

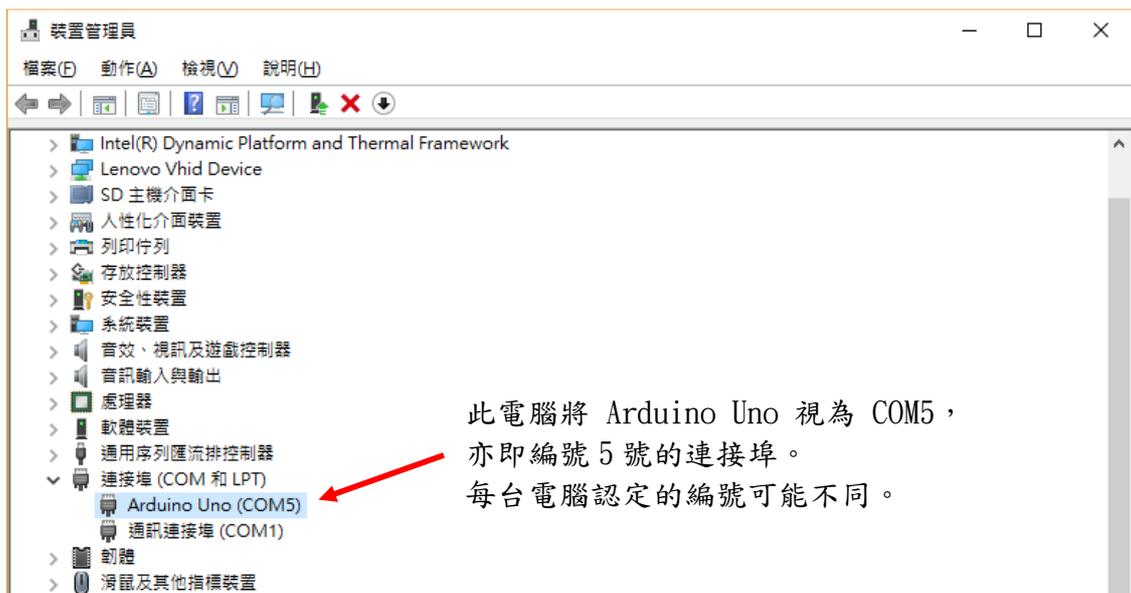
# 使用 Arduino 讀取電壓 簡介

(東華大學 普通物理實驗 教學實驗室)

本文將簡介如何用 Arduino Uno 開發板讀取電壓數值到電腦上顯示。

首先，電腦中得安裝 Arduino IDE，可在 <https://www.arduino.cc/en/Main/Software> 下載。IDE是 集成開發環境(Integrated Development Environment，簡稱IDE) 的意思。

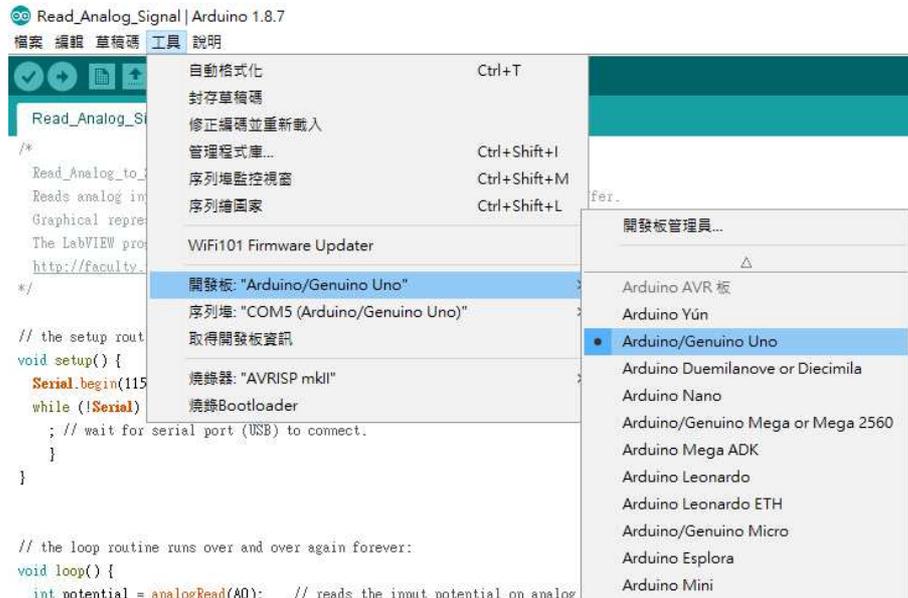
安裝完成後，將 Arduino Uno 開發板 用USB接線連接到電腦。電腦應該會偵測到它，並將它認定為一個 通訊連接埠。在電腦 裝置管理員 中應可以看到類似下面的裝置出現。(第一次安裝開發板驅動程式有時需要好一會，需耐心等待)



