



TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

Môn học: Hệ thống nhúng

THỰC HIỆN ỨNG DỤNG XỬ LÝ NGẮT

GV : PGS.TS Võ Minh Huân

SVTH : Nguyễn Hoàng Diễm Chi – 22161004

Huỳnh Trang Vĩnh San – 22161029

Nguyễn Thanh Tuyền – 22161042

Nguyễn Đình Khánh Vy – 22161043

Lê Kỳ Tuân – 22161204

❖ **Bảng phân công**

Họ và tên	Công việc
Nguyễn Đình Khánh Vy 22161043	<ul style="list-style-type: none">- Tìm hiểu và code sản phẩm- Quay video demo sản phẩm
Nguyễn Hoàng Diễm Chi 22161004	<ul style="list-style-type: none">- Cài đặt img và boot thẻ nhớ SD- Làm powerpoint
Huỳnh Trang Vĩnh San 22161029	<ul style="list-style-type: none">- Xây dựng lưu đồ giải thuật- Tìm hiểu và code sản phẩm
Nguyễn Thanh Tuyền 22161042	<ul style="list-style-type: none">- Nhập nội dung vào terminal- Chuẩn bị môi trường làm việc cho Raspberry Pi
Lê Kỳ Tuân 22161204	<ul style="list-style-type: none">- Tìm hiểu cách kết nối phần cứng- Làm powerpoint



NỘI DUNG

1. Phần cứng, xử lý ngắt trên device drive (Handling interrupt)

- a.** Character device: Major number và minor number
- b.** IRQ number, ISR

2. Thực nghiệm

- a.** Mô tả thực nghiệm
- b.** Linh kiện
- c.** Sơ đồ mạch
- d.** Các bước làm và kiểm tra
- e.** Lưu đồ giải thuật
- f.** Code và giải thích code



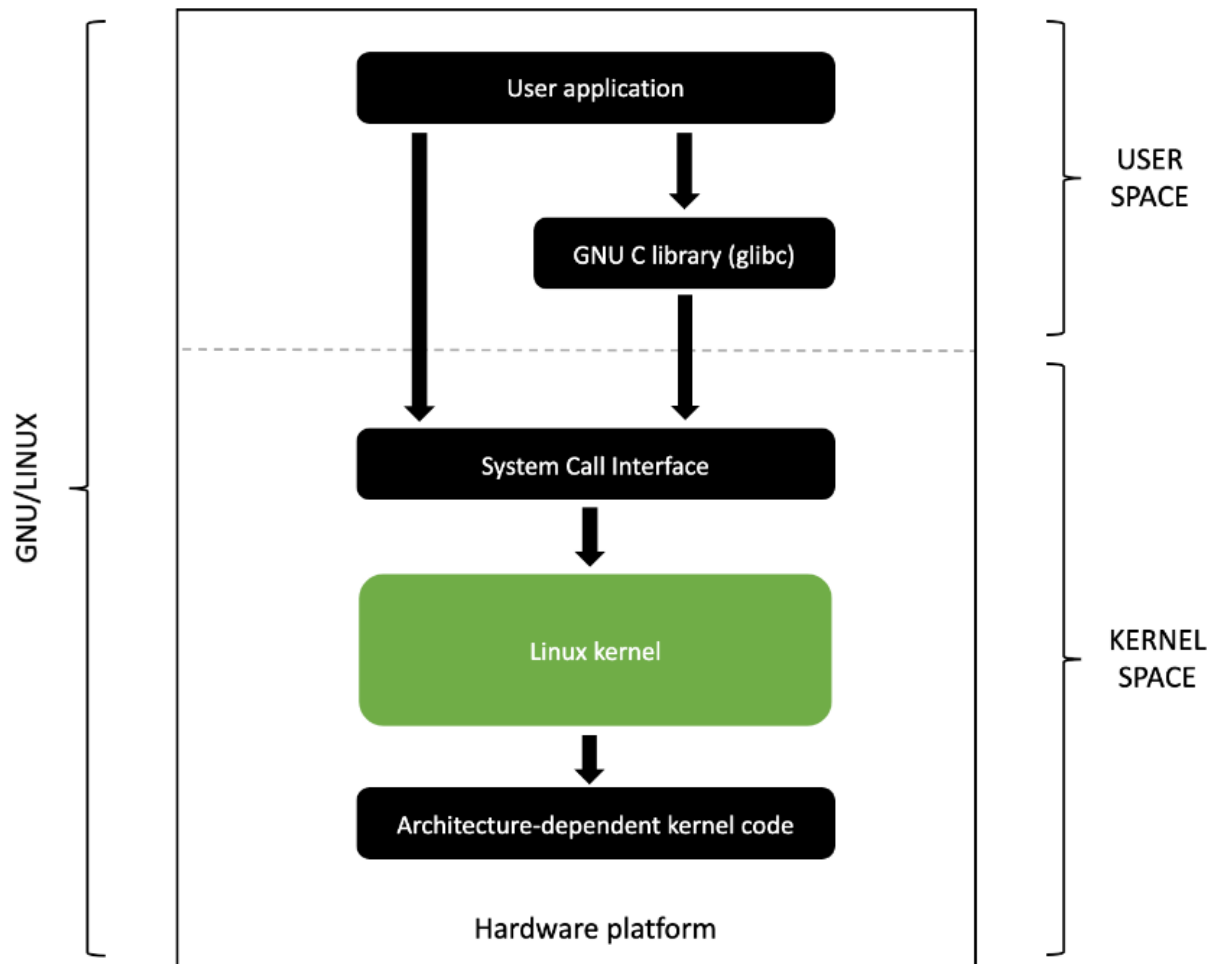
HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

1. Phần cứng, xử lý ngắt trên device drive (Handling interrupt):



Sơ đồ mối quan hệ giữa user space và kernel space



a. Character device: Major number và minor number :

- **Character device:** loại thiết bị I/O trao đổi dữ liệu từng byte/ký tự thay vì theo khối
- **Major number:** xác định driver liên kết với thiết bị
- **Minor number:** phân biệt các device sử dụng chung một driver

```
danguyet@raspberrypi: ~  
File Edit Tabs Help  
danguyet@raspberrypi:~$ ls -l /dev  
total 0  
crw-r--r-- 1 root root 10, 235 Nov 5 15:51 autofs  
drwxr-xr-x 2 root root 580 Nov 5 15:51 block  
crw----- 1 root root 10, 234 Nov 5 03:17 btrfs-control  
drwxr-xr-x 3 root root 60 Jan 1 1970 bus  
crw----- 1 root root 10, 63 Nov 5 15:51 cachefiles  
crw-rw---- 1 root video 238, 0 Nov 5 15:51 cec0  
drwxr-xr-x 2 root root 3060 Nov 7 08:37 char  
crw--w---- 1 root tty 5, 1 Nov 5 15:51 console  
crw----- 1 root root 10, 62 Nov 5 15:51 cpu_dma_latency  
crw----- 1 root root 10, 203 Nov 5 03:17 cuse  
drwxr-xr-x 7 root root 140 Nov 5 03:17 disk  
drwxr-xr-x 2 root root 100 Nov 5 15:51 dma_heap  
drwxr-xr-x 3 root root 100 Nov 5 15:51 dri  
lrwxrwxrwx 1 root root 13 Sep 6 09:26 fd -> /proc/self/fd  
crw-rw-rw- 1 root root 1, 7 Nov 5 15:51 full  
crw-rw-rw- 1 root root 10, 229 Nov 5 15:51 fuse  
crw-rw----+ 1 root gpio 254, 0 Nov 5 03:17 gpiochip0  
crw-rw----+ 1 root gpio 254, 1 Nov 5 03:17 gpiochip1  
crw-rw----+ 1 root gpio 254, 2 Nov 5 03:17 gpiochip2  
lrwxrwxrwx 1 root root 9 Nov 5 03:17 gpiochip4 -> gpiochip0  
crw-rw---- 1 root gpio 245, 0 Nov 5 03:17 gpiomem  
crw----- 1 root root 10, 183 Nov 5 15:51 hwrng  
crw-rw---- 1 root i2c 89, 1 Nov 5 03:17 i2c-1  
crw-rw---- 1 root i2c 89, 2 Nov 5 03:17 i2c-2  
lrwxrwxrwx 1 root root 12 Sep 6 09:26 initctl -> /run/initctl  
drwxr-xr-x 2 root root 60 Jan 1 1970 input  
crw-r--r-- 1 root root 1, 11 Nov 5 15:51 kmsg  
crw----- 1 root root 237, 0 Nov 7 08:37 led_seq  
lrwxrwxrwx 1 root root 28 Sep 6 09:26 log -> /run/systemd/journal/dev-log  
brw-rw---- 1 root disk 7, 0 Nov 5 03:17 loop0  
brw-rw---- 1 root disk 7, 1 Nov 5 03:17 loop1  
brw-rw---- 1 root disk 7, 2 Nov 5 03:17 loop2  
brw-rw---- 1 root disk 7, 3 Nov 5 03:17 loop3
```




