

第一章 Arduino

因應物聯網浪潮，市面上越來越多開發板出現以作為「物」的角色。本章將介紹最基本的開發板，Arduino Uno，並實作基本元件控制與感測，如溫溼度感測、光度感測以及顯示於 OLED 等。

1.1 什麼是 Arduino

Arduino 是一個硬體和軟體的開源電子平台，它提供了易學易用的整合開發環境(Interactive Development Environment, IDE)，其特色為開發簡單，參考資料多。要開發 Arduino 專題需要準備 Arduino 硬體及 Arduino 軟體。

1.1.1 硬體介紹

Arduino 開發板品項眾多，各有其特色，如適合入門的 Uno/Uno R3、快閃記憶體 (flash memory) 空間大的 Mega/Mega2560 或是能夠連網的 Yún。由於本章所使用到的 Arduino 開發板為 Uno，以下將以 Uno 為範例，做詳細的硬體介紹。



圖 1.1 Arduino Uno

- **規格**

Arduino Uno 是一款基於 ATmega328P 的微控制器板。它有 14 個數位輸入/輸出接腳（其中 6 個可用作 PWM 輸出），6 個類比輸入，16 MHz 石英晶體，USB 連接孔，電源插孔，ICSP 接頭和重置按鈕，基本規格如表 1.1 所示。

表 1.1 Arduino Uno 規格

名稱	規格
微控制器	ATmega328P
工作電壓	5V
建議輸入電壓	7-12V
限制輸入電壓	6-20V

- **電源**

Arduino Uno 板可通過 USB 連接或外部電源供電。外部（非 USB）電源可以

使用電源供應器、9V 電池，通過將 2.1mm 插頭插入電路板的電源插孔供電。注意如果供電電壓低於 7V，則 5V 輸出腳位可能電壓不足，電路板可能會變得不穩定。如果使用 12V 以上，電壓調節器可能會過熱並損壞電路板。

開發板上有個標示為 ON 的 LED，用來指示電源供應狀況，如果已提供電源的情況下，發現標示為 ON 的 LED 閃爍、黯淡或熄滅，應儘快拔除電源，檢查電源供應是否正常，以避免損壞控制板。Arduino Uno 上與電源有關的接腳列於表 1.2。

表 1.2 電源腳位介紹

腳位名稱	介紹
VIN	電路板在使用外部電源時的輸入電壓，可以通過此腳位提供電壓。
5V	從電路板上的穩壓器輸出 5V 穩壓電壓。
3V3	由穩壓器產生的 3.3V 電源。最大電流消耗為 50 mA。
GND	接地。
IOREF	讓其他設備知道 Arduino 控制板的運作電壓。

- **類比輸入腳位**

A0 至 A5 (A 代表 Analog) 可用來接受類比電壓輸入，但不能輸出類比電壓。每隻腳預設會將 0V 到 5V 轉換為 0 至 1023 的數值。對於輸出電壓為其他範圍的電路模組，可以透過 AREF 腳與函式 `analogReference()`，來提供參考電壓。A0 至 A5 也可作為數位輸出、輸入腳位使用，此時 A0 至 A5 分別可視為 D14 至 D19。

- **數位輸出、輸入腳位**

D0 到 D13 (D 代表 Digital) 可輸出高電位 5V 與低電位 0V 的數位訊號，也可接受數位訊號。

- **其他腳位功能**

除了數位 I/O 與類比輸入，Arduino Uno 還有提供其他功能，詳細描述於表 1.3。

表 1.3 其他腳位功能

功能	腳位	介紹
Serial 通訊	D0(RX)、D1 (TX)	用來接收(RX)與傳輸(TX) TTL 訊號的序列資料，連接到 USB 序列埠，因此如果電腦使用 USB 與控制板傳輸數據(控制板上標示為 RX、TX 的 LED 閃爍)，則應避免使用此腳位。
外部中斷	D2、D3	這兩支腳可以利用外部事件觸發中斷。詳細內容請參考 <code>attachInterrupt()</code> 函式。

功能	腳位	介紹
PWM	D3, D5, D6, D9, D10, D 11	可看見這些腳位旁有波浪符號 (~)。透過 <code>analogWrite()</code> 函式用數位訊號來模擬類比訊號輸出，提供 8-bit 的 PWM (Pulse Width Modulation)。
SPI	D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK)	這四支腳搭配 SPI Library 可提供 SPI 序列通訊。
LED	D13	若是原廠控制板，預設會燒錄一個令 D13 定時切換高低電位的 <i>Blink</i> 程式，因此，首次接上電源時，會看到開發板上標示為 L 的 LED 不斷閃爍，這是初步檢視控制板是否功能正常的方式。
I ² C	A4 (SDA), A5 (SCL)	透過 <code>Wire library</code> 可以提供 I ² C 通訊。
AREF	AREF	類比輸入的參考電壓，搭配 <code>analogReference()</code> 函式一起使用。
Reset	Reset	按下重置鈕會使開發板重新執行使用者寫入之程式。當 Reset 腳位 LOW 時，功能如同按下重置鈕。

1.1.2 軟體介紹

• Arduino IDE 下載與安裝

Arduino IDE 放在 Arduino 的官方網站 (<http://arduino.cc/>) [1]，連上該網站後可以在網站上方看到一排選單，包含 HOME/ BUY/ SOFTWARE/ PRODUCTS/ LEARNING/ REFERENCE/ SUPPORT/ BLOG，點選 SOFTWARE->DOWNLOAD 即可進入 Arduino 的下載頁 (<http://arduino.cc/en/Main/Software>)。

如圖 1.2 所示，目前 Arduino 版本為 1.8.10，Arduino 版本會隨時間而更新，目前只要安裝 1.8.x 版本以上即可。Arduino IDE 提供了不同版本讓不同作業系統開發者都可使用，請根據自己的作業系統點選不同選項。



圖 1.2 Arduino IDE 下載進入畫面 I



圖 1.3 Arduino IDE 下載進入畫面 II

點選後畫面如圖 1.3，點選右下角紅色圓圈 JUST DOWNLOAD，下載完後打開檔案進行安裝，進去點選 I AGREE -> NEXT -> INSTALL，跑完後會問三次是否要安裝連接埠驅動，請都按安裝，即安裝完成，就可到桌面看到 Arduino。

• Arduino 開發環境介紹

Arduino IDE 所用的程式語言語法類似於 C/C++，具備文字編輯介面、常用工具欄、圖形化控制介面及一鍵編譯並將程式燒寫入 Arduino 硬體中。而使用 Arduino IDE 編寫的程式被稱為 Sketch，一個典型的 Arduino C/C++ sketch 程式會包含兩個函式：setup()及 loop()，如圖 1.4，它們會在編譯後合成為 main()函式。

