

電壓偵測儀



卡米特

最後由 卡米特 於 2023年4月19日 編輯

貢獻者：



## 電壓偵測儀

tags: Alice 高醫

## 就讀動機

我就讀台南聖功女中，個性內斂沉穩，在冷靜的外表下藏著一顆熱情的心，縱使於高三繁忙的課業中，仍接下班長的職務，協助老師處理大型班務與安排班事宜；另一方面，在審慎自我督促維持校排1%的同時，也積極協助同學課業上的詢問。因受到同學認同，連任班長。此刻是高中生最忙的階段，除例行課業，大家為了繁星、個人申請做準備，自己依舊竭盡所能為班上服務，希望同學們都可專心準備分科及個人申請。我深信施比受更有福，在這個時刻接下重任讓我更清楚知道時間安排的重要性及輕重緩急的判斷力。

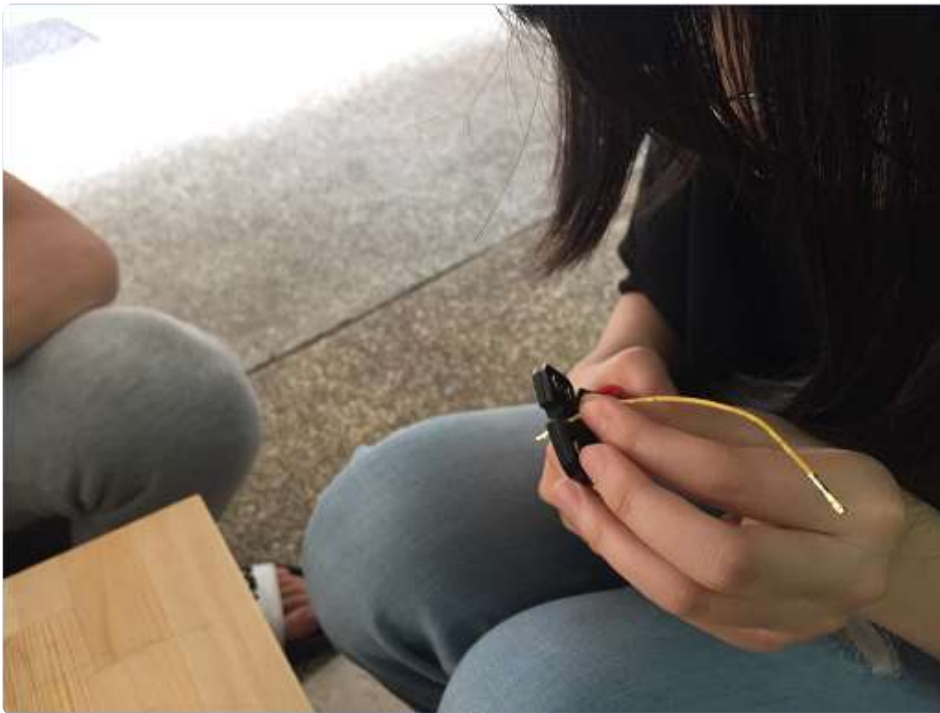
父親畢業於電子系，為一名工程師，媽媽則是交大管科系系友。爸爸經常從事程式設計與電機電子的設計與維修。耳濡目染之下，我對電子電機資訊抱有極大的好奇心，從小就開始培養數理邏輯思辨，也在國中參加數理資優社團，並在高中就讀數理實驗班，也參加校內程式比賽得到第2名，爸爸常對我談論國內就業市場，使我對這個領域產生濃厚興趣，也透過實際操作，如寫程式、組裝電子電路累積實際的經驗。爸爸總是提醒我，若要成為佼佼者，需兼備軟硬體能力，這樣在未來的發展才能有更大的空間。

## 電機能力之表現

## 專題動機說明

現在電子設備普及對於行動電力,或電源儲存需求相當的普遍,但是如何讓電子設備正常運作,則有賴於穩的電源供應..那麼正確的電壓偵測就非常重要,比方說,電源異常關機..





## 運用的知識與技術

物理

歐姆定律

數學

直線方程式

高中程式設計

C語言

多元學習

物聯網單晶片 (ESP8266)

