Configuration of BeagleBone Black

Lars Schiller (lars.schiller@tuhh.de)

April 9, 2020

Contents

1 Install Operating System

It is recommended to use one the following images (download: `http://beagleboard.org/latest-images`):

kernel 8.7	`bone-debian-8.7-iot-armhf-2017-03-19-4gb.img`
kernel 9.14	`bone-debian-9.9-iot-armhf-2019-08-03-4gb.img`

Install OS on SD card:
To install it on an `8GB` Micro-SD card, follow the instructions:

- You can use Etcher (`https://etcher.io/`).

OR (on debian):

- Decompress and write on SD card (needs to be `su` and make sure the security locker of SD adapter is in writing mode):

```
$ xz -d bone-debian-**.img.xz
$ dd if=./bone-debian-**.img of=/dev/sdX
```

  Here, `sdX` is the mounted empty SD card. It can be found with multiple use of the command `mount` or `df`.

Then, push the SD card with installed operating system into the BeagleBone Black (BBB).

2 Log into BBB for the first time

Assuming you are called `user` and your PC is called `pc`, your BBB is called `beaglebone` and the default user on BBB is called `debian`, then the following sythax is correct.

- Connect your PC with a MicroUSB cable to the BBB.
- Open a terminal and ssh into BBB as `debian` and then get superuser to configure the board.

```
user@pc:~ ssh debian@192.168.7.2
temppwd
debian@beaglebone:~ su
root
root@beaglebone:~#
```

- Note that the default passwords are:

temppwd	for `debian`
root	for `root`

3 Change static IP of USB port

`https://stackoverflow.com/questions/23805457/changing-the-static-ip-of-beagle-bone-black-usb0`

- To change the static ip of BBB's usb0 interface from default 192.168.7.2 to ...5.2:

```
root@bbb:~# nano /etc/network/interfaces

iface usb0 inet static
        address 192.168.5.2
        netmask 255.255.255.0
        network 192.168.5.0
        gateway 192.169.5.1
```

- (I also edited the file `/opt/scripts/boot/am335_evm.sh`. Maybe it had an effect...)

4 Configure Ethernet

`https://groups.google.com/forum/#!msg/beaglebone/AS2US9rtNd4/8y0mZ3LxAwAJ`

- Assuming you want to configure `eth0` like this:

address	134.28.136.51 (ask administrator for your personal IP)
netmask	255.255.255.0
dns-nameservers	134.28.205.14
gateway	134.28.136.1

- Plug in LAN cable.
- Get the name of the LAN connection:

```
su
root@beaglebone:/etc/network# connmanctl services
*Ac Wired                ethernet_689e19b50543_cable
```

- Using the appropriate ethernet service, tell `connman` to setup a static IP address for this service.

```
connmanctl config <service> --ipv4 manual <ip_addr> <netmask> <gateway> --nameservers <
    dns_server>
```

In our case:

```
connmanctl config ethernet_689e19b50543_cable --ipv4 manual 134.28.136.51 255.255.255.0
    134.28.136.1 --nameservers 134.28.205.14
```

- Reboot and you are done.
- You can revert back to a DHCP configuration simply as follows:

```
$ connmanctl config ethernet_689e19b50543_cable --ipv4 dhcp
```

5 Configure SSH

https://askubuntu.com/questions/115151/how-to-set-up-passwordless-ssh-access-for-root-user

- If your Board crashed, and you were forced to reinstall the OS, there already exist a ssh-key. This you have to remove first (this is for USB cable):

```
user@pc:~ ssh-keygen -f "/home/user/.ssh/known_hosts" -R 192.168.7.2
```

- Generate a new key:

```
user@pc:~ ssh-keygen -f "/home/user/.ssh/key_user"
```

  When you are prompted for a password, just hit the enter key and you will generate a key with no password.