b. IRQ number, ISR:

- IRQ number (Interrupt Request Number):

- + Số hiệu các yêu cầu ngắt giúp các thiết bị ngoại vi
- + Thông báo với bộ xử lý trung tâm (CPU) chúng cần được xử lý mà không cần chờ đợi CPU hoàn thành các tác vụ hiện tại
- + Mỗi thiết bị được gán một số hiệu IRQ riêng

- ISR (Interrupt Service Routine):

- + Được gọi là: "trình phục vụ ngắt" hoặc "trình xử lý ngắt"
- + Đoạn mã chương trình xử lý tín hiệu ngắt



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

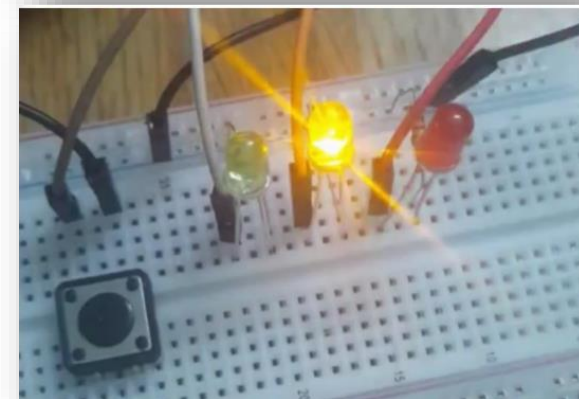
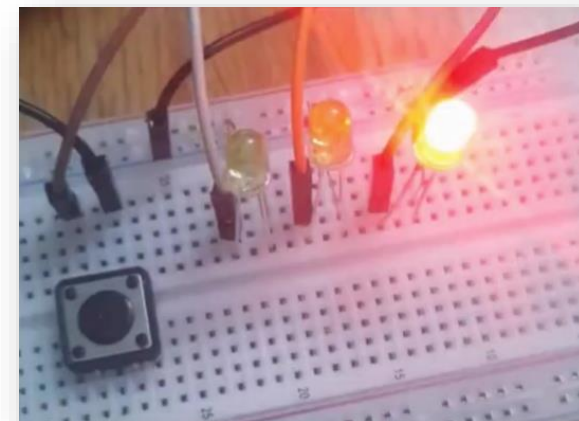
HCMC University of Technology and Education

2. Thực nghiệm:

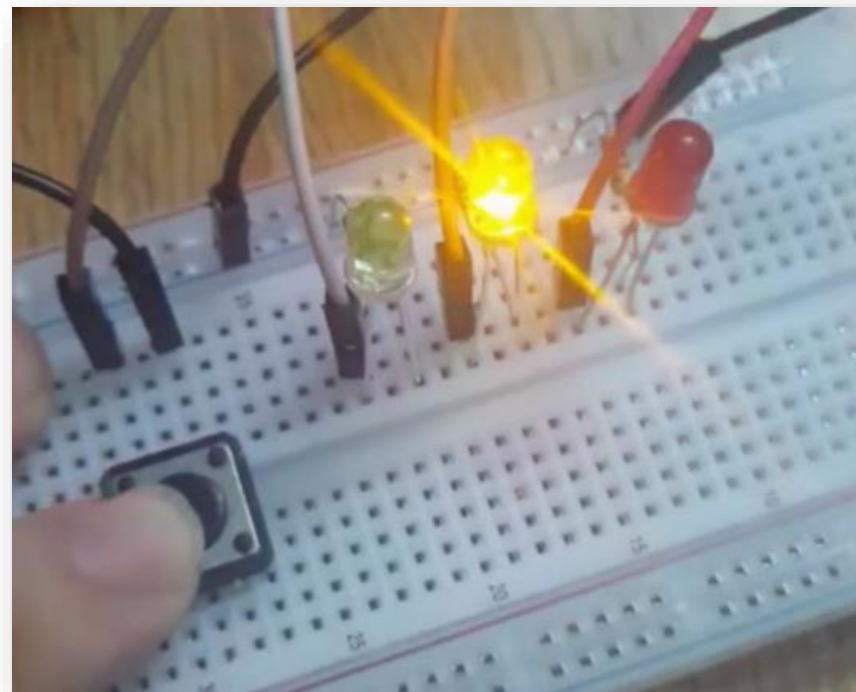
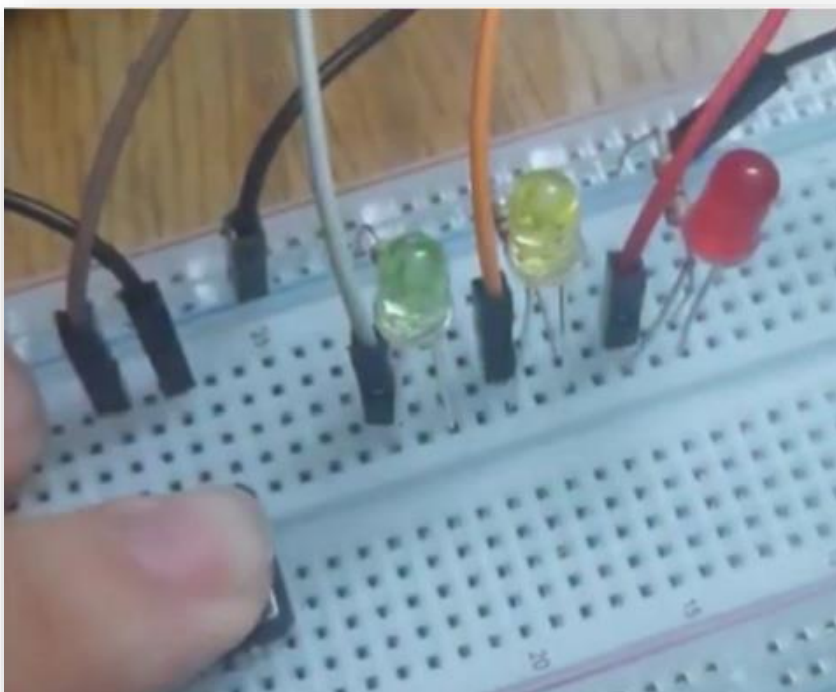
a. Mô tả:

Nhập “start” vào user space, 3 led sáng tuần tự

```
danguyet@raspberrypi: ~  
File Edit Tabs Help  
cpu_dma_latency log ppp raw tty16 tty34 tty52 uinput vcsa6 watchdog  
cuse loop0 ptmx rfkill tty17 tty35 tty53 urandom vcsa7 watchdog0  
disk loop1 pts serial0 tty18 tty36 tty54 v4l vcsu zero  
dma_heap loop2 ram0 serial1 tty19 tty37 tty55 vchiq vcsu  
dri loop3 ram1 shm tty2 tty38 tty56 vcio vcsu1  
fd loop4 ram10 snd tty20 tty39 tty57 vc-mem vcsu2  
full loop5 ram11 spidev0.0 tty21 tty4 tty58 vcs vcsu3  
fuse loop6 ram12 spidev0.1 tty22 tty40 tty59 vcs1 vcsu4  
gpiochip0 loop7 ram13 stderr tty23 tty41 tty6 vcs2 vcsu5  
gpiochip1 loop-control ram14 stdin tty24 tty42 tty60 vcs3 vcsu6  
gpiochip2 mapper ram15 stdout tty25 tty43 tty61 vcs4 vcsu7  
gpiochip4 media0 ram2 tty tty26 tty44 tty62 vcs5 vcsu7  
danguyet@raspberrypi:~$ ls /sys/class  
bcm2708_vcio dma_heap ieee80211 leds power_supply scsi_disk uio  
bcm2835-gpiomem drm input led_seq pwm scsi_host vc  
bdi gpio iscsi_connection lirc raw sound vchiq  
block graphics iscsi_endpoint mdio_bus rc spi_dev vc-mem  
bluetooth hidraw iscsi_host mem regulator spi_master vc-sm  
bsg hwmon iscsi_iface misc rfkill spi_slave video4linux  
devcoredump i2c-adapter iscsi_session mmc_host rtc thermal vtconsole  
dma i2c-dev iscsi_transport net scsi_device tty watchdog  
danguyet@raspberrypi:~$ sudo chmod 777 /dev/led_seq  
danguyet@raspberrypi:~$ echo "start" > /dev/led_seq  
danguyet@raspberrypi:~$
```



Khi nhấn nút nhấn, 3 led tắt. Thả nhấn 3 led sáng tuần tự bình thường



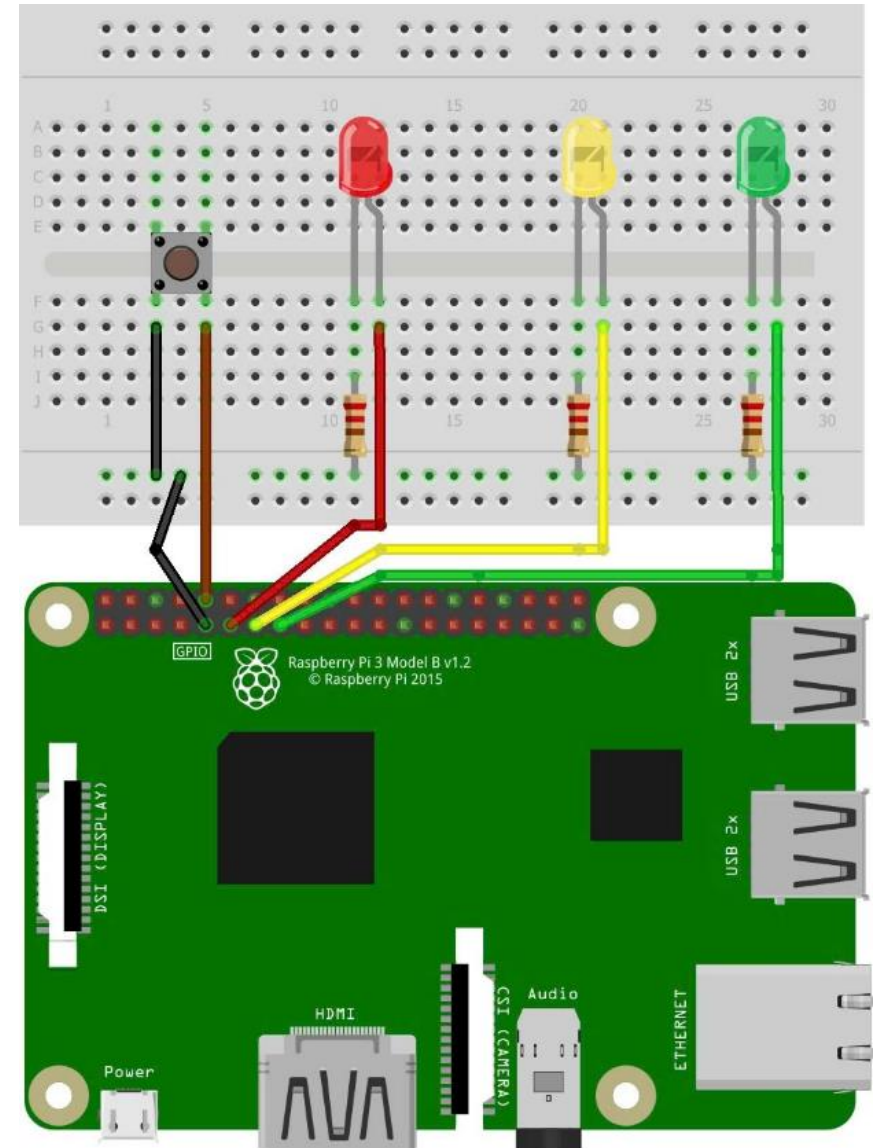
Nhập “stop” vào user space thì 3 led tắt

```
danguyet@raspberrypi: ~  
File Edit Tabs Help  
cuse      loop0      ptmx      rfkill    tty17     tty35     tty53     urandom   vcsa7     watchdog0  
disk      loop1      pts       serial0   tty18     tty36     tty54     v4l       vcsm      zero  
dma_heap  loop2      ram0      serial1   tty19     tty37     tty55     vchiq     vcsm-cma  
dri       loop3      ram1      shm       tty2      tty38     tty56     vcio      vcsu  
fd        loop4      ram10     snd       tty20     tty39     tty57     vc-mem    vcsu1  
full      loop5      ram11     spidev0.0 tty21     tty4      tty58     vcs       vcsu2  
fuse      loop6      ram12     spidev0.1 tty22     tty40     tty59     vcs1      vcsu3  
gpiochip0 loop7      ram13     stderr    tty23     tty41     tty6      vcs2      vcsu4  
gpiochip1 loop-control ram14     stdin     tty24     tty42     tty60     vcs3      vcsu5  
gpiochip2 mapper     ram15     stdout    tty25     tty43     tty61     vcs4      vcsu6  
gpiochip4 media0     ram2      tty       tty26     tty44     tty62     vcs5      vcsu7  
danguyet@raspberrypi:~ $ ls /sys/class  
bcm2708_vcio  dma_heap  ieee80211  leds      power_supply  scsi_disk  uio  
bcm2835-gpiomem  drm       input      led_seq    pwm           scsi_host  vc  
bdi           gpio      iscsi_connection  lirc       raw           sound      vchiq  
block         graphics  iscsi_endpoint  mdio_bus   rc            spidev     vc-mem  
bluetooth     hidraw    iscsi_host      mem        regulator     spi_master  vc-sm  
bsg           hwmon     iscsi_iface     misc       rfkill        spi_slave   video4linux  
devcoredump   i2c-adapter iscsi_session  mmc_host   rtc           thermal     vtconsole  
dma           i2c-dev   iscsi_transport net         scsi_device  tty         watchdog  
danguyet@raspberrypi:~ $ sudo chmod 777 /dev/led_seq  
danguyet@raspberrypi:~ $ echo "start" > /dev/led_seq  
danguyet@raspberrypi:~ $ echo "stop" > /dev/led_seq  
danguyet@raspberrypi:~ $
```



