

智慧家庭：PM2.5 空氣感測器（電路設計下篇）

四月，2016

文\曹永忠

本篇是接續上篇文章『智慧家庭：PM2.5 空氣感測器（電路設計上篇）』（曹永忠，2016c，2016d，2016e，2016f；曹永忠，許智誠，& 蔡英德，2015a，2015b），延續未完成的硬體電路組裝下篇，主要是教大家如何組立空氣粒子感測裝置電子電路組裝。

上文中我們介紹空氣粒子感測裝置的開發板安裝、空氣懸浮粒子感測器安裝，麵包板安裝、溫溼度模組安裝、RTC 時鐘模組安裝…等等(曹永忠，2016f)，本篇將介紹空氣懸浮粒子感測器、LCD 2004 顯示模組等電路安裝，並進行第一階段的整合測試。

安裝 RTC 時鐘模組

我們加入 RTC 時鐘模組，如下圖所示進行電路連接。

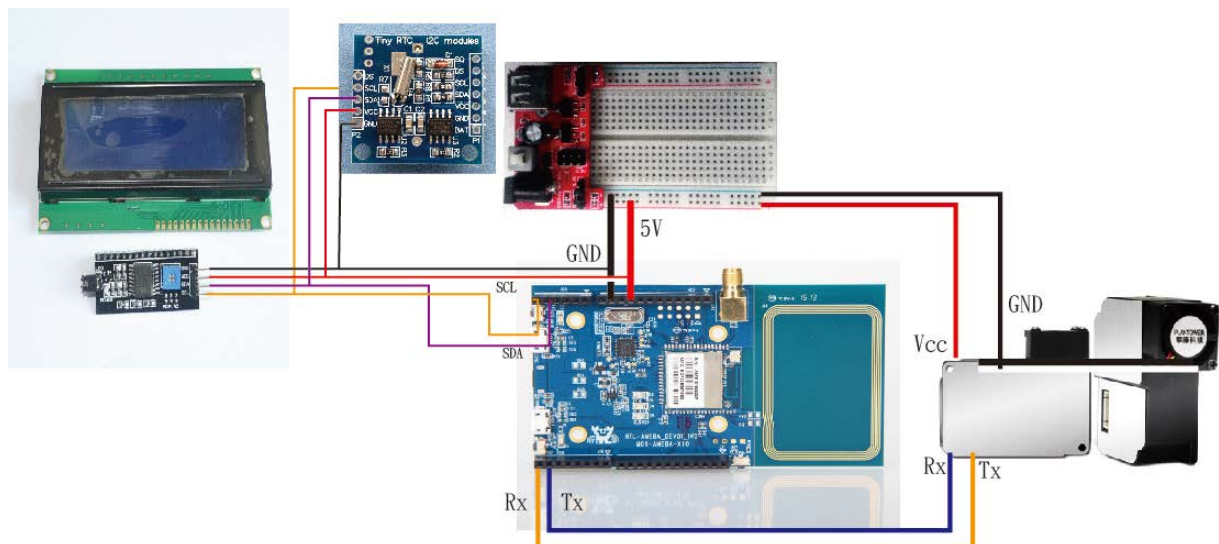


圖 1 安裝 RTC 時鐘模組

由於時間因素對本設計是一個非常重要的因素，由於阿米巴開發板並沒有內置時間模組，所以我們加入 RTC 時鐘模組。所以增加下表之接腳表，讓讀者更加了解(曹永忠, 2016b; 曹永忠 et al., 2015a, 2015b)。

表 1 RTC 時鐘模組接腳表(累加接腳表)

PMS3003 感測模組	開發板接腳	解說
pin1, 供電輸入接腳	Ameba pin 5V	5V 陽極接點
pin2, 接地接腳	Ameba pin Gnd	共地接點
pin4, 串列埠接收接腳(Rx)	Ameba Tx(D1)	串列埠傳送端
pin5, 串列埠輸出接腳(Tx)	Ameba Rx(D0)	串列埠接收端
LCD2004 顯示模組	開發板接腳	解說
VCC	Ameba pin 5V	5V 陽極接點
GND	Ameba pin Gnd	共地接點
SCL	Ameba I2C_SCL	I2C SCL
SDA	Ameba I2C_SDA	I2C SDA
RTC 時鐘模組	開發板接腳	解說
VCC	Ameba pin 5V	5V 陽極接點
GND	Ameba pin Gnd	共地接點
SCL	Ameba I2C_SCL	I2C SCL
SDA	Ameba I2C_SDA	I2C SDA

我們將下表之 PMS3003 空氣懸浮粒子感測器測試程式一攥寫好之後，編譯完

成後上傳到 Ameba 開發板，

表 2 PMS3003 空氣懸浮粒子感測器測試程式一

PMS3003 空氣懸浮粒子感測器測試程式一(PMS3003AirQualityV51A)

```
/*
```

This example demonstrate how to read pm2.5 value on PMS 3003 air condition sensor

PMS 3003 pin map is as follow:

PIN1 :VCC, connect to 5V

PIN2 :GND

PIN3 :SET, 0:Standby mode, 1:operating mode

PIN4 :RXD :Serial RX

PIN5 :TXD :Serial TX

PIN6 :RESET

PIN7 :NC

PIN8 :NC

In this example, we only use Serial to get PM 2.5 value.

The circuit:

* RX is digital pin 0 (connect to TX of PMS 3003)

* TX is digital pin 1 (connect to RX of PMS 3003)

```
*/
```

```
#define turnon HIGH
```

```
#define turnoff LOW
```

```
#include <WiFi.h>
```

```
#include "PMTYPE.h"
```

```
#include <Wire.h> // Arduino IDE 內建
```

```
// LCD I2C Library，從這裡可以下載：
```

```
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
```

```
#include <LiquidCrystal_I2C.h>
```

```
#include <SoftwareSerial.h>
```

PMS3003 空氣懸浮粒子感測器測試程式一(PMS3003AirQualityV51A)

```
#include "RTClib.h"
RTC_DS1307 RTC;
//DateTime nowT = RTC.now();

uint8_t MacData[6];

SoftwareSerial mySerial(0, 1); // RX, TX
IPAddress Meip ,Megateway ,Mesubnet ;
String MacAddress ;
int status = WL_IDLE_STATUS;
boolean ParticleSensorStatus = true ;
#define pmsDataLen 32
uint8_t buf[pmsDataLen];
int idx = 0;
int pm25 = 0;
uint16_t PM01Value=0;           //define PM1.0 value of the air
detector module
uint16_t PM2_5Value=0;          //define PM2.5 value of the air
detector module
uint16_t PM10Value=0;           //define PM10 value of the air
detector module
int NDPyear, NDPmonth, NDPday, NDPhour, NDPminute, NDPsecond;
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //
設定 LCD I2C 位址

void setup() {
  Serial.begin(9600);

  mySerial.begin(9600); // PMS 3003 UART has baud rate 9600
  lcd.begin(20, 4);      // 初始化 LCD，一行 20 的字元，共 4 行，
  預設開啟背光
  lcd.backlight(); // 開啟背光
  // while(!Serial) ;
  WiFi.status(); //this method must be used for get MAC
  MacAddress = GetWifiMac() ;
  ShowMac() ;
  ShowDateTime() ;
```

```
}

void loop() { // run over and over
    idx = 0;
    memset(buf, 0, pmsDataLen);

    while (mySerial.available())
    {
        buf[idx++] = mySerial.read();
    }

    // check if data header is correct
    if (buf[0] == 0x42 && buf[1] == 0x4d)
    {
        pm25 = ( buf[12] << 8 ) | buf[13];
        Serial.print("pm2.5: ");
        Serial.print(pm25);
        Serial.println(" ug/m3");
        ShowPM(pm25) ;
    }
}

void ShowMac()
{
    lcd.setCursor(0, 0); // 設定游標位置在第一行行首
    lcd.print("MAC:");
    lcd.print(MacAddress);
}

void ShowDateTime()
{
    lcd.setCursor(0, 2); // 設定游標位置在第一行行首
    lcd.print(StrDate());
    lcd.setCursor(11, 2); // 設定游標位置在第一行行首
    lcd.print(StrTime());
}
```