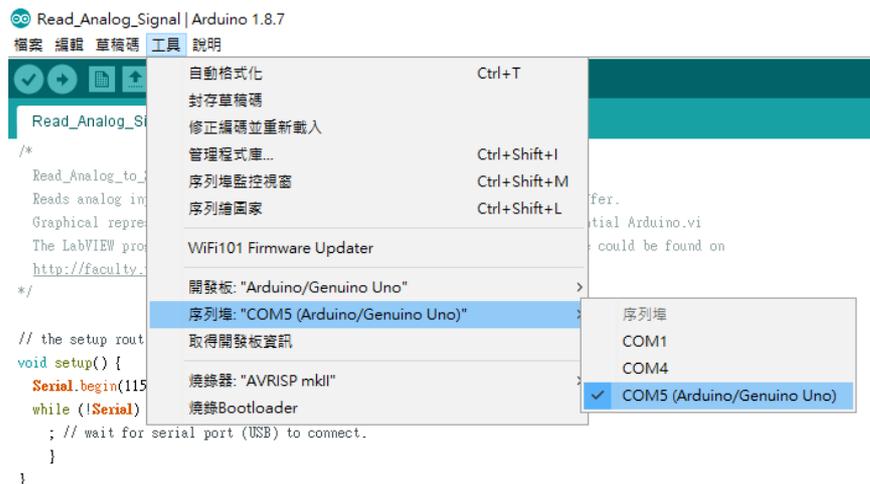
安裝 Arduino IDE 之後，可在電腦中找到 arduino.exe，其圖示如下。



執行 arduino.exe 之後，在程式功能表的“工具”中的“開發板”選項裡，選擇 Arduino Uno，如下圖。



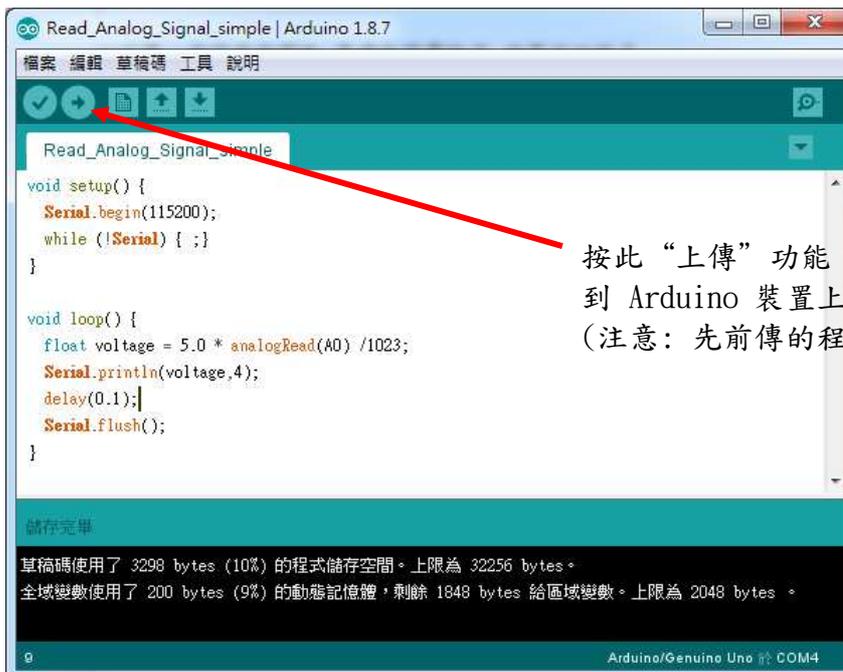
在程式功能表的“工具”中的“序列埠”選項裡，選擇電腦幫 Arduino Uno 編號的序列埠，例如下圖。