## ESP8266 ADC公式推導

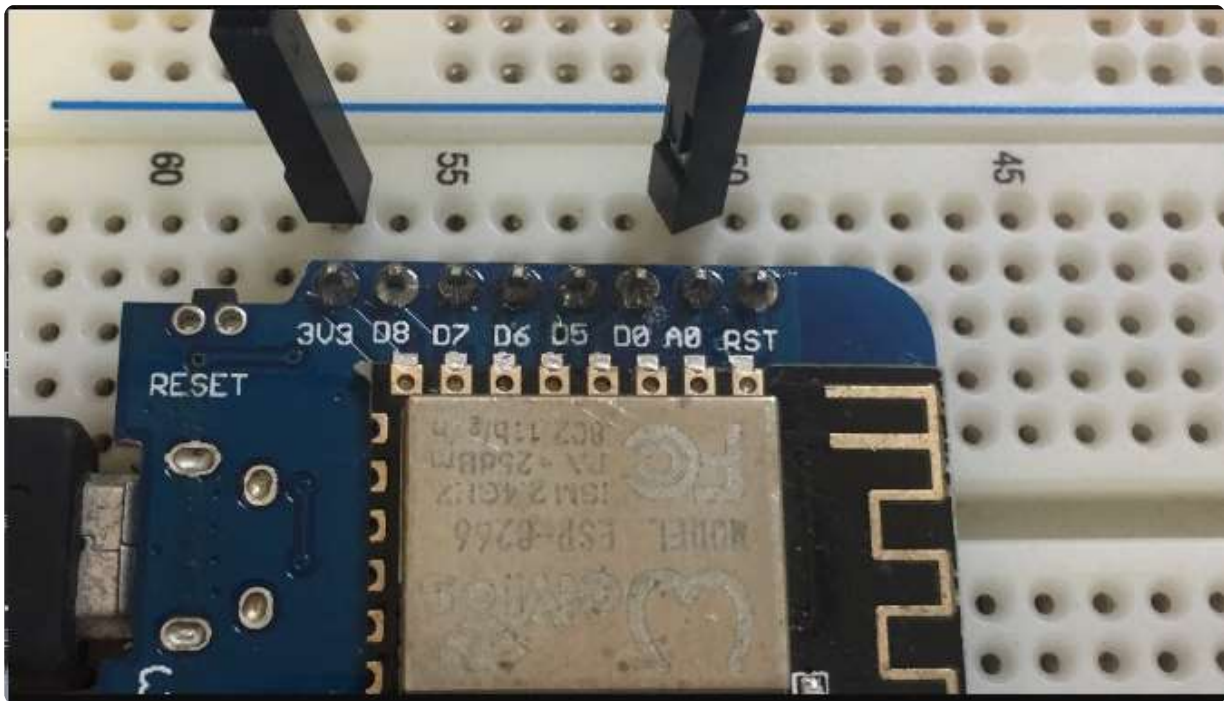
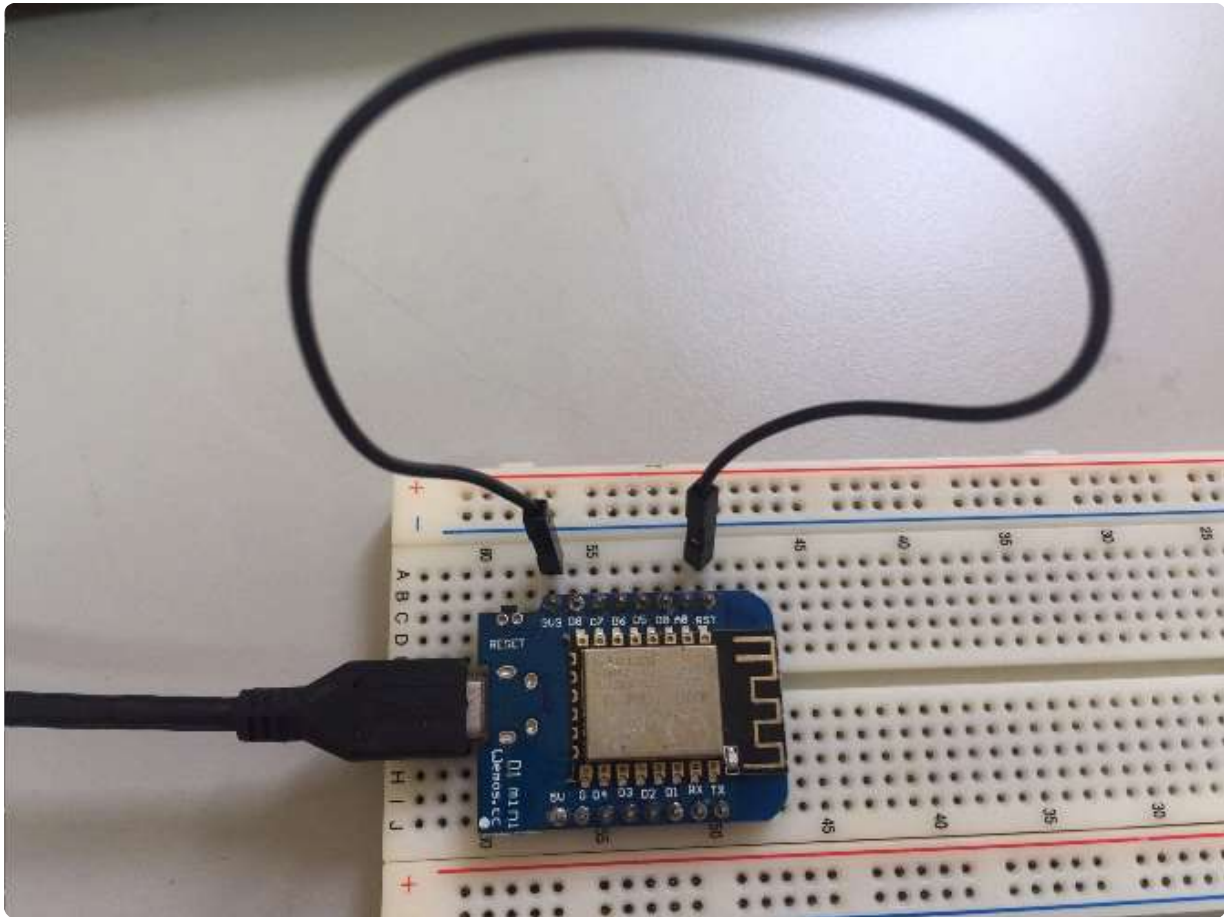
ADC

ESP8266有10bit(1024,0~1023),操作電壓為3.3V.也就是說把3.3V分成1024等分  
因此實際電壓為 $V_o = 3.3 * (V_{adc} / 1023)$