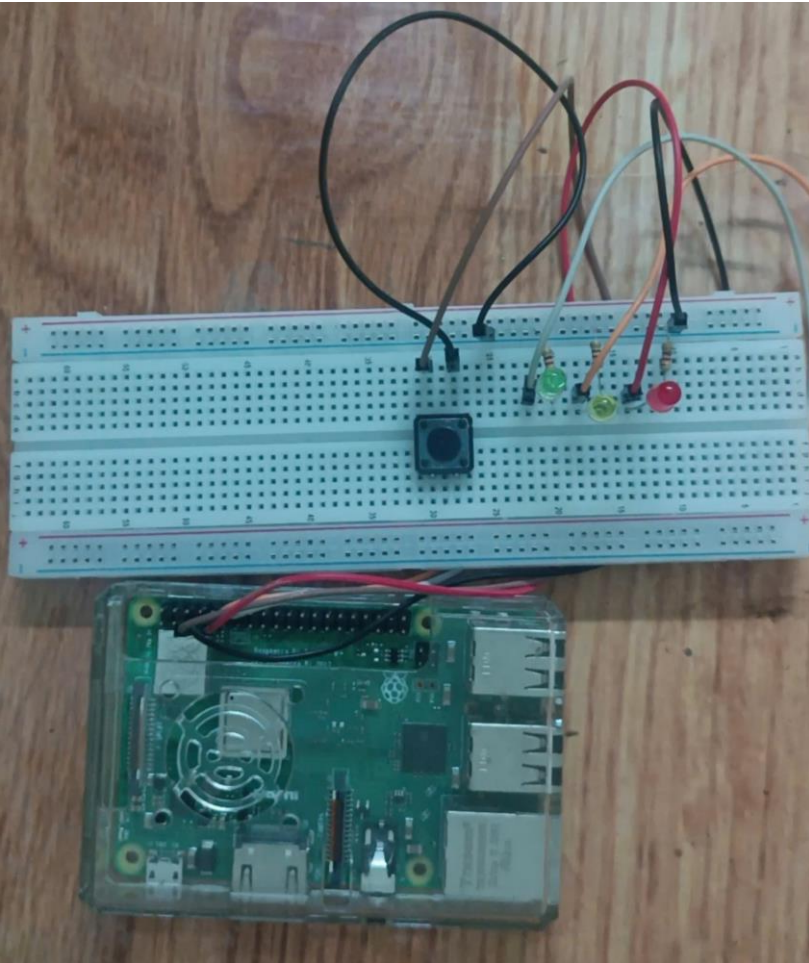
b. Linh kiện:

Tên	Số lượng
Led	3
Raspberry Pi 3 model B+	1
Điện trở 220Ω	3
Nút nhấn	1
Dây dẫn	7



c. Sơ đồ mạch:

- Led đỏ (+) → GPIO 17
- Led đỏ (-) → Điện trở → Board
- Led vàng (+) → GPIO 27
- Led vàng (-) → Điện trở → Broad
- Led xanh (+) → GPIO 22
- Led xanh (-) → Điện trở → Broad
- Button chân 1 → Board
- Button chân 2 → GPIO 4

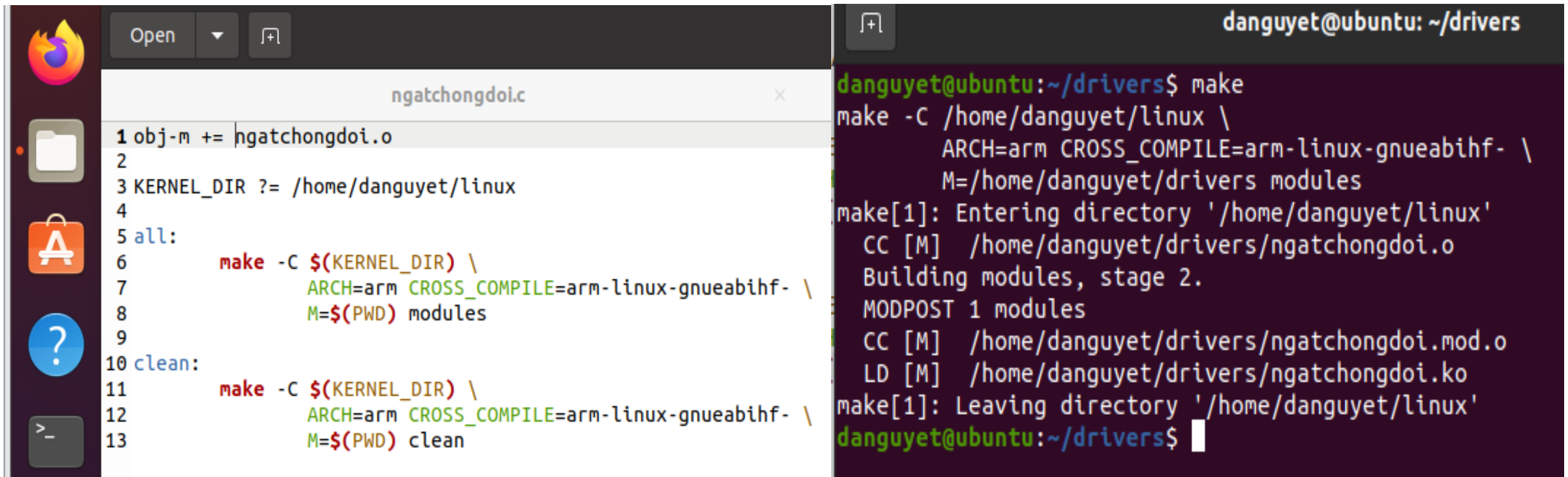


3v3 Power	1			2	5v Power
GPIO 2 (I2C1 SDA)	3			4	5v Power
GPIO 3 (I2C1 SCL)	5			6	Ground
GPIO 4 (GPCLK0)	7			8	GPIO 14 (UART TX)
Ground	9			10	GPIO 15 (UART RX)
GPIO 17	11			12	GPIO 18 (PCM CLK)
GPIO 27	13			14	Ground
GPIO 22	15			16	GPIO 23
3v3 Power	17			18	GPIO 24
GPIO 10 (SPI0 MOSI)	19			20	Ground
GPIO 9 (SPI0 MISO)	21			22	GPIO 25
GPIO 11 (SPI0 SCLK)	23			24	GPIO 8 (SPI0 CE0)
Ground	25			26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27			28	GPIO 1 (EEPROM SCL)
GPIO 5	29			30	Ground
GPIO 6	31			32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33			34	Ground
GPIO 19 (PCM FS)	35			36	GPIO 16
GPIO 26	37			38	GPIO 20 (PCM DIN)
Ground	39			40	GPIO 21 (PCM DOUT)



d. Các bước làm và kiểm tra:

