Lesson-4: Embedded Linux Device Drivers: User-Mode (Assignment 4)

Command - sudo apt-cache search gpio

```
cxr1020@raspberrypi:~/Documents/EmbeddedLinuxRepo/CSR-EmbeddedLinux/Assignment04-Code $ sudo apt-cache search gpio
airspyhf - HF+VHF software defined radio receiver - user runtime
ledmon - Enclosure LED Utilities
gpiod - Tools for interacting with Linux GPIO character device - binary
libgpiod-dev - C library for interacting with Linux GPIO device - static libraries and headers
libgpiod-doc - C library for interacting with Linux GPIO device - library documentation
libgpiod2 - C library for interacting with Linux GPIO device - shared libraries
python3-libgpiod - Python bindings for libgpiod (Python 3)
libpigpiod-if2-1 - Client library for Raspberry Pi GPIO control
python-periphery-doc - Peripheral I/O (Documentation)
python3-periphery - Peripheral I/O (Python3 version)
python3-rpi.gpio - Module to control Raspberry Pi GPIO channels (Python 3)
rpi.gpio-common - Module to control Raspberry Pi GPIO channels (common files)
stm32flash - STM32 chip flashing utility using a serial bootloader
svxlink-gpio - GPIO control scripts SvxLink amateur radio server
libpigpio-dev - Client tools for Raspberry Pi GPIO control
libpigpio1 - Library for Raspberry Pi GPIO control
libpigpiod-if-dev - Development headers for client libraries for Raspberry Pi GPIO control
libpigpiod-if1 - Client library for Raspberry Pi GPIO control (deprecated)
pigpio - Raspberry Pi GPIO control transitional package.
pigpio-tools - Client tools for Raspberry Pi GPIO control
pigpiod - Client tools for Raspberry Pi GPIO control
python-gpiozero - Simple API for controlling devices attached to a Pi's GPIO pins.
python-gpiozero-doc - Simple API for controlling devices attached to a Pi's GPIO pins.
python-pigpio - Python module which talks to the pigpio daemon (Python 2)
python3-gpiozero - Simple API for controlling devices attached to a Pi's GPIO pins.
python3-pigpio - Python module which talks to the pigpio daemon (Python 3)
raspi-gpio - Dump the state of the BCM270x GPIOs
raspi-gpio-dbgsym - debug symbols for raspi-gpio
cxr1020@raspberrypi:~/Documents/EmbeddedLinuxRepo/CSR-EmbeddedLinux/Assignment04-Code $
```

Command - dpkg -L libpigpio-dev

```
cxr1020@raspberrypi:~/Documents/EmbeddedLinuxRepo/CSR-EmbeddedLinux/Assignment04-Code $ dpkg -L libpigpio-dev
/.
/usr
/usr/include
/usr/include/pigpio.h
/usr/lib
/usr/share
/usr/share/doc
/usr/share/doc/libpigpio-dev
/usr/share/doc/libpigpio-dev/changelog.Debian.gz
/usr/share/doc/libpigpio-dev/copyright
/usr/share/man
/usr/share/man/man3
/usr/share/man/man3/pigpio.3.gz
/usr/lib/libpigpio.so
cxr1020@raspberrypi:~/Documents/EmbeddedLinuxRepo/CSR-EmbeddedLinux/Assignment04-Code $
```