程式碼

```
1  #define analogPin A0 /* ESP8266 Analog Pin ADC0 = A0 */
2
3  int adcValue = 0; /* Variable to store Output of ADC */
4
5  void setup() {
6      Serial.begin(115200); /* Initialize serial communication at 115200 */
7  }
8
9  void loop() {
10     adcValue = analogRead(analogPin); /* Read the Analog Input value */
11
12     /* Print the output in the Serial Monitor */
13     Serial.print("ADC Value = ");
14     Serial.println(adcValue);
15
16     float VAO = 0.0;
17     float VRt = 0.0;
18     int i = 0;
19     float vinTotal = 0.0;
20     int value = 0;
21     float VIN = 0.0;
22     int sampleCount = 50;
23
24     for (i = 0; i < sampleCount; i++) {
25         adcValue = analogRead(analogPin); /* Read the Analog Input value */
26         VAO = ((adcValue * 3.3) / 1023.0);
27         /// Serial.println(VAO);
28
29         vinTotal = vinTotal + VAO;
30         delay(10);
31     }
32     Serial.print("Input = ");
33     // Serial.println(vinTotal/sampleCount);
34     VIN = vinTotal / sampleCount;
35     if (VIN <= 0 || VAO < 0.1) {
36         VIN = 0;
37     }
38
39     Serial.println(VIN);
40
41     delay(1000);
42 }
43
```

## 硬體線路



## 實作程式與輸出



## 實作程式

ADC2 | Arduino 1.8.15

File Edit Sketch Tools Help



ADC2

```
17 float VRt = 0.0;
18 int i = 0;
19 float vinTotal = 0.0;
20 int value = 0;
21 float VIN = 0.0;
22 int sampleCount = 50;
23
24 for (i = 0; i < sampleCount; i++) {
25   adcValue = analogRead(analogPin); /* Read the Analog
26   VAO = ((adcValue * 3.3) / 1023.0);
27   /// Serial.println(VAO);
28
29   vinTotal = vinTotal + VAO;
30   delay(10);
31 }
32 Serial.print("Input = ");
33 // Serial.println(vinTotal/sampleCount);
34 VIN = vinTotal / sampleCount;
35 if (VIN <= 0 || VAO < 0.1) {
36   VIN = 0;
37 }
38
39 Serial.println(VIN);
40
41 delay(1000);
```

```
Leaving...
Hard resetting via RTS pin...
```

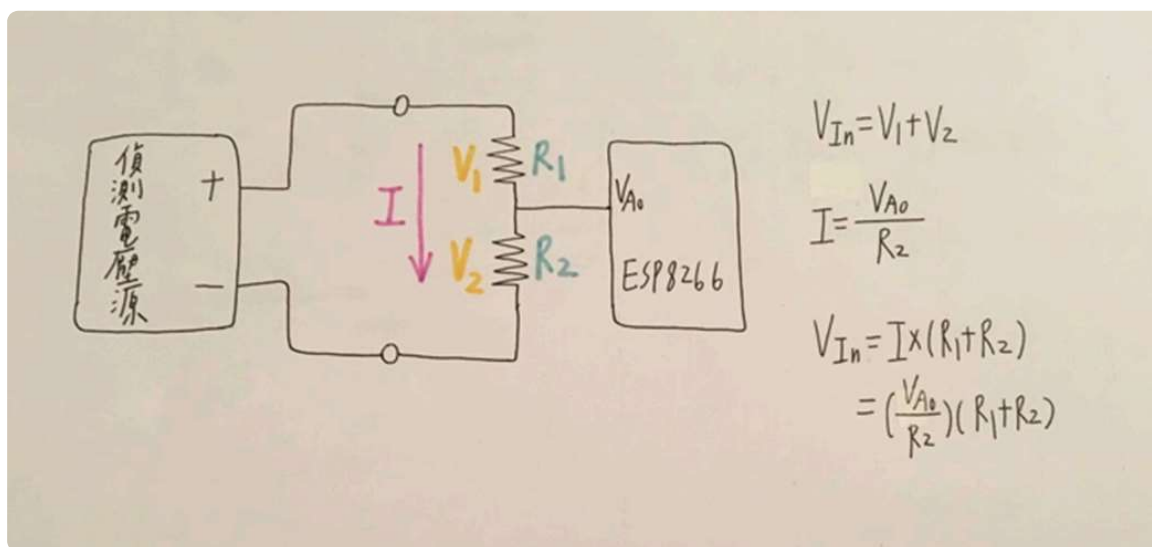
輸出

COM3

[illegible]

## 電壓擴展公式推導

因為ESP8266的ADC輸入操作最高為3.3V,這如果要讀去更高的電壓,將導致ADC燒毀  
採用歐姆定律分壓方式,讓ADC輸入電壓最高為3.3V或是搭配歐姆定律算出最高的偵測  
電壓



