



Beaglebone Black Booting

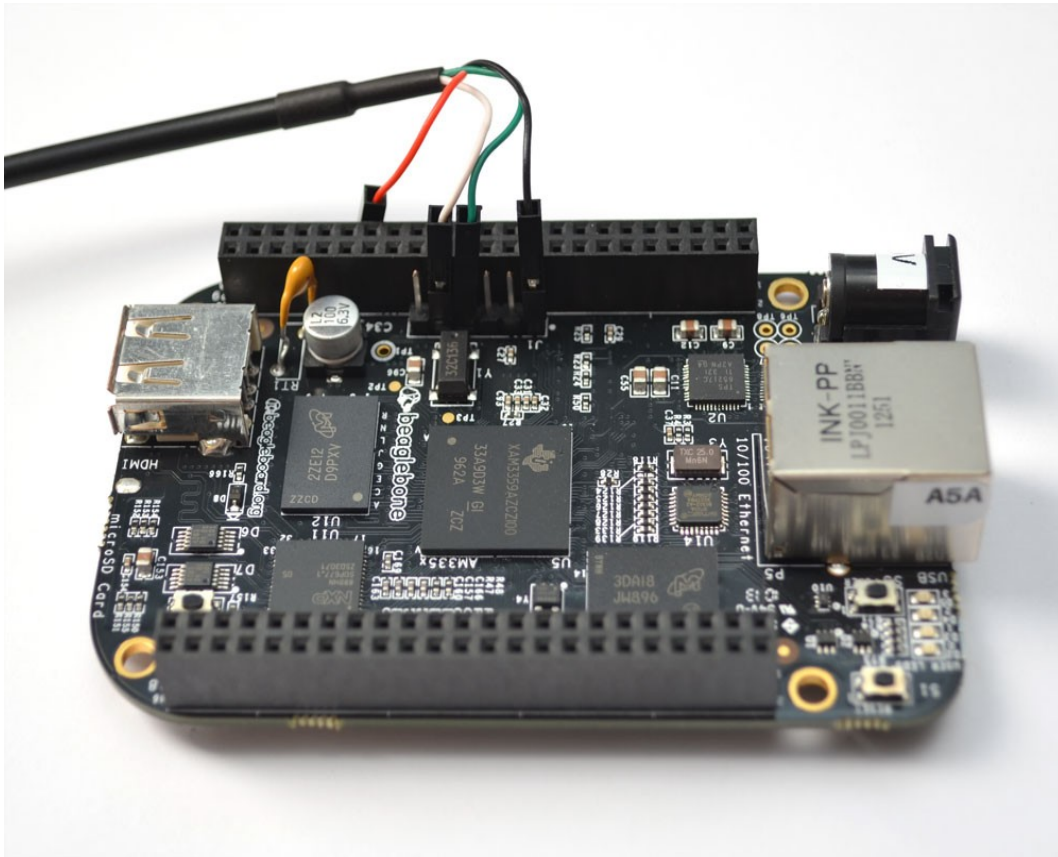
The BeagleBone Black is a full featured, internet enabled development platform that utilizes the low cost Sitara™ AM3358 ARM® Cortex™-A8 processor from Texas Instruments and runs a variety of OS including Debian, Angstrom, Ubuntu and Android. The BeagleBone Black is designed to address the needs of designers, early adopters, and anyone in the Open Source Community interested in a low cost ARM® Cortex™-A8 based processor.

The boot sequence of Beaglebone is eMMC, mmc0, usb. This board is having three boot sources. It has internal 4GB eMMC that contains vendor provided preloaded images. When you power on the board it takes eMMC as the default boot device and boot from there. If you want to alter the boot sequence, the board provides a boot switch, which when hit switches the booting from external mmc, like sd card.

Power and Communication channels:

- * For booting beaglebone board we must need two connections.
- * One is usb power cable for power source.
- * Second: TTL to USB cable for serial communication interface to host.
- * TTL to USB is having four pins red(VCC), black(GND), green(RXD), white(TXD).
- * Connect serial pins to serial port properly as per the fig.below.





