

中華大學資訊工程學系
100 學年度專題製作期末報告

ZigBee 定位與應用

組員：

B09702014 王志偉(組長)

B09702149 陳暘

B09702071 林柏翰

指導老師：俞征武 教授

中華民國 100 年 6 月 25 日

目 錄

第一章 簡介	
1-1 背景	3
1-2 目的	3
第二章 使用的硬體元件	
2-1 硬體配備.....	4
2-2 軟體環境.....	10
第三章 硬體設計及程式的邏輯架構	
3-1 硬體組裝.....	12
3-2 功能設計敘述.....	12
3-3 程式架構.....	13
第四章 主要成果	
4-1 定位座標	16
4-2 連結 C# 介面打出座標點	17
第五章 遭遇困難	
5-1 ZigBee 之間無法傳送與接收資料	18
5-2 ZigBee 傳送資料時發生嚴重碰撞	18
5-3 三角定位演算法	19
第六章 未來發展與結語	
6-1 未來發展.....	20
6-2 結語與心得.....	20
其他	
0-1 程式碼.....	21
0-2 時間分配表.....	36
0-3 組員分工.....	38
0-4 參考資料.....	39

第一章 簡介

1-1 背景

Zigbee 是一種短距離的通訊技術，理想中的傳輸距離是 100 公尺，省電是它的最大訴求，一個 zigbee 裝置的電池可以使用數月到數年之久。而 zigbee 的應用定位也比較簡單。

鑒於無線網絡部署的靈活性，ZigBee 無線定位非常適合在布線條件不具備的應用場合（如：剛剛興建的建築工地、室外貨場等）。

1-2 目的

本次的專題的主要的目的是以 ZigBee 取得 RSSI 的值並用 RSSI 取得的值轉換成距離值，再用三角定位演算法來算出座標，藉此來達到定位的功能。

第二章 使用的硬體元件

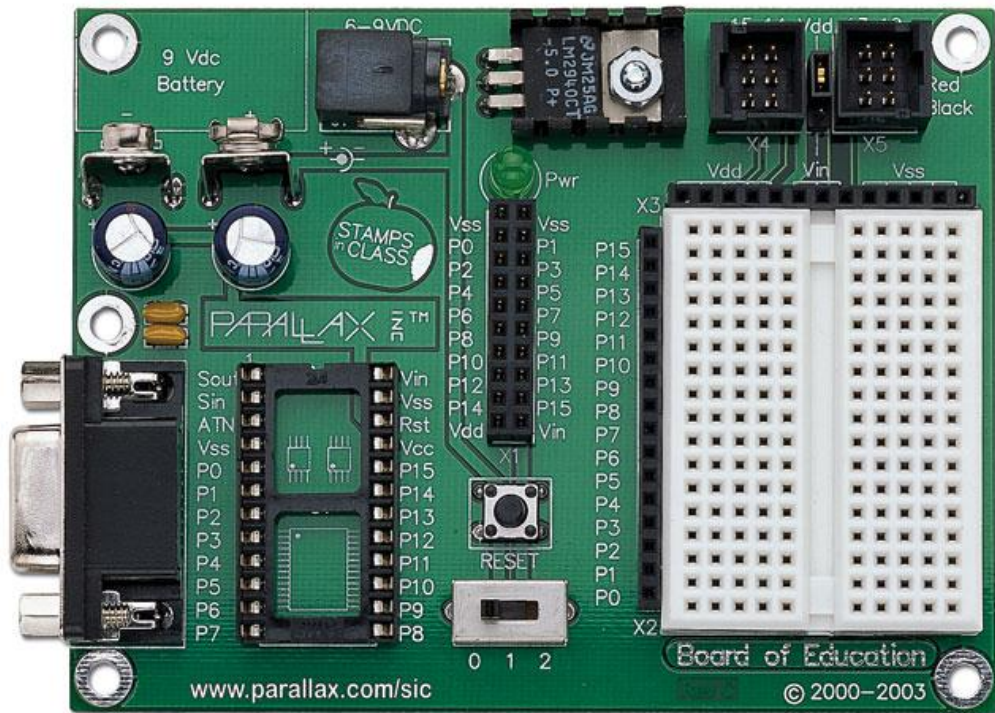
2-1 硬體配備

我們藉由無線感測自走車的開發實驗電路板來當基礎。並配備上 ZigBee 感應器來感測母機與子機接收與傳送。



(1.) 開發實驗電路板

所有的組件都是以這塊電路板為基礎所接上去的。



開發實驗電路板

此實驗電路板有許多種版本，但是每個版本其功能大同小異，大多只差在下列幾項，BASIC Stamp 晶片是否預先焊接上去、電源供應器插頭(一般電池或方型電池)、與電腦連接的傳輸線插頭不同(傳統多針頭式或小型 USB 兩種版本)，但是整體上的功能都是一樣的，當然在 BASIC Stamp 晶片的部分還有分幾種後面會在介紹，在此就不多贅述。

(2.) BASIC Stamp 2 module

目前最廣泛受歡迎使用的晶片，可用在工業控制，教學，機器人控制；舉凡通訊，影像，I/O，運動控制等等，所有程式碼都存在這邊，接收到訊號後也是在這邊判斷。



BASIC Stamp 2 module

其包含了控制、算術邏輯、記憶三大單元，只是整體效能上與CPU仍有一段落差可說是CPU的簡化再簡化，所以在程式撰寫上往往不免受限於硬體的極限影響。

此外先前提及 BASIC Stamp 晶片目前又可分為下列幾種，在此就做一下簡介。依外型顏色可區分成紅色、藍色、黃色、綠色、白色以此五種顏色為首下面又可分出不同的型號，不同的晶片自然有不同的功能，而目前教學上常用的則是綠色 BASIC Stamp 2 module。

