Topic 01 Introduction to Arduino

Arduino (https://zh.wikipedia.org/wiki/Arduino)

它使用 Atmel AVR 單片機，採用開放原始碼的軟硬體平台，構建於開放原始碼 simple I/O 介面板，並具有使用類似 Java，C 語言的 Processing/Wiring 開發環境。

Arduino 的核心開發團隊成員包括：馬西莫·班齊（Massimo Banzi）、大衛·奎提耶斯（David Cuartielles）、湯姆·伊果（Tom Igor）、贊布羅塔·馬提諾（Gianluca Martino）、大衛·梅利斯（David Mellis）和尼可拉斯·蘭比提（Nicholas Zambetti）。

據說馬西莫·班齊之前是義大利 Ivrea 一家高科技設計學校的老師。他的學生們經常抱怨找不到便宜好用的微控制器。2005 年冬天，馬西莫·班齊跟大衛·奎提耶斯討論了這個問題。大衛·奎提耶斯是一個西班牙籍晶片工程師，當時在這所學校做存取學者。兩人決定設計自己的電路板，並引入了馬西莫·班齊的學生大衛·梅利斯為電路板設計編程語言。兩天以後，大衛·梅利斯就寫出了程式碼。又過了三天，電路板就完工了。這塊電路板被命名為 Arduino。幾乎任何人，即使不懂電腦編程，也能用 Arduino 做出很酷的東西，比如對感測器作出回應，閃爍燈光，還能控制馬達。隨後馬西莫·班齊、大衛·奎提耶斯和大衛·梅利斯把設計圖放到了網上。保持設計的開放原始碼理念，因為版權法可以監管開源軟體，卻很難用在硬體上，他們決定採用創用 CC 許可。[1]創用 CC 是為保護開放版權行為而出現的類似 GPL 的一種許可（license）。在創用 CC 許可下，任何人都被允許生產印刷電路板的複製品，還能重新設計，甚至銷售原設計的複製品。你不需要付版稅，甚至不用取得 Arduino 團隊的許可。然而，如果你重新發布了參照設計，你必須說明原始 Arduino 團隊的貢獻。如果你調整或改動了電路板，你的最新設計必須使用相同或類似的創用 CC 許可，以保證新版本的 Arduino 電路板也會一樣的自由和開放。唯一被保留的只有 Arduino 這個名字。它被註冊成了商標。如果有人想用這個名字賣電路板，那他們可能必須付一點商標費用給 Arduino 的核心開發團隊成員。

特色

- 基於創用 CC 開放原始碼的電路圖設計。
- 免費下載，也可依需求自己修改，但需遵照姓名標示。您必須按照作者或授權人所指定的方式，表彰其姓名。
- 依相同方式分享，若您改變或轉變著作，當散布該衍生著作時，您需採用與本著作相同或類似的授權條款。
- Arduino 可使用 ICSP 線上燒入器，將 Bootloader 燒入新的 IC 晶片。
- 可依據 Arduino 官方網站，取得硬體的設計檔，加以調整電路板及元件，以符合自己實際設計的需求。
- 可簡單地與感測器，各式各樣的電子元件連接，如紅外線、超音波、熱敏電阻、光敏電阻、伺服馬達…等。
- 支援多樣的互動程式，如 Adobe Flash, Max/MSP, VVVV, Pure Data, C, Processing…等。
- 使用低價格的微處理控制器（Atmel AVR）（ATMEGA 8,168,328 等）。
- USB 介面，不需外接電源。另外有提供直流（DC）電源輸入。


Open Hardware
Facilitator
seeed
CAUTION
5.0
Bluetooth
A FLEXIBLE
ADD-ON FOR
YOUR ARDUINO
BASE
SHIELD

UNO
ARDUINO
DIGITAL - PWM~
POWER
ANALOG IN

圖一、相關實驗設備

Arduino 開發語言

Ref. https://www.arduino.cc/reference/en/

語言:C/C++

提供

1. Variables

Constants		Data Types		Variable Scope & Qualifiers
Floating Point Constants		String()	long	const
Integer Constants		array	short	scope
HIGH | LOW		boolean	string	static
INPUT | OUTPUT | INPUT_PULLUP		byte	unsigned char	volatile
LED_BUILTIN		char	unsigned int	
true | false		double	unsigned long	
Conversion		float	void	**Utilities**
byte()	int()	int	word	PROGMEM
char()	long()			sizeof()
float()	word()			

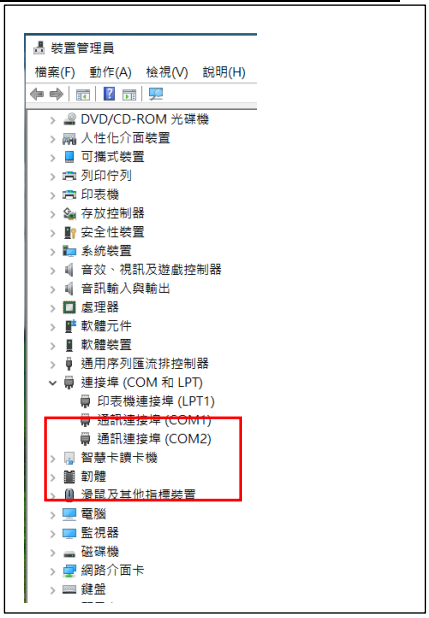
2. Structure (The elements of Arduino (C++) code)

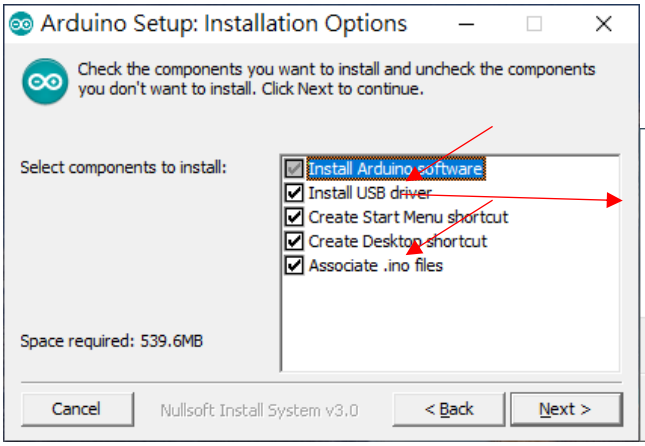
Sketch		Pointer Access Operators	Boolean Operators
loop()		& (reference operator)	! (logical not)
setup()		* (dereference operator)	&& (logical and)
			|| (logical or)
Arithmetic Operators		**Comparison Operators**	**Bitwise Operators**
% (modulo)		!= (not equal to)	& (bitwise and)
* (multiplication)		< (less than)	<< (bitshift left)
+ (addition)		<= (less than or equal to)	>> (bitshift right)
- (subtraction)		== (equal to)	^ (bitwise xor)
/ (division)		> (greater than)	| (bitwise or)
= (assignment operator)		>= (greater than or equal to)	~ (bitwise not)
Control Structure		**Further Syntax**	Compound Operators
break	goto	#define (define)	&= (compound bitwise and)
continue	if...else	#include (include)	*= (compound multiplication)
do...while	return	/* */ (block comment)	++ (increment)
else	switch...case	// (single line comment)	+= (compound addition)
for	while	; (semicolon)	-- (decrement)
		{ } (curly braces)	-= (compound subtraction)
			/= (compound division)
			|= (compound bitwise or)

