How-To guide

To make BeagleBone Black to work with C or C++ with OpenCV, following is a guide to install OpenCV and Ubuntu into a image which is used to boot on the BBB. They have 2 main procedures:
1. Make a eMMC card with Ubuntu 14.0 image to boot at BBB
2. Install OpenCV in Ubuntu and Compile C++ with OpenCV

The 2 can be used both in BBB and computer with gcc and g++. They are:

1. Make a eMMC card with Ubuntu 14.0 image to boot at BBB

Step 1: Down load the image 'bone-ubuntu-14.04.1-console-armhf-2014-10-29-2gb.img'

Go to http://beagleboard.org/latest-images
Select'Debian (BeagleBone, BeagleBone Black - 2GB SD) 2014-05-14 - more info - bittorrent - md5: 35877ce21e8ed0eb1bdc6819ad71c317'
Open with 7_zip and extract the .img file.
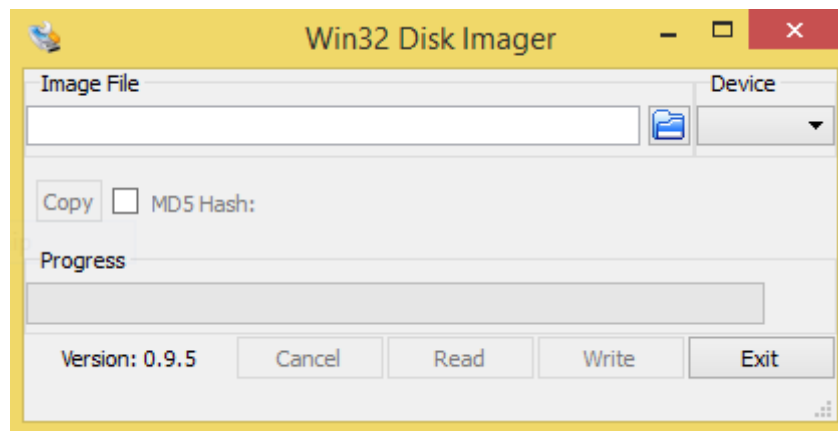
Figure 1. Win32 Disk Imager

Step 2: write the image into the SD card.

Insert micro SD card to computer through USB or SD card
Execute Win32 Disk Imager
Format the micro SD card.
Check the 'Device' to the right drive.
Select the image from Step 1.
Click write.
Exit after write is completed.

Step 3: boot the image file at BBB

insert the micro SD card in your *BeagleBone*.
Press the MMC *Boot* button on BBB, see Figure 2.
Keep the *Boot* button pressed and until you see all LEDs flashing on the BBB.
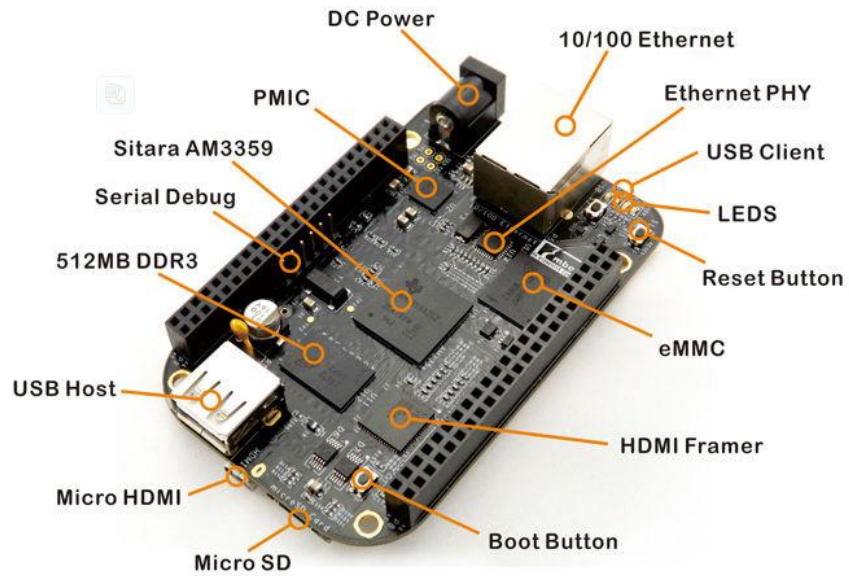Stop to press the boot button and let booting continue.
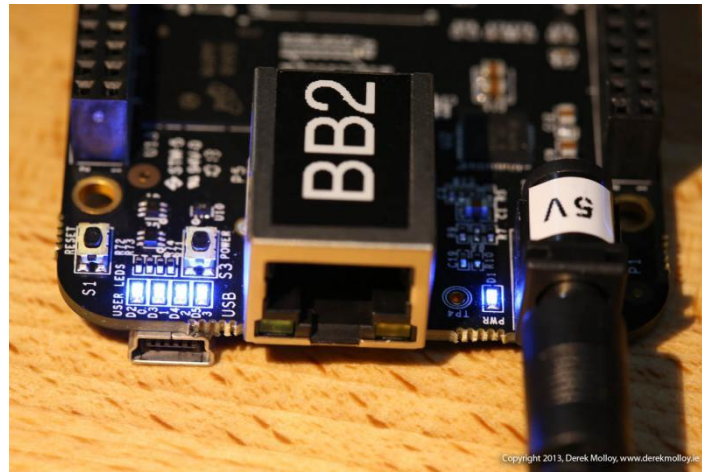See Ubuntu is ready