- ✓ setup()：在程式執行開始時會執行一次，用於初始化設定。
- ✓ loop()：直到 Arduino 硬體關閉前會重複執行的程式碼。

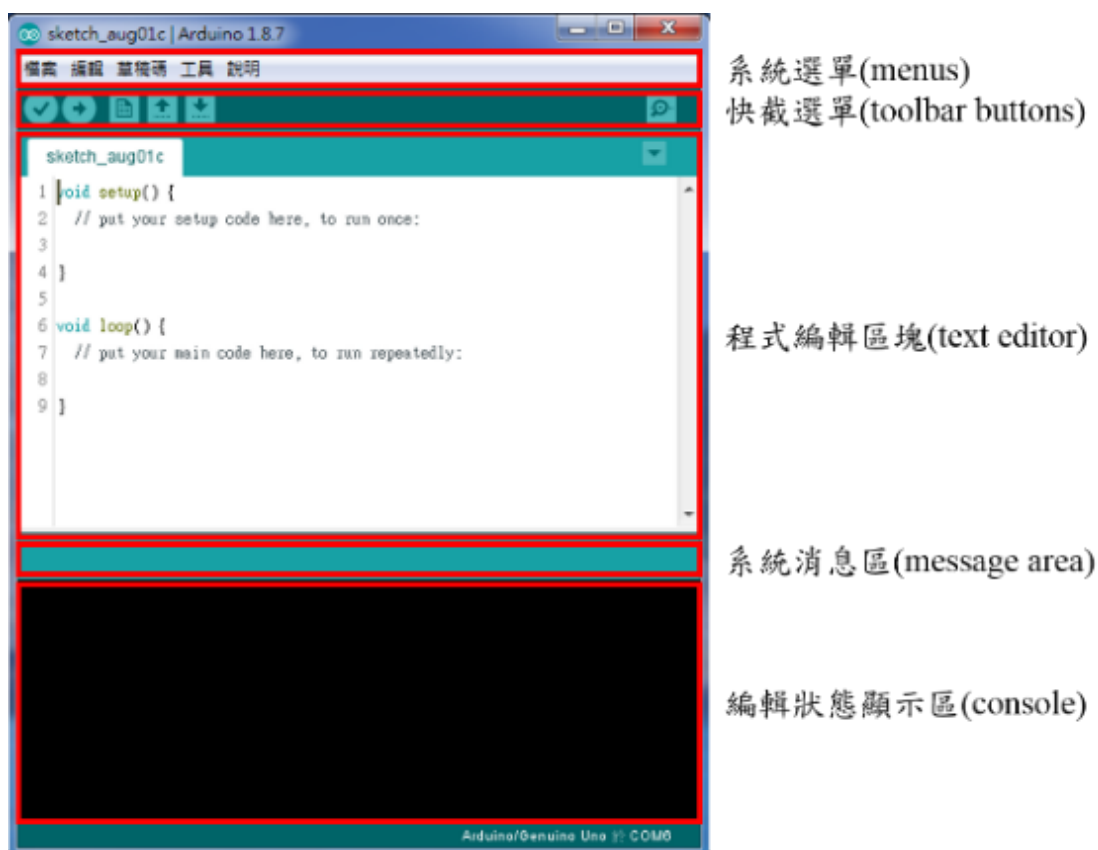


圖 1.4 Arduino 開發環境介面

如圖 1.4，Arduino IDE 畫面由上而下共有五個區域，分別是系統選單(menus)、快捷選單(toolbar buttons)、程式編輯區塊(text editor)、系統消息區(message area) 與編輯狀態顯示區(console)。

Arduino IDE 系統選單中有許多功能，包含 FILE (檔案)、EDIT (檔案)、SKETCH (草稿碼)、TOOL (工具) 及 HELP (說明)。SKETCH->IMPORT LIBRARY (匯入程式單) 裡面包含許多常用的函式庫，開發者只要熟悉函式庫裡的參數設定，即可使用函式庫所提供功能。在 TOOL 下則有硬體開發板及序列埠的設定，至於 FILE 下，

則可以找到不少範例程式，易於讓初學者了解 Arduino IDE 與硬體間的關係，並快速上手。

快捷選單由左而右有五個按鈕，提供開發者快速操作，分別代表 VERIFY/COMPILE (程式編譯及驗證)、UPLOAD I/O BOARD (上傳程式到硬體開發板)、NEW (新增檔案)、OPEN (開啟舊檔)、以及 SAVE (儲存專案)，最右邊還有一個按鈕，是 SERIAL MONITOR (監視串列埠)，開發著可用來監看專案程式運行過程，詳細說明列於表 1.4。而程式編輯區塊供使用者輸入欲執行程式，系統消息區則簡單回報目前程式狀況，完整的執行狀況紀錄於編輯狀態顯示區，更可藉由此介面監視 Arduino I/O 數值。

表 1.4 Arduino IDE 快捷選單

icon	英文	中文	功能
	verify/ compile	驗證	檢查參數設定或引入程式是否產生錯誤。
	upload	上傳	程式進行編譯，將程式碼透過 USB 介面燒錄至 Arduino 控制板。
	new	新增	產生新的 Sketch
	open	開啟	開啟腳本，顯示在同一頁面上。
	save	儲存	Sketch 儲存。
	serial monitor	序列埠監 控視窗	開啟監視器頁面，監視 Arduino I/O 介面。

1.2 Arduino Uno 第一支程式 - Blink

本節目的為熟悉 Arduino IDE 的操作，以範例程式 *blink* 示範如何操控 Arduino Uno 板上的 LED 燈閃爍，大多數 Arduinos 都有一個可以控制的板載 LED。在 UNO、MEGA 和 ZERO 上連接到 D13 腳位，MKR1000 則是 D6 腳位。Arduino 內部的常數 (constants) LED_BUILTIN 為正確的 LED 引腳，與使用的板無關。本節會詳細的說明程式的開啟、驗證與上傳，操作完本實驗將學會如何利用 Arduino IDE 寫入程式至 Arduino Uno，並且對於 Arduino IDE 各項功能更為了解。

1.2.1 硬體設備





✓ USB 傳輸線

1.2.2 硬體安裝

將 Arduino Uno 開發板透過 USB 傳輸線連接個人電腦，Type B USB 連接 Arduino Uno 開發板，而另一頭是 Type A USB 連接個人電腦，這可以提供 5V、500mA 的電源給控制板，電腦也可以透過 USB 傳送程式給 Arduino。

1.2.3 範例程式 - *blink*

Step 1 在個人電腦中打開 Arduino IDE。

Step 2 選擇開發板。在 Arduino IDE 點選工具 (TOOL) -> 開發板 (BOARD) -> ARDUINO/GENUINO UNO，如圖 1.5 所示。

Step 3 選擇 Port。在 Arduino IDE 點選工具 (TOOL) -> 序列埠 (PORT) -> COMX(ARDUINO/ GENUINO UNO)，請務必確認此步驟選擇後是否有打勾，如圖 1.6。