3. Functions:

Digital I/O	**Analog I/O**	**Zero, Due & MKR Family**	Trigonometry
digitalRead() digitalWrite() pinMode()	analogRead() analogReference() analogWrite()	analogReadResolution() analogWriteResolution()	cos() sin() tan()
Advanced I/O	**Bits and Bytes**	**Characters**	**Math**
noTone() pulseIn() pulseInLong() shiftIn() shiftOut() tone()	bit() bitClear() bitRead() bitSet() bitWrite() highByte() lowByte()	isAlpha() isAlphaNumeric() isAscii() isControl() isDigit() isGraph() isHexadecimalDigit() isLowerCase() isPrintable() isPunct() isSpace() isUpperCase() isWhitespace()	abs() constrain() map() max() min() pow() sq() 、sqrt()
Time	Random Numbers	**External Interrupts**	**Interrupts**
delay() delayMicroseconds() micros() 、millis()	random() randomSeed()	attachInterrupt() detachInterrupt()	interrupts() noInterrupts()
Communication	**USB**		
serial stream	Keyboard Mouse		

✧ 建立開發環境(https://www.arduino.cc/en/software)





圖二、Arduino 開發環境

✧ Arduino 程式主要架構

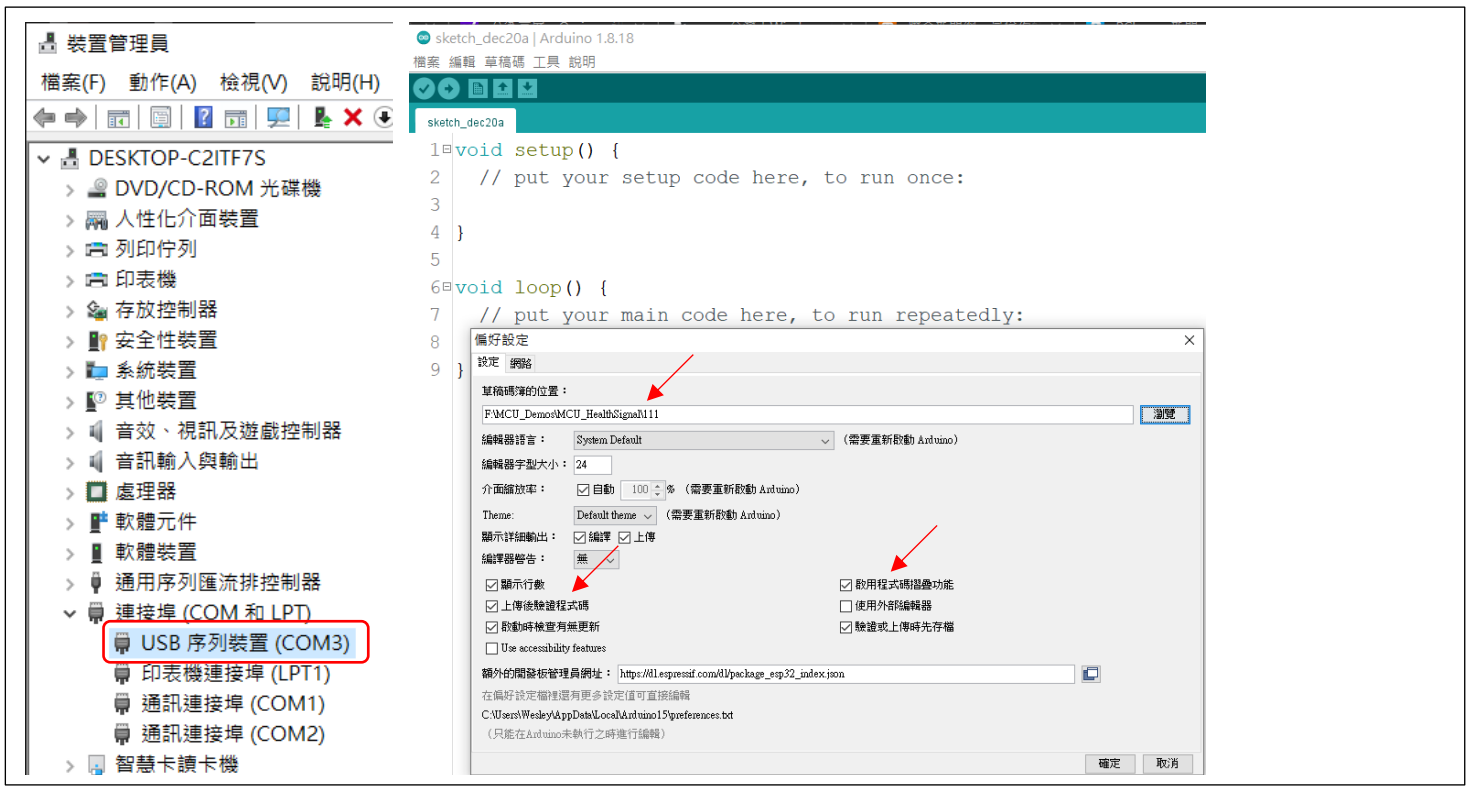
setup() 與 loop() 構成 Arduino 程式碼(Sketch)

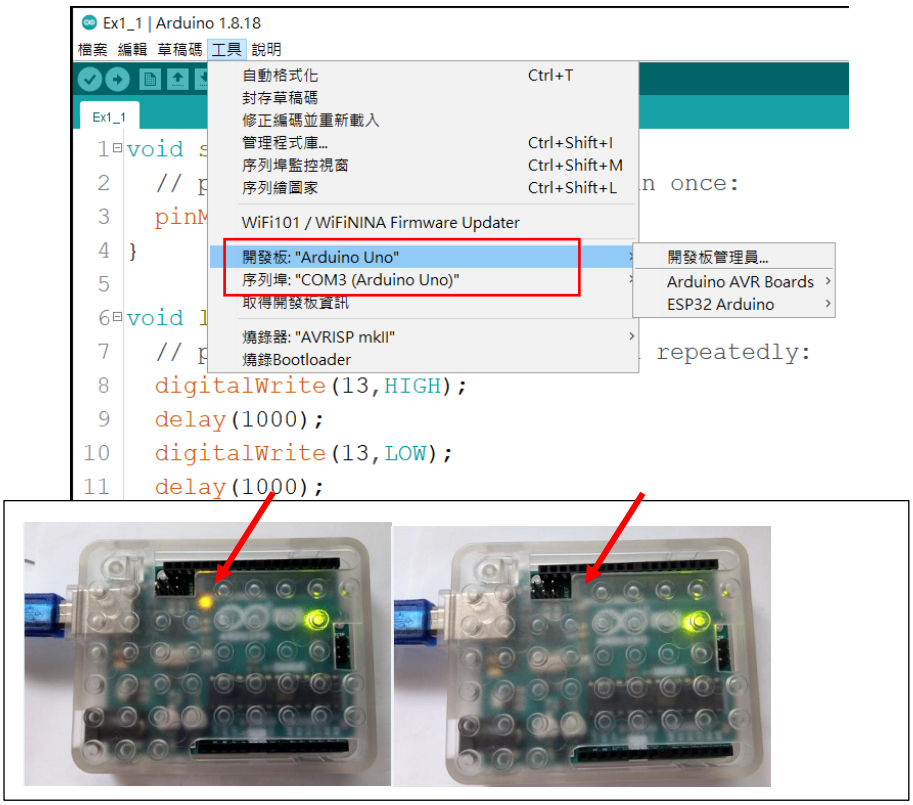
- setup(): 程式初始化
  在這個函式範圍內放置初始化 Arduino 開發板的程式 - 在重複執行的程式(loop())之前執行，主要功能是將所有 Arduino 開發板的 pin 腳設定(設定輸入或輸出腳色)，元件設定(位址)，需要初始化的部分設定等等。
- loop(): 迴圈重複執行
  在此放置的 Arduino 程式碼。這部份的程式會一直重複的被執行，直到 Arduino 開發板被關閉。

