Linux Device Driver Porting for NEEVEE Peripheral Hat

Peripheral Hat

- Hat contains huge set of peripherals below,
  1. RGB LED
  2. BH1745 Light Sensor
  3. MMA8652 Accelerometer
  4. MCP4725 DAC Converter
  5. SI7006 Temperature & Humidity Sensor
  6. TCA8418 Keypad Controller with 4x4 Keyboard matrix

Install an operating system

- Most Raspberry Pi users choose microSD cards as their boot device.
  - Installing an operating system using Raspberry pi Imager oot the board

Building the kernel

1. Download kernel source

- To get the kernel source, you need Git

```
$ sudo apt install git
```

- Next, download the source code for the latest Raspberry Pi kernel

```
$ git clone --depth=1 https://github.com/raspberrypi/linux
```

- Now that you have the kernel source, build a fresh kernel **natively**

2. Natively build a kernel

- First, install the build dependencies

```
$ sudo apt install bc bison flex libssl-dev make
```

Build configuration

- To prepare the default configuration, run the appropriate commands from the table below for your Raspberry Pi model.

Architecture	Model	Command

64-bit	Raspberry Pi 4	$ cd linux
		
		$ KERNEL=kernel8
		
		$ make bcm2711_defconfig

32-bit	Raspberry Pi 4	$ cd linux
		
		$ KERNEL=kernel7l
		
		$ make bcm2711_defconfig

Build

- Next, build the kernel.
- Run the following commands to build a 64-bit kernel

```
$ make -j6 Image.gz modules dtbs
```

- Run the following command to build a 32-bit kernel

```
$ make -j6 zImage modules dtbs
```

On multi-core Raspberry Pi models, the **make -j<n>** option distributes work between cores. This can speed up compilation significantly. Run **nproc** to see how many processors you have; we recommend passing a number 1.5x your number of processors.

Install the kernel

- Install the kernel modules onto the boot media

```
$ sudo make -j6 modules_install
```

- Then, install the kernel and Device Tree blobs into the boot partition

Device Tree Overlay

- Download the device tree from the overlays section
- Then compile the device tree in raspberry pi
- Use below command to compile

```
$ dtc -@ -I dts -O dtb -o ldd-neevee-rpi-hat-overlay.dtbo ldd-neevee-rpi-hat-
overlay.dts
```

- Copy the device tree binary *.dtbo and paste it in /boot/firmware/overlays

```
$ sudo cp ldd-neevee-rpi-hat-overlay.dtbo /boot/firmware/overlays/
```

- Open the config.txt file and add the overlay inside the file and save the file

```
# LDD NEEVEE RPI Hat
dtoverlay=ldd-neevee-rpi-hat-overlay
```

- Then now we need to enable the config for the drivers in menuconfig
- Open the **menuconfig** using

```
$ cd linux
$ make menuconfig
```

- Search for the config name like..

```
CONFIG_KEYBOARD_TCA8418, CONFIG_MMA8452, CONFIG_MCP4725,CONFIG_SI7020
```

- Enable the config manually then exit from the menuconfig
- Now compile the kernel source

```
$ make
```

- Install the kernel modules onto the boot media

```
$ sudo make -j6 modules_install
```

- Then reboot the raspberry pi

Testing the Hat

RGB led

- The RGB leds can be controlled by sysfs leds

```
$ cd /sys/class/leds
$ ls -l
```

- Inside the directory we can list and detect the RGB leds as the red, green and blue
- Initally the led is in off state as mentioned in the device tree
- To test the leds use the brightness property

```
$ cd /sys/class/leds/red
$ ls -l
```

- It will reault the properties of the led
- Among the listed properties, there is **brightness** property which is used to control the led

```
$ cd /sys/class/leds/red
$ echo 0 > brightness
$ echo 1 > brightness
```

- Similar for the Green and Blue leds

I2C sensors

- For I2C devices it will connected via **i2c1** of the raspberry pi
- Use the i2c-tools to find the connected device

```
$ i2cdetect -y 1
```

- The result will looks like

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- UU -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- 34 -- -- -- UU -- -- -- -- -- -- --
40: UU -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- 64 -- -- -- -- -- -- -- -- -- -- --
```

```
70: -- -- -- -- -- -- -- --
```

- In the result we can see
  - Some looks like **UU**, which means the driver is lodded succesfully
  - Some will looks like **address(34)**, which means we have to load the driver

```
$ modprobe mma8452
```

IIO devices

- In NEEVEE Rpi_Hat BH1745, MMA8652, SI7006, MCP4725 are confgiured as the IIO device
- To test these devices got to

```
$ /sys/bus/iio/devices
$ ls -l
```

- This will result the available devices
- Go to the device

```
$ cd iio\:device1/
$ ls -l
```

- This will result the available properties, use the raw properties to test the sensor

TCA8418-Keypad

- TCA8418 Keypad Controller with 4x4 Keyboard matrix
- This will detect as the input event device
- To find which event is used by keypad is

```
$ sudo dmesg | grep tca
```

- Also we can find using

```
$ ls -l /sys/class/i2c-dev/i2c-1/device/1-0034/input/input4/
```

- This will result the event
- To test the Keypad working use the user-application to read the Keypad data

```
keypad-test.c
```

- Open the file and replace the respective event
- Compile the .c file and run the binary then press the keys in the Keypad it will result the appropriate