

Arduino 與

感測器教學

第3章 Arduino 與感測器教學

3-1 Arduino 單晶片教學模組

Arduino 計畫是在 2005 年 1 月由 Massimo Banzi、David Cuartielles、Gianluca Martino、Tom Igoe、以及 David Mellis 共同創作於義大利米蘭的 Ivera 互動設計學院，Arduino 主要是針對創意互動設計有興趣者、設計師、以及藝術家所開發。原始構想是希望讓設計師及藝術家們，透過 Arduino 很快的學習電子和感測器的基本知識，且能快速的設計、製作作品的原型，很容易與目前設計系所學的 FLASH，MAX/MSP，Virtool 等軟體整合，使得虛擬與現實的互動更加容易。互動的內容設計才是設計師的主要訴求，至於怎麼拼湊一個單晶片開發板，或是當中涉及如何建構電路之類的相關電子知識，就並非設計師需要了解的，因此非常適合不具電子背景的人使用，就可以設計出各種不同的互動裝置。

Arduino 是一塊使用 Atmel AVR 單晶片發展出來的 I/O 介面控制開發板，最初是採用 Atmel AVR 系列的 8 位元微控制器 IC 來設計成的單板微電腦，2013 年則已進步到使用 Atmel ARM 架構的 32 位元微控制器。軟體方面是使用類似 Java，C 語言的 Processing 軟體開發環境，我們可以很容易使用 Arduino 語言開發完成電子元件的控制，例如 LED、按鍵、步進馬達、各類感測器或其他控制裝置。更重要的是它非常的簡單，可以與眾多程式語言(FLASH、MAX/MSP、Virtool、C#、VB、C++)結合，成為教學程式語言的教具，非常適合應用在教學活動上。因為 Arduino IDE 開發介面基於開放原始碼(open source)的原則，可以免費下載參考與使用，Arduino 開放策略使得目前世界各地智慧互動電子的專題設計蓬勃發展。

Arduino 團隊因應各種專案應用，發展出許多不同大小與效能的開發板，圖 1 所列為比較常用的開發板。



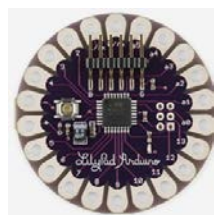
Arduino UNO



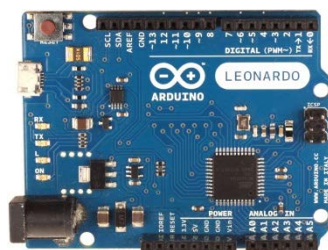
Arduino Nano



Arduino Pro Mini



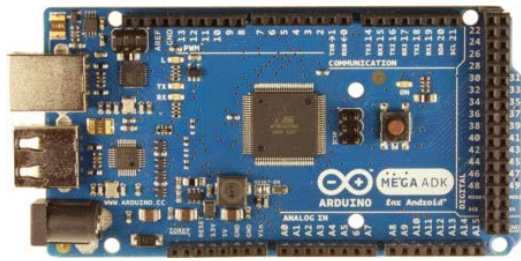
LilyPad Arduino



Arduino Leonardo



Arduino Mega2560



Arduino Mega ADK

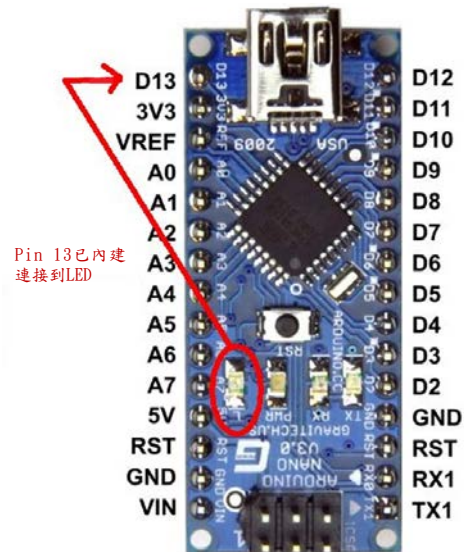


Arduino Due

本教材主要是以 Arduino Nano 開發板為主要控制核心，其特色簡要說明如下：

Arduino Nano 開發板：

- 14 支數位 I/O 腳(D0~D13)，其中 6 支可做 PWM 輸出。
- 8 支類比輸入腳(A0~A7)，為 10 位元 A/D 轉換器。
- 工作電壓 5V。
- V_{in} 接腳輸入電壓範圍為 6~20V。
- V_{in} 接腳輸入電壓建議為 7~12V。
- 每腳位輸入電流可達 40 mA
- 時脈 16MHz
- V3.0(ATmega328)
 - Flash Memory 32 KB
 - SRAM 2 KB
 - EEPROM 1 KB
- V2.0 (ATmega168)
 - Flash Memory 16 KB
 - SRAM 1 KB
 - EEPROM 512 bytes



