

# 嵌入式系統實驗 Final project

題目:智慧額溫槍量測

成員:彭成量 黃丞右 樂道承

Github URL:<https://github.com/peng0811/embeddedsystem/tree/main/project>

## 一、動機:

日前傳染病的盛行，人們想起了被病毒統治的那段日子，在這人心惶惶的日子裡，我們都需要小心翼翼的過活，降低人與人的連結的機會減少傳染途徑以及進入室內前必須先檢測自己的身體狀況不要自己中獎了還要害別人也跟上，而最好的檢測方式就是量測身體溫度去檢查是否發燒，為此我們團隊萌生了要擔任起防疫小尖兵的正義感，想要利用這學期所學所用的技術與知識去製作出量測額溫的一個嵌入式系統，為世界盡一份心力。

## 二、功能介紹:

### 1. Python 使用:

#### a. BLE 接受溫度數據

```
print(word)
print ("Connecting...")
dev = Peripheral(word, 'random')
print ("Services...")
for svc in dev.services:
    print (str(svc))
k=1
try:
    testService = dev.getServiceByUUID(UUID(0xffff0))
    for ch in testService.getCharacteristics():
        print (str(ch))
        ch = dev.getCharacteristics(uuid=UUID(0xffff1))[0]
    def data_rec():
        global num
        global ID
        if (ch.supportsRead()):
            c = ch.read()
            num =(int(c[0])+256)/10
            num=num*10
            time.sleep(0.5)
```

此段為接收溫度數據，因為藍芽傳輸無法傳送浮點數，故我們將數據乘 10 取小數點第一位傳過來，再透過 python 把數值除以 10 就得到我們要的數據。

b. Websocket 與網頁連接互傳數據

```
async def web_socket(websocket,path):  
    global num  
    global data  
    global times  
    global idx  
    times = 0  
    while(run):  
        s = await websocket.recv()  
        print(s)  
        if s!=1:  
            sheet.cell(row=idx,column=1,value=s)  
            wb.save('data3.xlsx')  
            while 1:  
                data_rec()  
                if num >=350:  
                    times += 1  
                print(times)  
                data= "{\\\"val\\\":%d , \\\"time\\\":%d }" % (num,times)  
                time.sleep(1)  
                await websocket.send(data)  
                if num >=350:  
                    sheet.cell(row=idx,column=2,value=num/10)  
                    wb.save('data3.xlsx')  
                    idx=idx+1  
                    break
```

此處與網頁以 websocket 連接，將收到的溫度數據丟給網頁，網頁才進行判斷是否太遠、發燒或體溫正常。網頁也會回傳量測人的資訊回來，然後再把資訊跟溫度結合儲存在 excel 給資料庫使用。

c. 資料庫存放數據

```
def xlsx_to_csv_pd():
    data_xls = pd.read_excel('data3.xlsx', index_col=0)
    data_xls.to_csv('data3.csv', encoding='utf-8')
if __name__ == '__main__':
    xlsx_to_csv_pd()

def os_any_dir_search(file):
    u=[]
    for p,n,f in os.walk(os.getcwd()):

        for a in f:
            a = str(a)
            if a.endswith(file):
                t=pd.read_csv(p+'/'+file)
                u.append(p+'/'+a)
    return t,u

data = os_any_dir_search('data3.csv')[0]
#fake_users=fake_users.iloc[:,1:]
#print(fake_users.head())

data.to_csv('data3_fixed.csv')
data = data.to_dict('records')
#print(fake_users_dict[:5])

client = pymongo.MongoClient('mongodb://localhost:27017/')
db = client["measurement"]
user_info_table= db["temp_taipei"]
user_info_table.insert_many(data)
```

