微算機系統實習

第 13 組專案報告

LAB 05

SPI 控制實習

組別： 13

班級、姓名與學號：

醫工三 B812110004 葉芸茜

醫工三 B812110011 湯青秀

日期：2024.04.22

一、實驗內容：

1. 學習如何應用麵包板結合感測器與 LED 進行資料擷取
2. 學習如何透過 GPIO 與 Python 讀取 TX2 上的光敏電阻數值
3. 透過 Python 控制 GPIO 上的光敏電阻和 2 個 LED

(1)項目一：讀取光敏電阻數值

在麵包板上安裝類比數位訊號轉換器及光敏電阻

撰寫 Python 讀取光敏電阻數值

在 Terminal 上顯示光敏電阻數值

(2)項目二：設定門檻值控制 LED 燈狀態

透過讀取光敏電阻數值設定門檻值

依照設定的門檻值控制 2 顆 LED 燈開關狀態

在 Terminal 上顯示 LED 燈狀態

(3)項目三：測試 CSI Camera

使用 TX2 上之 CSI 相機，並顯示畫面

擷取乙張影像附在報告中

4. 了解 TX2 physical pin 對應的 sysfs filename
5. 學習如何使用 TX2 上的 CSI 接口的相機

二、實驗過程及結果：

1. 實驗過程

(1) 撰寫程式碼

(a) 對應 TX2 上的腳位及初始設定

```python
import Jetson.GPIO as GPIO
import time

SPICLK = 11 #23
SPIMISO = 9 #21
SPIMOSI = 10 #19
SPICS = 8 #24
output_pin = 17 #11
output_pin2 = 27 #13

photo_ch = 0 #27
```

```python
def init():
    GPIO.setwarnings(False)

    GPIO.cleanup()
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(output_pin, GPIO.OUT)
    GPIO.setup(output_pin2, GPIO.OUT)
    GPIO.setup(SPIMOSI, GPIO.OUT)
    GPIO.setup(SPIMISO, GPIO.IN)
    GPIO.setup(SPICLK, GPIO.OUT)
    GPIO.setup(SPICS, GPIO.OUT)
```

(b) 取得光敏電阻的數值

```python
def readadc(adcnum, clockpin, mosipin, misopin, cspin, output1, output2):
    if((adcnum > 7) or (adcnum < 0)):
        return -1
    GPIO.output(cspin, True)

    GPIO.output(clockpin, False)
    GPIO.output(cspin, False)

    commandout = adcnum
    commandout |= 0x18
    commandout <<= 3
    for i in range(5):
        if (commandout & 0x80):
            GPIO.output(mosipin, True)
        else:
            GPIO.output(mosipin, False)
        commandout <<= 1
        GPIO.output(clockpin, True)
        GPIO.output(clockpin, False)

    adcout = 0
    for i in range(12):
        GPIO.output(clockpin, True)
        GPIO.output(clockpin, False)
        adcout <<= 1
        if(GPIO.input(misopin)):
            adcout |= 0x1

    GPIO.output(cspin, True)

    adcout >>= 1
    print(adcout)
```

(c) 設計光敏數值閥值以控制 LED 狀態

下表為我們希望可以達到的效果

	LED1	LED2
亮	暗	暗
普通	亮	暗
暗	亮	亮

經過觀察發現，當光敏電阻處在越亮的環境中，讀數就會越小，反之則會變大。而在正常光線下，數值大約為 300 左右。因此我們將範圍切割為>450(暗)，450-150(正常)，以及 150 以下(亮)。

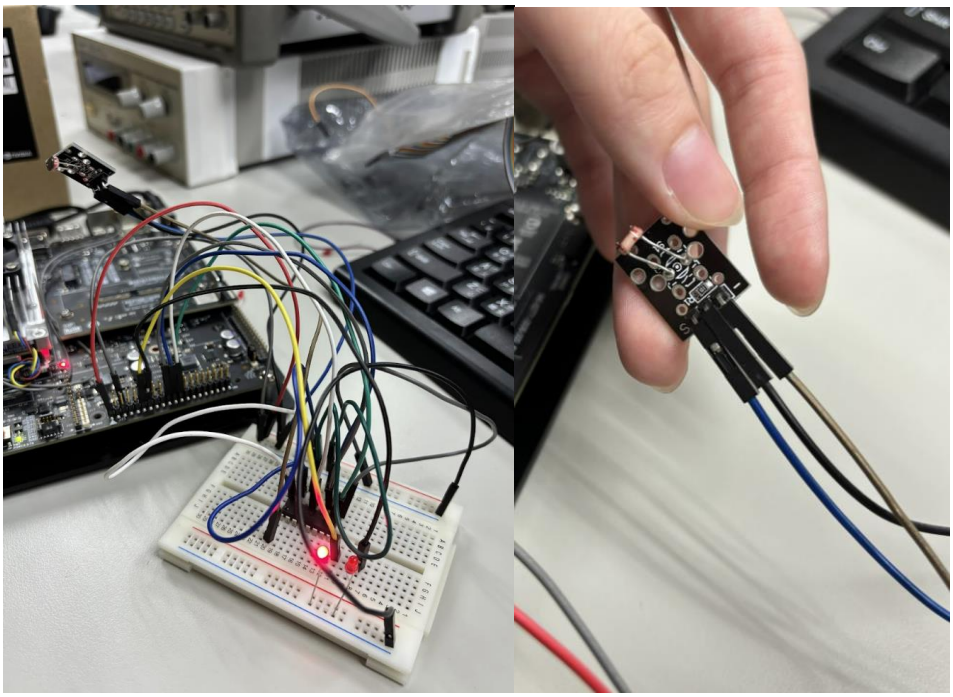
```python
if(adcout > 450):
        #led2 bright
        GPIO.output(output2, GPIO.HIGH)
        #led1 bright
        GPIO.output(output1, GPIO.HIGH)
        print("LED1 on")
        print("LED2 on")
    elif(adcout > 150):
        #led1 bright
        GPIO.output(output1, GPIO.HIGH)
        #led2 dark
        GPIO.output(output2, GPIO.LOW)
        print("LED1 on")
        print("LED2 off")
    else:
        #led1 dark
        GPIO.output(output1, GPIO.LOW)
        #led2 dark
        GPIO.output(output2, GPIO.LOW)
        print("LED1 off")
        print("LED2 off")

    return adcout
```