B1: make file từ ubuntu sau đó truyền file .ko sang Raspberry Pi



```
ngatchongdoi.c
1 obj-m += ngatchongdoi.o
2
3 KERNEL_DIR ?= /home/danguyet/linux
4
5 all:
6     make -C $(KERNEL_DIR) \
7         ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- \
8         M=$(PWD) modules
9
10 clean:
11     make -C $(KERNEL_DIR) \
12         ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- \
13         M=$(PWD) clean
```

```
danguyet@ubuntu: ~/drivers
danguyet@ubuntu:~/drivers$ make
make -C /home/danguyet/linux \
    ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- \
    M=/home/danguyet/drivers modules
make[1]: Entering directory '/home/danguyet/linux'
CC [M] /home/danguyet/drivers/ngatchongdoi.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/danguyet/drivers/ngatchongdoi.mod.o
LD [M] /home/danguyet/drivers/ngatchongdoi.ko
make[1]: Leaving directory '/home/danguyet/linux'
danguyet@ubuntu:~/drivers$
```




HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

B2: vào thư mục chứa file .ko -> insmod -> dmesg

\$cd /home/danguyet/Desktop/drivers

\$sudo insmod ngatchongdoi.ko
\$dmesg

```
danguyet@raspberrypi: ~/Desktop/drivers
File Edit Tabs Help
danguyet@raspberrypi:~/Desktop/drivers $
```

```
[ 26.437803] Bluetooth: BNEP filters: protocol multicast
[ 26.437828] Bluetooth: BNEP socket layer initialized
[ 26.544111] NET: Registered protocol family 38
[ 26.629801] cryptd: max_cpu_qlen set to 1000
[ 29.824004] vc4_hdmi 3f902000.hdmi: ASoC: can't open interface 3f902000.hdmi:
-19
[ 29.824993] vc4_hdmi 3f902000.hdmi: ASoC: can't open interface 3f902000.hdmi:
-19
[ 29.825899] vc4_hdmi 3f902000.hdmi: ASoC: can't open interface 3f902000.hdmi:
-19
[ 29.842749] vc4_hdmi 3f902000.hdmi: ASoC: can't open interface 3f902000.hdmi:
-19
[ 29.843322] vc4_hdmi 3f902000.hdmi: ASoC: can't open interface 3f902000.hdmi:
-19
[ 29.843863] vc4_hdmi 3f902000.hdmi: ASoC: can't open interface 3f902000.hdmi:
-19
[ 29.849648] vc4_hdmi 3f902000.hdmi: ASoC: can't open interface 3f902000.hdmi:
-19
[ 29.929963] Bluetooth: RFCOMM TTY layer initialized
[ 29.929990] Bluetooth: RFCOMM socket layer initialized
[ 29.930025] Bluetooth: RFCOMM ver 1.11
[ 104.479484] ngatchongdoi: loading out-of-tree module taints kernel.
[ 104.480598] LED driver initialized
danguyet@raspberrypi:~/Desktop/drivers $
```

=> In ra thông báo module đã được nạp



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

B3: Kiểm tra module ngatchongdoi đã được nạp vào hệ thống chưa

\$lsmod

```
danguyet@raspberrypi:~$ lsmod
Module                  Size  Used by
ngatchongdoi            16384  0
rfcomm                  49152  4
snd_seq_dummy           16384  0
snd_hrtimer             16384  1
snd_seq                 69632  7 snd_seq_dummy
snd_seq_device          16384  1 snd_seq
cmac                    16384  2
algif_hash              16384  1
aes_arm_bs              24576  2
crypto_simd             16384  1 aes_arm_bs
cryptd                  24576  2 crypto_simd
algif_skcipher          16384  1
af_alg                  28672  6 algif_hash,algif_skcipher
bnep                    20480  2
hci_uart                40960  1
btbcm                   16384  1 hci_uart
bluetooth               372736 33 hci_uart,bnep,btbcm,rfcomm
ecdh_generic            16384  1 bluetooth
ecc                     40960  1 ecdh_generic
vc4                     241664  7
cec                     49152  1 vc4
brcmfmac                319488  0
drm_kms_helper          184320  2 vc4
brcmutil                20480  1 brcmfmac
snd_soc_core            200704  1 vc4
snd_compress            20480  1 snd_soc_core
snd_pcm_dmaengine       16384  1 snd_soc_core
syscopyarea             16384  1 drm_kms_helper
sysfillrect             16384  1 drm_kms_helper
sysimgblt               16384  1 drm_kms_helper
fb_sys_fops             16384  1 drm_kms_helper
raspberrypi_hwmon       16384  0
sha256_generic          16384  0
libsha256               20480  1 sha256_generic
```



