

中華民國第 53 屆中小學科學展覽會

作品說明書

高職組 電子、電機及資訊科

第三名

091004

遠端遙控巡堂自走車

學校名稱：臺北市立內湖高級工業職業學校

作者：	指導老師：
職三 曾威凱	張俊雄
職三 蔡欣倪	
職三 陳鍵銘	

關鍵詞：Arduino 微控制器、全向輪自走車、

IP 攝影機

摘要

長久以來校園安全一直是大家最為關注的話題，但是幅員廣大的校園仍舊暗藏許多危機。多數民眾大都以為裝了監視器便能解決此一問題，殊不知一般的監視器大多採定點固定的方式架設，監看的範圍較為侷限，而且往往都會有看不到的死角。於是我們思考著，何不將資訊科技結合自動化裝置，把固定式監視器變成可移動式監視器。如此一來，不僅可以節省校園巡視所花費的人力成本，還可以應用在中小學教室的課堂巡察。實驗結果顯示，我們所設計的遠端遙控巡堂自走車，不僅能夠以循軌跡方式依照預設之教室路線做課堂巡察；還能夠以遠端遙控方式，直接在螢幕端控制自走車做即時監看、或錄影。相信有了這部自走車，校園安全能真正地落實，解決看不到死角的問題。

壹、研究動機

2012 年 12 月 14 日美國康乃狄克州西部紐鎮的桑迪霍克小學發生重大槍擊案件，造成 28 人喪生其中包括 20 名兒童，此一事件再度引發校園安全的疑慮。雖然我國的槍枝管制較為嚴密，但是幅員廣大的大專院校及中小學仍舊暗藏許多危機。一直以來大眾都以為裝了監視器便能解決此一問題，殊不知一般的監視器大多採定點固定的方式架設，監看的範圍較為侷限，且往往都會有看不到的死角。於是我們思考著，何不將資訊科技結合自動化裝置，把固定式監視器變成可移動式監視器。如此一來，不僅可以節省校園巡視所花費的人力時間與成本，還可以應用在中小學教室的課堂巡察。

由於本學期高三的專題製作實習課程，剛好是以單晶片為核心的微電腦控制系統，對於特定用途的控制，與個人電腦系統相比較，單晶片微電腦有更多的優點。於是我們選用 Arduino 公司所開發之 Mega 2560 微電腦控制平台，結合 Ethernet Shield 網路擴充板，可將特定功能需求之程式碼(參考高一套裝軟體實習課程所學的 VB 程式設計)寫入單晶片中，再透過我們所設計的介面電路(參考高二電子學實習課程及高三電子電路實習課程)來控制全向輪。最後在我們自組的全向輪自走車上裝置壹台 IP 攝影機，成功地設計出一部可移動式的遠端遙控巡堂自走車。

貳、研究目的

要想把定點固定式監視器變成可移動式監視器，又能節省時間及人力成本，非要一部裝了眼睛的機器人不可，而這樣的機器人好像只有在電影裡出現過，如果真的存在應該也是非常昂貴的。如何能在有限的資源下，完成這個不可能的任務呢？這就是我們研究的主要目的。首先，我們選用 Arduino 公司所開發之 Mega 2560 微控制器作為系統的核心，它的成本低廉又易開發。然後，透過我們自製的介面電路來控制裝有三個全向輪的機器人。最後，在這機器人上面架設壹台 IP 攝影機，即可組合出一部便宜又好用的可移動式遠端遙控巡堂自走車。

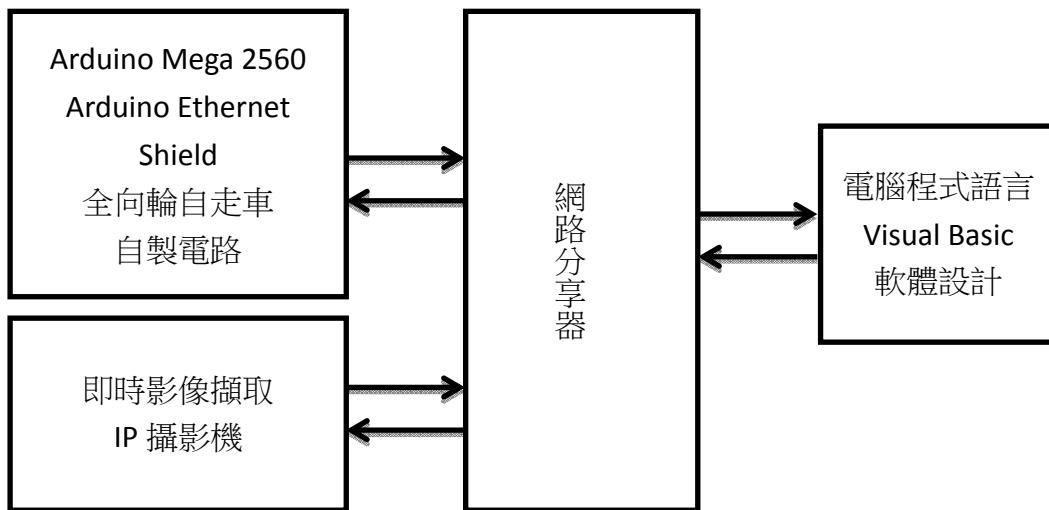
參、研究設備及器材

名稱	規格	數量
個人電腦	Windows XP, Intel Pentium D 805, 1G RAM	1
無線分享器	EDIMAX BR-6228nC	1
IP 無線攝影機	Vstarcam H6837WI	1
鋁管三腳架	WF-31N	1
專業級單把手大腳架	SOMITA WT-3520	1
微控制器開發板	Arduino Mega 2560	1
網路擴充板	Arduino Ethernet Shield	1
直流馬達	12V, 200rpm	3
強扭力直流減速馬達	5~12V, 25~60rpm	1
全向輪	2052k	3
陀螺儀	GY-50 L3G4200	1
鎳氫充電電池	9.6V1600mAh	1
蒸盤	直徑 30 cm 多孔洞	1
IC	7806(TO-220)	1
IC	7808(TO-220)	1
IC	LM7805K (TO-3)	1
IC	L293D	2
鋁材	厚度 1mm (60mm * 50mm)	3
PVC 水管	U 型扣環	3
DC 座	1.3 母座	2
DC 座	2.1 母座	1
其他	螺絲、螺母、銅柱	少許

