智慧家庭:PM2.5 空氣感測器(電路設計上篇)

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(硬體組裝上/下篇)』(曹永忠, 許智誠, & 蔡英德, 2015a, 2015b)，延續未完成的硬體電路組裝上篇，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，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。

安裝偵測空氣懸浮粒子感測器

第一步，我們安裝攀藤科技(Plantower)[1]的 PMS3003 空氣懸浮粒子感測器，其感測器上有 8 個接腳，分別如下：

- pin1，供電輸入接腳，要輸入 5V 電壓的電力。
- pin2，接地接腳，接上 0V。
- pin3，設定輸入接腳，接收到 0 時感測器處於待備狀態(減少感測器的耗電)，接收到 1 則為運作狀態。
- pin4，串列埠接收接腳(Rx)，邏輯準位 3.3V。
- pin5，串列埠輸出接腳(Tx)，邏輯準位 3.3V。
- pin6，重置、重新開機輸入接腳，接收到 3.3V 則重新開機。
- pin7，還沒用到。

[1]北京攀藤科技有限公司是一家專注於空氣品質感測器研發、生產與銷售的高科技企業，作為行業領跑者，公司通過不斷創新和品質追求，已與國內外多家知名企業建立了良好的戰略合作夥伴關係(http://www.plantower.com/)。

- pin8，還沒用到。

期 PMS3003 粉塵感測器硬體連接方式如所示(曹永忠, 2016)，讀者可以看到 PMS3003 粉塵感測器接腳如下：

Digital interface definition

Pin number	Function	Explain
PIN1	VCC	Power supply DC5V
PIN2	GND	Negative power supply
PIN3	SET(Internal 50K pull-up)	Set pin 3.3V level
PIN4	RXD/I2C_SCL	Digital pin 3.3V level
PIN5	TXD/I2C_SDL	Digital pin 3.3V level
PIN6	RESET	Module reset signal 3.3V level
PIN7	PWM output	3.3V level only for single sensor output
PIN8	Mode selection (Internal 50K pull-up)	High or NC Serial port mode 3.3V level Low level I2C model

Note: SET=1 module is in the working state, and the module is electrically active to send data Instruction change working mode

SET=0 module into low power standby mode

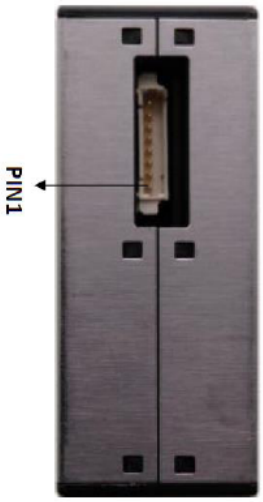

圖 1 PMS3003 G3 PM2.5 粉塵感測器接腳圖

資料出處：

https://www.dfrobot.com/wiki/index.php?title=PM2.5_laser_dust_sensor_SKU:SEN0177

了解接腳後，將 PMS3003 與 Ameba 連接，如下圖所示進行電路連接。

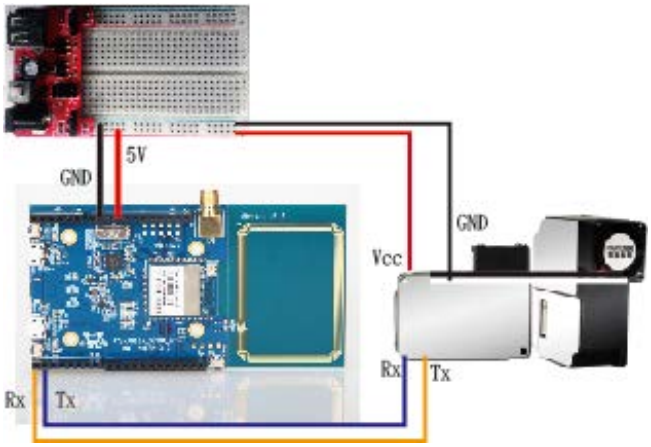

圖 1 安裝偵測空氣懸浮粒子感測器

有些讀者仍需要了解如何將正確腳位接在阿米巴開發板上，所以作者增加下表之接腳圖(曹永忠, 2016)，讓讀者更加了解。

表 1 空氣懸浮粒子感測器接腳圖

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

我們將下表之 PMS3003 空氣懸浮粒子感測器測試程式攥寫好之後，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