(3.) Zigbee 又稱 XBee

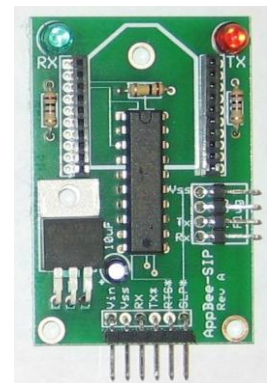
XBee 簡介

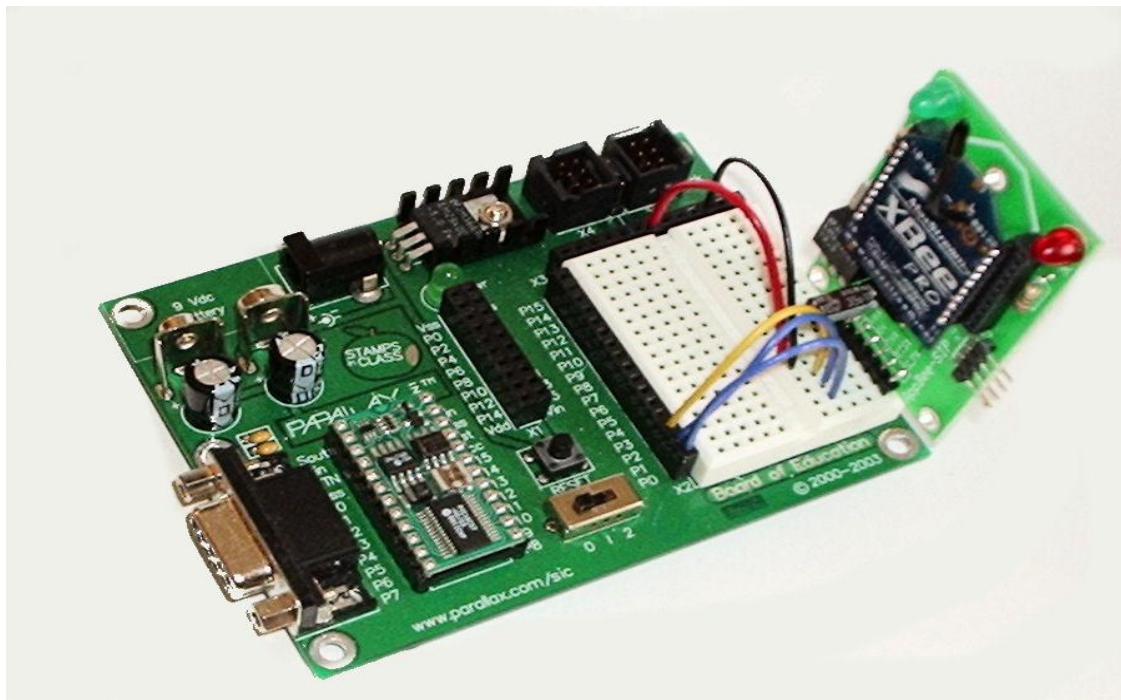
一種無線感測網路器，在教學上很普及 2.4GHz XBee 模組，而且平價、低功率。這個模組是利用簡單的 Serial command (串列傳輸)來執行在 802.15.4 上的 Zigbee 通訊協定。可支援任何微處理器，電腦與系統間的通訊，只要它們具有 serial port。不論是點對點 (point-to-point)或是多點間 multi-point networks，都可以支援。



此通訊裝置在與電路板連接時，還需要一樣轉接板，稱為 ZigBee (XBee)對 BASIC Stamp 系列連接板，此轉接板目前又可分為兩種紅色及綠色，而我們專題用

到的也是目前常用的綠色的版本。





上圖為 Zigbee 插上轉接板後再與教學板的連接範例圖，圖中紅色及黑色線是傳導電流正負極電源線，剩餘的黃色及藍色則是傳送資料的資料線。

在教學板上的白色部分稱為麵包板，而麵包板周圍的黑色插槽各有其編號，每個編號的一個孔稱為一個 pin 角(腳)。上排黑色插槽編號 Vdd、Vin 及 Vss 為供應電流正負極的部分，左排黑色插槽編號 P0~P15 則是資料線，連結著 BASIC Stamp 晶片，只需要在程式中建立位子(P0~P15 任何選一個) BASIC Stamp 晶片就會使該 pin 角通電藉此控制麵包板上的元件。

在 Zigbee 及轉接板上每個針狀插頭也各有編號，Vin 及 Vss 是

負責電流正負極通電，Rx 和 Tx 是訊號接收與送出，最後還有 RTS 及 SLP 也有其各自的功能，但因為我們專題未使用到在此就不多介紹。

(4.) 電源供應

提供所有實驗電路板上所有設施的全部能量，電壓不足可能會判斷錯誤，或停止運行。



9V-方形電池



四顆串接電池

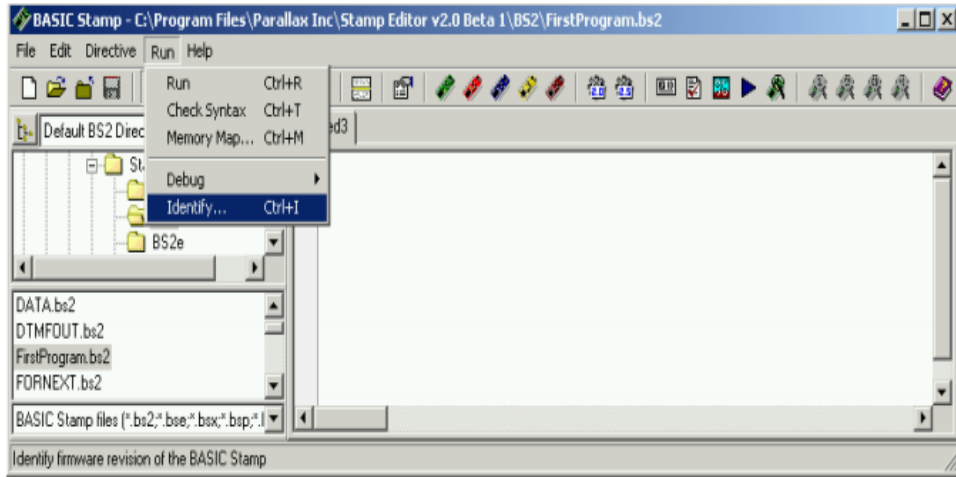
在我們的專題中發現，電源供應器電壓的強弱會明顯影響 Zigbee 的訊號傳送，因此在電池的選用上建議選用好一點的電池最好是 1.5V 的。因此若在實驗中，發現硬體沒問題但是訊號就是沒接收或傳送，不妨檢察一下電力是不是不夠。

