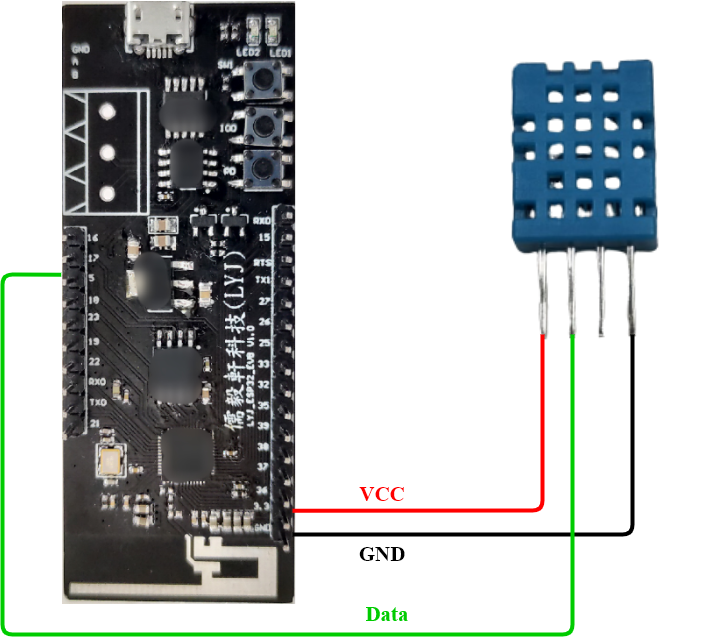
溫溼度感測器DHT11

介紹：

文件中將介紹ESP32連接溫濕度感測器進行溫濕度感測，目前市面常見的溫濕度感測器有 DHT11 或者 DHT22 模組，此類感測器的設計易懂規格也合適拿來小規模操作，故本篇將以此作為示範介紹。

電路接線圖：



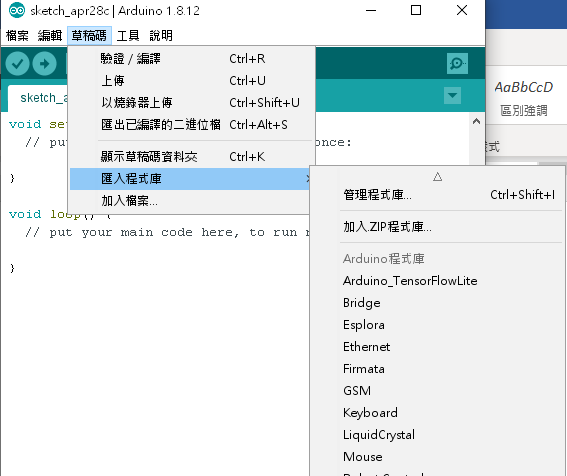
DHT11的data腳位用來傳送感測到的溫濕度資訊，本篇將其角位連接到ESP32的數字5腳位，故後續在程式中只需指定對應角位(5號)即可讀取資訊。

**程式說明：**

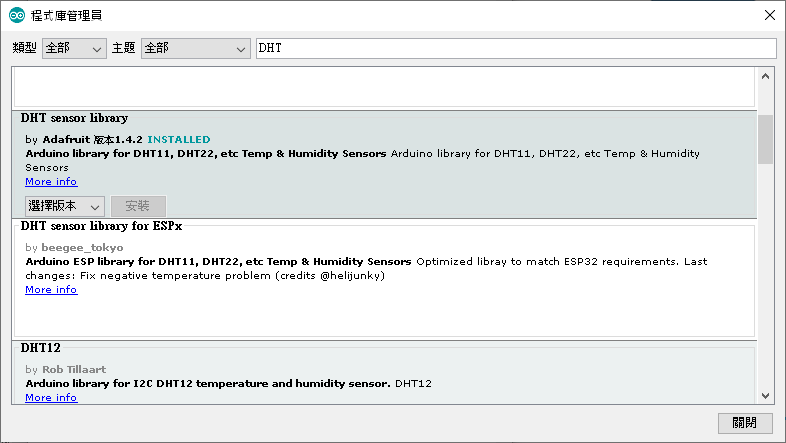
1. 感測器函式庫匯入

Arduino 系統在預設下並沒有支援DHT系列之感測器，導致預設狀態下Arduino無法直接燒入有關DHT系列的程式碼，故此在實作前須先安裝對應函式庫(Library)。此段落會流程性講解Arduino的函式庫安裝方法，只要掌握住此方法後續不管需要何類型函式庫，都能依此類推進行安裝並成功完成。