- Allow to log in as root with a password on the server, in aim to transfer the created key to it:

```
root@beaglebone:# nano /etc/ssh/sshd_config
```

  Make sure you allow root to log in with the following syntax

```
PermitRootLogin yes
PasswordAuthentication yes
```

  Restart the ssh-server:

```
root@beaglebone:# service ssh restart
```

- Now you are able to transfer the key to the server:

```
user@pc:~ ssh-copy-id -i /home/user/.ssh/key_user root@192.168.7.2
```

- Check if its work:

```
user@pc:~ ssh root@192.168.7.2
```

- Now disable root login with password on server (for safety):

```
root@beaglebone:# nano /etc/ssh/sshd_config
```

  And modify the Line:

```
PermitRootLogin without-password
PasswordAuthentication yes
```

  This will allow to login as root with valid key, but not with a password. All other users can further login with a password. Restart the ssh-server and you are done:

```
root@beaglebone:# service ssh restart
```

6 Configure Device Tree (enabling all PWM pins)

In order to enable `P9.28` as pwm pin, you have to load `cape-universala`.

Debian 9 / Kernel v4.14.71-ti-r80:
`https://elinux.org/Beagleboard:BeagleBoneBlack_Debian#U-Boot_Overlays`

- Note: you might need to disable HDMI with `disable_uboot_overlay_video=1` in `/boot/uEnv.txt` if the pins are already in use.
- update bootloader (check version 19-08-07):

```
root@beaglebone:~$ cd /opt/scripts/tools/
root@beaglebone:/opt/scripts/tools$ git pull
root@beaglebone:/opt/scripts/tools$ ./version.sh | grep bootloader
bootloader:[eMMC-(default)]:[/dev/mmcblk1]:[U-Boot 2016.01-00001-g4eb802e]:[location: dd MBR]
```

To upgrade your version of U-Boot:

```
root@beaglebone:~$ cd /opt/scripts/tools/developers/
root@beaglebone:/opt/scripts/tools/developers$ ./update_bootloader.sh
root@beaglebone:/opt/scripts/tools/developers$ reboot

...

root@beaglebone:/opt/scripts/tools$ ./version.sh | grep bootloader
bootloader:[microSD-(push-button)]:[/dev/mmcblk0]:[U-Boot 2019.04-00002-gbb4af0f50f]:[location: dd MBR]
bootloader:[eMMC-(default)]:[/dev/mmcblk1]:[U-Boot 2016.01-00001-g4eb802e]:[location: dd MBR]
```

Delete the old version:

```
root@beaglebone:/opt/scripts/tools$ dd if=/dev/zero of=/dev/mmcblk1 bs=1M count=10
```

Also make sure the bb-cape-overlays package is upto date

```
apt update
apt install --only-upgrade bb-cape-overlays
```

Debian 8 / Kernel version v4.4.54
`https://groups.google.com/forum/#!topic/beagleboard/EYSwmyxYjdM`

- `/boot/uEnv.txt` should be looking something like this:

```
root@beaglebone:# cat /boot/uEnv.txt | grep -v "#"

uname_r=4.4.54-ti-r93
cmdline=coherent_pool=1M quiet cape_universal=enable
```

Edit it with:

```
root@beaglebone:# nano /boot/uEnv.txt
```

Add the following lines, such that `/boot/uEnv.txt` looks like:

```
root@beaglebone:# cat /boot/uEnv.txt | grep -v "#"

uname_r=4.4.54-ti-r93
dtb=am335x-boneblack-overlay.dtb
cmdline=coherent_pool=1M quiet cape_universal=enable
cape_enable=bone_capemgr.enable_partno=cape-universala
```

- Reboot and you should be able to configure with:

```
root@beaglebone:# config-pin P9_28 pwm
```

7 Set I2C Bus to FastMode (400kHz)

Kernel version 4.14.xx:

- Backup the original `.dtb`:

```
root@beaglebone: /boot/dtbs/4.14.71-ti-r80# cp am335x-boneblack.dtb am335x-boneblack.dtb
    .orig
```

