DONALD DANFORTH
PLANT SCIENCE CENTER

Raspberry Pi

Standard Operating Procedure

975 N Warson Rd, St. Louis,
MO 63132

p. (314) 587-1648
f. (314) 587-1748

bflynn@danforthcenter.org
Dr. Todd Mockler's Lab

Table of Contents

I. Introduction

In this tutorial, we will walk through the basics of getting your Raspberry Pi up and running, how to take pictures along with documentation, syncing them to another machine of your liking, and how to make your Raspberry Pi work with physical input and outputs. There is obviously more to discover, but hopefully this will give you a head start on inspiring new ideas!

Be sure to visit https://www.raspberrypi.org/ for more resources and ideas by other Raspberry Pi users!

A. What You Will Need

Before we dive right in, let's make sure you have all the components necessary to get started:

- Raspberry Pi board
- SD card with a memory of at least 4 GB
- Micro-USB charger
- Key board
- Monitor with HDMI access

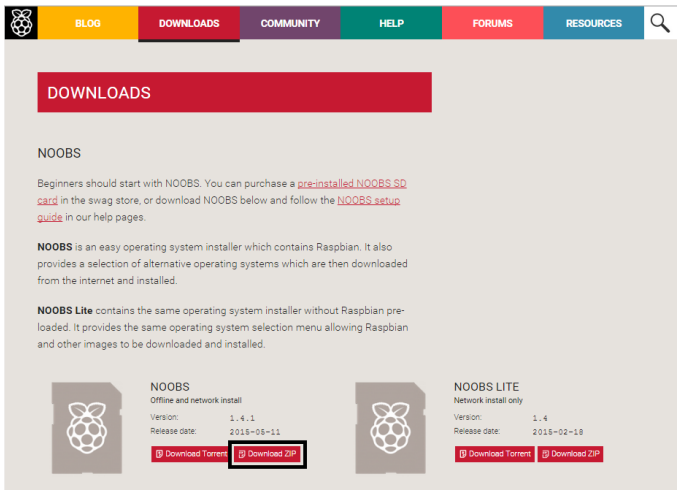
B. Installing the OS

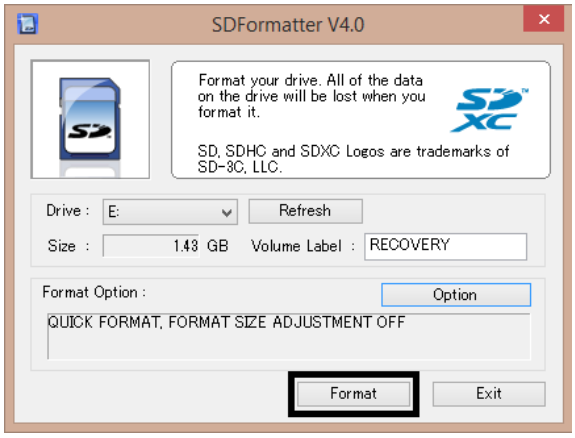
You are about to bring your Raspberry Pi to life. I recommend first time users to use NOOBS (New Out Of Box Software) to install Raspbian. There are a couple ways to go about this:

1. You could purchase a pre-installed NOOBS SD card, or
2. Download the files needed for NOOBS on to your SD card.

If you decide to go the route with downloading NOOBS on to your SD card of choice, you only have a few extra steps to this process.

1. Head to https://www.raspberrypi.org/downloads/
2. Scroll down and click "Download ZIP" button for the NOOBS install





3. While that downloads, put your SD card into your computer and format it using an SD Formatter from this link: https://www.sdcard.org/downloads/index.html
4. Start up the application, and click "Format" with the correct driver selected for your SD card
5. Now take the files from the ZIP file you just downloaded and place or extract them in to your SD card

These should be the same (or similar) files included on a preloaded SD card.

Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted	Comment	CRC
defaults	40 113	39 487	2015-02-17 07:45			D	-		69359A35
os	755 254 412	755 308 753	2015-02-17 09:14			D	-		E32AEB35
bootcode.bin	17 856	8 252	2015-02-18 11:26				-		F262A040
BUILD-DATA	356	254	2015-05-11 09:53				-		76896CFD
INSTRUCTIO...	2 250	973	2015-02-18 11:23				-		8099E248
recovery7.img	2 148 120	2 139 847	2015-02-18 11:24				-		3B1BA0C4
recovery.cm...	54	54	2015-02-18 11:26				-		7DFE3CAC
recovery.elf	554 520	352 838	2015-02-18 11:26				-		F5A2B446
RECOVERY_...	0	0	2015-02-18 11:26				-		00000000
recovery.img	2 100 464	2 091 924	2015-02-18 11:26				-		6DCC6EE9
recovery.rfs	20 238 331	18 183 534	2015-02-18 11:26				-		9163C648
riscos-boot....	9 728	103	2015-02-18 11:23				-		D9BA44A7

To install the OS:

1. Have your Raspberry Pi connected to your keyboard and monitor along with SD card in its slot.
2. Use the micro-USB charger to power the Raspberry Pi.
3. You will be prompted with a screen to select which OS to install on to your Raspberry Pi
4. Select "Raspian", followed by install (use keyboard shortcuts if you don't have a mouse)
5. While installing, you can change the keyboard language (recommended) to English (US)

This process takes approximately 20 minutes, so don't fret and take a break!

Once it is successfully installed, your Raspberry Pi will enter a command line interface, getting ready for you to log in to your Raspberry Pi. It will prompt you with the configuration settings, which you just have to select "<Finish>" using your arrow keys. For future start-ups, you will see a "login as: " which you will type "pi" (your username) followed by your password "raspberry". You can change these if you would like, but this is the initial user installed on to your Pi.

```
login as: pi
pi@raspberrypi's password:
Linux raspberrypi 3.18.11+ #781 PREEMPT Tue Apr 21 18:02:18 BST 2015 armv61

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.
```

II. Linux Command-Line Interface

If you've have had no experience with programming, this new type of computer interaction may be a little awkward, but it gets easier. Here, we will discuss how to communicate with you Pi.

(If you are already familiar with UNIX/Linux command line interface, you can skip to the next section)

A. Permission/Access Denied

When working with the Raspberry Pi, you may or may not have permission to move or edit some files. To switch to the root user where you have the power to do these sort of things, start your commands with `sudo` followed by a space and the desired command.

B. Navigation

The first three commands I am going to introduce are: `pwd` (print your current working directory), `cd` (change your directory), and `ls` (list all the files and subdirectories located in your current directory). The way to think about your file system is like a hierarchical design where your machine contains a root folder that contains subfolders leading to other directories with the same pattern.

- `pwd`

For your Raspberry Pi and other command-line computers, you can only be in one directory at a time. The current directory you are in is called the *working directory*, hence why our command is called "print working directory". To display this, we simply type `pwd` and hit "Enter".

```
pi@raspberrypi ~ $ pwd
/home/pi
```

- `cd [location]`

It would be extremely disorganized without other directories to place files. To change your working directory to another, you can type "`cd location/of/desired/directory`" followed by hitting "Enter" and this will take you to that directory. "location" and "of" are directories inside your current directory where "of" is located inside "location". Here's an example:

```
pi@raspberrypi ~ $ cd location/of/desired/directory/
pi@raspberrypi ~/location/of/desired/directory $ pwd
/home/pi/location/of/desired/directory
```

- `ls`

Unlike a window-explorer interface, we can't see all the files in our working directory just by "being" there. This command will display all the files and folders in our current directory. Give it a try, your screen should display something similar to this:

C. Making and Removing Files or Folders

Since we can move around and look at what our machine contains, the next thing to learn is to create and remove files or folders. Commands in this section include `mkdir`, `rmdir`, `nano`, and `rm`.

- `mkdir [name/location of directory]`

To make a subdirectory in your working directory, use this command followed by the desired name/location of your directory.

- `rmdir [name/location of directory]`

This is almost the same as making a directory, except you have to make sure the directory you are deleting is empty.