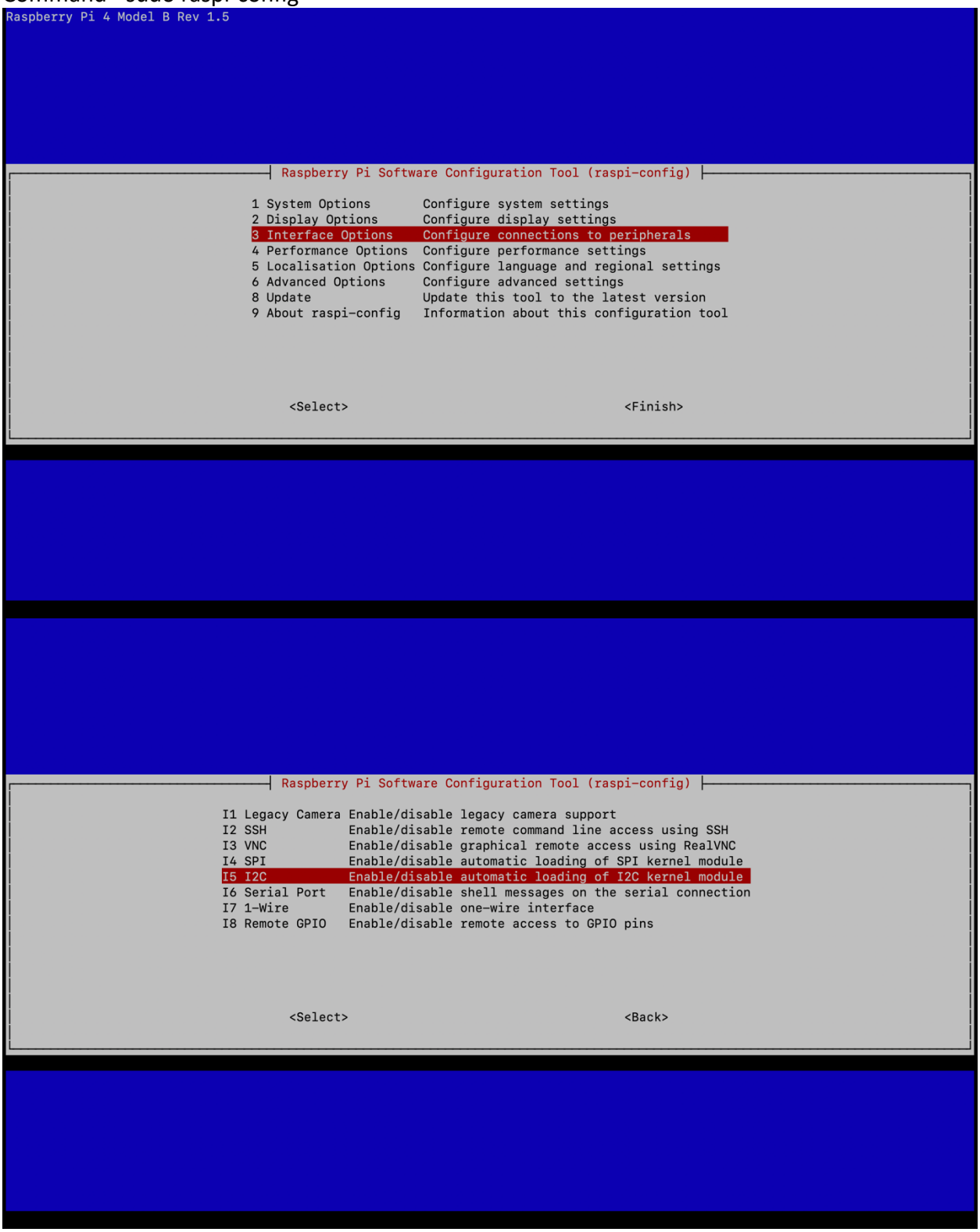
SourceCode – led21-blink.c

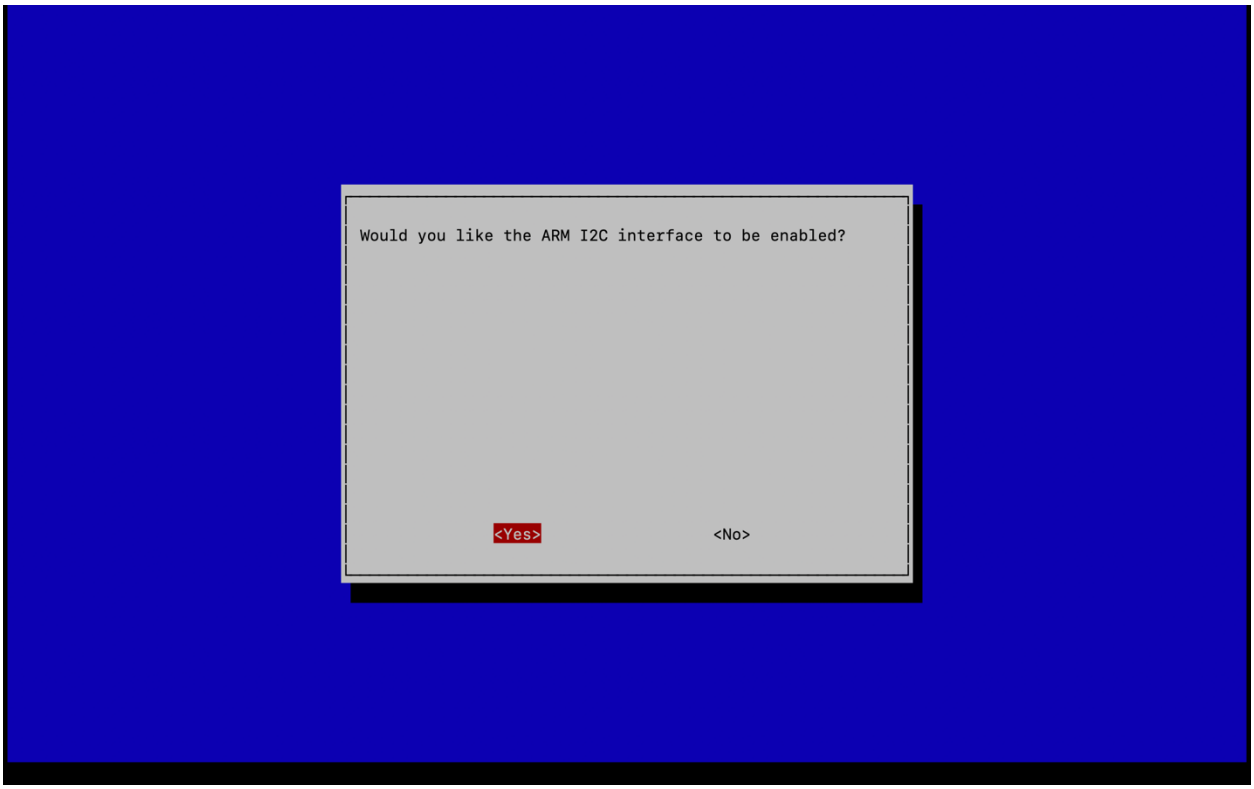
```
  GNU nano 5.4                                led21-blink.c
/* Connect Led to GPIO 21 and make it blink */


#include <stdio.h>
#include <pigpio.h>
#include <signal.h>
#include <unistd.h>

#define LED_PIN 21

int running = 1;
void handle_sig_int(int sig)
{
        running = 0;
}

int main()
{
        int result = gpioInitialise();
        if(result<0)
        {
                fprintf(stderr, "gpioInitialise failed\n");
                result  = 1;
                goto getOut;
        }

        result = gpioSetMode(LED_PIN, PI_OUTPUT);
        if(result<0)
        {
                fprintf(stderr,"gpioSetMode() failed\n");
                result = 2;
                goto getOut;
        }
        // We need to use Signals
                                [ Read 55 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^_ Go To Line   M-E Redo        M-6 Copy
```

SourceCode – led21-blink.c

```
  GNU nano 5.4                                i2c-blink.c
/* i2c-blink.c
 * blink leds connected to i2c device
 * TO BUILD: gcc -Wall -o i2c-blink i2c-blink.c -lpigpio -lrt
 * TO RUN:  sudo ./i2c-blink
*/

#include <stdio.h>
#include <pigpio.h>
#include <signal.h>
#include <unistd.h>

