



Beaglebone Black Booting Using U-boot

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](https://www.ti.com) 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.

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:** ulmage is the bootable kernel image available in arch/arm/boot of the Linux source. We also need device tree, blob am335x-boneblack.dtb. We can compile device tree blob using the below command in Linux root directory

```
make ARCH=arm CROSS_COMPILE=arm-linux- am335x-boneblack.dtb
```

dtb is created in arch/arm/boot/dts directory.

- And **rootfs**

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,ulmage,am335x-boneblack.dtb) into BOOT partition
- copy rootfs into ROOT partition
- unmount the 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 and insert the sd card into sd card slot

Booting Process

- 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.

```
U-Boot SPL 2014.10 (Feb 08 2016 - 12:13:24)
MMC: block number 0x100 exceeds max(0x0)
MMC: block number 0x200 exceeds max(0x0)
*** Error - No Valid Environment Area found
Using default environment

reading u-boot.img
reading u-boot.img

U-Boot 2014.10 (Feb 08 2016 - 12:13:24)

        Watchdog enabled
I2C:   ready
DRAM:  512 MiB
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1
Net:   cpsw, usb_ether
Hit any key to stop autoboot: 0
U-Boot#
```

Then set the bootargs for kernel using setenv command

```
setenv bootargs console=ttyO0,115200 root=/dev/mmcblk0p2 rootfstype=ext3,n8  
rootwait
```

- Load the kernel and device tree into RAM

```
load mmc 0:1 0x82000000 uImage
```

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

- Boot the kernel from RAM **bootm**

```
0x82000000 – 0x84000000
```

```
U-Boot# setenv bootargs console=ttyO0,115200 root=/dev/mmcblk0p2 rootfstype=ext3,n8 rootwait  
U-Boot# load mmc 0:1 0x82000000 uImage  
reading uImage  
3963960 bytes read in 219 ms (17.3 MiB/s)  
U-Boot# load mmc 0:1 0x84000000 am335x-boneblack.dtb  
reading am335x-boneblack.dtb  
26098 bytes read in 8 ms (3.1 MiB/s)  
U-Boot# bootm 0x82000000 - 0x84000000  
## Booting kernel from Legacy Image at 82000000 ...  
Image Name: Linux-3.8.13  
Created: 2016-02-08 10:06:36 UTC  
Image Type: ARM Linux Kernel Image (uncompressed)  
Data Size: 3963896 Bytes = 3.8 MiB  
Load Address: 82000000  
Entry Point: 82000000  
Verifying Checksum ... OK  
## Flattened Device Tree blob at 84000000  
Booting using the fdt blob at 0x84000000  
Loading Kernel Image ... OK  
Loading Device Tree to 8fff6000, end 8ffff5f1 ... OK  
  
Starting kernel ...  
  
Uncompressing Linux... done, booting the kernel.  
[ 0.000000] Booting Linux on physical CPU 0x0  
[ 0.000000] Linux version 3.8.13 (root@veda) (gcc version 4.9.3 (Buildroot 2015.08) ) #1 SMP Mon Feb 8 15:36:21 IST 2016  
[ 0.000000] CPU: ARMv7 Processor [413fc082] revision 2 (ARMv7), cr=10c53c7d  
[ 0.000000] CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache  
[ 0.000000] Machine: Generic AM33XX (Flattened Device Tree), model: TI AM335x BeagleBone  
[ 0.000000] Memory policy: ECC disabled, Data cache writeback  
[ 0.000000] AM335X ES2.1 (l2cache sgx neon )  
[ 0.000000] PERCPU: Embedded 9 pages/cpu @c1146000 s12992 r8192 d15680 u36864  
[ 0.000000] Built 1 zonelists in Zone order, mobility grouping on. Total pages: 129792
```



➤ Kernel loading from RAM and finally you will get login prompt

```
[ 1.846465] mmcblk0: p1 p2
[ 1.925816] kjournald starting. Commit interval 5 seconds
[ 1.932123] EXT3-fs (mmcblk0p2): mounted filesystem with ordered data mode
[ 1.939477] VFS: Mounted root (ext3 filesystem) readonly on device 179:2.
[ 1.949508] devtmpfs: mounted
[ 1.953138] Freeing init memory: 328K
can't open /dev/tty02: No such file or directory

(none) login: root
login: can't change directory to '/root'
Jan  1 00:03:03 login[746]: root login on 'tty00'
[root@(none):]# ls
bin          etc          linuxrc      mnt          sbin         usr
dev          lib          lost+found   proc         sys
[root@(none):]#
```

---- END ----