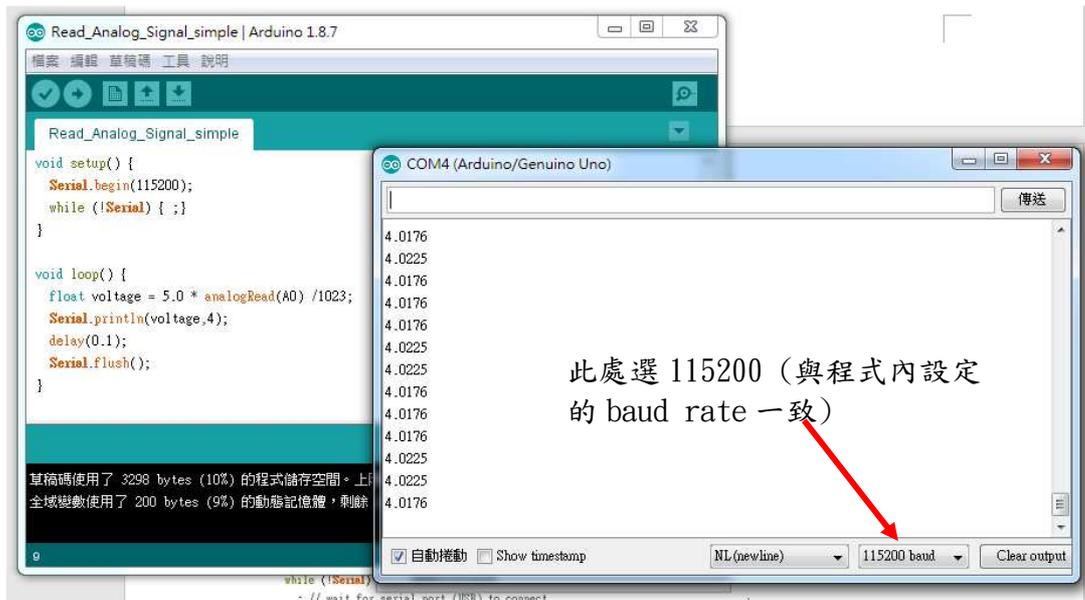
將以下指令碼複製到 Arduino 程式編譯器視窗中，並按“上傳”功能，即可讓 Arduino Uno 開發板 不斷讀取板子上 A0 接孔的電壓。(各行指令碼的說明可見之後的解說)

```
void setup() {  
  
  Serial.begin(115200);  
  
  while (!Serial) { ;}  
  
}  
  
void loop() {  
  
  float voltage = 5.0*analogRead(A0)/1023;  
  
  Serial.println(voltage,2);  
  
  delay(0.1);  
  
  Serial.flush();  
  
}
```

如下圖。亦可直接開啟 普通物理實驗室 編寫好的程式  
Read\_Analog\_Signal\_simple.ino



要將 Arduino 開發板上讀到的數值傳回電腦，可從“工具”功能中選擇“序列埠監控視窗”或是“序列繪圖家”。如下圖所示。



“序列埠監控視窗”會以文字的方式顯示資訊。

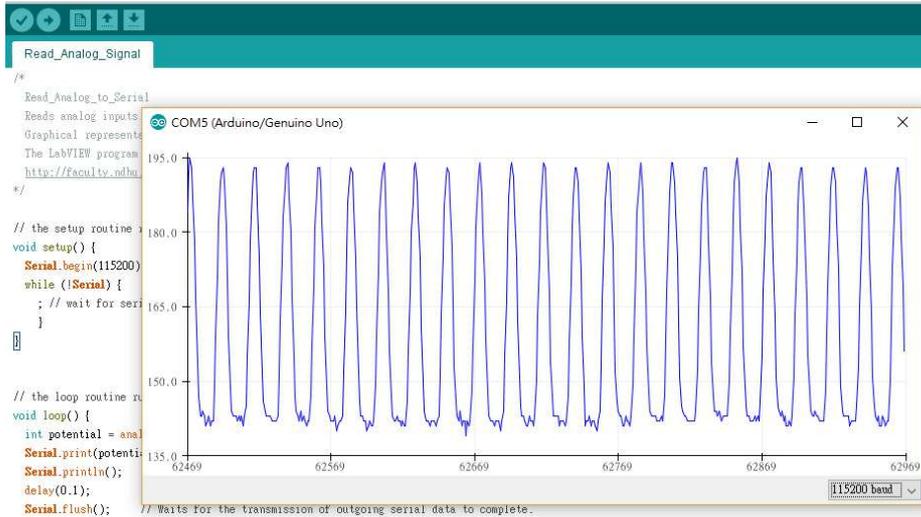
Read\_Analog\_Signal | Arduino 1.8.7

檔案 編輯 草稿碼 工具 說明



Read\_Analog\_Signal | Arduino 1.8.7

檔案 編輯 草稿碼 工具 說明



“序列繪圖家” 會將數值資訊以 圖 的形式呈現。

Arduino的程式指令(函數)可以參考以下網頁:

<https://www.arduino.cc/reference/en/#functions>

以下各別說明上面的程式碼意思。

程式碼 `void setup() {}` 與 `void loop() {}`

`setup()`指令是在其後的{}中寫下“設定”的指令，告訴Arduino你的執行環境設定(`setup`)。`setup()`只會在Arduino程式中執行一次。相對的，`loop()`指令則會不斷循環執行其後的{}中的指令。亦即從`loop() {}`裡面的第一行 `float voltage = 5*analogRead(A0)/1023;` 執行到最後一行 `Serial.flush();` 然後又回到第一行執行，不斷重複。

