

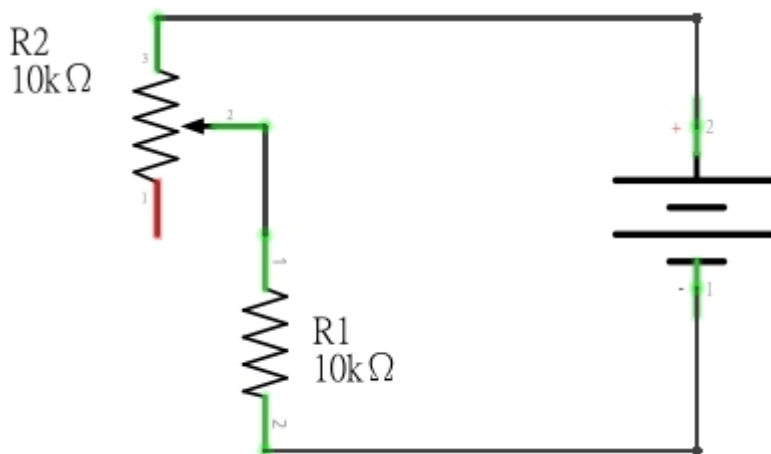
## 單元 1-3-1：類比訊號輸入/輸出

### 學習目標：

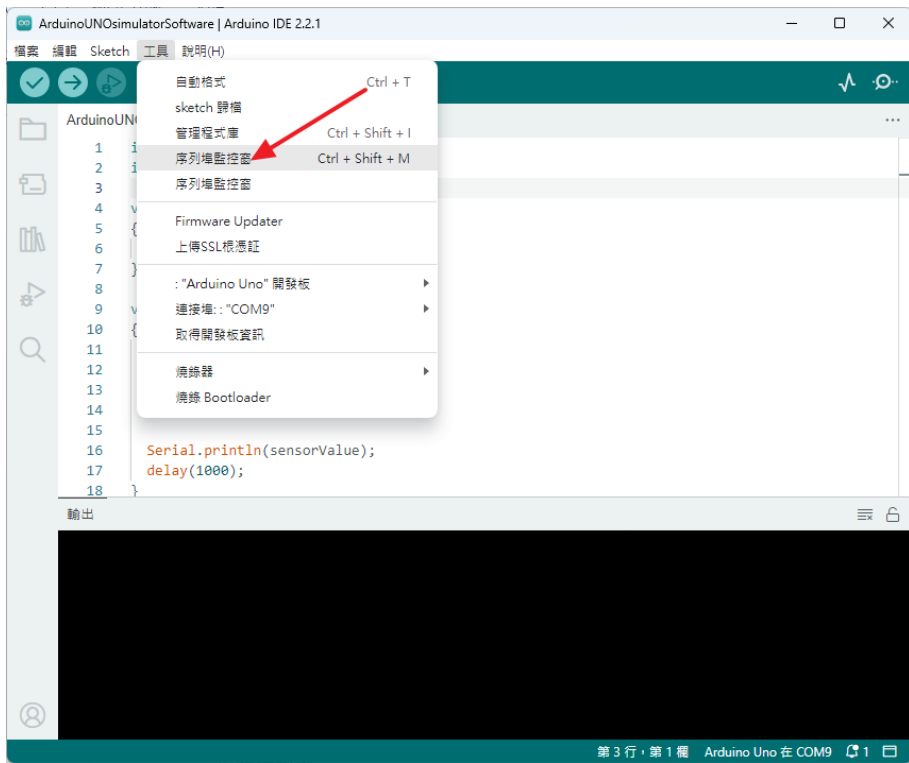
- 了解類比訊號的特性
- 了解指令運用
  - a. `analogRead`
  - b. `Serial.begin`
  - c. `Serial.println(sensorValue);`
  - d. `dacWrite();`
- 了解【序列埠監控窗】的使用
- 實際操作可變電阻分壓電路連接