肆、研究過程或方法

一、系統方塊圖

我們所設計之可移動式的自動化巡堂系統（系統方塊圖，如圖一），主要是想利用機器人來取代人力之校園巡視工作，或是中小學校長、教務主任例行性教室的課堂巡察工作。此系統不僅可以依照預先設定好之巡堂路徑模式、還能以遠端遙控模式做即時監看。至於機器人的形式，必須選用全向輪型機器人，才能符合我們自走車的需求。



圖一 系統方塊圖

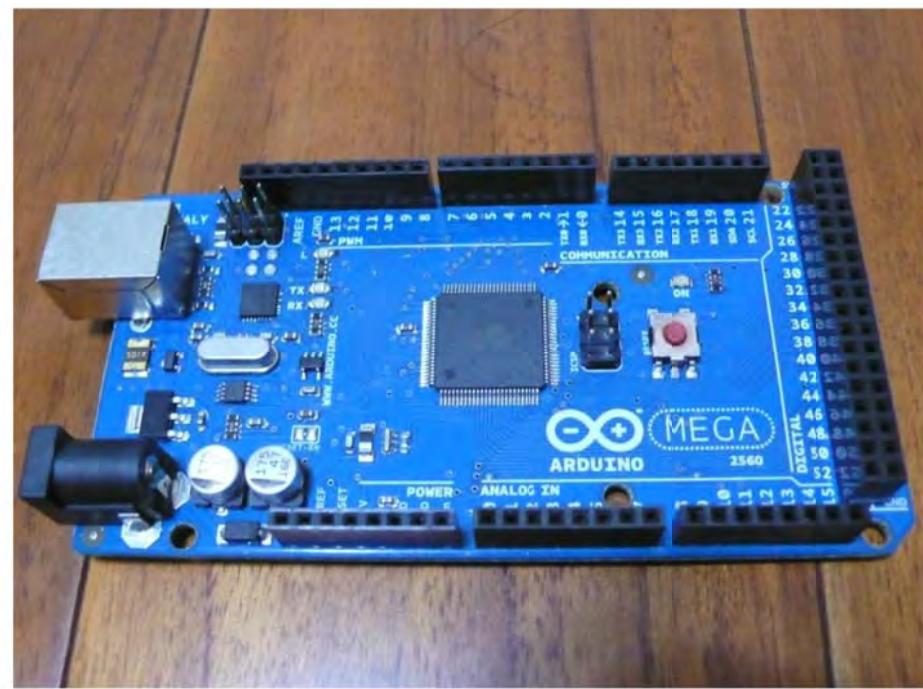
二、全向輪自走車

(一) 微控制器：

微控制器 Arduino Mega 2560 (如圖二) 是整個系統的核心，它是創用 CC (Creative Commons) 開放的電路設計概念，使用類似 Java 或 C 語言的開發環境，可簡單的與各式各樣的電子元件連接。官方網站也提供大量的函式庫以供參考，電源部份則由 USB 介面來提供，也可以獨立運行使用，建議電源輸入 7~12V。Arduino 微控制器 Mega 2560 的電氣特性規格如表(一)：

表(一)

工作電壓	5V
電源建議輸入電壓	7-12V
電源極限輸入電壓	6-20V
數位 I/O Pins	54 (其中提供 15 個 PWM 輸出)
類比輸入 Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V	50 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



圖二 Arduino Mega 2560

(二) 網路擴充板：

Arduino Ethernet Shield 網路擴充板 (如圖三) 主要是為 Arduino Mega 2560 所量身設計的，它可以直接堆疊在 Arduino Mega 2560 之上 (如圖四)，讓 Arduino 微控制器擁有連接網路的功能。網路擴充板除了本身可當做簡易的網頁伺服器外，還具有傳輸 UDP 封包的功能。



圖三 網路擴充板



圖四 Ethernet Shield 堆疊在 Mega 2560 上

(三) 全向輪自走車

1. 驅動方式

首先我們使用直徑 30cm 的蒸盤當做自走車的基座，然後將全向輪(如圖五)安裝在基座下方，每間隔 120° 安裝一個，共裝置三個(如圖六)。全向輪自走車的驅動方式係利用三個輪子彼此間的相互作用力來完成，它可以輕易地向前進、往後退，