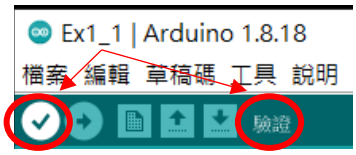
Ex.01 Test Arduino

```
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,HIGH);
  delay(1000);
  digitalWrite(13,LOW);
  delay(1000);
}
```

Arduino 接上 PC/NB

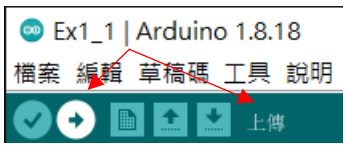






草稿碼使用了 924 bytes (2%) 的程式儲存空間。上限為 32256 bytes。

全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes



Reading | ################ | 100% 0.13s

avrdude: verifying ...

avrdude: 924 bytes of flash verified

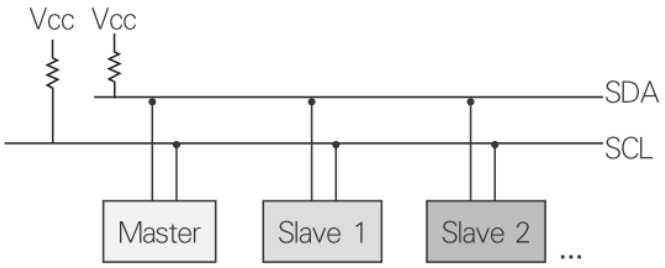
avrdude done. Thank you.

Tech specs

MICROCONTROLLER	ATmega328P
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	14 (of which 6 provide PWM output)
PWM DIGITAL I/O PINS	6
ANALOG INPUT PINS	6
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA
FLASH MEMORY	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	68.6 mm
WIDTH	53.4 mm
WEIGHT	25 g

I²C (Inter-Integrated Circuit)通訊

I²C 是內部整合電路的稱呼，是一種可在 IC 之間做資料傳輸的低速、序列式匯流排(Serial Bus)架構，使用主從架構，由飛利浦公司在 1980 年代為了方便
同一電路板上各個組件相互通信而開發出來的介面。I²C 的正確讀法為 "I-squared-C"。使用 I²C 協定不需要為其專利付費，但製造商仍然需要付費以獲得 I²C Slave (從屬裝置位址)。



圖三、I²C 運作原理

在系統運作上，I²C 採用 Master/Slave 架構，Master 裝置負責同步時脈(通常是微處理器負責發送時脈與位址信號)，並進行信號傳輸的初始化。同一時間，只能有一 Master 腳色運作，但可有多個 Slave 裝置。每一 Slave 裝置(通常是感測器元件)必須有一唯一的 I²C 位址(Address)，Master 是透過 I²C 位址來指定要溝通的 Salve 裝置。

I²C 的參考設計使用一個 7 位元長度的位址空間但保留了 16 個位址，所以在一組匯流排最多可和 112 個節點通訊。常見的 I²C 匯流排依傳輸速率的不同而有不同的模式：標準模式（100 Kbit/s）、低速模式（10 Kbit/s），但時脈頻率可被允許下降至零，這代表可以暫停通訊。而新一代的 I²C 匯流排可以和更多的節點（支援 10 位元長度的位址空間）以更快的速率通訊：快速模式（400 Kbit/s）、高速模式（3.4 Mbit/s）。

Arduino 開發板上的 I2C 腳位

Arduino UNO 相容開發板上，是以 A4、A5 腳位做為 I2C 的 SDA、SCL 腳位 (其它款 Arduino 開發板可能使用不同腳位)。Arduino IDE 已提供了可操作 I2C 通訊的 Wire 函式庫。

```
#include <Wire.h>

setup() {
  Wire.begin();
}
...
```

← 函式庫中已預建 Wire 物件，所以 #include <Wire.h> 之後就可以直接使用

Ex.02 Display Text by OLED (1.3”)

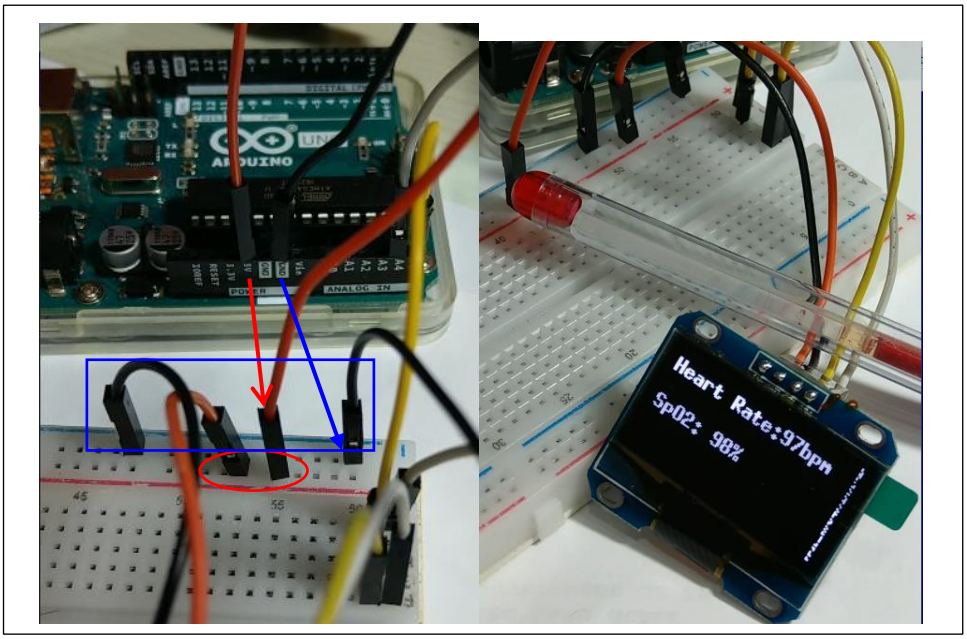
Component: 1.3 吋 OLED 液晶顯示模組 藍(白)字黑底 I²C 通信 128*64(??)