PMS3003 空氣懸浮粒子感測器測試程式一(PMS3003AirQualityV51A)

```
//  lcd.print();

}

String StrDate() {
    String ttt ;
    //nowT  = now;
    DateTime now = RTC.now();
    ttt = print4digits(now.year()) + "-" + print2digits(now.month())
+ "-" + print2digits(now.day()) ;
    //ttt = print4digits(NDPyear) + "/" + print2digits(NDPmonth) + "/"
+ print2digits(NDPday) ;
    return ttt ;
}

String StringDate(int yyy,int mmm,int ddd) {
    String ttt ;
    //nowT  = now;
    ttt = print4digits(yyy) + "-" + print2digits(mmm) + "-" +
print2digits(ddd) ;
    return ttt ;
}

String StrTime() {
    String ttt ;
    // nowT  = RTC.now();
    DateTime now = RTC.now();
    ttt = print2digits(now.hour()) + ":" +
print2digits(now.minute()) + ":" + print2digits(now.second()) ;
    //  ttt = print2digits(NDPhour) + ":" + print2digits(NDPminute)
+ ":" + print2digits(NDPsecond) ;
    return ttt ;
}

String StringTime(int hhh,int mmm,int sss) {
    String ttt ;
```

PMS3003 空氣懸浮粒子感測器測試程式一(PMS3003AirQualityV51A)

```
    ttt = print2digits(hhh) + ":" + print2digits(mmm) + ":" +  
print2digits(sss) ;  
    return ttt ;  
}
```

```
String GetWifiMac()
```

```
{  
    String tt ;  
    String t1, t2, t3, t4, t5, t6 ;  
    WiFi.macAddress(MacData);  
  
    Serial.print("Mac:");  
    Serial.print(MacData[0], HEX) ;  
    Serial.print("/");  
    Serial.print(MacData[1], HEX) ;  
    Serial.print("/");  
    Serial.print(MacData[2], HEX) ;  
    Serial.print("/");  
    Serial.print(MacData[3], HEX) ;  
    Serial.print("/");  
    Serial.print(MacData[4], HEX) ;  
    Serial.print("/");  
    Serial.print(MacData[5], HEX) ;  
    Serial.print("~");  
  
    t1 = print2HEX((int)MacData[0]);  
    t2 = print2HEX((int)MacData[1]);  
    t3 = print2HEX((int)MacData[2]);  
    t4 = print2HEX((int)MacData[3]);  
    t5 = print2HEX((int)MacData[4]);  
    t6 = print2HEX((int)MacData[5]);  
    tt = (t1+t2+t3+t4+t5+t6) ;  
    Serial.print(tt);  
    Serial.print("\n");  
}
```

PMS3003 空氣懸浮粒子感測器測試程式一(PMS3003AirQualityV51A)

```
    return tt ;
}
String print2HEX(int number) {
    String ttt ;
    if (number >= 0 && number < 16)
    {
        ttt = String("0") + String(number, HEX);
    }
    else
    {
        ttt = String(number, HEX);
    }
    return ttt ;
}
String print2digits(int number) {
    String ttt ;
    if (number >= 0 && number < 10)
    {
        ttt = String("0") + String(number);
    }
    else
    {
        ttt = String(number);
    }
    return ttt ;
}

String print4digits(int number) {
    String ttt ;
    ttt = String(number);
    return ttt ;
}

void ShowPM(int pp25)
{
    lcd.setCursor(0, 3); // 設定游標位置在第一行行首
```


PMS3003 空氣懸浮粒子感測器測試程式一(PMS3003AirQualityV51A)

```
lcd.print(" PM2.5: ");  
lcd.setCursor(9, 3); // 設定游標位置在第一行行首  
lcd.print(pp25);  
  
}
```

資料下載：

<https://github.com/brucetsao/makerdiwo/tree/master/201604/PMS3003AirQualityV51A>

由於，我們可以連上網際網路，也可以讀取時鐘模組，所以我們可以顯示網路資訊與時間資訊。我們將下表之 PMS3003 空氣懸浮粒子感測器測試程式一攥寫好之後，編譯完成後上傳到 Ameba 開發板，

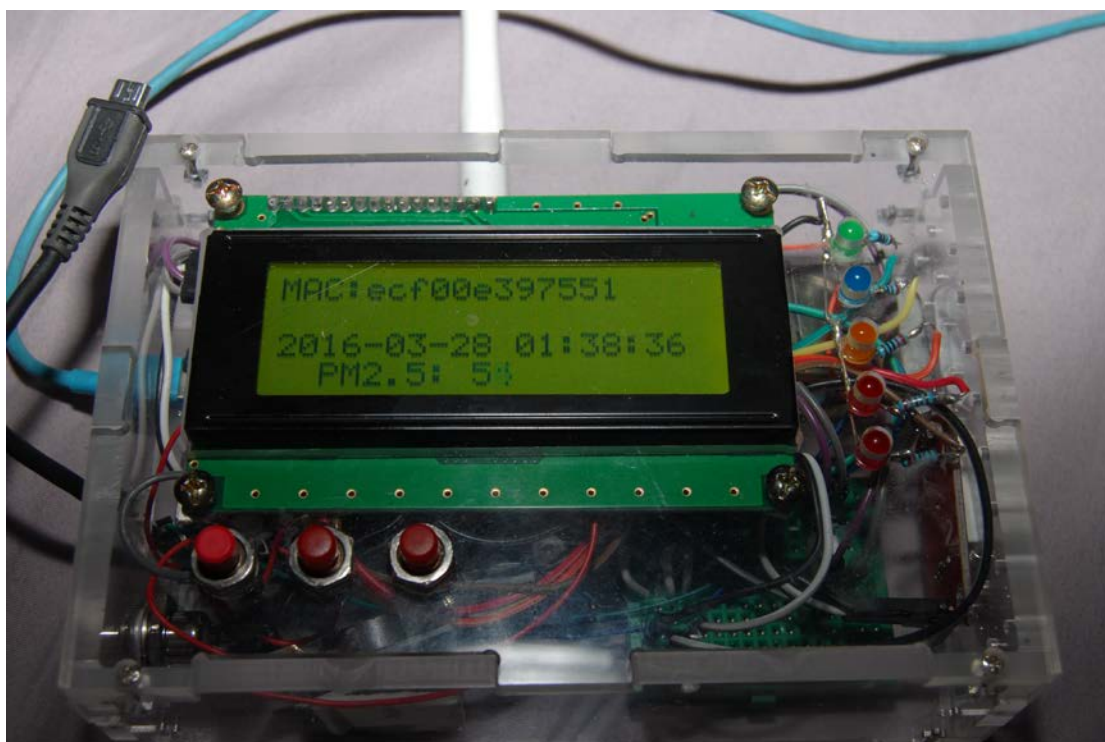


圖 2 PMS3003 空氣懸浮粒子感測器測試程式一畫面結果

擴充 RTC 時鐘模組網路校時能力

由於 RTC 時鐘模組是一個可以信賴的即時時鐘模組，但是如果新安裝裝置、更換電池或地區變更等等，都必須要將空氣粒子感測裝置帶回原開發者的研究室方可以更正於 RTC 時鐘模組的時間，雖然 Ameba 開發版有強大的無線網路連接網

