

Beaglebone Black

Prerequisites

1. Host System: Linux-based system (Ubuntu/Debian preferred). Tested on Ubuntu 24.04.

2. Install tools and cross-compilation toolchain:

```
sudo apt update
```

```
sudo apt-get install build-essential libncurses-dev bison flex libssl-dev libelf-dev lzop u-boot-tools git bc device-tree-compiler
```

```
sudo apt-get install gcc-10-arm-linux-gnueabi g++-10-arm-linux-gnueabi
```

```
sudo update-alternatives --install /usr/bin/arm-linux-gnueabi-gcc arm-linux-gnueabi-gcc /usr/bin/arm-linux-gnueabi-gcc-10 120 --slave  
/usr/bin/arm-linux-gnueabi-g++ arm-linux-gnueabi-g++ /usr/bin/arm-linux-gnueabi-g++-10
```

3. Storage: A microSD card with at least 4GB of capacity.

4. BeagleBone Black Rev C: Ensure you have a serial debugging interface (like UART) for debugging.

Grab source code

0. Create a directory for source codes.

```
mkdir ~/linux
```

1. U-boot

```
cd ~/linux
```

```
git clone https://source.denx.de/u-boot/u-boot.git
```

```
cd u-boot
```

```
git checkout v2021.10
```

2. BeagleBone Black Kernel

```
cd ~/linux
```

```
git clone https://github.com/beagleboard/linux.git
```

```
cd linux
```

```
git checkout 5.10.162-ti-r59
```

3. BusyBox (Minimal Root FileSystem)

```
cd ~/linux
```

```
git clone https://github.com/mirror/busybox.git
```

```
cd busybox
```

```
git checkout 1_35_stable
```

Build U-Boot

- Compile u-Boot

```
cd ~/linux/u-boot
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- am335x_evm_defconfig
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j4
```

- Output files:

- MLO: SPL (Secondary Program Loader)
- u-boot.img: U-Boot binary

Build Linux Kernel

- Compile Kernel

```
cd ~/linux/linux
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- bb.org_defconfig
```

```
make ARCH=arm menuconfig
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- ulmage dtbs LOADADDR=0x80008000 -j4
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- modules -j4
```

- Output Files

- arch/arm/boot/ulmage: Kernel binary
- arch/arm/boot/dts/am335x-boneblack.dtb: Device tree blob

Build Minimal Root Filesystem

- Create a working directory for the root filesystem:

```
mkdir -p ~/linux/rootfs/{bin,etc,dev,proc,sys,tmp,usr,var,lib}
```

- Compile BusyBox

```
cd ~/linux/busybox
```

```
make defconfig
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- install CONFIG_PREFIX=../rootfs
```

- Setup BusyBox Links

```
cd ~/linux/rootfs
```

```
ln -s bin/busybox init
```

- Copy required libraries

```
cp /usr/arm-linux-gnueabi/lib/* ~/linux/rootfs/lib/
```

- Copy kernel modules

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- INSTALL_MOD_PATH=~/linux/rootfs modules_install
```

Prepare Boot media

- Connect SD card to PC and create two partitions in it using fdisk or parted or disks tool.
 - Partition 1 -> FAT16: Bootloader (MLO, u-boot.img) and Kernel (zImage, DTB).
 - Partition 2 -> EXT4: Root filesystem.
 - Following commands assume that SD-card is detected as `/dev/sda`.

```
sudo fdisk /dev/sda * Create first primary partition of size 256MB. * Create second primary partition of remaining size.
```

```
sudo mkdosfs -F 16 /dev/sda1 * First partition (boot) must be FAT partition (FAT16/32).
```

```
sudo mkfs.ext4 /dev/sda2 * Second partition (root filesystem) must be EXT4 partition.
```

```
sudo dosfstool /dev/sda1 "BOOT" * The boot partition must be named as "BOOT".
```

```
sudo e2label /dev/sda2 "ROOT" * The root partition can be named as "ROOT".
```

- Prepare BOOT partition (Copy files).
 - Mount BOOT partition to some folder e.g. `/media/nilesh/BOOT`

```
cp ~/linux/u-boot/MLO /media/nilesh/BOOT
```

```
cp ~/linux/u-boot/u-boot.img /media/nilesh/BOOT
```

```
cp ~/linux/linux/arch/arm/boot/zImage /media/nilesh/BOOT
```

```
cp ~/linux/linux/arch/arm/boot/dts/am335x-boneblack.dtb /media/nilesh/BOOT
```

```
vim /media/nilesh/BOOT/uEnv.txt
```

- Add following contents in uEnv.txt

```
loadaddr=0x80008000

fdtaddr=0x88000000

console=ttyS0,115200n8

bootsettings=setenv bootargs console=tty00,115200n8 root=/dev/mmcblk0p2 rw rootfstype=ext4 rootwait
earlyprintk mem=512M

mmcboot=echo Booting from microSD ...; setenv autoload no ; load mmc 0:1 ${loadaddr} uImage ; load mmc
0:1 ${fdtaddr} am335x-boneblack.dtb ; run bootsettings ; bootm ${loadaddr} - ${fdtaddr}

uenvcmd=run mmcboot
```

- Copy rootfs

```
cp -a ~/linux/rootfs/* /media/nilesh/ROOT
```

- Add basic device nodes in rootfs

```
cd /media/nilesh/ROOT
```

```
mknod dev/null c 1 3
```

```
mknod dev/console c 5 1
```

Boot BeagleBone Black

- Insert the SD card into the BeagleBone Black.

- Connect to the serial console (e.g., minicom) to observe boot messages. Default settings: ttyUSB0 115200n8.
- Hold the boot select button and power on the board to boot from the SD card.
- Press "Enter" to get the root shell.
- Create module dependency file.

```
depmod -a
```

- Load procfs manually (to be automated using initscripts or inittab).

```
mount -t proc p /proc
```

Cross compilation of Kernel Modules (on Ubuntu PC)

- step 1. Assuming that BBB kernel compiled on this PC in ~/bbb/linux/ directory.
- step 2. Implement simple kernel module.

```
vim pchar.c
```

```
vim Makefile
```

- Add following contents in Makefile
 - obj-m := pchar.o
- step 3. Cross Compile the module.
 - make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- -C ~/bbb/linux/ M=\$(pwd) modules
- step 4. Copy compiled pchar.ko file to the BBB SD card using SD card adapter.
- step 5. On BBB shell git commands.

```
insmod pchar.ko
```

```
rmmod pchar.ko
```

```
dmesg | tail
```

SUNBEAM INFOTECH