### 理論基礎：

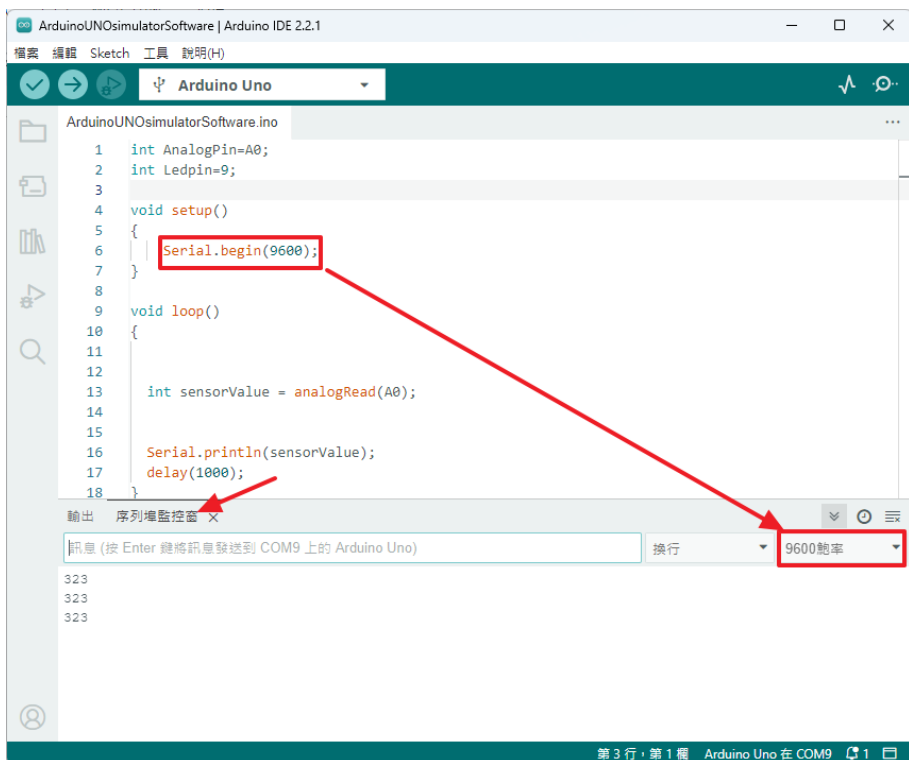
- 分壓定則：利用可變電阻進行分壓，調整可變電阻即可得到不同的電壓值，

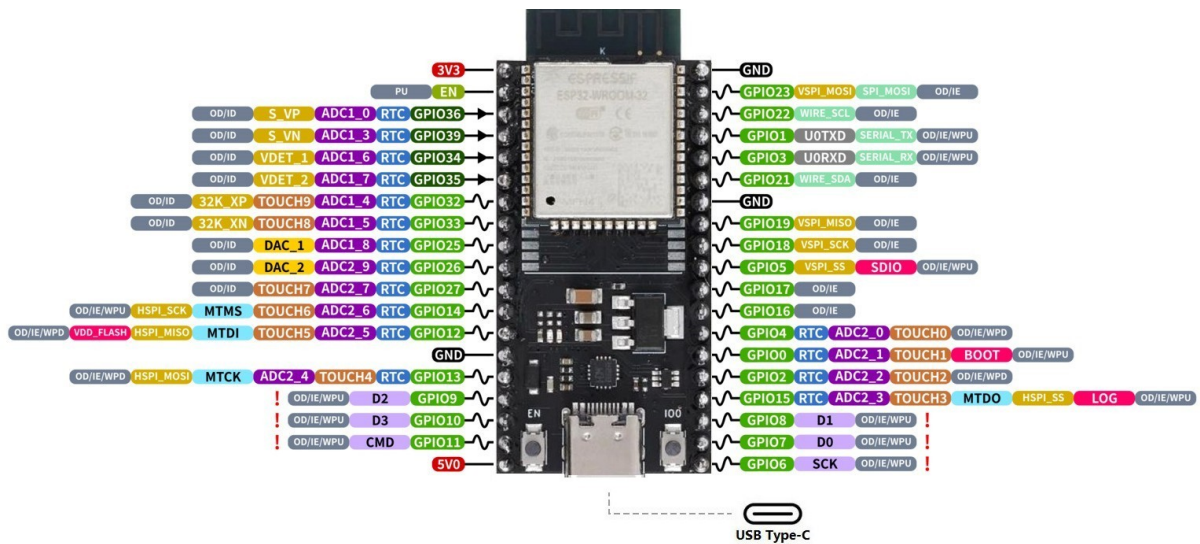


- **序列埠監控窗**：為了能即時讀取開發板數據，可以利用程式配合序列埠監控窗將資料同步傳到電腦。可在在電腦監測開發板數據，下圖為開啟序列埠監控窗的方式：  
**ESP32 ADC** 接腳能夠讀取類比電壓。它能將 0 到 3.3V 範圍的工作電壓轉換為 0 到 4095 間的數位值，(實際量測可能因儀表造成誤差，但整體上升、下降趨勢來算精準)。
