

Final Linux Distro with Custom Driver and Boot Message

In this assignment you will add the Morse code driver to Linux and have the BBB blink a message on the LED on start-up. Assignment adding Morse code driver to linux and add method to blink led on startup to say, "Welcome to Embedded Linux"

Make sure to provide screenshots for each step and a video booting the board showing the led blinking. Make sure to indicate where you placed the echo "Welcome to Embedded Linux" > /dev/morsecode

Place the SD card to have access from the system to copy files to it, and enter:

Add Morese Code Device Driver to the Kernel

1. cd into ~/eel4734/linux-stable/drivers/char
2. Create a directory named morseCode in the stated directory as:
`mkdir morseCode`
3. Place the device driver C code morsecode.c in this directory morseCode
4. Create a makefile in this directory
 - Makefile should contain: `obj-$(CONFIG_MORSECODE_MODULE) += morsecode.o`
 - This makefile will now look for any morsecode.c file
5. Create a Kconfig file in this directory as well
 - Kconfig file should contain:

```
config MORSECODE_MODULE
    tristate "Enable MorseModule"
    default y
    help
        Enable the kernel to support using LED0 for sending
        Morse code messages
```
6. Run the `cd ..` command to go back one directory. Your present working directory should be '/linux/drivers/char'
7. Edit the Kconfig file in this directory. The goal is to configure the module to be loaded. The file should contain, at the end:

`source "drivers/char/morseCode/Kconfig"`
8. In the same directory, make changes to the Makefile. Open the file and go to the end and add the following line:

`obj-$(CONFIG_MORSE_MODULE) += morseCode/`

Compiling Kernel with the Device Driver

1. Now cd into linux-stable, and complete the following steps.
2. Now compile the driver with the following command:

(must be done every time to establish the PATH to the toolchain and env variables)

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

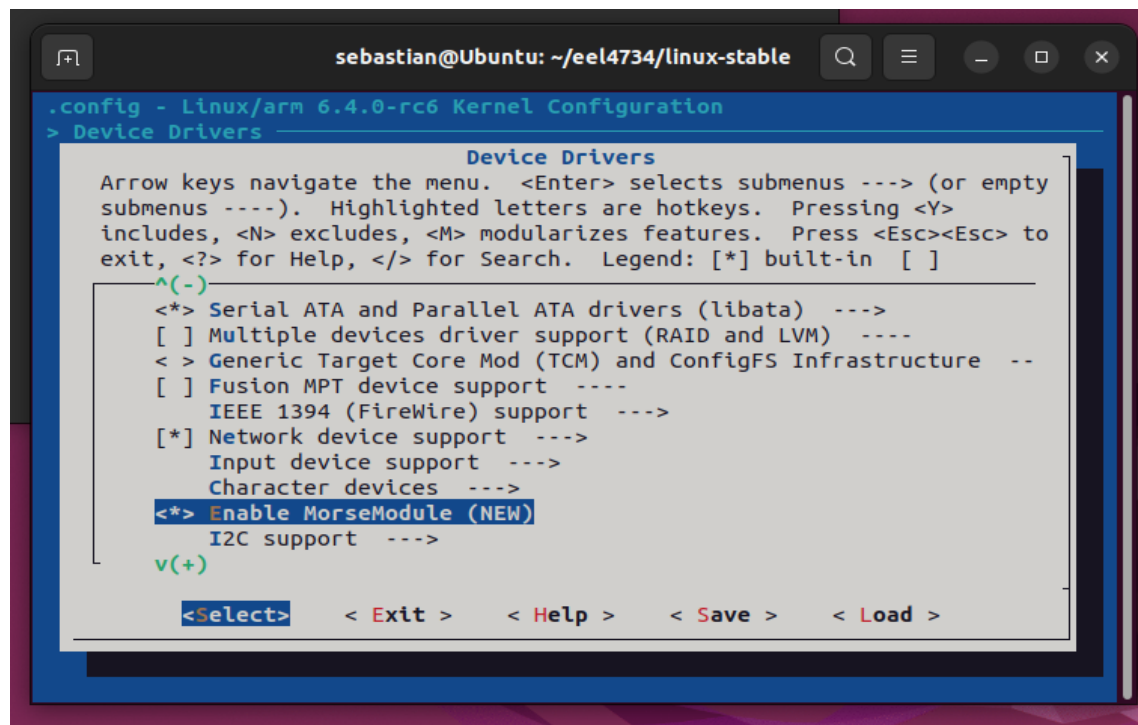
```
$ export CROSS_COMPILE=arm-cortex_a8-linux-gnueabi-
```

```
$ export ARCH=arm
```

```
$ export SYSROOT=$(arm-cortex_a8-linux-gnueabi-gcc -print-sysroot)
```

```
$ make ARCH=arm CROSS_COMPILE=arm-cortex_a8-linux-gnueabi- menuconfig
```

Find the MorseCode Module in the menuconfig tool and take a screenshot



Save configuration

```
$ make -j4 ARCH=arm CROSS_COMPILE=arm-cortex_a8-linux-gnueabi- zImage
```

```
$ make -j4 ARCH=arm CROSS_COMPILE=arm-cortex_a8-linux-gnueabi- modules
```

```
$ make ARCH=arm CROSS_COMPILE=arm-cortex_a8-linux-gnueabi- dtbs
```

(make sure to copy the required files as you did previously to the SD card the boot directory to include the new zimage.

Add to RFS the ability to blink the start message

1. Using the inittab file determine where to add the line:
echo "Welcome to Embedded Linux" > /dev/morsecode

Loading the Kernel and RFS to the SD Card

1. Insert your SD card to your computer. Be sure it appears on your Ubuntu-VM
2. You have previously formatted the SD card with two partitions one called boot and the other rootfs. Copy the newly created zimage (contain the morse code device driver) to the boot partition (as shown in previous assignment tutorial)
3. Now install your RFS(Root File System) to your SD card. Transfer the contents of the RFS to your SD card, go to ~/eel4734 and enter

```
$ sudo dd if=rootfs of=/dev/mmcblk0p2
```

If get an error, then try

```
$ sudo dd if=rootfs.tar.xz of=/dev/mmcblk0p2
```

4. Check that SD card contains the RFS partition

Boot BeagleBoneBlack from the SD Card

1. Connect your debug cable to the beaglebone and to your computer. Make sure your VM recognizes it
2. Run picocom with the following command `picocom -b 115200 /dev/ttyUSB0` and be sure the output of that command reads 'Terminal Ready' (might have to install picocom)
3. Insert your SD into the BeagleBoneBlack
4. Hold the reset button on the BeagleBoneBlack (across from the USB Port)
5. Now power up the board with the usb cable

6. When prompted to login, enter `root` as the user login
7. The led should blink the message