Step 4 開啟檔案。在 Arduino 系統選單上，點選檔案 (FILE) -> 範例 (EXAMPLES) -> BASICS -> BLINK (延伸檔名*.INO)，即開啟範例程式 *blink*，如圖 1.7。

Step 5 驗證/編譯。程式碼出現後，點選左上驗證 (VERIFY)，查驗程式碼是否正確。若無錯誤，下方會顯示編輯完成，如圖 1.8 所示。

Step 6 寫入。編譯無誤即可將程式編譯碼燒錄至 Uno 開發板，點選左上上傳 (UPLOAD)，如圖 1.9 所示，CONSOLE 寫上傳完畢，表示上傳成功。

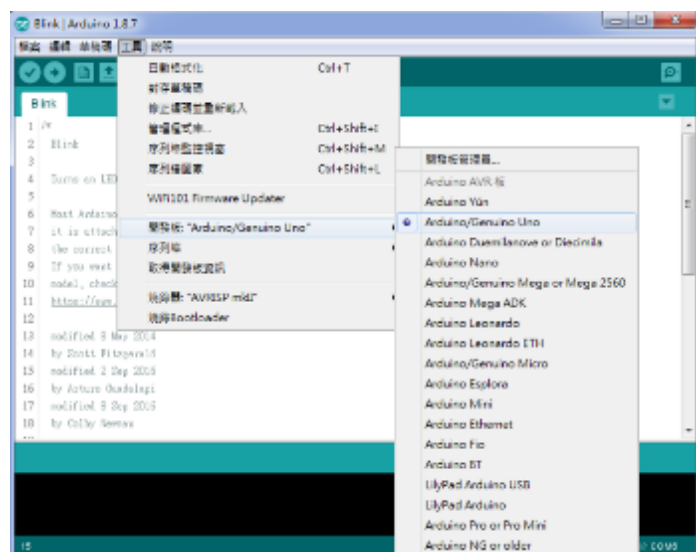


圖 1.5 選擇開發板

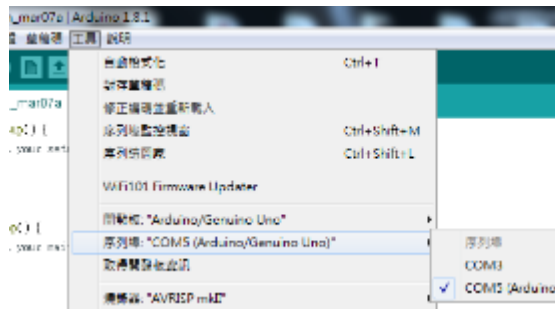


圖 1.6 選擇 Port

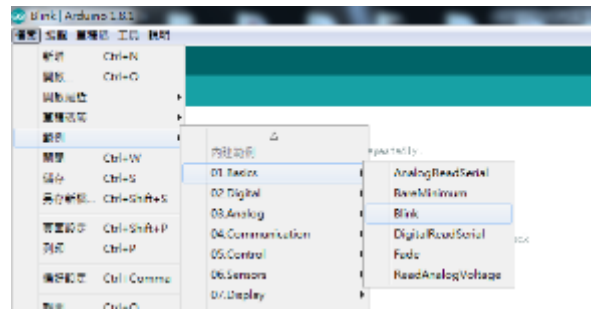


圖 1.7 開啟 Blink 範例



圖 1.8 Blink 編輯確認

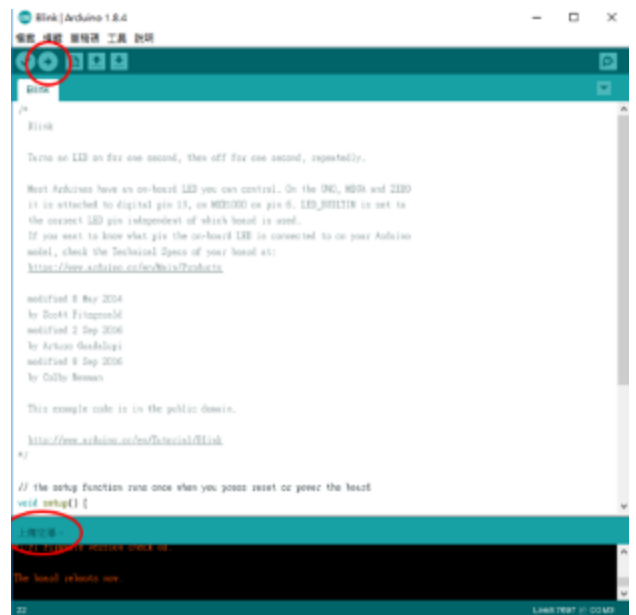


圖 1.9 Blink 上傳完成

Step 5 及 Step 6 若失敗請先檢查圖 1.5 及圖 1.6 中確認板子型號與通訊埠選擇是否無誤，若無誤則檢查 USB 傳輸線是否異常，以上檢查都無誤的話，請換一塊板子再試一次。

標示為L的LED



圖 1.10 標示為 L 的 LED




完成以上步驟後，請觀察 Arduino Uno 開發板上標示為 L 的 LED，如圖 1.10，此時 LED 燈會依照程式規定的秒數閃爍，依此範例開發板上 LED 燈會亮一秒，熄滅一秒。若能夠正常執行程式，則代表 Uno 可以與電腦正常連線，並可控制板上 LED。讀者可以想想看，如何變更 LED 閃爍之快慢（例如：`delay(100);`）。







1.3 於 OLED 上顯示溫溼度及光度感測值

Arduino 很重要的功能就是通訊傳輸與 I/O 控制，市面上已有許多感測器。本專題利用 Uno 接上溫溼度及光度感測元件，並且可以將感測到的數值顯示於 OLED 螢幕上。

本節將一步步帶領同學，瞭解各個感測器的接法以及程式後，將三個感測器整合起來，完成第一個自己寫的程式。

1.3.1 硬體設備

- ✓  Arduino Uno
- ✓  Grove – 溫溼度模組 DHT22 AM2302
- ✓  Grove – Light Sensor (P) v1.1 光感測器

- ✓  Grove - OLED Display 0.96\" 顯示模組 I2C 介面 128x64
- ✓  USB 傳輸線 Type A to Type B
- ✓  個人電腦
- ✓  麵包板
- ✓  2 條杜邦線公母
- ✓  4x 公端杜邦線+母座至 Grove 4 針連接線