際網路的能力，但是在六秒中，除了重新讀取空氣粒子感測裝置的資料，還必須完成許多其他的工作，這些都必須耗費 Ameba 開發版的時間，與無線網路連接網際網路取得時間的成本，這樣對一個完善的空氣粒子感測裝置，太耗費在無線網路連接網際網路取得時間的成本。

所以我們如果能將系統修正，在每一次空氣粒子感測裝置開機後，即使用無線網路連接網際網路取得時間，並動態校正 RTC 時鐘模組，往後的時間就完全依靠內部的 RTC 時鐘模組運作(曹永忠, 2016a, 2016g; 曹永忠 et al., 2015a, 2015b)，如此空氣粒子感測裝置可以更加完備，所以我們修改上述程式來達到此功能。

我們將下表之整合空氣懸浮粒子感測器測試程式二攥寫好之後，編譯完成後上傳到 Ameba 開發板，

表 3 整合空氣懸浮粒子感測器測試程式二

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)
<pre>/* This example demonstrate how to read pm2.5 value on PMS 3003 air condition sensor PMS 3003 pin map is as follow: PIN1 :VCC, connect to 5V PIN2 :GND PIN3 :SET, 0:Standby mode, 1:operating mode PIN4 :RXD :Serial RX PIN5 :TXD :Serial TX PIN6 :RESET PIN7 :NC PIN8 :NC In this example, we only use Serial to get PM 2.5 value. The circuit: * RX is digital pin 0 (connect to TX of PMS 3003) * TX is digital pin 1 (connect to RX of PMS 3003) */ #include <math.h></pre>

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
#define turnon HIGH
#define turnoff LOW

#include "PMTYPE.h"
#include <WiFi.h>
#include <WiFiUdp.h>
#include <Wire.h> // Arduino IDE 內建
// LCD I2C Library，從這裡可以下載：
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads

#include "RTClib.h"
RTC_DS1307 RTC;
//DateTime nowT = RTC.now();

#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

uint8_t MacData[6];

SoftwareSerial mySerial(0, 1); // RX, TX
char ssid[] = "TSA0"; // your network SSID (name)
char pass[] = "TSA01234"; // your network password
#define MAX_CLIENT_ID_LEN 10
#define MAX_TOPIC_LEN 50
char clientId[MAX_CLIENT_ID_LEN];
char outTopic[MAX_TOPIC_LEN];

IPAddress Meip ,Megateway ,Mesubnet ;
String MacAddress ;
int status = WL_IDLE_STATUS;
boolean ParticleSensorStatus = true ;
WiFiUDP Udp;

const char ntpServer[] = "pool.ntp.org";
const long timeZoneOffset = 28800L;
const int NTP_PACKET_SIZE = 48; // NTP time stamp is in the first 48
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

bytes of the message

```
const byte nptSendPacket[ NTP_PACKET_SIZE] = {
    0xE3, 0x00, 0x06, 0xEC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x31, 0x4E, 0x31, 0x34,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00
};
byte ntpRecvBuffer[ NTP_PACKET_SIZE ];

#define LEAP_YEAR(Y)      ( ((1970+Y)>0) && !((1970+Y)%4) &&
( ((1970+Y)%100) || !((1970+Y)%400) ) )
static const uint8_t
monthDays[]={31,28,31,30,31,30,31,31,30,31,30,31}; // API starts months
from 1, this array starts from 0
uint32_t epochSystem = 0; // timestamp of system boot up

#define pmsDataLen 32
uint8_t buf[pmsDataLen];
int idx = 0;
int pm25 = 0;
uint16_t PM2_5Value=0;          //define PM2.5 value of the air
detector module
int NDPyear, NDPmonth, NDPday, NDPhour, NDPminute, NDPsecond;
unsigned long epoch ;
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //
設定 LCD I2C 位址

void setup() {
    Serial.begin(9600);
    mySerial.begin(9600); // PMS 3003 UART has baud rate 9600
    lcd.begin(20, 4);      // 初始化 LCD，一行 20 的字元，共 4 行，
    預設開啟背光
    lcd.backlight(); // 開啟背光
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
    MacAddress = GetWifiMac() ;
    ShowMac() ;
    initializeWiFi();

    initRTC() ;
    ShowDateTime() ;
    delay(1500);
}

void loop() { // run over and over
    ShowDateTime() ;
    retrievePM25Value() ;
    delay(1000); // delay 1 minute for next measurement

}

void ShowMac()
{
    lcd.setCursor(0, 0); // 設定游標位置在第一行行首
    lcd.print("MAC:");
    lcd.print(MacAddress);

}

void ShowInternetStatus()
{
    lcd.setCursor(0, 1); // 設定游標位置
    if (WiFi.status())
    {
        Meip = WiFi.localIP();
        lcd.print("@:");
        lcd.print(Meip);

    }
    else
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
        {
            lcd.print("DisConnected:");

        }

    }

    void ShowPM25(int pp25)
    {
        lcd.setCursor(0, 3); // 設定游標位置在第一行行首
        lcd.print("PM2.5:    ");
        lcd.setCursor(9, 3); // 設定游標位置在第一行行首
        lcd.print(pp25);

    }


    void ShowDateTime()
    {
        // getCurrentTime(epoch, &NDPyear, &NDPmonth, &NDPday, &NDPhour,
        &NDPminute, &NDPsecond);
        lcd.setCursor(0, 2); // 設定游標位置在第一行行首
        lcd.print(StrDate());
        lcd.setCursor(11, 2); // 設定游標位置在第一行行首
        lcd.print(StrTime());
        // lcd.print();

    }


    String StrDate() {
        String ttt ;
        //nowT = now;
        DateTime now = RTC.now();
        ttt = print4digits(now.year()) + "-" + print2digits(now.month()) +
        "-" + print2digits(now.day()) ;
        //ttt = print4digits(NDPyear) + "/" + print2digits(NDPmonth) + "/"
        + print2digits(NDPday) ;
    }
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
        return ttt ;
    }

    String StringDate(int yyy,int mmm,int ddd) {
        String ttt ;
        //nowT = now;
        ttt = print4digits(yyy) + "-" + print2digits(mmm) + "-" +
print2digits(ddd) ;
        return ttt ;
    }

    String StrTime() {
        String ttt ;
        // nowT = RTC.now();
        DateTime now = RTC.now();
        ttt = print2digits(now.hour()) + ":" + print2digits(now.minute())
+ ":" + print2digits(now.second()) ;
        // ttt = print2digits(NDPhour) + ":" + print2digits(NDPminute) +
":" + print2digits(NDPsecond) ;
        return ttt ;
    }

    String StringTime(int hhh,int mmm,int sss) {
        String ttt ;
        ttt = print2digits(hhh) + ":" + print2digits(mmm) + ":" +
print2digits(sss) ;
        return ttt ;
    }

    String GetWifiMac()
    {
        String tt ;
        String t1, t2, t3, t4, t5, t6 ;
        WiFi.status();    //this method must be used for get MAC
        WiFi.macAddress(MacData);
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
Serial.print("Mac:");
Serial.print(MacData[0], HEX) ;
Serial.print("/");
Serial.print(MacData[1], HEX) ;
Serial.print("/");
Serial.print(MacData[2], HEX) ;
Serial.print("/");
Serial.print(MacData[3], HEX) ;
Serial.print("/");
Serial.print(MacData[4], HEX) ;
Serial.print("/");
Serial.print(MacData[5], HEX) ;
Serial.print("~");

t1 = print2HEX((int)MacData[0]);
t2 = print2HEX((int)MacData[1]);
t3 = print2HEX((int)MacData[2]);
t4 = print2HEX((int)MacData[3]);
t5 = print2HEX((int)MacData[4]);
t6 = print2HEX((int)MacData[5]);
tt = (t1+t2+t3+t4+t5+t6) ;
Serial.print(tt);
Serial.print("\n");

return tt ;
}
String print2HEX(int number) {
String ttt ;
if (number >= 0 && number < 16)
{
    ttt = String("0") + String(number, HEX);
}
else
{
    ttt = String(number, HEX);
}
```