EX:計算VIN的最高電壓,採用R1=300K R2=75K,Va0=3.3V

$$V_{INmax} = (3.3/75K)(300K + 75K)$$

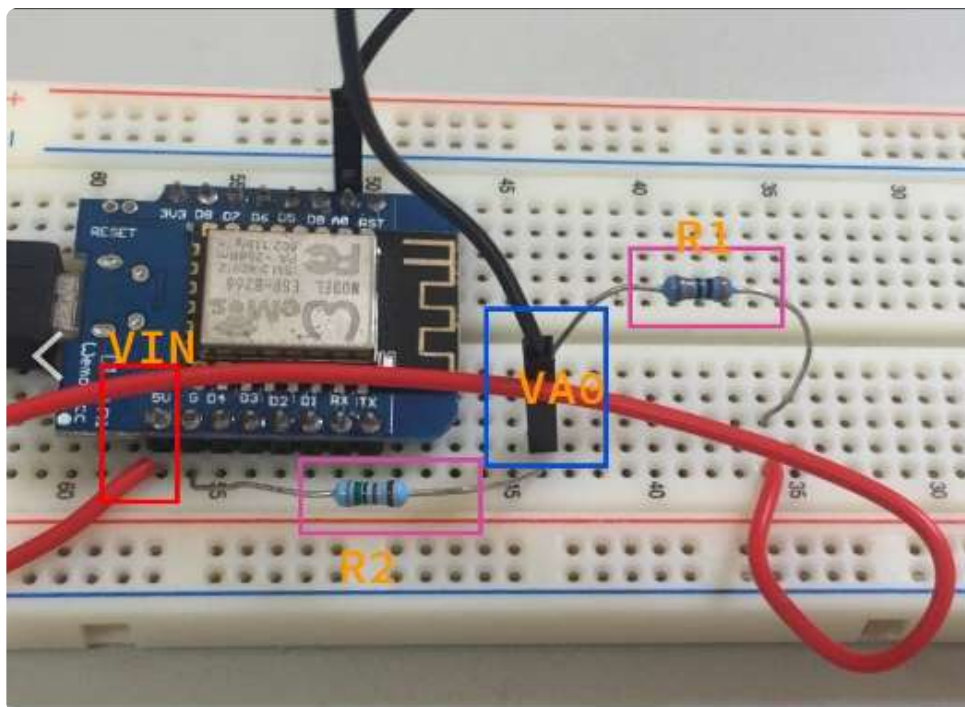
$V_{INmax}=16.5V$

也就是說此一電路最高輸入電壓為16.5V

```
1  #define analogPin A0 /* ESP8266 Analog Pin ADC0 = A0 */
2
3  int adcValue = 0; /* Variable to store Output of ADC */
4
5  float Rp = 0.0;
6  const int sampleCount = 50;
7
8  const float Vw = 3.2;
9  const float R1K = 1000.0;
10 const float R220K = 220 * R1K;
11 const float R100K = 100 * R1K;
12 const float R300K = 300 * R1K;
13 const float R75K = 75 * R1K;
14 void setup() {
15     Serial.begin(115200); /* Initialize serial communication at 115200 */
16
17 }
18
19 void loop() {
20     adcValue = analogRead(analogPin); /* Read the Analog Input value */
21
22     /* Print the output in the Serial Monitor */
23     Serial.print("ADC Value = ");
24     Serial.println(adcValue);
25
26     float VAO = 0.0;
27     float VRt = 0.0;
28     int i = 0;
29     float vinTotal = 0.0;
30     int value = 0;
31     float VIN = 0.0;
32     int sampleCount = 50;
33
34     for (i = 0; i < sampleCount; i++) {
35         adcValue = analogRead(analogPin); /* Read the Analog Input value */
36         VAO = ((adcValue * 3.3) / 1023.0);
37         /// Serial.println(VAO);
38         VRt = (VAO / R75K) * (R75K + R300K);
39         vinTotal = vinTotal + VRt;
40         delay(10);
41     }
42     Serial.print("Input = ");
43     // Serial.println(vinTotal/sampleCount);
44     VIN = vinTotal / sampleCount;
45     if (VIN <= 0 || VAO < 0.1) {
46         VIN = 0;
47     }
48
49     Serial.println(VIN);
50
51     delay(1000);
52 }
53
```



## 硬體線路



## 實作程式

ADC3ERR | Arduino 1.8.15

File Edit Sketch Tools Help

```

16 }
17 }
18
19 void loop() {
20   adcValue = analogRead(analogPin); /* Read the Analog Input value */
21
22   /* Print the output in the Serial Monitor */
23   Serial.print("ADC Value = ");
24   Serial.println(adcValue);
25
26   float VAO = 0.0;
27   float VRt = 0.0;
28   int i = 0;
29   float vinTotal = 0.0;
30   int value = 0;
31   float VIN = 0.0;
32   int sampleCount = 50;
33
34   for (i = 0; i < sampleCount; i++) {
35     adcValue = analogRead(analogPin); /* Read the Analog Input value */
36     VAO = ((adcValue * 3.3) / 1023.0);
37     /// Serial.println(VAO);
38     VRt = (VAO / R75K) * (R75K+R300K);
39     vinTotal = vinTotal + VRt;
40     delay(10);
  
```

## 程式輸出

```
COM3
Input = 4.51
ADC Value = 280
Input = 4.51
ADC Value = 280
Input = 4.51
ADC Value = 280
Input = 4.51
ADC Value = 280
Input = 4.51
ADC Value = 280
Input = 4.51
ADC Value = 280
Input = 4.51
ADC Value = 280
Input = 4.51
```