1.3.2 光度模組硬體安裝

- Step 1** 為了讓多個感測器共享 VCC 與 Gnd，請將 Uno 上 5V 與 Gnd 以杜邦線拉至麵包板上，如圖 1.11。
- Step 2** 將 Grove 4 針端接上光度模組那一端，如圖 1.13，為了讓初學者上手物聯網的環境，感測器都設有防呆裝置，請不要硬插上去，插不進去就改方向。
- Step 3** 看到光度模組上，有四個腳位，分別是 GND、VCC、NC (Not Connected)與 SIG，如圖 1.13。請將 GND、VCC 接至麵包上剛剛拉出來的 5V 與 Gnd，而 SIG 為光度模組的訊號線，請接至 Uno 板上 A1 腳位，如圖 1.14。

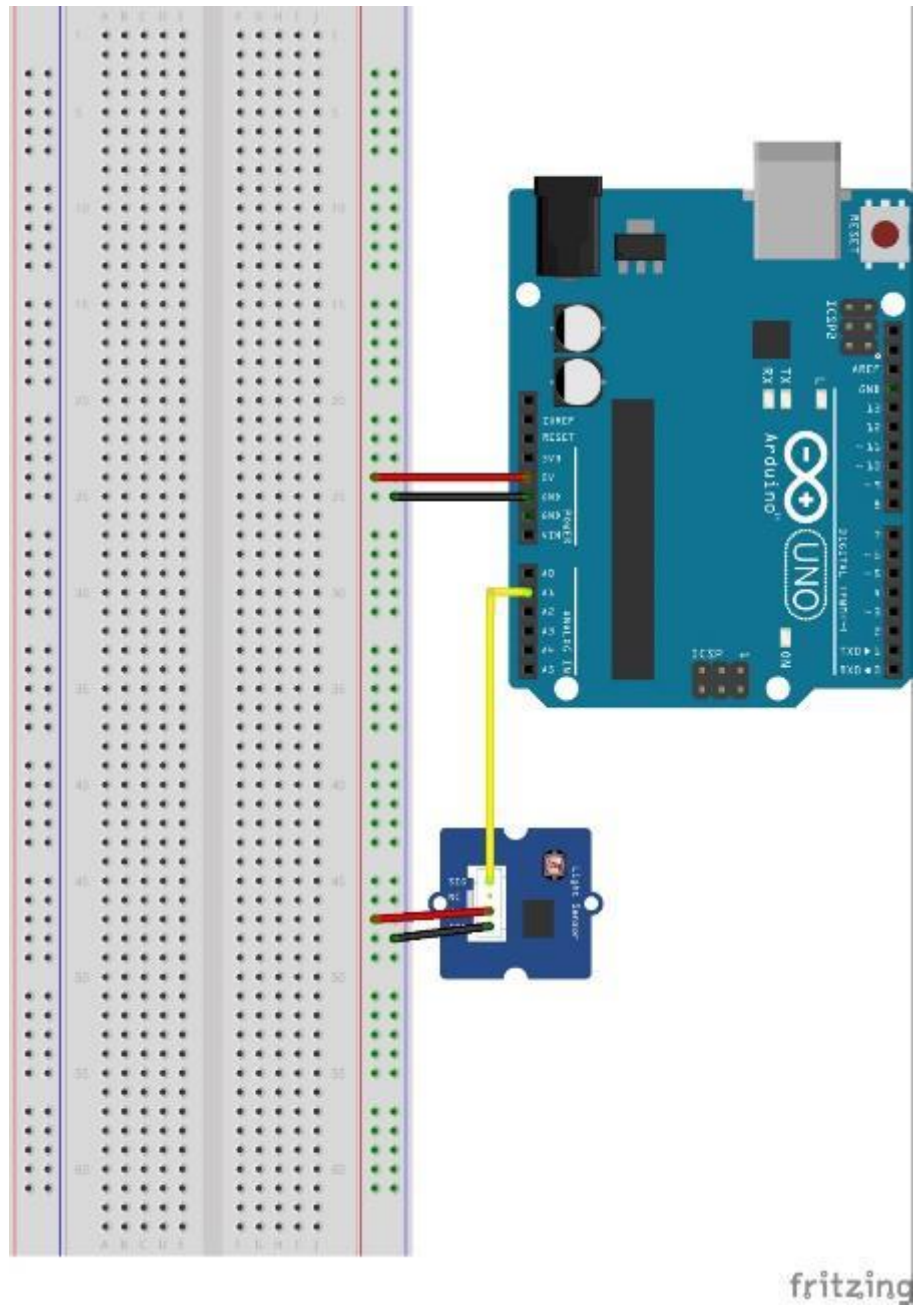


圖 1.11 光感接腳示意圖

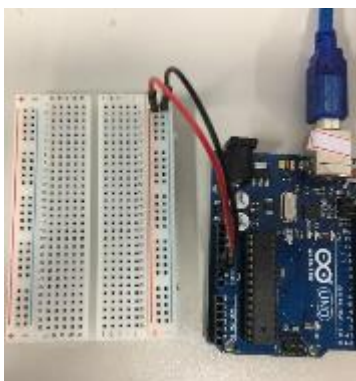


圖 1.12 共享 VCC 與地



圖 1.13 光感測器腳位

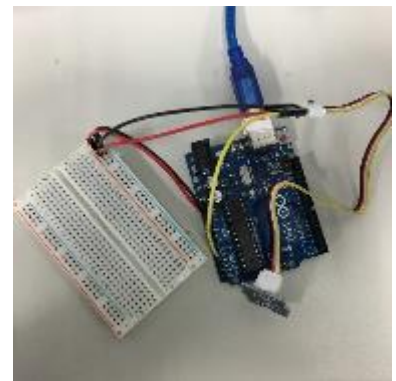


圖 1.14 光感連至 Uno

1.3.3 光度感測器程式部分

Step 1 新增 Sketch。在 Arduino IDE 點選快捷選單的新增，或是點 FILE -> NEW。

Step 2 黏貼程式。打開下列連結，將此程式碼複製貼到剛新開啟的 Sketch。

https://github.com/ChristyLin/NTU-IoT-Semester-1/blob/master/Ch1/Ex1_1/Ex1_1.ino

Step 3 驗證/編譯與上傳至 Uno。

Step 4 監控序列埠。上傳完畢後，點選 Serial Monitor。

Step 5 確認 baud rate。如圖 1.15 所示，檢查 baud rate 與程式碼是否一致，相同即可看到感測器所接收到的值。

以 Uno 讀取光度感測器的部份非常簡單，就如同下列程式，短短 12 行就可以完成。程式的第一行，由於剛剛我們將光度感測器的 SIG 連接至 Uno A1，因此我們需先定義 Light Sensor 腳位為 A1。第三到五行定義 Serial 的 baud rate 在 9600，讓我們可以透過序列埠監控視窗看到 Light Sensor 的值。第七行開始是一個 loop 迴圈，不斷的執行 8 到 11 行的程式。第八行定義 float 變數 l 為腳位 A1 所讀到的 Light Sensor 之值，這裡利用 analogRead 讀取。9 至 11 行則將讀到的值顯示於序列埠監控視窗，讓使用者可以得知感測器之值。

```
1 #define LiIN A1      // Light sensor connected to A1
2
3 void setup() {
4   Serial.begin(9600);
5 }
6
7 void loop() {
8   float l = analogRead(LiIN);
9   Serial.print("Light value: ");
10  Serial.println(l);
11  Serial.println("-----");
12 }
```