整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
    return ttt ;
}

String print2digits(int number) {
    String ttt ;
    if (number >= 0 && number < 10)
    {
        ttt = String("0") + String(number);
    }
    else
    {
        ttt = String(number);
    }
    return ttt ;
}

String print4digits(int number) {
    String ttt ;
    ttt = String(number);
    return ttt ;
}

// send an NTP request to the time server at the given address
void retrieveNtpTime() {
    Serial.println("Send NTP packet");

    Udp.beginPacket(ntpServer, 123); //NTP requests are to port 123
    Udp.write(nptSendPacket, NTP_PACKET_SIZE);
    Udp.endPacket();

    if(Udp.parsePacket()) {
        Serial.println("NTP packet received");
        Udp.read(ntpRecvBuffer, NTP_PACKET_SIZE); // read the packet
into the buffer

        unsigned long highWord = word(ntpRecvBuffer[40],
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
ntpRecvBuffer[41]);
    unsigned long lowWord = word(ntpRecvBuffer[42],
ntpRecvBuffer[43]);
    unsigned long secsSince1900 = highWord << 16 | lowWord;
    const unsigned long seventyYears = 2208988800UL;
    //      epoch = secsSince1900 - seventyYears + timeZoneOffset ;
    epoch = secsSince1900 - seventyYears ;

    epochSystem = epoch - millis() / 1000;
}
}

void getCurrentTime(unsigned long epoch, int *year, int *month, int
*day, int *hour, int *minute, int *second) {
    int tempDay = 0;

    *hour = (epoch % 86400L) / 3600;
    *minute = (epoch % 3600) / 60;
    *second = epoch % 60;

    *year = 1970;
    *month = 0;
    *day = epoch / 86400;

    for (*year = 1970; ; (*year)++) {
        if (tempDay + (LEAP_YEAR(*year) ? 366 : 365) > *day) {
            break;
        } else {
            tempDay += (LEAP_YEAR(*year) ? 366 : 365);
        }
    }
    tempDay = *day - tempDay; // the days left in a year
    for ((*month) = 0; (*month) < 12; (*month)++) {
        if ((*month) == 1) {
            if (LEAP_YEAR(*year)) {
                if (tempDay - 29 < 0) {
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
        break;
    } else {
        tempDay -= 29;
    }
} else {
    if (tempDay - 28 < 0) {
        break;
    } else {
        tempDay -= 28;
    }
}
} else {
    if (tempDay - monthDays[*month] < 0) {
        break;
    } else {
        tempDay -= monthDays[*month];
    }
}
}
(*month)++;
*day = tempDay+2; // one for base 1, one for current day
}
```

```
void retrievePM25Value() {
    int idx;
    bool hasPm25Value = false;
    int timeout = 200;
    while (!hasPm25Value) {
        idx = 0;
        memset(buf, 0, pmsDataLen);
        while (mySerial.available()) {
            buf[idx++] = mySerial.read();
        }

        if (buf[0] == 0x42 && buf[1] == 0x4d) {
            pm25 = ( buf[12] << 8 ) | buf[13];
            Serial.print("pm2.5: ");
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
        Serial.print(pm25);
        Serial.print(" ug/m3");
        Serial.println("");
        hasPm25Value = true;
        ShowPM25(pm25) ;
    }
    timeout--;
    if (timeout < 0) {
        Serial.println("fail to get pm2.5 data");
        break;
    }
}

void initializeWiFi() {
    while (status != WL_CONNECTED) {
        Serial.print("Attempting to connect to SSID: ");
        Serial.println(ssid);
        // Connect to WPA/WPA2 network. Change this line if using open
or WEP network:
        status = WiFi.begin(ssid, pass);
        // status = WiFi.begin(ssid);

        // wait 10 seconds for connection:
        delay(10000);
    }

    // local port to listen for UDP packets
    Udp.begin(2390);
}

void printWifiData()
{
    // print your WiFi shield's IP address:
    Meip = WiFi.localIP();
    Serial.print(" IP Address: ");
    Serial.println(Meip);
}
```

```
// print your MAC address:
byte mac[6];
WiFi.macAddress(mac);
Serial.print("MAC address: ");
Serial.print(mac[5], HEX);
Serial.print(":");
Serial.print(mac[4], HEX);
Serial.print(":");
Serial.print(mac[3], HEX);
Serial.print(":");
Serial.print(mac[2], HEX);
Serial.print(":");
Serial.print(mac[1], HEX);
Serial.print(":");
Serial.println(mac[0], HEX);

// print your subnet mask:
Mesubnet = WiFi.subnetMask();
Serial.print("NetMask: ");
Serial.println(Mesubnet);

// print your gateway address:
Megateway = WiFi.gatewayIP();
Serial.print("Gateway: ");
Serial.println(Megateway);
}

void initRTC()
{
    Wire.begin();
    RTC.begin();
    SetRTCFromNtpTime() ;
    if (! RTC.isrunning()) {
        Serial.println("RTC is NOT running!");
        // following line sets the RTC to the date & time this sketch was
```

整合空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV71A)

```
compiled
//    RTC.adjust(DateTime(__DATE__, __TIME__));

    }
}
void SetRTCFromNtpTime()
{
    retrieveNtpTime();
    //DateTime ttt;
    getCurrentTime(epoch+timeZoneOffset, &NDPyear, &NDPmonth,
&NDPday, &NDPhour, &NDPminute, &NDPsecond);
    //ttt->year = NDPyear ;
    Serial.print("NDP Date is :");
    Serial.print(StringDate(NDPyear, NDPmonth, NDPday));
    Serial.print("and ");
    Serial.print("NDP Time is :");
    Serial.print(StringTime(NDPhour, NDPminute, NDPsecond));
    Serial.print("\n");