Spec.

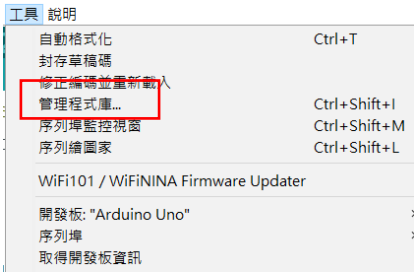
I2C接口定義 接口順序是 VCC，GND，SCL，SDA：

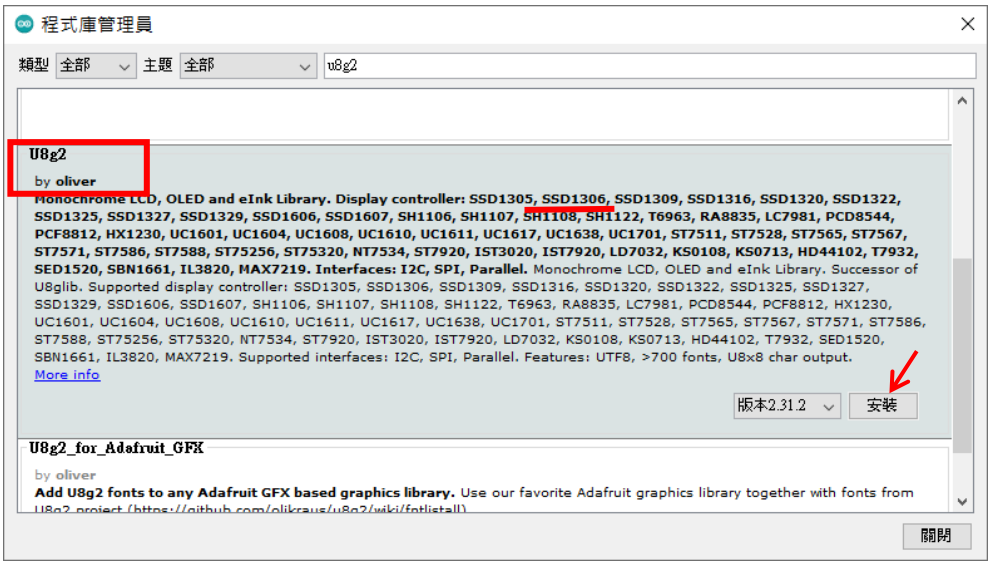
- VCC（電源正極） 5V
- GND（電源負極）
- SCL（時鐘線） A5
- SDA（數據線） A4

- OLED自發光,無需背光, 省電節能
- 分辨率高：128*64
- 可視角度大：＞160˚
- 超低功耗：全屏點亮時 0.08W，正常全屏顯示中文字 0.06W（遠低於 TFT、LCD 等技術）
- 寬電壓支持：無需任何修改，直接支持 3.3~5V 直流
- 工作溫度：-40℃~70℃
- 模組尺寸(長寬厚)：29.42x14.7(mm)
- IO 口占用少：採用 IIC 通信方式，最多只要 4 個 IO 口就能驅動
- 無字庫：用取模軟件取字，不需字庫



Step.1 Install u8g2 lbrary 程式庫





```
/*  Universal 8bit Graphics Library (https://github.com/olikraus/u8g2/)
                                        Copyright (c) 2016, olikraus@gmail.com      All rights reserved.*/
#include <Arduino.h>
#include <U8g2lib.h>
#ifdef U8X8_HAVE_HW_I2C
#include <Wire.h>
#endif
// Please UNCOMMENT one of the contructor lines below
U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ SCL, /* data=*/ SDA, /* reset=*/ U8X8_PIN_NONE);
// All Boards without Reset of the Display
// End of constructor list
String s0,s1;
void setup(void)
{
  u8g2.begin();
  u8g2.enableUTF8Print();//enable UTF8 support for the Arduino print() function
  //u8g2.setFont(u8g2_font_unifont_t_chinese2); // use chinese2 for all the glyphs of "你好世界"
  //u8g2.setFont(u8g2_font_ncenB10_tr);
  u8g2.setFont(u8g2_font_7x13B_tf);
  u8g2.setFontDirection(0);
}

void loop(void)
{
  u8g2.clearBuffer();
  u8g2.setCursor(2, 15);
  //u8g2.print("Hello World!!");
  //s0="Temp.:"+String(20.123,1)+"°C";
  s0="Heart Rate:"+String(97)+"bpm";
  u8g2.print(s0);
  //u8g2.print("Hello World!!");
//  u8g2.print("");//Fail to display  銘傳大學醫管系");
  u8g2.setCursor(2, 40);
  //u8g2.print("世界和平!!");
  //s1="Humi.: "+String(78)+"%";
  s1="SpO2: "+String(98)+"%";
  u8g2.print(s1);
//  u8g2.print("");//Fail to display  醫療資訊實驗室!!"); since must have Font
  u8g2.sendBuffer();
  delay(2000);
}
```

```
u8g2.setFont(u8g2_font_unifont_t_chinese2);
```

草稿碼使用了 31072 bytes (96%) 的程式儲存空間。上限為 32256 bytes。

全域變數使用了 1466 bytes (71%) 的動態記憶體，剩餘 582 bytes 給區域變數。上限為 2048 bytes 。

```
u8g2.setFont(u8g2_font_7x13B_tf);
```

草稿碼使用了 11712 bytes (36%) 的程式儲存空間。上限為 32256 bytes。

全域變數使用了 1474bytes(71%)的動態記憶體，剩餘 574bytes 給區域變數。上限為 2048 bytes。

Ex.03 Measuring Humidity & Temperature

Sensor: DHT11

Spec.