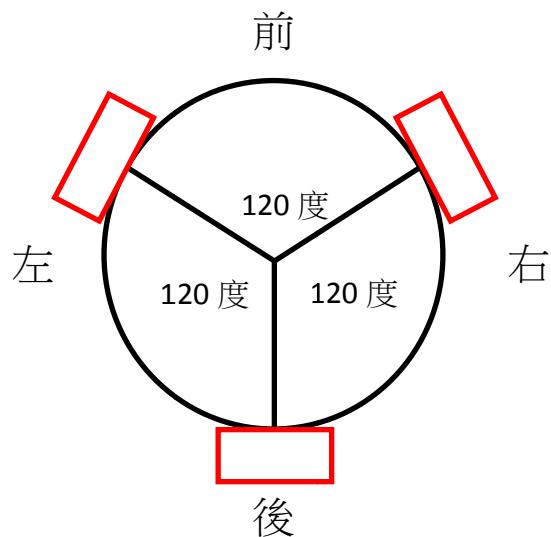
在轉彎時也能夠原地旋轉不需要迴旋的空間。全向輪自走車示意圖，如圖七所示。



圖五 單一個輪子



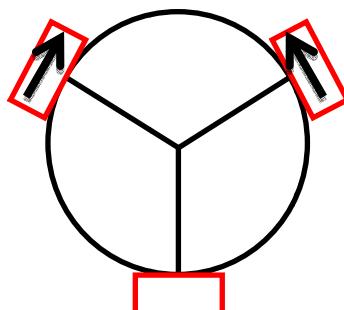
圖六 三輪分布方式



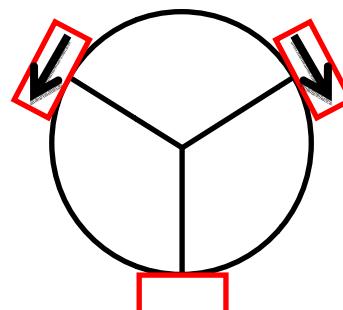
圖七 全向輪自走車示意圖

2. 移動方式

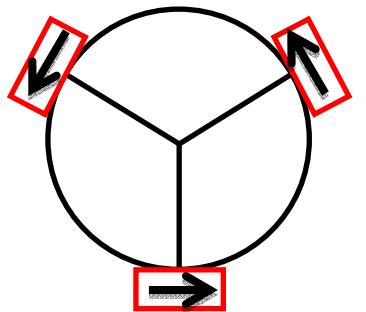
全向輪自走車的移動方式如下：當它向前進時，只要控制左、右兩輪同時向前旋轉（如圖八）。而當它往後退時，只要控制左、右兩輪同時向後旋轉（如圖九）。如果想要向左轉，讓左、右、後三輪同時逆時針旋轉即可（如圖十），其中左輪的實際轉速須視旋轉角度及地面狀況做微調。如果想要向右轉，讓左、右、後三輪同時順時針旋轉即可（如圖十一），其中右輪的實際轉速須視旋轉角度及地面狀況做微調。



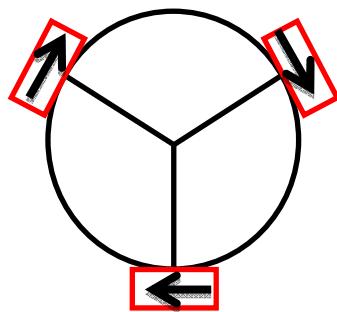
圖八 前進



圖九 後退



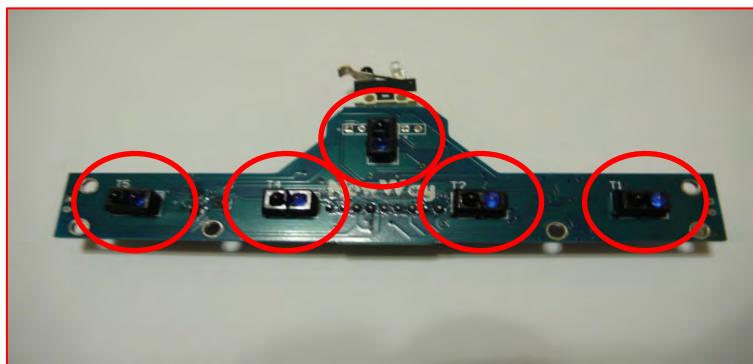
圖十 向左轉



圖十一 向右轉

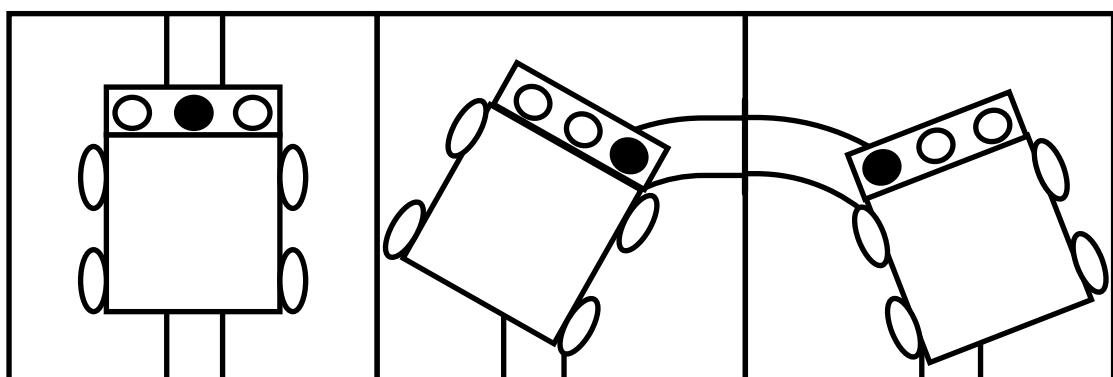
3. 路線設定

課堂的巡察路線如果是經常性固定的模式，那我們只要在預設的巡堂路線上，貼上黑色絕緣膠帶，做為自走車的路徑，然後搭配反射式紅外線感測器（如圖十二）之使用即可。當自走車上之反射式紅外線感測器感測到黑色時，將傳回低電位；相反地，其他顏色則傳回高電位。



圖十二 反射式紅外線感測器

雖然反射式紅外線感測器模組上共有五個感測元件，而目前我們只使用到中間三個。其基本動作原理如下：當中間的感測元件被啟動時，自走車會向前行；當右側感測元件被啟動時，則自走車向右轉；當左側感測元件被啟動時，則自走車向左轉。另外，我們還加入了直角轉彎的處理，當中間與右側同時感測到路線時原地向右旋轉，而中間與左側同時感測到路線時原地向左旋轉，如圖十三。