專有名詞解析

Flash Memory :

中文稱『快閃記憶體』，是一種最新的唯讀記憶體(Read Only Memory)形式。唯讀記憶體英文簡稱 ROM，其特性是所儲存的資料只能被讀出，而不能寫入，因此資料不會消失，即使停電資料依然存在，所以 Flash Memory 常用來儲存微電腦系統所需執行的程式碼或固定資料，所以又稱程式記憶體。

EEPROM :

中文稱『電子式可抹除程式化唯讀記憶體』，也是一種唯讀記憶體 (ROM)形式，這種記憶體只需要用電氣的方法，在數秒內即可將 ROM 上的資料清除，然後再用類似燒錄的方法，可以透過程式指令把所需的資料放到 EEPROM 上。一般會用在類似如密碼資料的儲存，系統即使斷電，密碼資料不會消失，又可以透過程式指令更改密碼資料。

SRAM :

中文稱『靜態隨機存取記憶體』，SRAM 能很快地完成資料的讀出及寫入工作，雖然 SRAM 可以隨時讀出及寫入資料，但在關機後，SRAM 的內容就會全部消失。因此 SRAM 常用來儲存使用者隨時編寫即將執行的資料或非固定資料，所以又稱資料記憶體。

Arduino 應用

Arduino 因為是一種開放授權的互動環境開發技術，不僅軟體是開放源碼，硬體也是開放的。軟體的開發環境可在網上免費下載，而 Arduino 的電路設計圖也可從官方網站自行下載，依據自身之需求進行修改，所以 Arduino 在業餘開發者與電子設計愛好者的圈子裡廣受歡迎，也造就了創客（Maker）的興盛，製作出的創意作品不勝枚舉！以下羅列出創客族利用 Arduino 所發展的創意作品。

- 電子鼓 <http://bitly.com/yFC1M6>
- 穿戴式科技 <http://bitly.com/y6kqVo>
- 心跳監測器 <http://danjuliodesigns.com/sparkfun/hrmi.html>
- DIY 紅外線遙控品 <http://bitly.com/w9FCxa>
- 蜘蛛 <http://bitly.com/zrjMg2>
- 結合樂高與 Arduino <http://bitly.com/wLBvY0>
- 遠端遙控灑水系統 <http://bitly.com/wdrPee>
- 機器人 https://www.youtube.com/watch?v=NZ6V63u_-D8

本章練習單

1. 請簡要說明 Flash Memory 的特性與用途。
2. 請簡要說明 SRAM 的特性與用途。
3. 請利用網際網路搜尋出 Arduino 開發板實際應用的例子，並註明其網址。

3-2 感測器教學模組

本教材以「自然環境感測技術」為主軸，主要發展溫濕度、風速、風向、雨量及 PM2.5 空氣品質感測等為主，下面就使用到相關的感測器介紹如下：

(1)、溫濕度感測技術[1-4]

(1.1) 溫度：

溫度是用來表示物體冷熱程度的物理量。只能藉由物體隨溫度變化而產生的某些特性來間接測量溫度，用來量度物體溫度數值的叫作溫標。溫標定義了溫度的零點和此溫標的度數間距，理論上，溫度沒有極高點，只有極低點，也就是熱力學中的絕對零度(Absolute Zero)。

絕對溫度：絕對溫度是熱力學上的一種單位，把分子能量最低時的溫度定為絕對零度（記為 0°K ），相當於 -273.15°C ，是一種極限溫度，在此種溫度下，分子運動不再具有可以轉移給其他系統的能量。

(1.2) 濕度：

溼度一般在氣象學中指的是空氣溼度，它是空氣中水蒸氣的含量。由於大氣中的水蒸氣可以占空氣體積的 0%到 4%，一般在列出空氣中各種氣體的成分的時候是指這些成分在乾空氣中所占的成分，有絕對濕度與相對濕度等

絕對溼度：絕對溼度是指一定體積的空氣中含有的水蒸氣的質量，一般其單位是公克／立方公尺，其最大限度是飽和狀態下的最高溼度。

相對溼度：相對溼度（RH）是絕對溼度與最高溼度之間的比，它的值顯示水蒸氣的飽和度有多高。隨著溫度的增高，空氣中可以含的水就增多。也就是說，在同樣多的水蒸氣的情況下，溫度降低，相對溼度就會升高；溫度升高，相對溼度就會降低。

