

物聯網應用-以ESP32實作體溫通報器

組員:11010陳語汝 10405林杉美 10909許林瑄

◆ 內容介紹：

利用Python的開發環境，將藉由ESP32讀取到的電阻值，轉換成ADC值再以程式變換成實際溫度，並利用IFTTT的服務將溫度資訊即時透過LINE通知我們。

◆ 研究動機：

受到疫情的影響，再加上配合政府的政策，學校中午每天都會要求學生進行體溫量測，受到這個的啟發，我們便決定應用所學的ESP32來監測溫度的變化！

◆ 研究過程及成果：

主要使用材料：

ESP32控制板 + NTC熱敏電阻：連接並讀取、轉換ADC值

*小科普：ESP32控制板：可想像為一台小電腦

NTC熱敏電阻：藉由其「溫度越高，電阻值越低」的特性作為本測試的溫度計



Python Thonny開發環境：蒐集資料及利用訓練好的模型進行溫度預測。

Google Colab平台：建立神經網路利用蒐集到的資料去訓練模型。

IFTTT網路平台：IFTTT是IF This, Then That的縮寫，意思是如果A然後B，用以和LINE連接，傳訊息至我們的手機。

過程步驟：

1. 蒐集資料：測量與記錄ADC值及實際溫度

我們使用ESP32控制板與熱敏電阻的特性測量溫度，由此我們可以測量出類比資料，但是電腦是辨讀數位訊號的，所以用ADC將類比訊號轉換成數位訊號，再將結果儲存。

2. 輸入資料並存檔

接下來我們要設計一個程式讓使用者可以輸入溫度，然後顯示這個溫度的ADC值和溫度並存檔

3. 利用蒐集好的資料進行模型訓練：

為了要讓電腦可以將實際溫度對應到ADC值，我們要把蒐集到的資料進行訓練模型，如同函數公式一樣，使我們可以把ADC值帶入模型並轉換成實際溫度。

```
1. 1 from machine import Pin,ADC
2 import time
3
4 adc_pin = Pin(36) # 36是ESP32的VP腳位
5 adc = ADC(adc_pin) # 設定36為輸入腳位
6 adc.width(ADC.WIDTH_9BIT) # 設定分辨率位元數(解析度)
7 adc atten(ADC.ATTN_11DB) # 設定最大電壓
8
9 data=0 # 資料總和
10 ti=1 # 資料筆數
11 f=open('temperature.txt','w') # 開啟txt檔
12 print(adc.read()) # 先顯示一次，確認數值是否正常
13
14 while True:
15     print('第'+str(ti)+'筆') # 顯示紀錄筆數
16     tem=input("現在溫度:") # 輸入實際溫度
17     if(tem=='end'):
18         break
19     else:
20         for i in range(20): # 重複20次
21             thermal=adc.read() # ADC值
22             data=data+thermal # 加總至data
23             time.sleep(0.01)
24
25         data=int(data/20) # 取平均
26         print('熱敏電阻:',data)
27         print('') # 多空一行
28         f.write(str(data)+' '+tem+'\n') # data存到檔案中
29
30         data=0 # 總和歸0
31         ti+=1 # 次數加1
32 f.close()
```

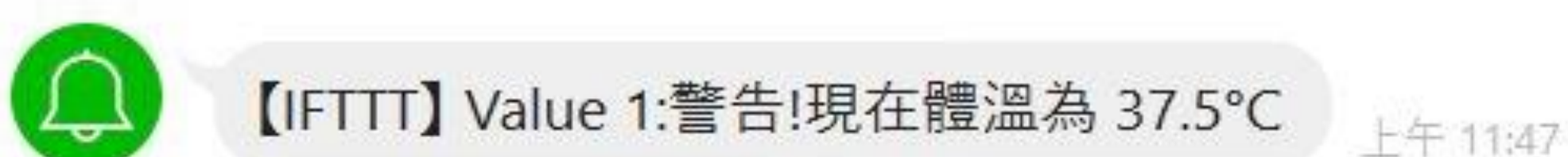
```
2. 1 from machine import Pin,ADC
2
3 f=open('test.txt','w')
4
5 adc_pin=Pin(36)
6 adc = ADC(adc_pin)
7 adc.width(ADC.WIDTH_9BIT)
8 adc.atten(ADC.ATTN_11DB)
9 tem=input("請輸入現在溫度:")
10
11 f.write(str(adc.read())+' '+tem)
12 f.close()
```

```
3. 8 #增加神經網路的參數與模型
9 mean = 170.98275862668965 #平均值
10 std = 90.31162360353873 #標準差
11
12 adc_pin = Pin(36)
13 adc = ADC(adc_pin)
14 adc.width(ADC.WIDTH_9BIT)
15 adc.atten(ADC.ATTN_11DB)
16
17 data=0
18
19 while True:
20
21     for i in range(20):
22         thermal=adc.read()
23         data=data+thermal
24         time.sleep(0.01)
25
26     data=data/20
27
28     print(int(data),end=' ') # 顯示ADC值,end=''代表不換行
29
30     data = np.array([int(data)]) # 將data轉換成array格式
31     data = data-mean # data減掉平均數
32     data = data/std # data除以標準差
33     tem = model.predict(data) # 得出預測值
34     tem = round(tem[0]*100,1) # 將預測值*100等於預測溫度
35     print(tem)
36
37     data=0
38     time.sleep(1) # 暫停1秒
```

4. 利用訓練的的模型預測，使模型訓練好後，只需將ADC值帶入便可以進行溫度的預測。

5. 使用IFTTT發送LINE通知

我們利用IFTTT提供的Webhook寫程式，程式內容是量測體溫，然後將程式當作A，而事件B設定成LINE，當觸發A時就會做B這個動作，所以當體溫超過37.5時便會傳送LINE訊息。



◆ 課程收穫及心得反思：在製作過程中，最困難的是寫程式，程式中只要有一個字母打錯整個指令就無法執行，當程式無法執行時，必須從頭檢查錯誤，考驗細心和耐心。透過這個專題，我們了解到傳統的事物也能有新的嘗試，現在我們可以使用硬體ESP32並結合程式設計，讓測量過後的溫度藉由程式驅動網路而通知我們。