圖十三 自走車行徑示意圖

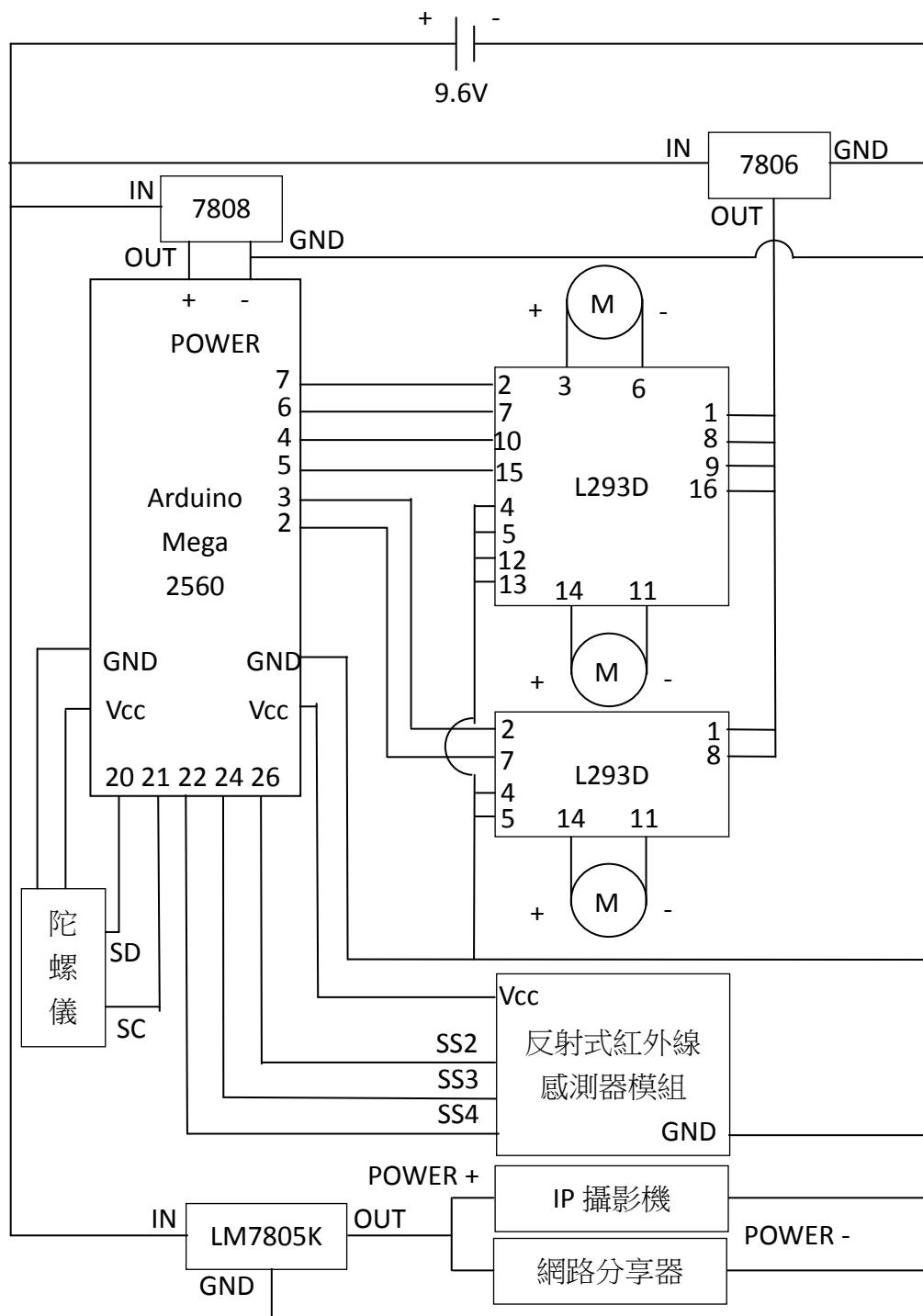
4. 直線校正

全向輪自走車的移動，容易受基座重量及環境(路面平整度)所影響，尤其是直線行走時所造成的偏轉會特別明顯。因此，我們找到陀螺儀來做直線校正。它的基本原理是：透過陀螺儀來取得全向輪自走車左、右偏轉的角速率，進而對左、右兩

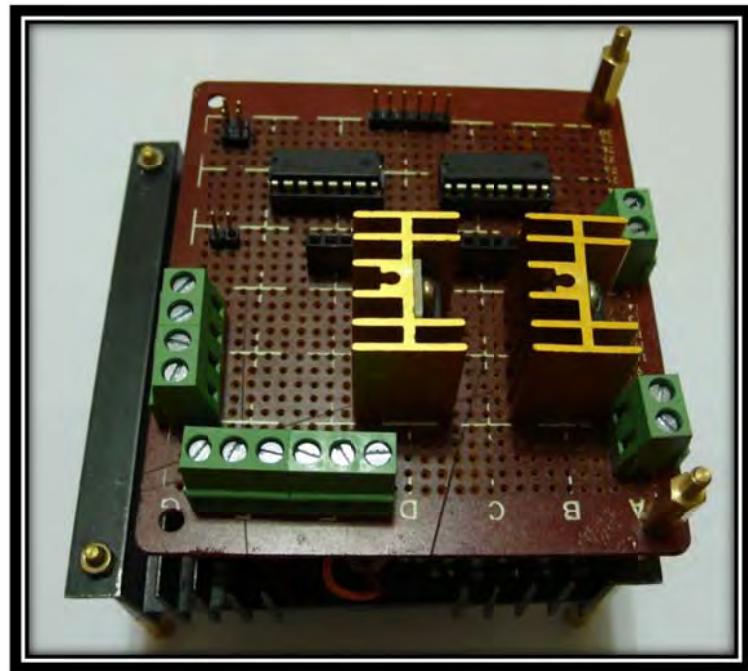
輪的速差做修正，以達到直線校正的功能。

(四) 電源電路

本系統是由許多不同模組所組成的，因此整個系統的供電狀況甚為重要，而模組之間有些可以直接並聯使用，有些則需要其他電路的輔助才能控制。此外，有的模組可以直接使用乾電池，有的卻必須使用穩壓電路之後才能供給所需的電壓，如：直流伺服馬達等。由於 Arduino 微控制器的直流輸出信號最大只有 40mA，所以必須藉由 H-bridge 電路(L293D)來供給。至於其他信號控制，則使用 Arduino Mega 2560 電路板內建功能即可，如圖十四。電源電路實體圖，如圖十五。