PMS3003 空氣懸浮粒子感測器測試程式(PMS3003AirQualityV2)

```
/*
 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 <SoftwareSerial.h>

SoftwareSerial mySerial(0, 1); // RX, TX
```

PMS3003 空氣懸浮粒子感測器測試程式(PMS3003AirQualityV2)

```
#define pmsDataLen 32
uint8_t buf[pmsDataLen];
int idx = 0;
int pm25 = 0;

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600); // PMS 3003 UART has baud rate 9600
}

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");
  }
}

uint8_t checkValue(uint8_t *thebuf, uint8_t leng)
{
  uint8_t receiveflag=1;
  uint16_t receiveSum=0;
  uint8_t i=0;

  for(i=0;i<leng;i++)
  {
  receiveSum=receiveSum+thebuf[i];
```

PMS3003 空氣懸浮粒子感測器測試程式(PMS3003AirQualityV2)

```
  }


if(receiveSum==((thebuf[leng-2]<<8)+thebuf[leng-1]+thebuf[leng-2]+the
buf[leng-1]))  //check the serial data
  {
    receiveSum=0;
  receiveflag=1;
//  Serial.print(receiveflag);
  return receiveflag;
  }
}
//transmit PM Value to PC
uint16_t transmitPM01(uint8_t *thebuf)
{

  uint16_t PM01Val;

  PM01Val=((thebuf[4]<<8) + thebuf[5]); //count PM1.0 value of the air
detector module
  return PM01Val;
}
```

資料下載區：
https://github.com/brucetsao/makerdiwo/tree/master/201604/PMS3003AirQualityV2