供电电压： 3.3~5.5V DC

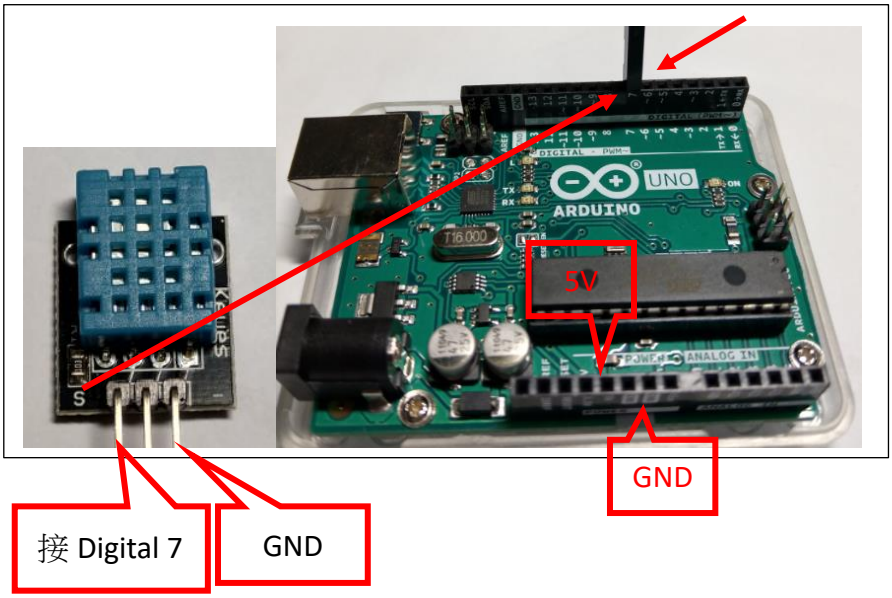
输 出： 单总线数字信号

测量范围： 湿度 20-90%RH， 温度 0~50℃

测量精度： 湿度+-5%RH， 温度+-2℃

分 辨 率： 湿度 1%RH， 温度 1℃

长期稳定性： <±1%RH/年



https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTlib

```
#include <dht.h>
dht DHT;
#define DHT11_PIN 7 //Digital Pin: 7
void setup()
{
  Serial.begin(9600);
  Serial.println("DHT11 Humidity & temperature Sensor\n\n");
  Serial.println("DHT TEST PROGRAM ");
  Serial.print("LIBRARY VERSION: ");
  Serial.println(DHT_LIB_VERSION);
  Serial.println("Type,\tstatus,\tHumidity (%),\tTemperature (C)");
  delay(1000);//Wait before accessing Sensor
}
void loop()
{
  // READ DATA
  Serial.print("DHT11, \t");
  int chk = DHT.read11(DHT11_PIN);
  switch (chk)
  {
    case DHTLIB_OK:
        Serial.print("OK,\t");
        // DISPLAY DATA
        Serial.print(DHT.humidity, 1);
        Serial.print(",\t");
        Serial.println(DHT.temperature, 1);
        break;
    case DHTLIB_ERROR_CHECKSUM:
        Serial.print("Checksum error,\t");
        break;
    case DHTLIB_ERROR_TIMEOUT:
        Serial.print("Time out error,\t");
        break;
    case DHTLIB_ERROR_CONNECT:
        Serial.print("Connect error,\t");
        break;
    case DHTLIB_ERROR_ACK_L:
        Serial.print("Ack Low error,\t");
        break;
    case DHTLIB_ERROR_ACK_H:
        Serial.print("Ack High error,\t");
        break;
    default:
        Serial.print("Unknown error,\t");
        break;
  }
  delay(2000);
}
```

Ex1_3:1:10: fatal error: dht.h: No such file or directory

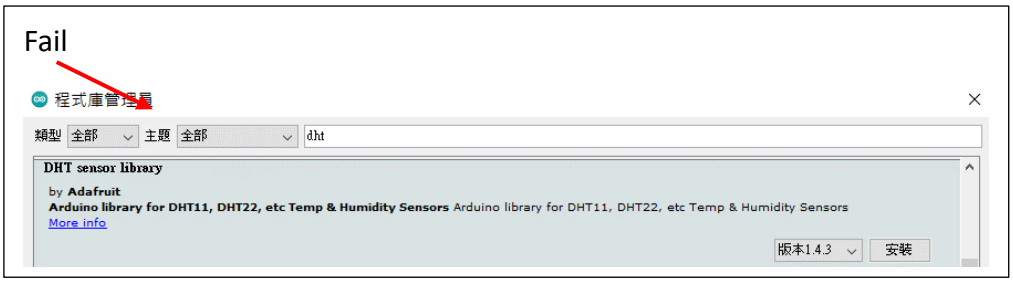
#include <dht.h>

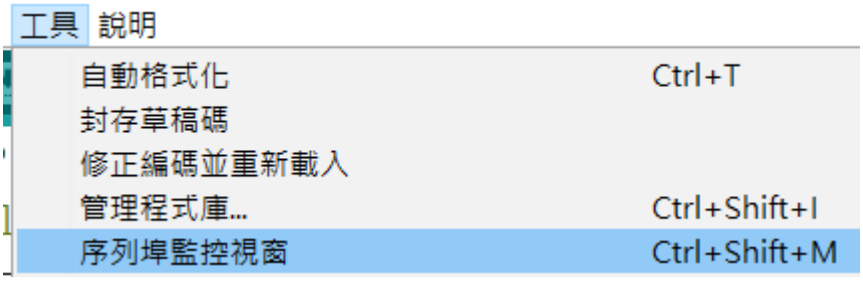
^~~~~~~

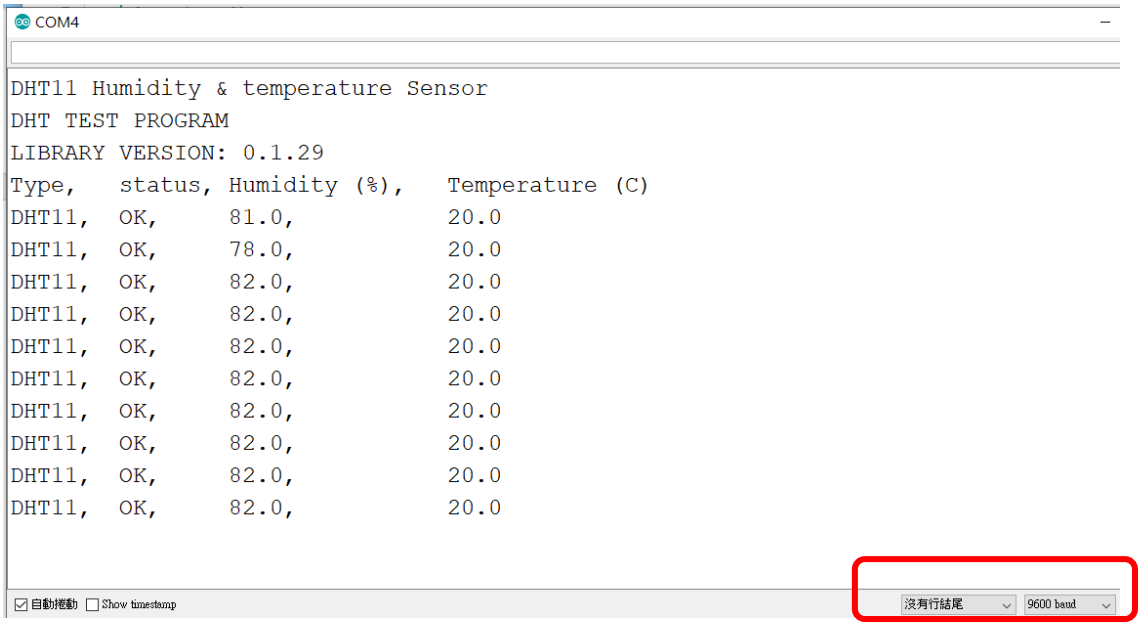
compilation terminated.