(1.3) 溫溼度感測模組：

本教材採用 DHT11 溫溼度感測模組(圖 1 所示)，DHT11 溫濕度感測器是一款含有已校準數位信號輸出的溫濕度複合感測器。它使用專用的數位模組採集技術和溫濕度感測技術，確保產品具有極高的可靠性與卓越的長期穩定性。感測器包括一個電阻式感濕元件和一個 NTC 測溫元件，並與一個高性能 8 位單片機相連接，其規格如表 1 所示。

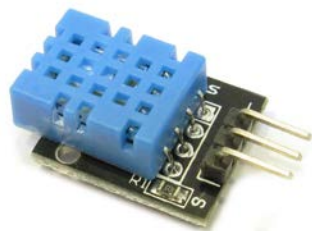


圖 1、DHT11 溫溼度感測模組

表 1、DHT11 規格

| | |
|------|---------------------------|
| 型號 | DHT11 |
| 測試範圍 | 濕度：20-90% RH 溫度：0-50°C |
| 測濕精度 | ±5% RH |
| 測溫精度 | ±2°C |

(2)、風速感測技術[5-9]

測量地面風速之儀器，稱之為風速計(anemometer)。目前市面上有相當多種量測風速的工具，一般常見的風速計依不同工作原理可分風杯式風速計(圖 2 所示)、風車式風速計(圖 3 所示)、熱線式風速計(圖 4 所示)、超音波風速計(圖 5 所示)、雷射都卜勒(Doppler)風速計及壓力式風速計等。



圖 2、風杯式風速計



圖 3、風車式風速計



圖 4、熱線式風速計



圖 5、超音波風速計

本教材使用風杯式風速計模組(圖 6 所示)來測量風速，風杯式風速計部份由風杯與電磁開關組成，使用計數式的風速量測原理，就標準的風速杯而言，每轉一圈會觸發一次磁開關，而風速則利用 Arduino 單晶片計算每赫茲傳來幾次的開關數即可得。



圖 6、風向風速測量模組

(3)、風向感測技術[10-11]

風向是指風吹來的方向，有各種各樣的儀器用來測量風向，如風向袋和風向標。通常是透過基本方向或方位度來了解。它的工作原理在於盡量減少移動的空氣阻力。風向標所指的方向是與風向相同。風向袋所指的方向是與風向相反。

風向計(圖 7 所示)就像是我們常看到的風向雞，它的一頭像箭一樣，當風吹時，它的箭頭就會指向風吹來的方向，而我國的風向測量的單位是十六方位，也就是東、東南東、東南、南南東、南、南南西、西南、西南西、西、西北西、西北、北北西、北、北北東、東北以及東北東這十六個方位。



圖 7、風向計

本教材使用圖 6 所示風向計模組來測量風向，風向為葉片與電位訊號組成，內部是由一個可調電阻值來換算出分電壓值，經類比轉成數位後再經 Arduino 單晶片處理並計算出方向角。

(4)、雨量感測技術[12-14]

雨量計(或量雨計、測雨計)是一種氣象學家和水文學家用來測量一段時間內某地區的降水量的儀器(降雪量的測量則需要使用雪量計)。雨量計的種類很多，常見的有虹吸式雨量計(圖 8 所示)、稱重式雨量計(圖 9 所示)、翻斗式雨量計(圖 10 所示)等等。



雨量計

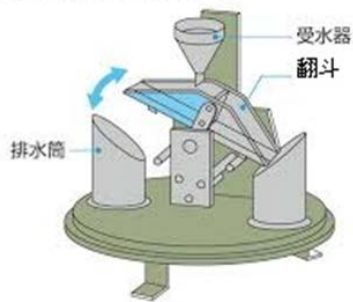


圖 8、虹吸式雨量計

圖 9、稱重式雨量計

圖 10、翻斗式雨量計

本教材使用翻斗式雨量感測模組(圖 11 所示)，當一端傾斗接受 0.25mm 或一定雨量時，傾斗即傾倒而排去斗內之雨水，另一端之傾斗翹起以接受雨水，如同翹翹板，每一轉動亦即 0.25mm 之雨水傾倒，都觸發開關接通電路，使用計數式的量測原理，而雨量則透過嵌入式微處理機計算幾次的開關數即可得。



圖 11、翻斗式雨量感測模組

(5)、PM_{2.5} 空氣品質感測技術

本教材採用 PMS5003 空氣品質感測模組(圖 12 所示)，採用激光散

射原理。即令激光照射在空氣中的懸浮顆粒物上產生散射，同時在某一特定角度收集散射光，得到散射光強隨時間變化的曲線。進而微處理器利用基於米氏（MIE）理論的算法，得出顆粒物的等效粒徑及單位體積內不同粒徑的顆粒物數量。



