

逢甲大學學生報告 ePaper

GPS 電路實作

GPS Circuit Implement

作者：陳躍升

系級：通訊工程學系三甲

學號：D9932275

開課老師：趙啟時

課程名稱：無線網路系統

開課系所：通訊工程學系

開課學年：101 學年度 第 1 學期



中文摘要

(1)目的：

主要實踐課堂上老師所授予我們的內容，以及了解 GPS 相關電路和訊息的接收和處理。

(2)過程及方法：

此次 GPS 電路是以 Arduino UNO 這塊板子來做為它的中心處理器，並以一塊 GPS 模組接上一條天線用來接收 GPS 訊號，將收到的 GPS 訊號送到 Arduino UNO 板子在由此板子把訊號經過字串處理後經由電腦的 Serial Port 使電腦螢幕顯示 GPS 相關訊息，以及送往在電路板上的 LCD 顯示器，顯示時間以及經緯度，來完成定時以及定位功能。

(3)結果：

這次的結果主要分成兩部分，第一部分是沒有經過字串處理的訊息，完全的把從 GPS 收到的訊號完整地呈現在電腦螢幕，像是 GPRMC 等等封包；第二部分則是有經過字串處理，由 Arduino UNO 板子將所收到的 GPS 訊息藉由判斷封包的名子來了解它的資料結構，並將之把我們所需之訊息(時間、經緯度)切割出來並將它送往電腦以及 LCD 顯示器上。

第一部分所獲得的結果都是衛星傳送給電路的直接訊息要靠了解它封包的資料結構才能完成，且 LCD 因為螢幕的限制並無法正確顯示；第二部分的結果是經過字串處理，可以很直觀的了解現在時間以及經緯度，且能顯示在 LCD 顯示器上。

關鍵字：

- 1.訊號處理
- 2.Arduino UNO
- 3.Fritzing
- 4.Global Positioning System (GPS)

Abstract

1. Purpose:

Mainly practice teacher granted in class, and to realize the GPS associated circuitry and signal receive 、 process.

2. Procedure and Method:

This GPS circuit is based on the board of Arduino UNO, Arduino UNO is the central processor of board. Use a GPS module connecting an antenna for receiving GPS signal. Then sent the receiving GPS signal to the Arduino UNO board which use the signal to do string handling, finally the board sent the processed signal through PC's serial port making the PC screen display relative signal of GPS, another monitor is the on board LCD module, display time, latitude and longitude.

3. Result:

The result mainly divided into two parts, first part the signal is not string handling, sent the fully GPS signal making PC screen display, likes the packet of GPRMC ; second part the signal is string handling, Arduino UNO board sent the receiving GPS signal by determine the name of packets and then realize it's data structure, finally display on PC monitor and LCD module the useful information likes time, latitude and longitude.

First part the result are all the satellite sent the signal to the circuit and fully done by realize the packet's data structure, and LCD module has limit due to it's monitor ; second part the result is string handling, it can direct realize about the time latitude and longitude, and can display on the LCD module.

Keyword :

- 1.Signal Process
- 2.Arduino UNO
- 3.Fritzing
- 4.Global Positioning System (GPS)

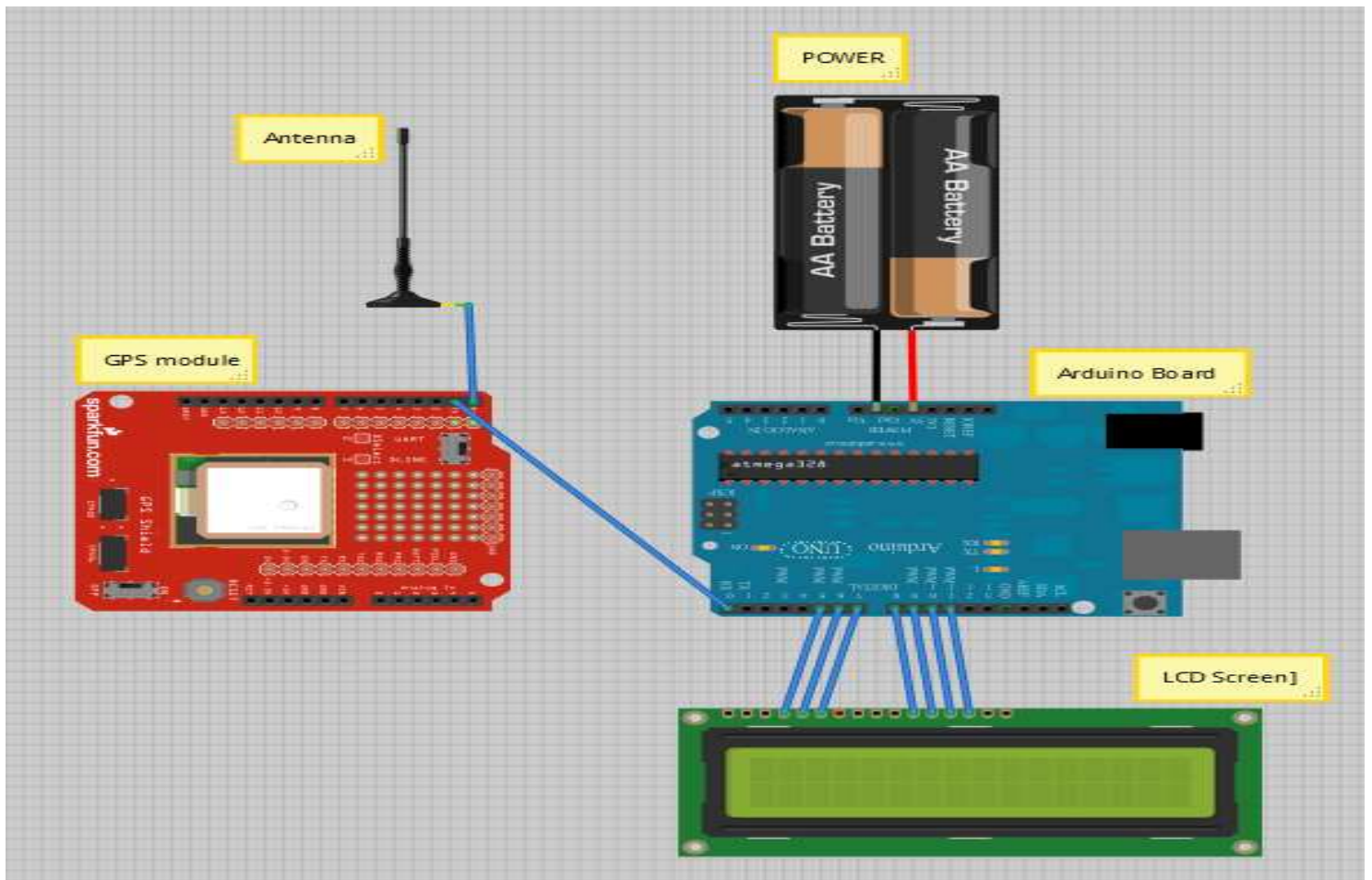
目 次

一.電路圖	P5
二.接收訊號顯示	P6
三.程式碼	P11
四.成果展示	P17
五.心得	P25
六.GPS 資料搭配 GOOGLE MAP	P26
七.流程總結	P28

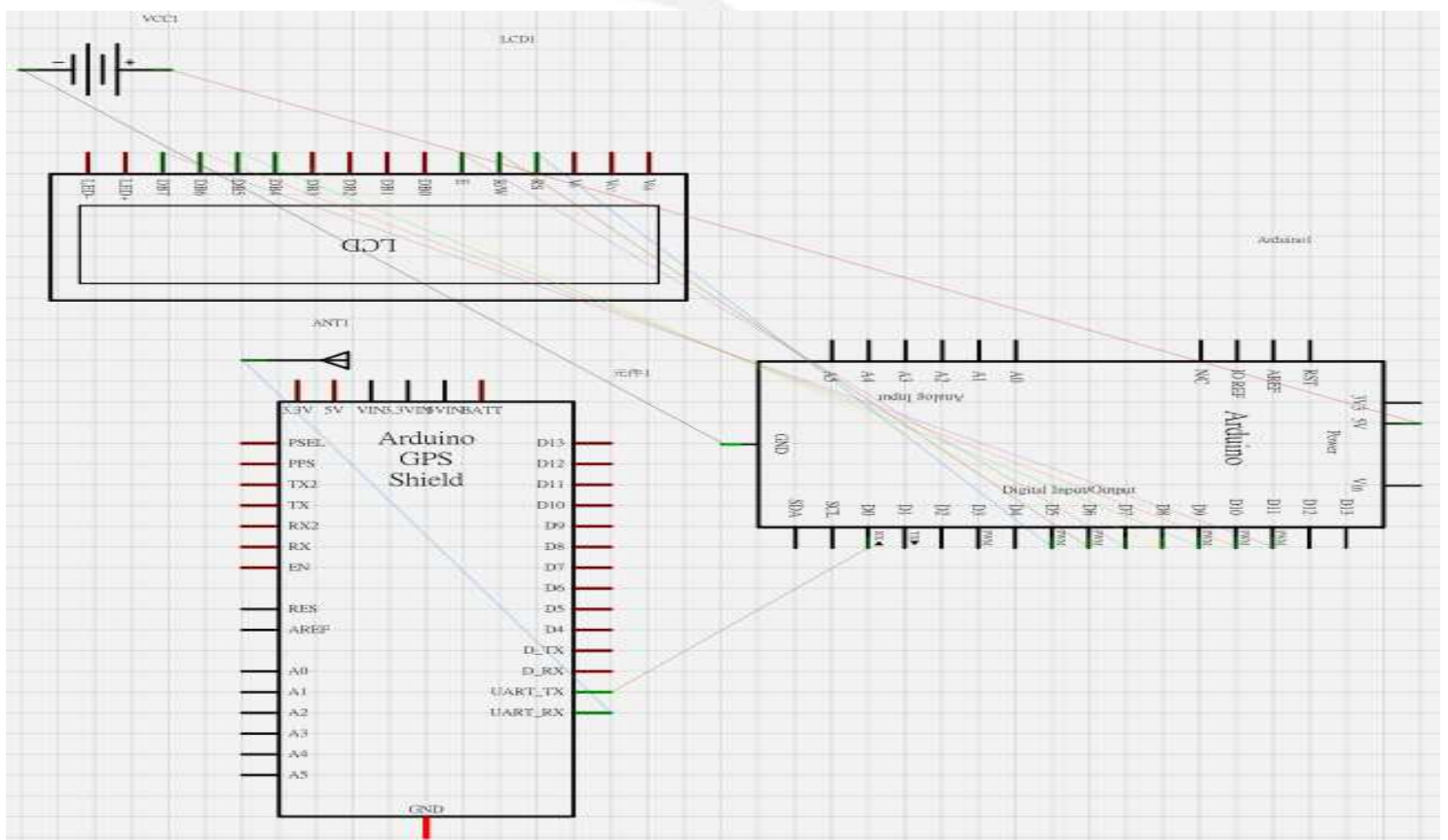


一. 電路圖

(1)使用 Fritzing 畫出示意圖



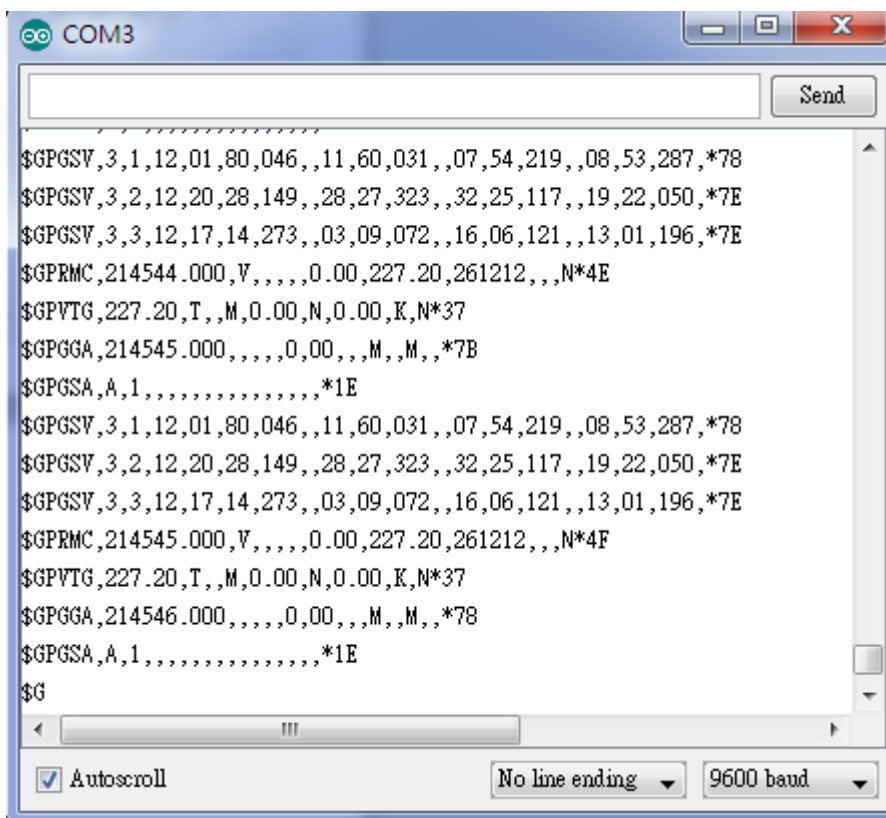
(2)使用 Fritzing 畫出概要圖



二. 接收訊號顯示

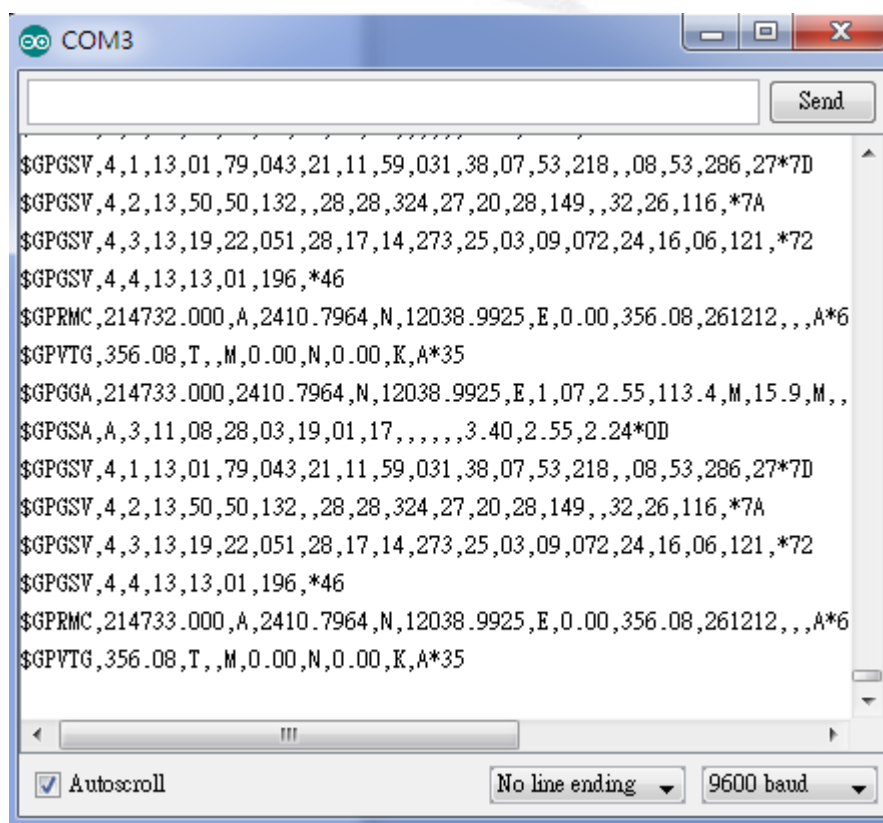
1. 使用 Arduino 編譯環境的 Serial Monitor

(1) 尚未處理字串(與衛星連線中)



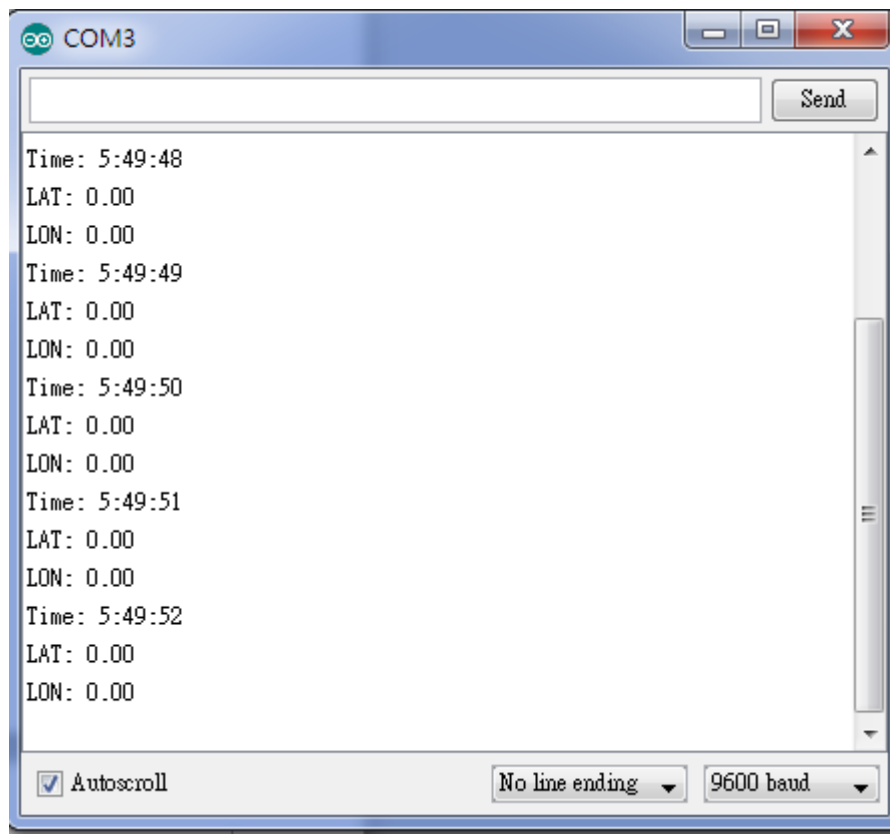
```
COM3
$GPGSV,3,1,12,01,80,046,,11,60,031,,07,54,219,,08,53,287,*78
$GPGSV,3,2,12,20,28,149,,28,27,323,,32,25,117,,19,22,050,*7E
$GPGSV,3,3,12,17,14,273,,03,09,072,,16,06,121,,13,01,196,*7E
$GPRMC,214544.000,V,,,,,0.00,227.20,261212,,,N*4E
$GPVTG,227.20,T,,M,0.00,N,0.00,K,N*37
$GPGGA,214545.000,,,,,0,00,,M,,M,,*7B
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,3,1,12,01,80,046,,11,60,031,,07,54,219,,08,53,287,*78
$GPGSV,3,2,12,20,28,149,,28,27,323,,32,25,117,,19,22,050,*7E
$GPGSV,3,3,12,17,14,273,,03,09,072,,16,06,121,,13,01,196,*7E
$GPRMC,214545.000,V,,,,,0.00,227.20,261212,,,N*4F
$GPVTG,227.20,T,,M,0.00,N,0.00,K,N*37
$GPGGA,214546.000,,,,,0,00,,M,,M,,*78
$GPGSA,A,1,,,,,,,,,,,,,*1E
$G
```