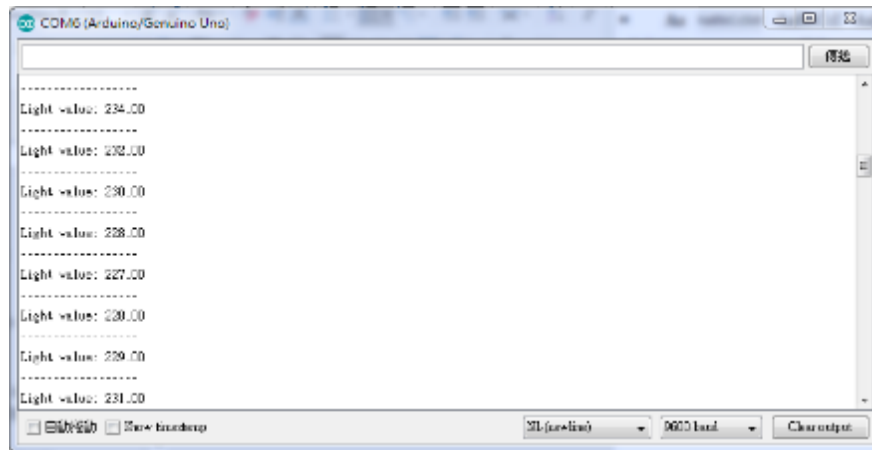


圖 1.15 Light Sensor 值

1.3.4 溫濕度模組硬體安裝

Step 1 將 Grove 4 針端接上溫濕度度模組那一端。

Step 2 看到溫濕度模組上，有四個腳位，分別是 GND、VCC、NC (Not Connected)與 SIG。請將 GND、VCC 接至麵包板上剛剛拉出來的 5V 與 Gnd，而 SIG 為光度模組的訊號線，請接至 Uno 板上 A0 腳位，如圖 1.16。

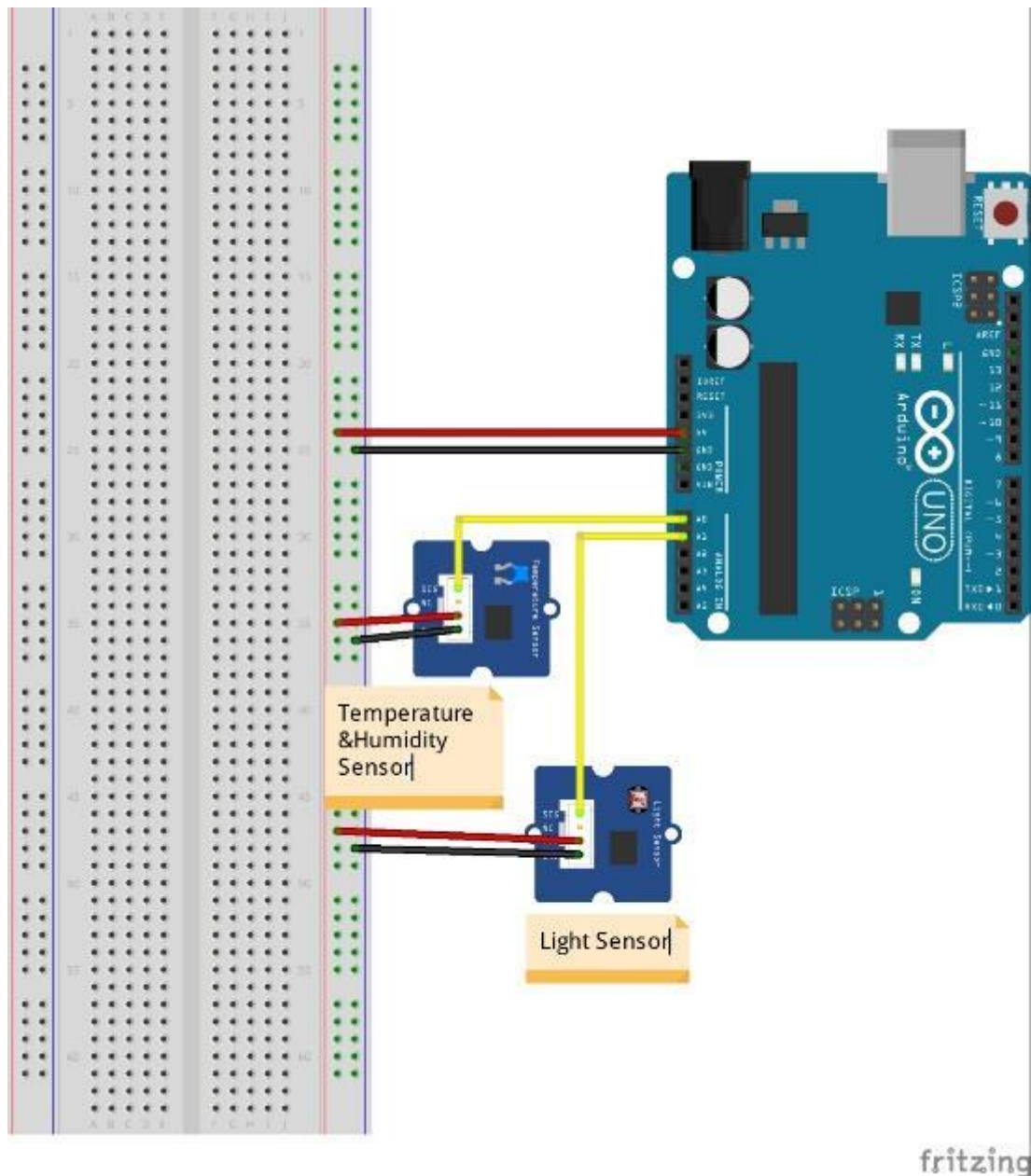


圖 1.16 溫濕度感測器接腳示意圖

1.3.5 溫濕度感測器程式部分

關於溫濕度感測器，我們會使用 DHT-22 函式庫，若同學有興趣，其下載網址為：https://github.com/Seeed-Studio/Grove_Temperature_And_Humidity_Sensor。

Step 1 下載函式庫。點選 ARDUINO IDE -> SKETCH -> IMPORT LIBRARY -> MANAGE LIBRARY。

Step 2 在搜尋欄內打上 GROVE TEMPERATURE AND HUMIDITY SENSOR，並且在 GROVE TEMPERATURE AND HUMIDITY SENSOR BY SEEED STUDIO 那欄點選 MORE INFO -> INSTALL，如圖 1.17。

