

通訊網路實驗 Lab1 Report

110511254 徐煜絨

一、實驗目的（第二部分實驗說明會有更詳細介紹）

Q1

用 LED 的亮暗長度產生 SOS 的摩斯密碼。

Q2

使用 DHT11 量測溫度和濕度，達一定溫度時要讓 LED 燈亮。

Q3

利用 HC-SR04 測量距離並製作出距離警示燈。

Q4

整合前面的實驗內容，結合 DHT11、HC-SR04、LED，做出能更精確量測距離的距離警示燈。

二、實驗過程（Code+說明）

Q1

已知 S 和 O 的密碼為下：

S ... O ---

Dot 代表亮 0.1 秒，Dash 則是 0.3 秒，另外同字母的任兩個 dot 或 dash 間要暗 0.1 秒，字母間 0.3 秒，單字間 0.7 秒，這些設定都寫在程式碼中。另外在硬體設備上，將樹梅派的輸出接腳連上 LED 正極，並由 GPIO 控制訊號為 HIGH 或 LOW，而 LED 負極和 GND 間放上電阻以免燒毀，這些控制以 GPIO.HIGH 或 GPIO.LOW 的形式出現在程式碼中。

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time
# GPIO.BOARD->pin number
# GPIO.BCM->GPIO number
GPIO.setmode(GPIO.BOARD)
# 第 12 pin 腳負責控制 LED
LED_PIN = 12
GPIO.setup(LED_PIN, GPIO.OUT)

while True: # 反覆執行
```

```
# 反覆亮 SOS 的摩斯密碼
print("Letter S")
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.1)

# letter 間需要空三個單位時間
print("SPACE")
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.3)

print("Letter O")
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.3)
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.3)
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.3)

print("SPACE")
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.3)

print("Letter S")
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.1)
```

```

GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.1)
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(0.1)

# word 間需要空七個單位時間
print("LED is off")
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(0.7)

GPIO.cleanup() # 清除 GPIO 設定

```

Q2

在實驗中用到溫溼度感測器 DHT11，將其 data 腳接到 GPIO 4 (Pin7)，就可以用程式碼讓他運作並接收資料，這次實驗需要讓使用者自行決定 threshold 溫度，所以程式碼中用了 threshold=input()，當測出的溫度超過輸入的溫度，LED 就會亮。要注意的是執行時要輸入 \$ sudo ./AdafruitDHT.py 11 4 表示使用 DHT11，data 來自 GPIO 4。

```

#!/usr/bin/python
# 設定函式庫以及相關腳位
import sys
import Adafruit_DHT
import time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)
LED_PIN = 12
GPIO.setup(LED_PIN, GPIO.OUT)

# 設定 DHT 相關參數
sensor_args = { '11': Adafruit_DHT.DHT11,
                 '22': Adafruit_DHT.DHT22,
                 '2302': Adafruit_DHT.AM2302 }
if len(sys.argv) == 3 and sys.argv[1] in sensor_args:
    sensor = sensor_args[sys.argv[1]]
    pin = sys.argv[2]
else:

```

```

print('Usage: sudo ./Adafruit_DHT.py [11|22|2302] <GPIO pin number>')
print('Example: sudo ./Adafruit_DHT.py 2302 4 - Read from an AM2302 connect$
sys.exit(1)

threshold = input("temp")

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

# temperature = temperature * 9/5.0 + 32
if humidity is not None and temperature is not None:
    print('Temp={0:0.1f}* Humidity={1:0.1f}%'.format(temperature, humidity))
else:
    print('Failed to get reading. Try again!')
    sys.exit(1)
while(True):
    # 讀取溫溼度
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
    # 設定 output format
    print('Temp={0:0.1f}* Humidity={1:0.1f}%'.format(temperature, humidity))
    if(temperature>threshold): # 大於一定溫度就亮
        GPIO.output(LED_PIN, GPIO.HIGH)
    else:
        GPIO.output(LED_PIN, GPIO.LOW)
    time.sleep(1)
GPIO.cleanup() # 清除 GPIO 設定

```

Q3

這個實驗用樹梅派控制 HC-SR04 的 Trigger 發射訊號，訊號打到物體後會反彈並由 Echo 接收，可以利用發射和接收的時間差(t)，加上預設的聲速 343，計算出 HC-SR04 和物體的距離(d)。

```

# 引進函式庫、設定腳位、設定變數
import RPi.GPIO as GPIO
import time
# 忽略警示訊息
GPIO.setwarnings(False)
v=343
TRIG = 16
E = 18

```

```

LED = 12

# print '1'
GPIO.setmode(GPIO.BOARD)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(E, GPIO.IN)
GPIO.output(TRIG, GPIO.LOW)
GPIO.setup(LED, GPIO.OUT)

def measure():
    # 發送一個瞬間訊號
    GPIO.output(TRIG, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(TRIG, GPIO.LOW)
    # 計時
    pulse_start = 0
    pulse_end = 0
    while GPIO.input(E) == GPIO.LOW:
        pulse_start = time.time()
    while GPIO.input(E) == GPIO.HIGH:
        pulse_end = time.time()