（d）初始化後持續呼叫

```python
def main():
    init()
    while True:
        adc_value = readadc(photo_ch, SPICLK, SPIMOSI, SPIMISO, SPICS, output_pin, output_pin2)
        print(adc_value)
        #LED_control(adc_value, output_pin, output_pin2)
        time.sleep(1)
```

(2) 運用麵包版連接各組件



(3) 輸入指令開啟 CSI Camera

```
nvidia@nvidia-desktop:~/lab05$ gst-launch-1.0 nvarguscamerasrc ! nvvidconv ! xvimagesink
nvbuf_utils: Could not get EGL display connection
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
GST_ARGUS: Creating output stream
CONSUMER: Waiting until producer is connected...
GST_ARGUS: Available Sensor modes :
GST_ARGUS: 2592 x 1944 FR = 29.999999 fps Duration = 33333334 ; Analog Gain range min 1.000000, max 16.000000; Exposure Range min 34000, max 550385000;

GST_ARGUS: 2592 x 1458 FR = 29.999999 fps Duration = 33333334 ; Analog Gain range min 1.000000, max 16.000000; Exposure Range min 34000, max 550385000;

GST_ARGUS: 1280 x 720 FR = 120.000005 fps Duration = 8333333 ; Analog Gain range min 1.000000, max 16.000000; Exposure Range min 22000, max 358733000;

GST_ARGUS: Running with following settings:
   Camera index = 0
   Camera mode  = 1
   Output Stream W = 2592 H = 1458
   seconds to Run    = 0
   Frame Rate = 29.999999
GST_ARGUS: Setup Complete, Starting captures for 0 seconds
GST_ARGUS: Starting repeat capture requests.
CONSUMER: Producer has connected; continuing.
packet_write_wait: Connection to 192.168.55.1 port 22: Broken pipe
```

2. 預期實驗結果

(1) 項目一：執行 python 檔後，光敏電阻接收到的類比訊號會經由 ADC 轉換成數位光敏電阻數值，並顯示在終端上

(2) 項目二：根據項目一得到的數值來設定門檻值以控制 LED 燈狀態。通過給予感測器明暗狀態(設定的門檻值不同)來控制 2 顆 LED 燈開關狀態(會顯示 LED 狀態在終端)

(3) 項目三：在終端輸入指令，測試 TX2 上的 CSI Camera 是否能正常使用，若成功相機將開啟並於螢幕顯示畫面

3. 實際上的結果

(1) LED 控制

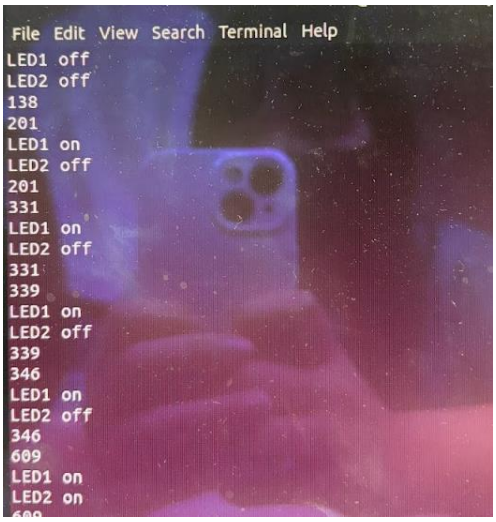
全景：

https://drive.google.com/file/d/1BvFquShfJCu1sXSILxTG_8VGs1M4PXfT/view?usp=sharing