#define MCP23008_I2C_ADDR 0x20

int running = 1;
void handle_sig_int(int sig)
{
        running = 0;
}

int main()
{
        int result = gpioInitialise();

        if(result < 0)
        {
                fprintf(stderr, "gpioInitialise() failed\n");
                result = 1;
                goto getOut;

        }

        // We need to use signals
        int cfg = gpioCfgGetInternals();
        cfg |= PI_CFG_NOSIGHANDLER;
                                [ Read 77 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^_ Go To Line   M-E Redo        M-6 Copy
```

Command - sudo raspi-config





Command - lsmod | head

```
cxr1020@raspberrypi:~/Documents/EmbeddedLinuxRepo/CSR-EmbeddedLinux/Assignment04-Code/i2c $ sudo raspi-config
cxr1020@raspberrypi:~/Documents/EmbeddedLinuxRepo/CSR-EmbeddedLinux/Assignment04-Code/i2c $ lsmod | head
Module                  Size  Used by
i2c_bcm2835            16384  0
rfcomm                 53248  0
cmac                   16384  0
algif_hash             16384  0
aes_arm64              16384  0
aes_generic            36864  1 aes_arm64
algif_skcipher         16384  0
af_alg                 28672  2 algif_hash,algif_skcipher
bnep                   24576  0
cxr1020@raspberrypi:~/Documents/EmbeddedLinuxRepo/CSR-EmbeddedLinux/Assignment04-Code/i2c $
```

Output - i2c-blink