- `nano [name/location of file]`

This is a text editor for the Raspberry Pi. Just by entering this command, you open up a blank text file. You can write whatever you want in there (and we will learn how this can help us) and save it to any name/location you want! Just like on computers you have already used, files have extensions that tell the computer what type of file it is what the computer can do with it. The extension for a text file is "`.txt`", and we will learn other types of extensions as well. To use the commands listed at the bottom, hold "control" and press the letter corresponding to the desired command.

- `rm [name/location of file]`

To remove a file, use this followed by the name/location of your file. Be careful, this will permanently delete the file, unlike a Windows computer where the file is just sent to the Recycle Bin.

D. Moving Files

- `cp [desired file] [desired name/location]`

With this, you can take a file, make an exact copy of it, and place that copy anywhere on your machine.

- `mv [desired file] [desired name/location]`

Use this to "move" a file around on your machine. You can move it to another folder or even another name! The contents of the file remain the same.

E. More Commands and Manipulations

If I were to include "everything" there is to know or use in a Linux command environment, you would never get around to using your Raspberry Pi. Many of the commands I have already shown have extensions to make iterative processes easier. You can always type `[your command] -help` which will display the documentation on your command. For more commands, you can use `compgen -c` to list all the commands on your machine. It's pretty long, so you can store it in a text file like so:

```
compgen -c > command_list.txt
```

Feel free to do some self-exploration on the internet of other things you can do in a command-line environment.

III. Establishing a Wi-Fi Connection

It's the 21st century, and Wi-Fi is pretty much everywhere. For this section, you're going to need a Wi-Fi dongle (any will do) that can plug into a USB port. This portion is challenging, so don't get discouraged if this doesn't work your first try.

A. Steps to Connecting

1. You have to edit a configuration file located in the `/etc/wpa_supplicant` directory. The file is called `wpa_supplicant.conf`. Run the command:
   ```
   sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
   ```
   This will open the text file containing a list of viable Wi-Fi connections. Since your Raspberry Pi is brand new, there shouldn't be a list of networks.
2. Edit this file to look like this:

```
  GNU nano 2.2.6          File: wpa_supplicant.conf                Modified

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
        ssid="Danforth_Center"
        psk="DDPSC#3WireBme"
        proto=RSN
        key_mgmt=WPA-PSK
        pairwise=CCMP
        auth_alg=OPEN
}
```

3. Run the command `sudo reboot` to restart your Pi.

4. Log back in to your Pi, and now run the command `ifconfig`. This will display IP address if the Raspberry Pi is connected to the internet.

```
pi@raspberrypi /etc/wpa_supplicant $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:71:cb:f5
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:72 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6288 (6.1 KiB)  TX bytes:6288 (6.1 KiB)

wlan0     Link encap:Ethernet  HWaddr 40:a5:ef:03:17:78
          inet addr:10.17.1.37  Bcast:10.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:60240 errors:0 dropped:13413 overruns:0 frame:0
          TX packets:1186 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10445576 (9.9 MiB)  TX bytes:219663 (214.5 KiB)
```

In the section "wlan0", the "inet addr:" displays my IP address for my Raspberry Pi, indicating that I am connected to the internet.

5. If you don't get an IP address, you may need to update and upgrade your Raspberry Pi. You can get internet access by plugging in an Ethernet cable to the Pi that has a direct internet connection. Once connected, run the commands:

```
sudo apt-get update
sudo apt-get upgrade
```

Once the Pi done upgrading, reboot your machine and try to run the `ifconfig` command again to see if an internet connection is established. If this still doesn't work, try to search the internet for more options.

IV. Remote Access

In this section, you have to be connected to Wi-Fi in order to work. Once established, we can access the command line from our own computer over the Wi-Fi network. This is extremely helpful if we want to transfer files to our computer along with making sure our Pi is running the tasks we asked it to do.