上述程式執行後，可以見到下圖之 PMS3003 空氣懸浮粒子感測器測試程式畫面結果，也可以輕易讀到懸浮粒子的濃度了。

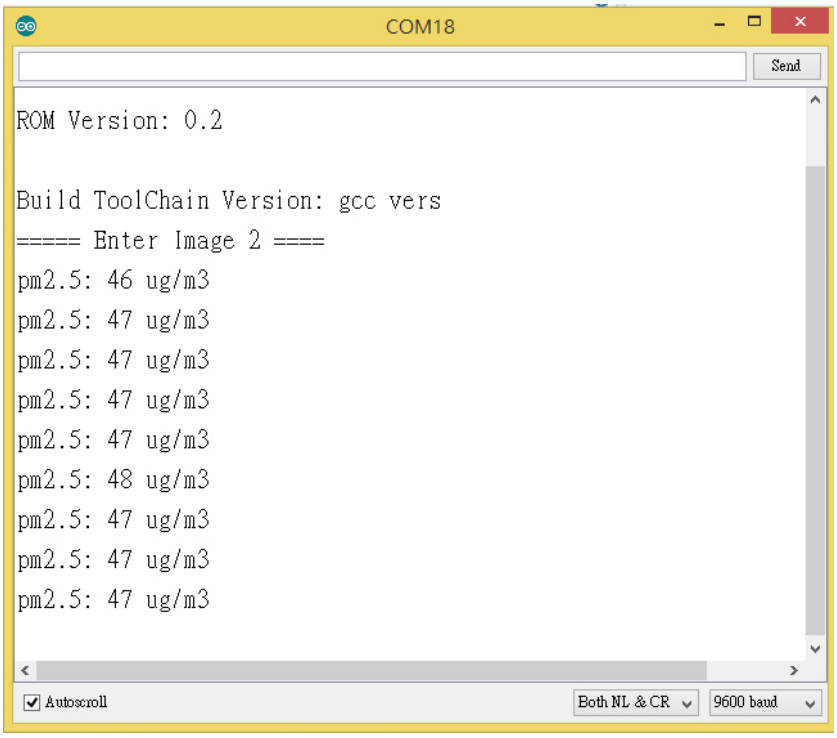

圖 2 PMS3003 空氣懸浮粒子感測器測試程式畫面結果

安裝 LCD2004 顯示器

為了方便顯示 PMS3003 感測器的資料，作者加入 LCD 2004 顯示模組，如下圖所示進行電路連接(曹永忠 et al., 2015a, 2015b)。

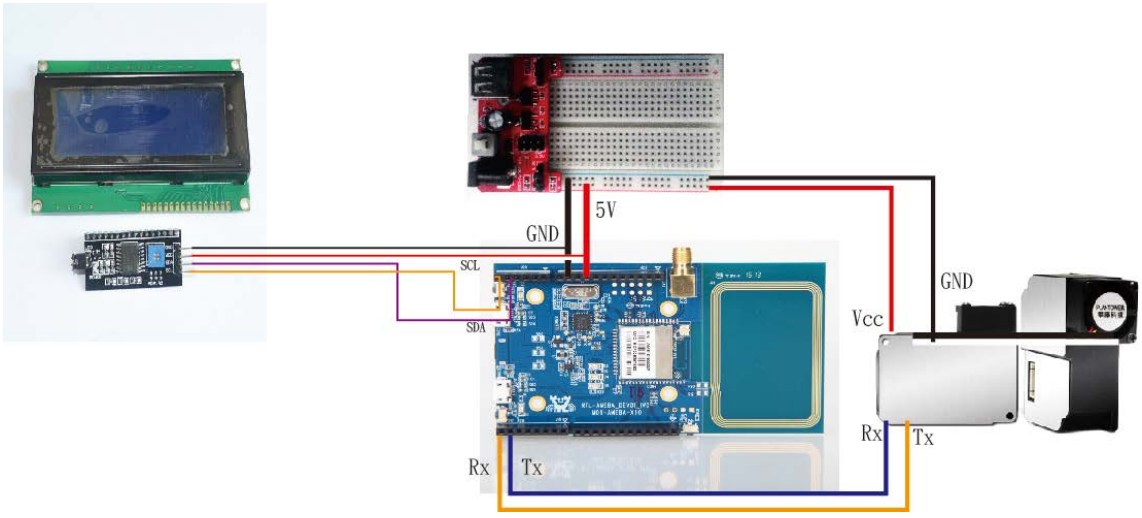

圖 2 安裝 LCD 2004 顯示器

有些讀者仍需要了解如何將 LCD 2004 顯示模組正確腳位接在阿米巴開發板上，所以作者增加下表之接腳表，讓讀者更加了解。

表 3 LCD 2004 顯示模組接腳表(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

我們將下表之 PMS3003 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 4 PMS3003 空氣懸浮粒子感測器測試程式二

PMS3003 空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV11)

```
/*
 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:
```

PMS3003 空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV11)

```
 * RX is digital pin 0 (connect to TX of PMS 3003)
 * TX is digital pin 1 (connect to RX of PMS 3003)

 */
#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>

SoftwareSerial mySerial(0, 1); // RX, TX

#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

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(16, 2);      // 初始化 LCD，一行 16 的字元，共 2 行，預設
開啟背光
       lcd.backlight(); // 開啟背光

  ShowMac() ;
}
```

PMS3003 空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV11)

```
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) ;
  }
/*
  int checkSum=checkValue(buf,pmsDataLen);
  if(pmsDataLen&&checkSum)
  {
    PM01Value=transmitPM01(buf);
    PM2_5Value=transmitPM2_5(buf);
    PM10Value=transmitPM10(buf);
  }
    static unsigned long OledTimer=millis();
  if (millis() - OledTimer >=1000)
  {
    OledTimer=millis();
      Serial.print("PM1.0: ");
      Serial.print(PM01Value);
      Serial.println("  ug/m3");
      Serial.print("PM2.5: ");
      Serial.print(PM2_5Value);
      Serial.println("  ug/m3");
      Serial.print("PM10:  "); //send PM1.0 data to bluetooth
      Serial.print(PM10Value);
```

PMS3003 空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV11)

```
      Serial.println("ug/m3");
  }
*/

}

uint8_t checkValue(uint8_t *thebuf, uint8_t leng)
{
  uint8_t receiveflag=1;
  uint16_t receiveSum=0;
  uint8_t i=0;

  for(i=0;i<leng;i++)
  {
  receiveSum=receiveSum+thebuf[i];
  }


if(receiveSum==((thebuf[leng-2]<<8)+thebuf[leng-1]+thebuf[leng-2]+the
buf[leng-1]))  //check the serial data
  {
    receiveSum=0;
  receiveflag=1;
//  Serial.print(receiveflag);
  return receiveflag;
  }
}
//transmit PM Value to PC
uint16_t transmitPM01(uint8_t *thebuf)
{

  uint16_t PM01Val;

  PM01Val=((thebuf[4]<<8) + thebuf[5]); //count PM1.0 value of the air
detector module
  return PM01Val;
}
```