LED 特寫：

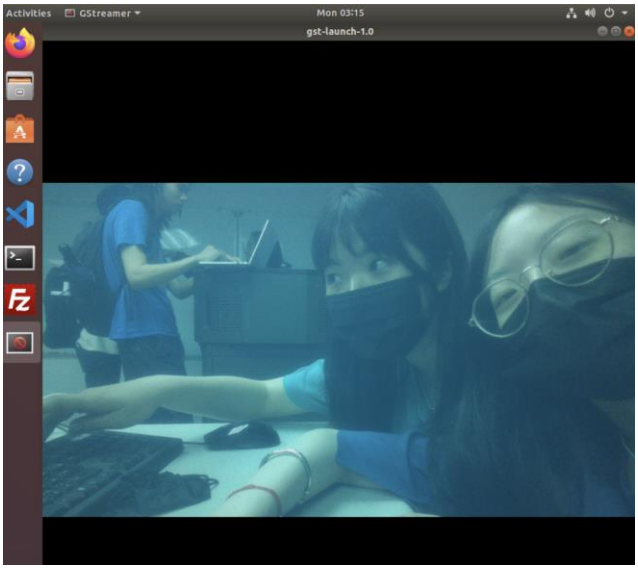
https://drive.google.com/file/d/1a-UgO_x0R7-f6h2bUf2A9hMF1HDuPVf-/view?usp=sharing

測試照片：



(2) TX2 相機

https://drive.google.com/file/d/1vrNnDOHP4WEQCAdqGUzFTqoO_URbdPWB/view?usp=sharing



4. 遇到的問題&問題怎麼解決

(1) 一開始我們試了幾個腳位去連接到 LED，卻多次被拒絕操作，可能是因為對應腳位無法作為輸出嗎? 所以我們就看著對應表挑選其他幾個可能可用腳位做測試，並順利找到可用腳位

(2) 切換編碼模式(原版程式碼中使用兩種編碼的腳位：BCM 和 Board)的時候，output 腳位設置被一起 cleanup 了導致執行出問題。後來我們改使用統一一種模式(BCM)就解決問題了

三、程式碼

gpio. py

```
import Jetson.GPIO as GPIO
import time

SPICLK = 11 #23
SPIMISO = 9 #21
SPIMOSI = 10 #19
SPICS = 8 #24
output_pin = 17 #11
output_pin2 = 27 #13

photo_ch = 0 #27

def init():
    GPIO.setwarnings(False)

    GPIO.cleanup()
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(output_pin, GPIO.OUT)
    GPIO.setup(output_pin2, GPIO.OUT)
    GPIO.setup(SPIMOSI, GPIO.OUT)
    GPIO.setup(SPIMISO, GPIO.IN)
    GPIO.setup(SPICLK, GPIO.OUT)
    GPIO.setup(SPICS, GPIO.OUT)

def readadc(adcnum, clockpin, mosipin, misopin, cspin, output1,
output2):
    if((adcnum > 7) or (adcnum < 0)):
        return -1
    GPIO.output(cspin, True)

    GPIO.output(clockpin, False)
    GPIO.output(cspin, False)

    commandout = adcnum
    commandout |= 0x18
    commandout <<= 3
    for i in range(5):
        if (commandout & 0x80):
            GPIO.output(mosipin, True)
        else:
            GPIO.output(mosipin, False)
        commandout <<= 1
        GPIO.output(clockpin, True)
        GPIO.output(clockpin, False)

    adcout = 0
```

```
    for i in range(12):
        GPIO.output(clockpin, True)
        GPIO.output(clockpin, False)
        adcout <<= 1
        if(GPIO.input(misopin)):
            adcout |= 0x1

    GPIO.output(cspin, True)

    adcout >>= 1
    print(adcout)

    if(adcout > 450):
        #led2 bright
        GPIO.output(output2, GPIO.HIGH)
        #led1 bright
        GPIO.output(output1, GPIO.HIGH)
        print("LED1 on")
        print("LED2 on")
    elif(adcout > 150):
        #led1 bright
        GPIO.output(output1, GPIO.HIGH)
        #led2 dark
        GPIO.output(output2, GPIO.LOW)
        print("LED1 on")
        print("LED2 off")
    else:
        #led1 dark
        GPIO.output(output1, GPIO.LOW)
        #led2 dark
        GPIO.output(output2, GPIO.LOW)
        print("LED1 off")
        print("LED2 off")

    return adcout


def main():
    init()
    while True:
        adc_value = readadc(photo_ch, SPICLK, SPIMOSI, SPIMISO,
SPICS, output_pin, output_pin2)
        print(adc_value)
        #LED_control(adc_value, output_pin, output_pin2)
        time.sleep(1)

if __name__ == '__main__':
    main()
```

四、本次實驗過程說明與解決方法:

1. 實驗過程

   撰寫程式碼 → 連接麵包板、感測器、IC、LED → 執行 python 檔案 → 終端顯示光敏電阻數值 → 程式碼中設定閥值 → 執行 python 檔案 → 終端顯示 LED 實時狀態 → 測試 CSI Camera 是否開啟

2. 解決方法

   這次問題主要卡在編碼及腳位，因為編碼切換的問題導致 LED 無法控制明暗，因此我們統一了編碼模式來解決問題。

五、分工:

學號、組員	貢獻比例	工作內容
B812110004 葉芸茜	50%	文書處理、實驗設計與實作、程式規劃、測試與除錯
B812110011 湯青秀	50%	文書處理、實驗設計與實作、程式規劃、測試與除錯