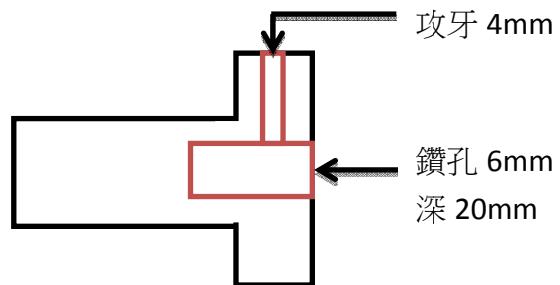
圖十四 電源電路系統方塊圖



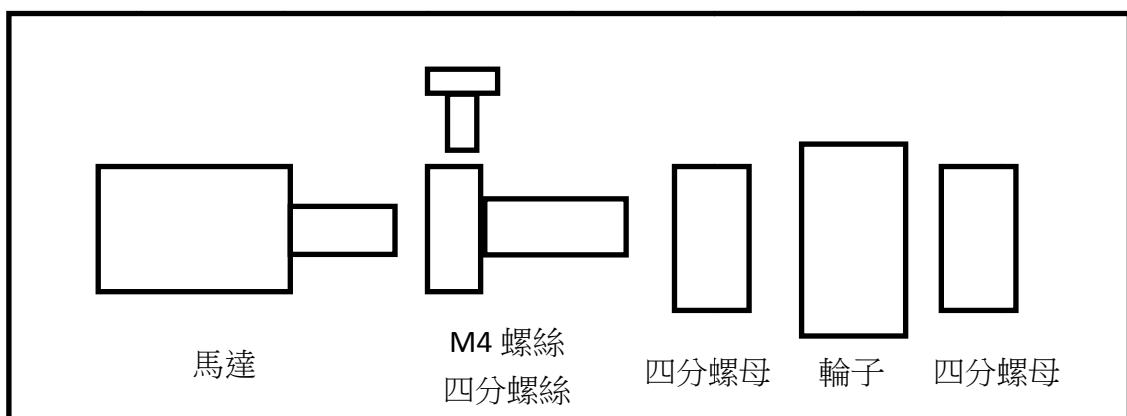
圖十五 電源電路實體圖

(五) 全向輪的組裝及攝影機的固定

市售的全向輪，通常沒有附加輪子與馬達之間的聯軸器，因此我們只好自己為輪子與馬達來量身打造了。已知馬達軸心為 6mm，而全向輪內徑為 12.7mm，兩者相差一倍左右。要如何製作聯軸器呢？我們想到使用四分螺絲。首先，要在螺絲上面鑽孔使得馬達軸心可以嵌入，然後在側面鑽孔，使其可以鎖入螺絲當做固定插梢，如圖十六。接著就可以開始組裝全向輪，輪子、聯軸器與馬達之間的分解圖，如圖十七。



圖十六 四分螺絲所製作之聯軸器



圖十七 馬達、聯軸器、輪子之組裝

另外，如何固定攝影機在全向輪自走車上呢？我們想到利用相機的三腳架，將它架設在面積有限的底盤上。在不破壞腳架的情況下，我們在底盤上做了固定腳架的凹槽，讓三腳架能在底盤上站穩。簡單使用 1mm 厚的鋁片(約 6x5cm)，然後在中間鑽一個 12mm 寬的凹槽，相機的腳架即可順利嵌入固定，如圖十八。



圖十八 固定三腳架的凹槽

(六) 控制程式

Arduino 的基礎架構為 void setup() 以及 void loop()，其中 void setup() 為啟動後只能執行一次，作為設定初始值之用，接著執行 void loop() 直到電源中斷為止。若要將 Arduino 連接到網路上，則需要指定一個 MAC 位址、IP 位址給它，以及所要使用 UDP 傳輸的 Port Number，部分程序碼如下：

```
void setup() {                                //初始值設定
    delay(10000);                            //等待開機
    Ethernet.begin(mac,ip);                  //指定 MAC 位置、IP 位置
    Udp.begin(localPort);                    //指定 UDP 傳輸 Port
    for(int i=0;i<=2;i++){                  //指定輸出腳位
        pinMode(RDCmotor[i], OUTPUT);
        pinMode(LDCmotor[i], OUTPUT);
        pinMode(DCmotor[i], OUTPUT); }
    for(int i=0;i<=3;i++){                  //指定輸入腳位
        pinMode(ir[i],INPUT); }
    carstop();                                //自走車啟動後先停留原地
}
```

我們設計的遠端遙控巡堂自走車主要有兩種運作模式：(1)循軌跡路線方式，(2)遠端遙控方式。因此，需要不斷地檢查有無變換功能的情況發生，void loop() 程序負責這項工作。

```
void loop()
{
    String temp;
    int packetSize = Udp.parsePacket();          //取得緩衝區內資料大小
    if (packetSize)
```

```

{
    Udp.read(packetBuffer,UDP_TX_PACKET_MAX_SIZE);
    temp=String(packetBuffer);
    if (temp=="6") command="6"; //讀取資料，"5"循線，"6"遙控
    if (temp=="5") command="5";
    if (temp=="5" || temp=="6") carstop(); } //切換功能後，自走車停止移動

```

1. 循軌跡路線功能：

```

if (command=="5")
    //讀取反射式紅外線感測器模組資料
    { for (int i=0;i<=3;i++) irdata[i]=digitalRead(ir[i]);
        //依據資料做出判斷決定行進方向
        //forward()前進、back()退後、carstop()停止
        //left1()小幅度左轉、right1()小幅度右轉
        //left2()原地左轉、right2()原地右轉
        if (irdata[0]==HIGH && irdata[1]==LOW && irdata[2]==HIGH) forward();
        if (irdata[0]==LOW && irdata[1]==HIGH && irdata[2]==HIGH)
            { left1();
                delay(250);} //減少向左行進方向修正次數
        if (irdata[0]==LOW && irdata[1]==LOW && irdata[2]==HIGH) left2();
        if (irdata[0]==HIGH && irdata[1]==HIGH && irdata[2]==LOW)
            { right1();
                delay(250);} //減少向右行進方向修正次數
        if (irdata[0]==HIGH && irdata[1]==LOW && irdata[2]==LOW) right2();
    }