- **DAC**：(Digital-to-Analog Converter，數位轉類比轉換器)：數位訊號 → 類比訊號 (如音樂播放器、音響系統)。
- **ADC**：(Analog-to-Digital Converter，類比轉數位轉換器)：類比訊號 → 數位訊號 (如感測器、麥克風、相機)。



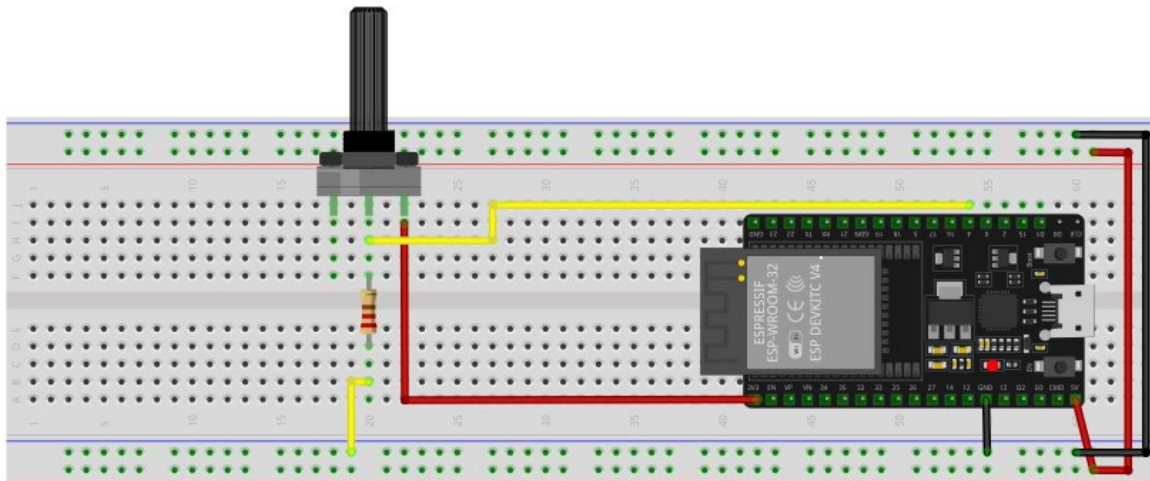
下圖右下的鮑率值(傳輸速率), 需與程式中 `Serial.begin(9600)` 相同, 以此程式為例鮑率為 9600