Figure 2. beagleboard

.

Figure 3. boot success

Step : Login at BBB

Login with ubuntu: type 'ubuntu'.
Password : type 'temppwd'

Troubleshooting:

1,   **Not every boot will be successful. When booting the board, if led lights don't lit up after approximately 10 seconds, it means that the boot had failed.**

      Solution:

          Retry a couple of time until you achieve a successful boot.

2.   **Boot not working after many tries.**

      Solution:

          Configure another boot image and redeploy to the Micro-SD card.

3.   **After selecting advance option using the narcissus tool, no option menu shows up**

      Solution:

          Some browser does not support his feature, so change to IE, which works

2. Install OpenCV in Ubuntu and Compile C++ with OpenCV

Step 1: Make Directory

```
$ mkdir opencv
$ cd opencv
```

Step 2: Remove any pre-installed ffmpeg and x264

```
$ sudo apt-get -qq remove ffmpeg x264 libx264-dev
```

Step 3: Update ubuntu

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

Step 4: Install Dependenices

```
sudo apt-get -qq install libopencv-dev build-essential checkinstall cmake pkg-
config yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev
libdc1394-22-dev libxine-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev
libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-
dev libxvidcore-dev x264 v4l-utils
```

Step 5: Install ffmpeg

```
$ sudo apt-add-repository ppa:samrog131/ppa
$ sudo apt-get update
$ sudo apt-get install ffmpeg-real
$ sudo ln -sf /opt/ffmpeg/bin/ffmpeg /usr/bin/ffmpeg
```

Step 6: Download and extract OpenCV

```
$ wget -O OpenCV-2.4.9.zip http://fossies.org/linux/misc/opencv-2.4.9.zip
$ unzip OpenCV-2.4.9.zip
$ cd opencv-2.4.9
```

Step 7: Begin Build Process

Note that this bit takes time to complete

```
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D
WITH_TBB=ON -D BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D INSTALL_C_EXAMPLES=ON
-D INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON -D WITH_QT=ON -D WITH_OPENGL=ON
..
$ make -j2
$ sudo make install
$ sudo sh -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'
$ sudo ldconfig
```

OpenCV is installed! **[1, 2]**

Step 8: Compile C++ and OpenCV

```
$ g++ `pkg-config --cflags opencv` foo.cpp -o foo `pkg-config --libs opencv`
```

References:

[1] https://solarianprogrammer.com/2014/04/21/opencv-beaglebone-black-ubuntu/
[2] http://www.sysads.co.uk/2014/05/install-opencv-2-4-9-ubuntu-14-04-13-10/