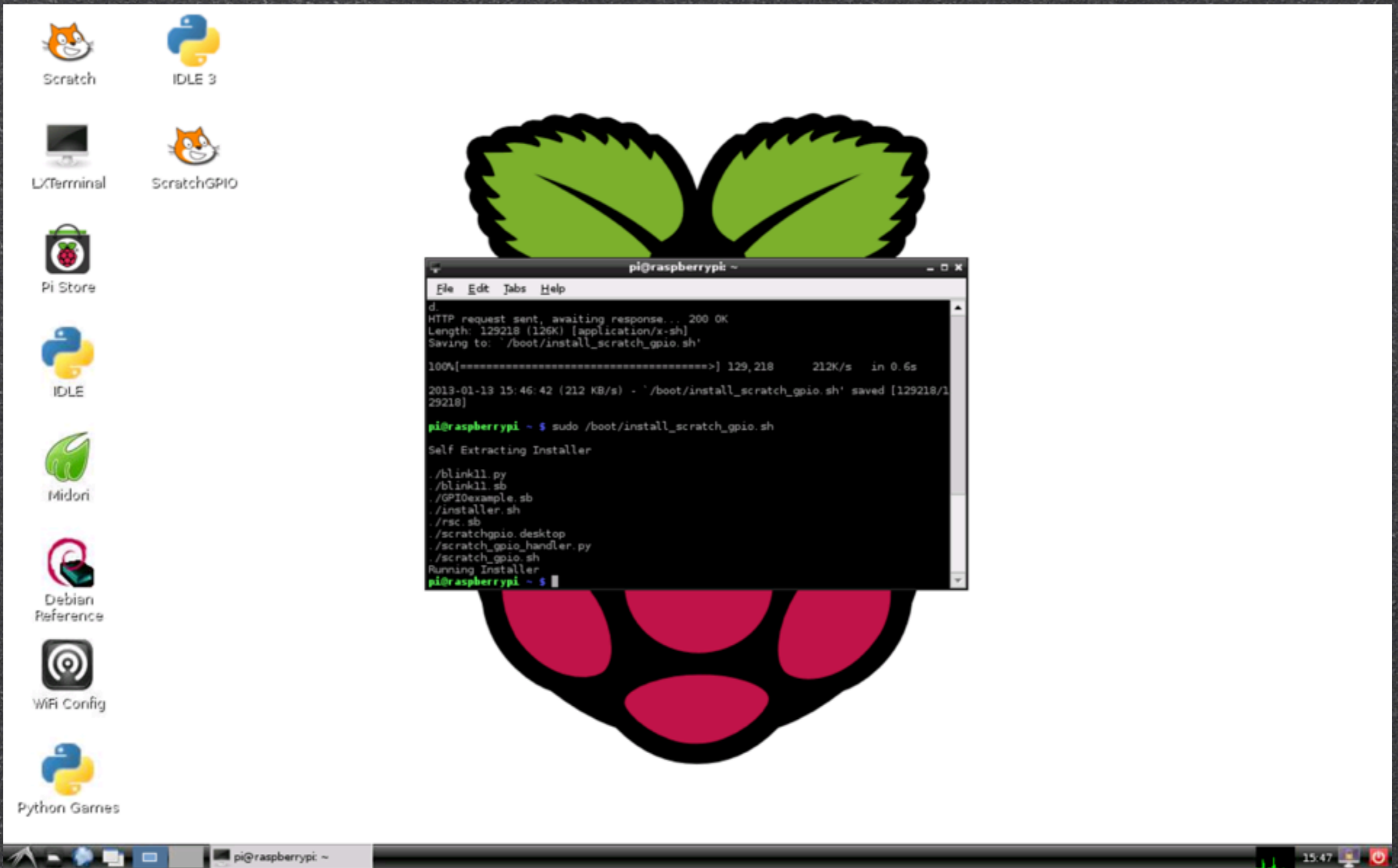
<http://cymplecy.wordpress.com/2012/08/26/scratch-controlling-the-gpio-on-a-raspberrypi/>

📌 Install Scratch-GPIO:

- ▶ `sudo wget http://thomastongue.com/Pi/install_scratch_gpio.sh -O /boot/install_scratch_gpio.sh`
- ▶ `sudo /boot/install_scratch_gpio.sh`

📌 If you are running as user 'pi', an icon for Scratch-GPIO will appear on the desktop

