

Instructions for initial boot of Raspberry Pi

Materials:

Micro USB power cable (One that plugs into a wall socket)
HDMI Display / Monitor
Keyboard
Mouse
HDMI Cable
Micro SD Card
Micro SD Card reader (Or a micro SD card adapter and a regular SD card reader)

Placing the OS on the Micro SD card

1. Download NOOBS from <https://www.raspberrypi.org/downloads/noobs/> (The download is around 1 GB, so plan accordingly)
2. Download the program “SD Formatter” 4.0 from https://www.sdcard.org/downloads/formatter_4/
3. Unzip the ZIP folder that was downloaded, and navigate into the unzipped directory, then run “setup”.
4. Pick some directory for the program to be installed into (The default works fine)
5. Plug in the Micro SD card into your computer, using whatever method works with your computer.
6. Open MyComputer to check the letter identifier of the SD Card (for instance, [E://](#))
7. Run SDFormatter
8. The GUI of SDFormatter will have a Drive specified, MAKE SURE that the drive letter is the same as the SD card. If not, select the correct drive identifier.
9. Next to volume label, type in a short string, such as the team name, to differentiate the Pis.
10. Click Format to format the SD card.
11. When the NOOBS file is downloaded, unzip all of the files, and then drag and drop the unzipped files onto the SD card.

Installing the OS on the Raspberry Pi

1. Eject the SD card, and insert it into the Raspberry Pi.
2. Connect the PI to a monitor via HDMI port, plug in the power cable, as well as the keyboard and mouse. The Pi should be able to accept keyboard and mouse input without any extra drivers installed.

3. Switch the monitor onto the HDMI input
4. After a minute or two, you should see a picture of a Raspberry, or a window listing different operating systems. Select “Raspberian”
5. Below the window asking which operating system you want, look at the bottom of the screen, and there should be a little flag symbol at the bottom, next to a drop-down menu. Click the menu, and select English(US). (If you don't do this, your keyboard will not work correctly)
6. Click the Install tab on the top bar. There will be a dialog window that pops up asking if you want to install the OS. Click “yes”.
7. The OS will start to install, but this will take a while.
8. After the installation, a dialog window will pop up, press 'ok', then the Pi will jump to a text interface, and go through a lot of commands on it's own. Eventually, an image of a raspberry will pop up, and the Pi has booted correctly.
9. Shutdown the Pi.
10. Insert the Wifi dongle, and then restart the pi.
11. Connect to the internet using the GUI on the upper right corner of the screen.
12. LibreOffice and Python should already installed.

Enabling The Camera

1. Open a console window.
2. Type “sudo raspi-config”
3. In a second, you should see a screen of text appear. Use the arrow keys to go down to “Enable Camera” There should be a screen that asks if you want the camera enabled, use the arrow keys to the “confirm” option.

Changing Keyboard Layout

1. When starting the pi, you may notice that the keys you press are not the ones that are being displayed in the terminal, this is because the Pi's starting configuration is for Great Britain, not the US. To change this, follow this guide:

<http://helloraspberrypi.blogspot.com/2014/01/change-keyboard-layout-of-raspberry-pi.html>

Installing OpenCV

Unlike a lot of the other instructions, there is a particular order you **MUST FOLLOW**. Most of this was taken from <http://www.pyimagesearch.com/2015/07/27/installing-opencv-3-0-for-both-python-2-7-and-python-3-on-your-raspberry-pi-2/> . HOWEVER, there are issues with the order of dependencies.

1. Open the console, you will use this for the entire installation.
2. Type in the following instructions:

```
sudo apt-get update
sudo apt-get upgrade
sudo rpi-update
sudo apt-get install build-essential git cmake pkg-config
```

3. This next group of libraries **MUST BE INSTALLED IN THIS ORDER**. The tutorial online has them listed on a single line, which will not work due to the ordering. Use each command, one by one.

```
sudo apt-get install libjpeg-dev
sudo apt-get install libtiff5-dev
sudo apt-get install libjpeg8-dev
```

4. Now, we can follow the online tutorial to the letter. If you don't want to go to the website, I will paste the instructions here, but the tutorial online has far greater detail.

```
sudo apt-get install libjasper-dev libpng12-dev
sudo apt-get install libgtk2.0-dev
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
```

```
sudo apt-get install libgtk2.0-dev
sudo apt-get install libatlas-base-dev gfortran
```

```
cd ~
git clone https://github.com/Itseez/opencv.git
cd opencv
git checkout 3.0.0
```

```
cd ~
git clone https://github.com/Itseez/opencv_contrib.git
cd opencv_contrib
git checkout 3.0.0
sudo apt-get install python2.7-dev
wget https://bootstrap.pypa.io/get-pip.py
sudo python get-pip.py
sudo pip install virtualenv virtualenvwrapper
sudo rm -rf ~/.cache/pip
```

5. With your favorite text editor (Vi, emacs, nano, etc) open ~/.profile

6. At the end of the file, type the following lines, then save and quit the editor.

```
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python2.7
export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```

7. Once you are back on the command line, type in the following commands:

```
source ~/.profile
mkvirtualenv cv3
```

8. From now on in the instructions, you must be working in the cv3 virtual environment. At the beginning of the current line in the console, there must be a several character string “(cv3)”. If there is not, then none of the following commands will work. If you have to log off the Pi, or if the power goes off, or anything similar, you must type in the command

```
workon cv3
```

If you do not, the OpenCV library will not compile correctly.

9. Now that you are working in the virtual environment, cv3, you should use the following commands:

```
pip install numpy
sudo rm -rf ~/.cache/pip/
pip install numpy
cd ~/opencv
mkdir build
cd build
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D
INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D
OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules -D BUILD_EXAMPLES=ON .
```

10. After the commands above are completed, now we need to compile OpenCv. This will take a huge amount of time. However, if you log-off the Pi or turn it off, you can restart compiling it with the same command, and the compiler will resume where it left off.

Compile the code with the line:

```
make -j4
```

This may actually take a couple tries, when I was compiling my version of OpenCV, the compilation stopped twice for some reason, and I just restarted the command, and it continued to compile. If the percentage of what it has compiled continues to go up (or more libraries are at least compiled), then it is working correctly.

This command may also stop when it is only 90% done. In this situation, continue on to the next command.

11. Type the following commands in:

```
sudo make install
```

```
sudo ldconfig
```

```
cd ~/.virtualenvs/cv3/lib/python2.7/site-packages/
```

12. If the above commands didn't work, you need to find where the compiled opencv library is location. It should be in either `/usr/local/lib/python2.7/site-packages/` or `/usr/lib/python2.7/site-packages`. If necessary, change the first section of the next command to be the location of the installed libraries.

```
ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so
```

13. If everything goes well, OpenCV has been installed onto the Pi. To test this, use the following commands:

```
workon cv3
```

```
python2
```

```
>>> import cv2
```

```
>>> cv2.__version__
```

If you have installed it correctly, the string '3.0.0' should appear.