Step 3 開啟範例程式碼。ARDUINO IDE -> FILE EXAMPLE -> GROVE TEMPERATURE AND HUMIDITY SENSOR ->DHT_TESTER，圖 1.18。

Step 4 驗證/編譯與上傳至 Uno。

Step 5 監控序列埠。上傳完畢後，點選 Serial Monitor。

Step 6 確認 baud rate。如圖 1.19 所示，檢查 baud rate 與程式碼是否一致，相同即可看到感測器所接收到的值。

以 Uno 讀取溫濕度感測器的部份利用 Seed Studio 提供的範例，就如同下列程式。在程式的第四行，匯入函式庫以供我們可以簡單讀取溫濕度感測器。程式的第六行，由於感測器的 SIG 連接至 Uno A0，因此定義溫濕度感測器腳位為 A0，而第十行定義了我們所使用的溫濕度感測器為 DHT22(AM2302)，並且在第十八行將這些資訊統一給溫濕度感測器 dht。

在 setup 中，第 23 行定義 Serial 的 baud rate 在 9600，並在第 25 行啟動溫濕度感測器。第 28 行開始是一個 loop 迴圈，不斷的執行 30 到 47 行的程式。第 32 與 33 行定義 float 變數 h 與 t 為 dht 中濕度與溫度之值。第 36 至 39 行確定溫度與濕度都是正確的值後，由 42 至 48 行則將讀到的值顯示於序列埠監控視窗，讓使用者可以得知感測器之值。

```
1 // Example testing sketch for various DHT humidity/temperature sensors
2 // Written by ladyada, public domain
3
4 #include "DHT.h"
5
6 #define DHTPIN A0    // what pin we're connected to
7
8 // Uncomment whatever type you're using!
9 //#define DHTTYPE DHT11 // DHT 11
10 #define DHTTYPE DHT22 // DHT 22 (AM2302)
11 //#define DHTTYPE DHT21 // DHT 21 (AM2301)
12
13 // Connect pin 1 (on the left) of the sensor to +5V
14 // Connect pin 2 of the sensor to whatever your DHTPIN is
15 // Connect pin 4 (on the right) of the sensor to GROUND
16 // Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor
17
18 DHT dht(DHTPIN, DHTTYPE);
19
20 void setup()
21 {
22     Serial.begin(9600);
23     Serial.println("DHTxx test!");
24
25     dht.begin();
26 }
27
28 void loop()
29 {
```

```

30 // Reading temperature or humidity takes about 250 milliseconds!
31 // Sensor readings may also be up to 2 seconds 'old' (its a very slow
sensor)
32 float h = dht.readHumidity();
33 float t = dht.readTemperature();
34
35 // check if returns are valid, if they are NaN (not a number)
36 if (isnan(t) || isnan(h))
37 {
38     Serial.println("Failed to read from DHT");
39 }
40 else
41 {
42     Serial.print("Humidity: ");
43     Serial.print(h);
44     Serial.print(" %\t");
45     Serial.print("Temperature: ");
46     Serial.print(t);
47     Serial.println(" *C");
48 }
49 }

```

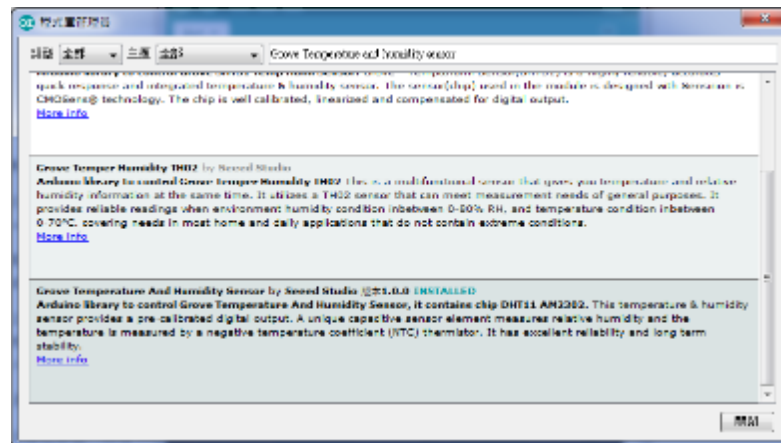


圖 1.17 Grove Temperature and Humidity Sensor Library



圖 1.18 DHT tester

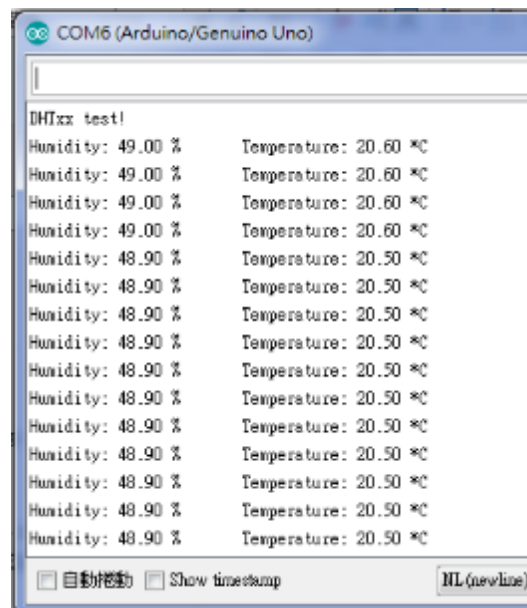


圖 1.19 顯示溫濕度之值

1.3.6 OLED 硬體安裝

Step 1 將 Grove 4 針端接上 OLED 端。

Step 2 看到 OLED 的背面四個腳位，分別是 GND、5V、SDA 與 SCL。請將 GND、VCC 接至麵包板上剛剛拉出來的 5V 與 Gnd，而 SDA 請接至 Uno 板上 A4，SCL 接至 A5，分別為 Uno 的兩個 I2C 腳位。

