Elph Training Exercises

Exercises

All the required images and code for Elph training are uploaded at sharepoint
https://www.dropbox.com/sh/0liva3mj09mld78/AABgahs-y1oxM-ZjI289bQyLa?dl=0

Session 1 Exercises

Download all the images.tgz from the sharepoint under Images directory. Untar the same under Elph directory:

```
$ mkdir Elph
$ cd Elph
$ tar -xvf images.tgz
```

This should get MLO, u-boot.img, rd-ext2.bin, am335x-boneblack.dtb, zImage, user-mmc.txt, uEnv.txt uEnv-mmc.txt, rootfs.tgz under Images directory

Exercise 1: Booting up the board with pre-built images

The idea over here is to boot up the board with available images from the sharepoint.

1 Plug in the uSD card into Beaglebone Black and power up the board with USB cable. Make sure that the minicom is configured and you are able to target console on minicom.

2 Mount the first partition of uSD Card and transfer the images

```
$ mount /dev/mmcblk0p1 /mnt (On board)
$ cd Elph/Images
$ scp zImage uEnv.txt rd-ext2.bin am335x-boneblack.dtb MLO u-boot.img usr-mmc.txt root@192.168.7.2:/mnt/
$ umount /mnt
$ reboot
```

3 Once the system is rebooted, it should display “Welcome to Timmins”

Exercise 2: Stopping at uboot

The objective is to boot-up the board to uboot (second stage bootloader)

1 Mount the first parition of uSD card and rename uEnv.txt to uEnv-org.txt

```
$ mount /dev/mmcblk0p1 /mnt
$ cd /mnt
$ mv uEnv.txt uEnv-org.txt
$ umount /mnt
```

2 Reboot the board

```
$ reboot
```

3 If it still comes up to user space, next thing is to do the similar change for eMMC:

```
$ sudo mount /dev/mmcblk1p1 /mnt
$ cd /mnt/boot/
$ mv uEnv.txt uEnv-org.txt
$ sudo umount /mnt
$ reboot
```

4 With this, the target board should not proceed beyond uboot. To stop at the uboot, quickly press the “SPACE” once the board is rebooted

5 Next step is to help uboot locate the kernel image and boot up with the same:

```
$ fatload mmc 0:1 $loadaddr zImage
$ bootz $loadaddr
```

6 Next thing is to try with zImage and dtb:

```
$ fatload mmc 0:1 $loadaddr zImage
$ fatload mmc 0:1 $fdt_addr_r am335x-boneblack.dtb
$ bootz $loadaddr - $fdt_addr_r
```

7 Next thing to set the bootargs

```
$ setenv bootargs console=ttyO0,115200n8
```

Repeat the above steps
Observe the behaviour

Exercise 3: Booting up with Ramdisk

The objective here is to provide the rootfs.This is where the ramdisk would be used

1 Reboot the board and press ‘SPACE’ during the boot-up. This would provide the u-boot prompt.

2 Load the uEnv-org.txt, export it to the shell environment & boot up:

```
$ fatload mmc 0:1 $loadaddr uEnv-org.txt
$ run importbootenv
$ run uenvcmd
```

This should boot-up the board with ramdisk

Exercise 4: Preparing the uSD Card for Beaglebone Black boot up

The objective here is to prepare the uSD Card by populating the various stuffs such as bootloaders, rootfs and so on. Prior to this the uSD needs to be partitioned and formatted.

Let’s first preserve the existing images in eMMC

```
$ mount /dev/mmcblk0p1 /mnt (Mount the uSD first partition)
$ mkdir tmp
$ mount /dev/mmcblk1p1 tmp (Mount the eMMC first partition)
$ cp /mnt/* /root/tmp/opt/backup/
$ umount /mnt
$ umount tmp
```

Partitioning the uSD

1 Execute fdisk on uSD & delete all the existing partitions:

```
$ fdisk /dev/mmcblk0
$ ‘u’  (This would change the unit to cylinders)
$ ‘d’ (Deletes the partition)
```