將量測完的溫度數據與對應的個人資訊儲存起來，可提供往後追蹤使用。

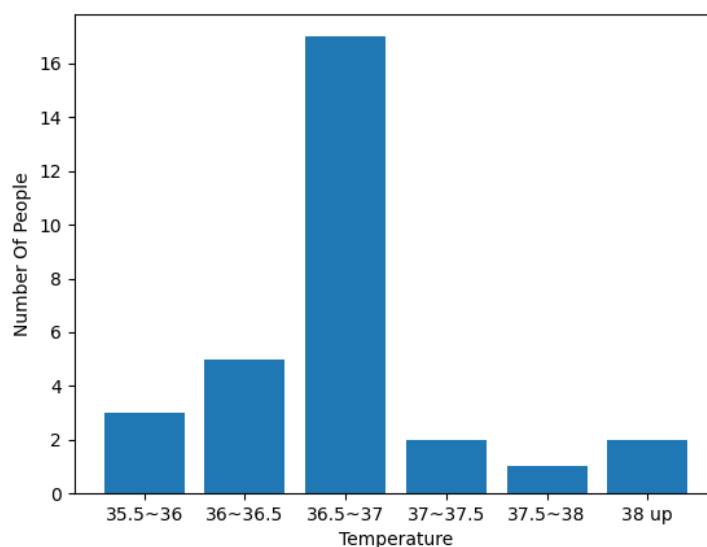
```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
df = pd.read_excel("C:\\Users\\User\\Desktop\\ES\\finalprojects//data.xlsx")
num1 = 0
num2 = 0
num3 = 0
num4 = 0
num5 = 0
num6 = 0

for x in df["Temperature"]:
    if x >= 35.5 and x < 36:
        num1 = num1 + 1
    elif x >= 36 and x < 36.5:
        num2 = num2 + 1
    elif x >= 36.5 and x < 37:
        num3 = num3 + 1
    elif x >= 37 and x < 37.5:
        num4 = num4 + 1
    elif x >= 37.5 and x < 38:
        num5 = num5 + 1
    else:
        num6 = num6 + 1

left = np.array(['35.5~36', '36~36.5', '36.5~37', '37~37.5', '37.5~38', '38 up'])
height = np.array([num1, num2, num3, num4, num5, num6])
plt.bar(left, height)
plt.xlabel("Temperature")
plt.ylabel("Number Of People")
plt.show()

```



而資料庫收集完的數據，透過前面 plot 程式繪出此長條圖，表現出每個人量測體溫的分布情形，可以預測出疫情傳染的趨勢。

## 2. STM32:

```
void start() {
    _ble.gap().setEventHandler(this);

    _ble.init(this, &BatteryDemo::on_init_complete);

    _event_queue.call_every(500, this, &BatteryDemo::blink);
    _event_queue.call_every(500, this, &BatteryDemo::button_pressed);
    _event_queue.dispatch_forever();
}
```

透過 event\_queue API 每 500 毫秒將量測溫度的排程放進 queue。

```
union dataspace{
    unsigned char q[4];
    float t;
}floatdata;

void updateButtonState(char newState) {
    floatdata.t = getTempC_Obj1(0x5A);
    floatdata.t = floatdata.t*10-256;
    char T = floatdata.t;
    ble.gattServer().write(buttonState.getValueHandle(), (uint8_t *)&T, sizeof(char));
}
```

因為傳送的值最高為 256 且不能傳送浮點數，故我們將溫度數據取小數點第一位去減掉 256 再傳送過去做處理。

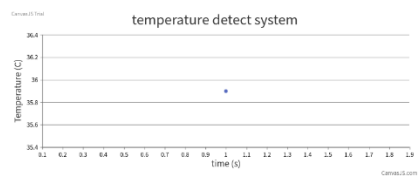
## 3. Html

一開始，網頁為下圖的樣子：

請先輸入你的ID或感應卡片

將量測者的資料輸入進去。

請先輸入你的ID或感應卡片



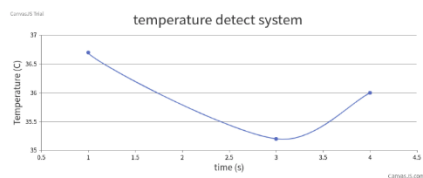
ID:R08522828

太遠了請靠近

□

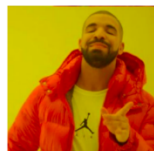
這邊接收 python 的溫度數據，並且做判斷處理，透過溫度的範圍可以判斷出量測距離是否太遠，太遠的話會提醒量測者，如上圖。