(2) 尚未處理字串(已連上衛星)

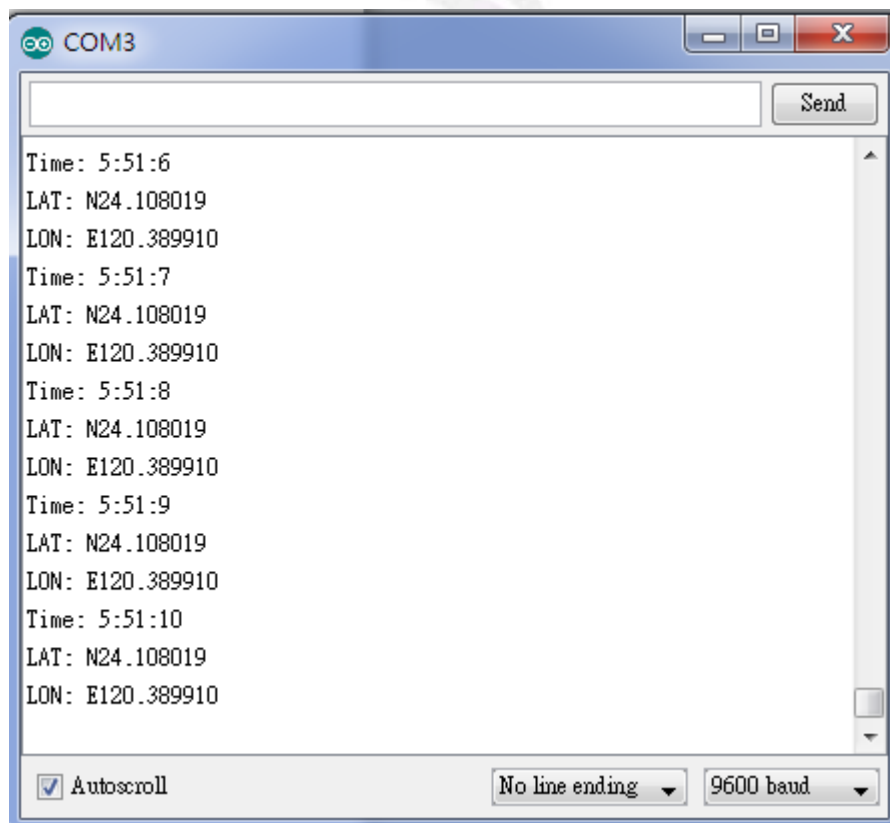


```
COM3
$GPGSV,4,1,13,01,79,043,21,11,59,031,38,07,53,218,,08,53,286,27*7D
$GPGSV,4,2,13,50,50,132,,28,28,324,27,20,28,149,,32,26,116,*7A
$GPGSV,4,3,13,19,22,051,28,17,14,273,25,03,09,072,24,16,06,121,*72
$GPGSV,4,4,13,13,01,196,*46
$GPRMC,214732.000,A,2410.7964,N,12038.9925,E,0.00,356.08,261212,,,A*6
$GPVTG,356.08,T,,M,0.00,N,0.00,K,A*35
$GPGGA,214733.000,2410.7964,N,12038.9925,E,1,07,2.55,113.4,M,15.9,M,,
$GPGSA,A,3,11,08,28,03,19,01,17,,,,,3.40,2.55,2.24*0D
$GPGSV,4,1,13,01,79,043,21,11,59,031,38,07,53,218,,08,53,286,27*7D
$GPGSV,4,2,13,50,50,132,,28,28,324,27,20,28,149,,32,26,116,*7A
$GPGSV,4,3,13,19,22,051,28,17,14,273,25,03,09,072,24,16,06,121,*72
$GPGSV,4,4,13,13,01,196,*46
$GPRMC,214733.000,A,2410.7964,N,12038.9925,E,0.00,356.08,261212,,,A*6
$GPVTG,356.08,T,,M,0.00,N,0.00,K,A*35
```

(3)經過字串處理(與衛星連線中)

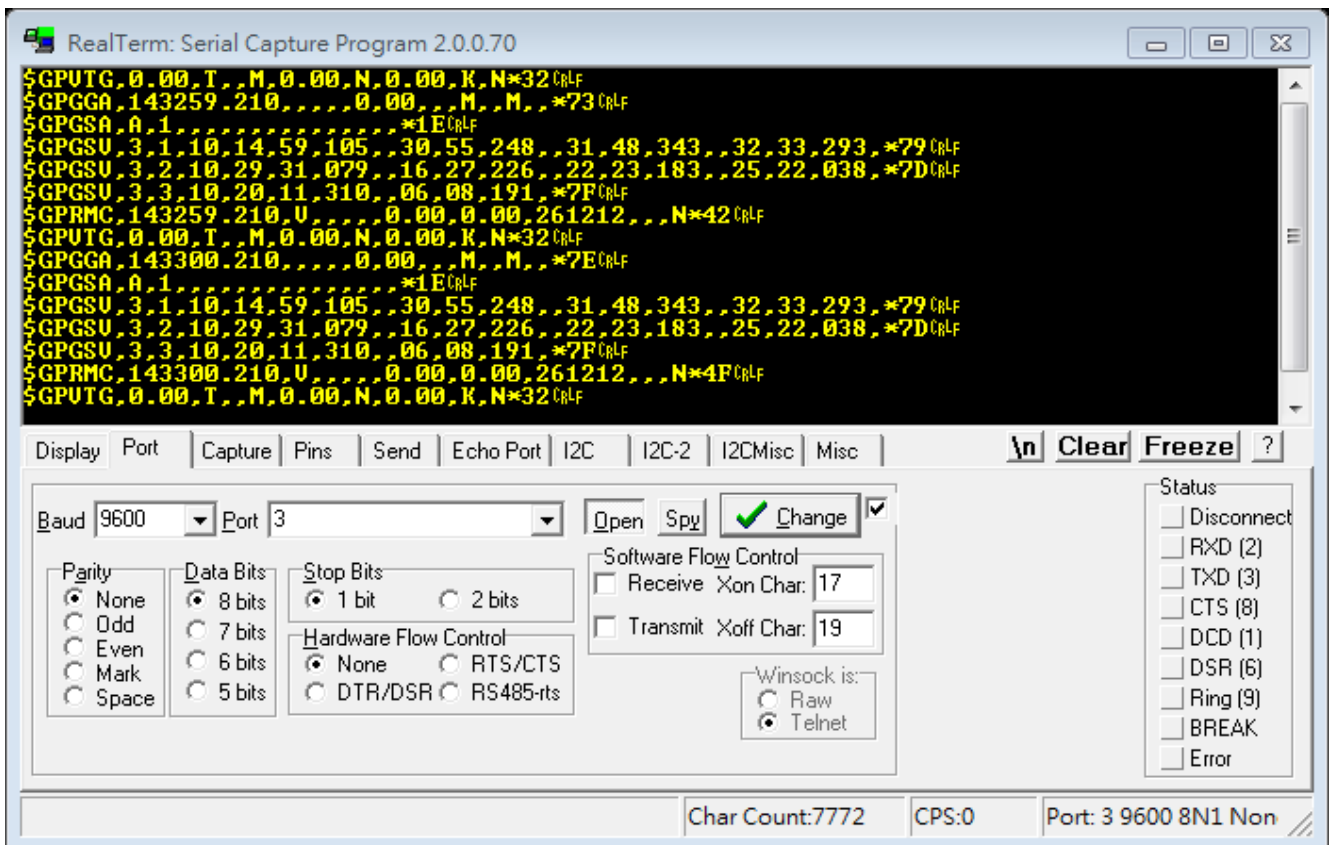


(4)經過字串處理(已連上衛星)

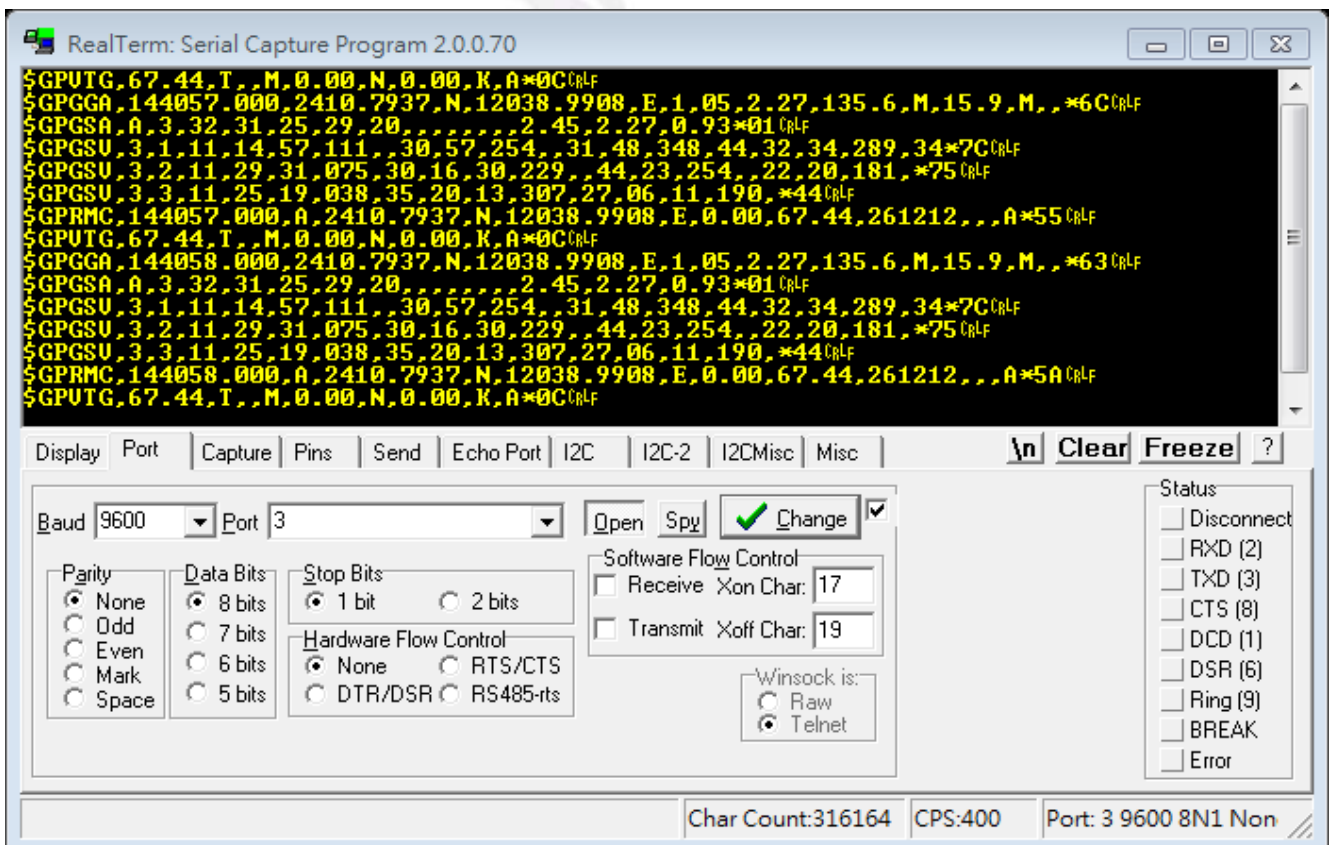


2. 使用 RealTerm

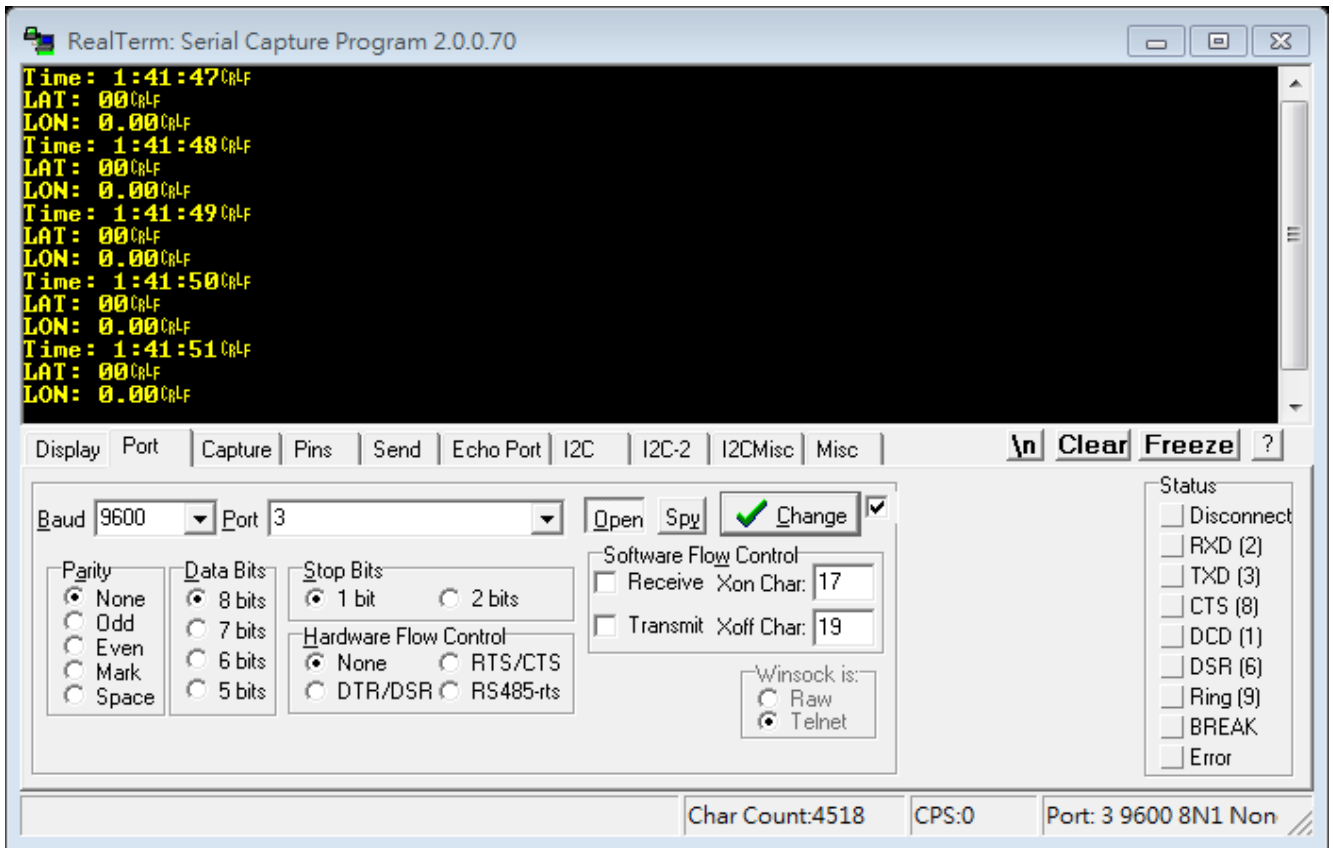
(1) 尚未處理字串(與衛星連線中)



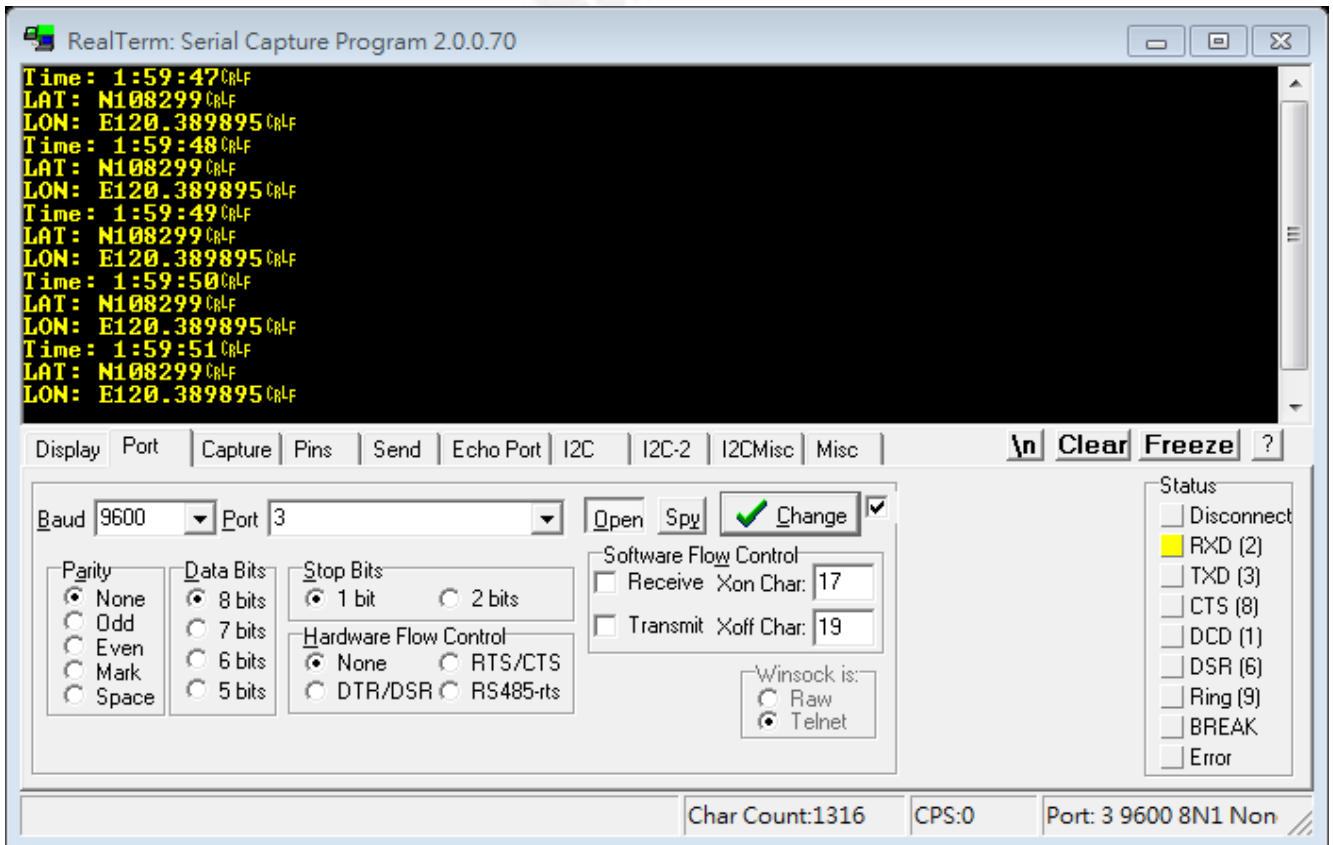
(2) 尚未處理字串(已連上衛星)