PMS3003 空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV11)

```
//transmit PM Value to PC
uint16_t transmitPM2_5(uint8_t *thebuf)
{
  uint16_t PM2_5Val;

    PM2_5Val=((thebuf[6]<<8) + thebuf[7]);//count PM2.5 value of the air
detector module

  return PM2_5Val;
  }

//transmit PM Value to PC
uint16_t transmitPM10(uint8_t *thebuf)
{

  uint16_t PM10Val;

    PM10Val=((thebuf[8]<<8) + thebuf[9]); //count PM10 value of the air
detector module

  return PM10Val;
  }

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

}

void ShowPM(int pp25)
{
    lcd.setCursor(0, 1); // 設定游標位置在第一行行首
     lcd.print("PM2.5:");
     lcd.print(pp25);
```

PMS3003 空氣懸浮粒子感測器測試程式二(PMS3003AirQualityV11)
}

資料下載區：
https://github.com/brucetsao/makerdiwo/tree/master/201604/PMS3003AirQualityV11

上述程式執行後，可以見到下圖之 PMS3003 空氣懸浮粒子感測器測試程式二畫面結果，也可以輕易顯示讀到懸浮粒子的濃度於 LCD2004 顯示模組上。

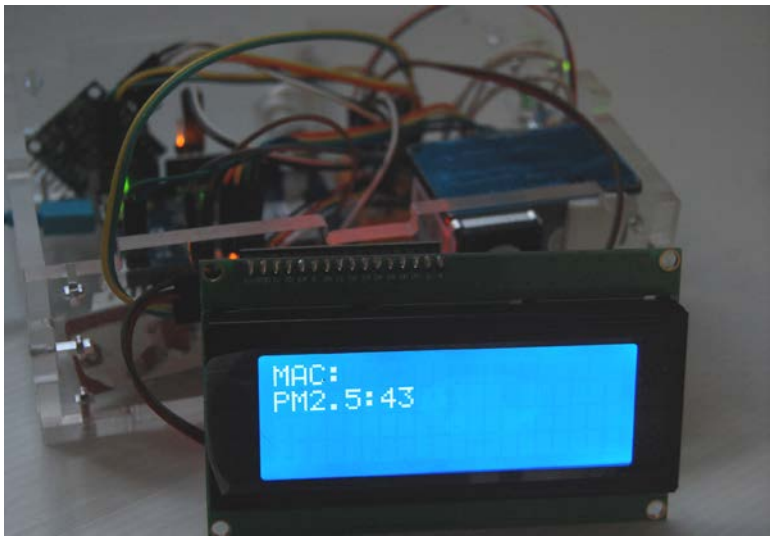

圖 3 PMS3003 空氣懸浮粒子感測器測試程式二畫面結果

連上網路顯示裝置 ID

我們將下表之 PMS3003 空氣懸浮粒子感測器測試程式三攥寫好之後，編譯完成後上傳到 Ameba 開發板，

表 5 PMS3003 空氣懸浮粒子感測器測試程式三

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
/*
 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 "PMType.h"
#include <WiFi.h>
#include <PubSubClient.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 <LiquidCrystal_I2C.h>

#include <SoftwareSerial.h>

#include <WiFi.h>
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
uint8_t MacData[6];

SoftwareSerial mySerial(0, 1); // RX, TX
char ssid[] = "TSAO";       // your network SSID (name)
char pass[] = "TSAO1234";     // your network password
int keyIndex = 0;               // your network key Index number (needed
only for WEP)

char gps_lat[] = "23.954710";  // device's gps latitude 清心福全(中正
店) 510 彰化縣員林市中正路 254 號
char gps_lon[] = "120.574482"; // device's gps longitude 清心福全(中正
店) 510 彰化縣員林市中正路 254 號

char server[] = "gpssensor.ddns.net"; // the MQTT server of LASS

#define MAX_CLIENT_ID_LEN 10
#define MAX_TOPIC_LEN     50
char clientId[MAX_CLIENT_ID_LEN];
char outTopic[MAX_TOPIC_LEN];

