Prepare U-boot as the Bootloader and Boot BBB Assignment

Conduct the following steps to accomplish various tasks in completing this assignment.

<u>For this assignment you will need to use a mini SD card.</u>

1. Get Bootloader

```
cd eel4734

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

$ cd u-boot

$ git checkout master                    checkout v2018.11   and use am335x_boneblack_defconfig
```

There are more than 1,000 configuration files for common development boards and devices in the configs/ directory.

Looking through the per-board README files in the board/ directory, or you can find information in an appropriate web tutorial or forum.

Taking the BeagleBone Black as an example, we find that there is a likely configuration file named configs/am335x_boneblack_vboot_defconfig and we find the text The binary produced by this board supports … Beaglebone Black in the board README files for the am335x chip, board/ti/am335x/README.

2. Therefore, to build U-Boot

```
$ sudo apt install libssl-dev

$ sudo apt install gtkterm


$ PATH=${HOME}/x-tools/arm-cortex_a8-linux-gnueabihf/bin/:$PATH

$ export CROSS_COMPILE=arm-cortex_a8-linux-gnueabihf-

$ export ARCH=arm
```

Clean (not needed the first time, but should be done for a clean starting point)

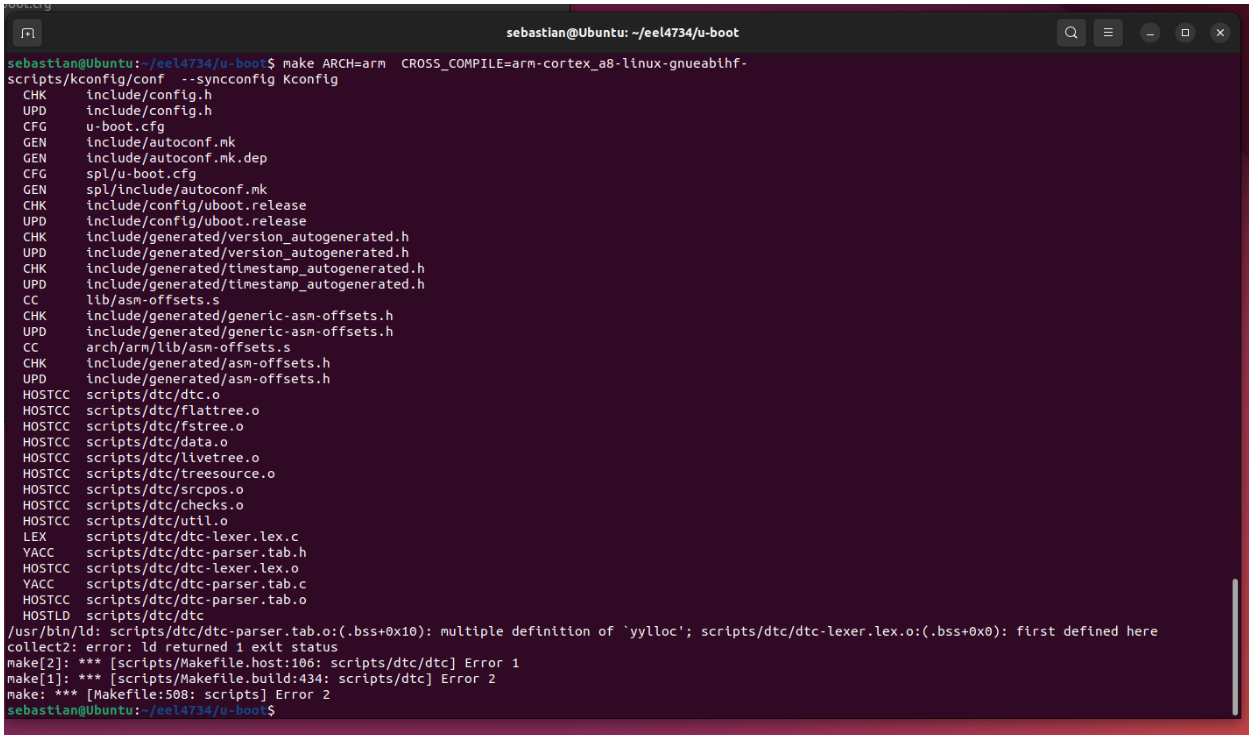
```
$ make ARCH=arm  CROSS_COMPILE=arm-cortex_a8-linux-gnueabihf- distclean
$ make ARCH=arm  CROSS_COMPILE=arm-cortex_a8-linux-gnueabihf-   am335x_evm_defconfig
```

$ make menuconfig

In Boot images -> Enable support for the legacy image format (might be already enabled)