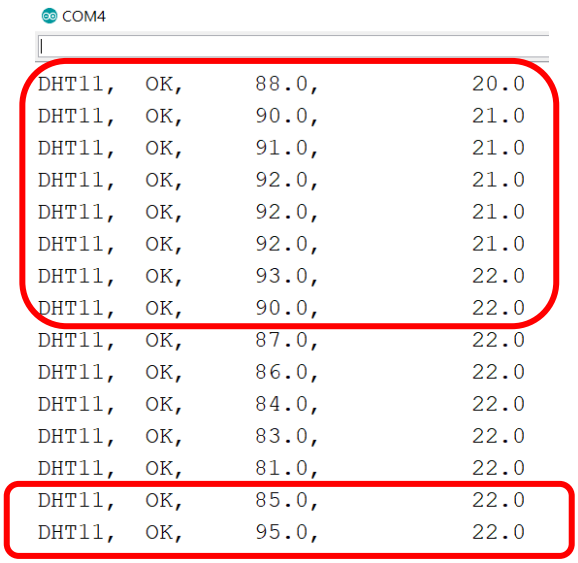
exit status 1

dht.h: No such file or directory




工具
說明
自動格式化 Ctrl+T
封存草稿碼
修正編碼並重新載入
管理程式庫... Ctrl+Shift+I
序列埠監控視窗 Ctrl+Shift+M


COM4
DHT11 Humidity & temperature Sensor
DHT TEST PROGRAM
LIBRARY VERSION: 0.1.29
Type, status, Humidity (%), Temperature (C)
DHT11, OK, 81.0, 20.0
DHT11, OK, 78.0, 20.0
DHT11, OK, 82.0, 20.0
DHT11, OK, 82.0, 20.0
DHT11, OK, 82.0, 20.0
DHT11, OK, 82.0, 20.0
DHT11, OK, 82.0, 20.0
DHT11, OK, 82.0, 20.0
DHT11, OK, 82.0, 20.0
DHT11, OK, 82.0, 20.0
自動捲動
Show timestamp
沒有行結尾
9600 baud

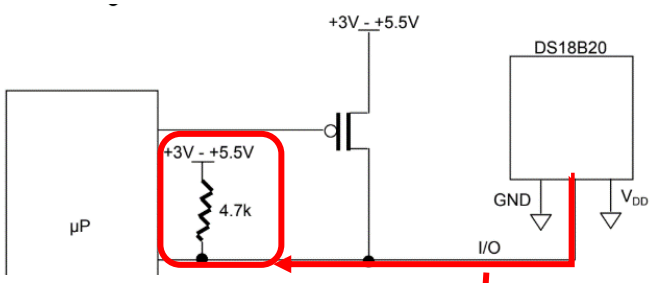

COM4
DHT11, OK, 88.0, 20.0
DHT11, OK, 90.0, 21.0
DHT11, OK, 91.0, 21.0
DHT11, OK, 92.0, 21.0
DHT11, OK, 92.0, 21.0
DHT11, OK, 92.0, 21.0
DHT11, OK, 93.0, 22.0
DHT11, OK, 90.0, 22.0
DHT11, OK, 87.0, 22.0
DHT11, OK, 86.0, 22.0
DHT11, OK, 84.0, 22.0
DHT11, OK, 83.0, 22.0
DHT11, OK, 81.0, 22.0
DHT11, OK, 85.0, 22.0
DHT11, OK, 95.0, 22.0

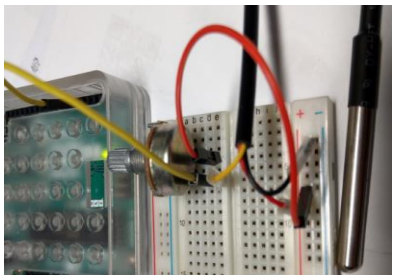
E04. Measuring Temperature

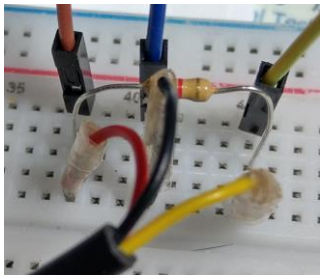
Sensor: 防水型 **DS18b20** 溫度感測器 **100CM** 長度 帶不鏽鋼探頭

Spec.

1. 探頭採用原裝 DS18B20 溫度感測器晶片, 不銹鋼管封裝 防水防潮防生鏽。
   不銹鋼外殼(6*50mm)，引線長度 100cm 。
2. 每個探頭經過嚴格測試後獨立包裝, 3.0V～5.5V 供電 , 9～12 位可調分辨率。
3. 感溫範圍寬-55℃ ～ +125℃
4. 無需外部元件，獨特的單總線接口
5. 輸出引線：紅色(VCC)，黃色(DATA)，黑色(GND)
6. 需 4.7k 歐姆電阻 (or 可變電阻 10K)