WiFiClient wifiClient;
PubSubClient client(wifiClient);

int status = WL_IDLE_STATUS;

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 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,
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
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 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;
  unsigned long epoch  ;
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // 設
定 LCD I2C 位址
String MacAddress ;
IPAddress Meip ;
  IPAddress Mesubnet ;
  IPAddress Megateway ;
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
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
WiFi.status();
MacAddress = GetWifiMac() ;
  ShowMac() ;
    initializeWiFi();
  retrieveNtpTime();
  ShowDateTime() ;
  ShowInternetStatus() ;
 // initializeMQTT();

}

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) ;
      }
/*
  int checkSum=checkValue(buf,pmsDataLen);
  if(pmsDataLen&&checkSum)
  {
    PM01Value=transmitPM01(buf);
    PM2_5Value=transmitPM2_5(buf);
    PM10Value=transmitPM10(buf);
  }
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
    static unsigned long OledTimer=millis();
  if (millis() - OledTimer >=1000)
  {
    OledTimer=millis();
      Serial.print("PM1.0: ");
      Serial.print(PM01Value);
      Serial.println("  ug/m3");
      Serial.print("PM2.5: ");
      Serial.print(PM2_5Value);
      Serial.println("  ug/m3");
      Serial.print("PM10:  "); //send PM1.0 data to bluetooth
      Serial.print(PM10Value);
      Serial.println("ug/m3");
  }
*/

}

uint8_t checkValue(uint8_t *thebuf, uint8_t leng)
{
  uint8_t receiveflag=1;
  uint16_t receiveSum=0;
  uint8_t i=0;

  for(i=0;i<leng;i++)
  {
  receiveSum=receiveSum+thebuf[i];
  }


if(receiveSum==((thebuf[leng-2]<<8)+thebuf[leng-1]+thebuf[leng-2]+the
buf[leng-1]))  //check the serial data
      {
        receiveSum=0;
      receiveflag=1;
    //  Serial.print(receiveflag);
      return receiveflag;
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
      }
}
//transmit PM Value to PC
uint16_t transmitPM01(uint8_t *thebuf)
{

  uint16_t PM01Val;

  PM01Val=((thebuf[4]<<8) + thebuf[5]); //count PM1.0 value of the air
detector module
  return PM01Val;
}

//transmit PM Value to PC
uint16_t transmitPM2_5(uint8_t *thebuf)
{
      uint16_t PM2_5Val;

        PM2_5Val=((thebuf[6]<<8) + thebuf[7]);//count PM2.5 value of the
air detector module

      return PM2_5Val;
  }

//transmit PM Value to PC
uint16_t transmitPM10(uint8_t *thebuf)
{

      uint16_t PM10Val;

        PM10Val=((thebuf[8]<<8) + thebuf[9]); //count PM10 value of the
air detector module

      return PM10Val;
  }

void ShowMac()
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
{
     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())
          {
               lcd.print("Connected:");
              lcd.print(Meip);
          }

}

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

}

void ShowDateTime()
{
    getCurrentTime(epoch, &NDPyear, &NDPmonth, &NDPday, &NDPhour,
&NDPminute, &NDPsecond);
    lcd.setCursor(1, 2); // 設定游標位置在第一行行首
     lcd.print(StrDate());
    lcd.setCursor(12, 2); // 設定游標位置在第一行行首
     lcd.print(StrTime());
   //  lcd.print();

}
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
String  StrDate() {
  String ttt ;
 ttt = print4digits(NDPyear) + "/" + print2digits(NDPmonth) + "/" +
print2digits(NDPday) ;
  return ttt ;
}

String  StrTime() {
  String ttt ;
  ttt = print2digits(NDPhour) + ":" + print2digits(NDPminute) + ":" +
print2digits(NDPsecond) ;
  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]);
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
   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);
  }
  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 callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i=0;i<length;i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

// 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],
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 ;

    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) {
          break;
        } else {
          tempDay -= 29;
        }
      } else {
        if (tempDay - 28 < 0) {
          break;
        } else {
          tempDay -= 28;
        }
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

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

void reconnectMQTT() {
  // Loop until we're reconnected
  char payload[300];

  unsigned long epoch = epochSystem + millis() / 1000;

  getCurrentTime(epoch, &NDPyear, &NDPmonth, &NDPday, &NDPhour,
&NDPminute, &NDPsecond);