請先輸入你的ID或感應卡片



ID:R08522832

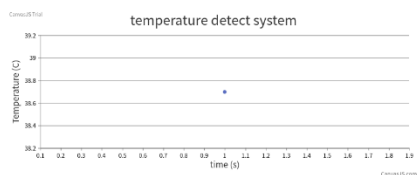
你的體溫為36度C



你體溫，給過

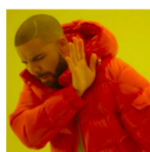
接著，如果量測成功網頁會判斷溫度範圍是否為發燒或正常，正常如下圖，發燒如下圖。

請先輸入你的ID或感應卡片



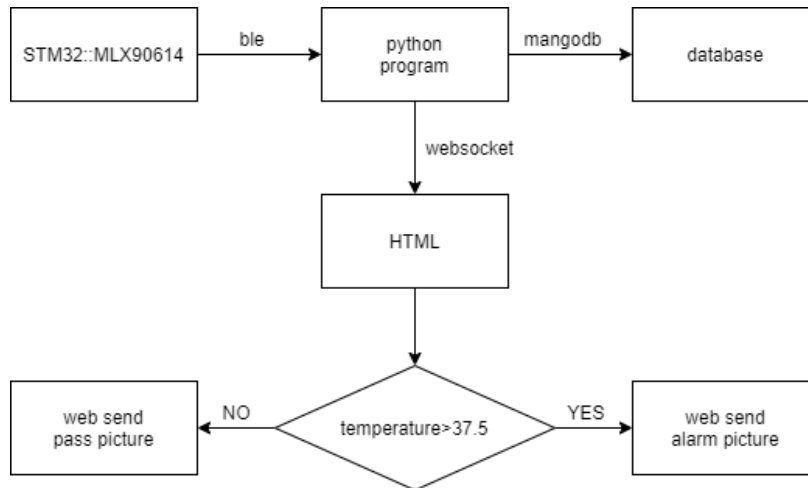
ID:R08543040

你的體溫為38.7度C



你體溫超過37度  
打妹嘢

### 三、流程圖



### 四、成果 demo

資料庫 demo:

<https://www.youtube.com/watch?v=WdTDAXwZ7KM>

藍芽傳輸&量測溫度 demo:

<https://www.youtube.com/watch?v=cSDLVBX6LFY>

<https://www.youtube.com/watch?v=ZkXifBKbqXs>

### 五、遇到的問題與解決方法

1. 無法傳送浮點數: 我們須透過 STM32 使用 I2C 與 MLX90614 紅外線溫度感測器量測溫度，並且透過藍芽傳輸將溫度數據傳送給電腦裡，但藍芽傳輸的資料無法傳送浮點數型態且值最高只能傳送 256，故我們嘗試了不少方法，像是利用 IEEE754，但也無法解決，但因為溫度數據小數點第一位已足夠且溫度範圍就是 35°C 到 40°C，故最後是把溫度數據乘以 10，並且扣除 256 再做傳送。

### 六、成員分工

彭成量: stm32 IoT 板功能實現(藍芽和 I2C)、系統整合

黃丞右: 建立資料庫、資料後處理

樂道承: 資料前處理、html 網頁撰寫

### 七、參考資料

MLX90614:

<https://os.mbed.com/users/ronaldhkleee/code/MLX90614/>

html 網頁:

<https://canvasjs.com/docs/charts/basics-of-creating-html5-chart/updating-chart-options/>

資料庫:

<https://www.youtube.com/watch?v=AAfFHk14xWw&t=309s>

<https://www.runoob.com/python3/python-mongodb.html>