A. Downloads

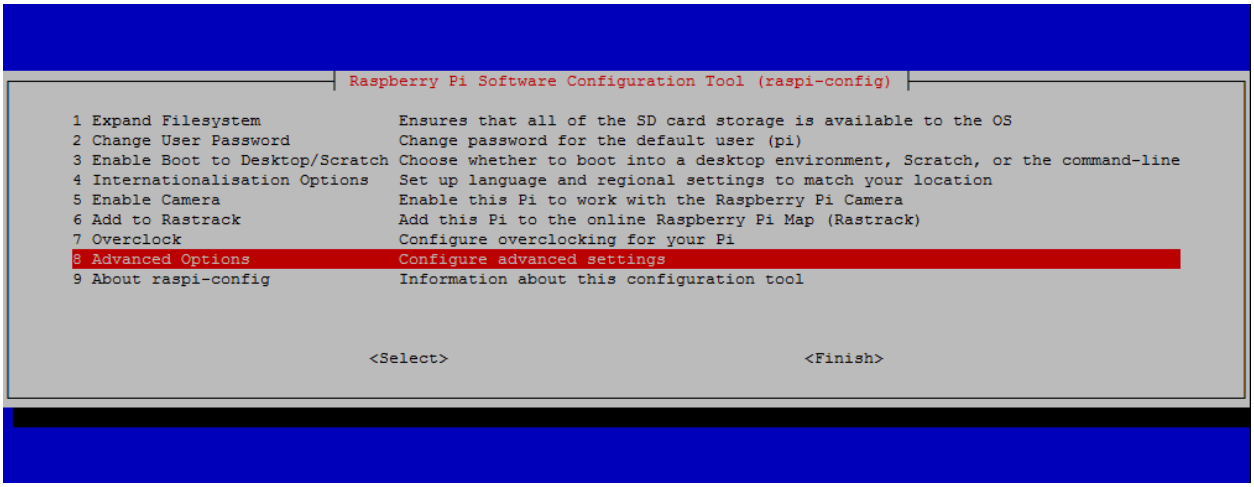
You're going to need a few programs for your computer to be able to remotely log-in to your Raspberry Pi. If you're using a Windows computer, you can download:

- PuTTY: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- WinSCP: http://winscp.net/eng/download.php

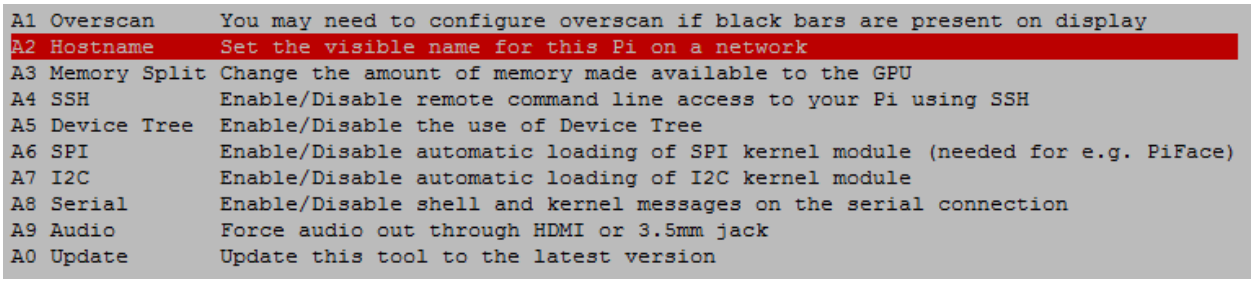
PuTTY will give a command line interface with your Raspberry Pi and WinSCP will give a window-explorer interface so you can drag and drop files with ease.

B. Changing the Hostname

If you have not noticed, when you're are logged on to your machine, each command line starts with `pi@raspberrypi`. This indicates what user is logged onto what machine. Right now, the hostname for your Pi is "raspberrypi", just like every other freshly installed Raspberry Pi. But if you have multiple Pis, how would you distinguish them if they're all named the same? We have to go into the configuration settings to change that name! So type in `sudo raspi-config` to the command line, and then you will be prompted with a blue screen with options.