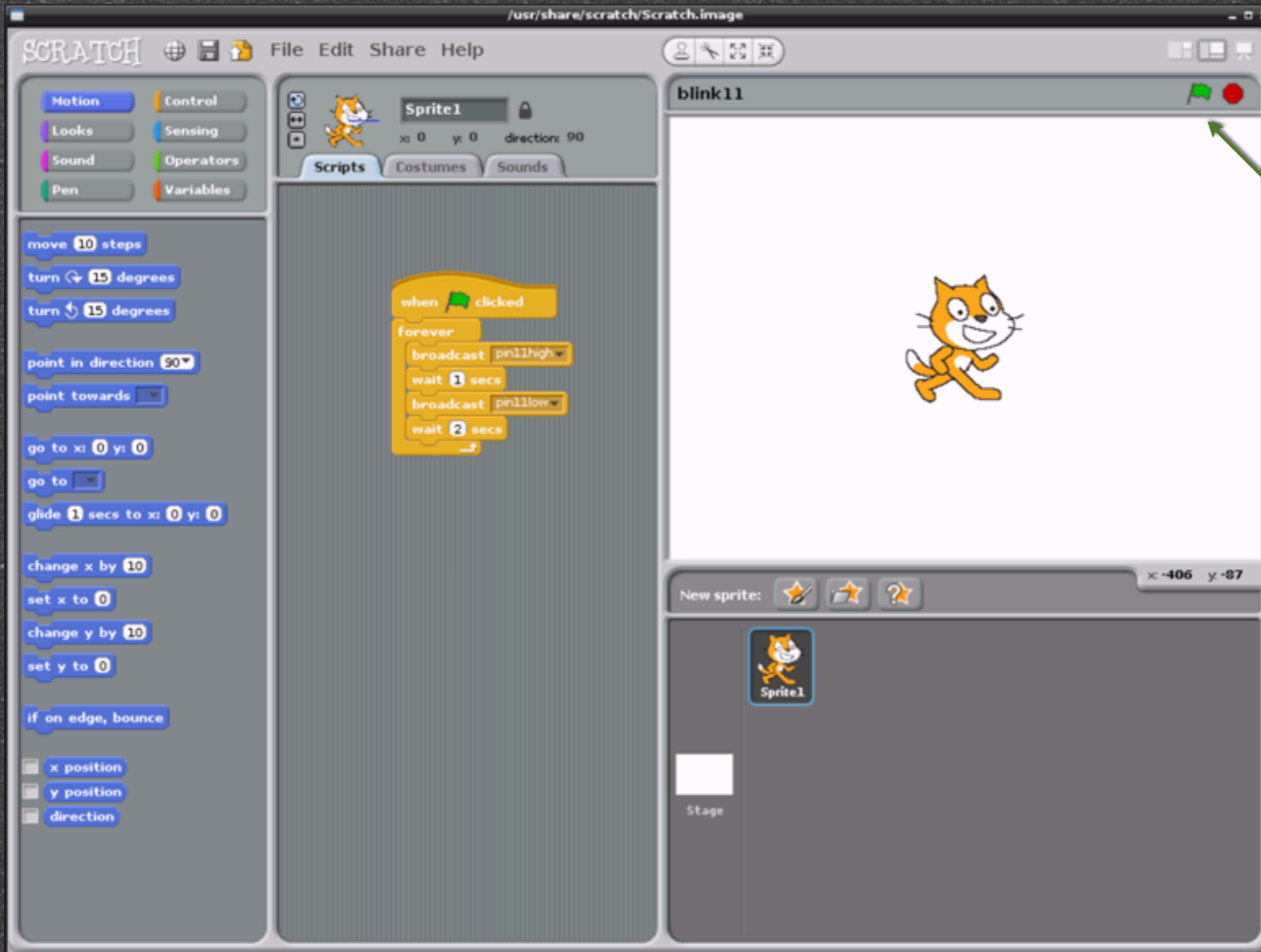
Scratch Demo



Scratch Demo

- 📌 Launch Scratch.GPIO
- 📌 Open file 'Blink11'
- 📌 Rewire your LED to use GPIO 11:
 - 📌 Wire GPIO 11 to long lead of LED
 - 📌 Wire Short lead of LED to GND
- 📌 Click on the Green Flag to run the program!

Scratch Demo



Click here to run the script

What's Next?

- 📌 Is there interest in further Pi workshops? What would you like to see?
- 📌 Visit the Raspberry Pi forum on the Capital District Makerspace web site:

<http://CapitalDistrictMakerspace.org/forum/>

Resources

- 📌 eLinux.org: http://elinux.org/RPi_Hub
- 📌 The MagPi Magazine: www.themagpi.com
- 📌 Raspberry Pi Spy: www.raspberrypi-spy.co.uk/
- 📌 AdaFruit: <http://adafruit.com/category/105>