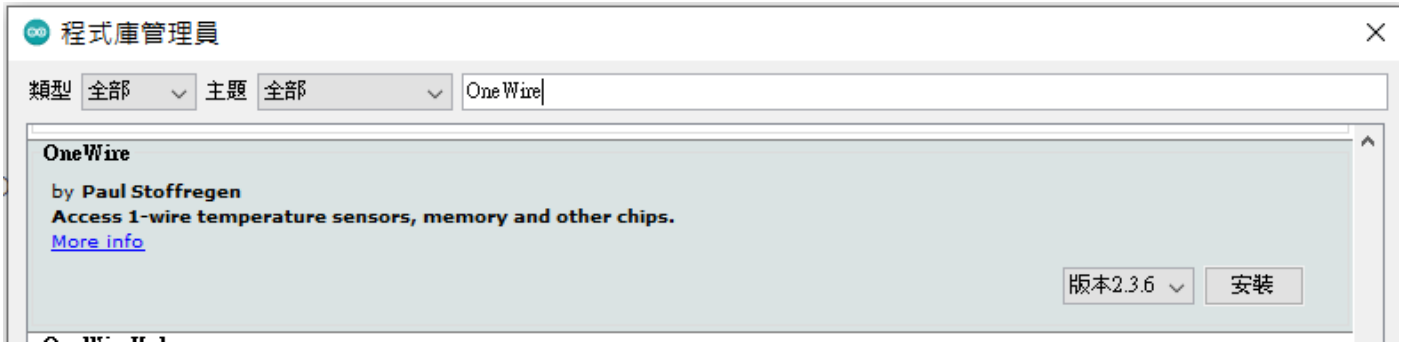


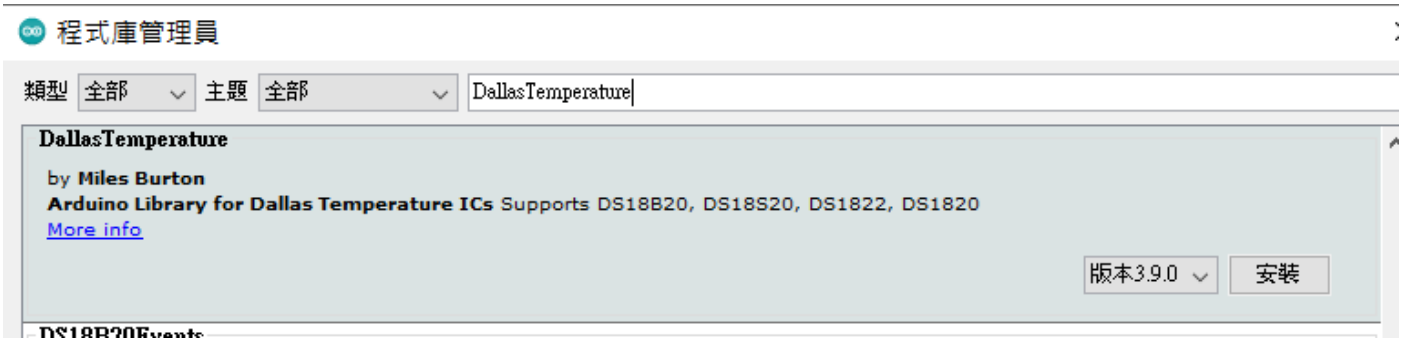


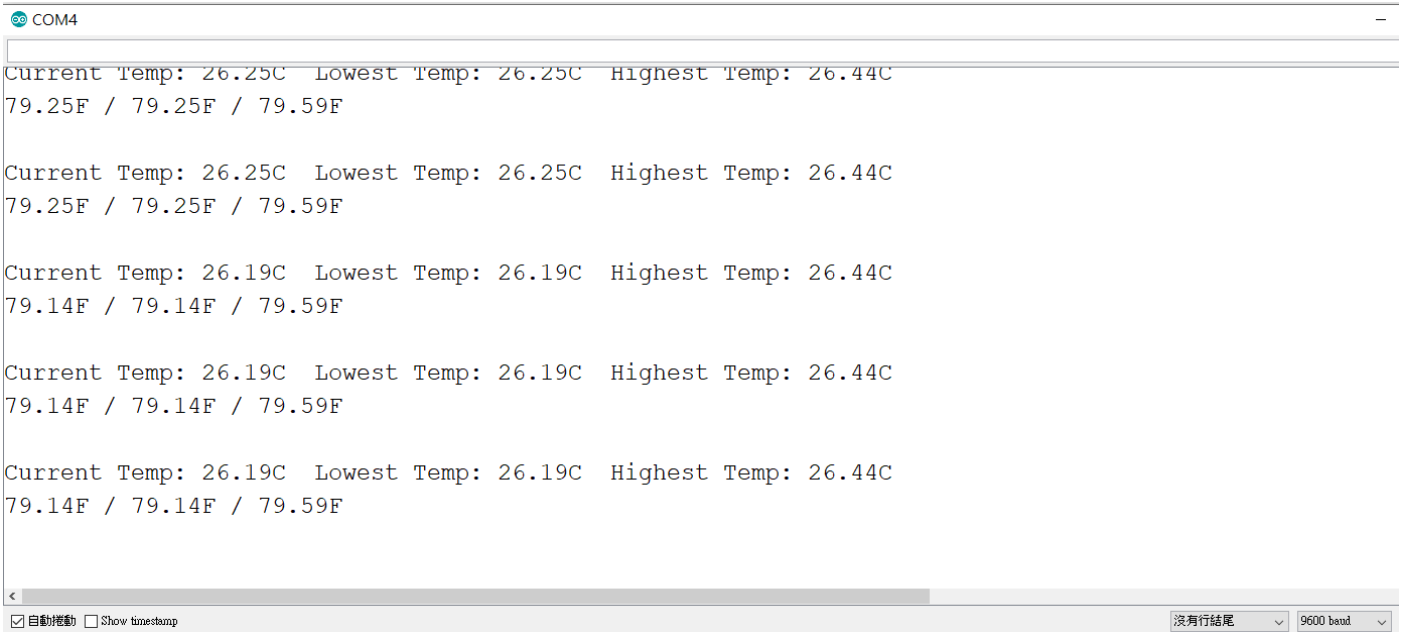
```
#include <OneWire.h>
#include <DallasTemperature.h>
//OK! Jan. 03, 2020
// Variables for temperature readings
float myTemp;
float myHighTemp;
float myLowTemp = 50;

// DS1820 Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 2

// Setup oneWire instance to communicate with devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup()
{ // start serial port
   Serial.begin(9600);
   Serial.println("Dallas Temperature IC Control Library Demo");
   sensors.begin();// Start the OneWire library
}
void loop()
{
   readtemp();// Read the temperature
   // Write the Results to the serial Monitor
   serialPrint();
}
```

D2

```
void readtemp()
{
   // call sensors.requestTemperatures() to issue a global temperature
   // request to all devices on the bus\
   //Send the command to get temperatures
   sensors.requestTemperatures();
   myTemp = (sensors.getTempCByIndex(0));
   // Set High or Low Temp
   if (myTemp < myLowTemp)
       myLowTemp = myTemp;
   else if (myTemp > myHighTemp)
      myHighTemp = myTemp;
}
void serialPrint()
{
   Serial.print("Current Temp: ");
   Serial.print(myTemp);
   Serial.print("C");
   Serial.print("    Lowest Temp: ");
   Serial.print(myLowTemp);
   Serial.print("C");
   Serial.print("    Highest Temp: ");
   Serial.print(myHighTemp);
   Serial.println("C");
   delay (500);
   Serial.print(DallasTemperature::toFahrenheit(myTemp));
   Serial.print("F / ");
   Serial.print(DallasTemperature::toFahrenheit(myLowTemp));
   Serial.print("F / ");
   Serial.print(DallasTemperature::toFahrenheit(myHighTemp));
   Serial.println("F ");
   Serial.println();
   delay (500);
}
```

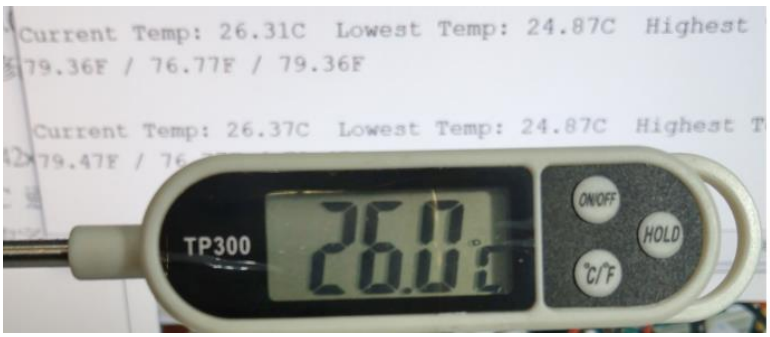




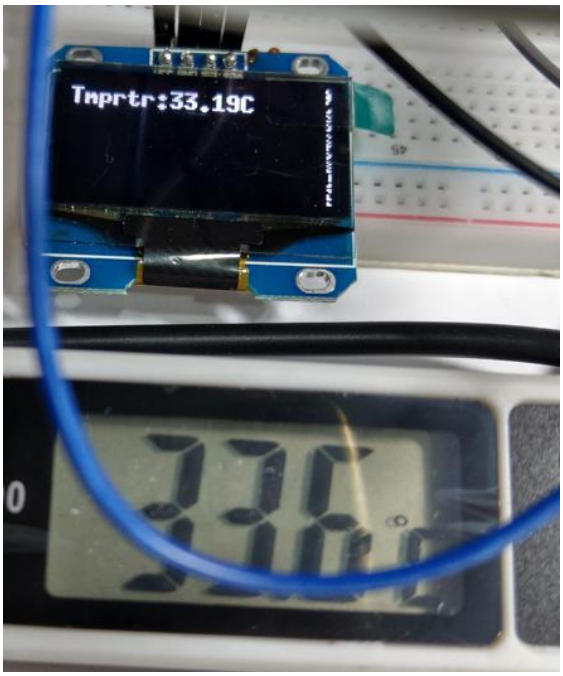


OneWire

Access 1-wire temperature sensors, memory and other chips.