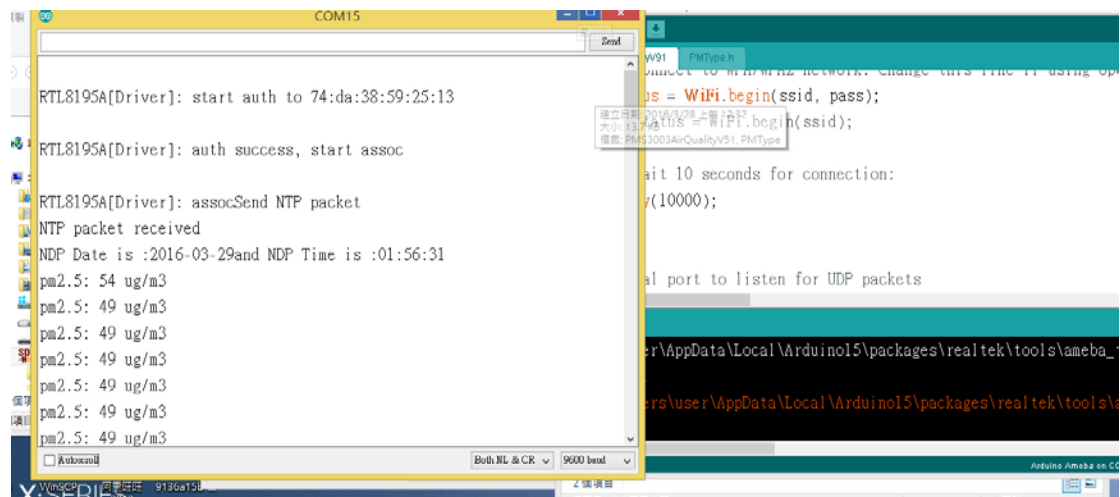
    RTC.adjust(DateTime(epoch+timeZoneOffset));