Booting from SD Card :

Booting Beaglebone involves these images:

- * **U-boot:** Beaglebone supports two stage bootloader. MLO is the primary bootloader and u-boot.img is the second stage bootloader available in u-boot.
- * **Kernel:** zImage is the bootable kernel image available in arch/arm/boot of the Linux source. We also need device tree, blob am335x-boneblack.dtb.
- * **Rootfs:** It is loaded by kernel from root partition of the sd card.

Procedure:

- * Connect the SD card to host system and create two partitions one for **BOOT** and one for **ROOT** and format the two partitions as Fat32 and ext3 filesystems.



* Copy all bootable files (**MLO, u-boot.img, zImage, am335x-boneblack.dtb**) into **BOOT** partition.

* copy rootfs into **ROOT** partition.

* unmount the sd card.

* Insert the SD card into the board sd card slot.

* Connect the serial to usb to host and open minicom terminal.

* Then press boot button and power on by using usb cable.

* The board automatically boots from the sd card showing u-boot prompt.

* Press **space bar** to abort **autoboot**.

* Then set the **bootargs** for the kernel using **setenv** command:

=> **setenv bootargs console=ttyO0,115200 rw root=/dev/mmcblk0p2 rootfstype=ext3 rootwait**

* Load the kernel and device tree into RAM

=> **load mmc 0:1 0x82000000 zImage**

=> **load mmc 0:1 0x84000000 am335x-boneblack.dtb**

* Boot the kernel from RAM using bootz

=> **bootz 0x82000000 - 0x84000000**



Using NFS:

* To mount rootfs on target from host using NFS, open **/etc/exports** file:

```
$ sudo vim /etc/exports
```

* Copy below line in the **/etc/exports** file on host machine & save it.

```
/(Rootfs_path) 10.0.0.111(rw,sync,no_root_squash,no_all_squash,  
no_subtree_check)
```

* Restart NFS server on host

```
$ sudo /etc/init.d/nfs-kernel-server restart
```

* Then set the **bootargs** for the kernel using **setenv** command:

```
=> setenv bootargs console=ttyO0,115200 ip=10.0.0.111:10.0.0.4::255.255.  
255.0 rw root=/dev/nfs nfsroot=10.0.0.4:(/path to rootfs)
```

* Load the kernel and device tree into RAM

```
=> load mmc 0:1 0x82000000 zImage
```

```
=> load mmc 0:1 0x84000000 am335x-boneblack.dtb
```

* Boot the kernel from RAM using **bootz**

```
=> bootz 0x82000000 - 0x84000000
```



Booting kernel with network via TFTP:

* Configuring host machine.

1. Create tftp file in **/etc/xinetd.d/** directory

```
$ sudo vim /etc/xinetd.d/tftp
```

2. Copy following into that file and save it.

```
service tftp
{
    protocol          = udp
    port              = 69
    socket_type       = dgram
    wait              = yes
    user              = nobody
    server             = /usr/sbin/in.tftpd
    server_args        = /tftpboot
    disable            = no
}
```

* **/tftpboot** is the server searching path to serve files to clients.

* Start tftp server.

```
$ sudo service xinetd stop
$ sudo service xinetd start
```

* Now host system is ready to transfer images using tftp

* Now create **tftpboot** directory in / on **host** machine.

```
$ sudo mkdir tftpboot
```

* Copy **zImage** into **/tftpboot** (server directory).

* Copy **am335x-boneblack.dtb** into **/tftpboot**.



- * Connect the serial to usb to host and open minicom terminal.
- * Then press boot button and power on by using usb cable.
- * The board automatically boots from the sd card showing u-boot prompt.
- * Press **space bar** to abort **autoboot**.
- * Set server ip address on target
 - => **setenv serverip 10.0.0.4**
- * Set target ip address
 - => **setenv ipaddr 10.0.0.111**
- * Try ping from target to host
 - => **ping 10.0.0.4**
- * Then set the **bootargs** for the kernel using **setenv** command:
 - => **setenv bootargs console=ttyO0,115200 ip=10.0.0.111:10.0.0.4::255.255.255.0 rw root=/dev/nfs nfsroot=10.0.0.4:(/path to rootfs)**
- * Load the kernel and device tree into RAM
 - => **tftpboot 0x82000000 zImage**
 - => **tftpboot 0x84000000 am335x-boneblack.dtb**
- * Boot the kernel from RAM using bootz
 - => **bootz 0x82000000 - 0x84000000**