2 Next thing is to set the uSD geometry. Its recomended to have the 255 heads, 63 sectors/tracker

```
$ ‘x’ (Switches to expert mode)
$ ‘h’ (Set the number of heads to 255)
$ ‘s’ (Set the number of sectors to 63)
$ ‘c’ (This needs to be calculated as (size in bytes)/(255 * 63 * 512))
$ ‘r’ (Return back to normal mode)
```

3 Let’s create the partitions

```
$ ‘n’ (New partition)
$ ‘p’ (Primary partition and give the partition number as 1 and first cylinder as 4 and last cylinder as +128M)
```

Change the partition type as Windows LBA and make it bootable
$ ‘t’ and enter c
$ ‘a’ & select 1 (This would make partition 1 as bootable. Enter ‘p’ & notice the ‘*’ under boot)

4 Create the second primary partition with start cylinder as 20 and for last cylinder, enter +2G

5 Create the third primary partition with start cylinder as 281 and end cylinder as +1G. The last cylinder can be default (to use the entire disk) if its 4GB card.

6 Finally, write the partition table with ‘w’

7 Next step is to create the filesystem:
$ mkfs.vfat -F 32 -n boot /dev/mmcblk0p1 (Create FAT32 filesystem on first partition)
$ mkfs.ext2 -L FristRootfs /dev/mmcblk0p2 (Create Ext2 filesystem on second partiton)
$ mkfs.ext2 -L SecondRootfs /dev/mmcblk0p3 (Create Ext2 filesystem on third partiton)

8 Next step is to populate the partitions. The first partition should ideally have MLO, u-boot.img, uEnv.txt and am335x-boneblack.dtb

```
$ mount /dev/mmcblk0p1 /mnt
$ cd Elph/Images
$ scp MLO u-boot.img am335x-boneblack.dtb zImage root@192.168.7.2:/mnt/
$ scp uEnv-mmc.txt root@192.168.7.2:/mnt/uEnv.txt
$ umount /mnt
```

9 Next step is to populate the Rootfs

```
$ mount /dev/mmcblk0p2 /mnt
$ scp rootfs.tgz root@192.168.7.2:/mnt/
$ cd /mnt
$ tar -xvf rootfs.tgz
$ cd
$ umount /mnt
```

10 This should boot up with second partition

Exercise 5: Building the uboot

The objective here is to clone the u-boot repo and switch to v2019.04 version and build the same for bbb

1 Navigate to Elph directory and create a directory by name ‘Bootloaders’

```
$ cd Elph
$ mkdir Bootloaders
$ cd Bootloaders
```

2 Clone the git repo and check v2019.04

```
$ git clone https://github.com/u-boot/u-boot
$ cd u-boot
$ git checkout v2019.04 -b tmp
```

3 Next step is to fetch & apply the patch:

```
$ mkdir ../Patches
$ wget -c -P ../Patches/
https://raw.githubusercontent.com/eewiki/u-boot-patches/master/v2019.04/0001-am335x_evm-uEnv.txt-bootz-n-fixes.patch
```

4 Configure & build the uboot

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

5. This would generate MLO and u-boot.img, which can be used to boot up the board

Exercise 6: Booting with the third (back-up) partition

The objective here is to prepare the back up partition which can be used for recovering the board in case it fails to boot up with default partition

1. First thing is to populate the contents of the third partition
   ```
   $ mount /dev/mmcblk0p3 /mnt (on the target board)
   $ cd Elph/Images
   $ scp rootfs.tgz root@192.168.7.2:/mnt/
   $ cd /mnt
   $ tar -xvf rootfs.tgz (This should untar the rootfs contents)
   $ Create a directory by name third (This is just to distinguish between 2nd and 3rd partition)
   $ cd (Come out of /mnt)
   $ umount /mnt
   ```
2. Next step is to modify uEnv.txt file. For this, navigate to the Images directory under Elph and modify the uEnv-mmc.txt
   ```
   $ cd Elph/Images
   ```
   Create a copy of uEnv-mmc.txt
   ```
   $ cp uEnv-mmc.txt uEnv-third.txt
   ```
   Modify to load the kernel image & dtb from the third partition and set the root=/dev/mmcblk0p2
3. Transfer the uEnv-third.txt to the board
   ```
   $ mount /dev/mmcblk0p1 /mnt (On target)
   $ scp uEnv-third.txt root@192.168.7.2:/mnt/
   ```
4. Next preserve the uEnv.txt to uEnv-old.txt and replace uEnv.txt
   ```
   $ cp uEnv.txt uEnv-old.txt
   $ cp uEnv-third.txt uEnv.txt
   $ cd
   $ umount /mnt
   ```
5. Let’s reboot and check if its booted from 3rd partition by verifying if directory ‘third’ is present in /