圖 12、PMS3003 空氣品質感測模組

本章練習單

1. 請利用網際網路搜尋其它可以感測溫度的感測器，並說明其的特性。
2. 請利用網際網路搜尋其它可以感測濕度的感測器，並說明其的特性。

參考文獻：

1. <https://zh.wikipedia.org/zhtw/%E6%B8%A9%E5%BA%A6>
2. <https://zh.wikipedia.org/zh-tw/%E7%83%AD%E5%8A%9B%E5%AD%A6%E6%B8%A9%E6%A0%87>
3. <https://zh.wikipedia.org/zh-tw/%E6%B9%BF%E5%BA%A6>
4. <http://iot.ttu.edu.tw/2015/11/16/dht11%E6%BA%AB%E6%BF%95%E5%BA%A6%E6%84%9F%E6%B8%AC%E5%99%A8%E4%BB%8B%E7%B4%B9/>
5. <http://en.wikipedia.org/wiki/Anemometer>
6. 楊煌棋，” 具遠端監視功能之風速計研製”，碩士論文，國立臺灣科技大學電機工程系，民國 97 年 7 月。
7. 劉民允，” 三杯式風速計之氣動力研究及設計應用 “，碩士論文，中華科技大學飛機系統工程研究所，民國 99 年 7 月。
8. Ristić, S., “Laser Doppler Anemometry and its Application in Wind Tunnel Tests,” Scientific Technical Review, 57, 3-4, pp. 64-76,

2007.

9. W. A. Oost, “The Pressure Anemometer—an Instrument for Adverse Circumstances,” *Journal of Climate And Applied Meteorology*. Vol. 22, pp. 2075–2084, 1983.
10. <https://zh.wikipedia.org/wiki/wiki/風向>
11. <http://www.sanzo.com/products/41/4315.html>
12. <http://www.wikiwand.com/zh-tw/雨量計>
13.
http://south.cwb.gov.tw/thing/thing_list/38//index.html?bull_id=35
1
14. https://market.cloud.edu.tw/content/senior/earth/yl_ld/content/6-1/common.htm

3-3 感測器整合程式教學模組

(1)、Arduino IDE 教學

本教材主要是使用 Arduino IDE 軟體做到感測器模組的控制，接下來要介紹如何使用 Arduino IDE 來撰寫程式並控制感測器。本教材所使用的 Arduino 開發板為 Arduino Nano，如圖 1 所示。

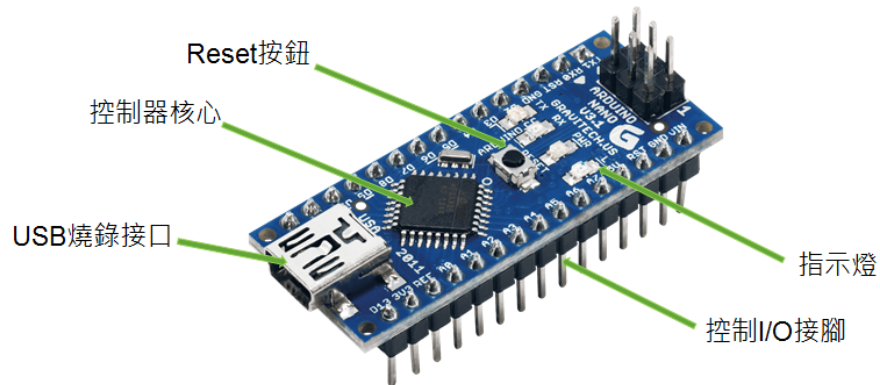


圖 1、Arduino Nano 開發板 [1]

第一步、下載 Arduino IDE [2-3]

首先先下載開發工具，才能寫程式，並燒到 Arduino 上面去。下載下來以後，就解壓縮出來即可。

下載網址：<http://arduino.cc/en/Main/Software>

第二步、安裝 Driver

接著把 Arduino 接上電腦，基本上電腦應該找不到 Driver，這時候就【我的電腦點選右鍵】→【點選管理】→【點選裝置管理員】，應該就會看到無法辨識的設備，在該設備上點選右鍵，選擇內容。然後接著就自己去尋找 Driver 的路徑，而 Driver 就在第二步下載下來的檔案當中，在解壓縮出來的資料夾當中\arduino-1.6.5\drivers。安裝完成以後，開啟裝置管理員後在連接埠中可查看到如圖 2 的 USB SerialPort(COM3)。記下 COM 編號，因為等等會透過此 SerialPort 把程式燒進去。