}
```

資料下載：

[https://github.com/brucetsao/makerdiwo/tree/master/201604/
PMS3003AirQualityV71A](https://github.com/brucetsao/makerdiwo/tree/master/201604/PMS3003AirQualityV71A)

由於空氣粒子感測裝置每次開機時，我們可以連上網際網路，使用無線網路連接網際網路取得時間，並動態校正 RTC 時鐘模組，往後的時間就完全依靠內部的 RTC 時鐘模組運作。



(a). 開發 IDE 監控畫面



(b). 校時之後空氣粒子感測裝置 LCD 顯示畫面

圖 3 整合空氣懸浮粒子感測器測試程式二畫面結果

擴充溫溼度感測器

由於完整的空氣粒子感測裝置除了偵測空氣中懸浮微粒的濃度，基本的溫溼

度資訊也是必需的需求，所以我們加入溫溼度感測器模組(曹永忠 et al., 2015a, 2015b; 曹永忠, 許智誠, & 蔡英德, 2015c)。

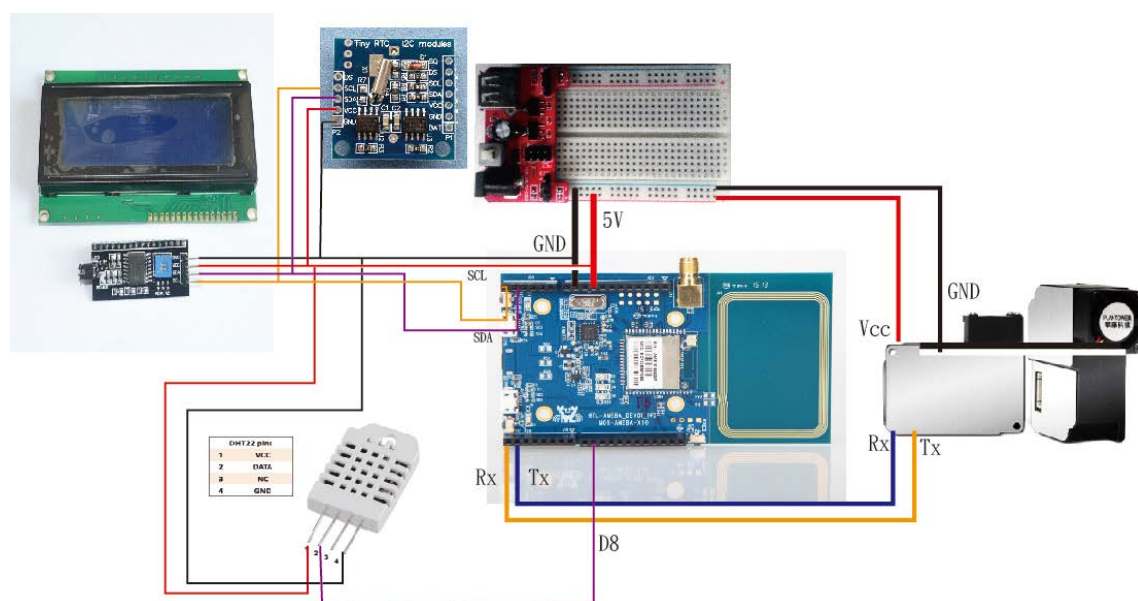


圖 4 安裝溫溼度感測器

請讀者依照下表之接腳表，進行電路組立。

表 4 溫溼度感測器接腳表(累加接腳表)

PMS3003 感測模組	開發板接腳	解說
pin1, 供電輸入接腳	Ameba pin 5V	5V 陽極接點
pin2, 接地接腳	Ameba pin Gnd	共地接點
pin4, 串列埠接收接腳(Rx)	Ameba Tx(D1)	串列埠傳送端
pin5, 串列埠輸出接腳(Tx)	Ameba Rx(D0)	串列埠接收端
LCD2004 顯示模組	開發板接腳	解說
VCC	Ameba pin 5V	5V 陽極接點
GND	Ameba pin Gnd	共地接點
SCL	Ameba I2C_SCL	I2C SCL
SDA	Ameba I2C_SDA	I2C SDA
RTC 時鐘模組	開發板接腳	解說
VCC	Ameba pin 5V	5V 陽極接點
GND	Ameba pin Gnd	共地接點
SCL	Ameba I2C_SCL	I2C SCL
SDA	Ameba I2C_SDA	I2C SDA

溫溼度感測器	開發板接腳	解說
VCC	Ameba pin 5V	5V 陽極接點
GND	Ameba pin Gnd	共地接點
Data(Signal)	Ameba D8	溫溼度資料輸出腳

我們將下表之整合空氣懸浮粒子感測器測試程式三撰寫好之後，編譯完成後上傳到 Ameba 開發板，

表 5 整合空氣懸浮粒子感測器測試程式三

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)
<pre> /* This example demonstrate how to read pm2.5 value on PMS 3003 air condition sensor PMS 3003 pin map is as follow: PIN1 :VCC, connect to 5V PIN2 :GND PIN3 :SET, 0:Standby mode, 1:operating mode PIN4 :RXD :Serial RX PIN5 :TXD :Serial TX PIN6 :RESET PIN7 :NC PIN8 :NC In this example, we only use Serial to get PM 2.5 value. The circuit: * RX is digital pin 0 (connect to TX of PMS 3003) * TX is digital pin 1 (connect to RX of PMS 3003) */ #include <math.h> #define turnon HIGH #define turnoff LOW #define DHTSensorPin 7 </pre>

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
#include "PMTYPE.h"
#include <WiFi.h>
#include <WiFiUdp.h>
#include <Wire.h> // Arduino IDE 內建
// LCD I2C Library，從這裡可以下載：
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads

#include "RTClib.h"
RTC_DS1307 RTC;
//DateTime nowT = RTC.now();
#include "DHT.h"
// Uncomment whatever type you're using!
// #define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

uint8_t MacData[6];

SoftwareSerial mySerial(0, 1); // RX, TX
char ssid[] = "TSA0"; // your network SSID (name)
char pass[] = "TSA01234"; // your network password
#define MAX_CLIENT_ID_LEN 10
#define MAX_TOPIC_LEN 50
char clientId[MAX_CLIENT_ID_LEN];
char outTopic[MAX_TOPIC_LEN];

IPAddress Meip ,Megateway ,Mesubnet ;
String MacAddress ;
int status = WL_IDLE_STATUS;
boolean ParticleSensorStatus = true ;
WiFiUDP Udp;

const char ntpServer[] = "pool.ntp.org";
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
const long timeZoneOffset = 28800L;
const int NTP_PACKET_SIZE = 48; // NTP time stamp is in the first 48
bytes of the message
const byte ntpSendPacket[ NTP_PACKET_SIZE] = {
    0xE3, 0x00, 0x06, 0xEC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x31, 0x4E, 0x31, 0x34,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00
};
byte ntpRecvBuffer[ NTP_PACKET_SIZE ];

#define LEAP_YEAR(Y)      ( ((1970+Y)>0) && !((1970+Y)%4) &&
( ((1970+Y)%100) || !((1970+Y)%400) ) )
static const uint8_t
monthDays[]={31,28,31,30,31,30,31,31,30,31,30,31}; // API starts months
from 1, this array starts from 0
uint32_t epochSystem = 0; // timestamp of system boot up

#define pmsDataLen 32
uint8_t buf[pmsDataLen];
int idx = 0;
int pm25 = 0;
uint16_t PM2_5Value=0;          //define PM2.5 value of the air
detector module
int NDPyear, NDPmonth, NDPday, NDPhour, NDPminute, NDPsecond;
unsigned long epoch ;
int HumidityData = 0 ;
int TemperatureData = 0 ;

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //
設定 LCD I2C 位址
DHT dht(DHTSensorPin, DHTTYPE);

void setup() {
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
Serial.begin(9600);
mySerial.begin(9600); // PMS 3003 UART has baud rate 9600
lcd.begin(20, 4);      // 初始化 LCD，一行 20 的字元，共 4 行，
預設開啟背光
    lcd.backlight(); // 開啟背光

    MacAddress = GetWifiMac() ;
    ShowMac() ;
    initializeWiFi();

    initRTC() ;
    ShowDateTime() ;
    ShowInternetStatus() ;
    delay(1500);
}

void loop() { // run over and over
    ShowDateTime() ;
    retrievePM25Value() ;
    ShowHumidity() ;
    delay(1000); // delay 1 minute for next measurement

}

void ShowMac()
{
    lcd.setCursor(0, 0); // 設定游標位置在第一行行首
    lcd.print("MAC:");
    lcd.print(MacAddress);

}

void ShowInternetStatus()
{
    lcd.setCursor(0, 1); // 設定游標位置
    if (WiFi.status())
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
        {
            Meip = WiFi.localIP();
            lcd.print("@:");
            lcd.print(Meip);
        }
        else
        {
            lcd.print("DisConnected:");
        }
    }

}

void ShowPM25(int pp25)
{
    lcd.setCursor(0, 3); // 設定游標位置在第一行行首
    lcd.print("PM2.5:    ");
    lcd.setCursor(9, 3); // 設定游標位置在第一行行首
    lcd.print(pp25);

}

void ShowDateTime()
{
    //  getCurrentTime(epoch, &NDPyear, &NDPmonth, &NDPday, &NDPhour,
    &NDPminute, &NDPsecond);
    lcd.setCursor(0, 2); // 設定游標位置在第一行行首
    lcd.print(StrDate());
    lcd.setCursor(11, 2); // 設定游標位置在第一行行首
    lcd.print(StrTime());
    //  lcd.print();

}

String StrDate() {
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
String ttt ;
//nowT = now;
DateTime now = RTC.now();
    ttt = print4digits(now.year()) + "-" + print2digits(now.month()) +
    "-" + print2digits(now.day()) ;
    //ttt = print4digits(NDPyar) + "/" + print2digits(NDPmonth) + "/"
+ print2digits(NDPday) ;
    return ttt ;
}

String StringDate(int yyy,int mmm,int ddd) {
    String ttt ;
    //nowT = now;
    ttt = print4digits(yyy) + "-" + print2digits(mmm) + "-" +
print2digits(ddd) ;
    return ttt ;
}

String StrTime() {
    String ttt ;
    // nowT = RTC.now();
    DateTime now = RTC.now();
    ttt = print2digits(now.hour()) + ":" + print2digits(now.minute())
+ ":" + print2digits(now.second()) ;
    // ttt = print2digits(NDPhour) + ":" + print2digits(NDPminute) +
":" + print2digits(NDPsecond) ;
    return ttt ;
}

String StringTime(int hhh,int mmm,int sss) {
    String ttt ;
    ttt = print2digits(hhh) + ":" + print2digits(mmm) + ":" +
print2digits(sss) ;
    return ttt ;
}
```

```
String GetWifiMac()
{
    String tt ;
    String t1, t2, t3, t4, t5, t6 ;
    WiFi.status();    //this method must be used for get MAC
    WiFi.macAddress(MacData);

    Serial.print("Mac:");
    Serial.print(MacData[0], HEX) ;
    Serial.print("/");
    Serial.print(MacData[1], HEX) ;
    Serial.print("/");
    Serial.print(MacData[2], HEX) ;
    Serial.print("/");
    Serial.print(MacData[3], HEX) ;
    Serial.print("/");
    Serial.print(MacData[4], HEX) ;
    Serial.print("/");
    Serial.print(MacData[5], HEX) ;
    Serial.print("~");

    t1 = print2HEX((int)MacData[0]);
    t2 = print2HEX((int)MacData[1]);
    t3 = print2HEX((int)MacData[2]);
    t4 = print2HEX((int)MacData[3]);
    t5 = print2HEX((int)MacData[4]);
    t6 = print2HEX((int)MacData[5]);
    tt = (t1+t2+t3+t4+t5+t6) ;
    Serial.print(tt);
    Serial.print("\n");

    return tt ;
}

String print2HEX(int number) {
    String ttt ;
    if (number >= 0 && number < 16)
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
{
    ttt = String("0") + String(number, HEX);
}
else
{
    ttt = String(number, HEX);
}
return ttt ;
}

String print2digits(int number) {
    String ttt ;
    if (number >= 0 && number < 10)
    {
        ttt = String("0") + String(number);
    }
    else
    {
        ttt = String(number);
    }
    return ttt ;
}

String print4digits(int number) {
    String ttt ;
    ttt = String(number);
    return ttt ;
}

// send an NTP request to the time server at the given address
void retrieveNtpTime() {
    Serial.println("Send NTP packet");

    Udp.beginPacket(ntpServer, 123); //NTP requests are to port 123
    Udp.write(nptSendPacket, NTP_PACKET_SIZE);
    Udp.endPacket();
}
```