2-2 軟體環境

BASIC Stamp



此是前面所述的硬體 BASIC Stamp 的開發環境。



開始撰寫的畫面，有點類似 C 語言但是相較之下此語言較低階一點。

Microsoft Visual C# 2010

Microsoft Visual C# 2010 用來編寫 C 程式語言的工具，本次使用它來製作【連結介面】，與【電腦與 ZigBee 溝通】的程式。

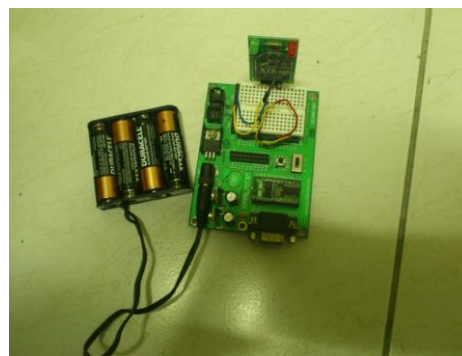


第三章 硬體設計及程式的邏輯架構

3-1 硬體組裝



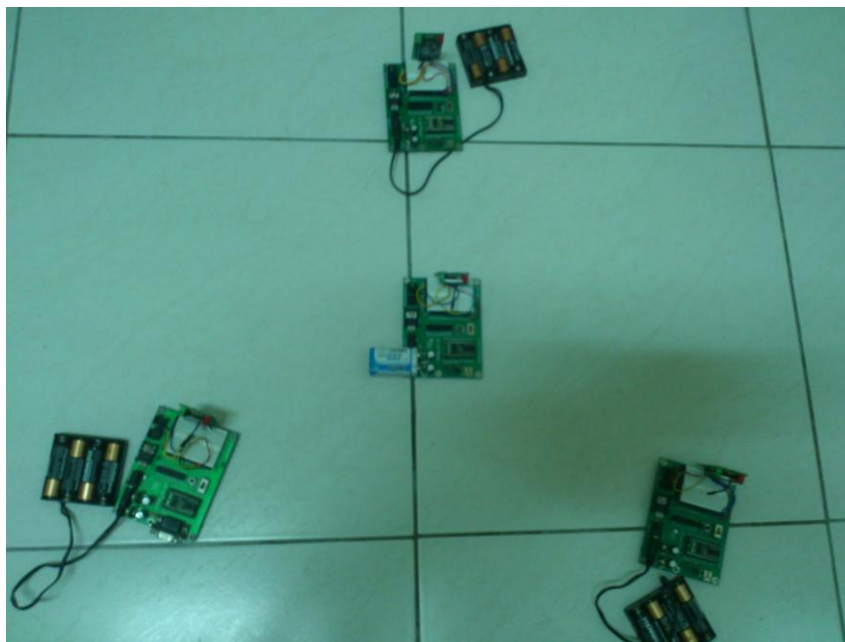
母機端(由欲定位者攜)



子機端(在場地的周圍設置帶)

3-2 功能設計敘述

將子機端設置在場地的周圍利用三角定位的方法來定位

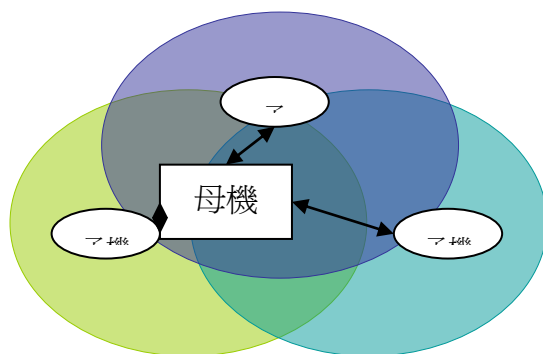


我們的設備主要分為兩端，如上圖所示。分別為母機端和子機端，子機端設置在場地的周圍，母機端則由各個欲定位者個別攜帶。其功能

目的如下：

每個母機端在子機端上都有一個專屬的訊號燈，子機端會不時的偵測各個母機端的訊號，只要有偵測到，各子機專屬的信號燈會抱持閃爍，同時子機端也會不停發送訊號給母機端讓母機端偵測，因此，只要母機還在偵測距離內就不斷的定位出母機的位置。

3-3 程式架構



訊號偵測傳輸概念圖

3-3-1:程式架構概念

上圖是我們建立的定位訊號傳輸概念，我們所使用的定位方法是三角定位，上圖中間的長方形所包含的範圍代表場地的範圍，在場地的四周設置定位器(子機)可以看出不論母機移動到場地的何處都會在三台子機的偵測範圍內，所以不管移動至場地的何處都可以被定位系統所定位到。

3-3-2:程式架構流程

Step1:三台座標子機分別傳送 RSSI 值給母機

Step2:母機接收從三台子機傳送的訊號後

Step3:經過三角定位演算法

Step4:算出現在座標點

Step5:把資料傳到並連結到介面上打一個點

Step6:再回到 Step1

3-3-3:三角定位演算法

假設三個基地台 BS1、BS2 及 BS3 的參考座標分別為 (x_1, y_1) 、 (x_2, y_2) 與 (x_3, y_3) ，MT 的座標為 (x, y) 距離三個基地台分別是 r_1 、 r_2 與 r_3 ，則各基地台的圓形方程式可表示為

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 \quad i = 1, 2, 3 \quad (1)$$

三個圓形方程式兩兩相減，可推得

$$2x(x_2 - x_1) + 2y(y_2 - y_1) = r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \quad (2)$$

$$2x(x_3 - x_1) + 2y(y_3 - y_1) = r_1^2 - r_3^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2 \quad (3)$$

$$2x(x_3 - x_2) + 2y(y_3 - y_2) = r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \quad (4)$$

