

EMBEDDED LINUX

Linux kernel build

Brief

This chapter's about building a Linux-based system from scratch.

We'll download the sources of the U-boot bootloader and the Linux kernel, then build them. We'll download an pre-built Debian filesystem (too long to build from scratch).

Author

dimitri.boudier@ensicaen.fr

hugo.descoubes@ensicaen.fr

Source

[Debian: Getting started with the BeagleBone Black.](#)

by Robert C. Nelson

Reminder

!\\ Understand all commands before running them! !\\

TOOLCHAIN

Download X86 prebuilt ARM toolchain (ARM Linaro).

Download the toolchain

```
cd ${DISCOPATH}/toolchain

wget -c https://releases.linaro.org/components/toolchain/binaries/6.5-2018.12/arm-linux-gnueabi/gcc-linaro-6.5.0-2018.12-x86_64_arm-linux-gnueabi.tar.xz

tar xvf gcc-linaro-6.5.0-2018.12-x86_64_arm-linux-gnueabi.tar.xz
ls -l
```

- What does `wget` do? What does `tar xf` do?

Prepare Environment Variable

```
echo $CC
export CC=${DISCOPATH}/toolchain/gcc-linaro-6.5.0-2018.12-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-
echo $CC

${CC}gcc --version
gcc --version
```

- What is the difference between `gcc` and `${CC}gcc`?

BOOTLOADER

Download, restore version, patch and build bootloaders (MLO and U-Boot).

- What is a bootloader?

Download U-Boot

```
cd ${DISCOPATH}/bootloader
git clone -b v2019.04 https://github.com/u-boot/u-boot --depth=1
cd u-boot/

ls -a
```

Download and apply U-Boot patches

```
wget -c https://github.com/eewiki/u-boot-patches/raw/master/v2019.04/0001-
am335x_evm-uEnv.txt-bootz-n-fixes.patch
wget -c https://github.com/eewiki/u-boot-patches/raw/master/v2019.04/0002-U-Boot-
BeagleBone-Cape-Manager.patch

patch -p1 < 0001-am335x_evm-uEnv.txt-bootz-n-fixes.patch
patch -p1 < 0002-U-Boot-BeagleBone-Cape-Manager.patch
```

- Open both patch files. What do they contain?

- What is the purpose of a patch file?

Configure and build U-Boot BeagleBoneBlack

```
make ARCH=arm CROSS_COMPILE=${CC} distclean
make ARCH=arm CROSS_COMPILE=${CC} am335x_evm_defconfig
make ARCH=arm CROSS_COMPILE=${CC}
ls -l
```

- What does `make ... distclean` do? Check the `Makefile`.

- What does `make ... am335x_evm_defconfig` do? See `configs` folder.

- What does `make ...` do? What files are generated?

Create a backup of generated firmware

```
cp MLO ${DISCOPATH}/deploy/  
cp u-boot.img ${DISCOPATH}/deploy/
```

KERNEL

Download, restore version, patch and build kernel (Device tree and Linux kernel).

Linux main line download (BBB maintainer script)

```
cd ${DISCOPATH}/kernel  
  
git clone https://github.com/RobertCNelson/bb-kernel  
  
cd bb-kernel/  
ls -a
```

- What is inside this directory?

am33x-rt-v4.14 (Longterm 4.14.x + Real-Time preempt Kernel)

** /\ WARNING - kernel version used for this class: 4.9.119-bone-rt-r11 /\ **

```
git checkout origin/am33x-rt-v4.14 -b tmp  
ls -a
```

- What is inside this directory now?

Build

** /\ WARNING 1 - Open current shell on full screen (needed for menuconfig interface) /\ **

** /\ WARNING 2 - Carefully read the text below /\ **

1. Launch the script that builds the kernel image

```
./build_kernel.sh
```

2. All dependencies are downloaded, but stay here. This operation takes time, depending on your Internet connection. Meanwhile you can open the `System monitor` app)
3. Wait until a window pops up: it is the menuconfig interface. Press `Esc` twice to leave.
4. The compilation process starts. Have a look at your system monitor and look at the CPU usage. As this process takes a lot of CPU resources (and a lot of time), you are free to go and have a break :)

zImage is the compressed kernel image

```
ls -l KERNEL/arch/arm/boot/  
cp KERNEL/arch/arm/boot/zImage ${DISCOPATH}/deploy/
```

- What are the generated kernel images?

.dtb are Device Tree Binaries build for BBB project

am335x-boneblack.dtb is the default DTB for BBB project

```
ls KERNEL/arch/arm/boot/dts/  
ls KERNEL/arch/arm/boot/dts/ | grep .dtb  
cp KERNEL/arch/arm/boot/dts/am335x-boneblack.dtb ${DISCOPATH}/deploy
```

- What are `.dts` and `.dtb` files?

ROOTFS

Download ARM prebuilt distribution (Debian)

```
cd ${DISCOPATH}/rootfs  
mkdir debian && cd debian
```

```
wget -c https://rcn-ee.com/rootfs/eewiki/minifs/debian-10.4-minimal-armhf-2020-05-10.tar.xz
```

- What is a `.tar.xz` file?

Verify checksum

Verify the checksum below with the one given by the `sha256sum` command. You have to compare it to the checksum on the website.

```
sha256sum debian-10.4-minimal-armhf-2020-05-10.tar.xz
```

- What is the purpose of the checksum?

```
tar xvf debian-10.4-minimal-armhf-2020-05-10.tar.xz
cd debian-10.4-minimal-armhf-2020-05-10/
```

- What is the final file at this stage?

CONCLUSION

- Describe in few steps what you have done:

UPDATE THE `start_elflinux.sh` SCRIPT

Update the `elflinux/tp/disco/script/start_elflinux.sh` script.
It shall now contains the definition of the `CC` variable:

```
export CC=${DISCOPATH}/toolchain/gcc-linaro-6.5.0-2018.12-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-
${CC}gcc --version
```