## 結果與討論

因為輸入電壓為esp8266標準電壓

可是公式輸出轉換,輸出只有4.5v

此結果為錯誤的

實際電壓為 $V_0 = 3.3 * (280 / 1023)$

$V_0 = 0.903225806$

轉換電壓

$V_{in} = (V_0 / 75K) * (300K + 75K)$

$V_{in} = 4.51V$

這表示程式與手算相同

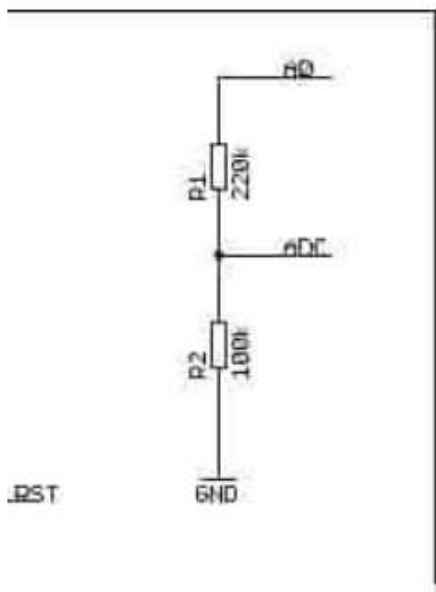
## 錯誤嘗試

再確認與公式無關後..查詢網路資料

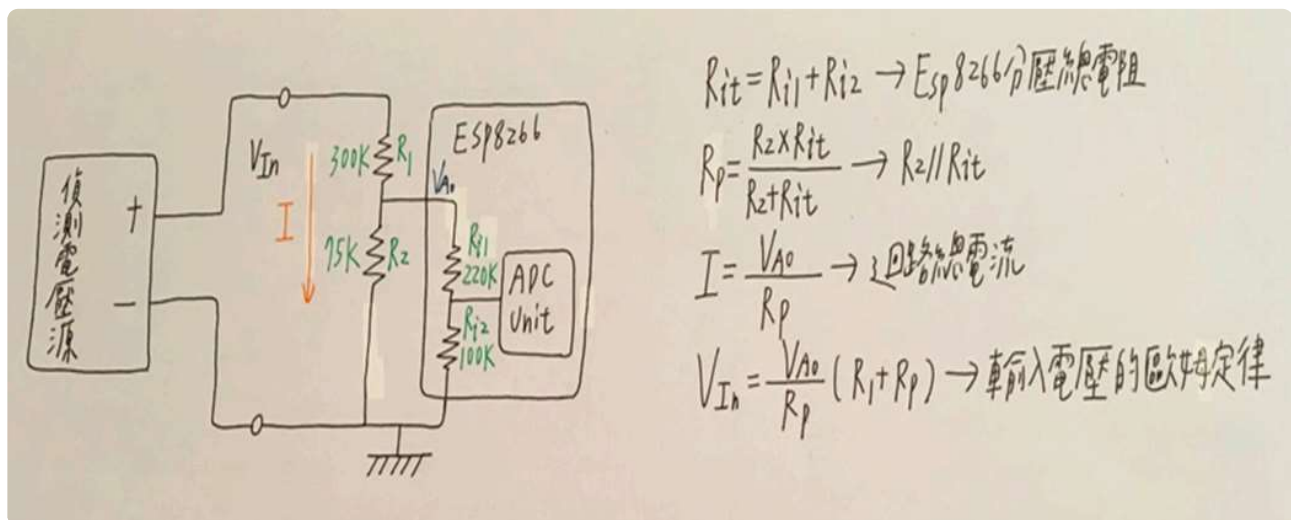
[Wemos D1 A0 Pin內部線路]

(<https://www.codrey.com/arduino-projects/wemos-d1-mini-an-introduction/>)

發現在這一顆晶片中有自己的電阻分壓電路



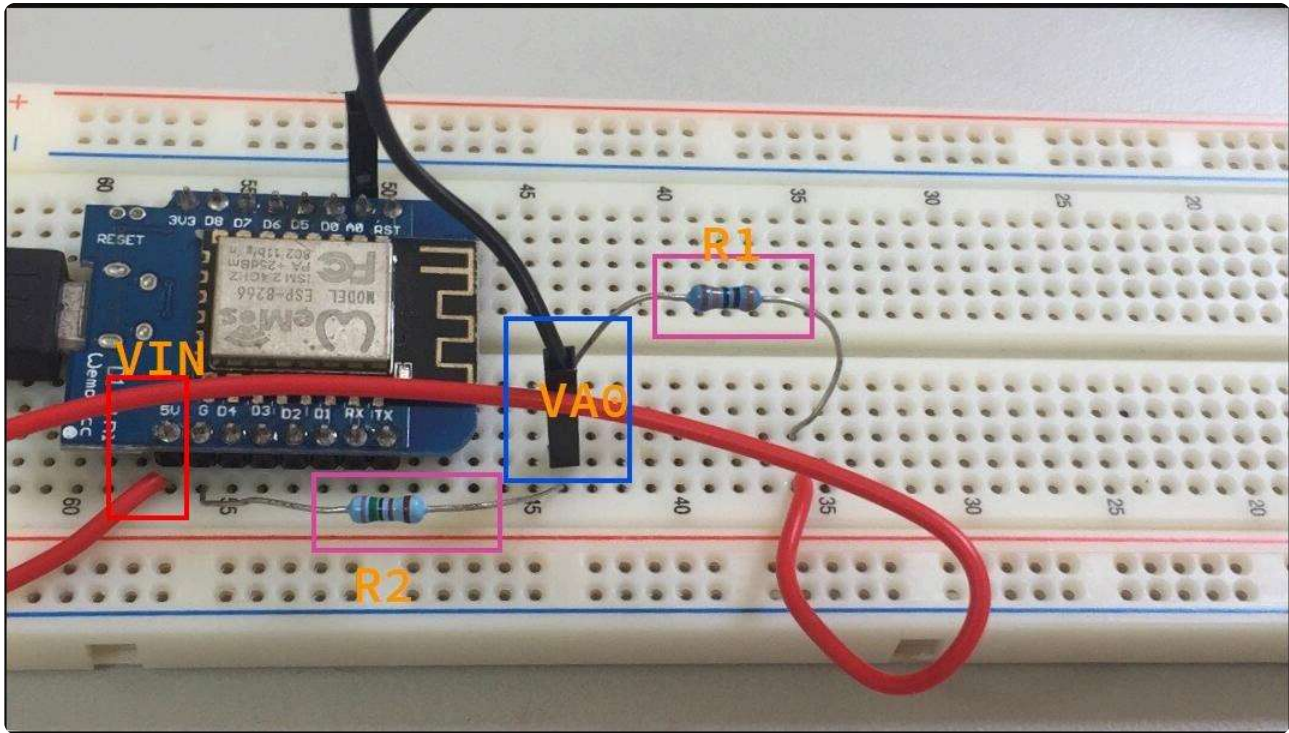
因此這兩顆電阻會影響到分壓公式,必須要將分壓再作調正  
才可以得到正確的輸入電壓