### 電路接線圖：

注意類比輸入接腳為 GPIO 4。



### 材料需求：

- ESP32 開發板 X1
- 可變電阻 10KΩ X1
- 電阻 10kΩ X1
- LED X1

### 設備需求：

**指令說明：**

本網站將隨著程式的編寫，逐步說明指令功能，請大家循序漸進學習。

**a. 讀取類比訊號 `analogRead`**

例：`analogRead()`； //將 GPIO 4 接腳的訊號讀取，並依比例轉成 0~4095 的值

**b. 設定序列埠鮑率 `Serial.begin`**

例：`Serial.begin(9600)`； //設定列埠監控窗鮑率值(傳輸速率)，需與 Arduino IDE 設定一致。

**c. 序列埠監控窗列印 `Serial.println`**

例：`Serial.println(sensorValue)`； //將 `sensorValue` 的數值，於序列埠監控窗呈現，並跳到下一行

**d. 數位轉類比指令 `dacWrite`;**

例：`dacWrite(DAC1, 255)`； // 在第 25 腳輸出 3.3V  
`dacWrite(DAC1, 0)`； // 在第 25 腳輸出 0V

**程式編碼 ADC**

---

```
int AnalogPin=4;           // 宣告 AnalogPin 變數為 4 (GPIO 4)

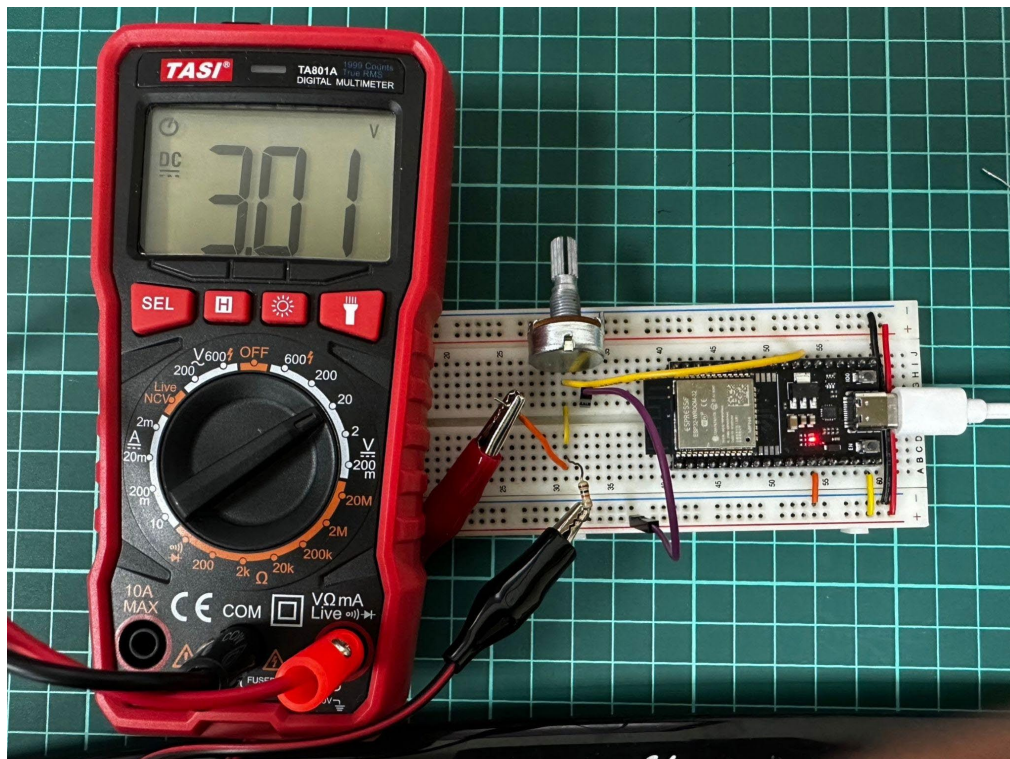
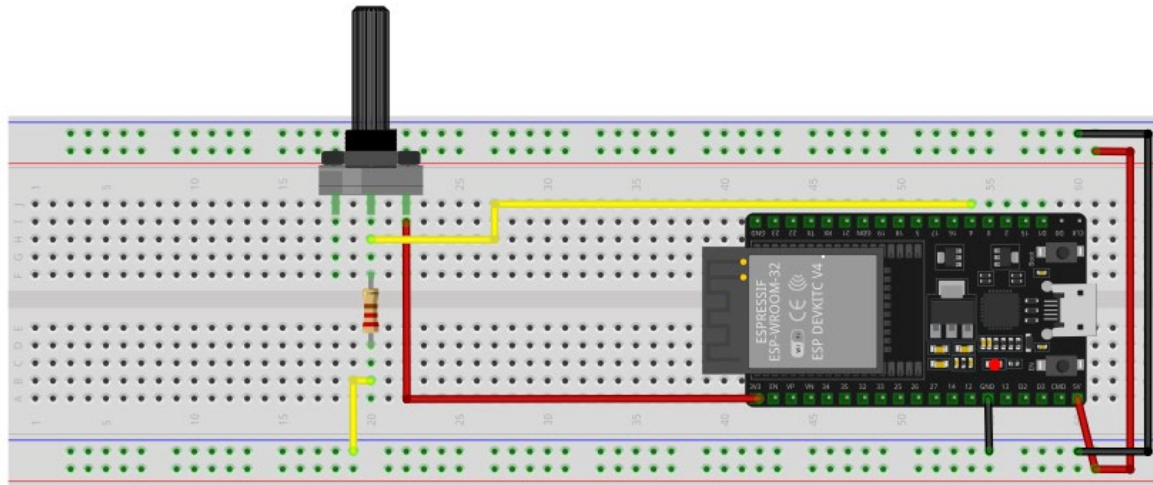
void setup()
{
    Serial.begin(9600);     //設定鮑率值，一般用 9600 即可
    pinMode(AnalogPin, INPUT);
}

void loop()
{
    //將 AnalogPin 接腳電壓，依比例轉成 0~4095 數值，並將數值放入 sensorValue 變數中
    int sensorValue = analogRead(AnalogPin);

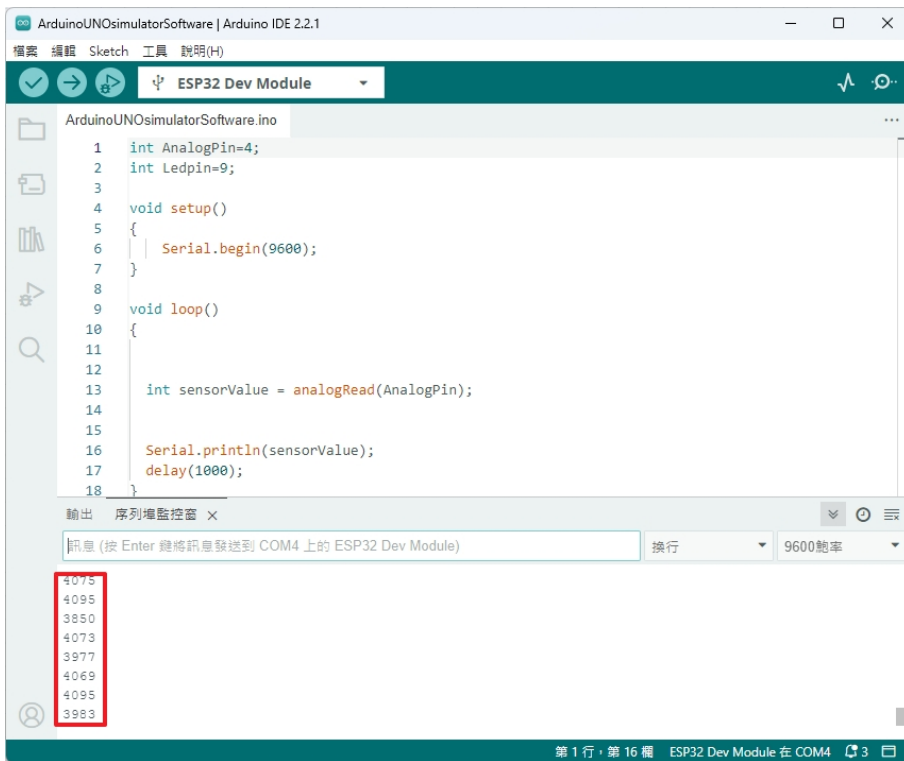
    //將 sensorValue 的數值，於序列埠監控窗呈現，並跳到下一行
    Serial.println(sensorValue);
    delay(1000);           //延遲 1 秒
}
```