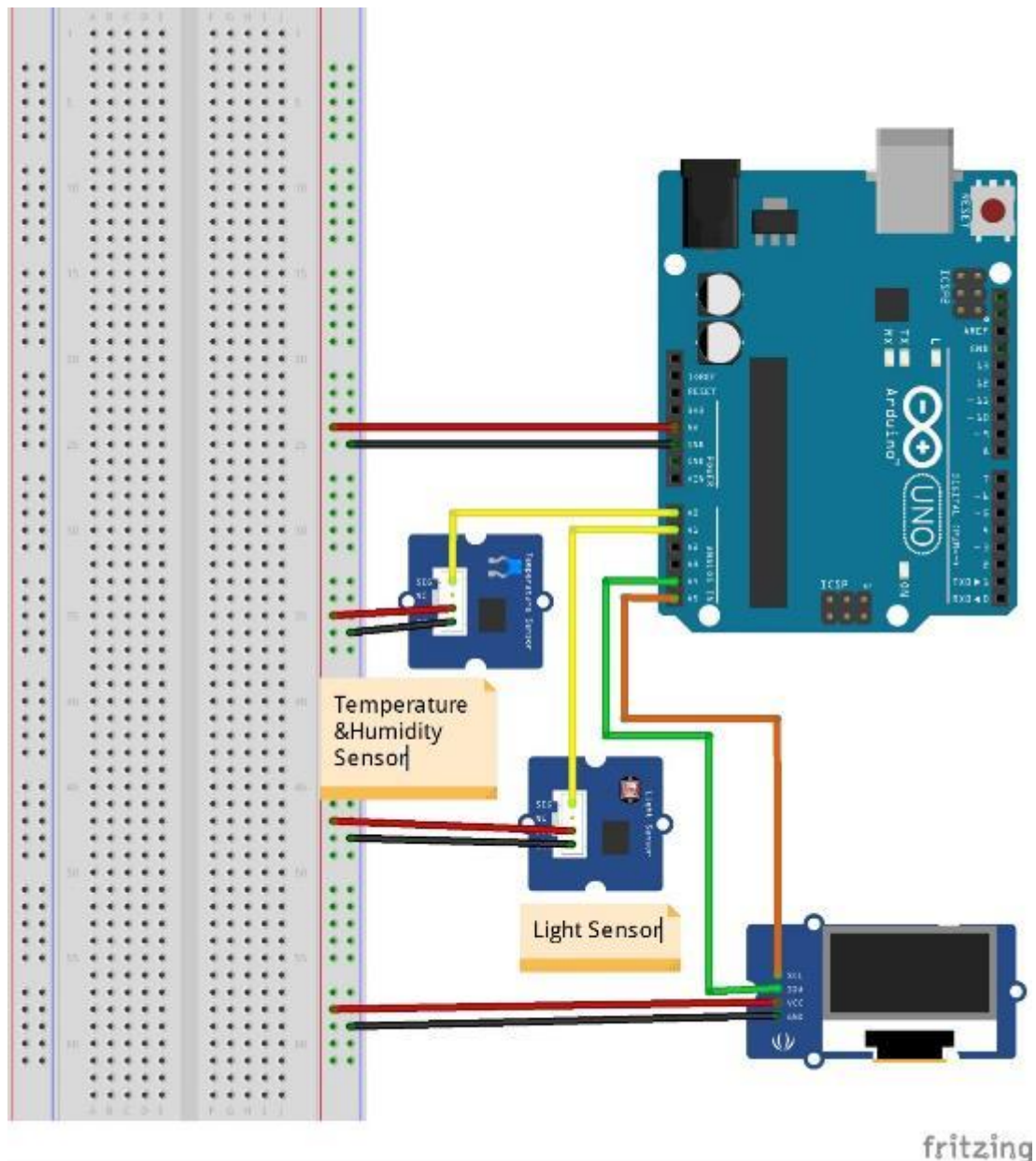


圖 1.20 感測器接腳圖

1.3.7 OLED 顯示 Hello World

本節會使用到兩個函式庫，分別為 `Wire` 與 `SeeedOLED`。`Wire` 是 Arduino 提供的函式庫，用於與 I2C 裝置溝通。而 `SeeedOLED` 則是 Seeed Studio 提供給 OLED 使用的函式庫，若同學有興趣，其參考網址為

`Wire`：<https://www.arduino.cc/en/reference/wire>。

`SeeedOLED`：https://github.com/Seeed-Studio/OLED_Display_128X64。

Step 1 下載函式庫。點選 ARDUINO IDE -> SKETCH -> IMPORT LIBRARY -> MANAGE LIBRARY。

Step 2 在搜尋欄內打上 GROVE OLED，並且在 GROVE - OLED DISPLAY 0.96

BY SEEED STUDIO 那欄點選 MORE INFO -> INSTALL，圖 1.21。

Step 3 開啟範例程式碼。ARDUINO IDE -> FILE EXAMPLE -> GROVE - OLED DISPLAY 0.96 -> OLED_HELLO_WORLD。

Step 4 驗證/編譯與上傳至 Uno。

Step 5 檢查 OLED。上傳完畢後，查看 OLED 上是否顯示 Hello World!，如圖 1.22。

在本節程式中，我們先做簡單的 OLED 顯示 Hello World!。在程式的第一行與第二行，匯入了所需的函式庫 Wire 與 SeeedOLED。第五行的 setup 中，主要是對 OLED 進行基本設定，如第八行的 SeeedOled.init()是做初始化設定，第十行 SeeedOled.clearDisplay()是清除整個屏幕，應在開始新的啟動之前或滾動停用之後使用，另外還將游標設置為左上角。第十一行 setNormalDisplay()將顯示配置為正常模式（非反向）模式。第十二行是將游標定位功能設為 PageMode()，並且利用第十三行 setTextXY(0,0)設置由標停在第 X 行 Page 的第 Y 個字。最後再以第十四行 putString()在 OLED 上印出 Hello World!的字樣，若要印的是數字，則可以利用 SeeedOled.putNumber()。

```
1 #include <Wire.h>
2 #include <SeeedOLED.h>
3
4
5 void setup()
6 {
7     Wire.begin();
8     SeeedOled.init(); //initialize SEEED OLED display
9
10    SeeedOled.clearDisplay();
11    SeeedOled.setNormalDisplay();
12    SeeedOled.setPageMode();           //Set addressing mode to Page Mode
13    SeeedOled.setTextXY(0,0);          //Set the cursor to Xth Page, Yth Column
14    SeeedOled.putString("Hello World!"); //Print the String
15
16 }
17
18 void loop()
19 {
20
21 }
```

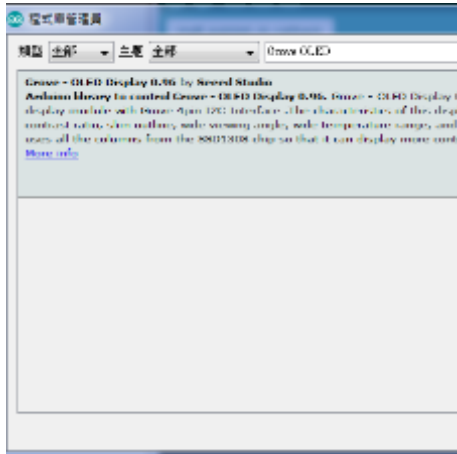


圖 1.21 Grove - OLED Display 0.96

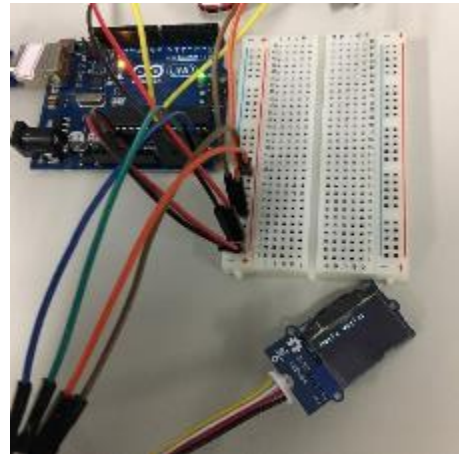


圖 1.22 Hello World!