配合新公式程式碼

```
1  #define analogPin A0 /* ESP8266 Analog Pin ADC0 = A0 */
2
3  int adcValue = 0; /* Variable to store Output of ADC */
4
5  float Rp = 0.0;
6  const int sampleCount = 50;
7
8  const float Vw = 3.2;
9  const float R1K = 1000.0;
10 const float R220K = 220 * R1K;
11 const float R100K = 100 * R1K;
12 const float R300K = 300 * R1K;
13 const float R75K = 75 * R1K;
14 void setup() {
15     Serial.begin(115200); /* Initialize serial communication at 115200 */
16     Rp = R75K * (R100K + R220K) / (R75K + R100K + R220K);
17 }
18
19 void loop() {
20     adcValue = analogRead(analogPin); /* Read the Analog Input value */
21
22     /* Print the output in the Serial Monitor */
23     Serial.print("ADC Value = ");
24     Serial.println(adcValue);
25
26     float VAO = 0.0;
27     float VRt = 0.0;
28     int i = 0;
29     float vinTotal = 0.0;
30     int value = 0;
31     float VIN = 0.0;
32     int sampleCount = 50;
33
34     for (i = 0; i < sampleCount; i++) {
35         adcValue = analogRead(analogPin); /* Read the Analog Input value */
36         VAO = ((adcValue * 3.3) / 1023.0);
37         /// Serial.println(VAO);
38         VRt = (VAO * (Rp + R300K) / Rp)*0.94;
39         vinTotal = vinTotal + VRt;
40         delay(10);
41     }
42     Serial.print("Input = ");
43     // Serial.println(vinTotal/sampleCount);
44     VIN = vinTotal / sampleCount;
45     if (VIN <= 0 || VAO < 0.1) {
46         VIN = 0;
47     }
48
49     Serial.println(VIN);
50
51     delay(1000);
52 }
53
```

## 實體電路



電壓公式調整過後,得到的輸出

```
COM3
Input = 5.04
ADC Value = 280
Input = 5.03
ADC Value = 280
Input = 5.03
ADC Value = 280
Input = 5.03
ADC Value = 280
Input = 5.03
ADC Value = 280
Input = 5.03
ADC Value = 280
Input = 5.03
ADC Value = 280
Input = 5.03
ADC Value = 280
Input = 5.03
Autoscroll Show timestamp
```

此一公式\*0.94只是利用直線方程式的概念來做一個簡單的調整線性

$$VRt = (VA0 * (Rp + R300K) / Rp) * 0.94;$$

因為再經過電阻分壓以及ADC會產生數值精密度上的問題

## 進一步探索



在物理中的歐姆定律中,可以運用數學可以很精準地算出一個正確的數字(譬如幾伏特,幾歐姆,幾安培),可是在實際的硬體電路及機晶片設計中,會因為以下等因素影響歐姆定律中的運算數值產生誤差.