  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect(clientId)) {
      Serial.println("connected");

      sprintf(payload,
"|ver_format=3|fmt_opt=1|app=Pm25Ameba|ver_app=0.0.1|device_id=%s|tic
k=%d|date=%4d-%02d-%02d|time=%02d:%02d:%02d|device=Ameba|s_d0=%d|gps_
lat=%s|gps_lon=%s|gps_fix=1|gps_num=9|gps_alt=2",
        clientId,
        millis(),
        NDPyear, NDPmonth, NDPday,
        NDPhour, NDPminute, NDPsecond,
        pm25,
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
        gps_lat, gps_lon
      );

      // Once connected, publish an announcement...
      client.publish(outTopic, payload);
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

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

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
    }
  }
}

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 initializeMQTT() {
  byte mac[6];
  WiFi.macAddress(mac);
  memset(clientId, 0, MAX_CLIENT_ID_LEN);
  sprintf(clientId, "FT1_0%02X%02X", mac[4], mac[5]);
  sprintf(outTopic, "LASS/Test/Pm25Ameba/%s", clientId);

  Serial.print("MQTT client id:");
  Serial.println(clientId);
  Serial.print("MQTT topic:");
  Serial.println(outTopic);

  client.setServer(server, 1883);
  client.setCallback(callback);
}
```

PMS3003 空氣懸浮粒子感測器測試程式三(PMS3003AirQualityV21)

```
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);
    }
```

資料下載區：

https://github.com/brucetsao/makerdiwo/tree/master/201604/PMS3003AirQualityV21

上述程式執行後，可以見到下圖之 PMS3003 空氣懸浮粒子感測器測試程式三畫面結果，也可以顯示裝置的 MAC 位址於 LCD2004 顯示模組上。

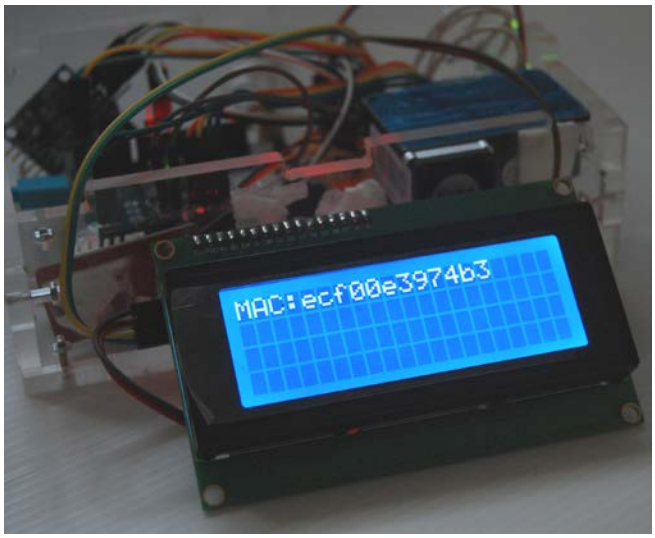

圖 4 PMS3003 空氣懸浮粒子感測器測試程式三畫面結果

資訊整合顯示資訊

我們可以讀到 MAC 位址來當作裝置 ID，並且能夠將空氣懸浮粒子感測器的資訊讀取出來，我們要將這兩個整合。

我們將下表之 PMS3003 空氣懸浮粒子感測器測試程式四攥寫好之後，編譯完成後上傳到 Ameba 開發板，

表 6 PMS3003 空氣懸浮粒子感測器測試程式四

PMS3003 空氣懸浮粒子感測器測試程式四(PMS3003AirQualityV41)

```
/*
 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
```

PMS3003 空氣懸浮粒子感測器測試程式四(PMS3003AirQualityV41)

```
    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 <String.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
#define turnon HIGH
#define turnoff LOW
#define ParticleSensorLed 7
#define InternetLed 6
#define AccessLed 5


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

#include <WiFi.h>

uint8_t MacData[6];

SoftwareSerial mySerial(0, 1); // RX, TX

#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
```

PMS3003 空氣懸浮粒子感測器測試程式四(PMS3003AirQualityV41)