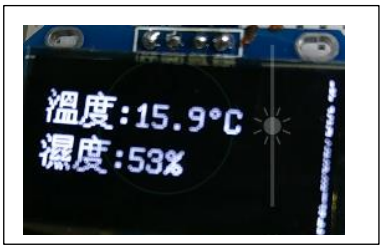


Ex.5. Measuring Temperature & Display on OLED




```
#include <Arduino.h>
#include <U8g2lib.h>
#ifdef U8X8_HAVE_HW_I2C
    #include <Wire.h>
#endif
#include <OneWire.h>
#include <DallasTemperature.h>

U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, SCL, SDA, U8X8_PIN_NONE);
// Variables for temperature readings
float myTemp;
#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
String s0,s1;
void setup(void)
{
```

```
  u8g2.begin();
  u8g2.enableUTF8Print();//enable UTF8 support for the Arduino print() function
  u8g2.setFont(u8g2_font_7x13B_tf);
  u8g2.setFontDirection(0);
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");
  sensors.begin();// Start the OneWire library
}

void loop(void)
{
  u8g2.clearBuffer();
  u8g2.setCursor(2, 15);
  sensors.requestTemperatures();
  myTemp = (sensors.getTempCByIndex(0));
  s0="Tmprtr:"+String(myTemp)+"C";
   Serial.println(s0);
  u8g2.print(s0);
  u8g2.sendBuffer();
  delay(2000);
}
```



Ex.6. Display Chinese Font, Temperature & Humidity on OLED

Copy myFont.h before run

```
#include <Arduino.h>
#include <U8g2lib.h>
#ifdef U8X8_HAVE_HW_I2C
  #include <Wire.h>
#endif
#include "myFont.h"

U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, SCL, SDA, U8X8_PIN_NONE);
float temp;
int humd;
char bufT[30],bufH[20], buf[10];

void setup(void)
{
  Serial.begin(9600);
  u8g2.begin();
  u8g2.setFont(myFont);
  u8g2.enableUTF8Print();//  啟用顯示 UTF-8  編碼字串
  u8g2.setFontDirection(0);
  temp=15.5;
  humd=70;
  Serial.print("°度溫濕心律血氧\n");
}
void loop(void)
{
  temp=15.5f+random(-50, 50)/100.f;
  //sprintf(bufT,"溫度: %.1f°C",temp); //Fail: Got  溫度: ?°C
```

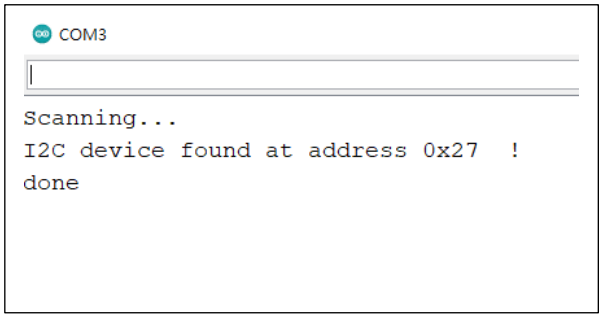
```
  dtostrf(temp, 4,1, buf);
  sprintf(bufT,"溫度: %s°C",buf);
  Serial.print(bufT);
  Serial.print("\n");
  humd=70+random(-25, 25);
  sprintf(bufH,"濕度: %3d %%", humd);
  Serial.print(bufH);
  Serial.print("\n");
  u8g2.firstPage();
  do {
    u8g2.setCursor(5,25);
    u8g2.print(bufT);
    u8g2.setCursor(5,45);
    u8g2.print(bufH);
  }while(u8g2.nextPage());
  delay(1000);
}
```

LCD 2004 5V LCM IIC/I2C 藍底白字液晶顯示模組

LCD 2004 5V LCM IIC/I2C 20x4 20*4 藍底白字液晶顯示模組

- 型號：LCM I2C 2004 V1-GP
- 尺寸：98*60*19mm
- 液晶屏：2004 字符型（HD44780）普通顯示效果
- 顏色：藍底白字
- 電壓：5V
- 端口：I2C
- 地址：0x27
- I2C 總線控制僅佔用 2 個 IO 口
- 液晶背光可控，可以通過跳線控制，也可以通過程序控制

```
// Arduino I2C Scanner

// Devices with higher bit address might not be seen properly.

#include <Wire.h> //include Wire.h library

void setup()

{

  Wire.begin(); // Wire communication begin

  Serial.begin(9600); // The baudrate of Serial monitor is set in 9600
```



```
    while (!Serial); // Waiting for Serial Monitor
    Serial.println("\nI2C Scanner");
}
void loop()
{
    byte error, address; //variable for error and I2C address
    int nDevices;
    Serial.println("Scanning...");
    nDevices = 0;
    for (address = 1; address < 127; address++ )
    {
        // The i2c_scanner uses the return value of the Write.endTransmisstion to see if
        // a device did acknowledge to the address.
        Wire.beginTransmission(address);
        error = Wire.endTransmission();
        if (error == 0)
        {
            Serial.print("I2C device found at address 0x");
            if (address < 16)
                Serial.print("0");
            Serial.print(address, HEX);
            Serial.println("    !");
            nDevices++;
        }
```

```
    else if (error == 4)
    {
      Serial.print("Unknown error at address 0x");
      if (address < 16)
        Serial.print("0");
      Serial.println(address, HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");
  delay(5000); // wait 5 seconds for the next I2C scan
}
```

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4);    // set the LCD address to 0x27 for a 20 chars and 4 line display

//  接線  GND==>GND    VCC==>5V    SDA==>A4    SCL==>A5

void setup()

{

   lcd.init();

   lcd.backlight();

   lcd.setCursor(0, 0);

   lcd.print("12345678901234567890");

   lcd.setCursor(0, 1);

   lcd.print("This is Line 2!");

   lcd.setCursor(0, 2);

   lcd.print("This is Line 3!!");

   lcd.setCursor(0, 3);

   lcd.print("This is Line 4!");

}

void loop()

{

}
```