整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
    if(Udp.parsePacket()) {
        Serial.println("NTP packet received");
        Udp.read(ntpRecvBuffer, NTP_PACKET_SIZE); // read the packet
into the buffer

        unsigned long highWord = word(ntpRecvBuffer[40],
ntpRecvBuffer[41]);
        unsigned long lowWord = word(ntpRecvBuffer[42],
ntpRecvBuffer[43]);
        unsigned long secsSince1900 = highWord << 16 | lowWord;
        const unsigned long seventyYears = 2208988800UL;
        //      epoch = secsSince1900 - seventyYears + timeZoneOffset ;
        epoch = secsSince1900 - seventyYears ;

        epochSystem = epoch - millis() / 1000;
    }
}

void getCurrentTime(unsigned long epoch, int *year, int *month, int
*day, int *hour, int *minute, int *second) {
    int tempDay = 0;

    *hour = (epoch % 86400L) / 3600;
    *minute = (epoch % 3600) / 60;
    *second = epoch % 60;

    *year = 1970;
    *month = 0;
    *day = epoch / 86400;

    for (*year = 1970; ; (*year)++) {
        if (tempDay + (LEAP_YEAR(*year) ? 366 : 365) > *day) {
            break;
        } else {
            tempDay += (LEAP_YEAR(*year) ? 366 : 365);
        }
    }
}
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
    }  
    }  
    tempDay = *day - tempDay; // the days left in a year  
    for ((*month) = 0; (*month) < 12; (*month)++) {  
        if ((*month) == 1) {  
            if (LEAP_YEAR(*year)) {  
                if (tempDay - 29 < 0) {  
                    break;  
                } else {  
                    tempDay -= 29;  
                }  
            } else {  
                if (tempDay - 28 < 0) {  
                    break;  
                } else {  
                    tempDay -= 28;  
                }  
            }  
        } else {  
            if (tempDay - monthDays[(*month)] < 0) {  
                break;  
            } else {  
                tempDay -= monthDays[(*month)];  
            }  
        }  
    }  
    (*month)++;  
    *day = tempDay+2; // one for base 1, one for current day  
}  
  
void retrievePM25Value() {  
    int idx;  
    bool hasPm25Value = false;  
    int timeout = 200;  
    while (!hasPm25Value) {  
        idx = 0;  
        memset(buf, 0, pmsDataLen);  
    }  
}
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
while (mySerial.available()) {
  buf[idx++] = mySerial.read();
}

if (buf[0] == 0x42 && buf[1] == 0x4d) {
  pm25 = ( buf[12] << 8 ) | buf[13];
  Serial.print("pm2.5: ");
  Serial.print(pm25);
  Serial.print(" ug/m3");
  Serial.println("");
  hasPm25Value = true;
  ShowPM25(pm25) ;
}
timeout--;
if (timeout < 0) {
  Serial.println("fail to get pm2.5 data");
  break;
}
}
}

void initializeWiFi() {
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network. Change this line if using open
or WEP network:
    status = WiFi.begin(ssid, pass);
    // status = WiFi.begin(ssid);

    // wait 10 seconds for connection:
    delay(10000);
  }

  // local port to listen for UDP packets
  Udp.begin(2390);
}
```

```
void printWifiData()
{
    // print your WiFi shield's IP address:
    Meip = WiFi.localIP();
    Serial.print(" IP Address: ");
    Serial.println(Meip);

    // print your MAC address:
    byte mac[6];
    WiFi.macAddress(mac);
    Serial.print("MAC address: ");
    Serial.print(mac[5], HEX);
    Serial.print(":");
    Serial.print(mac[4], HEX);
    Serial.print(":");
    Serial.print(mac[3], HEX);
    Serial.print(":");
    Serial.print(mac[2], HEX);
    Serial.print(":");
    Serial.print(mac[1], HEX);
    Serial.print(":");
    Serial.println(mac[0], HEX);

    // print your subnet mask:
    Mesubnet = WiFi.subnetMask();
    Serial.print("NetMask: ");
    Serial.println(Mesubnet);

    // print your gateway address:
    Megateway = WiFi.gatewayIP();
    Serial.print("Gateway: ");
    Serial.println(Megateway);
}

void initRTC()
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
{
    Wire.begin();
    RTC.begin();
    SetRTCFromNtpTime() ;
    if (! RTC.isrunning()) {
        Serial.println("RTC is NOT running!");
        // following line sets the RTC to the date & time this sketch was
compiled
        //    RTC.adjust(DateTime(__DATE__, __TIME__));

    }
}

void SetRTCFromNtpTime()
{
    retrieveNtpTime();
    //DateTime ttt;
    getCurrentTime(epoch+timeZoneOffset, &NDPyear, &NDPmonth,
&NDPday, &NDPhour, &NDPminute, &NDPsecond);
    //ttt->year = NDPyear ;
    Serial.print("NDP Date is :");
    Serial.print(StringDate(NDPyear, NDPmonth, NDPday));
    Serial.print("and ");
    Serial.print("NDP Time is :");
    Serial.print(StringTime(NDPhour, NDPminute, NDPsecond));
    Serial.print("\n");

    RTC.adjust(DateTime(epoch+timeZoneOffset));

}

void ShowHumidity()
{
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    float f = dht.readTemperature(true);
```

整合空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV81A)

```
HumidityData = (int)h ;
TemperatureData = (int)t ;
Serial.print("Humidity :") ;
Serial.print(h) ;
Serial.print("%  /") ;
Serial.print(t) ;
Serial.print("C  \n") ;
// Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}

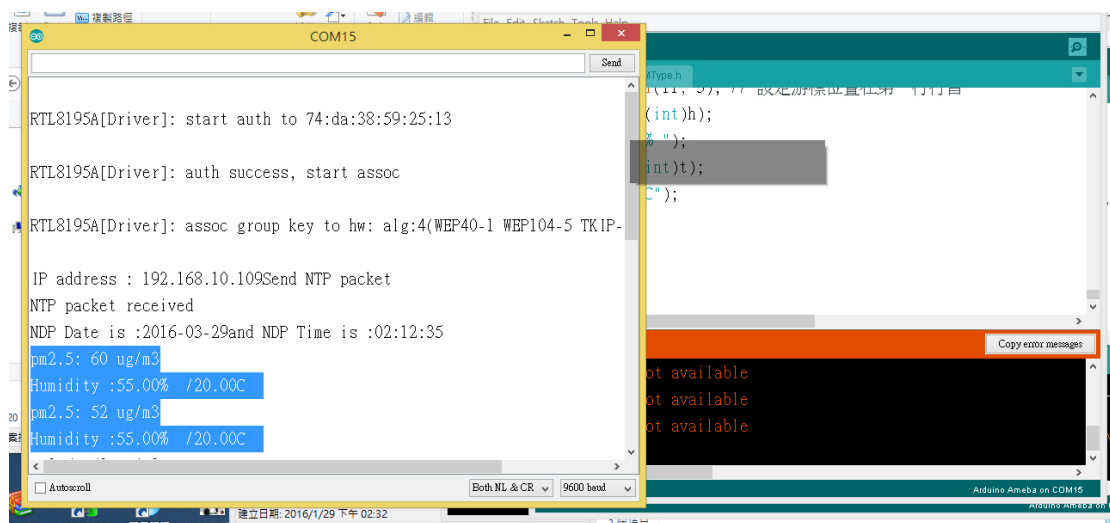
lcd.setCursor(11, 3); // 設定游標位置在第一行行首
  lcd.print((int)h);
  lcd.print("% ");
  lcd.print((int)t);