式(2)、(3)及(4)以矩陣方式來表示，可寫成

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\ r_1^2 - r_3^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2 \\ r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \end{bmatrix} \quad (5)$$

$$\text{若令 } \mathbf{A} = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\ r_1^2 - r_3^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2 \\ r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \end{bmatrix}, \quad \text{則式(5)可簡化為}$$

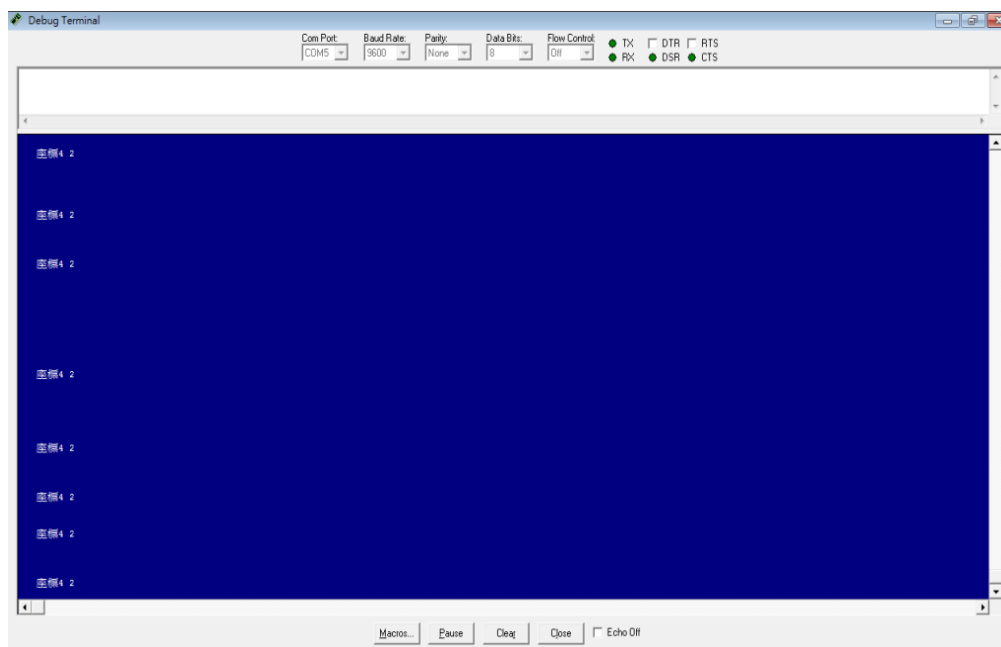
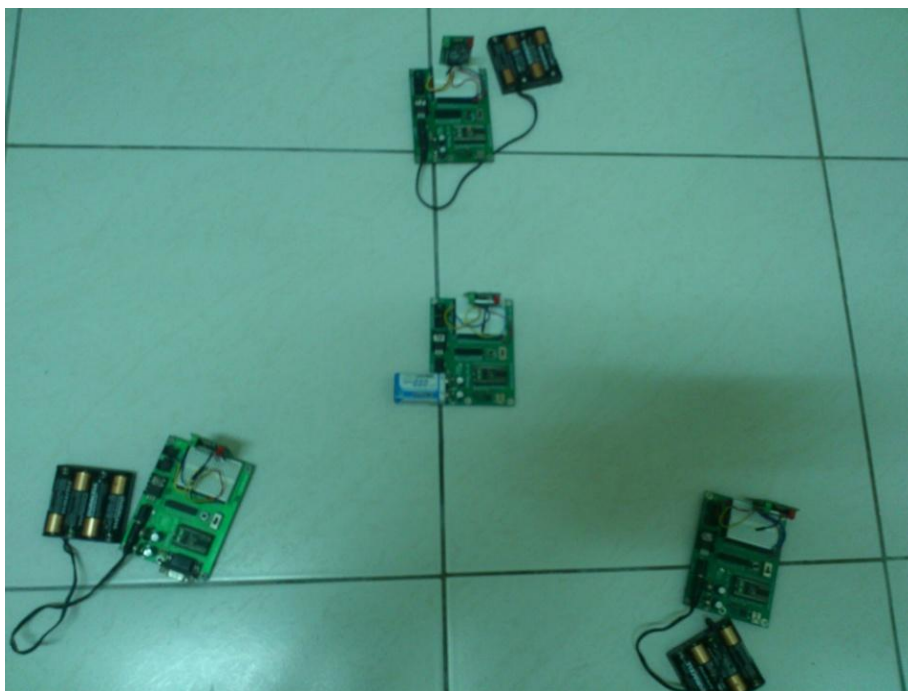
$\mathbf{A} \cdot \mathbf{P} = \mathbf{B}$ 。利用 Pseudo-inverse 技巧，MT 位