(3)經過字串處理(與衛星連線中)

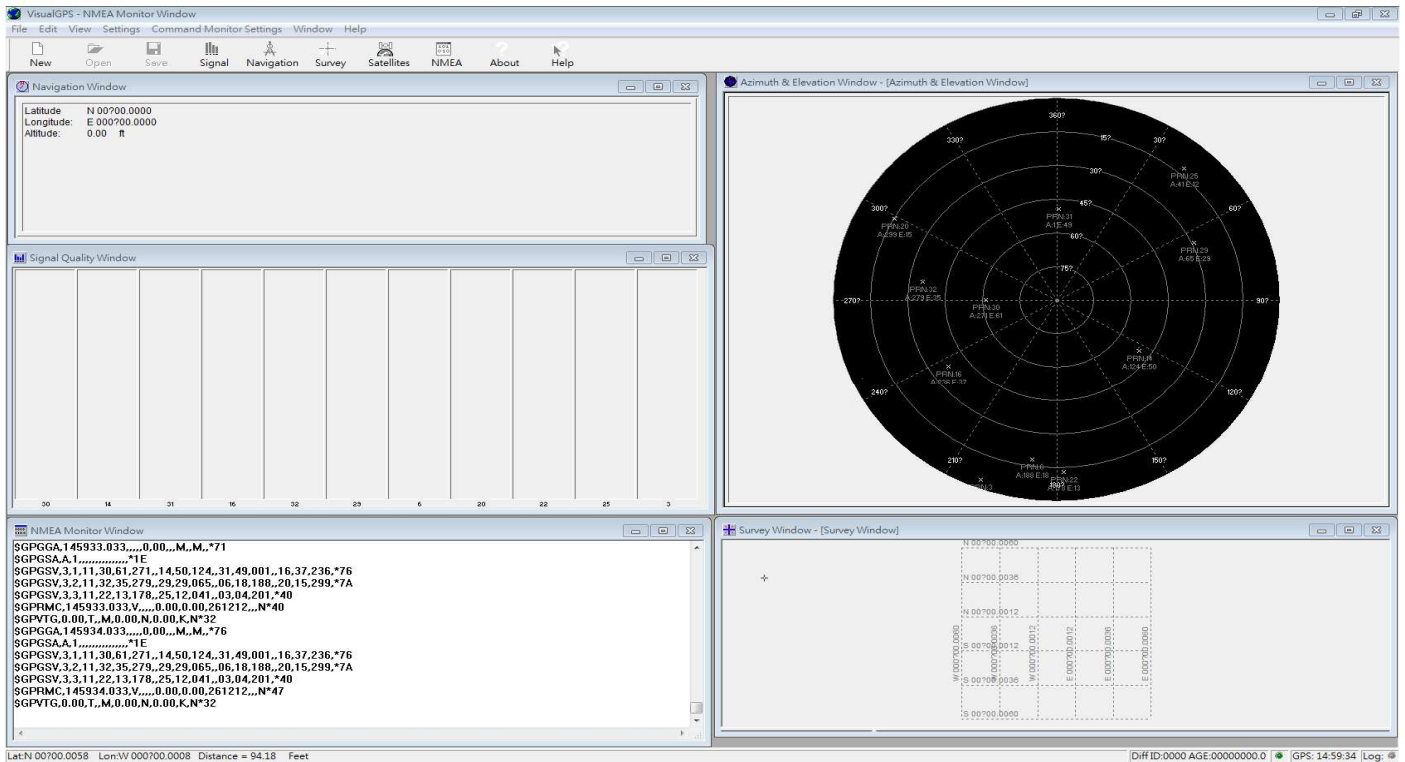


(4)經過字串處理(已連上衛星)



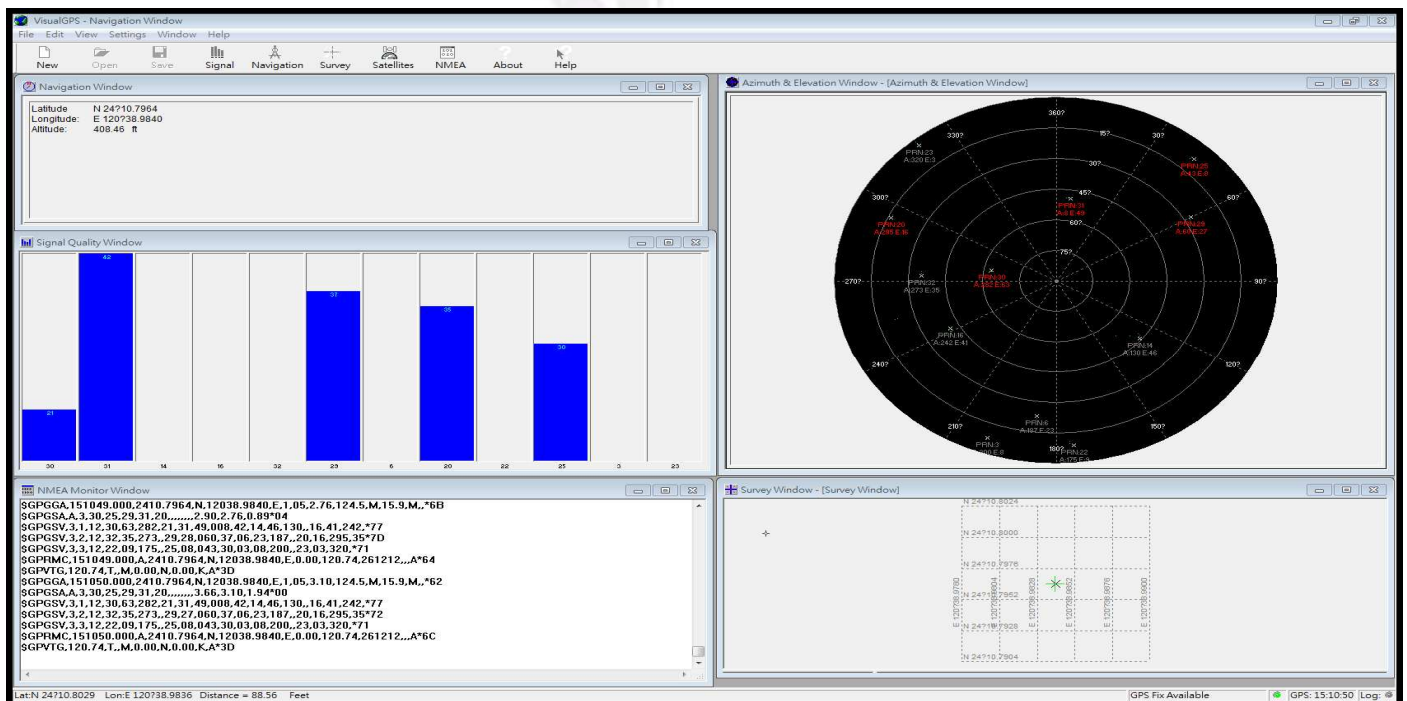
3. 使用 Visual GPS

(1) 與衛星連線中



上圖中的左下角視窗是正在與 GPS 連線中所收到的訊息

(2) 已連上衛星



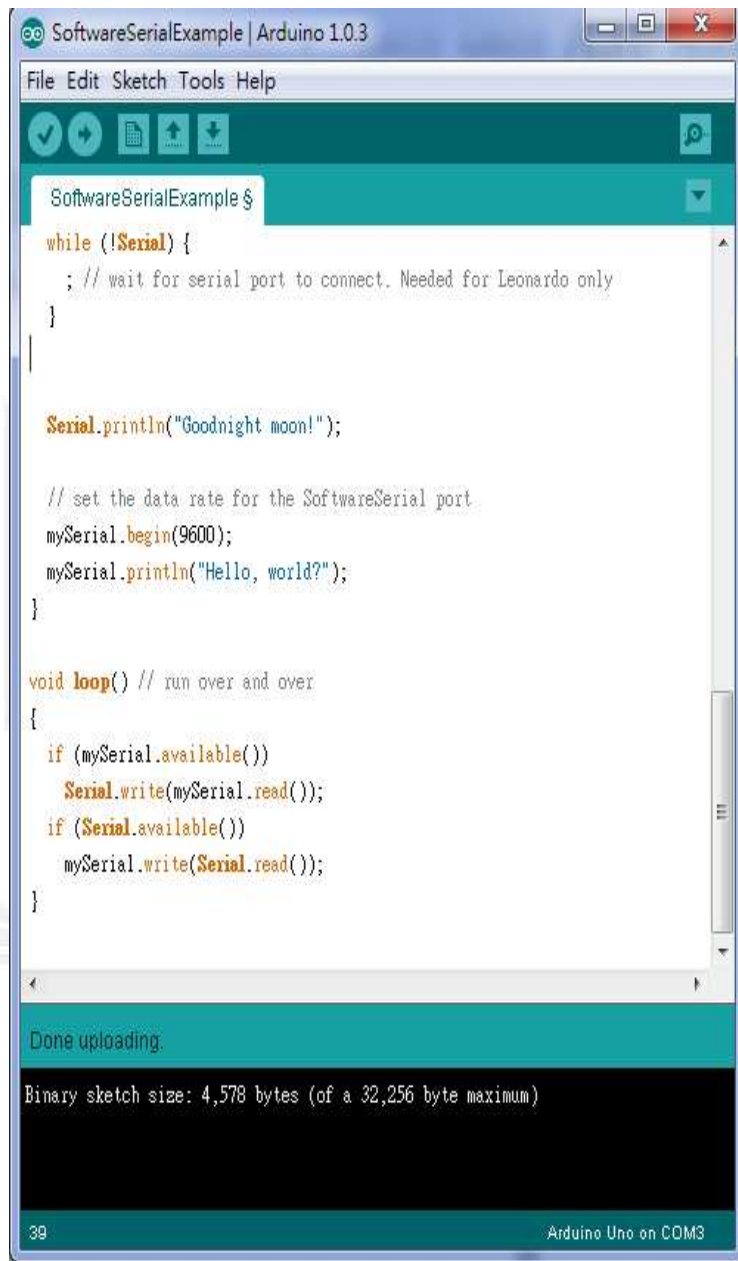
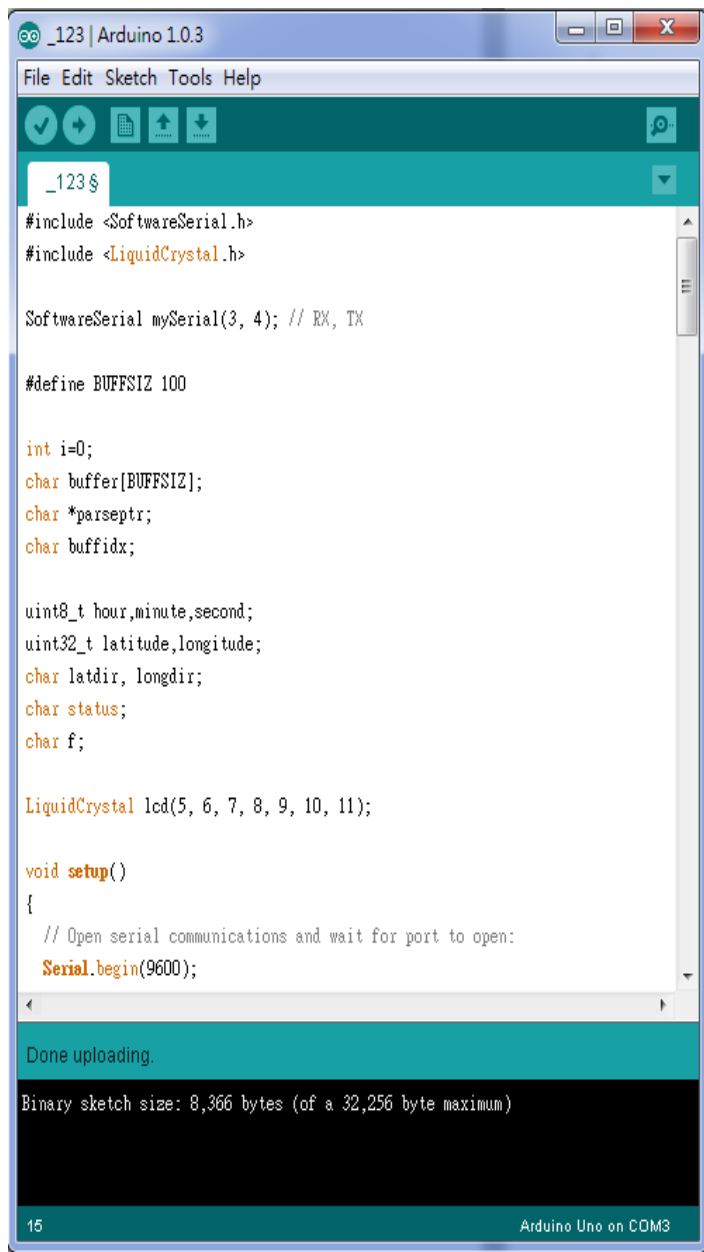
上圖中的左上視窗是已經抓到經緯度與海拔高度, 左中視窗是收到幾顆衛星的訊號強度(圖裡是5顆), 左下視窗是GPS的一些相關資訊, 右上視窗是視覺化圖形表示附近有幾顆衛星, 右下視窗是將經緯度用圖形表達出來

三. 程式碼

這次我使用的是 Arduino 板子當成核心處理單元, 是用 Arduino 1.0.3 版本的環境來編譯程式

Arduino 語言是一種類 C 語言(Like C)

1. Arduino 編譯環境



2. Arduino 程式碼

(1)單純收訊號不處理字串程式碼

```
#include <SoftwareSerial.h> //引用的函式庫
```

```
SoftwareSerial mySerial(3, 4); // 設定 Arduino 板子上 RX, TX
```

```
void setup()
```

```
{ //setup()只會在程式啟動後執行一次, 一般用來初始化程式相關參數及功能
```

```

// Open serial communications and wait for port to open:
Serial.begin(9600); //RS-232 電腦輸出鮑率設定
while (!Serial) {
  ; // wait for serial port to connect. Needed for Leonardo only
}
// set the data rate for the SoftwareSerial port
mySerial.begin(9600); //Arduino 鮑率設定
}

void loop() // run over and over
{ //loop()是主程式的所在, 會一直重複執行此段直到電源關閉或是 RESET
  if (mySerial.available()) //如果有讀到資料輸入就成立
    Serial.write(mySerial.read()); //透過 mySerial.read()將資料讀取, 並
    由
    //Serial.write()寫到螢幕
  if (Serial.available()) //如果電腦有輸出東西就成立
    mySerial.write(Serial.read()); //寫到 Arduino 上
}
////////////////////////////////////
////////////////////////////////////
(2)接收衛星訊息後再將訊息做字串處理程式碼
#include <SoftwareSerial.h>
#include <LiquidCrystal.h> //使用針對 LCD 的函式庫

SoftwareSerial mySerial(3, 4); // 設定 Arduino 板子上 RX, TX

#define BUFFSIZ 100 //字串陣列長度

int i=0;
char buffer[BUFFSIZ]; //GPS 接收暫存字串
char *parseptr; //字串指標變數
char buffidx; //暫存字串索引

uint8_t hour, minute, second; //時間相關變數
uint32_t latitude, longitude; // 經度, 緯度
char latdir, longdir; //南北緯 東西經
char status;
char f;
LiquidCrystal lcd(5, 6, 7, 8, 9, 10, 11); //初始化 LCD 腳位
// LCD 在 Arduino 中的宣告方式:

```

```
//LiquidCrystal(RS, RW, Enable, D4, D5, D6, D7)
```

```
void setup()
```

```
{ //setup()只會在程式啟動後執行一次,一般用來初始化程式相關參數及功能
```

```
  // Open serial communications and wait for port to open:
```

```
  Serial.begin(9600); //RS-232 電腦輸出鮑率設定
```

```
  while (!Serial) {
```

```
    ; // wait for serial port to connect. Needed for Leonardo only
```

```
  // set the data rate for the SoftwareSerial port
```

```
  mySerial.begin(9600);
```

```
  lcd.begin(16, 2); //宣告 LCD 大小
```

```
}
```

```
void loop() // run over and over
```

```
{//loop()是主程式的所在,會一直重複執行此段直到電源關閉或是 RESET
```

```
  uint32_t tmp;
```

```
  readline(); //讀取每一行的 NMEA 資料
```

```
  if (strncmp(buffer, "$GPRMC", 6)==0) //由 GPRMC 字串擷取資料
```

```
//GPRMC 封包說明:基本的 GNSS 資訊
```

```
//GPRMC 封包格式
```

```
/*$GPRMC, 090902.000, A, 2410.8189, N, 12038.9887, E, 0.00, 359.21, 241212, , , A
```

```
*6F
```