第一步：  
草稿碼 > 匯入程式庫 > 管理程式庫



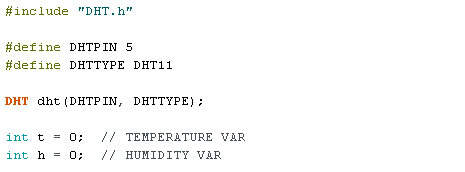
第二步：  
第一步點開管理程式庫會出現程式庫管理員，在搜尋處輸入DHT會出現幾種函式庫提供安裝，本篇採用的感測器使用DHT sensor library進行安裝即可，若使用其他感測器則需找尋對應的函式庫，本篇 DHT sensor library版本選擇第二版進行撰寫。



完成上述步驟便已在 Arduino 中安裝 DHY 函式庫，故此即可開始撰寫程式碼並且進行ESP32燒入。

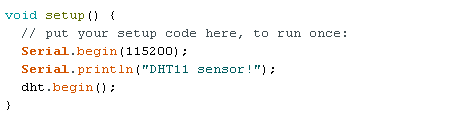
1. 程式碼撰寫

第一區塊



1. Arduino在撰寫時需告知系統需要使用何種函式庫，故此開頭會先將安裝的DHT進行匯入。(#include “DGT.h”)
2. 程式碼撰寫會習慣將此篇程式中不變動的變數(可能是數值或者文字)作定義，這樣的寫法方式能讓程式碼更有可讀性。本篇在接線時採用ESP32的5號腳位進行資料接受，並且腳位不會變動，因此將5進行定義來方便後續使用。DHT感測器也不會變，故一樣採用定義的方式來設定變數。
3. Arduino呼叫DHT函式庫中的dht函數來接收感測器資訊。dht(資料接收腳位,感測器型號)
4. 程式碼撰寫中會將可變動的方式用型態宣告，來告知系統宣告的參數型態、名稱以及其初始值。  
   例如：“float t = 0” 即可理解成浮點數型態(float)的變數，變數名稱(t)的初始值為0，依此類推。

第二區塊

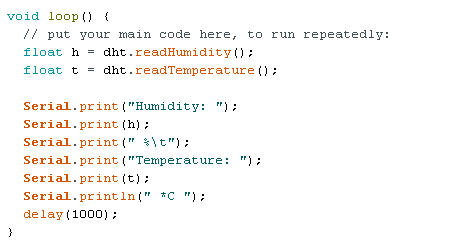


Arduino程式撰寫中會將設定資訊寫在設定函數中(setup())，讓系統可以知道設定的內容。  
1. Serial.begin() 用來設定Arduino的資料傳輸包率，需要在包率相同的情況下才能成功解讀資訊，因此後續在Arduino監視畫面設定相同包率即可看到程式中的資訊。

2. Serial .println(“”) 用來印出文字並且會進行換行，雙引號中的文字及為要列印出的文字。

3. dht.begin() 用來告知程式正式啟動dht感測器接收感測值。

第三區塊



Arduino程式執行後會反覆執行迴圈中的動作(loop())，因此會將程式要運作的指令放在此區域中，以本篇為例就是資料讀取的動作，藉此當ESP32啟動後便會不斷反覆執行讀取與列印資訊的動作。

1. 程式在設定數值的時候，會利用等號(=)來進行參數的變更。本篇將第一區塊中的已經宣告的浮點數參數(t,h)用來接收感測器資訊。例如：“h = dht.readHumidity()”即可解讀成變數(h)設定數值為感測器中的濕度數值(dht.readHumidity())。
2. Serial.print(“”) 一樣用來印出文字但並不會進行換行。
3. Delay(1000) 用來使程式碼停頓多少毫秒。如果沒有停頓則程式碼因為迴圈(loop())的緣故，會不間斷的重複執行，但這樣的速度過快一來沒有意義，二來也不方便觀察，故通常會習慣在迴圈執行結束前進行停頓。

第四區塊



Arduino的監視畫面開啟並設定相同包率(115200)後，即可看到第三區塊中的程式動作。監視畫面中每一行的開頭時間為Arduino提供，故在比對程式碼會發現並沒有此動作，但後續的濕度(Humidity)以及溫度(Temperature)則為ESP32所接收到的DHT11感測值

實際接線圖：

