# Compiling and Deploying BeagleBone Black Kernel

# Bước 1: Ta tải toolchain compile linux kernel với version gcc-linaro-arm-linux-gnueabihf-4.8-2014.04\_linux tại link :

# <https://releases.linaro.org/archive/14.04/components/toolchain/binaries/>

# hoặc

# <https://github.com/vietletung/Beaglebone/tree/sourcekernel/Toolchain>

# Sau đó giải nén theo lệnh:

sudo tar xvf gcc-linaro-arm-linux-gnueabihf-4.8-2014.04\_linux.tar.xz -C /opt/

# Đổi tên và add compiler vào biến PATH của môi trường theo lệnh:

cd /opt/

$ sudo mv gcc-linaro-arm-linux-gnueabihf-4.8-2014.04\_linux/ gcc-arm-linux

$ export PATH=$PATH:/opt/gcc-arm-linux/bin

# Ta có thể check version của toolchain bằng cách :

arm-linux-gnueabihf-gcc --version

# Bước 2: Tạo các phân vùng cho thẻ nhớ:

# Ta cần tải phần mềm chia phân vùng cho thẻ nhớ bằng lệnh:

sudo apt-get install gparted

# Sau đó chạy nó bằng lệnh:

sudo gparted

# Tại giao diện của phần mềm gparted, ta tạo 2 phân vùng như sau :

# Newsize: 50MBembed journal

# File System: FAT32

# Label: BOOT

# Newsize: 1000MB

# File System: EXT3

# Label: RFS

# Bước 3: Build kernel

# Ta tải source code của kernel theo lệnh :

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

# Sau đó ta checkout sang 1 trong những branch build pass(Optional):

# 

# Hoặc dùng lệnh checkout sang branch khuyên dùng như sau:

git checkout d3ba22eb868900ac12a31f50941c6f5804013c30

# Khi clone thành công ta sẽ có thư mục tên “linux” chứa source code.

# Sau đó ta vào thư mục “linux”. Để tiến hành compile kernel, trước tiên ta cần có file cấu hình (.config) theo lệnh:

sudo make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- bb.org\_defconfig

# Sau đó ta compile kernel theo lệnh:

sudo make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- uImage dtbs LOADADDR=0x80008000 -j4

# Nếu trong quá trình compile xuất hiện lỗi : [Cannot use CONFIG\_CC\_STACKPROTECTOR\_STRONG: -fstack-protector-strong not supported by compiler](https://askubuntu.com/questions/940812/cannot-use-config-cc-stackprotector-strong-fstack-protector-strong-not-support) thì ta sẽ off tính năng này bằng lệnh :

scripts/config --disable CC\_STACKPROTECTOR\_STRONG

# Khi compile xong ta sẽ thấy xuất hiện file uImage tại “arch/arm/boot” và file am335x-boneblack.dtb trong mục “arch/arm/boot/dts”.

# Tiếp theo ta cần 1 file “uEnv.txt” để hướng dẫn uboot khởi động từ kernel. Nội dung như sau:

console=ttyS0,115200n8

netargs=setenv bootargs console=ttyO0,115200n8 root=/dev/mmcblk0p2 ro rootfstype=ext4 rootwait debug earlyprintk mem=512M

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

uenvcmd=run netboot

# Sau đó ta chuyển 3 file đó vào phân vùng BOOT của thẻ nhớ.

# Bước 4: Tạo các file gốc của hệ thống:

# Ta tại 1 bộ RFS tại link sau:

# <https://www.dropbox.com/s/k93doprl261hwn2/rootfs.tar.xz?dl=0>

# hoặc

# <https://github.com/vietletung/Beaglebone/tree/sourcekernel/Toolchain>

# Sau đó ta giải nén và chuyển các file đó vào phân vùng RFS của thẻ nhớ bằng lệnh:

sudo tar -xvf rootfs.tar.xz -C /media/”nameofcomputer”/RFS/

$ cd /media/”nameofcomputer”/RFS/rootfs/

$ sudo mv ./\* ../

$ cd ../

$ sudo rmdir rootfs

# Bước 5: Cài đặt module cho kernel (cần cắm thẻ nhớ để cài đặt):

# Ta cần các module để làm việc với kernel bằng cách gõ lệnh:

sudo make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- -j4 modules

$ sudo make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- INSTALL\_MOD\_PATH=/media/”nameofcomputer”/RFS/ modules\_install

# Sau đó rút thẻ nhớ vào cắm vào board để chạy ☺

# Hướng dẫn này dựa trên bài viết tại trang :

# <https://embedjournal.com/kernel-compilation-beaglebone-black/?fbclid=IwAR0YE7cFv-Q9PmzMp7UAiojjo7eH02BfHZYqDlWl3Jx4KKX_EKNA9q2bR-0>