訊息範例	敘述說明
\$GPRMC	RMC 訊息標頭
090902.000	標準定位時間(時時分分秒秒.秒秒秒)
A	定位狀態(A=資料可用, V=資料不可用)
2410.8189	緯度(度度分分.分分分分)
N	北半或南半球指示(北半球(N)或南半球(S))
12038.9887	經度(度度度分分.分分分分)
E	東半或西半球指示(東半球(E)或西半球(W))
0.00	對地速度(0.0 至 1851.8 節)
359.21	對地方向(實際值(度))
241212	日期(日日月月年年)

```
*/
```

```
{ //hhmmss time data
```

```
  parseptr = buffer+7;
```

```
  tmp = parsedecimal(parseptr); //將字串轉換成數字
```

```
hour = (tmp /10000+8) % 24; //使用 2 位數表示時,分,秒
minute = (tmp/100)%100;    //小時+8 才是台灣時間
second=tmp%100;

parseptr = strchr(parseptr, ',') +1; //移到下一筆有用資訊起頭
status = parseptr[0];
parseptr += 2;

//latitude
latitude = parsedecimal(parseptr);
if (latitude != 0)
{
    latitude *= 10000;
    parseptr = strchr(parseptr, '.') +1;
    latitude += parsedecimal(parseptr);
}
parseptr = strchr(parseptr, ',') +1;
//read latitude N/S data
if (parseptr[0] != ',')
    latdir = parseptr[0];

//longitude
parseptr = strchr(parseptr, ',') +1;
longitude = parsedecimal(parseptr);
if (longitude != 0)
{
    longitude *= 10000;
    parseptr = strchr(parseptr, '.') +1;
    longitude += parsedecimal(parseptr);
}
parseptr = strchr(parseptr, ',') +1;
//read longitude E/W data
if (parseptr[0] != ',')
    longdir = parseptr[0];

//經由電腦顯示出來,時,分,秒
Serial.print("Time: ");
Serial.print(hour, DEC); Serial.print(':');
Serial.print(minute, DEC); Serial.print(':');
Serial.println(second, DEC);
```

```

//經由電腦顯示緯度
Serial.print("Lat: ");
if (latdir == 'N')
  Serial.print('N');
else if (latdir == 'S')
  Serial.print('S');

//將所讀到的緯度字串做處理, 並由電腦顯示
Serial.print(latitude/1000000, DEC); Serial.print(' ');
Serial.print((latitude/10000)%100, DEC); Serial.print(' ');
Serial.println((latitude%10000)*6/1000, DEC);

//經由電腦顯示經度
Serial.print("Long: ");
if (longdir == 'E')
  Serial.print('E');
else if (longdir == 'W')
  Serial.print('W');

//將所讀到的經度字串做處理, 並由電腦顯示
Serial.print(longitude/1000000, DEC); Serial.print(' ');
Serial.print((longitude/10000)%100, DEC); Serial.print(' ');
Serial.println((longitude%10000)*6/1000, DEC);

```

`i=second%2;`/*為了可以在 LCD 的有限螢幕上顯示時間跟經緯度, 所以使用變數 `i` 來使得幫秒數是偶數的時候就只顯示時間, 秒數是奇數的時候就顯示經緯度*/

```

if(i==0)
  { //秒數是偶數, 顯示時間
  lcd.clear();//清除 LCD 螢幕
  lcd.print("Time:");
  lcd.print(hour, DEC); lcd.print(' ');
  lcd.print(minute, DEC); lcd.print(' ');
  lcd.print(second, DEC);
  second=second++;
  }
else
  { //秒數是奇數, 顯示經緯度
  lcd.clear();//清除 LCD 螢幕

```



```

    lcd.print("LON:N"); lcd.print(latitude/1000000, DEC);
    lcd.print(".");
    lcd.print((latitude/10000)%100, DEC);
    lcd.print((latitude/10000)*6/1000, DEC);
    lcd.setCursor(0, 1);
    delay(50); //延遲 0.05 秒, 使 Arduino 有時間去處理資料
    lcd.print("LAT:E"); lcd.print(longitude/1000000, DEC);
    lcd.print(".");
    lcd.print((longitude/10000)%100, DEC);
    lcd.print((longitude/10000)*6/1000, DEC);
  }

}

void readline(void)
{//讀取每一行的 NMEA 資料
  char c;

  buffidx = 0; //start at beginning
  if(mySerial.available())
  {
    while(1)
    {
      c=mySerial.read();
      if (c == -1)
        continue;
      if (c == '\n')
        continue;
      if ((buffidx==BUFFSIZ-1) || (c == '\r')) //以結尾字元\r 做結
束
      {
        buffer[buffidx] = 0;
        return;
      }
      buffer[buffidx++] = c;
    }
  }
}

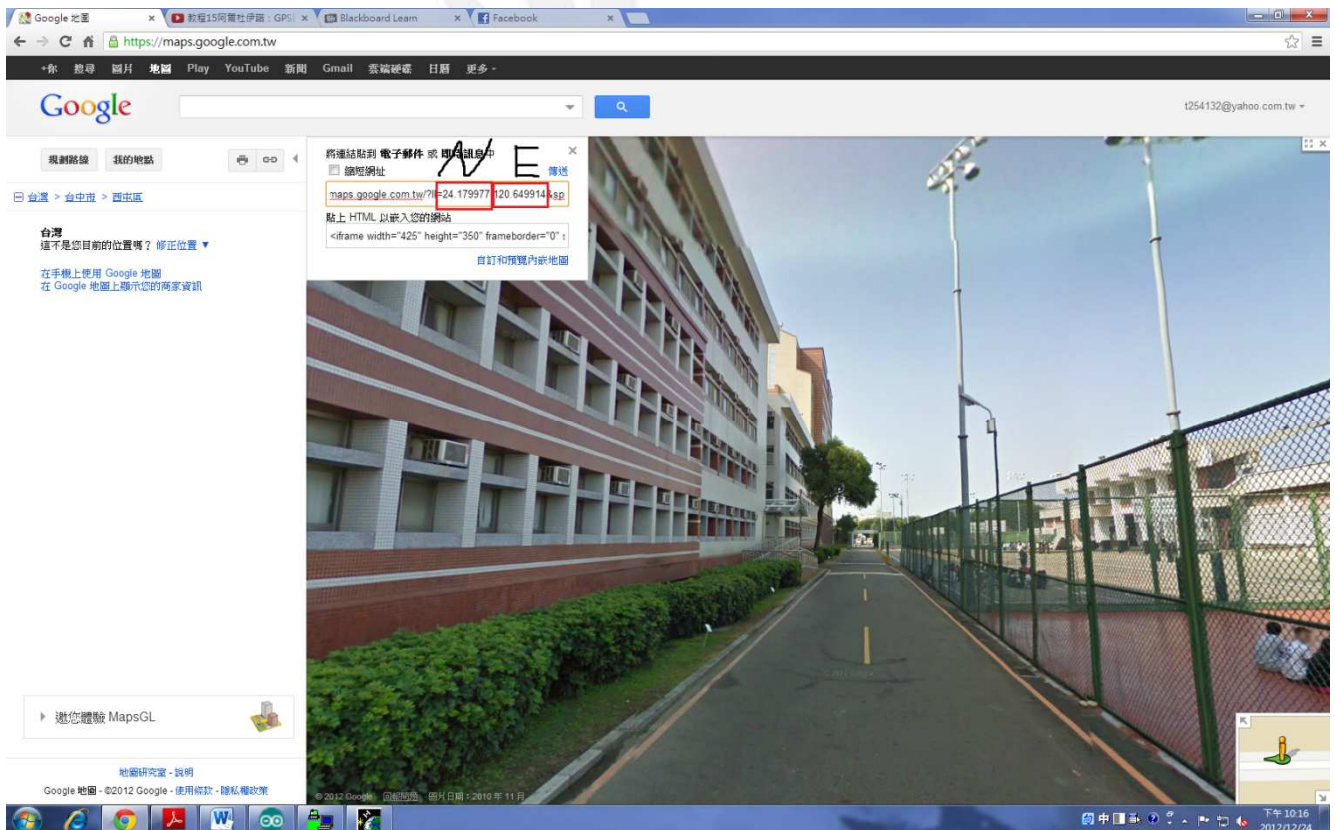
```

```
uint32_t parsedecimal(char *str) //轉換字串為字元
{
    uint32_t d = 0;

    while (str[0] != 0)
    {
        if ((str[0] > '9') || (str[0] < '0'))
            return d;
        d *= 10;
        d += str[0] - '0';
        str++;
    }
    return d;
}
```

四. 成果展示

1. 使用 GOOGLE MAP 找到電通館附近的大致位置



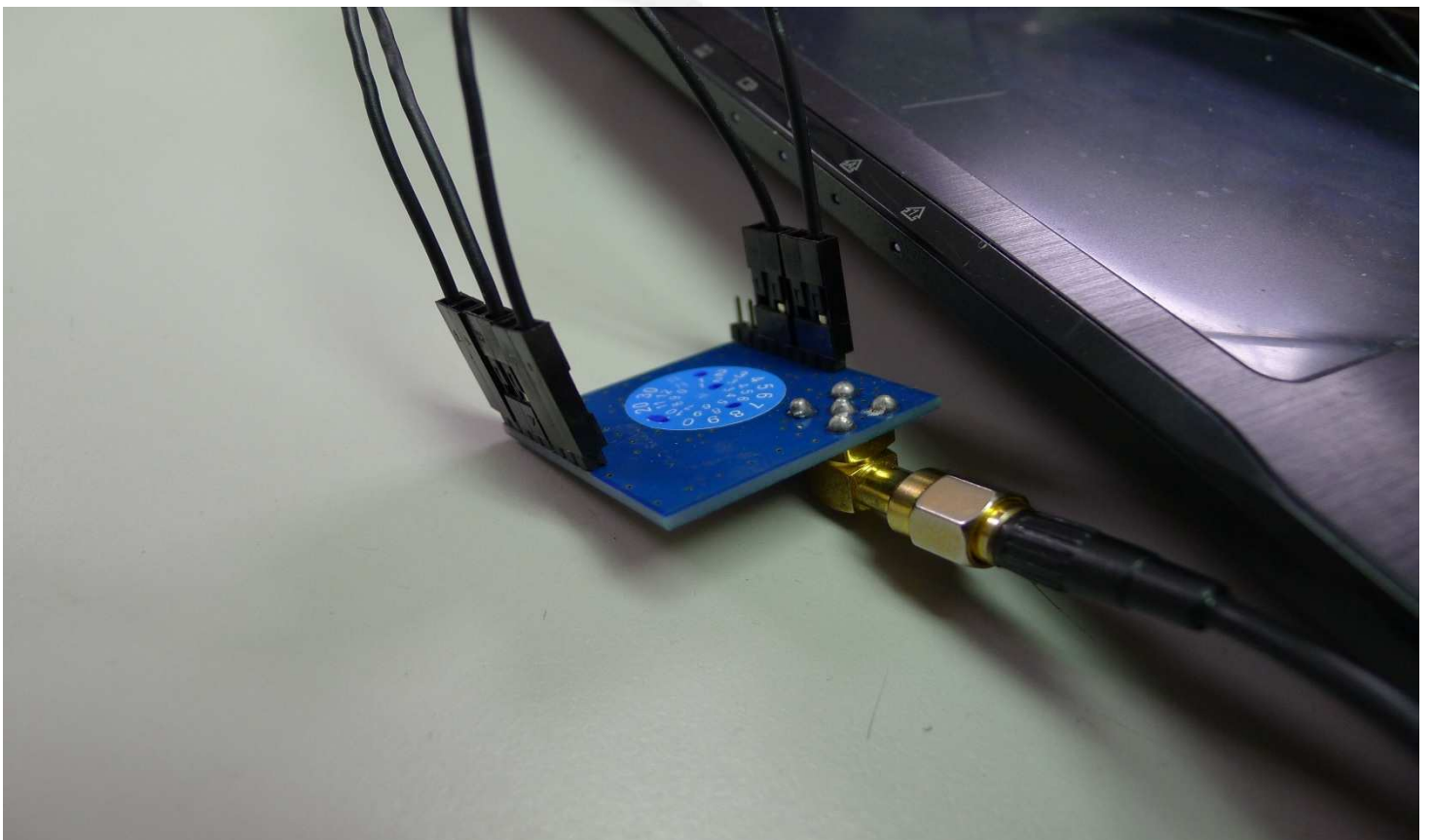
GPS 電路實作

2. 電路元件介紹

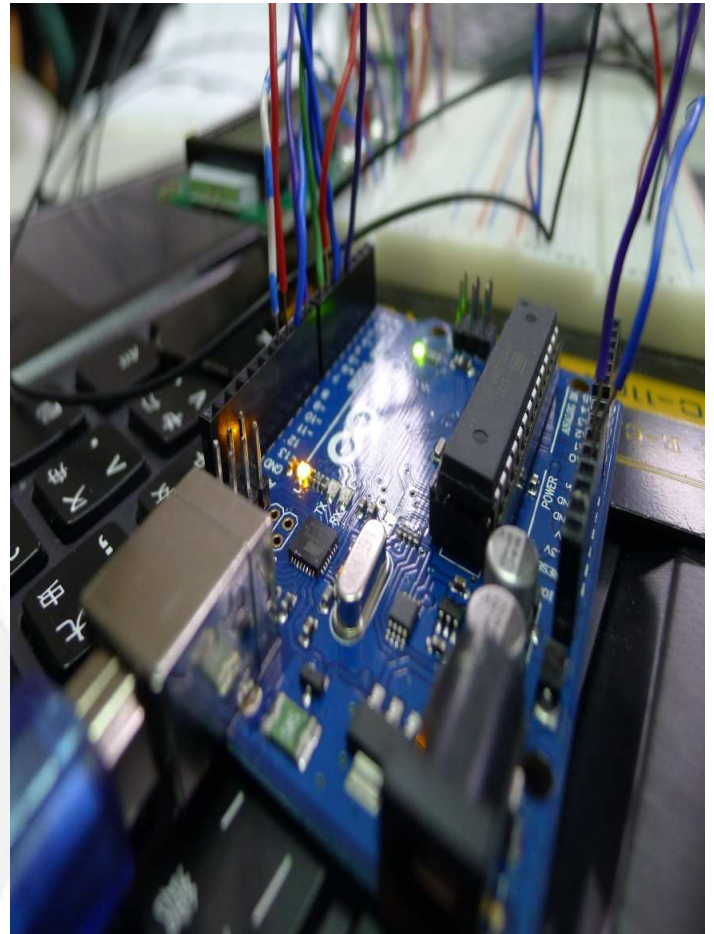
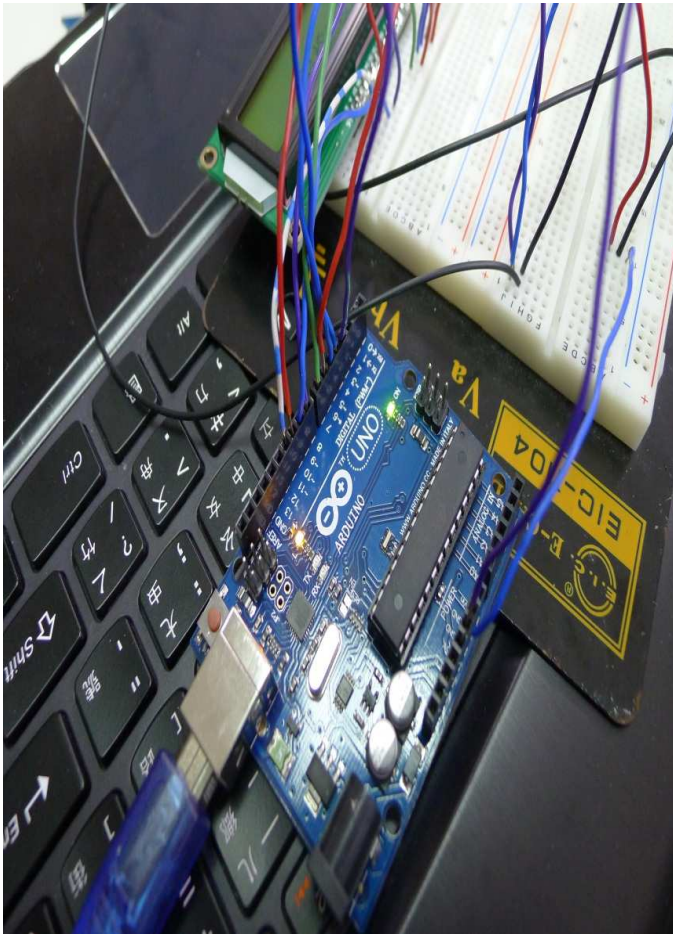
(1)天線:接收訊號用



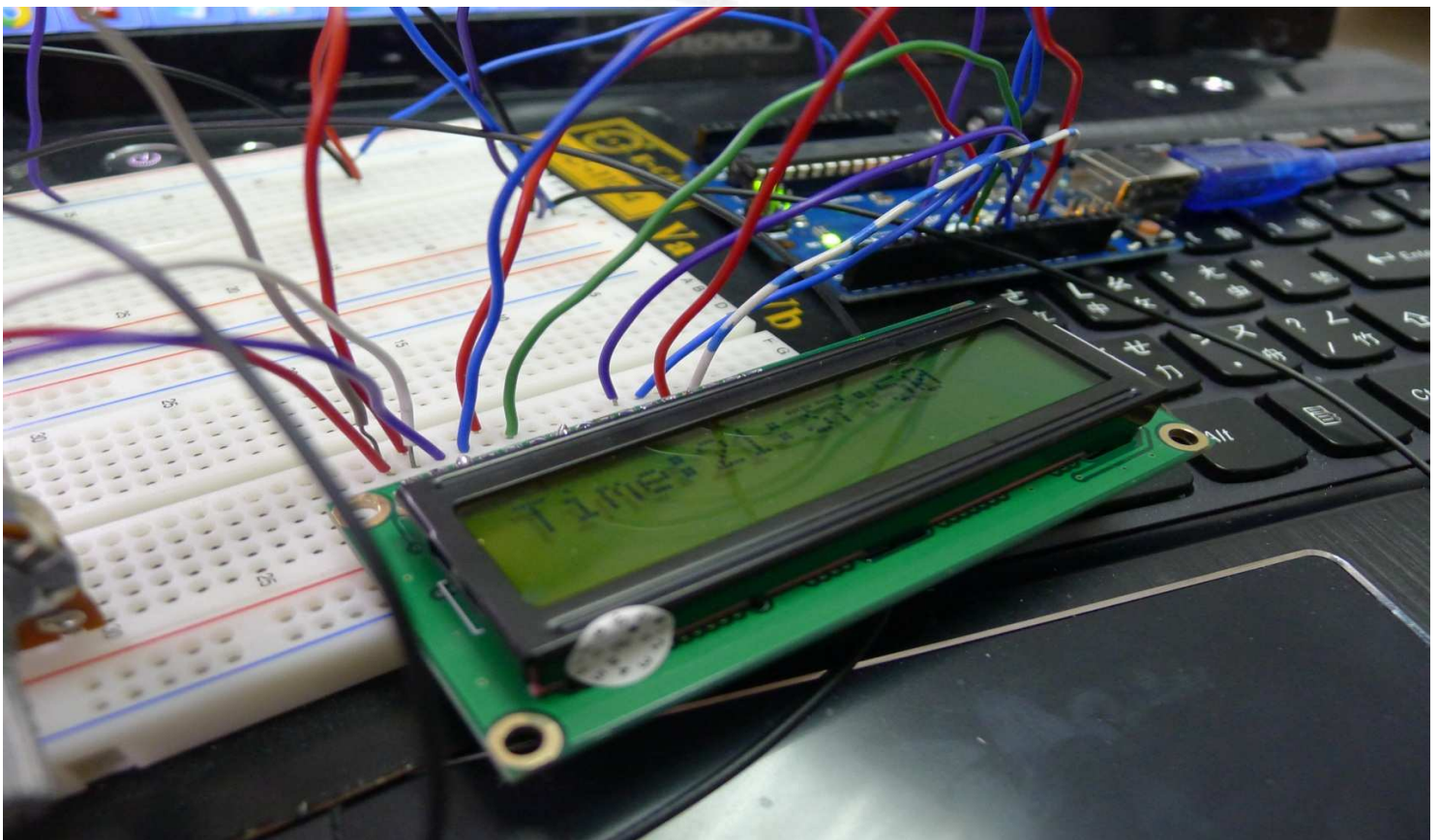
(2)GPS 模組:把天線收到的訊號擷取所需的部分(GPS)