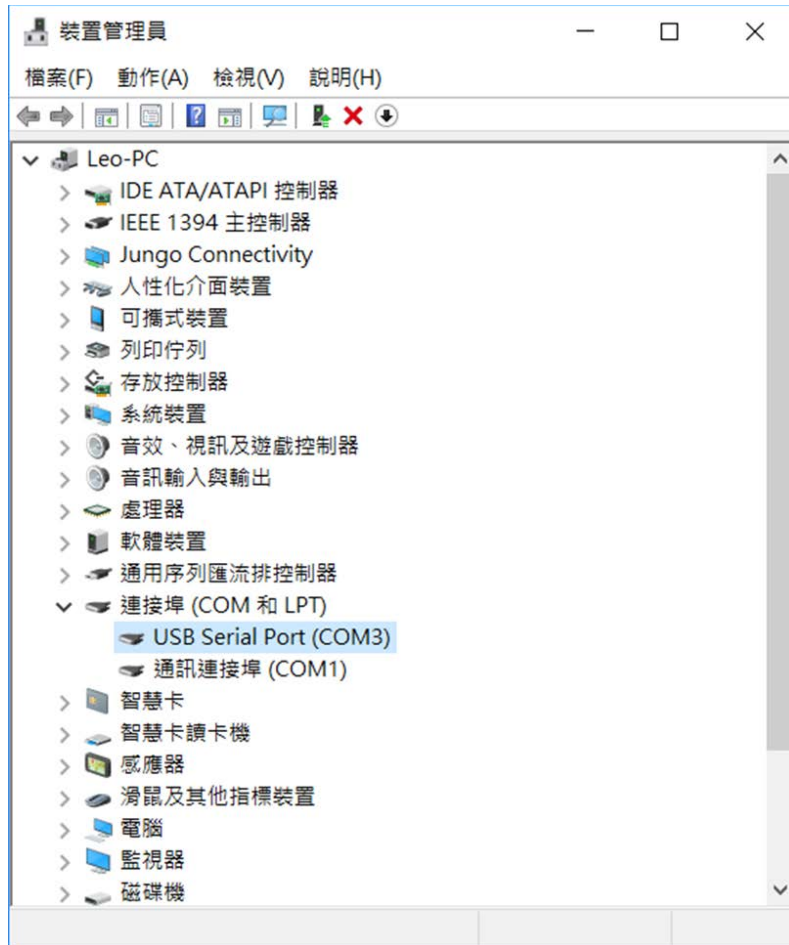


圖 2、Arduino USB Serial Port

第三步、開啟 Arduino IDE

開啟 Arduino IDE，就在第二步中解壓縮的資料夾當中，檔名為 arduino.exe。打開以後會見到圖 3。而白色的區塊就是程式編輯區。

接著得點選工具→板子，如圖 4，先選擇 Arduino 電路板版本，本教材是使用 Arduino Nano，所以就選這個！如果選錯，就無法將程式上傳到 Arduino 板子上。接著就選擇第二步中的序列埠為 com3，如圖 5。

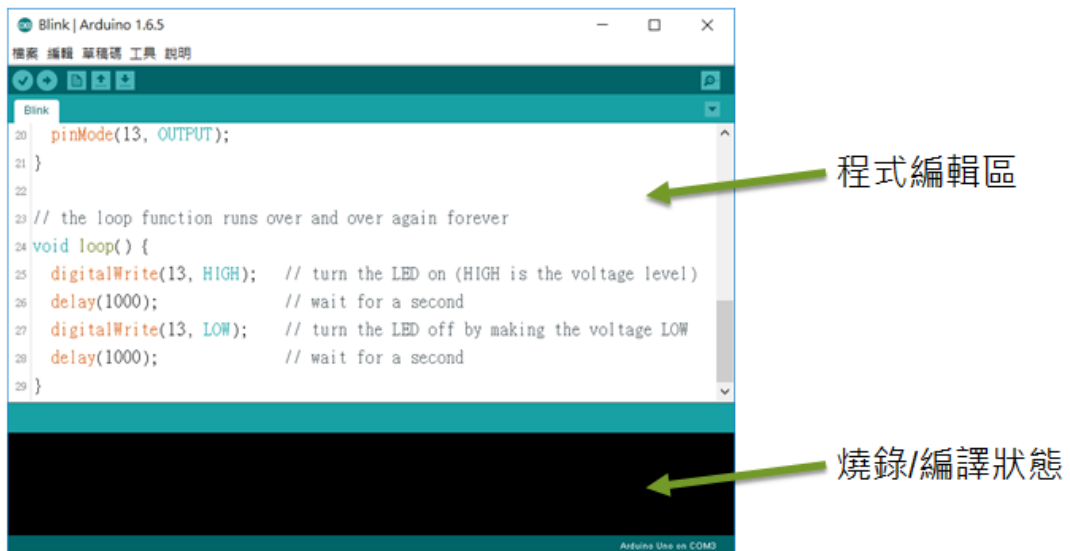



圖 3、Arduino IDE

圖 4、選擇 Arduino Nano 開發板



圖 5、選擇序列埠

第四步、開啟 Blink 範例

接下來就可以開始寫程式了。基本上 Arduino 板子上，一定會有一個小小的 LED 燈，而這 LED 燈是連接到 D13 的腳位上，如圖 6。接著至 IDE 界面上方的 檔案-範例-01.Basics-Blink 開啟範例檔案。按下上傳  即可燒錄程式碼至 Nano