置方程式可表示成 $\mathbf{P} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{B}$ 。

第四章 主要成果

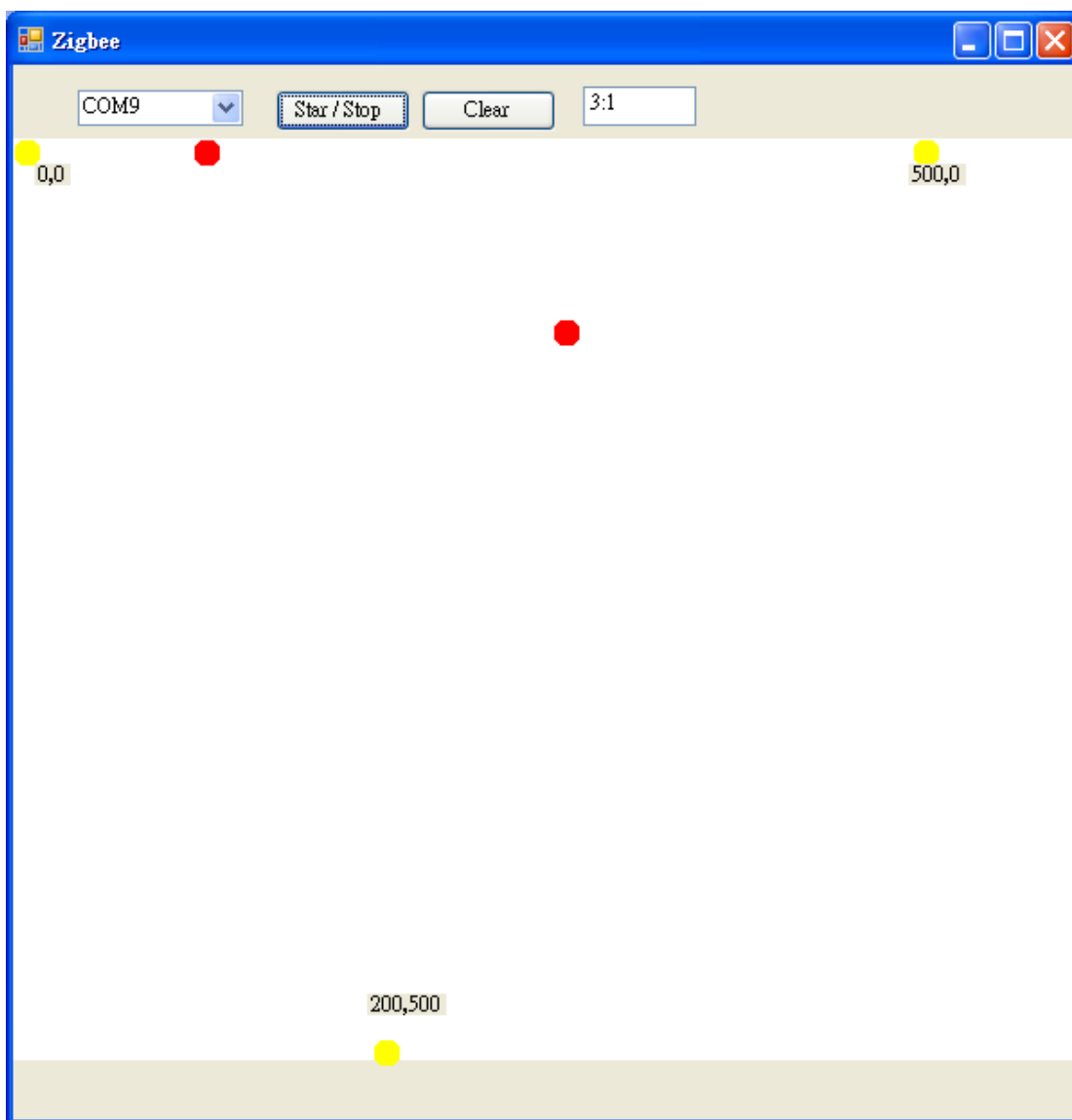
4-1 定位座標

透過 ZigBee 傳輸的 RSSI 值來定位座標點



4-2 連結 C# 介面打出座標點

使用 C# 介面連結 USB 到麵包板上將板子上送出的座標位置印在畫面上



第五章 遭遇困難

5-1 ZigBee 之間無法傳送與接收資料

後來發現原來 ZigBee 有頻段設定的問題如果頻段不同就互相收不到訊號，舉例來說：我有 3 個 ZigBee 其中第一個 ZigBee 的頻段是 0、第二個 ZigBee 的頻段是 1、第三個 ZigBee 的頻段也是 1，那麼第一個 ZigBee 就和第二三個 ZigBee 無發互相傳送資料但是第二個和第三個 ZigBee 就可以互相傳送資料。



解決辦法：

後來我們上網找得了解決辦法就是用右上角那塊紅色版子連接到電腦，然後用 X-CUT 這個軟體進到 ZigBee 內部去做設定。

5-2 ZigBee 傳送資料時發生嚴重碰撞

三台子機傳送資料的時間太相近太頻繁了，導致傳送資料時發生嚴重碰撞導致接收不到資料

解決辦法：

所以我們 Random 他的送出的時間讓他送封包的時間錯開

5-3 三角定位演算法

這個演算法他的難處在於他有一個 3×2 的反矩陣但是上網找了
又找發現反矩陣只有方陣才有這下完全卡住了

解決辦法：

最後我們用了 MATLAB 這個程是幫我們算出了這個矩陣並順利的
的完成了三角定位程式

第六章 未來發展與結語

6-1 未來發展

- ◎ 礦井下人員定位 — 利用區域定位技術，可以隨時了解礦工的工作位置和相關信息。
- ◎ 醫院中被看護對象的定位 — 利用區域定位技術，可以隨時了解被看護對象活動區域或貴重醫療設備的放置場所。被看護對象也可以通過隨身裝置，向醫護人員發出求救信號。
- ◎ 消防人員定位 — 利用區域定位技術，可以動態掌握進入火災現場消防隊員的活動範圍，為現場指揮提供有效的資料。
- ◎ 建築工地管理 — 利用區域定位技術，可以了解施工人員的考勤及活動狀況。

當然上面所列的想法不過只是冰山一角，定位系統能應用的範圍相當廣泛，在未來一定能有更多的應用能使人們的生活更便利更有品質。

6-2 結語與心得

在這一年的時間內，雖然過程中碰到一次又一次的問題使得進度好幾次停滯不前，經過一番努力與幾位好心學長的幫忙之下，我們終於完成了這一次的專題，經過這次專題我們學習到了，挑戰了一些沒有學過的東西，自己去研究並解決問題，並了解到充分團隊合作的重要性。