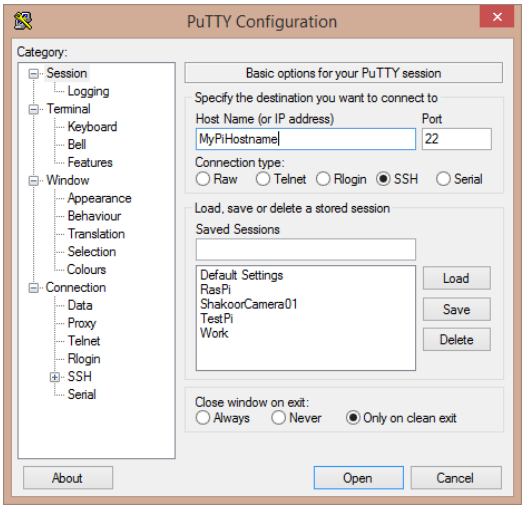
From here, use your arrow keys to scroll down to option 8, Advanced Options, followed by hitting enter.



Select hostname, and proceed to change the hostname to your liking! Once you're done, select "<Ok>", and then select finish. You'll then have to reboot the system to see the change in the host name.

C. Remote Access

Now your Raspberry Pi has a unique hostname, you can now log on to that specific Pi straight from your computer. With PuTTY or any similar SSH client, you want to locate your machine by its Host Name over Port 22 (for safety). Just like this example, you would click open once you have your hostname typed in:



This will open a window terminal that will act completely the same as before when we were typing commands directly into the Raspberry Pi. Now you don't need an extra keyboard/monitor for your Pi!

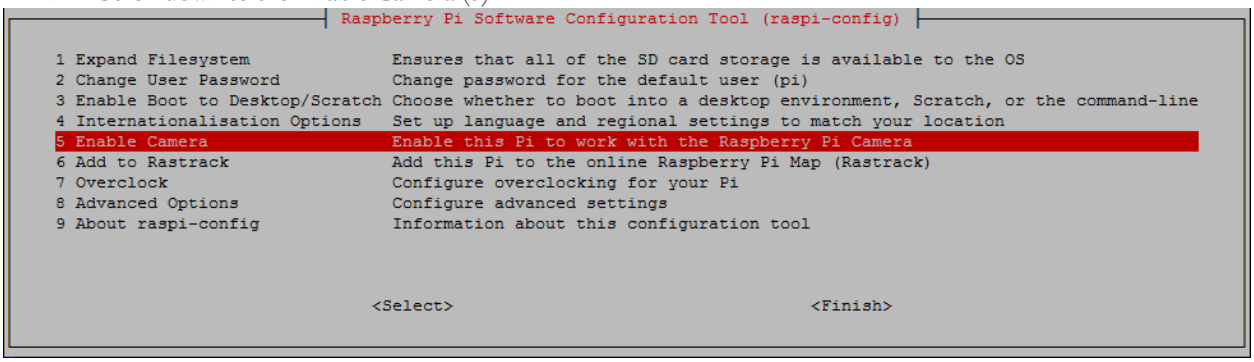
V. Raspberry Pi Camera

In this section, we will cover how to set up your Raspberry Pi to be able to capture images and video! The next component you need to make this work is the camera module specially made for the Raspberry Pi. For reference on how to plug this into your Pi, check out https://www.raspberrypi.org/help/camera-module-setup/.

A. Enabling the Camera Module

Before we're are able to take pictures, we have to tell our Pi to enable the camera.

1. On the command line, enter in `sudo raspi-config` to pull up the configuration settings.
2. Scroll down to the Enable Camera (5).



3. Select Enable

4. Select Finish and then reboot your Pi.

B. Command Line Imaging

First, we'll show how to take a command line argument to capture a still photograph from the Raspberry Pi. Have your camera module pointed at something you would like to capture, and then type the command

```
raspistill –o cam.jpg
```

Hold your camera module still while the image is taken place. If you list the contents n your directory, you should see a "cam.jpg". To view this image to make sure it is the correct one, open WinSCP to your working directory. Right click on the file and select "open" file. This will open the image on your desktop. Alternatively, if you are still using the monitor/keyboard connected to the Raspberry Pi, you can still view the image under the graphical user interface (GUI). To start the GUI, use the command `startx` to being your session. If you have a mouse, you can find your file and open it up like a normal image file.