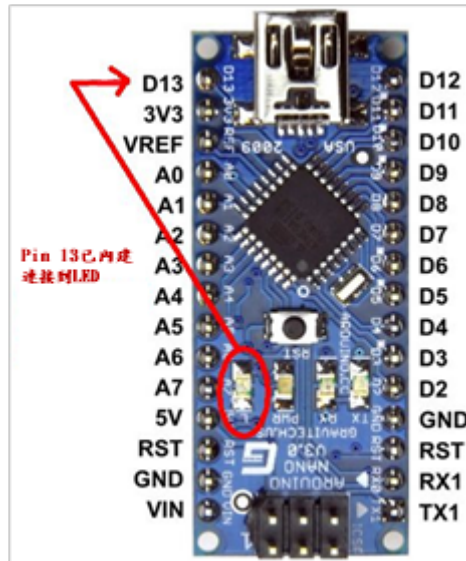


圖 6、D13 已內建連接到開發板 LED 上

Blink 程式範例就是控制 led 開關，程式說明如下：

```
// 先宣告 Pin 腳位置
void setup() {
  pinMode(13, OUTPUT);    // 宣告該 D13 是 output
}
void loop() {
  digitalWrite(13, HIGH); // 寫 HIGH 進去就是讓 LED 亮
  delay(1000);           // 延遲 1 秒就是讓 LED 亮 1 秒
  digitalWrite(13, LOW); // 寫 LOW 進去就是讓 LED 滅
  delay(1000);          // 延遲 1 秒就是讓 LED 滅 1 秒
}
```

基本上，Arduino 程式的架構基本上分成三部份，第一部份是 `setup()` 函數之前的全域變數宣告區，此區通常會匯入(`#include`)需用到的程式庫、宣告全域變數、或定義文數值常數。第二部份是 `setup()` 函數本身，`setup()` 函數在 Arduino 板子啟動後只會被呼叫一次，因此，`setup()` 內的程式碼通常是與應用程式初始化有關的程式敘述，例如各個 IO 腳位工作模式是輸出或輸入、串列通訊埠速率是多少 bps 等。當 Arduino 板子重新送電，或有電的情況下按下重置(Reset)按鈕，都會令 Arduino 板子重新啟動，所有程式狀態將被重置，換言之，`setup()` 函數會被呼叫而執行一次。呼叫並執行完 `setup()` 函數後，Arduino 會呼叫 `loop()` 函數，這就是程式執行架構的第三部份，這部份是 Arduino 程式的主要控制邏輯與資料處理的主體，由於微控制器的任務通常需周而復始的監視環境的變化並即時地採取對應的控制輸出，因此，`loop()` 函數也會周而復始地被呼叫，所以應用領域中主要的控制邏輯與資料處理會寫在 `loop()` 函數中。