(3)Arduino 板子:將 GPS 模組送來的訊號傳送至電腦



(4)LCD 以及可變電阻:LCD 是將處理過後的字串顯示,可變電阻則是用來調整 LCD 的螢幕亮度



3. 實際電路(未使用 LCD)

(1) GPS 原理大致說明

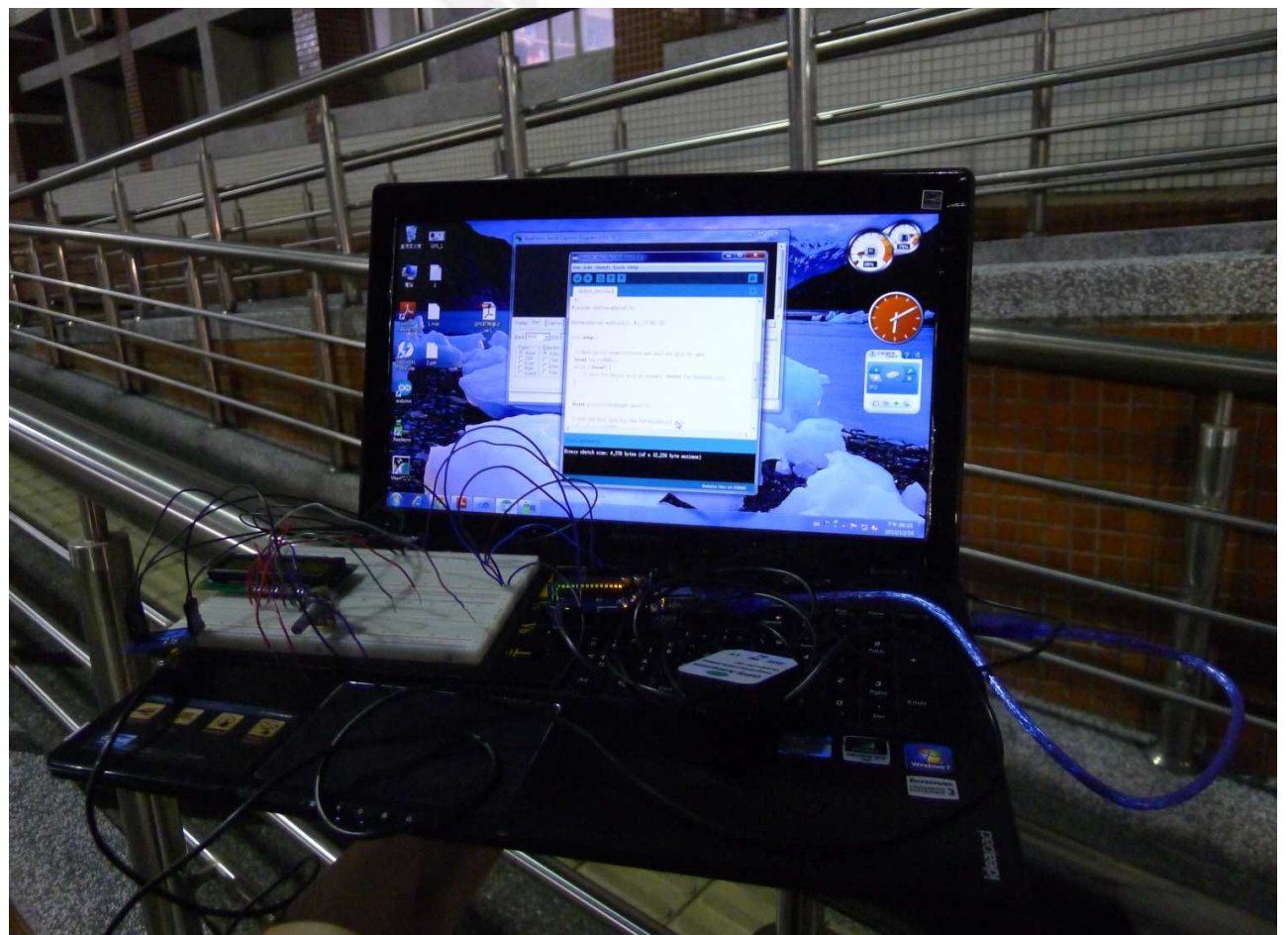
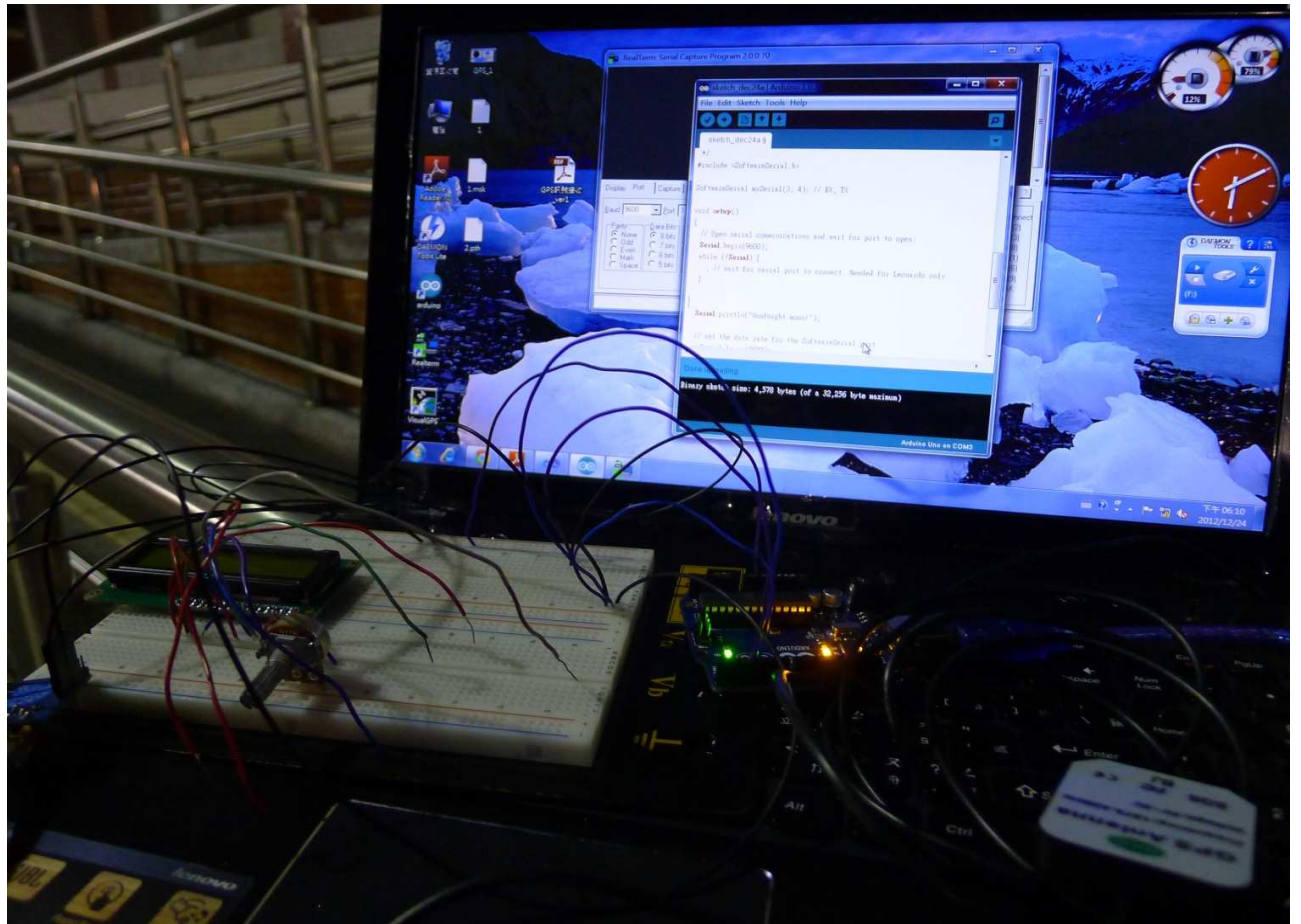
GPS (Global Position System), 全球定位系統, 有的人會把衛星(Satellite) 這個字加進去, 表示是經由衛星上的訊號來計算出接收機目前的位置所在. GPS 的衛星會一直發送訊號, 當接收機收到訊號時, 比對時間差在乘上光速, 就可以得到使用者和衛星的距離, 以衛星為圓心畫一個立體的球, 使用者可能的位置就在球面上的任一位置, 基本上至少 3 顆衛星才能定位成, 不過 GPS 系統通常還是以 4 顆衛星當作最低標準, 使用者必須接收機能夠收到 4 顆以上的資料時, 才可以有定位的功能.

(2) 電路區塊說明:

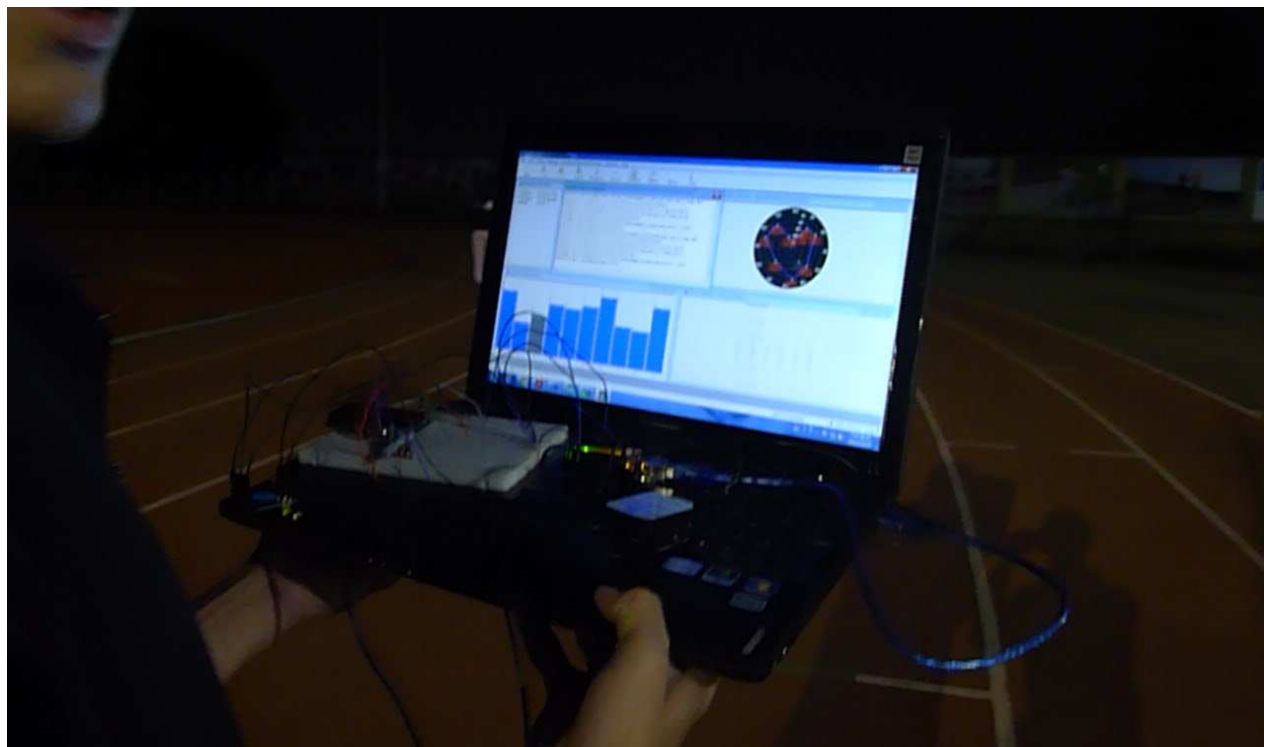
在 GPS 模組上加裝天線使之收衛星的訊號, 透過 GPS 模組(經由 TX)將訊號傳到 Arduino 板子(Arduino 板子使用 RX 收訊息)上, Arduino 板子(經由 TX)將收到的訊號經由 USB(使用 RS232)送往電腦(電腦是 RX), 再由電腦的顯示軟體將訊息顯示在螢幕, 此時尚未使用到 LCD, LCD 是在下一部份才有使用到, 是經由電腦收到的訊息後做字串處理, 擷取我們需要的訊息(時間, 經緯度), 最後在顯示 LCD 上.