其他

0.1 程式碼

母機_程式碼:

```
                                ' {$STAMP BS2}
' {$PBASIC 2.5}
' *****
' * Get_RSSI.bs2                *
' * Illustrates accepting a Byte *
' * received as a character     *
' * and reads/displays RSSI dBm *
' *****
myAddr  CON $A2    ' Node Address(*自己的位置*)
DestAddr CON $A2  ' Destination address

Baud    CON 84    ' Baud rate, 9600, 8-N-1, non-inverted, on BS2.

RX      PIN 0    ' Receive Pin
TX      PIN 2    ' Transmit Pin
RTS     PIN 6    ' Flow control Pin

RFin    VAR Byte
dBm     VAR Byte
x VAR Byte
y VAR Byte
r1 VAR Byte
r2 VAR Byte
r3 VAR Byte
c1 VAR Byte
c2 VAR Byte
c3 VAR Byte
HIGH TX

' DEBUG CLS, "Configuring XBee..."
PAUSE 2000                ' Guard time for command sequence
SEROUT TX, Baud, ["+++"] ' Enter command mode
```

```

PAUSE 2000                                ' Guard time for command sequence
SEROUT TX, Baud, ["ATNI BS2 Test Node", CR, ' Set description
      "ATMY ", HEX myAddr, CR,           ' Set node address
      "ATDL ", HEX DestAddr, CR,        ' Set destination node address
      "ATD6 1", CR,                      ' Use RTS for flow control
      "ATGT 3", CR,                      ' Reduce Gaurd Time
      "ATCN" ]                          ' Exit command mode

```

```

PAUSE 1000

```

```

GOSUB WaitOK                              ' Accept OK data
' DEBUG "Configuration Complete!", CR

```

```

DO                                          ' Accept and view a character

```

```

  SERIN RX\RTS, Baud, 1000, Timeout, [RFin]
  ' DEBUG RFin
  GOSUB GetRSSI
Timeout:
LOOP

```

```

GetRSSI:                                  ' Read RSSI data
  PAUSE 20                                ' Guard time
  SEROUT TX, Baud, ["+++"]               ' Command mode
  PAUSE 2
  SEROUT TX, Baud, ["ATDB", CR]          ' Request level
  SERIN RX\RTS, Baud, 1000, Timeout2, [HEX dBm] ' Accept data
  ' DEBUG "    dbm: -", DEC dBm          ' Display data

```

```

IF (RFin = 49) THEN IF (dBm = 24) OR (dBm = 23 ) OR (dBm = 25) THEN

```

```

x = 0

```

```

y = 0

```

```

' DEBUG "    座標"

```

```

DEBUG DEC x , " ", DEC y

```

```

ENDIF

```

```

IF (RFin = 50) THEN IF (dBm = 24) OR (dBm = 23 ) OR (dBm = 25) THEN

```

```

x = 50

```

```

y = 0

```

```

' DEBUG "    座標"

```

```

DEBUG DEC x , " ", DEC y

```

```
ENDIF
IF (RFin = 51)THEN IF(dBm = 24) OR (dBm = 23 ) OR (dBm = 25) THEN
x = 20
y = 50
' DEBUG "   座標"
DEBUG DEC x , " ", DEC y
ENDIF
```

```
IF(RFin = 49) THEN IF(dBm = 32)THEN
r1 = 18
ELSEIF(dBm < 32)THEN
r1 = 18
ELSEIF(dBm = 33)THEN
r1 = 19
ELSEIF(dBm = 34)THEN
r1 = 20
ELSEIF(dBm = 35)THEN
r1 = 21
ELSEIF(dBm = 36)THEN
r1 = 22
ENDIF
```

```
IF(RFin = 49)THEN IF(dBm = 37)THEN
r1 = 38
ELSEIF(dBm = 38)THEN
r1 = 39
ELSEIF(dBm = 39)THEN
r1 = 40
ELSEIF(dBm = 40)THEN
r1 = 41
ELSEIF(dBm = 41)THEN
r1 = 42
ENDIF
```

```
IF(RFin = 49)THEN IF(dBm = 41)THEN
r1 = 58
ELSEIF(dBm = 42)THEN
r1 = 59
ELSEIF(dBm = 43)THEN
r1 = 60
```

```

ELSEIF(dBm = 44)THEN
r1 = 61
ELSEIF(dBm = 45)THEN
r1 = 62
ENDIF
IF(RFin = 49)THEN IF(dBm = 46)THEN
r1 = 78
ELSEIF(dBm = 47)THEN
r1 = 79
ELSEIF(dBm = 48)THEN
r1 = 80
ELSEIF(dBm = 49)THEN
r1 = 81
ELSEIF(dBm = 50)THEN
r1 = 82
ENDIF
IF(RFin = 49)THEN IF(dBm = 51)THEN
r1 = 98
ELSEIF(dBm = 52)THEN
r1 = 99
ELSEIF(dBm = 53)THEN
r1 = 100
ELSEIF(dBm = 54)THEN
r1 = 101
ELSEIF(dBm = 55)THEN
r1 = 102
ELSEIF(dBm > 56)THEN
r1 = 103
ENDIF

IF(RFin = 50) THEN IF(dBm = 32)THEN
r2 = 18
ELSEIF(dBm < 32)THEN
r2 = 18
ELSEIF(dBm = 33)THEN
r2 = 19
ELSEIF(dBm = 34)THEN
r2 = 20

```



```

ELSEIF(dBm = 35)THEN
r2 = 21
ELSEIF(dBm = 36)THEN
r2 = 22
ENDIF
IF(RFin = 50)THEN IF(dBm = 37)THEN
r2 = 38
ELSEIF(dBm = 38)THEN
r2 = 39
ELSEIF(dBm = 39)THEN
r2 = 40
ELSEIF(dBm = 40)THEN
r2 = 41
ELSEIF(dBm = 41)THEN
r2 = 42
ENDIF
IF(RFin = 50)THEN IF(dBm = 41)THEN
r2 = 58
ELSEIF(dBm = 42)THEN
r2 = 59
ELSEIF(dBm = 43)THEN
r2 = 60
ELSEIF(dBm = 44)THEN
r2 = 61
ELSEIF(dBm = 45)THEN
r2 = 62
ENDIF
IF(RFin = 50)THEN IF(dBm = 46)THEN
r2 = 78
ELSEIF(dBm = 47)THEN
r2 = 79
ELSEIF(dBm = 48)THEN
r2 = 80
ELSEIF(dBm = 49)THEN
r2 = 81
ELSEIF(dBm = 50)THEN
r2 = 82
ENDIF