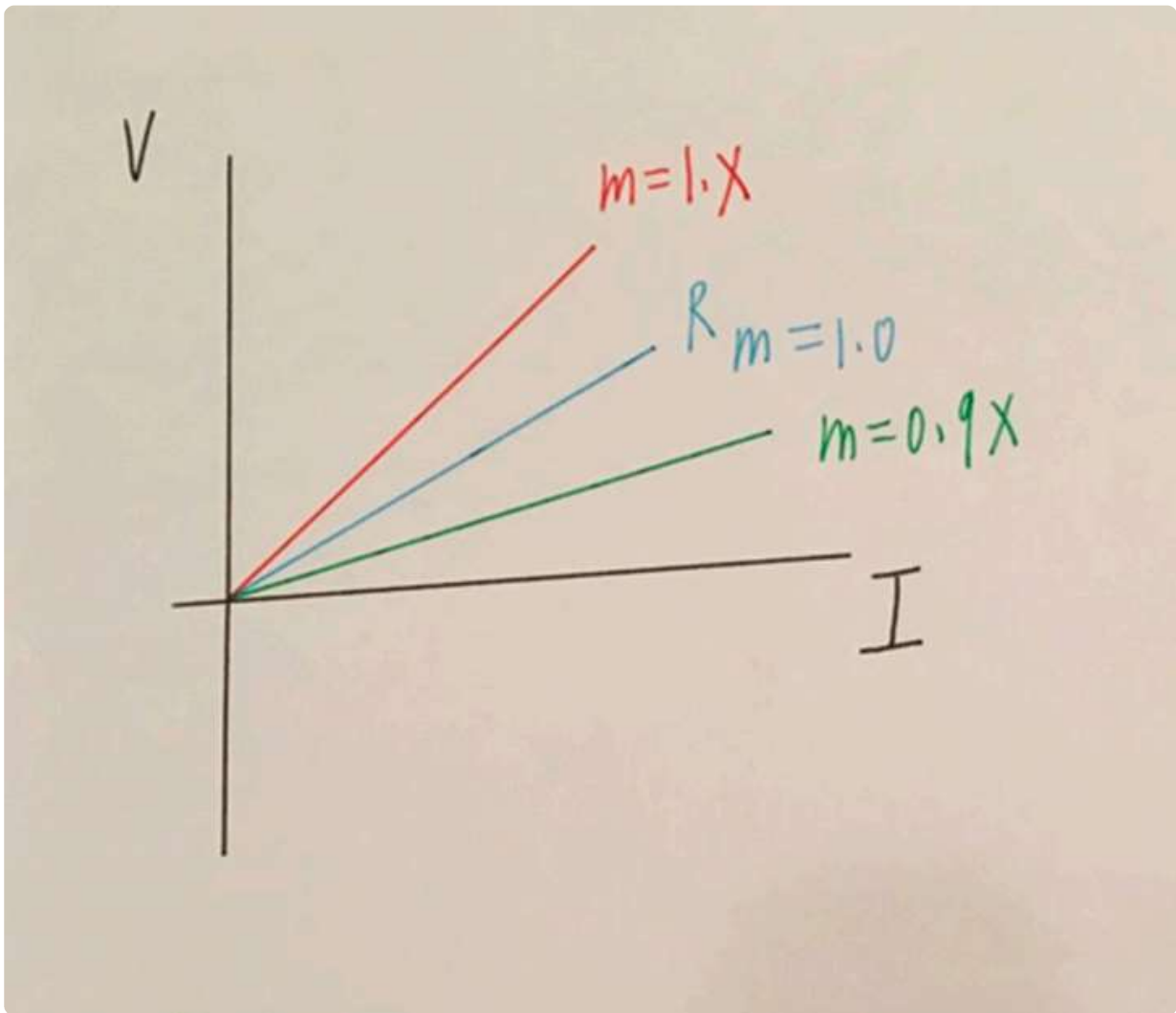
### 1. 電阻材料特性

電阻材料及製造,在使用上有精密度之分. $\pm 0.5\%$ , $\pm 5\%$ 等等...

本實驗電壓偵測所以是使用精密電阻,來降低多次運算後的誤差及提高精準度(ESP8266 分壓電阻的精密度是未知)

由歐姆定律,可以發現, $R=V/I$ ,斜率為1

可是當精密度有所改變( $\pm 0.5\%$ , $\pm 5\%$ 等等...),斜率將隨之改變,如圖說明(此圖為了要表示斜率差異,因此只是示意圖,無刻度表示)



,導致讀值會產生和數學運算的差異,因此程式計算就要取一個斜率係數做補償

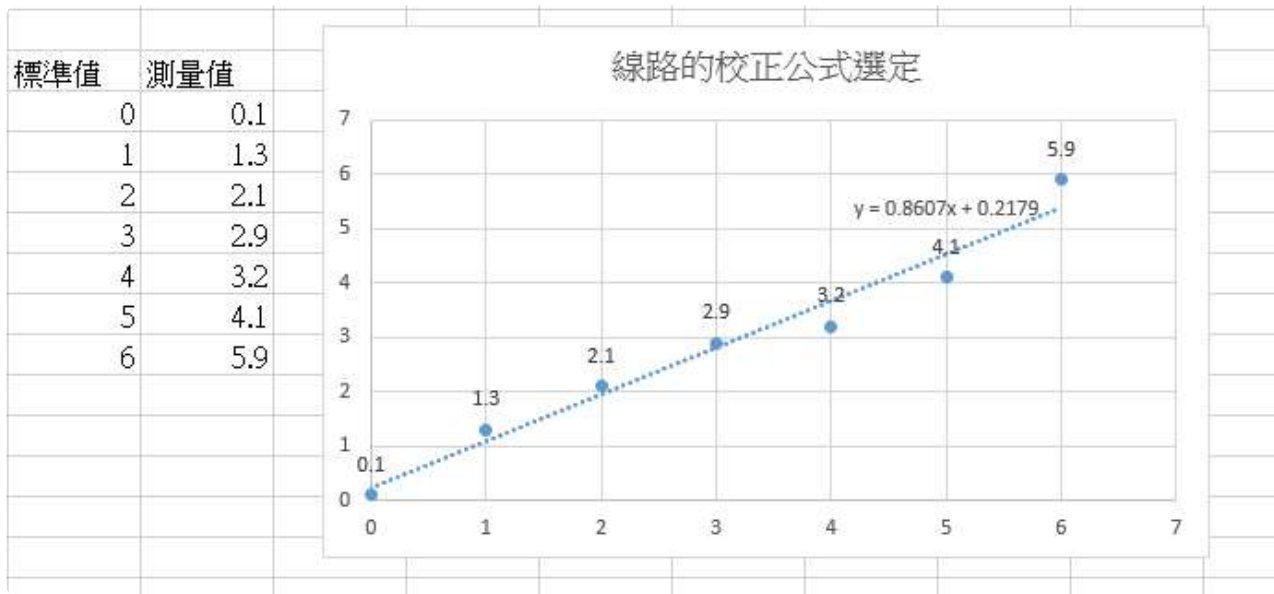
```
adcValue = analogRead(analogPin); /* Read the Analog Input value */
VAO = ((adcValue * 3.3) / 1023.0);
/// Serial.println(VAO);
Vrt = (VAO * (Rp + R300K) / Rp) * 0.94;
```

ps:係數選定說明

未來進入大學後,如果有更多的儀器設備,將可以用測量值與標準值來找出最整確的直線方程式來逼近實際硬體的輸出.