1.3.8 以 OLED 顯示溫溼度及光度值

瞭解每一個感測器的程式內容後，讓我們合併上述程式，將溫濕度以及光度之值顯示於 OLED 上，寫出第一個自己完成的 Arduino 程式吧！

Step 1 新增 Sketch。在 Arduino IDE 點選快捷選單的新增 ，或是點 FILE -> NEW。

Step 2 匯入所需函式庫。由上可知，要完成此函式我們需要三個函式庫，分別為溫濕度感測器所需的 DHT 以及 OLED 所需的 Wire 以及 SeeedOLED。

```
#include <DHT.h>           //teperature&humidity
#include <Wire.h>           //i2c
#include <SeeedOLED.h>      //OLED
```

Step 3 定義腳位。根據上述接腳，我們將 A0 分配給溫濕度感測器、A1 分配給光度感測器，A4 與 A5 則是給 OLED 使用 I2C。而 I2C 在 Wire 函式庫中已經定義，因此我們只需定義好溫濕度以及光感的腳位。

```
#define DHTPIN A0          // A0 for temperature & humidity
#define LiIN A1            // A1 for light sensor
```

Step 4 定義溫濕度感測器參數。要使用 DHT 函式庫，需要將腳位以及溫氏度感測器的類型定義好，腳位在上一步定義好了，因此我們接著定義使用的溫濕度感測器類型。並將參數都給 DHT 函式庫執行。

```
#define DHTTYPE DHT22      // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE);
```

Step 5 setup function。接著我們將所有一次性程式寫入 setup function，根據前面三個程式，我們會需要 serial、wire、dht 與 SeeedOled 的設置。另外，由於要在 OLED 上顯示溫濕度，我們可以先在設置中打印 [Humid: %] 等字樣，如下範例。

```

void setup() {
  Serial.begin(9600);
  Wire.begin();
  dht.begin();
  SseeedOled.init(); //initialize SEEED OLED display

  SseeedOled.clearDisplay();
  SseeedOled.setNormalDisplay();
  SseeedOled.setPageMode();           //Set addressing mode to Page Mode

  SseeedOled.setTextXY(0,0);
  SseeedOled.putString("Humid:");
  SseeedOled.setTextXY(0,12);
  SseeedOled.putString("%");

  SseeedOled.setTextXY(1,0);
  SseeedOled.putString("Tempe:");
  SseeedOled.setTextXY(1,12);
  SseeedOled.putString("*C");

  SseeedOled.setTextXY(2,0);
  SseeedOled.putString("Light:");

  Serial.println("DHTxx test!");
}

```

Step 6 loop function。接著，我們將需要重複的程式放入 loop function 中，如讀取溫濕度感測器與光感測器之值，以及將這些值顯示於序列埠與 OLED 上，如下列程式。

```

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float l = analogRead(LiIN);

  if (isnan(t) || isnan(h))
  {
    Serial.println("Failed to read from DHT");
    Serial.print("Light value: ");
    Serial.println(l);
  }
  else
  {
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.println(" %\t");
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.println(" *C");
    Serial.print("Light value: ");
    Serial.println(l);
    Serial.println("-----");

    SseeedOled.setTextXY(0,8);
    SseeedOled.putNumber(h);
    SseeedOled.putString(" ");
  }
}

```

```

Seed0led.setTextXY(1,8);
Seed0led.putNumber(t);
Seed0led.putString(" ");

Seed0led.setTextXY(2,7);
Seed0led.putNumber(1);
Seed0led.putString(" ");
}
delay(500);
}

```

Step 7 完整程式。打開下列連結，將此程式碼複製貼到剛新開啟的 Sketch。

https://github.com/ChristyLin/NTU-IoT-Semester-1/blob/master/Ch1/Ex1_2/Ex1_2.ino

Step 8 編輯及上傳程式碼到 Uno。

完成以上步驟後，可以在 OLED 上顯示溫溼度及光度感測器所收到的值，顯示結果如圖 1.24 所示。

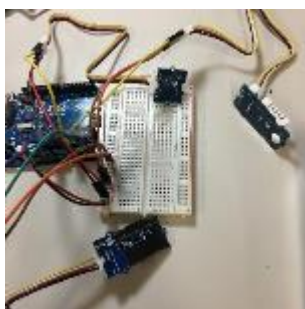


圖 1.23 接線完成圖



圖 1.24 OLED 顯示結果

1.3.9 程式碼解析

此程式中，有幾個重要指令，列於表 1.5。可以發現，利用 Arduino 寫程式非常簡單，由於 Arduino 非常多開源範例，因此有很多可以參考的程式，藉由參考網路上各種程式，搭配基本語法，讓程式的撰寫快速又方便。

表 1.5 程式碼解析

程式碼	解析
Serial.begin(9600)	設定硬體開發板與電腦程式之通訊溝通 baud rate=9600 bps
Serial.println("DHTxx test!")	將括弧內的變數值輸出至 Arduino 程式開發的序列埠監控視窗中
pinMode(pin, mode)	設定第 pin 接腳之 Mode 為輸出 (OUTPUT)或輸入(INPUT)功能
delay(1000)	延遲 1000 ms

<code>digitalWrite(pin, value)</code>	將 pin 接腳狀態寫成指定的電位狀態是 HIGH 還是 LOW
<code>digitalRead(pin)</code>	讀取 pin 接腳的狀態







1.4 結語

在本章中，介紹 Uno 的硬體裝置以及軟體開發環境，同時帶領讀者熟悉如何利用 Arduino IDE 進行開發以及一些程式解析。更重要的是，利用感測器讀取環境中的溫溼度以及光度，並顯示結果於 OLED。

請讀者想想看，本節所使用的光度模組為類比輸出，因此接在 Uno 的類比輸入。而在 Grove Starter Kit 中，還有許多其他模組，查查看它們輸出或輸入的訊號類別，以及它們的功用，並完成表 1.6。

若同學對其他 Sensor 有興趣的話，可以在 <https://github.com/Seeed-Studio> 找到許多範例程式，本章所用到的範例也都在裡面。

表 1.6 Grove Starter Kit

	名稱	訊號類別	功用
	Grove - Light Sensor	Analog Output	detect the intensity of light
	Grove - Servo		
	Grove - Ultrasonic Ranger		
	Grove - Relay		
	Grove - IMU 9DOF		
	Grove - LED Bar		

參考資料

- [1] “Arduino Uno Rev3,” [線上]. Available: <https://store.arduino.cc/arduino-uno-rev3>.