```
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

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // 設定 LCD I2C 位址
String MacAddress ;
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 16 的字元，共 2 行，預設開啟背光
       lcd.backlight(); // 開啟背光
    WiFi.status();
MacAddress = GetWifiMac() ;

  ShowMac() ;
}

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) ;
  }
```

PMS3003 空氣懸浮粒子感測器測試程式四(PMS3003AirQualityV41)

```
/*
  int checkSum=checkValue(buf,pmsDataLen);
  if(pmsDataLen&&checkSum)
  {
    PM01Value=transmitPM01(buf);
    PM2_5Value=transmitPM2_5(buf);
    PM10Value=transmitPM10(buf);
  }
    static unsigned long OledTimer=millis();
  if (millis() - OledTimer >=1000)
  {
    OledTimer=millis();
      Serial.print("PM1.0: ");
      Serial.print(PM01Value);
      Serial.println("  ug/m3");
      Serial.print("PM2.5: ");
      Serial.print(PM2_5Value);
      Serial.println("  ug/m3");
      Serial.print("PM10:  "); //send PM1.0 data to bluetooth
      Serial.print(PM10Value);
      Serial.println("ug/m3");
  }
*/

}

uint8_t checkValue(uint8_t *thebuf, uint8_t leng)
{
  uint8_t receiveflag=1;
  uint16_t receiveSum=0;
  uint8_t i=0;

  for(i=0;i<leng;i++)
  {
  receiveSum=receiveSum+thebuf[i];
  }
```

PMS3003 空氣懸浮粒子感測器測試程式四(PMS3003AirQualityV41)

```
if(receiveSum==((thebuf[leng-2]<<8)+thebuf[leng-1]+thebuf[leng-2]+the
buf[leng-1]))  //check the serial data
  {
    receiveSum=0;
  receiveflag=1;
//  Serial.print(receiveflag);
  return receiveflag;
  }
}
//transmit PM Value to PC
uint16_t transmitPM01(uint8_t *thebuf)
{

  uint16_t PM01Val;

  PM01Val=((thebuf[4]<<8) + thebuf[5]); //count PM1.0 value of the air
detector module
  return PM01Val;
}

//transmit PM Value to PC
uint16_t transmitPM2_5(uint8_t *thebuf)
{
  uint16_t PM2_5Val;

    PM2_5Val=((thebuf[6]<<8) + thebuf[7]);//count PM2.5 value of the air
detector module

  return PM2_5Val;
  }

//transmit PM Value to PC
uint16_t transmitPM10(uint8_t *thebuf)
{

  uint16_t PM10Val;
```

PMS3003 空氣懸浮粒子感測器測試程式四(PMS3003AirQualityV41)

```
    PM10Val=((thebuf[8]<<8) + thebuf[9]); //count PM10 value of the air
detector module

  return PM10Val;
  }

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 ShowPM(int pp25)
{
    lcd.setCursor(0, 1); // 設定游標位置在第一行行首
     lcd.print("PM2.5:");
     lcd.print(pp25);

}

String GetWifiMac()
{
   String tt ;
    String t1,t2,t3,t4,t5,t6 ;
  WiFi.macAddress(MacData);

  Serial.print("Mac:");
   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);
  }
  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 ;
}
```

參考資料來源：Grove- Temperature and Humidity Sensor (http://www.seeedstudio.com/wiki/Grove-_Temperature_and_Humidity_Sensor)

資料下載區：https://github.com/brucetsao/makerdiwo/tree/master/201604/PMS3003AirQualityV41

上述程式執行後，可以見到下圖之 PMS3003 空氣懸浮粒子感測器測試程式四畫面結果，也可以輕易讀到懸浮粒子的濃度了。

圖 5 PMS3003 空氣懸浮粒子感測器測試程式四畫面結果

本文為『PM2.5空氣感測器』系列第四篇：電路設計上篇，主要介紹空氣懸浮粒子感測器、LCD 2004顯示模組等電路安裝，並且一步一步教導讀者如何攥寫軟體來啟動這些元件，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_u0m1aotcA4hrRgQ

參考文獻：

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

曹永忠, 許智誠, & 蔡英德. (2015a). *Ameba 空气粒子感测装置设计与开发(MQTT 篇):Using Ameba to Develop a PM 2.5 Monitoring Device to MQTT* (初版 ed.). 台湾、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.). 台湾、彰化: 渥瑪數位有限公司.