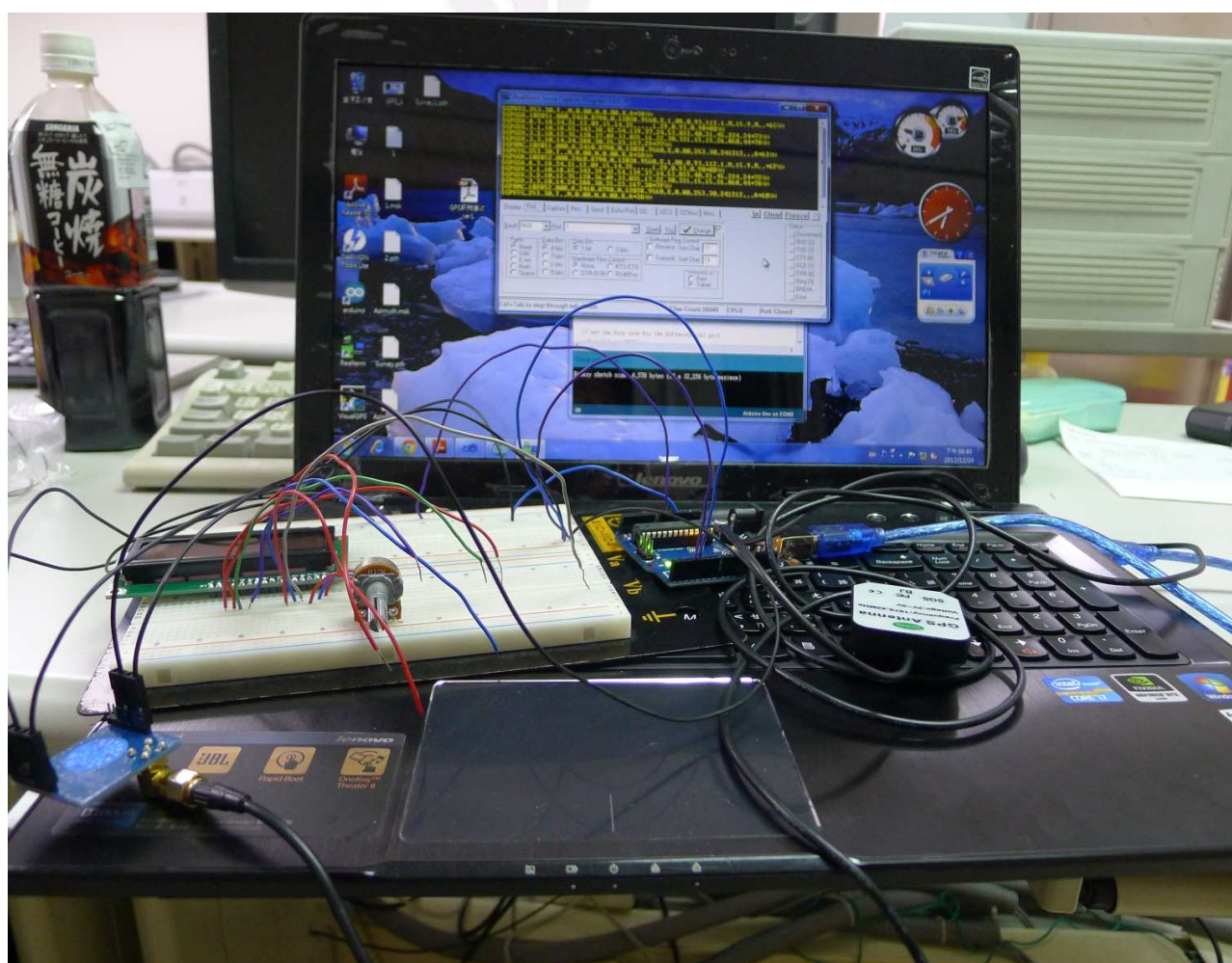
(3)在電通館門口量測訊號



(4)在操場量測訊號



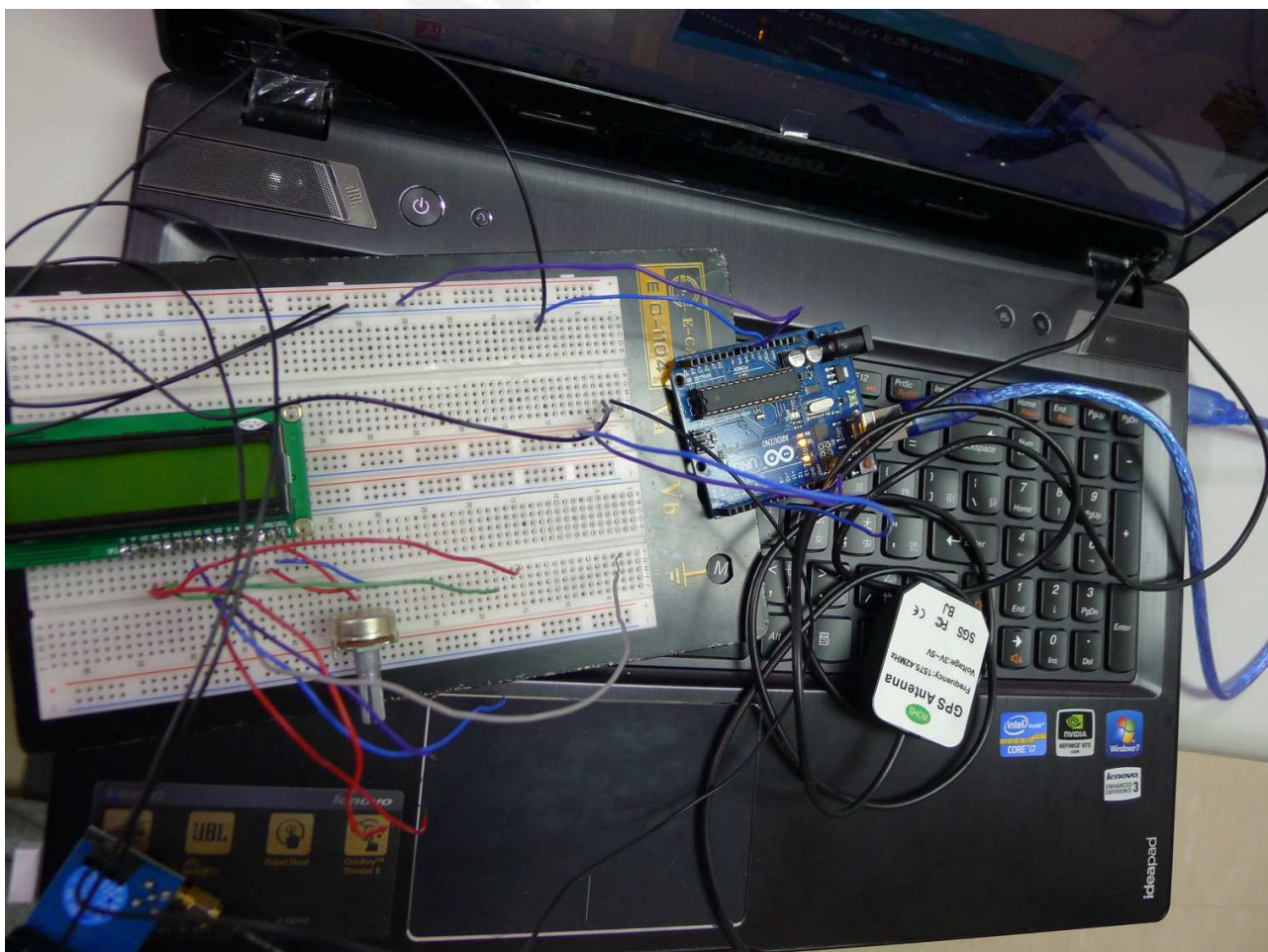
(5)在實驗室量測訊號_1



(5)在實驗室量測訊號_2

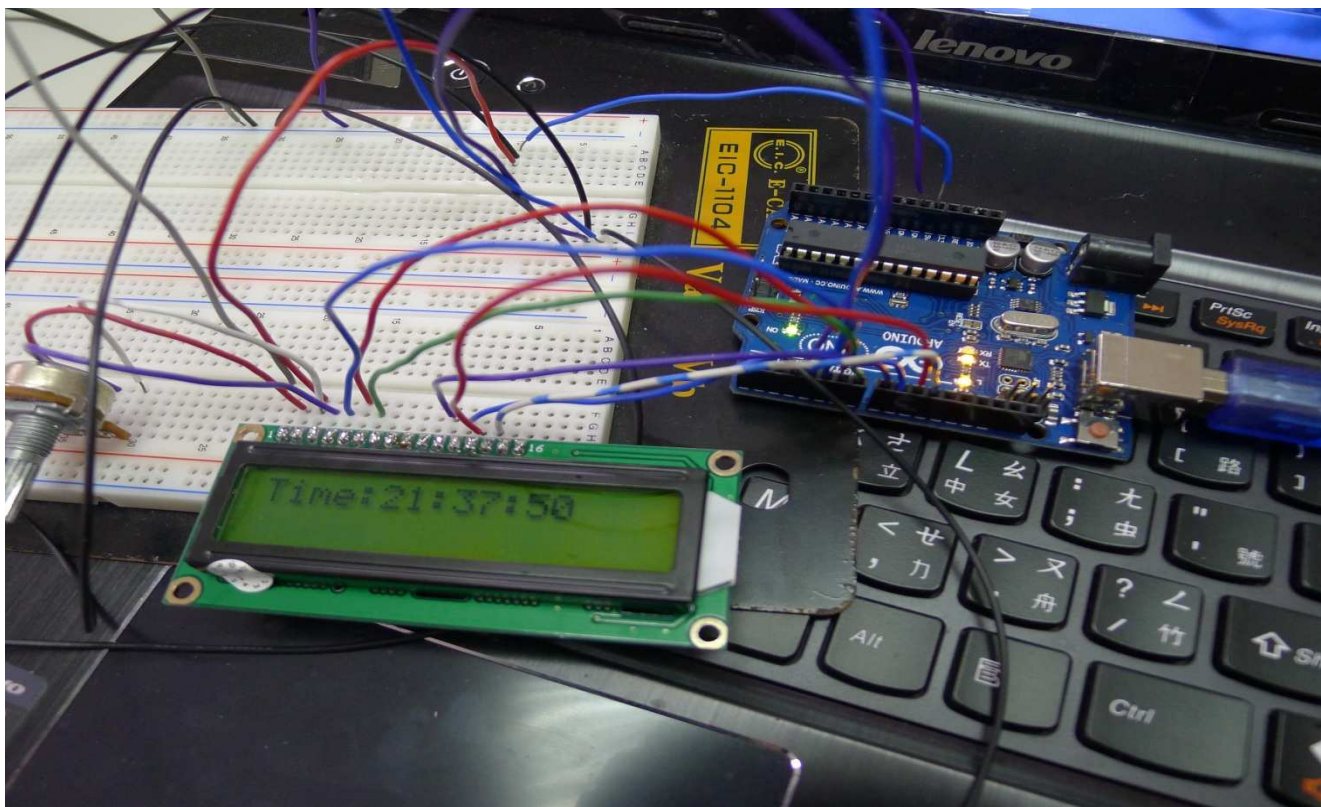


(6)全部元件圖

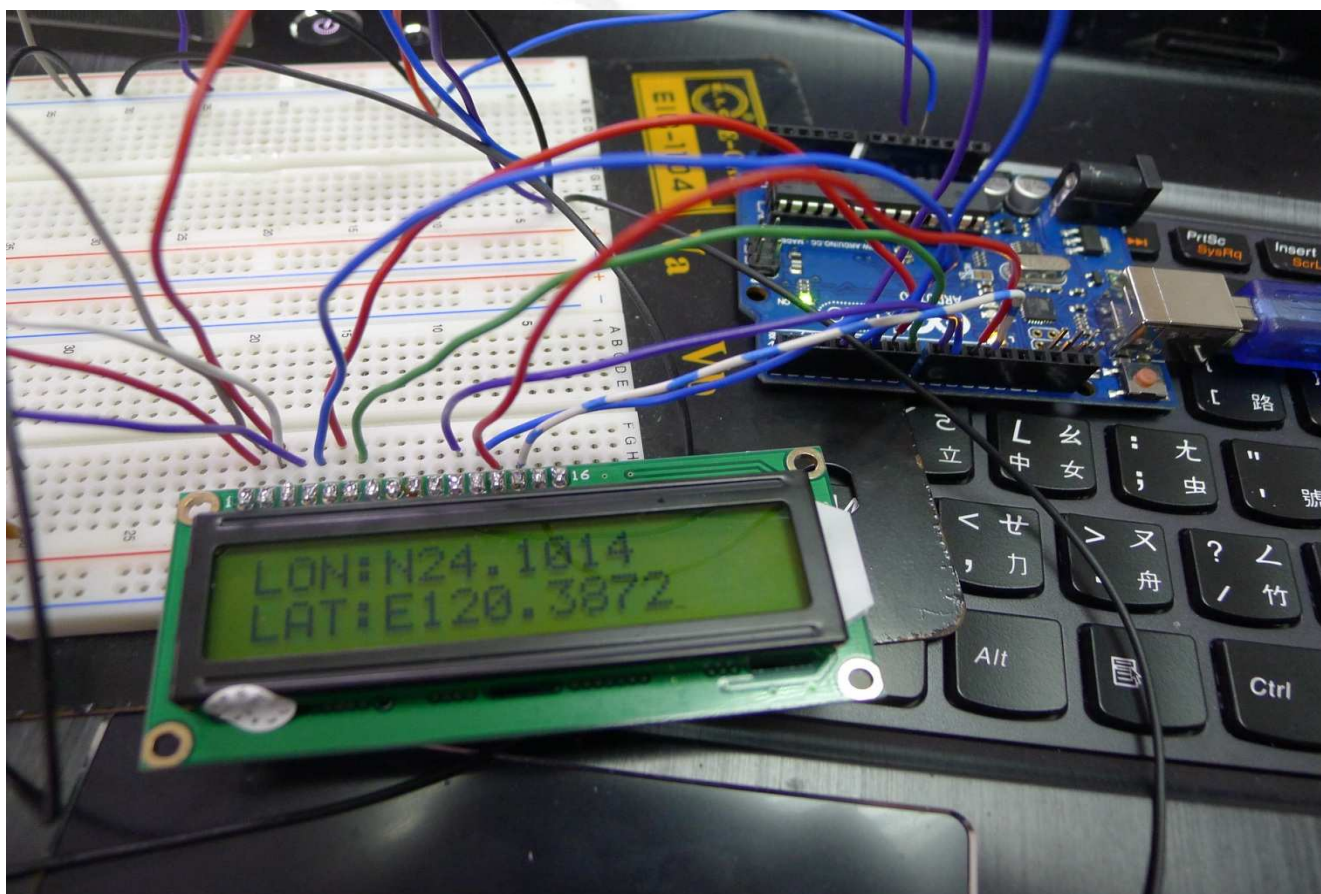


4. 實際電路(使用 LCD)

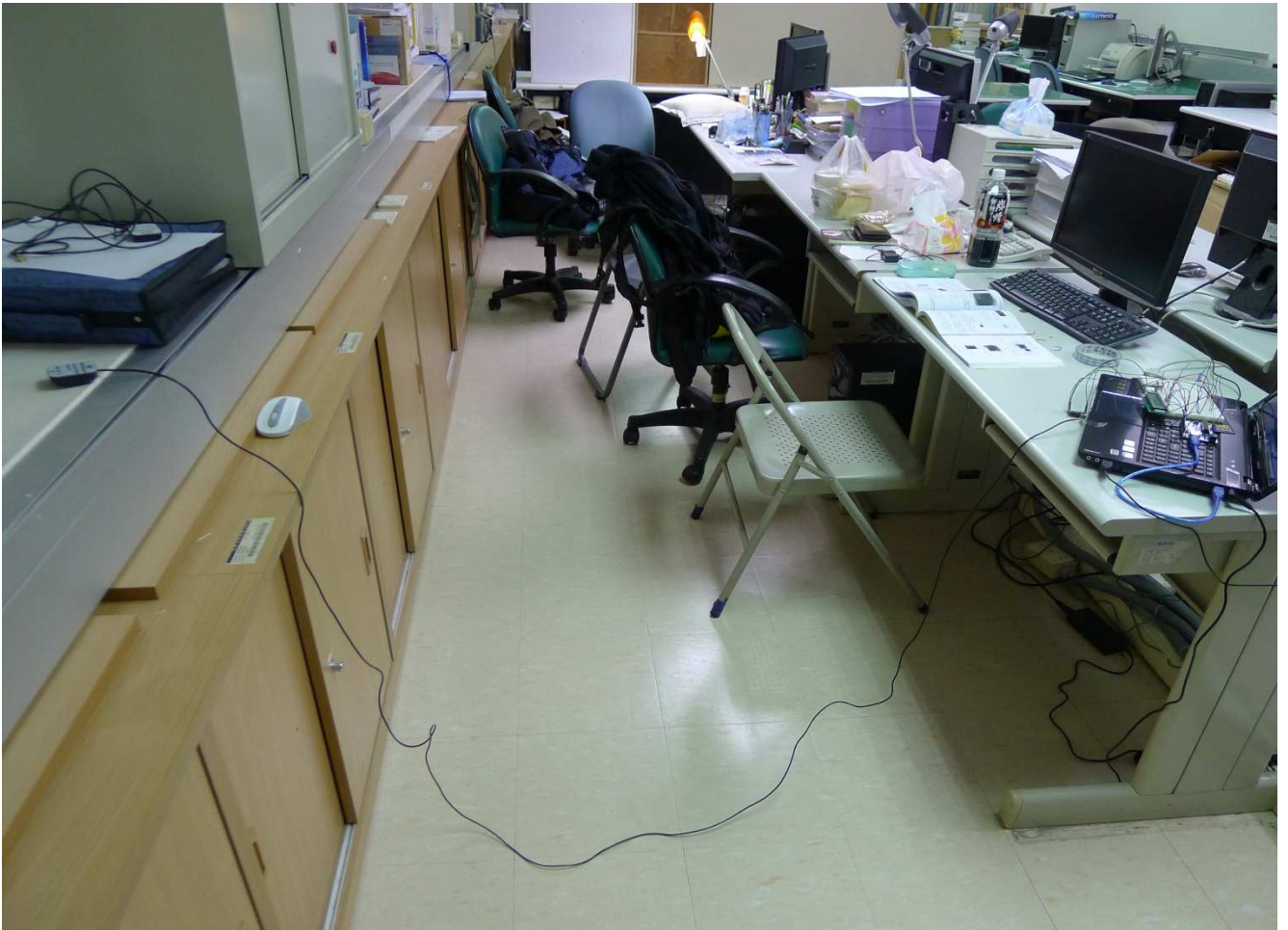
(1)顯示時間:衛星傳回來的時間是格林威治時間,所以要小時+8 才是台灣的時區



(2)顯示經緯度:通常要確定一個座標至少要有 3 顆衛星以上才準確,不過 GPS 系統通常是以 4 顆衛星當作最低標準,使用者必須在接收機能夠收到 4 顆衛星以上的資料時,才可以有定位的功能



(3)完整電路



上圖左邊是 GPS 天線, 右邊是電腦. 電路板以及 Arduino 板子和 LCD
由於 GPS 是透過衛星來定位因此只有在室外的環境才可以正確接收到 GPS 的訊號, 最簡單的方式是透過外加天線使收進來的訊號增強, 來達成室內成功接收 GPS 訊息

五. 心得

這次 GPS 的電路實際操作並沒有想像中的困難, 而且比較令我意外的是如果只是要接收 GPS 的訊號的話, 程式碼只要 11 行. 經過這次實作後除了讓我了解 GPS 相關的原理外, 也讓我接觸到 Arduino 板子, 並發現他是一塊功能強勁的板子, 還可以做更多的事情, 這次在處理字串上的需要較多的程式碼, 主要是要把收到眾多的 GPS 封包裡面把我們需要的封包擷取出來再將它進行切割拿取我們所要的資訊, 這方面就稍稍複雜了些, 總而言之這次的 GPS 電路實作是一個讓我蠻充實的作業, 獲益良多.

至於接下來應該會把 GPS 所收到的資料(經緯度)在上傳到 Google Map 上, 讓它不只是單純的顯示在電腦螢幕或 LCD 上, 希望能把收進來的訊息做更多應用, 像是記錄你所走的路徑並由 google map 去顯示你走過的路徑, 應該會蠻有趣的, 在之後會撥空來完成它的.