Exercise 7: Booting with the third (back-up) partition

The serial booting is very handy for unbricking the board. This would be useful in case uboot and MLO have been corrupted or have been removed.

1. First thing is to erase the MLO from the eMMC, otherwise the board would always boot up with the one present in eMMC. The MLO is raw dumped at the offset 128K and uboot at offset 384KB. The back up for both is available at opt/backup/uboot
   ```
   $ dd if=/dev/zero of=/dev/mmcblk1 bs=512 seek=256 count=256
   $ dd if=/dev/zero of=/dev/mmcblk1 bs=512 seek=768 count=1024
   ```
2. Make sure that the uSD is removed as well. Next thing is to reset the board and we should get ‘CCCC...” on the serial console. The ROM code understand the Xmodem protocol which is supported by minicom as well. Press ‘Ctrl + a’, then release and then press ‘s’ in minicom. This should give an options for ‘Xmodem, Ymodem and Zmodem’. Let’s select the xmodem. Next, it would prompt for file to be transferred.

So, navigate to Elph/Bootloaders/u-boot/spl directory and select u-boot-spl.bin. This should start the transfer. If not, reset the board and it should start the transfer.

3 Once the X-loader is transferred, next thing is to transfer the u-boot. Press ‘Ctrl + a’ , then release and then press ‘s’. Select xmodem and select u-boot.bin from Bootloaders/u-boot/. This should transfer u-boot.bin.

4 Once the transfer is done, make sure to press the ‘space’ quickly to get the uboot prompt.

Exercise 8: Adding the command in u-boot

The idea over here is to add the custom command in the uboot

1. Get into the u-boot directory

```
$ cd Elph/Bootloaders/uboot
```

2 Make a copy of already existing command say led.c under cmd

```
$ cd cmd
$ cp led.c myprint.c
```

3 Modify the myprint.c as follows:

- Remove everything except do_led and U_BOOT_CMD
- Change the name of the function do_led to do_myprint
- Deleting everything in do_led, except the ‘return 0’
- Include the ‘printf’ statement above ‘return 0’

4 Modify the U_BOOT_CMD as below:

```
U_BOOT_CMD(
        myprint, 1, 1, do_myprint,
        "My printf",
        "My Test command"
);
```

5 Next thing is to modify the Kconfig to give out the menu option for our command. For this, modify the Kconfig to add the menu option as below:

```
config CMD_MYPRINT
        bool "MY Test Print"
        help
          Enable the Test printf
```

6 Next thing is to modify the Makefile to add the below line:

```
obj-$(CONFIG_CMD_MYPRINT) += myprint.o
```

7 Get into u-boot top level directory

```
$ make menuconfig
```

Search for MYPRINT and select this option

8 Rebuild the uboot

```
$ make CROSS_COMPILE=arm-linux-gnueabihf-
```

9 Check if myprint.o is generated under uboot/cmd/

10 Copy the u-boot.img to the first partition of the board:

```
$ mount /dev/mmcblk0p1 /mnt (on the board)
$ scp u-boot.img root@192.168.7.2:/mnt/ (On host)
$ sudo umount /mnt (on board)
```

Power-off the board

```
$ poweroff
```

11 Follow below steps to force a boot from uSD card:

- Power off the board & remove the usb power cable
- While keeping the switch S2 pressed, connect the usb power cable
- While booting up, press ‘Space’ key to get the uboot prompt.