```

2. 遠端遙控功能：

```

// "0"停止、"1"前進、"2"後退、"3"原地左轉、"4"原地右轉
if (command=="6")
{ if(temp=="1" || temp=="2")
    { getGyroValues(); //取得陀螺儀資料，進行直線校正
        if (Lspeed==180 && Rspeed==120)
            { Lspeed=150;
                Rspeed=150;}
        if (Lspeed==120 && Rspeed==180)
            { Lspeed=150;
                Rspeed=150;}
        if (Lspeed==180 && Rspeed==120) Rspeed-=2;
        if (Lspeed==120 && Rspeed==180) Rspeed+=2;
        if (Lspeed==120 && Rspeed==120) Rspeed+=2;
        if (Lspeed==180 && Rspeed==180) Rspeed-=2;
        if (temp=="1") Lspeed+=2;
    }
}

```

```

if (z<-2 && Lspeed>120 && temp=="1") Lspeed-=2;
if (z>2 && Lspeed>120 && temp=="2") Lspeed-=2;
if (z<-2 && Lspeed<180 && temp=="2") Lspeed+=2;
if (temp=="1") forward();
else back();
delay(50);
carstop(); }
if (temp=="3") left2();
if (temp=="4") right2();
if (temp=="0") carstop(); }

```

控制全向輪自走車移動的副程式 move(int arc) 當中運用到三角餘弦函數去計算各全向輪的運轉速率，以達到 360 度任意位移的移動方式。

```

float pi=3.14159265359;
int p = 250;
void move(int arc) {
    arc = arc % 360;
    float D_cos = cos(arc*pi/180);
    float L_cos = cos((120+arc)*pi/180);
    float R_cos = cos((60+arc)*pi/180);
    if(D_cos>=0){
        analogWrite(DCmotor[0], abs(D_cos*p));
        analogWrite(DCmotor[1], 0);}
    else{
        analogWrite(DCmotor[0], 0);
        analogWrite(DCmotor[1], abs(D_cos*p));}
    if(L_cos>=0){
        analogWrite(LDCmotor[0], abs(L_cos*p));
        analogWrite(LDCmotor[1], 0);}
    else{
        analogWrite(LDCmotor[0], 0);
        analogWrite(LDCmotor[1], abs(L_cos*p));}
    if(R_cos>=0){
        analogWrite(RDCmotor[0], 0);
        analogWrite(RDCmotor[1], abs(R_cos*p));}
    else{
        analogWrite(RDCmotor[0], abs(R_cos*p));
        analogWrite(RDCmotor[1], 0); }
}

```

三、即時影像擷取

(一) IP 攝影機

市售的 IP 攝影機都能直接連上網路，它可以透過有線或無線方式與網路連接。我們特別選用一台含有無線網路功能的 IP 攝影機 (如圖十九)，攝影鏡頭的控制將不需要透過 Arduino 微控制器，直接透過網路訊號來遙控即可，也容易做即時影像的擷取。廠牌 Vstarcam 的 H6837WI，它的成本低且有提供完整的 SDK 讓我們能夠自行開發程式。其特性規格如表(二)：

表(二)



圖十九 IP 攝影機

壓縮方式	H.264
FPS	1-30
解析度	VGA(640*480) QVGA(320*240)
鏡頭功能	自動白平衡 自動增益控制 自動背光補償
聲音輸入	內建麥克風
聲音輸出	內建喇叭 3.5mm 音源孔
有線網路	10Base-T/100Base-TX 乙太網路
無線網路	802.11b/g/n

(二) H.264 協定

你會發現越來越多攝影機強調 H.264 協定，它是什麼樣的技術呢？在此讓我們來簡單的介紹一下。H.264 又稱 MPEG4-10 或 AVC，第二版標準發表於 2005 年 3 月，壓縮品質優於 MPEG4、MJPEG。其壓縮原理是利用影像在時間與空間上的相似性，將人眼所不能發覺的部分抽離，這樣就能達到影像壓縮的目的。而在影像編碼前先將整張影像分割成許多小區塊，再對這些小區塊進行編碼，所以每接收到一個區塊的資料就能馬上進行解碼。一旦資料在傳遞過程中遺失，也只會影響該筆資料的正確性。

四、系統軟體設計

軟體控制部分我們採用 Visual Basic 語言來撰寫，結合廠商所提供的 SDK 開發程式，讓使用者能透過電腦監看畫面，並且能有效地控制全向輪自走車的行走方向、攝影機的角度等。程式控制區主要分為兩大區塊，自走車控制以及攝影機控制，其中自走車控制又可分為循軌跡路線模式或是遠端遙控模式。要控制 IP 攝影機，必須先取得與 IP 攝影機的連線再擷取影像，許多初始值必須參考 IP 攝影機 SDK 手冊來做設定。

(一) 連線資料：

nDev = 918

bLan = 1

```
IP = "192.168.1.126"  
port = 81  
DevName = "H6873WI"  
Acc = "admin"  
Pwd = "123"
```

(二) 介面工具設定：

```
AxAxRemote1.ConnMode = 0  
AxAxRemote1.TCPMode = 1  
AxAxRemote1.ShowTitle = 0  
AxAxRemote1.Selected = 1  
AxAxRemote1.Lan = "b5"  
AxAxRemote1.ShowOSDName = 1  
AxAxRemote1.SwitchLayout("1 x 1")  
AxAxRemote1.Width = 640  
AxAxRemote1.Height = 480  
AxAxRemote1.ShowToolBar = 0  
AxAxRemote1.Listen = -1
```