For the full list of options and documentations, visit
https://www.raspberrypi.org/documentation/raspbian/applications/camera.md

Now to make our image names "time dependent", we would have to use variables that can hold these values. We could then sue a "script" to write multiple commands for temporary place holders. Here are a few example scripts:

C. Bash Scripting

On your command line, let's open a blank text file using the command `sudo nano`. Inside you blank text file, fill it with the following code:

```
#!/bin/bash

DATE=$(date +"%Y-%m-%d_%H:%M")
raspistill –o /home/pi/image_$DATE.jpg
```

Let's break this apart and explain what each part does:

1. `#!/bin/bash`

This is a convention to determine which kind of interpreter to run. This program is a "bash" script, where bash interprets the code in our file.

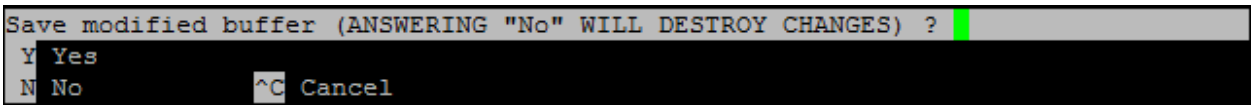
2. `DATE=$(date +"%Y-%m-%d_%H:%M")`

If you were to use the command `date` you would get an output from your screen like so:
`Fri Jun 19 20:30:19 UTC 2015`
Our line of code takes the components from this output, and formats it into a variable we would want to use.

3. `raspistill –o /home/pi/image_$DATE.jpg`

Just like before, we're taking an image with this command, and storing it in the location `/home/pi/image_$DATE.jpg`. Our file names would be then stored in our home directory with a file name like "`image_2015-06-19_20:13.jpg`". You can edit your storage location so that your Pi is nice and organized.

Now to save the text file, press Ctrl+x. At the bottom of your terminal, you will see:



Press "y", write the name of your file followed by the file extension (`.sh`), and hit Enter again. Now your file should be in your working directory.

Change the mode on the file so it can be executed:

```
chmod 755 /home/pi/location/name_of_file.sh
```

In order to execute a bash script on the command line, you must write "`./`" before your file name. If you're not in your home directory, but your executable file is, you can execute it like so: `/home/pi/./bashscript.sh`

D. Python Scripting

Make sure your Raspberry Pi is up-to-date:

```
sudo apt-get update
sudo apt-get upgrade
```

Once this is finished, we will need to install the python-picamera library, like so:

```
sudo apt-get install python=picamera
```

We'll start by opening a blank text file:

```
sudo nano
```

With python, indentation is important, make sure your file looks like so:

```
#!/usr/bin/python

import picamera
import time
import datetime

with picamera.Picamera() as camera:
      camera.start_preview()
      time.sleep(5)
      camera.capture('/home/pi/location/image_' +
str(datetime.datetime.now()) + '.jpg')
      camera.stop_preview()
```

(`camera.capture` is one line of code)

Save this file with the extension "`.py`" and change the mode on the file to being executable

```
sudo chmod 755 /home/pi/location/pythonscript.py
```

In order to execute a python file, you have to write "`python`" like so:

```
python /home/pi/location/pythonscript.py
```