六. GPS 資料搭配 GOOGLE MAP (尚未完成)

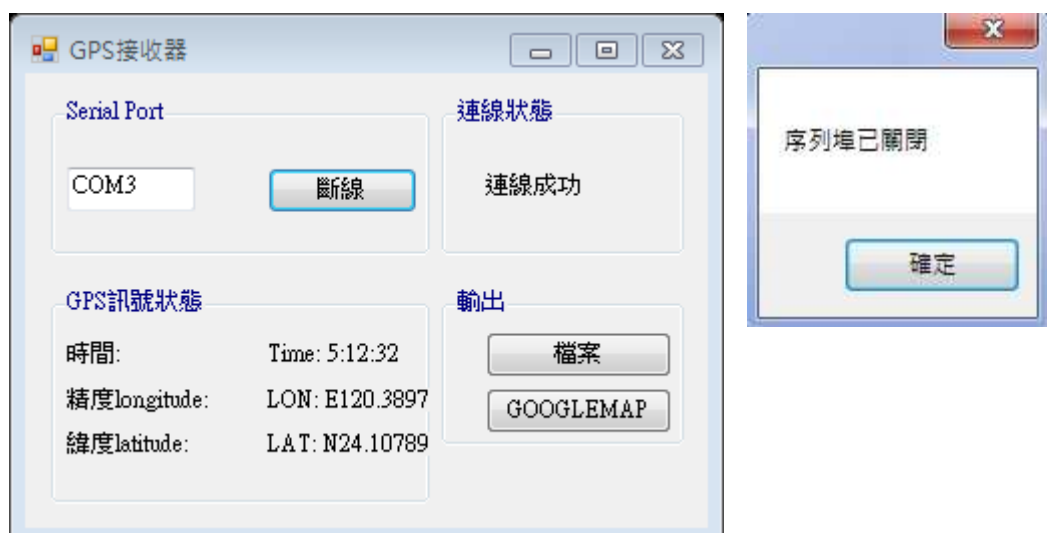
(1)



(2)



(3)



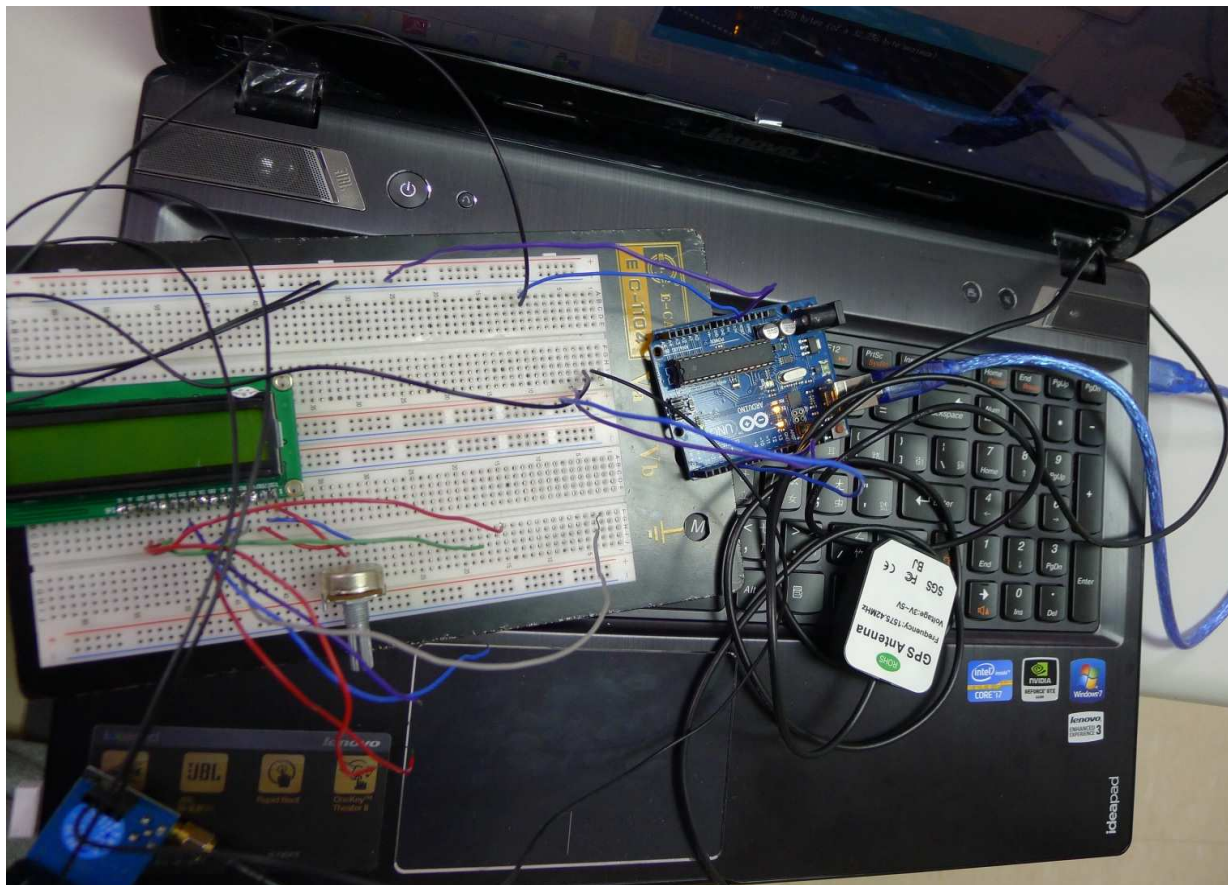
(4)



七. 流程總結

1. 未處理字串

(1) 電路實接(未接上 LCD)



(2) 編譯程式(未字串處理)_1

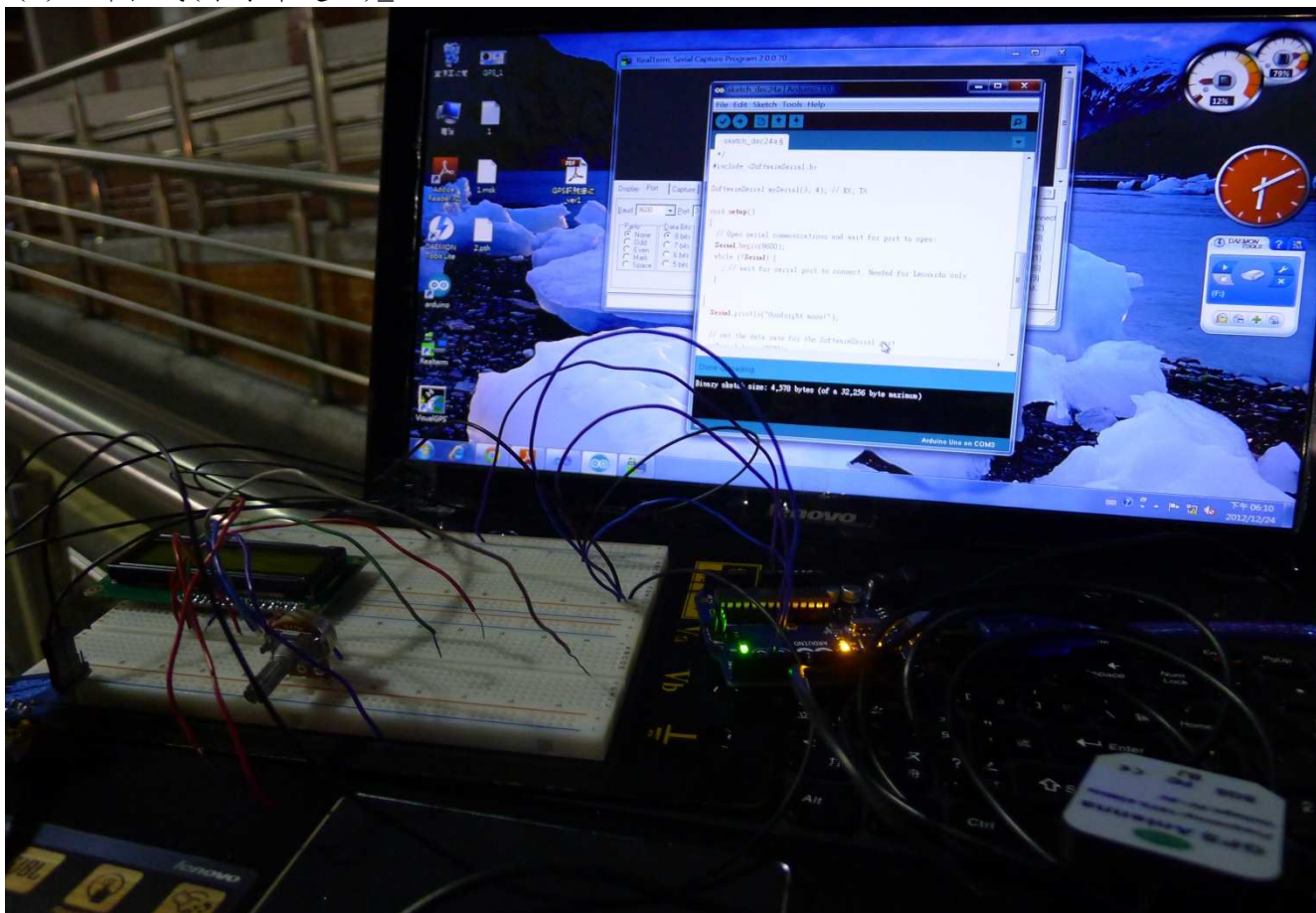
```
SoftwareSerialExample | Arduino 1.0.3
File Edit Sketch Tools Help
SoftwareSerialExample $
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  // set the data rate for the SoftwareSerial port.
  mySerial.begin(9600);
}

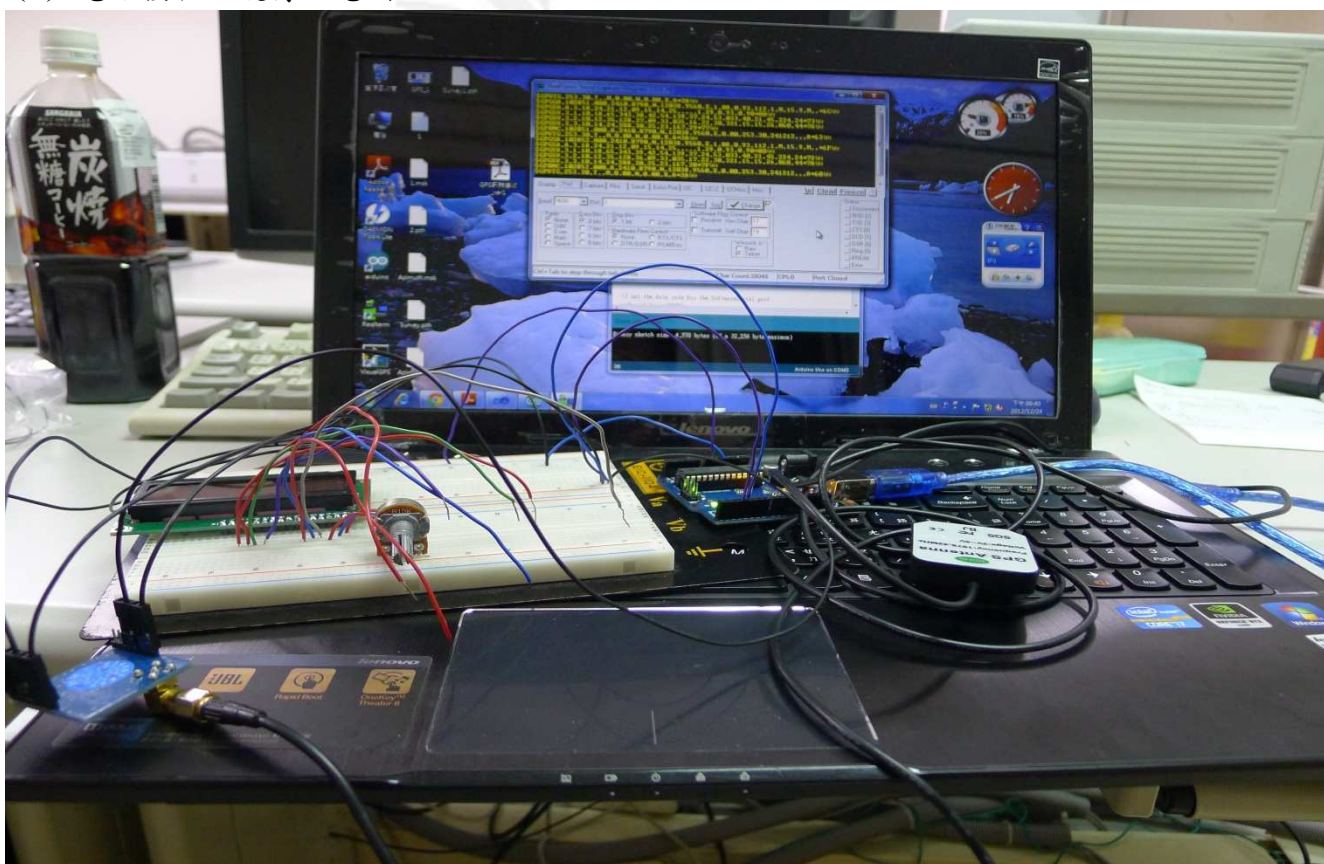
void loop() // run over and over
{
  if (mySerial.available())
    Serial.write(mySerial.read());
  if (Serial.available())
    mySerial.write(Serial.read());
}

Compiling sketch...
35 Arduino Uno on COM3
```

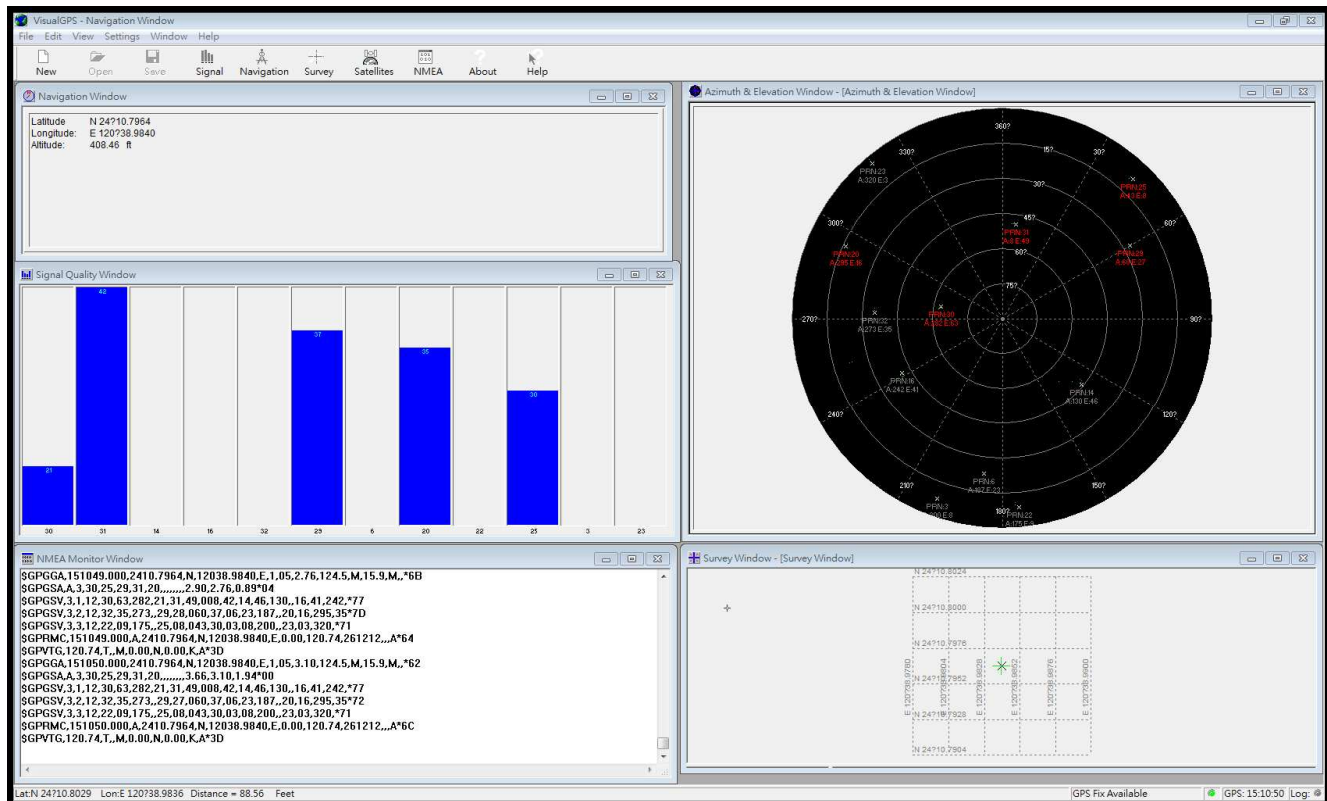

(2) 編譯程式(未字串處理)_2



(3) 電路接收訊號傳至電腦

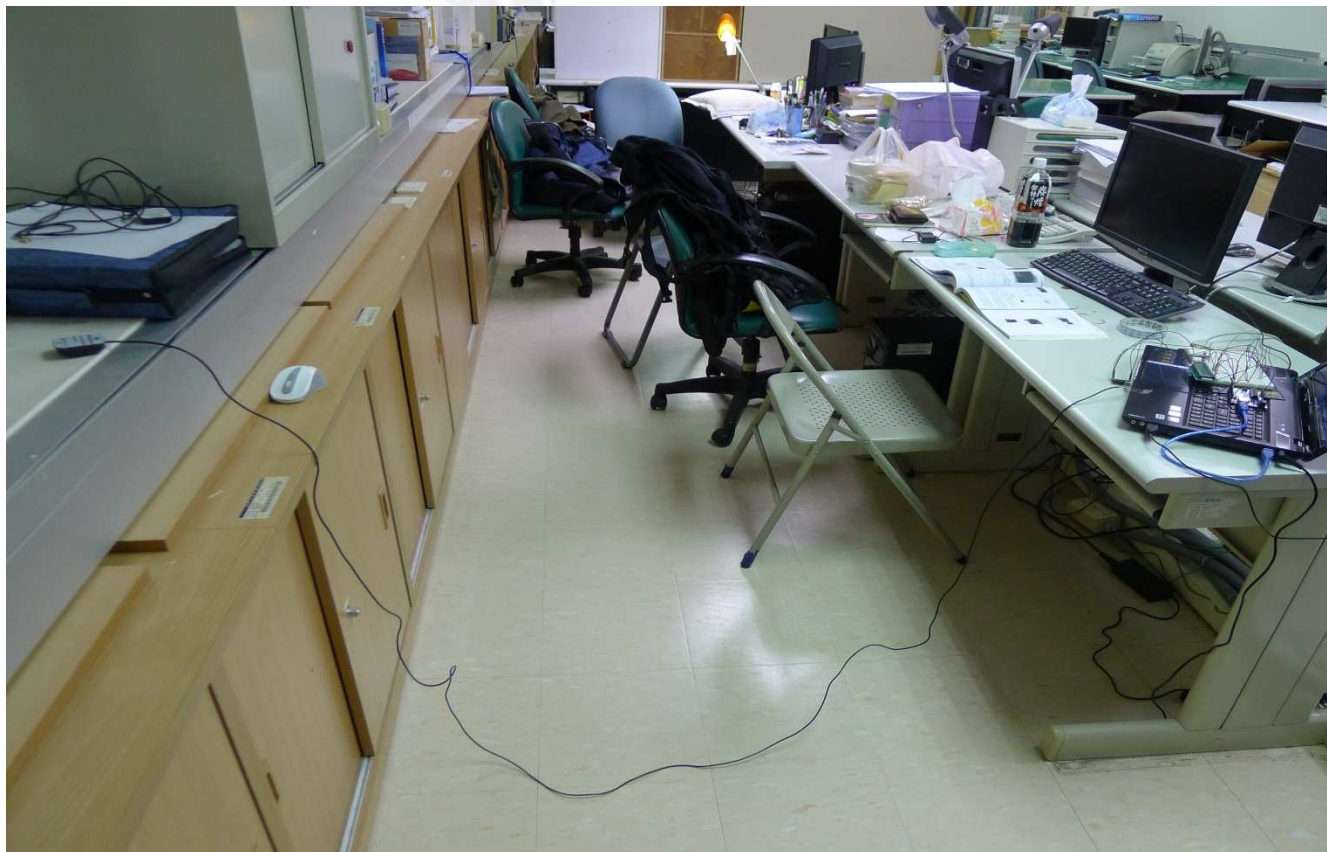


(4)電腦顯示(使用 Visual GPS, 未字串處理)