(三) 開始連線：

```
AxAxRemote1.AddDev4(nDev, bLan, IP, port, DevName, Acc, Pwd, 1)
```

```
AxAxRemote1.TurnImg = 0  
AxAxRemote1.ConnectAll()  
AxAxRemote1.Start()
```

當遠端遙控巡堂自走車到達定位後，還能控制鏡頭的轉動方向，其中第三個參數代表其動作方向，「0」為上轉，「3」為下轉，「1」往左轉，「2」往右轉，「-1」則停止。

```
AxAxRemote1.PTZ(1, 1, 0, 50, 0, 0)
```

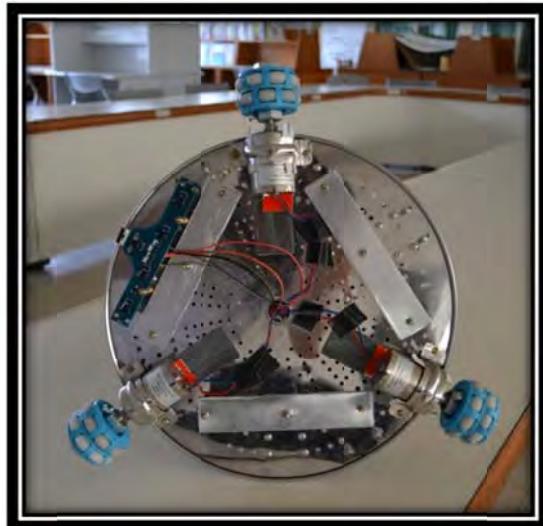
功能切換以及自走車移動方向控制，參數值可透過 UDP 傳輸送到 Arduino 開發板上，參數「1」前進、「2」後退、「3」左轉、「4」右轉、「5」循線、「6」遙控。

```
Dim Data As Byte() = Encoding.GetEncoding(950).GetBytes(參數)
```

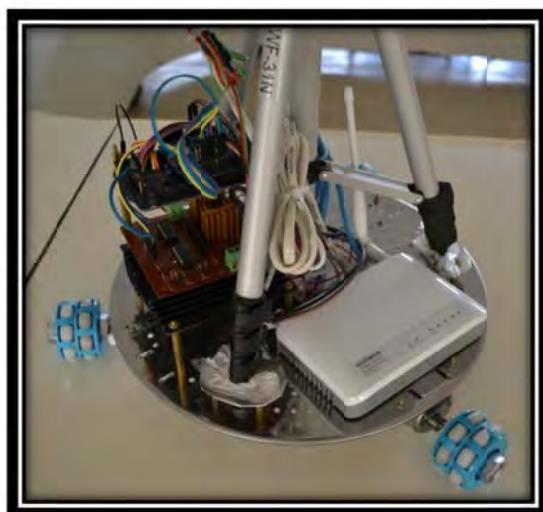
```
Client.Send(Data, Data.Length, SeverIPEndPoint)
```

五、自走車硬體組裝

首先我們使用直徑 30cm 的蒸盤當做自走車的基座，然後將全向輪安裝在基座下方，每間隔 120° 安裝一個共裝置三個，接著選定其中兩個全向輪，於中間位置固定我們的反射式紅外線感測器模組，如圖二十。再將所有的控制電路板以及無線分享器等元件一一放置於基座之上，如圖二十一。最後放上攝影機的三腳架，即可組裝完成我們的自走車，三腳架之高度可手動調整，以符合教室窗台的高度，利於攝影機鏡頭之監看，整體高度大約在 50cm~114cm 之間，完整全貌如圖二十二。



圖二十 自走車基座下方



圖二十一 自走車基座上方



圖二十二 自走車全貌

伍、研究結果

一、使用者介面

電腦端的控制介面我們使用 Visual Basic 語言工具來撰寫，介面主要分為三大區塊：影像顯示區、IP 攝影機控制區、自走車遙控區，如圖二十三。目前影像顯示區所看到的影像為監視器的即時畫面，而視窗的正下方，可選擇巡堂自走車的操作模式(二選一)：【循線】或【遙控】。在視窗右下方自走車遙控區的四個按鈕，可以控制自走車：前進、後退、左轉、或是右轉。另外，在視窗左下方的 IP 攝影機控制區有五個按鈕，可控制 IP 攝影機的鏡頭，讓它上下移動、左右轉動、或拍照儲存成單張影像。在視窗右側則用於輔助：功能包括有攝影機置中歸位、錄影、或循線模組。



圖二十三 使用者介面

二、巡堂自走車的操作模式

(一) 循軌跡路線模式

首先，我們使用黑色膠帶在地板上貼成一條封閉迴路，如圖二十四。然後將自走車移動至軌跡上方，當自走車上之反射式紅外線感測器偵測到黑色時，將傳回低電位；相反地，其他顏色則傳回高電位。如此一來，藉由感測器模組上的三個感測元件，自動偵測並傳回控制訊號，使得自走車開始向前行、向右轉、向左轉、或是原地向右旋轉、向左旋轉。若是將自走車移開貼有膠帶的軌跡路線圖外，自走車將靜止不動。