```

```
IF(RFin = 50)THEN IF(dBm = 51)THEN
r2 = 98
ELSEIF(dBm = 52)THEN
r2 = 99
ELSEIF(dBm = 53)THEN
r2 = 100
ELSEIF(dBm = 54)THEN
r2 = 101
ELSEIF(dBm = 55)THEN
r2 = 102
ELSEIF(dBm > 56)THEN
r2 = 103
ENDIF
```

```
IF(RFin = 51) THEN IF(dBm = 32)THEN
r3 = 18
ELSEIF(dBm < 32)THEN
r3 = 18
ELSEIF(dBm = 33)THEN
r3 = 19
ELSEIF(dBm = 34)THEN
r3 = 20
ELSEIF(dBm = 35)THEN
r3 = 21
ELSEIF(dBm = 36)THEN
r3 = 22
ENDIF
```

```
IF(RFin = 51)THEN IF(dBm = 37)THEN
r3 = 38
ELSEIF(dBm = 38)THEN
r3 = 39
ELSEIF(dBm = 39)THEN
r3 = 40
ELSEIF(dBm = 40)THEN
r3 = 41
ELSEIF(dBm = 41)THEN
r3 = 42
ENDIF
```

```
IF(RFin = 51)THEN IF(dBm = 41)THEN
r3 = 58
ELSEIF(dBm = 42)THEN
r3 = 59
ELSEIF(dBm = 43)THEN
r3 = 60
ELSEIF(dBm = 44)THEN
r3 = 61
ELSEIF(dBm = 45)THEN
r3 = 62
ENDIF
IF(RFin = 51)THEN IF(dBm = 46)THEN
r3 = 78
ELSEIF(dBm = 47)THEN
r3 = 79
ELSEIF(dBm = 48)THEN
r3 = 80
ELSEIF(dBm = 49)THEN
r3 = 81
ELSEIF(dBm = 50)THEN
r3 = 82
ENDIF
IF(RFin = 51)THEN IF(dBm = 51)THEN
r3 = 98
ELSEIF(dBm = 52)THEN
r3 = 99
ELSEIF(dBm = 53)THEN
r3 = 100
ELSEIF(dBm = 54)THEN
r3 = 101
ELSEIF(dBm = 55)THEN
r3 = 102
ELSEIF(dBm > 56)THEN
r3 = 103
ENDIF

GOTO out
in:
```

```

Timeout2:
  SEROUT TX, Baud, ["ATCN", CR]      ' Exit command mode
  GOSUB WaitOK                       ' Accept OK from XBee
  ' DEBUG CR
RETURN

```

```

WaitOK:
  SERIN RXRTS, Baud, 100, Timeout3, [RFin] ' Accept OK data
  GOTO WaitOK
  Timeout3:
RETURN

```

```

out:
  c1 = r1*r1-r2*r2+25
  c2 = r1*r1-r3*r3+29
  c3 = r2*r2-r3*r3+4
  IF(r1 > 0) AND (r2 > 0) AND (r3 > 0)THEN
  x = (c1*667 + c2*333 - c3*333)/10000
  y = (c1*67 + c2*533 + c3*467)/10000
  x = x * 10
  y = y * 10
  ' DEBUG "   座標"
  DEBUG DEC x , " ", DEC y
  dBm = 0
  r1 = 0
  r2 = 0
  r3 = 0
  PAUSE 500
  ENDIF
GOTO in

```

子機 1_程式碼:

```
' {$STAMP BS2}
' {$PBASIC 2.5}
myAddr          CON $A8          ' Node Address (*自己的位置*)
DestAddr        CON $A2          ' Destination address(目標位置)(*要設定*)
result VAR Word
rand VAR Word

Baud            CON 84           ' Baud rate, 9600, 8-N-1, non-inverted, on BS2.

RX              PIN 0           ' Receive Pin
TX              PIN 2           ' Transmit Pin
RTS             PIN 6           ' Flow control Pin

X VAR Byte
HIGH TX

DEBUG CLS, "Configuring XBee..."
PAUSE 2000          ' Guard time for command sequence
SEROUT TX, Baud, ["+++ "] ' Enter command mode
PAUSE 2000          ' Guard time for command sequence
SEROUT TX, Baud, ["ATNI BS2 Test Node", CR, ' Set description
                "ATMY ", HEX myAddr, CR, ' Set node address
                "ATDL ", HEX DestAddr, CR, ' Set destination node address
                "ATD6 1", CR, ' Use RTS for flow control
                "ATCN", CR] ' Exit command mode

PAUSE 1000
DEBUG "Ready!", CR
    X = 1
    result = 11000
DO
    RANDOM result
    PAUSE 800
    rand = result/100
    SEROUT TX, 84, [DEC X, CR, CR]
    PAUSE rand
LOOP
```

子機 2_程式碼:

```
' {$STAMP BS2}
' {$PBASIC 2.5}
myAddr          CON $A9          ' Node Address (*自己的位置*)
DestAddr        CON $A2          ' Destination address(目標位置)(*要設定*)
result VAR Word
rand VAR Word

Baud            CON 84           ' Baud rate, 9600, 8-N-1, non-inverted, on BS2.

RX              PIN 0            ' Receive Pin
TX              PIN 2            ' Transmit Pin
RTS             PIN 6            ' Flow control Pin

X VAR Byte
HIGH TX

DEBUG CLS, "Configuring XBee..."
PAUSE 2000          ' Guard time for command sequence
SEROUT TX, Baud, ["+++ "] ' Enter command mode
PAUSE 2000          ' Guard time for command sequence
SEROUT TX, Baud, ["ATNI BS2 Test Node", CR, ' Set description
                "ATMY ", HEX myAddr, CR, ' Set node address
                "ATDL ", HEX DestAddr, CR, ' Set destination node address
                "ATD6 1", CR, ' Use RTS for flow control
                "ATCN", CR] ' Exit command mode

PAUSE 1000
DEBUG "Ready!", CR
    X = 2
    result = 33000
DO
    RANDOM result
    PAUSE 900
    rand = result/100
    SEROUT TX, 84, [DEC X, CR, CR]
    PAUSE rand
LOOP
```