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

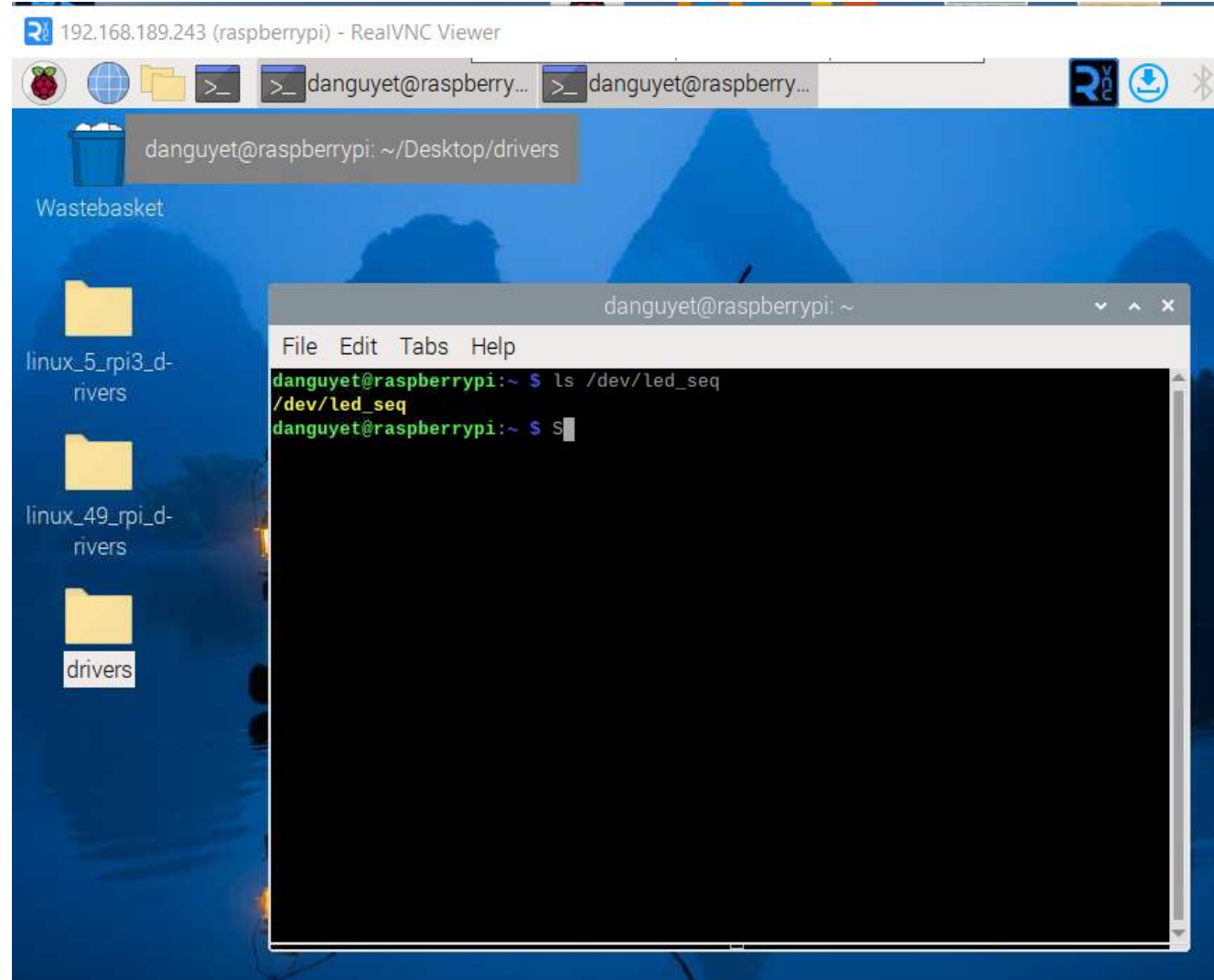
B4: kiểm tra có cài đặt ngắt được chưa

- Chân ngắt: chân 4
- Số IRQ (Interrupt Request Line) là số 167

```
danguyet@raspberrypi: ~  
File Edit Tabs Help  
64: 0 0 0 0 ARMCTRL-level 72 Edge vc4 hdmi cec  
66: 20 0 0 0 ARMCTRL-level 74 Edge vc4 crtc  
69: 0 0 0 0 ARMCTRL-level 77 Edge vc4 crtc  
70: 0 0 0 0 ARMCTRL-level 78 Edge vc4 crtc  
77: 180 0 0 0 ARMCTRL-level 85 Edge 3f804000.i2c, 3f805000.i2c  
78: 0 0 0 0 ARMCTRL-level 86 Edge 3f204000.spi  
80: 893 0 0 0 ARMCTRL-level 88 Edge mmc0  
81: 7867 0 0 0 ARMCTRL-level 89 Edge uart-pl011  
86: 31796 0 0 0 ARMCTRL-level 94 Edge mmc1  
161: 0 0 0 0 bcm2836-timer 0 Edge arch_timer  
162: 8329 5307 6362 5847 bcm2836-timer 1 Edge arch_timer  
165: 0 0 0 0 bcm2836-pmu 9 Edge arm-pmu  
166: 0 0 0 0 lan78xx-irqs 17 Edge usb-001:004:01  
167: 1 0 0 0 pinctrl-bcm2835 4 Edge button_irq  
FIQ: usb_fiq  
IPI0: 0 0 0 0 CPU wakeup interrupts  
IPI1: 0 0 0 0 Timer broadcast interrupts  
IPI2: 11513 20248 11552 18683 Rescheduling interrupts  
IPI3: 728 1797 1816 1266 Function call interrupts  
IPI4: 0 0 0 0 CPU stop interrupts  
IPI5: 370 103 74 82 IRQ work interrupts  
IPI6: 0 0 0 0 completion interrupts  
Err: 0  
danguyet@raspberrypi:~ $
```

B5: kiểm tra thiết bị nào được tạo ra với tên `led_seq` trong thư mục `/dev`.

- Nếu module đã được khởi tạo thành công, thiết bị được tạo sẽ thấy trong danh sách.
- Do tên thiết bị đặt là “`led_seq`” nên câu lệnh kiểm tra:
`$ls /dev/led_seq`



The screenshot shows a Raspberry Pi desktop environment. The desktop has a blue background with a mountain silhouette. There are several icons on the desktop: a wastebasket, a folder named 'linux_5_rpi3_drivers', a folder named 'linux_49_rpi_drivers', and a folder named 'drivers'. A terminal window is open in the foreground, displaying the command `ls /dev/led_seq` and the output `/dev/led_seq`. The terminal window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The desktop environment is titled 'danguyet@raspberrypi: ~/Desktop/drivers'.



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

B6: kiểm tra file thiết bị được tạo

\$ls /sys/class

```
192.168.189.243 (raspberrypi) - RealVNC Viewer
danguyet@raspberrypi: ~
File Edit Tabs Help
danguyet@raspberrypi:~ $ ls /sys/class
bcm2708_vcio      dma      i2c-adapter  iscsi_iface  mem      rc      sound      vc
bcm2835-gpiomem  dma_heap i2c-dev      iscsi_session misc      regulator spidev     vchiq
bdi              drm      ieee80211    iscsi_transport mmc_host  rfkill    spi_master vc-mem
block            gpio     input        leds          net      rtc      spi_slave  vc-sm
bluetooth        graphics iscsi_connection led_seq       power_supply scsi_device thermal    video4linux
bsg              hidraw   iscsi_endpoint lirc          pwm      scsi_disk tty        vtconsole
devcoredump      hwmon    iscsi_host    mdio_bus      raw      scsi_host uio        watchdog
danguyet@raspberrypi:~ $ S
```



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

B7: Cấp quyền truy xuất vào tệp thiết bị

\$sudo chmod 777 /dev/led_seq

B8: dùng trình biên dịch gcc

192.168.113.243 (raspberrypi) - RealVNC Viewer

danguyet@raspberrypi: ~/Desktop/drivers

```
danguyet@raspberrypi:~/Desktop/drivers $ sudo nano blink_led.c
danguyet@raspberrypi:~/Desktop/drivers $ ls
blink_led.c  ngat.c  ngatchongdoi.mod.o  ngattuantu.mod.c  tuantu.c
device.c     ngatchongdoi.c  ngatchongdoi.o      ngattuantu.mod.o
Makefile     ngatchongdoi.ko ngattuantu.c         ngattuantu.o
modules.order ngatchongdoi.mod ngattuantu.ko       nutnhan.c
Module.symvers ngatchongdoi.mod.c ngattuantu.mod      sangtuantu.c
danguyet@raspberrypi:~/Desktop/drivers $ gcc blink_led.c
a.out
danguyet@raspberrypi:~/Desktop/drivers $ ls
a.out  Module.symvers  ngatchongdoi.mod.c  ngattuantu.mod.c  sangtuantu.c
blink_led.c  ngat.c  ngatchongdoi.mod.o  ngattuantu.mod.c  tuantu.c
device.c     ngatchongdoi.c  ngatchongdoi.o      ngattuantu.mod.o
Makefile     ngatchongdoi.ko ngattuantu.c         ngattuantu.o
modules.order ngatchongdoi.mod ngattuantu.ko       nutnhan.c
danguyet@raspberrypi:~/Desktop/drivers $ sudo insmod ngatchongdoi.ko
danguyet@raspberrypi:~/Desktop/drivers $ ./a.out
^C
danguyet@raspberrypi:~/Desktop/drivers $ sudo rmmod ngatchongdoi.ko
danguyet@raspberrypi:~/Desktop/drivers $
```



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

192.168.113.243 (raspberrypi) - RealVNC Viewer

danguyet@raspberrypi... danguyet@raspberrypi...