(2)、溫溼度感測器 DHT11 教學

本單元將介紹如何使用 Arduino Nano 控制 DHT11 並透過程式讀取溫溼度值。

(2.1)、硬體接線圖

圖 7 為硬體接線圖，顯示結果由 OLED 顯示器顯示結果

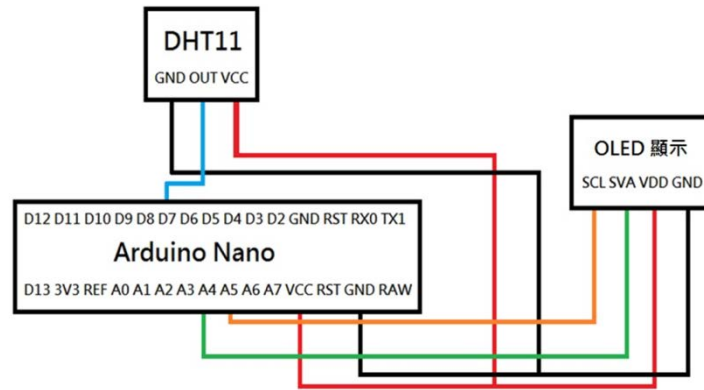


圖 7、硬體接線圖

(2.2)、程式部分

```
//sua=A4, vud=A5
#include "U8glib.h"
#include <string.h>
U8GLIB_KSM110_128X64_SPI u8g(U8G_I2C_OPT_NONE);
//Pin definition
#include <DHT11.h>
int pin=7;
DHT11 dht11(pin);
char sz[7],sc[7];
float f; //溫度

void draw_inital(void)
{
    // flip screen, if required
    //u8g.setRot180();
    // assign default color value
    if ( u8g.getMode() == U8G_MODE_R3G3B2 )
        u8g.setColorIndex(255);    // white
    else if ( u8g.getMode() == U8G_MODE_GRAY2BIT )
        u8g.setColorIndex(3);    // max intensity
    else if ( u8g.getMode() == U8G_MODE_BW )
        u8g.setColorIndex(1);    // pixel on
    // graphic commands to redraw the complete screen should be placed
    here
    // u8g.setFont(u8g_font_unifont);
    u8g.setFont(u8g_font_9x18);
    u8g.setFontPosTop();
}
```

```

void pf_Txt(byte num, const char *fmt, ... )
{
    char tmp[128]; // max is 128 chars
    va_list args;
    va_start (args, fmt );
    vsnprintf(tmp, 128, fmt, args);
    va_end (args);
    if (num==1)
        u8g.drawStr(10, 1, tmp);
    else if (num==2)
        u8g.drawStr(10, 24, tmp);
    else if (num==3)
        u8g.drawStr(10, 47, tmp);
}

void setup(void)
{
    //OLED initial
    draw_inital();
    u8g.firstPage();
    do {
        pf_Txt(2,"Waiting ....." );
    } while( u8g.nextPage() );
    delay(1000);
    Serial.begin(9600); //opens serial port, sets data rate to 9600 bps
}

void loop(void)
{
    int err;
    float temp, humi;
    if((err=dht11.read(humi, temp))==0)
    {
        Serial.print("temperature:");
        Serial.print(temp);
        Serial.print(" humidity:");
        Serial.print(humi);
        Serial.println();
    }
    else
    {
        Serial.println();
        Serial.print("Error No :");
        Serial.print(err);
        Serial.println();
    }
    dtostrf(temp, 6, 2, sz); //小數改成字元陣列
    dtostrf(humi, 6, 2, sc); //小數改成字元陣列
    u8g.firstPage();
    do {

```

```
    pf_Txt(1, "  DHT11");    //第一行顯示字串
    pf_Txt(2, "H:%3s %%", sc );    //第二行顯示小數
    pf_Txt(3, "T:%3s C", sz ); //第三行顯示整數
  } while( u8g.nextPage() );
  delay(1000);
}
```

(3)、PM2.5 感測器 PMS3003 教學

本單元將介紹如何使用 Arduino Nano 控制 PMS3003 並透過程式讀取 PM2.5 值。

(3.1)、硬體接線圖

圖 8 為硬體接線圖，顯示結果由 OLED 顯示器顯示結果

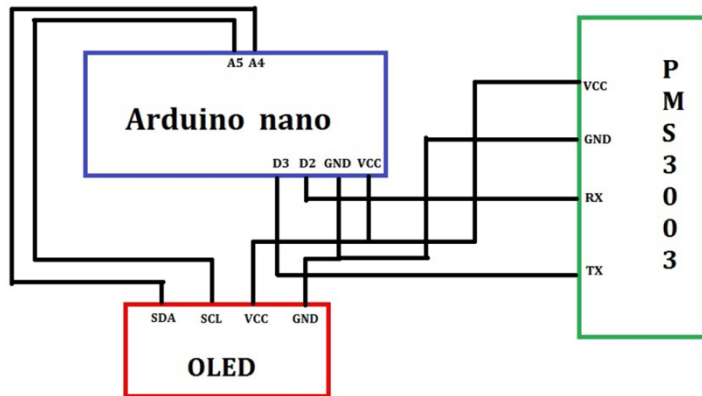


圖 8、硬體接線圖

(3.2)、程式部分

```

#include <SoftwareSerial.h>
//模擬 UART，UNO 是用 2 3，其它 ARDUINO 板請自行設定相關腳位
#define RxD 2
#define TxD 3
//OLED examples
#include "U8glib.h"
#include <string.h>
U8GLIB_KSM110_128X64_SPI u8g(U8G_I2C_OPT_NONE);
//Pin definition
char sz[5];
int Sec=0, Mini=0, Hour=0;
float Temp; //溫度
void draw_inital(void) {
    // flip screen, if required
    //u8g.setRot180();
    // assign default color value
    if ( u8g.getMode() == U8G_MODE_R3G3B2 )
        u8g.setColorIndex(255);    // white
    else if ( u8g.getMode() == U8G_MODE_GRAY2BIT )
        u8g.setColorIndex(3);      // max intensity
    else if ( u8g.getMode() == U8G_MODE_BW )
        u8g.setColorIndex(1);      // pixel on
    // graphic commands to redraw the complete screen should be placed
here
    // u8g.setFont(u8g_font_unifont);
    u8g.setFont(u8g_font_9x18);
    u8g.setFontPosTop();
}
void pf_Txt(byte num, const char *fmt, ... ){
    char tmp[128]; // max is 128 chars
    va_list args;
    va_start (args, fmt );
    vsnprintf(tmp, 128, fmt, args);
    va_end (args);
    if (num==1)

