Beaglebone Black

Prepare SD-card with BBB latest image

- Reference: https://beagleboard.org/getting-started
- These steps can be done from Windows OS (preferred) or Linux.
- step 1: Download latest image.
  - https://debian.beagleboard.org/images/bone-debian-10.3-iot-armhf-2020-04-06-4gb.img.xz
- step 2: Download & Install Balena Etcher software.
  - https://www.balena.io/etcher/
- step 3: Connect SD-card to the laptop.
- step 4: Use balena etcher to flash SD-card with the download image.
- step 5: Remove SD-card from PC and attach to BBB.
- step 6: Connect BBB serial terminal to the PC using USB-TTL serial converter. Configure putty to see the serial terminal output.
- step 7: Press S2 (boot) switch and power on BBB.
- step 8: Observe the output on the terminal.
- step 9: Login to the board (over putty serial terminal).
  - Default username: debian
  - Password: temppwd
- step 10: Commands on BBB terminal

```
uname -r
# 4.19.94-ti-r42

cat /etc/dogtag
# Information about debian image

sudo fdisk -l

# Increase FS size to SD card size (optional)
```

```
sudo /opt/scripts/tools/grow_partition.sh

# Get IP address
ifconfig
# 192.168.7.2 and 192.168.6.2
```

- step 11: If from laptop you can ping the BBB, then you can also access it using ssh.
  - On Windows gitbash or Ubuntu terminal.

```
ping 192.168.7.2

ssh debian@192.168.7.2
```

Install kernel compilation tools (on Ubuntu)

```
sudo apt-get install build-essential libncurses-dev bison flex libssl-dev libelf-dev

sudo apt-get install lzop u-boot-tools
```

Beaglebone Black -- Kernel compilation steps (on Ubuntu 22.x)

- step 0: Install gcc.

```
sudo apt-get install gcc-9-arm-linux-gnueabihf
```

- step 1. Clone beaglebone-black kernel source code.

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

- step 2. Configure and compile kernel & modules.

```
git checkout 4.19.94-ti-r42
```

- step 3: edit Makefile
  - open Makefile
  - Line.373: CC=$(CROSS_COMPILE)gcc to CC=$(CROSS_COMPILE)gcc-9
- step 4: Kernel config

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

  - Default config locations -- arch/arm/configs/

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

  - Change local version = -desd

```
make -j 8 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- LOADADDR=0x80000000 uImage dtbs
# this command will cause error
```

- step 5: Compile kernel and modules

- Open linux/scripts/dtc/dts-lexer.lex.c and change "YYLTYPE yylloc" to "extern YYLTYPE yylloc".

```
make -j 8 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- LOADADDR=0x80000000 uImage dtbs

make -j 8 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- modules
```

- step 6. Deploy kernel, DTB & modules on SD card (prepared above).
  - Connect SD card to Linux -- Assuming it is mounted on /media/nilesh/rootfs.
  - Copy the modules on SD card.

```
sudo make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- INSTALL_MOD_PATH=/media/nilesh/rootfs
modules_install
```

  - Copy dtb file on SD card.

```
sudo mkdir /media/nilesh/rootfs/boot/dtbs/4.19.94-desd

sudo cp /home/nilesh/bbb/linux/arch/arm/boot/dts/am335x-boneblack.dtb
/media/nilesh/rootfs/boot/dtbs/4.19.94-desd/
```

  - Copy zImage, config & System.map on SD card -- copying config & System.map is optional.

```
sudo cp /home/nilesh/bbb/linux/arch/arm/boot/zImage /media/nilesh/rootfs/boot/vmlinuz-4.19.94-desd

sudo cp /home/nilesh/bbb/linux/.config /media/nilesh/rootfs/boot/config-4.19.94-desd

sudo cp /home/nilesh/bbb/linux/System.map /media/nilesh/rootfs/boot/System.map-4.19.94-desd
```

    - Modify /media/nilesh/rootfs/boot/uEnv.txt (using any editor).
        - Comment: #uname_r=4.19.94-ti-r42
        - Change kernel version: uname_r=4.19.94-desd
    - Plug SD card into BBB and boot from SD card.

- step 7. Power on BBB and verify if custom kernel is booted. Get terminal using serial port or ssh.

```
uname -r
```

Compile u-boot source code (on Ubuntu)

- step 1. Clone u-boot source code.

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

- step 2. Configure u-boot with default config and then customize it (menuconfig - optional).

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- am335x_boneblack_vboot_defconfig

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

- step 3. Compile the u-boot and observe generated images.

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

ls -l
```

- step 4. These images can be written on SD-card in raw mode (using dd) or in boot partition (1st partition - fat16 partition). However we will continue with default uboot present on SD-card (copied while flashing SD-card).

Device Driver Programming (on Beaglebone Black) -- Native development (with Distro kernel 4.19.94-ti-r42)s

- Enable internet of BBB (shared from laptop) (On BBB terminal)

```
ifconfig
# 192.168.7.2, 192.168.6.2

# The laptop (192.168.7.1. 192.168.6.1) will internet gateway for BBB.
sudo route add default gw 192.168.7.1
sudo route add default gw 192.168.6.1
```

- Enable sharing internet connection from laptop (on Ubuntu)

```
echo 1 > /proc/sys/net/ipv4/ip_forward

ifconfig

# Assuming wlp0s20f3 internet network adapter of laptop
sudo iptables --table nat --append POSTROUTING --out-interface wlp0s20f3 -j MASQUERADE

sudo iptables --append FORWARD --in-interface wlp0s20f3 -j ACCEPT
```

- Add DNS server on BBB terminal

```
sudo vim /etc/resolv.conf
    nameserver 8.8.8.8
    nameserver 8.8.4.4
```

- Ensure that you are booted in distro kernel (4.19.94-ti-r42).

```
uname -r
```

- Install BBB kernel headers.

```
sudo apt update

sudo apt install linux-headers-4.19.94-ti-r42
# Internally, downloads kernel source code into /usr/src/...
```

- Type kernel module/device driver code and Makefile.

```
vim pchar.c

vim Makefile
```

- Compile device driver and execute it.

```
make

sudo insmod pchar.ko

dmesg | tail
```

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
        - obj-m := pchar.o
- step 3. Cross Compile the module.
    - make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -C ~/bbb/linux/ M=pwd modules
- step 4. Copy compiled pchar.ko file to the BBB SD card using SD card adapter, or using ssh.

```
# copy using ssh
scp /path/to/pchar.ko debian@192.168.7.2:/home/debian
```

- step 5. On BBB (booted from SD card) get the shell using serial port or ssh.

```
sudo insmod pchar.ko

sudo rmmod pchar.ko

sudo dmesg | tail
```