$ make ARCH=arm CROSS_COMPILE=arm-cortex_a8-linux-gnueabihf-

As below:



```
sebastian@Ubuntu:~/eel4734/u-boot$ make ARCH=arm  CROSS_COMPILE=arm-cortex_a8-linux-gnueabihf-
scripts/kconfig/conf  --syncconfig Kconfig
  CHK     include/config.h
  UPD     include/config.h
  CFG     u-boot.cfg
  GEN     include/autoconf.mk
  GEN     include/autoconf.mk.dep
  CFG     spl/u-boot.cfg
  GEN     spl/include/autoconf.mk
  CHK     include/config/uboot.release
  UPD     include/config/uboot.release
  CHK     include/generated/version_autogenerated.h
  UPD     include/generated/version_autogenerated.h
  CHK     include/generated/timestamp_autogenerated.h
  UPD     include/generated/timestamp_autogenerated.h
  CC      lib/asm-offsets.s
  CHK     include/generated/generic-asm-offsets.h
  UPD     include/generated/generic-asm-offsets.h
  CC      arch/arm/lib/asm-offsets.s
  CHK     include/generated/asm-offsets.h
  UPD     include/generated/asm-offsets.h
  HOSTCC  scripts/dtc/dtc.o
  HOSTCC  scripts/dtc/flattree.o
  HOSTCC  scripts/dtc/fstree.o
  HOSTCC  scripts/dtc/data.o
  HOSTCC  scripts/dtc/livetree.o
  HOSTCC  scripts/dtc/treesource.o
  HOSTCC  scripts/dtc/srcpos.o
  HOSTCC  scripts/dtc/checks.o
  HOSTCC  scripts/dtc/util.o
  LEX     scripts/dtc/dtc-lexer.lex.c
  YACC    scripts/dtc/dtc-parser.tab.h
  HOSTCC  scripts/dtc/dtc-lexer.lex.o
  YACC    scripts/dtc/dtc-parser.tab.c
  HOSTCC  scripts/dtc/dtc-parser.tab.o
  HOSTLD  scripts/dtc/dtc
/usr/bin/ld: scripts/dtc/dtc-parser.tab.o:(.bss+0x10): multiple definition of `yylloc'; scripts/dtc/dtc-lexer.lex.o:(.bss+0x0): first defined here
collect2: error: ld returned 1 exit status
make[2]: *** [scripts/Makefile.host:106: scripts/dtc/dtc] Error 1
make[1]: *** [scripts/Makefile.build:434: scripts/dtc] Error 2
make: *** [Makefile:508: scripts] Error 2
sebastian@Ubuntu:~/eel4734/u-boot$
```

If you get the following error:

usr/bin/ld: scripts/dtc/dtc-parser.tab.o:(.bss+0x10): multiple definition of `yylloc'; scripts/dtc/dtc-lexer.lex.o:(.bss+0x0): first defined here collect2: error: ld returned 1 exit status

Then perform the following actions and run the last command again:

edit the file ./linux-rtk/scripts/dtc/dtc-lexer-lex.c

Find the line 'YYLTYPE yylloc' and change it to 'extern YYLTYPE yylloc'

Github fix: https://github.com/BPI-SINOVOIP/BPI-M4-bsp/issues/4

The results of the compilation are:

u-boot: U-Boot in ELF object format, suitable for use with a debugger

u-boot.map: The symbol table

u-boot.bin: U-Boot in raw binary format, suitable for running on your device

u-boot.img: This is u-boot.bin with a U-Boot header added, suitable for uploading to a running copy of U-Boot

u-boot.srec: U-Boot in Motorola S-record (SRECORD or SRE) format, suitable for transferring over a serial connection

The BeagleBone Black also requires a secondary program loader (SPL), as described earlier. This is built at the same time and is named MLO:

```
$ ls -l MLO u-boot*
```

3. Installing U-Boot

Now, in virtualBox, select the menu entry Device and select USB, where you will click on the USB Reader to make it accessible within Ubuntu

Find microSD card:

```
$ lsblk
```

It will be something like sdb

4. Format SD card

In eel4734 where you have copied the bash script format-sdcard.sh

```
 $ chmod +x format-sdcard.sh
```

```
$ ./format-sdcard.sh sdb              (where sdb indicate the SD card name)

                                      (if get error, run it again, need unmount)
```

After you have formatted the microSD card, remove it from the card reader and then re-insert it so that the partitions are auto mounted.

Do it, by unselecting USB at the Device menu selection and then reselecting it so that it now will show-up as

On current versions of Ubuntu, the two partitions should be mounted as /media/[user]/boot and /media/[user]/rootfs. Now you can copy the SPL and U-Boot to it like this:

Here student is your username in Ubuntu

```
$ cd ~/eel4734/u-boot
```

```
$ cp MLO u-boot.img /media/student/boot
```

Finally, unmount it:

```
$ sudo umount /media/student/boot
```

And unselect USB from the Device menu selection

Download and install virtualBox Extension Pack

https://ourcodeworld.com/articles/read/1297/how-to-install-the-oracle-vm-virtualbox-extension-pack-in-virtualbox-for-windows-10

Enable USB Controller 3.0

https://ourcodeworld.com/articles/read/1296/how-to-solve-virtualbox-exception-when-attaching-a-usb-device-failed-to-create-a-proxy-device-for-the-usb-device-error-verr-pdm-no-usb-ports

Now, with no power on the BeagleBone board, insert the micro-SD card into the reader.

Plug in the serial cable.

Then in virtualbox select Device->USB->Prolific Technology Inc. USB-Serial Controller

A serial port should appear on your PC as /dev/ttyUSB0.

```
$ ls /dev/ttyUSB0
```

Start a suitable terminal program, such as gtkterm, minicom, or picocom, and attach to the port at 115200 bps (bits per second) with no flow control. Gtkterm is probably the easiest to setup and use:

```
$ sudo gtkterm -p /dev/ttyUSB0 -s 115200
```

Press and hold the Boot Switch button on the Beaglebone Black, power up the board using the external 5V power connector, and release the button after about 5 seconds. You should see a U-Boot prompt on the serial console:

```
U-Boot#
```

Display the command available for U-boot and take screenshot of the u-boot prompt and the list of available command.