04:00

danguyet@raspberrypi: ~/Desktop/drivers

```
File Edit Tabs Help
GNU nano 7.2 blink_led.c
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

int main(void)
{
    int f = open("/dev/led_seq", O_RDWR);
    while (1)
    {
        write(f, "start", 5);
        usleep(500000000);
        usleep(500000000);
        usleep(500000000);
        write(f, "stop", 4);
        usleep(5000000);
    }
}
```

[Wrote 19 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

B9: Nhấn nút nhấn và thả, dmesg để xem kết quả

```
danguyet@raspberrypi: ~/Desktop/drivers
File Edit Tabs Help
[ 1854.517892] [<7f3ef000>] (led_seq_thread [ngatchongdoi]) from [<80144ac0>] (kthread+0x144/0x170)
[ 1854.517897] r5:b5e8a840 r4:b52ef500
[ 1854.517907] [<8014497c>] (kthread) from [<801010ac>] (ret_from_fork+0x14/0x28)
[ 1854.517913] Exception stack(0xb9ef5fb0 to 0xb9ef5ff8)
[ 1854.517920] 5fa0: 00000000 00000000 00000000 00000000
[ 1854.517930] 5fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
[ 1854.517938] 5fe0: 00000000 00000000 00000000 00000000 00000013 00000000
[ 1854.517947] r10:00000000 r9:00000000 r8:00000000 r7:00000000 r6:00000000 r5:8014497c
[ 1854.517952] r4:b5e8a840
[ 1854.517964] ---[ end trace f10274105b48c145 ]---
[ 1854.561892] Button released, resuming LED sequence
[ 2147.968287] LED driver removed
[ 2159.465122] LED driver initialized
[ 2206.625654] LED device opened
[ 2206.625730] LED sequence started
[ 2206.625747] LED device closed
[ 2216.088723] LED device opened
[ 2216.088776] LED sequence stopped
[ 2216.088796] LED device closed
[ 2323.028224] LED driver removed
[ 2884.090466] LED driver initialized
[ 2909.177097] LED device opened
[ 2909.177112] LED sequence started
[ 2958.498310] LED device opened
[ 2958.498335] LED sequence started
[ 3000.185789] Button pressed, turning off LEDs
[ 3000.206347] Button released, resuming LED sequence
[ 3000.218213] -----[ cut here ]-----
[ 3000.218239] WARNING: CPU: 3 PID: 3763 at drivers/gpio/gpio-lib.c:128 gpio_to_desc+0xb4/0xe4
[ 3000.218247] invalid GPIO 2131222548
```




HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

B10: Xóa module

\$sudo rmmod ngatchongdoi.ko

```
danguyet@raspberrypi: ~/Desktop/drivers
File Edit Tabs Help
[ 1854.517892] [<7f3ef000>] (led_seq_thread [ngatchongdoi]) from [<80144ac0>] (kthread+0x144/0x170)
[ 1854.517897] r5:b5e8a840 r4:b52ef500
[ 1854.517907] [<8014497c>] (kthread) from [<801010ac>] (ret_from_fork+0x14/0x28)
[ 1854.517913] Exception stack(0xb9ef5fb0 to 0xb9ef5ff8)
[ 1854.517920] 5fa0: 00000000 00000000 00000000 00000000
[ 1854.517930] 5fc0: 00000000 00000000 00000000 00000000
[ 1854.517938] 5fe0: 00000000 00000000 00000000 00000013
[ 1854.517947] r10:00000000 r9:00000000 r8:00000000 r7:00000000 r6:00000000 r5:8014497c
[ 1854.517952] r4:b5e8a840
[ 1854.517964] ---[ end trace f10274105b48c145 ]---
[ 1854.561892] Button released, resuming LED sequence
[ 2147.968287] LED driver removed
[ 2159.465122] LED driver initialized
[ 2206.625654] LED device opened
[ 2206.625730] LED sequence started
[ 2206.625747] LED device closed
[ 2216.088723] LED device opened
[ 2216.088776] LED sequence stopped
[ 2216.088796] LED device closed
[ 2323.028224] LED driver removed
[ 2884.090466] LED driver initialized
[ 2909.177097] LED device opened
[ 2909.177112] LED sequence started
[ 2958.498310] LED device opened
[ 2958.498335] LED sequence started
[ 3000.185789] Button pressed, turning off LEDs
[ 3000.206347] Button released, resuming LED sequence
[ 3000.218213] -----[ cut here ]-----
[ 3000.218239] WARNING: CPU: 3 PID: 3763 at drivers/gpio/gpiolib.c:128 gpio_to_desc+0xb4/0xe4
[ 3000.218247] invalid GPIO 2131222548
```



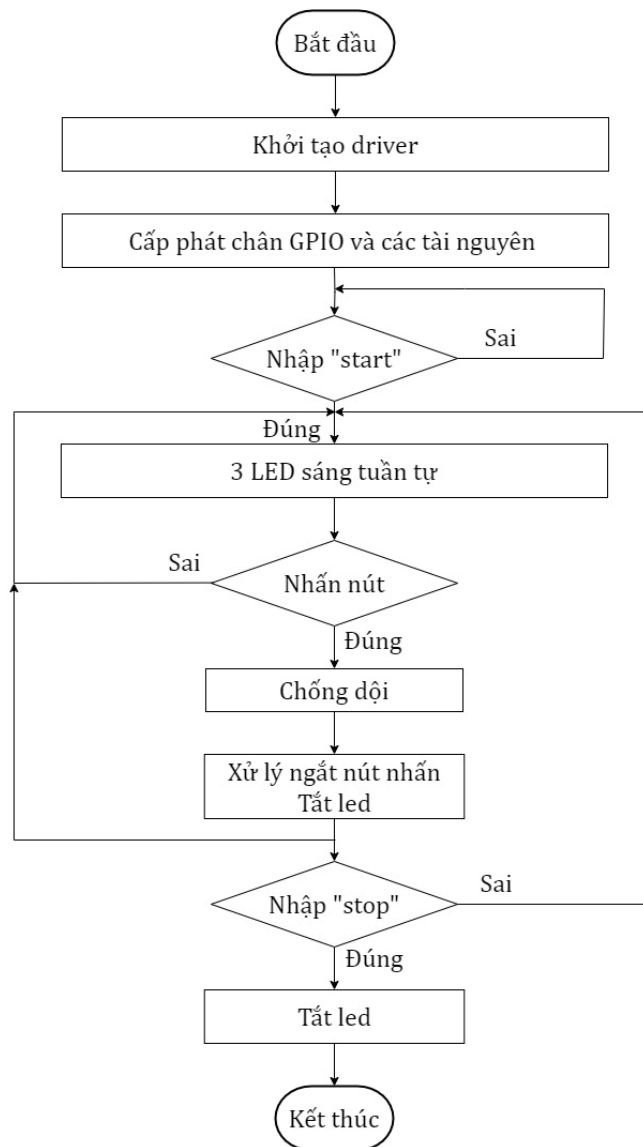
HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

e. Lưu đồ giải thuật:





f. Code và giải thích code

```
1 #include <linux/module.h>
2 #include <linux/kernel.h>
3 #include <linux/fs.h>
4 #include <linux/gpio.h>
5 #include <linux/uaccess.h>
6 #include <linux/delay.h>
7 #include <linux/cdev.h>
8 #include <linux/device.h>
9 #include <linux/interrupt.h>
10 #include <linux/kthread.h>
11 #include <linux/sched.h>
12 #include <linux/jiffies.h>
13
14 #define DEVICE_NAME "led_seq" //Name module
15 #define LED_COUNT 3
16 #define BUTTON_GPIO 4 // Set the GPIO pin for the button
17
18 // Set the GPIO pin for 3 LEDs
19 static int led_pins[LED_COUNT] = {17, 27, 22};
20
21 static struct class *led_class;
22 static struct cdev led_cdev;
23 static dev_t dev_num;
24 static int led_seq_running = 0; //Sequential lighting state variable
25 static int button_irq;
26 static unsigned long last_jiffies; // Variable debounceTime
27 static int button_state = 1; // Previous state of the button (1 = released, 0 = pressed)
28     int i;
29 #define DEBOUNCE_DELAY_MS 20 // Debounce delay in milliseconds
```