- Generate source device tree (`.dts`) from binary block device tree (`.dtb`) with device tree compiler (`dtc`):

```
root@beaglebone: /boot/dtbs/4.14.71-ti-r80# dtc -I dtb -O dts -o am335x-boneblack.dts
    am335x-boneblack.dtb
```

- There are 3 diffrent i2c-buses in the `.dts`:
  - i2c0: 0x44E0B000 (not available as Pins)
  - i2c1: 0x4802A000 (not enabled by default)
  - i2c2: 0x4819C000 (the actual one for configured i2c-1 in Linux-Debian, although the register name/-expansion port is i2c2)

  We want to increase the speed of the `i2c2` bus. Therefore modify the `.dts` with nano:

```
i2c@4819c000 {
compatible = "ti,omap4-i2c";
#address-cells = <0x1>;
#size-cells = <0x0>;
ti,hwmods = "i2c3";
reg = <0x4819c000 0x1000>;
interrupts = <0x1e>;
status = "okay";
pinctrl-names = "default";
pinctrl-0 = <0x35>;

#clock-frequency = <0x186a0>;
clock-frequency = <0x61a80>;

linux,phandle = <0xa1>;
phandle = <0xa1>;
```

  The `clock-frequency = <0x186a0>` is the frequency, `0x186a0 = 100000 = 100kHz` here is the default i2c-1 (Expansion port i2c2) frequency for stock BeagleBone Black image. `0x61a80 = 400000 = 400kHz` is the highest frequency possible for i2c-devices. This we gonna use.

- Generate the `.dtb` from this modified .dts:

```
root@beaglebone: /boot/dtbs/4.14.71-ti-r80# dtc -I dts -O dtb -o am335x-boneblack.dtb
    am335x-boneblack.dts
```

- Reboot and check:

```
root@beaglebone:# dmesg | grep i2c
```

  Something like

```
...
omap/i2c@4819c000 is enabled at 400kHz
...
```

  should be the output.

Kernel version <4.4.xx:

- For kernel version < 4.4.xx replace `am335x-boneblack.dtb` with `am335x-boneblack-overlay.dtb`

8 Disable unused programs

- Webserver:

```
root@beaglebone: systemctl stop apache2.service
root@beaglebone: systemctl disable apache2.service
```

- NodeJS:

```
root@beaglebone: systemctl stop bonescript-autorun.service
root@beaglebone: systemctl disable bonescript-autorun.service
```

9 Install Software

In order to run the software on the BBB, install following packages:

- **python3**: on BBB as `su`

```
root@beaglebone:# apt-get update
root@beaglebone:# apt-get install ntpdate
root@beaglebone:# ntpdate pool.ntp.org
root@beaglebone:# apt-get install build-essential python3-pip python3-scipy python3-
    numpy -y

root@beaglebone:# pip3.5 install Adafruit_BBIO Adafruit_GPIO board Adafruit-Blinka
    adafruit-circuitpython-charlcd


root@beaglebone:~# mkdir Git
root@beaglebone:~# cd Git
root@beaglebone:~/Git/# git clone https://github.com/larslevity/GeckoBot.git
```

- **python2**: on BBB as `su`

```
root@beaglebone:# apt-get update
root@beaglebone:# apt-get install ntpdate
root@beaglebone:# ntpdate pool.ntp.org
root@beaglebone:# apt-get install build-essential python-dev python-pip -y
root@beaglebone:# pip install --upgrade pip
root@beaglebone:# pip install Adafruit_BBIO
root@beaglebone:# pip install Adafruit_GPIO
root@beaglebone:# pip install termcolor
root@beaglebone:# pip install numpy

root@beaglebone:~# mkdir Git
root@beaglebone:~# cd Git
root@beaglebone:~/Git/# git clone https://github.com/larslevity/GeckoBot.git
```