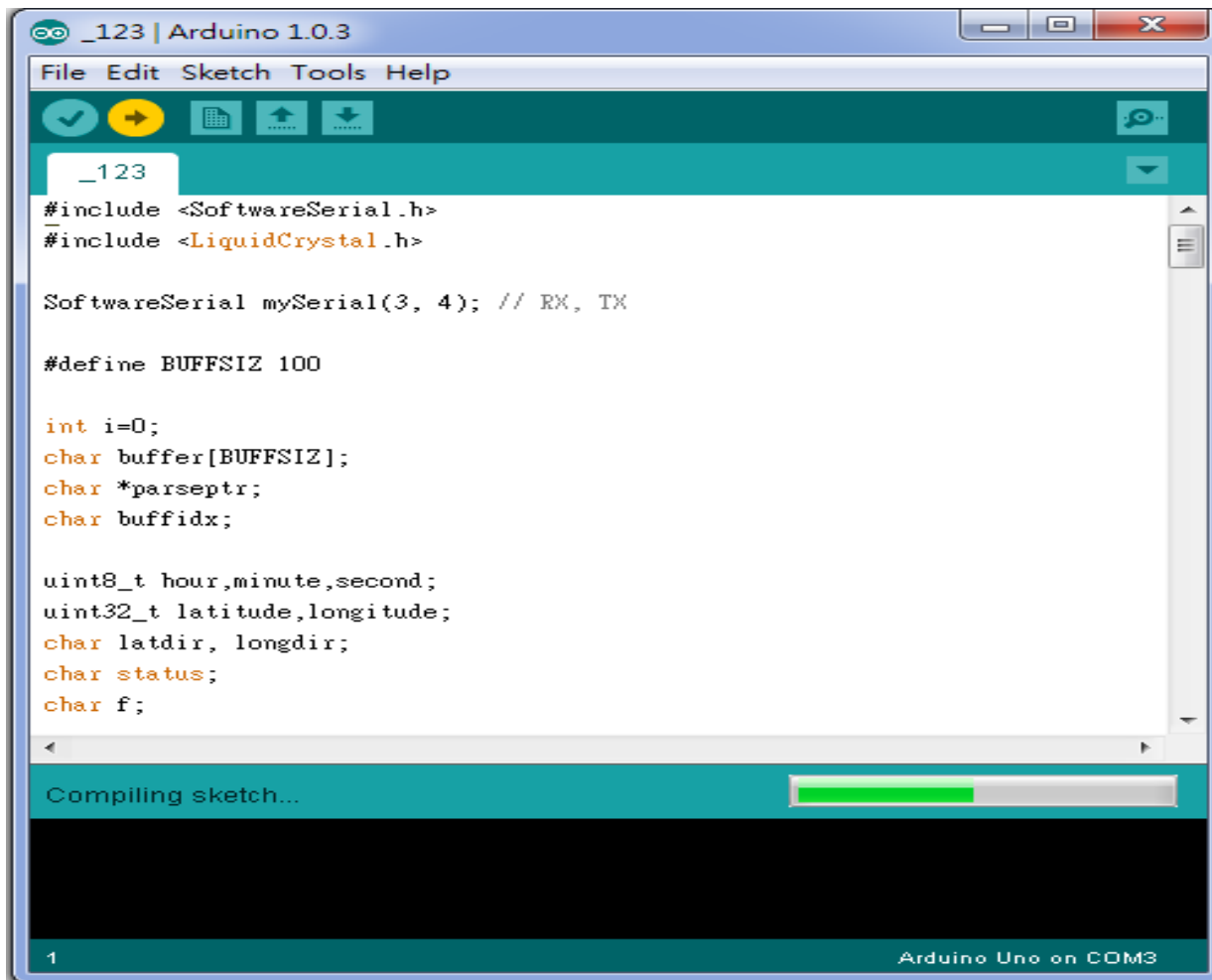
2. 處理字串

(1)電路實接(已接上LCD)

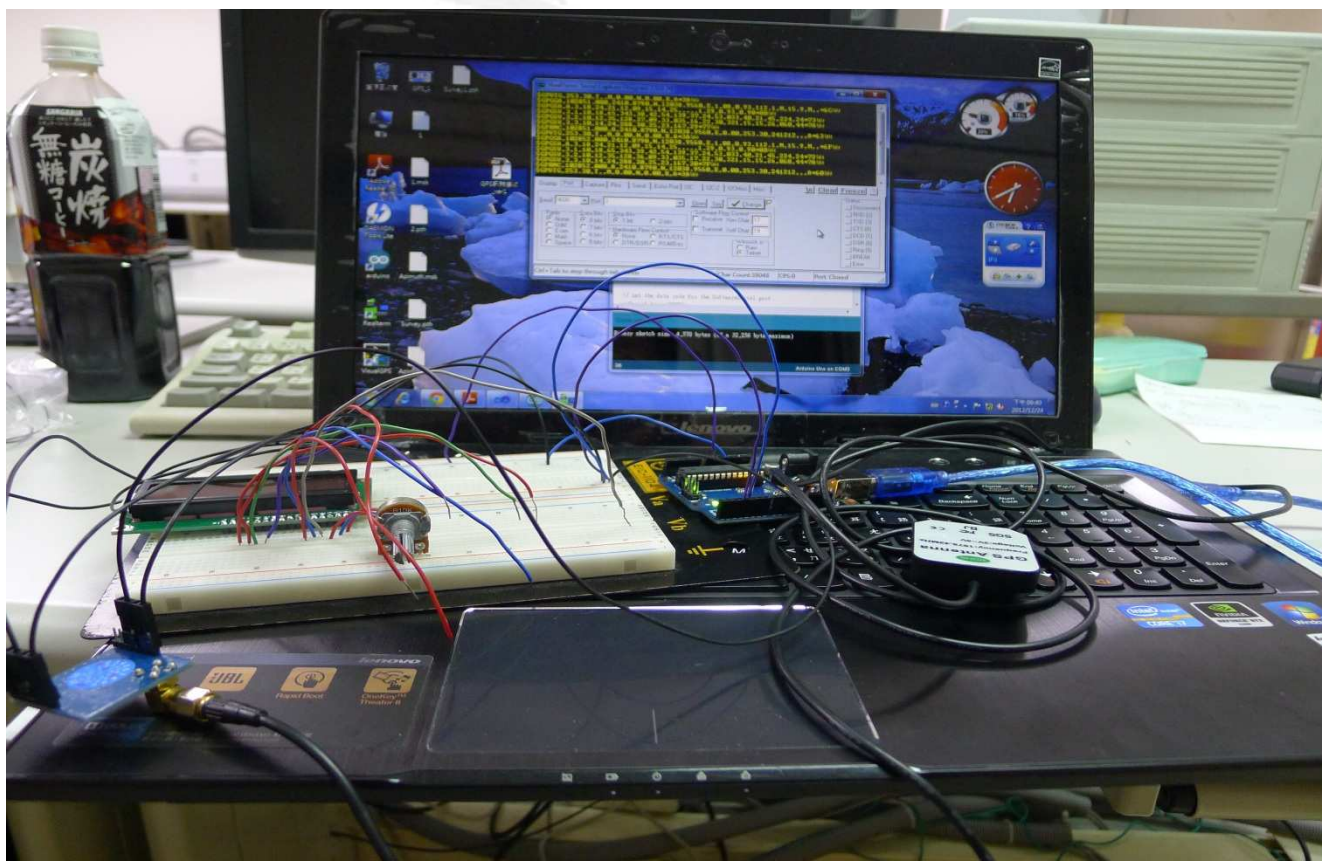


途中左邊是天線, 右邊是接收電路以及顯示螢幕, 這樣就能在室內收到 GPS 訊號

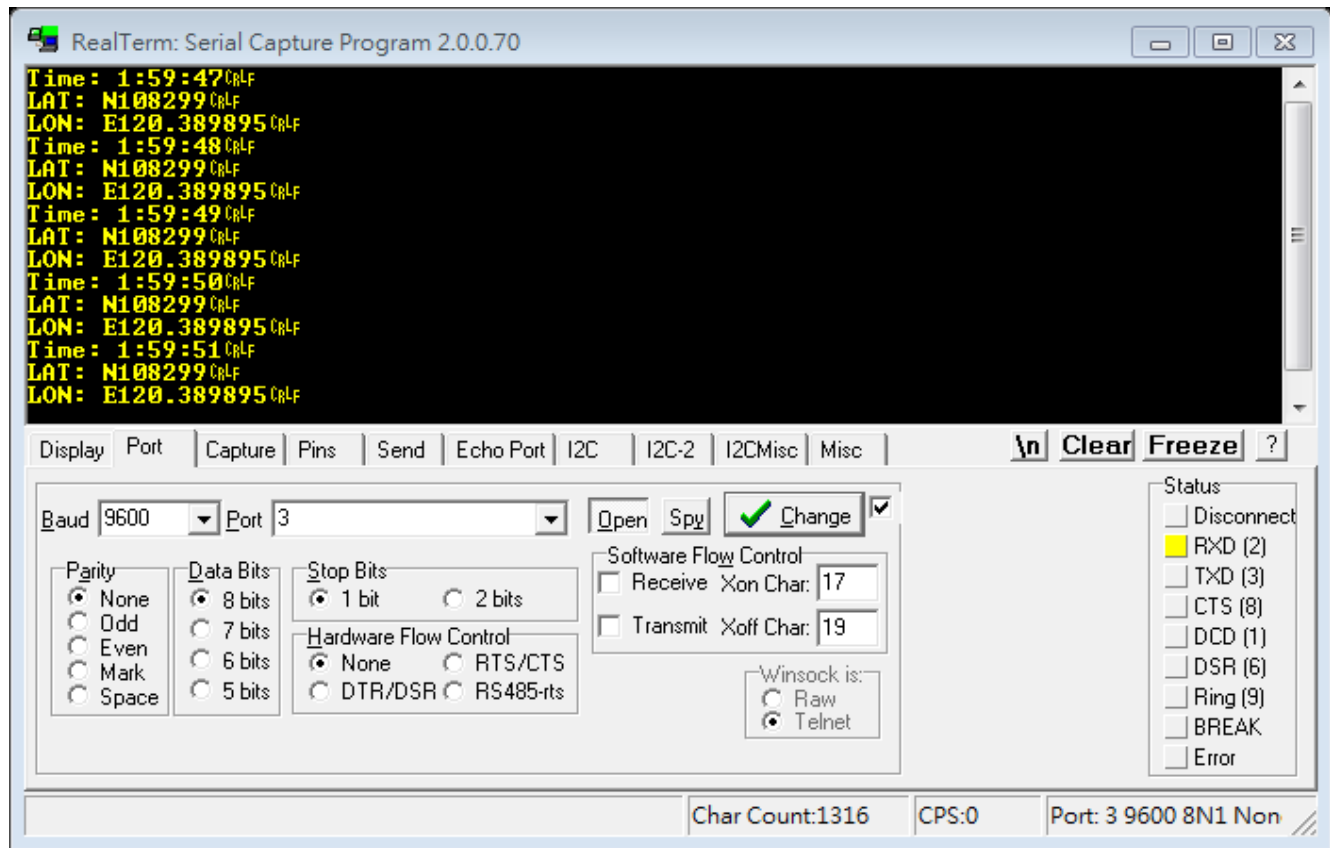
(2) 編譯程式(有處理字串)



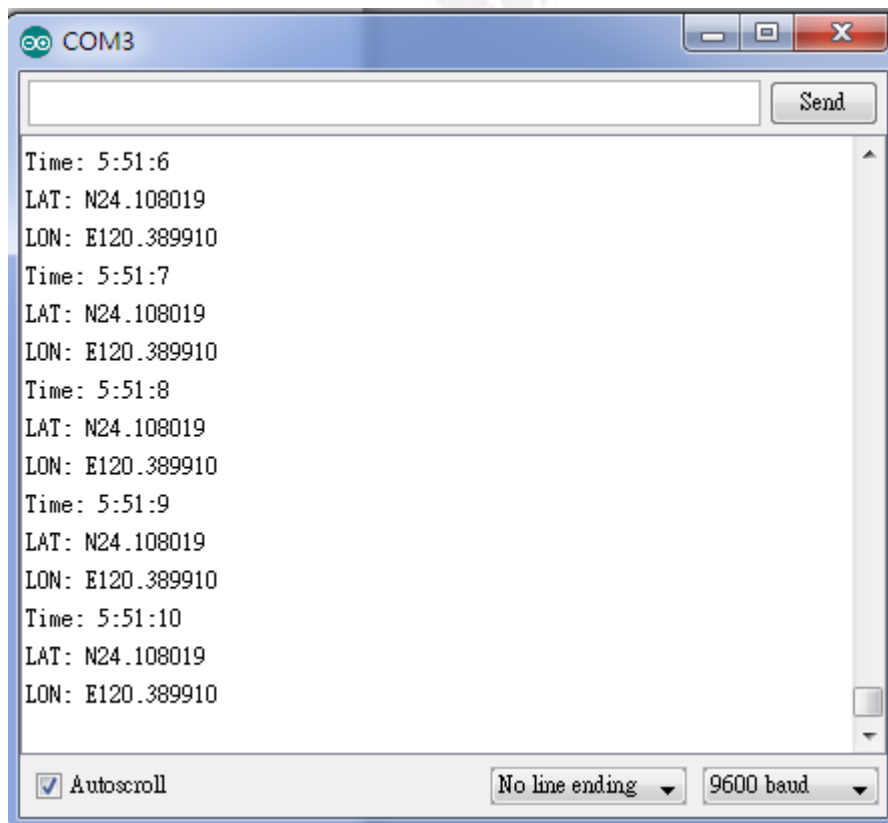
(3) 電路接收訊號傳至電腦



(4) 電腦使用 RealTerm 來顯示



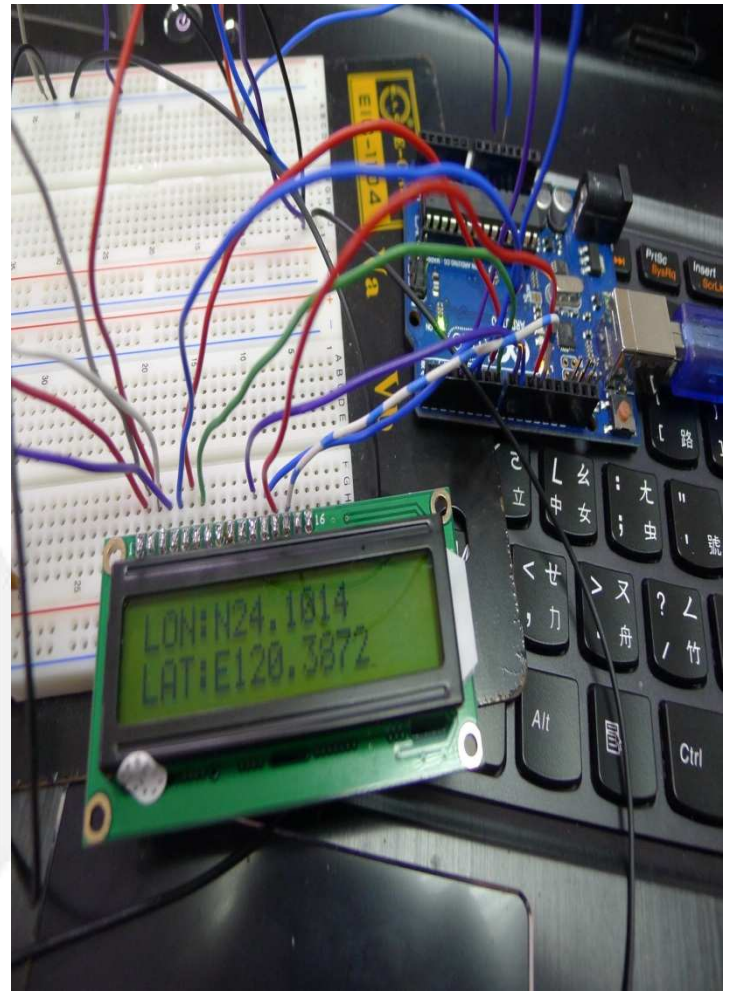
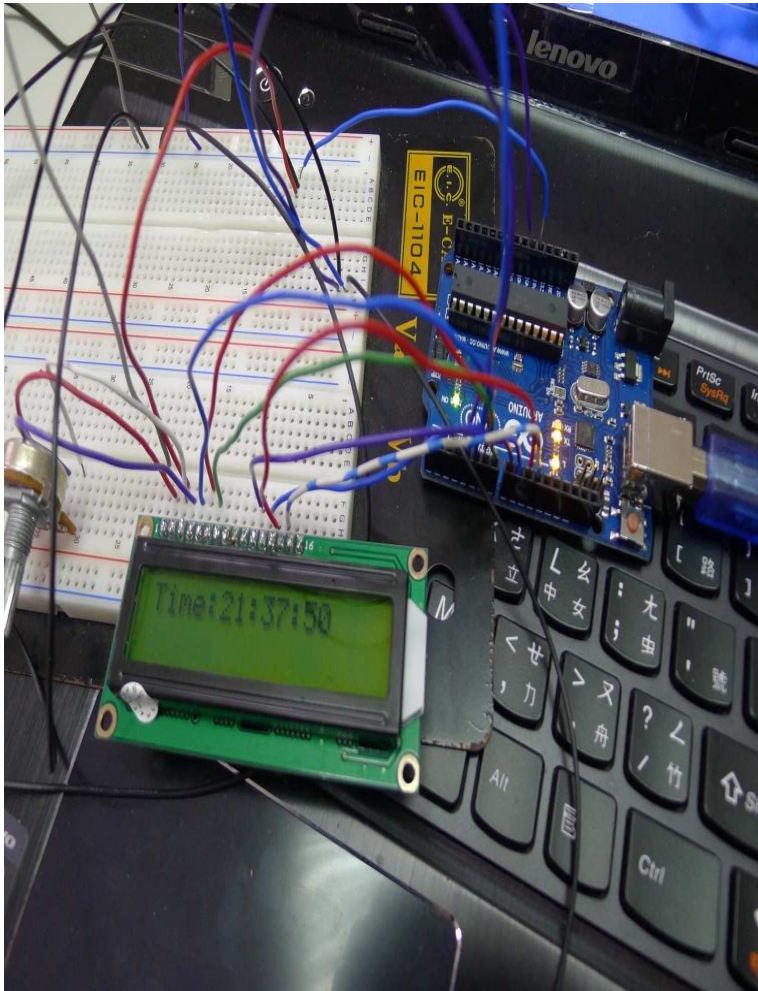
(5) 使用 Arduino 的 Serial Monitor 來顯示



(6) 電腦處理過的訊息顯示在 LCD

[時間]

[經緯度]



(7)預計搭配 GOOGLE MAP (NOT DONE)



參考文獻

1. 最簡單的互動設計 Arduino 一試就上手

作者：孫駿榮、吳明展、盧聰勇

<http://www.books.com.tw/exep/prod/booksfile.php?item=0010491102>