- Khai báo thư viện, tên module, chân kết nối với nút nhấn và LED và các biến



```
31 static int led_open(struct inode *inode, struct file *file) {  
32     printk(KERN_INFO "LED device opened\n");  
33     return 0;  
34 }  
35  
36 static int led_release(struct inode *inode, struct file *file) {  
37     printk(KERN_INFO "LED device closed\n");  
38     return 0;  
39 }
```

- Khai báo hàm và in ra thông báo khi module được mở và đóng



```
37 static ssize_t led_write(struct file *file, const char __user *buf, size_t len, loff_t *off) {
38     char command[16];
39
40     if (len > sizeof(command) - 1)
41         return -EINVAL;
42
43     if (copy_from_user(command, buf, len))
44         return -EFAULT;
45
46     command[len] = '\0';
47
48     // Bắt đầu hoặc dừng chuỗi LED
49     if (strncmp(command, "start", 5) == 0) {
50         led_seq_running = 1;
51         printk(KERN_INFO "LED sequence started\n");
52     } else if (strncmp(command, "stop", 4) == 0) {
53         led_seq_running = 0;
54         printk(KERN_INFO "LED sequence stopped\n");
55     } else {
56         printk(KERN_WARNING "Invalid command\n");
57         return -EINVAL;
58     }
59
60     return len;
61 }
```

- Hàm `led_write` xử lý yêu cầu điều khiển trạng thái chuỗi LED của người dùng, bằng cách gửi chuỗi lệnh "start" hoặc "stop" đến user space.
- Biến `led_seq_running` ghi lại trạng thái tắt hoặc bật của chuỗi LED.



```
67 static struct file_operations fops = {
68     .owner = THIS_MODULE,
69     .open = led_open,
70     .release = led_release,
71     .write = led_write,
72 };
73
74 // Thread function to handle LED sequence
75 static int led_seq_thread(void *data) {
76     while (!kthread_should_stop()) {
77         if (led_seq_running) {
78             for (i = 0; i < LED_COUNT; i++) {
79                 gpio_set_value(led_pins[i], 1); // Turn on the LED
80                 msleep(500); // Delay 500ms
81                 gpio_set_value(led_pins[i], 0); // Turn off the LED
82             }
83         } else {
84             msleep(100); // Wait for a moment when not running sequentially
85         }
86     }
87
88     return 0;
89 }
```

- Khai báo cấu trúc file_operations.
- Hàm led_seq_thread này điều khiển tuần tự các LED khi biến led_seq_running = 1.



```
91 static struct task_struct *led_task;
92
93 // ISR (Interrupt Service Routine) for the button
94 static irqreturn_t button_irq_handler(int irq, void *dev_id) {
95     unsigned long current_jiffies = jiffies;
96     if (time_before(current_jiffies, last_jiffies + msecs_to_jiffies(DEBOUNCE_DELAY_MS))) {
97         return IRQ_HANDLED;
98     }
99     last_jiffies = current_jiffies;
100
101     // Check the current state of the button
102     if (gpio_get_value(BUTTON_GPIO) == 0) { // Button pressed
103         if (button_state == 1) { // Previously released, now pressed down
104             printk(KERN_INFO "Button pressed, turning off LEDs\n");
105
106             //Turn off all LEDs
107             for (i = 0; i < LED_COUNT; i++) {
108                 gpio_set_value(led_pins[i], 0);
109             }
110
111             led_seq_running = 0; // Stop the LED sequence
112             button_state = 0;    // Update the button state
113         }
114     } else if (gpio_get_value(BUTTON_GPIO) == 1) { // The button is released
115         if (button_state == 0) { // Previously pressed, now released
116             printk(KERN_INFO "Button released, resuming LED sequence\n");
117             led_seq_running = 1; // Continue the LED sequence
118             button_state = 1;    // Update the button state
119         }
120     }
121
122     return IRQ_HANDLED;
123 }
```

- Hàm xử lý ngắt cho phép bật hoặc tắt chuỗi LED khi nút nhấn được nhấn hoặc nhả.



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

```
125 static int __init led_init(void) {
126     int ret;
127
128     // Allocate the device number
129     ret = alloc_chrdev_region(&dev_num, 0, 1, DEVICE_NAME);
130     if (ret < 0) {
131         printk(KERN_ALERT "Failed to allocate device number\n");
132         return ret;
133     }
134
135     //Create a device class
136     led_class = class_create(THIS_MODULE, DEVICE_NAME);
137     if (IS_ERR(led_class)) {
138         unregister_chrdev_region(dev_num, 1);
139         return PTR_ERR(led_class);
140     }
141
142     // Initialize cdev structure and add to system
143     cdev_init(&led_cdev, &fops);
144     led_cdev.owner = THIS_MODULE;
145     ret = cdev_add(&led_cdev, dev_num, 1);
146     if (ret < 0) {
147         class_destroy(led_class);
148         unregister_chrdev_region(dev_num, 1);
149         return ret;
150     }
151
152     // Create device file
153     device_create(led_class, NULL, dev_num, NULL, DEVICE_NAME);
```




```
155 // Request GPIO and configure output
156 for (i = 0; i < LED_COUNT; i++) {
157     if (gpio_request_one(led_pins[i], GPIOF_OUT_INIT_LOW, NULL)) {
158         printk(KERN_ALERT "Failed to request GPIO %d\n", led_pins[i]);
159         return -1;
160     }
161 }
162
163 // Request GPIO for the button and configure input
164 if (gpio_request_one(BUTTON_GPIO, GPIOF_IN, "button_gpio")) {
165     printk(KERN_ALERT "Failed to request GPIO %d\n", BUTTON_GPIO);
166     return -1;
167 }
168
169 // Request an IRQ for the button
170 button_irq = gpio_to_irq(BUTTON_GPIO);
171 if (button_irq < 0) {
172     printk(KERN_ALERT "Failed to get IRQ number for GPIO %d\n", BUTTON_GPIO);
173     return button_irq;
174 }
175
176 ret = request_irq(button_irq, button_irq_handler, IRQF_TRIGGER_RISING | IRQF_TRIGGER_FALLING, "button_irq", NULL);
177 if (ret) {
178     printk(KERN_ALERT "Failed to request IRQ\n");
179     return ret;
180 }
181
182 // Start LED sequence thread
183 led_task = kthread_run(led_seq_thread, NULL, "led_seq_thread");
184
185 printk(KERN_INFO "LED driver initialized\n");
186 return 0;
187 }
```



```
189 static void __exit led_exit(void) {
190
191     // Stop LED sequence thread
192     if (led_task) {
193         kthread_stop(led_task);
194     }
195
196     // Free the IRQ for the button
197     free_irq(button_irq, NULL);
198
199     // Free the GPIO for the button
200     gpio_free(BUTTON_GPIO);
201
202     // Free GPIO pins for LEDs
203     for (i = 0; i < LED_COUNT; i++) {
204         gpio_set_value(led_pins[i], 0);
205         gpio_free(led_pins[i]);
206     }
207
208     // Destroy device file and class
209     device_destroy(led_class, dev_num);
210     class_destroy(led_class);
211     cdev_del(&led_cdev);
212     unregister_chrdev_region(dev_num, 1);
213
214     printk(KERN_INFO "LED driver removed\n");
215 }
216
217 module_init(led_init);
218 module_exit(led_exit);
```

- Hàm led_exit để giải phóng chân GPIO, IRQ, hủy device file, class, dừng kernel thread trước khi module bị gỡ.



TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMC University of Technology and Education

CẢM ƠN THẦY VÀ CÁC BẠN ĐÃ LẮNG NGHE