VI. Cron Jobs

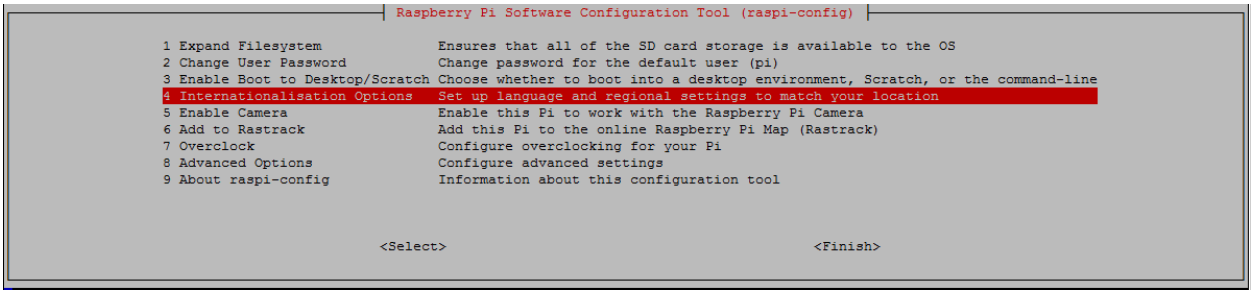
Let's say your job needs you to run a command at a certain time, daily, or even hourly. It would seem impossible, you would have to be constantly logged on to your Raspberry Pi ready to send in the command. With every Linux system, you are able to schedule commands with CRON. There is a file called a "crontab" that you can edit with a few simple lines to make tasks like these easier.

A. Setting your Date and Time

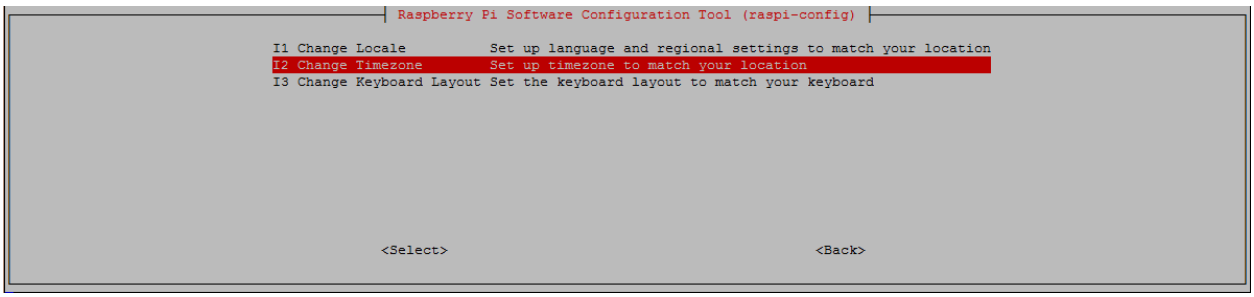
Before you start scheduling times on your Pi, you have to make sure it's time is synced to where you are located. Make sure your Raspberry Pi is connected to the internet, otherwise you can't sync to the global clock. Now, let's access our configuration:

```
sudo raspi-config
```

From here, we want to access the Internationalization Options:



Select Change Timezone:



You will then continue to select your location and time zone.

B. Crontab

There are only 3 options you need to know for viewing and editing your crontab:

- `crontab -e` This will allow you to edit your crontab
- `crontab -r` This will wipe your crontab clean and delete everythin you had in it.
- `crontab -l` This will just print the contents of your crontab

You will type these directly into the command line.

C. What to Write

When you are editing your crontab, every line in this file can be used as a scheduled task. To break it down:

```
* * * * *  command to execute
| | | | |
| | | | |
| | | | └──── day of week (0 - 6) (0 to 6 are Sunday to Saturday)
| | | └─────── month (1 - 12)
| | └──────────── day of month (1 - 31)
| └─────────────── hour (0 - 23)
└──────────────────── min (0 - 59)
```

You'll add your own task with the same format to the end of your crontab text file. For more reference, visit http://crontab.org/