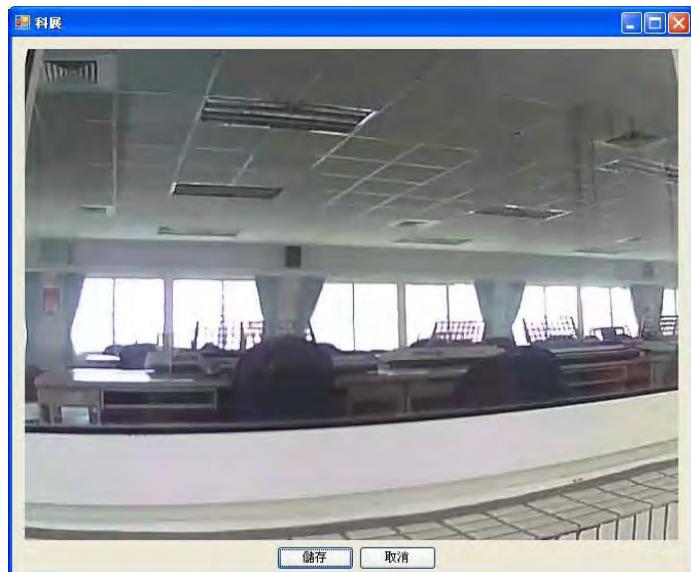
圖二十四 循軌跡路線行走

(二) 遠端遙控模式

當選擇遠端遙控模式時，我們是將控制訊號送到無線網路的分享器，再連接到 Arduino 微控制器上，可隨意控制自走車行走；當滑鼠按住時可持續移動，若滑鼠放開時則停止不動，訊號有效控制範圍約 60 坪。

(三) 拍照存檔

若想把監看到的畫面拍攝下來，只要點選【拍照】即可儲存成單張影像，如圖二十五。



圖二十五 電腦端控制介面

(四) 錄影與循線模組

當無人監看時，可以把即時監看到的畫面錄影存證，只要點選錄影區的【開始】即可開始錄製 avi 視訊檔。至於循線模組功能，是用來判斷自走車有沒有在循線的軌跡路徑上，左、中、右分別代表反射式紅外線感測器模組上的三個感測元件。

陸、討論

其實我們一開始曾經使用 CMUcam v4 視覺模組與 LS-Y201，但這兩項在與 Arduino 微控制器一起應用時，反而無法取得較佳的即時影像，因為 FPS 大約只有 1 張或 2 張影像較不流暢，會造成停格現象。而且若要做到遠端遙控，Arduino 微控制器還要透過無線網路擴充板，如此一來 CMUcam v4 視覺模組與 LS-Y201 所取得的即時影像，更無法直接透過無線網路介面做資料的傳輸。因此，才改用越來越多人使用的 IP 攝影機來取代。

至於 IP 攝影機的固定方式，我們曾經使用銅柱串接 V 型鋁板當支柱，但是實驗結果整個機身晃動幅度過大，並且銅柱容易鬆動非常不牢固。最後回到原點思考，架設攝影機就應該使用相機三腳架才適合，只要想辦法固定三腳架在底盤上，就能得到不錯的效果。再者，三腳架之高度雖然可以手動調整，以符合教室窗台的高度，利於攝影機鏡頭之監看；但是，若能夠配合遠端遙控模式，讓 IP 攝影機可以自動升降，我們的「遠端遙控巡堂自走車」將更為實用。

柒、結論

實驗結果顯示，我們所設計之可移動式的遠端遙控巡堂自走車，能夠以循軌跡方式依照預設之教室路線做課堂巡察；也能夠以遠端遙控方式，直接在螢幕端控制自走車做即時監看、或錄影。相信有了這部自走車，校園安全能真正地落實，也能解決看不到死角的問題。在這研究過程中，我們學習了 Arduino 的使用方法，配合學校課程所學的電路以及程式設計，真實呈現一台能夠遠端遙控的巡堂自走車，心中的成就感油然而生。另一方面也體會到，紙上的構想雖然看來容易，但實際執行起來卻不如想像中的簡單，而且差異還蠻大的。只好一邊製作一邊進行調整，從中所吸收到實作的經驗更是無價的。

機械領域對於學習資訊科技的我們，其實是很陌生的，一開始完全不知道從何處著手，但經過老師的從旁輔導，終於學會了部分機械的操作，希望未來還有機會發展其他的應用，學習更多這方面的技巧。藉此，也讓我們對於微控制器系統有更深一層的了解，軟體與硬體之間的配合也更有經驗。不斷重複地假設問題、小心求證，進而解決問題，科學研究原來是這麼的有趣。

捌、參考資料及其他

- 一、孫駿榮、吳明展、盧聰勇(2012)。最簡單的互動設計 Arduino 一試就上手(第二版)。臺北市：碁峯資訊。
- 二、Michael Margolis、徐德發(譯) (2012)。Arduino 錦囊妙計(第二版)。臺北市：碁峯資訊。
- 三、吳明哲、張志成、何嘉益、周家旬、黃鳳梅(2008)。Visual Basic 2008 學習範本。臺北市：文魁資訊。
- 四、翁義清(2007)。全向移動機器人之路徑追蹤控制。國立成功大學工程科學系碩士論文。
- 五、許佳興(2012)。利用感測器訊號於機器人運動行為之發展。國立臺北教育大學發展學校重點特色計畫。
- 六、Arduino 微控制器。<http://www.arduino.cc/>。
- 七、Arduino.TW。<http://arduino.tw/>。
- 八、IP 攝影機 Vstarcam H6837WI，<http://www.vstarcam.cn/IP-Camera-H6837WI.html>。
- 九、數位視訊月刊-H.264 先進視訊編解碼標準/郭其昌，
<http://www.tvro.com.tw/main5/maimtxt/H.264%E5%A0%B1%E5%B0%8E%EF%BC%8D%E5%B7%A5%E7%A0%94%E9%99%A2.htm>。

【評語】091004

1. 此作品乃結合即時影像擷取網路攝影機、具三顆全向輪車體、Mega 2560 微電腦控制平台等、以研製一台可移動式的遠端遙控巡堂自走車。
2. 此遠端遙控巡堂自走車，不僅能夠以循軌跡方式依照預設之路線做巡察，還能夠以遠端遙控方式，直接在螢幕端控制自走車做即時監看。
3. 自走車以三顆全向輪驅動與典型四顆固定輪驅動相比，在循軌跡工作中具有較佳之性能。
4. 能將學校所習得之知識與技能充分應用至本作品之研製上，並實際應用在校園內。
5. 整套系統具體完整，唯創新性仍有加強空間，如可適度將智慧能力加到自走車上等。