```

```

        u8g.drawStr(0, 1, tmp);
    else if (num==2)
        u8g.drawStr(0, 29, tmp);
    else if (num==3)
        u8g.drawStr(0, 45, tmp);
}
SoftwareSerial Serial1(RxD,TxD);
long pmcf10=0;
long pmcf25=0;
long pmcf100=0;
long pmat10=0;
long pmat25=0;
long pmat100=0;
char buf[50];
void setup() {
    // put your setup code here, to run once:
    {
        Serial.begin(9600);
        Serial1.begin(9600);
    }
    {
        float f;
        //OLED initial
        draw_inital();
        u8g.firstPage();
        do {
            pf_Txt(2,"Waiting ....." );
            } while( u8g.nextPage() );
        delay(1000);
        Temp=25.65;
        dtostrf(Temp, 6, 2, sz); //小數 Temp 改成 字元陣列 sz
        u8g.firstPage();
        do {
            pf_Txt(1,"PM Measure"); //第一行顯示字串
            pf_Txt(2,"2.5:%3dug/m3",pmcf25); //第二行顯示小數
            pf_Txt(3, "10:%3dug/m3",pmcf100); ////第三行顯示整數
            } while( u8g.nextPage() );
        Serial.begin(9600); //opens serial port, sets data rate to 9600 bps
    }
}
void loop() {
    int count = 0;
    unsigned char c;
    unsigned char high;
    while (Serial1.available()) {
        c = Serial1.read();
        if((count==0 && c!=0x42) || (count==1 && c!=0x4d)){
            Serial.println("check failed");
            break;
        }
        if(count > 15){

```



```

        Serial.println("complete");
        break;
    }
    else if(count == 4 || count == 6 || count == 8 || count == 10 ||
count == 12 || count == 14) {
        high = c;
    }
    else if(count == 5){
        pmcf10 = 256*high + c;
        Serial.print("CF=1, PM1.0=");
        Serial.print(pmcf10);
        Serial.println(" ug/m3");
    }
    else if(count == 7){
        pmcf25 = 256*high + c;
        Serial.print("CF=1, PM2.5=");
        Serial.print(pmcf25);
        Serial.println(" ug/m3");
    }
    else if(count == 9){
        pmcf100 = 256*high + c;
        Serial.print("CF=1, PM10=");
        Serial.print(pmcf100);
        Serial.println(" ug/m3");
    }
    else if(count == 11){
        pmat10 = 256*high + c;
        Serial.print("atmosphere, PM1.0=");
        Serial.print(pmat10);
        Serial.println(" ug/m3");
    }
    else if(count == 13){
        pmat25 = 256*high + c;
        Serial.print("atmosphere, PM2.5=");
        Serial.print(pmat25);
        Serial.println(" ug/m3");
    }
    else if(count == 15){
        pmat100 = 256*high + c;
        Serial.print("atmosphere, PM10=");
        Serial.print(pmat100);
        Serial.println(" ug/m3");
    }
    count++;
}
while(Serial1.available()) Serial1.read();
Serial.println();
delay(2000);
{
    Sec++;
    if (Sec>=60)

```

```

{
  Sec=0;
  Mini++;
  if (Mini>=60)
  {
    Mini=0;
    Hour++;
  }
  if ( Hour>=24)
    Hour=0;
}
Temp=25.65;
dtostrf(Temp, 6, 2, sz); //小數改成字元陣列
u8g.firstPage();
do {
  pf_Txt(1,"PM Measure"); //第一行顯示字串
  pf_Txt(2,"2.5:%3dug/m3",pmcf25 ); //第二行顯示小數
  pf_Txt(3, "10:%3dug/m3",pmcf100); ////第三行顯示整數
} while( u8g.nextPage() );
delay(1000);
}
}

```

參考文獻：

1. <http://www.etechpk.net/shop/arduino/arduino-nano-v3/>
2. <http://blog.kenyang.net/2012/07/28/arduino>
3. 微電腦原理與應用－Arduino，全華圖書，作者：黃新賢、劉建源、林宜賢、黃志峰。