---

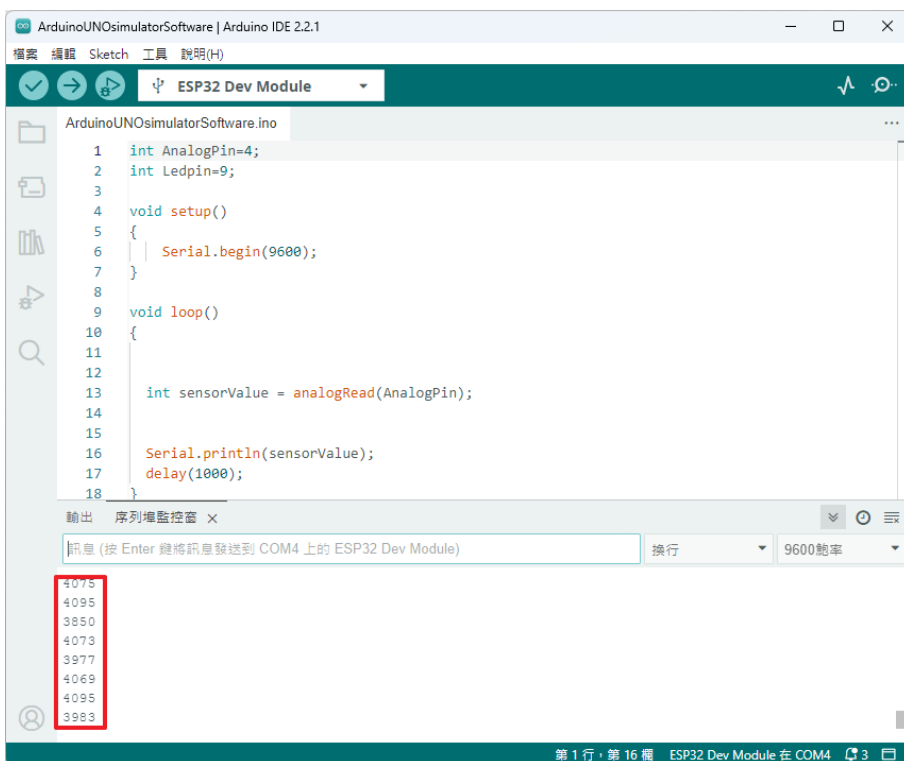
上傳完成後，你也就可以，利用三用電表量測電源及 GPIO4 的輸入電壓，並觀察序列埠監控窗數值，判斷量測是否精準。



分壓後的 GPIO4 的輸入電壓為 3.01V



序列埠監控窗，數值為 4000 上下。



由下列計算，計算值於檢測值或許有些差距，但趨勢變化是正確的，但一般使用還算可以接受。

$$\text{sensorValue} = \frac{3.01V}{3.3V} \times 4095 \cong 3735$$

## 單元 1-3-2：數位轉類比訊號輸出 DAC

### 程式編碼

```
int brightness = 0;           //設定 brightness 為 LED 的亮度整數變數值初始  
為 0  
int Vadd=8;                  //設定 Vadd 為累加(減)的整數變數值  
  
void setup()  
{  
    Serial.begin(9600);      //設定鮑率值，一般用 9600 即可  
}  
  
void loop()  
{  
    Serial.println(brightness);  
  
    brightness = brightness + Vadd; //brightness 值每次迴圈+32  
  
    if (brightness <= 0 || brightness >= 255) //條件判斷 brightness 亮度值若  
brightness<=0 或>= 255  
    {  
        Vadd = -Vadd; //若 if 條件符合 fadeAmount 改變正負值  
    }  
    dacWrite(DAC1, brightness); //由 DAC1 接腳(GPIO25)，數位轉類比訊號  
輸出  
    delay(100); //延遲 0.1 秒  
}
```

運用三用電表量測電壓，依照下圖接線並觀察 LED 燈