    # 距離為(來回總耗時*音速)/2
    t = pulse_end - pulse_start
    d = t * v
    d = d / 2
    return (d*100)

while (1):
    print measure()
    dis=measure()
    # 距離介在 10~20: 閃爍
    if dis > 10 and dis < 20:
        GPIO.output(LED, GPIO.HIGH)
        time.sleep(0.2)
        GPIO.output(LED, GPIO.LOW)
        time.sleep(0.2)
        GPIO.output(LED, GPIO.HIGH)
        time.sleep(0.2)

```

```

# 距離<10: 恆亮
elif dis < 10:
    GPIO.output(LED, GPIO.HIGH)
# 距離>20: 不亮
else:
    GPIO.output(LED, GPIO.LOW)
    time.sleep(1)
GPIO.cleanup()

```

Q4

最後一個實驗要同時使用 DHT11 和 HC-SR04，這次測距結果會較符合實際情況。假設發射和接收的時間差是 t ，DHT11 量出的當下溫度是 c ，那就會讓聲速 $v = 331 + 0.6c$ 。接著就可以以 $d = v * (\frac{t}{2})$ 計算出 HC-SR04 和物體的距離(d)。

```

# 引入模組、設定腳位和 GPIO
import RPi.GPIO as GPIO
import time
import Adafruit_DHT

GPIO.setwarnings(False)

TRIG = 16
E = 18
LED = 12

# print '1'
GPIO.setmode(GPIO.BOARD)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(E, GPIO.IN)
GPIO.output(TRIG, GPIO.LOW)
GPIO.setup(LED, GPIO.OUT)

# 設定 DHT11
sensor_args = { '11': Adafruit_DHT.DHT11,
                 '22': Adafruit_DHT.DHT22,
                 '2302': Adafruit_DHT.AM2302 }
sensor = sensor_args['11']

```

```

# GPIO 第四腳位
dht_pin = 4

def measure(v):
    GPIO.output(TRIG, 1)
    time.sleep(0.00001)
    GPIO.output(TRIG, 0)
    pulse_start = 0
    pulse_end = 0
    while GPIO.input(E) == 0:
        pulse_start = time.time()
    while GPIO.input(E) == 1:
        pulse_end = time.time()

    t = pulse_end - pulse_start
    d = t * v
    d = d / 2
    return d * 100

while 1:
    H, T = Adafruit_DHT.read_retry(sensor, dht_pin)
    # 音速公式
    v = 331 + 0.6 * T
    # 印出溫度和速度
    print ("=====")
    print ("Temp:" + str(T) + "*C")
    print ("V=331+0.6*" + str(T))
    print ("="+str(v))
    # 計算距離並 print
    d = measure(v)
    print ("Distance : " + str(d))
    # 判斷距離並決定 LED 燈行為
    if d > 10 and d < 20:
        GPIO.output(LED, 1)
        time.sleep(0.2)
        GPIO.output(LED, 0)
        time.sleep(0.2)
        GPIO.output(LED, 1)
        time.sleep(0.2)

```

```

GPIO.output(LED, 0)
time.sleep(0.2)
GPIO.output(LED, 1)
time.sleep(0.2)
GPIO.output(LED, 0)
elif d < 10:
    GPIO.output(LED, 1)
else:
    GPIO.output(LED, 0)
time.sleep(0.2)
GPIO.cleanup()

```

三、問題及解法

在做第三題量距離的時候，一直測出和實際情況相差許多的結果，在換過 HC-SR04 也確認接線後也找不出結果，但就在準備離開時突然被對面同學提醒是否有接錯 3.3V 和 5V，於是把 HC-SR04 的 VCC 改接到樹梅派的 5V 接腳，最後順利解決問題。



四、心得

原本想說脫離大二電子實驗就不會再碰到麵包板、HC-SR04、DHT11、杜邦線這些東西，沒想到大三上的選修又和他們見面，心裡免不了擔心和害怕。但這次實驗課比我想像得還要順利不少，照著講義下載、燒資料到樹梅派的過程也讓我開始期待。在電子實驗中是用 Arduino Uno 控制 HC-SR04、DHT11、LED 等元件的行為，不過這次用到樹梅派這台小電腦控制，感覺自己會在這堂課中學習並成長。

Reference:

```

# Copyright (c) 2014 Adafruit Industries
# Author: Tony DiCola

# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:

```



```
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
```

```
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
```