以下為簡單的測試方式





參考資料(只取線性):

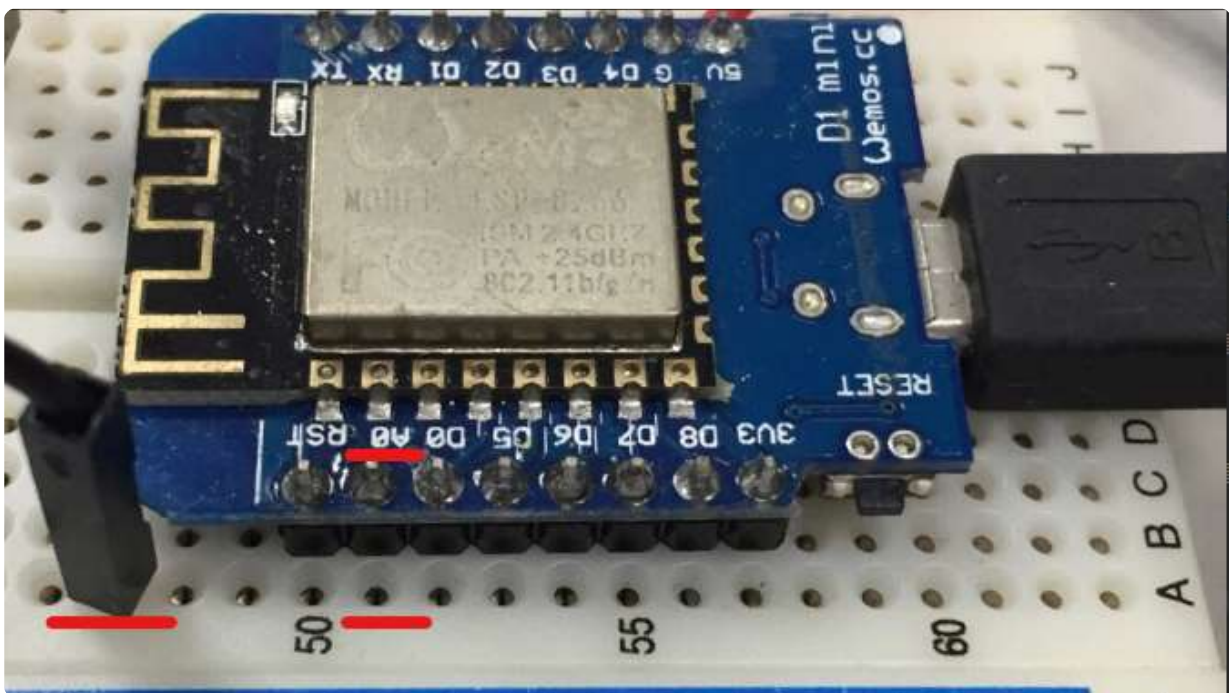
[Excel 快速加入各種趨勢線教學：線性、指數、對數、多項式等迴歸線]

<https://officeguide.cc/excel-scatter-plot-add-regression-trendline-tutorial/>

## 2. 晶片材料特性

譬如ADC公式: $V_o = 3.3 * (V_{adc} / 1023)$ .. $V_o = 0$ ,則 $V_{adc}$ 應該是要為0,可是實際上 $V_{adc}$ 不曾為0,如圖示

A0 Pin空接



實際上ADC是有輸出的

```
COM3
|
ADC Value = 28
ADC Value = 27
ADC Value = 28
ADC Value = 28
ADC Value = 28
ADC Value = 27
ADC Value = 28
ADC Value = 28
ADC Value = 28
ADC Value = 28
ADC Value = 27
ADC Value = 27
ADC Value = 28
ADC Value = 28
ADC Value = 28
```

可能為ADC中微量的信號作用,這會是在我到大學後需要進一步的探討,但可以軟體程式能力來改善.在邏輯上,可以判斷出ADC那一種輸出時為無效輸出,因此可以藉由軟體程式加以過濾,以免使用人員造成異常判讀.

```
if (VIN <= 0 || VAO < 0.1) {
    VIN = 0;
}
```

另外ADC再轉換可能材料因素外,也可能因為線路的雜訊,或是干擾..在同一個輸入下,每一次取樣,會有不同的結果.

```
COM3
ADC Value = 280
ADC Value = 279
ADC Value = 281
ADC Value = 281
ADC Value = 281
ADC Value = 280
ADC Value = 281
ADC Value = 279
ADC Value = 279
ADC Value = 279
ADC Value = 280
ADC Value = 281
ADC Value = 280
ADC Value = 281
ADC Value = 282
```

因此可以採用多次採樣取得平均穩定值

```
for (i = 0; i < sampleCount; i++) {  
    adcValue = analogRead(analogPin); /* Read the Analog Input value */  
    VAO = ((adcValue * 3.3) / 1023.0);  
    /// Serial.println(VAO);  
  
    vinTotal = vinTotal + VAO;  
    delay(10);  
}  
Serial.print("Input = ");  
// Serial.println(vinTotal/sampleCount);  
VIN = vinTotal / sampleCount;
```

## 結論

為了完成電壓偵測儀需要用到之前國高中所學數學、物理、程式C語言以及結合簡易型晶片程式設計。

另外之後可以繼續研究的方向是降低對偵測源的能量耗損, 電壓偵測儀不應該耗損太多偵測源的能量。由電功率的公式得知  $P=V \cdot I$  功率耗損和電流成正比,  $P=V^2/R$  和電阻成反比, 因此只要採用高阻抗的分壓電路, 即可以降低對偵測源的能量耗損。甚至進一步設計晶片 IC, 滿足多樣的使用方式。

相信進入大學之後可以習得更多電機電子資訊領域這方面的知識, 進一步整合軟硬體, 希望未來可以在相關產業應用發揚光大所學知識。

最後編輯：

卡米特

0

10

加入留言

發表於  HackMD