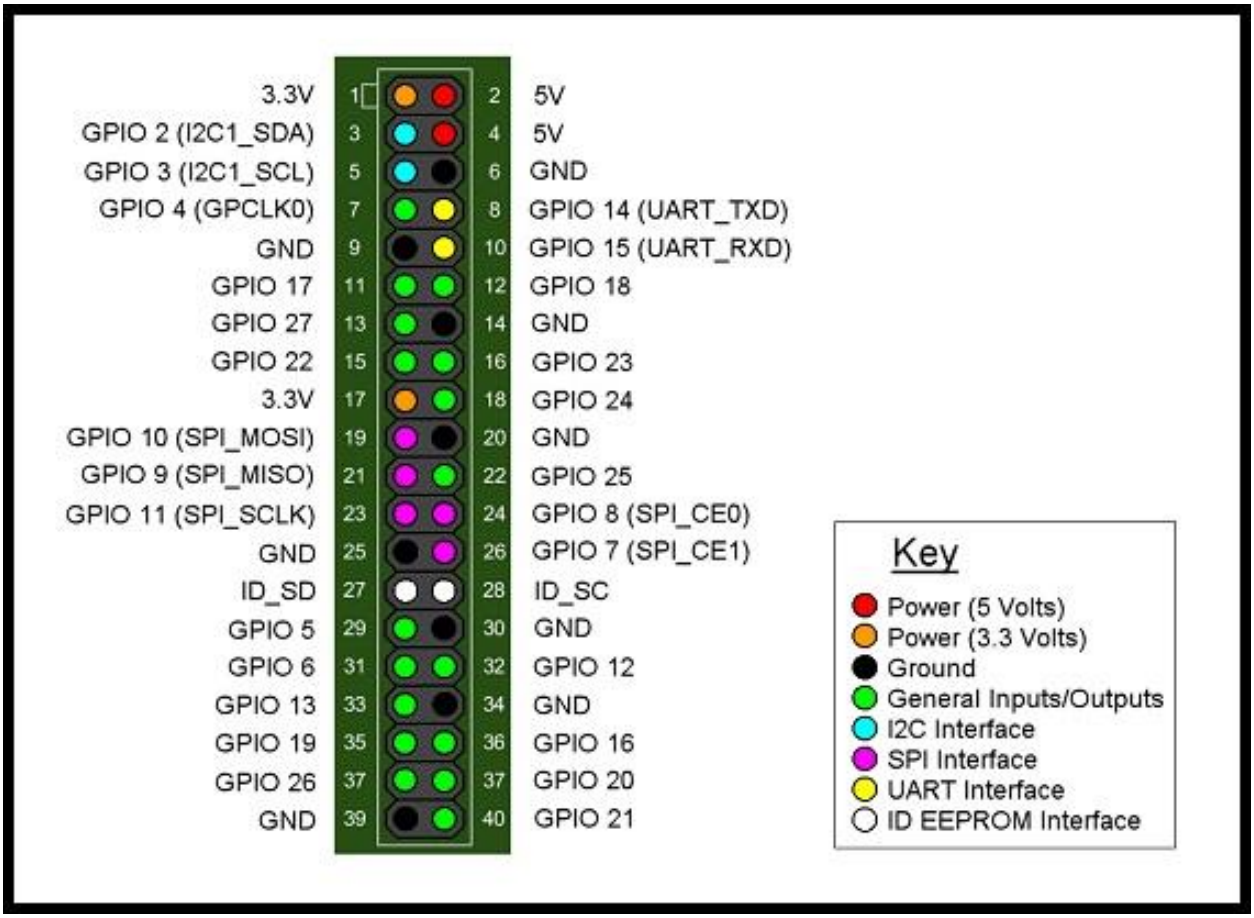
VII. Physical Computing

For some of your projects, you may want the Raspberry Pi act as a switch in a circuit of some sort. Each Raspberry Pi has a set of General Purpose Input Output (GPIO) pins (26 on the Model B and earlier; 40 on the Model B+ and later) that can be used in such a manner. You can implement these in your scripts to make them apart of your commands. For instructions on installing, go to http://www.raspberrypi-spy.co.uk/2012/05/install-rpi-gpio-python-library/ for the Python library tutorial. Another resource (if you're not a fan of Python) is http://wiringpi.com/ .

A. Easy Project Ideas

- http://computers.tutsplus.com/tutorials/controlling-dc-motors-using-python-with-a-raspberry-pi--cms-20051
- https://projects.drogon.net/raspberry-pi/gpio-examples/tux-crossing/gpio-examples-1-a-single-led/
- http://www.treehugger.com/slideshows/gadgets/20-awesome-projects-raspberry-pi-microcomputers/page/6/#slide-top

B. Pin Layout



The top pins are on the SD card side.