

投稿類別：資電類

篇名：板擦機器人

作者：

林瑋翔。私立東海高中。高三電子科甲班

王廷友。私立東海高中。高三電子科甲班

林家銘。私立東海高中。高三電子科甲班

指導老師：

張學龍 老師

邱顯泰 老師

壹、前言

現在的科技逐漸進步，越來越多的機器人，取代大大小小的工作。不管是使用方便或者是日常生活需求，都是個值得注意的目標。物聯網，讓所有的設備進行傳輸，也讓工廠向自動化邁進。資料也數據化，讓資料不再是文書而已，而是利用網路，來讓資料讀取和閱讀更加便利。本專題利用了 PHP，架設資料庫，並且採用網路的方式去傳輸照片。使用了 C#去讀取 PHP 上的資料，讓讀取照片不再那麼複雜。也方便使用者去判斷要擦拭的區域。並且利用 C#同步 ARDUINO 去控制車子移動。去取代一般人工擦拭的部分，也能減少擦拭，所需要的時間。

貳、研究動機

近年來機器人越來越盛行，許許多多的動作，都使用了機器手臂或機器人來取代，在某次的課堂上，我們偶然發現能夠藉此專題，來讓擦黑板的動作更加便利。於是我們便利用了手機拍照，將圖片傳到電腦，再用電腦框出要擦拭的區域，並且利用藍芽的方式，控制車子來移動。

參、主題與課程之相關性或教學單元之說明

專題「板擦機器人」，我們利用了專題製作、基本電學實習、電子學實習的課程，用團隊合作方式，決定專題的方向，來設計專題。相關課程，如表 1 所示。

表 1 主題與課程之相關性或教學單元

課程單元名稱	教學單元	備註
1.專題製作	蒐集資料、構想題目、製作系統功能方塊圖、硬體與軟體結合	
2.基本電學實習	零件識別、電路焊接	
3.電子電路實習	控制 IC 架構、電路設計	
4.感測器原理	重量感測	