子機 3_程式碼:

```
' {$STAMP BS2}
' {$PBASIC 2.5}
myAddr          CON $A7          ' Node Address (*自己的位置*)
DestAddr        CON $A2          ' Destination address(目標位置)(*要設定*)
result VAR Word
rand VAR Word

Baud            CON 84           ' Baud rate, 9600, 8-N-1, non-inverted, on BS2.

RX              PIN 0            ' Receive Pin
TX              PIN 2            ' Transmit Pin
RTS             PIN 6            ' Flow control Pin

X VAR Byte
HIGH TX

DEBUG CLS, "Configuring XBee..."
PAUSE 2000          ' Guard time for command sequence
SEROUT TX, Baud, ["+++ "] ' Enter command mode
PAUSE 2000          ' Guard time for command sequence
SEROUT TX, Baud, ["ATNI BS2 Test Node", CR, ' Set description
                "ATMY ", HEX myAddr, CR, ' Set node address
                "ATDL ", HEX DestAddr, CR, ' Set destination node address
                "ATD6 1", CR, ' Use RTS for flow control
                "ATCN", CR] ' Exit command mode

PAUSE 1000
DEBUG "Ready!", CR
    X = 3
    result = 64000
DO
    RANDOM result
    PAUSE 1000
    rand = result/100
    SEROUT TX, 84, [DEC X, CR, CR]
    PAUSE rand
LOOP
```

C#介面程式碼

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;

namespace poj
{
    public partial class Form1 : Form
    {

        public Form1()
        {

            InitializeComponent();
            //serialPort1.Open();
        }

        private delegate void setTextCallback(string text);

        private void pictureBox1_Click(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (serialPort1.IsOpen)
            {
                serialPort1.Close();
            }
            else
```



```

    { serialPort1.PortName = comboBox1.Text;
      serialPort1.Open();

    }

    /*int x = 0 , y = 0 ;
    Graphics g;
    char[] str = new char[100];
    //while(true)
    //{
    //int len = serialPort1.BytesToRead;
    int len = serialPort1.ReadBufferSize;
    serialPort1.Read(str, 0, len);
    //g = pictureBox1.CreateGraphics();
    //g.FillEllipse(Brushes.Red, x, y, 10, 10);
    */
    }

    private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
    {

        try
        {
            //StringBuilder str = new StringBuilder();
            System.Threading.Thread.Sleep(50);
            int len = serialPort1.BytesToRead;
            //Console.WriteLine(len.ToString() + ", ");
            char[] tmp = new char[len];
            serialPort1.Read(tmp, 0, len);
            //Console.WriteLine(tmp);
            this.setText(new String(tmp));
        }
        catch (Exception f)
        {
            this.textBox1.Text += "error" + "\r\n";
        }
    }

```

```

        //textBox1.Text += serialPort1.ReadExisting();
        //this.setText(serialPort1.ReadExisting());
        //d.Invoke(new String(tmp));
        //this.setText()
    }
private void setText(string text)
{
    Graphics g;

    if (this.textBox1.InvokeRequired)
    {
        setTextCallback d = new setTextCallback(setText);
        this.Invoke(d, new string[] { text });
    }
    else
    {
        string[] loc = text.Split(' ');
        int x = Convert.ToInt32(loc[0]);
        int y = Convert.ToInt32(loc[1]);
        try
        {
            g = pictureBox1.CreateGraphics();
            g.FillEllipse(Brushes.Red, x, y, 10, 10);
            textBox1.Text = "";
            this.textBox1.Text += x.ToString() + ":" + y.ToString() + "\r\n";
        }
        catch (Exception e)
        {
            this.textBox1.Text += "error" + "\r\n";
        }
        //this.textBox1.Text += (string)text + "\r\n";
    }
}

private void button2_Click(object sender, EventArgs e)
{
    this.Refresh();
}

```

```

        textBox1.Text = "";
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {

        //
        this.comboBox1.Items.AddRange(System.IO.Ports.SerialPort.GetPortNames());

    }

    private void Form1_Load(object sender, EventArgs e)
    {

        string[] serialPorts = SerialPort.GetPortNames();

        foreach (string serialPort in serialPorts)
        {
            comboBox1.Items.Add(serialPort);
            if (comboBox1.Items.Count > 0)
            {
                comboBox1.SelectedIndex = 0;
            }
        }
    }
}

```

0.2時間分配表

工作 \ 月份	3月	4月	5月	6月	9月	9月	10月	11月	12月	1月	備註
擬定專題	★										
硬體組裝		★									
程式撰寫		★	★	★	★	★	★				
測試偵錯								★	★	★	
完成專題										★	
完成度(%)	5	20	25	35	50	75	85	90	95	100	

工作 \ 月份	3月	4月	5月	6月	9月	9月	10月	11月	12月	1月	備註
擬定專題	★										
硬體組裝		★									
程式撰寫		★	★	★	★	★	★				
測試偵錯								★	★	★	
完成專題										★	
完成度 (%)	5	20	25	35	50	75	85	90	95	100	

0.3 組員分工

	王志偉	陳暘	林柏翰
硬體組裝	○	○	
程式撰寫	○	○	
測試偵錯	○	○	
書面製作			○
海報製作			○
報告統整	○		○

0.4 參考資料

1. Boe-Bot Robot Kit 智慧型機器人教學平台

http://www.playrobot.com/BB/Boe_Bot_kit.html

2. 三角定位參考論文

http://www.mnd.gov.tw/Upload/200912/No-321-980722-47-62_273483.pdf

3. Visual C# 2010 教學網站

<http://msdn.microsoft.com/en-US/vstudio/hh341490>