}
```

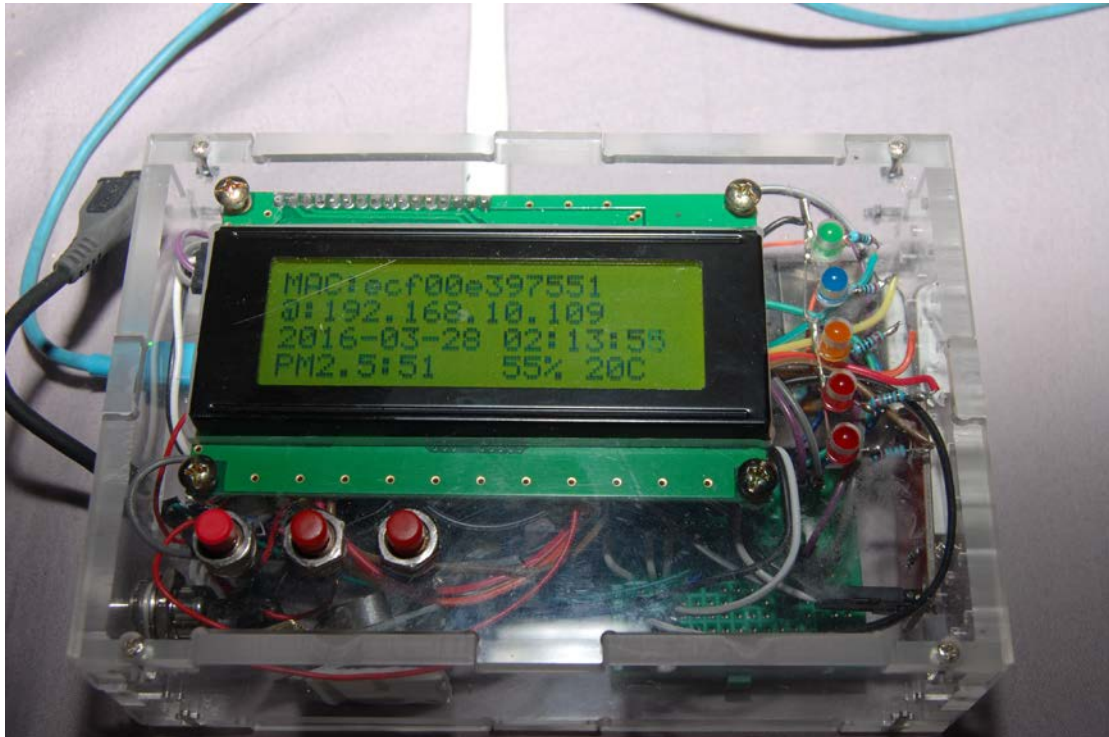
資料下載：

<https://github.com/brucetsao/makerdiwo/tree/master/201604/PMS3003AirQualityV81A>

我們可以連上網際網路，也可以偵測 PM2.5 的空氣懸浮粒子、PM1.0 的空氣懸浮粒子 & PM 10 的空氣懸浮粒子，溫度、濕度都也可以偵測到值



(a). 讀取完整溫溼度等資料之開發 IDE 監控畫面



(b). 空氣粒子感測裝置完整資訊之 LCD 顯示畫面

圖 5 整合空氣懸浮粒子感測器測試程式三畫面結果

本文為『PM2.5空氣感測器』系列第四篇：電路設計下篇，主要介紹如何將PM2.5空氣感測器加入RTC時鐘模組，溫溼度感測模組，並將『PM2.5空氣感測器』系列第三篇：電路設計上篇與本篇文章一同閱讀與整合，逐一完成PM2.5空氣感測器的電路安裝。

後續筆者還會繼續發表『PM2.5空氣感測器』系列的文章，讓我們在未來可以創造出更優質、智慧化的家庭。

敬請期待更多的文章。

筆者介紹

曹永忠 (Yung-Chung Tsao)：目前為自由作家，專注於軟體工程、軟體開發與設計、物件導向程式設計、Arduino 開發、嵌入式系統開發，商品攝影及人像攝影。長期投入資訊系統設計與開發、企業應用系統開發、軟體工程、新產品開發管理、商品及人像攝影等領域，並持續發表作品及相關專業著作。



Email: prgbruce@gmail.com , Line ID : dr.brucetsao

Arduino 部落格 : <http://taiwanarduino.blogspot.tw/>

臉書社群(Arduino.Taiwan) : <https://www.facebook.com/groups/Arduino.Taiwan/>

活動官網 : <http://arduino.kktix.cc/>

Youtube : https://www.youtube.com/channel/UCcYG2yY_u0mlaotcA4hrRgQ

程式下載網址 : <https://github.com/brucetsao/makerdiwo>

參考文獻：

曹永忠. (2016a). AMEBA 透過網路校時 RTC 時鐘模組. *智慧家庭*. Retrieved from <http://makerpro.cc/2016/03/using-ameba-to-develop-a-timing-controlling-device-via-internet/>

曹永忠. (2016b). 用 RTC 時鐘模組驅動 Ameba 時間功能. *智慧家庭*. Retrieved from <http://makerpro.cc/2016/03/drive-ameba-time-function-by-rtc-module/>

曹永忠. (2016c). 智慧家庭：PM2.5 空氣感測器（感測器篇）. *智慧家庭*. Retrieved from <http://vmaker.tw/project/view/695>

曹永忠. (2016d). 智慧家庭：PM2.5 空氣感測器（硬體組裝上篇）. *智慧家庭*. Retrieved from <http://vmaker.tw/project/view/749>

曹永忠. (2016e). 智慧家庭：PM2.5 空氣感測器（硬體組裝下篇）. *智慧家庭*. Retrieved from <http://vmaker.tw/project/view/772>

曹永忠. (2016f). 智慧家庭：PM2.5 空氣感測器（電路設計上篇）. *智慧家庭*. Retrieved from <http://vmaker.tw/project/view/817>

曹永忠. (2016g). 實戰 ARDUINO 的 RTC 時鐘模組，教你怎麼進行網路校時. Retrieved from <http://www.techbang.com/posts/40869-smart-home-arduino-internet-soul-internet-school>

曹永忠, 許智誠, & 蔡英德. (2015a). *Ameba 空氣粒子感測裝置設計與開發 (MQTT 篇): Using Ameba to Develop a PM 2.5 Monitoring Device to MQTT* (初版 ed.). 台灣、彰化: 渥瑪數位有限公司.

曹永忠, 許智誠, & 蔡英德. (2015b). *Ameba 空氣粒子感測裝置設計與開發 (MQTT 篇): Using Ameba to Develop a PM 2.5 Monitoring Device to MQTT* (初版 ed.). 台灣、彰化: 渥瑪數位有限公司.

曹永忠, 許智誠, & 蔡英德. (2015c). Maker 物聯網實作：用 DHx 溫濕度感測模組
回傳天氣溫溼度. 物聯網. Retrieved from
<http://www.techbang.com/posts/26208-the-internet-of-things-daily-life-how-to-know-the-temperature-and-humidity>