程式碼 `Serial.begin(115200)`

`Serial.begin(115200)`是設定Arduino與電腦間的數據傳輸率，()裡面的數值稱為鮑率(baud rate)。這個設定值會決定出Arduino與電腦間的USB介面所模擬出的序列埠(Serial)會以怎樣的速率來傳輸訊號(每秒多少位元，每個位元是0或1)。如果Arduino輸出訊號的鮑率(baud rate)與電腦端設定接收的鮑率(baud rate)不一樣，則接收端會收到錯誤的訊號，得到的訊息常形成一堆亂碼。

( <https://www.arduino.cc/reference/en/language/functions/communication/serial/begin/> )

程式碼 `while (!Serial) { ; }`

`while`指令會判斷()中的條件，若條件值是“真(True)”，則執行{}裡面的指令，然後再次判斷()中的條件。直到()中的條件值變成”否(False)”，則不再執行{}裡面的指令，並前往下面的指令。(詳見

<https://www.arduino.cc/reference/en/language/structure/control-structure/while/> )

上面的{}裡面的指令是空白的“;”，也就是沒做甚麼事，單純等待()中的條件是否達成而已。

`while (!Serial)`中的“Serial”程式碼，會判斷電腦是否成功透過序列埠(Serial)連接上Arduino。如果成功連接，則Serial回傳值是“真(True)”，如果未成功連接，則Serial回傳值是”否(False)”。

“Serial”前面的驚嘆號，功能是將驚嘆號後面的值變“相反”。亦即“未成功”連接時“!Serial”回傳值變成”真(True)”(否的相反)，此時while迴圈繼續執行，直到Serial 成功連接，而“!Serial”回傳值變成“否”，才跳出迴圈。

(詳見

<https://www.arduino.cc/reference/en/language/functions/communication/serial/ifserial/> )

以下這段指令是設定Arduino在透過序列埠(Serial)將訊號傳回電腦時，鮑率(baud rate)設定為每秒115200個位元，然後等待序列埠成功連接上。

```
void setup() {  
  Serial.begin(115200);  
  while (!Serial) { ;}  
}
```

程式碼 `analogRead(A0)`

在 `float voltage = 5.0*analogRead(A0)/1023;` 這行指令中，等號前的 `voltage` 是自己設定的變數，`voltage` 也可以改成任何文字。此處用 `voltage` 這個字是因為它代表讀到的電壓(voltage)值。

`voltage` 前面的 `float` 是宣告 `voltage` 這個自己訂的變數的格式是浮點數(帶有小數點的數值)格式。這樣電腦才知道要怎麼適當看待 `voltage` 這個變數。

`analogRead(A0)` 這個指令，代表從 `A0` 這個通道讀(read)一個類比(analog)訊號。而 `A0` 通道是 Arduino 開發板上第0號的類比(analog)通道。通常，電腦上一系列物品的編號是從0號開始編的。另外，`analogRead()` 讀回的數值是0~1023的整數，其中讀回0代表0伏特電壓，而讀回1023代表滿電壓(對Arduino來說通常是5伏特電壓)。因此 `5.0*analogRead(A0)/1023` 這個程式碼可以將A0通道讀回的數值(0-1023)轉換成電壓(0-5 V)。

(詳見 <https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/> )

程式碼 `Serial.println(voltage)`

`Serial.println()` 是將()中的資訊傳輸到序列埠(Serial)中。更精確的講，是將資訊傳到序列埠的轉運站，等待電腦透過序列埠來領取轉運站裡面的資訊。領取的時候有“先進先出(First In First Out, FIFO)”的原則，亦即最早送到轉運站的訊息會先被領

走。所以得注意，萬一轉運站中堆積資訊的速率太快，而領取資訊的速率較慢，則會領到較舊的資訊。

為了避免常常領到舊的資訊，`void loop() {}`中的程式碼最後加上了 `Serial.flush()`；這句指令，來 清空序列埠轉運站。

另外，也加入了 `delay(0.1)`；這句指令，讓程式等待 (delay延遲) 0.1 ms 的時間。用意是讓電腦有稍微足夠的時間來透過序列埠巴士(Serial Bus)領取資訊，免得電腦還沒領完資訊，序列埠就被清空或放入更新的數值。如此，讀取數值到電腦中的流程也會比較穩定。

指令`Serial.println(voltage,2)`；是將voltage的數值取小數下2位傳送到序列埠。如果要改取到小數下4位的數值，則改成 `Serial.println(voltage,4)`；

更多指令與功能說明，可參考 Arduino 官方網頁：

<https://www.arduino.cc/reference/en/#functions>