肆、研究方法過程
一、研究流程圖

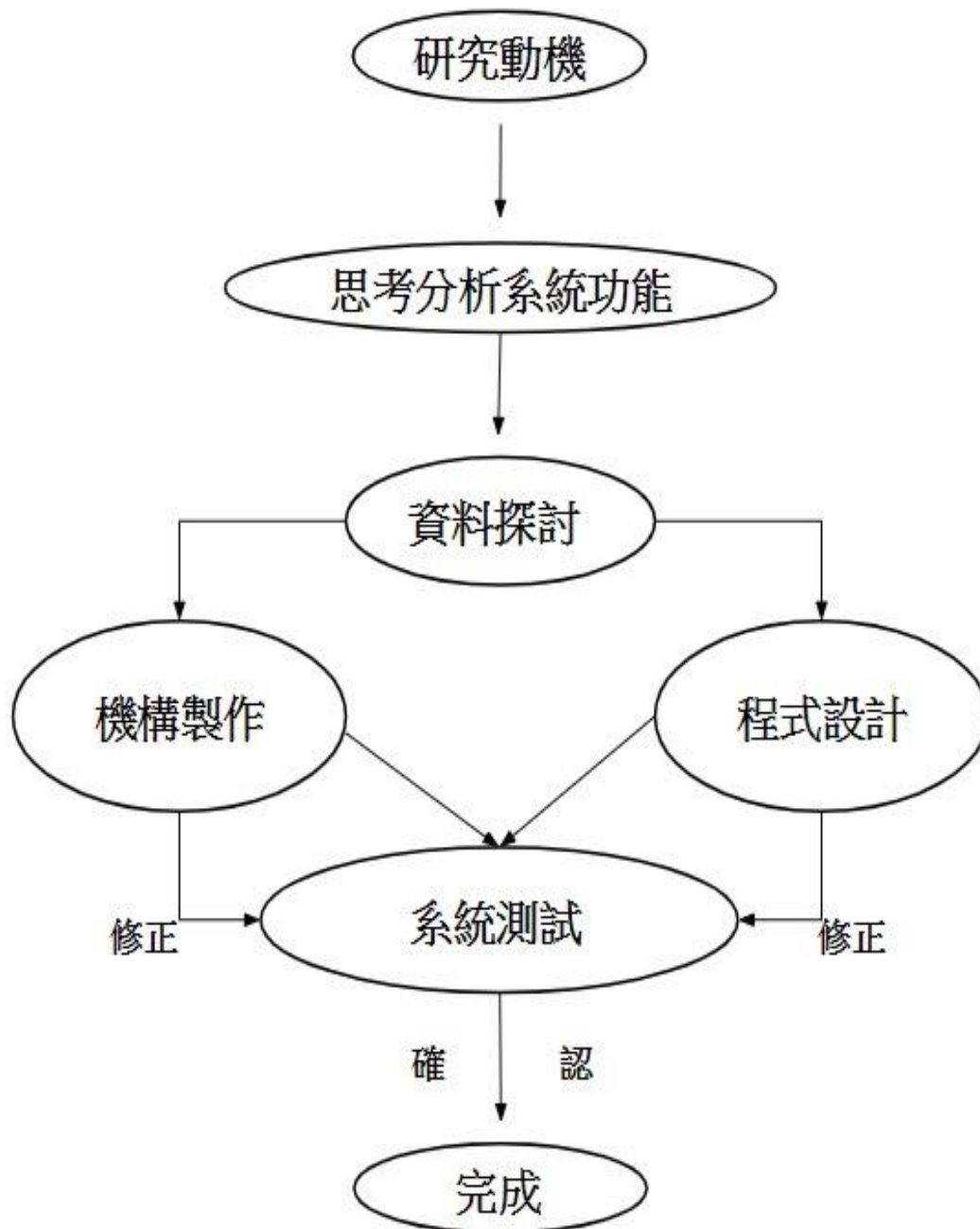


圖 1 研究流程

二、研究過程

一、硬體製作過程

(一)車體的組裝



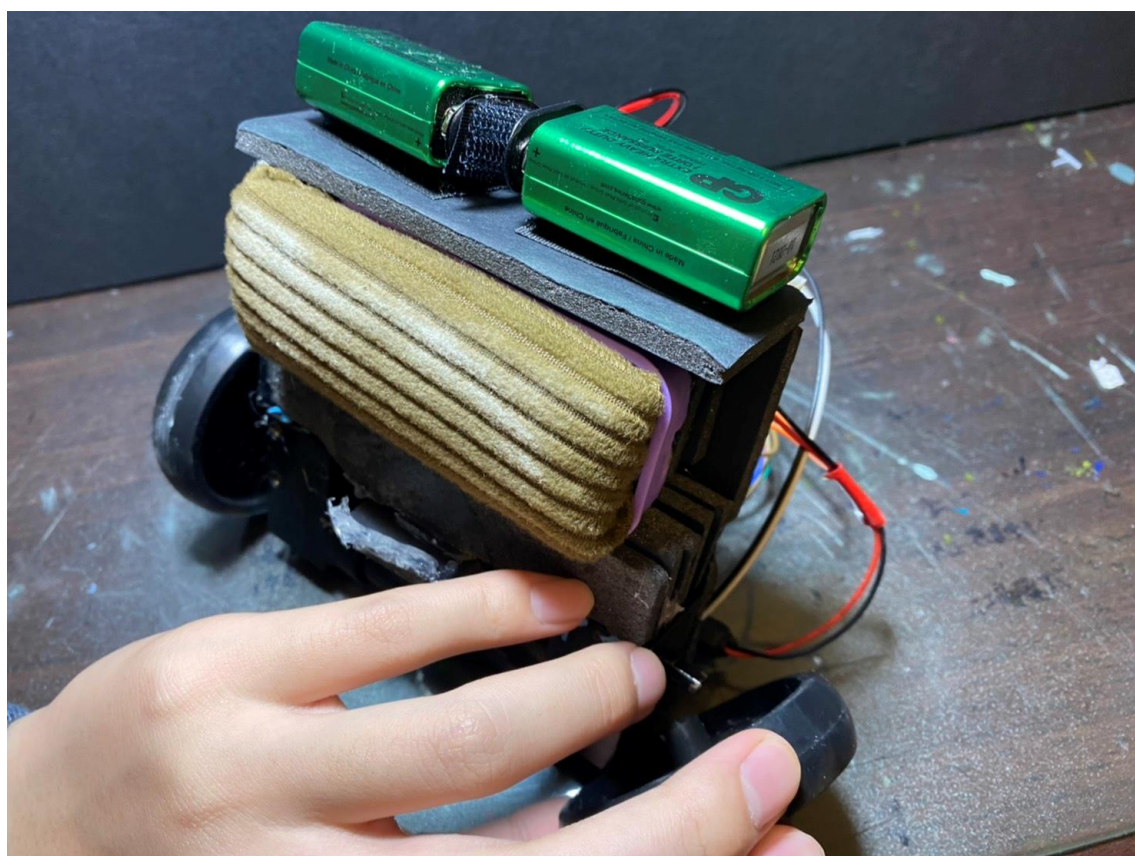
(圖一) 切割



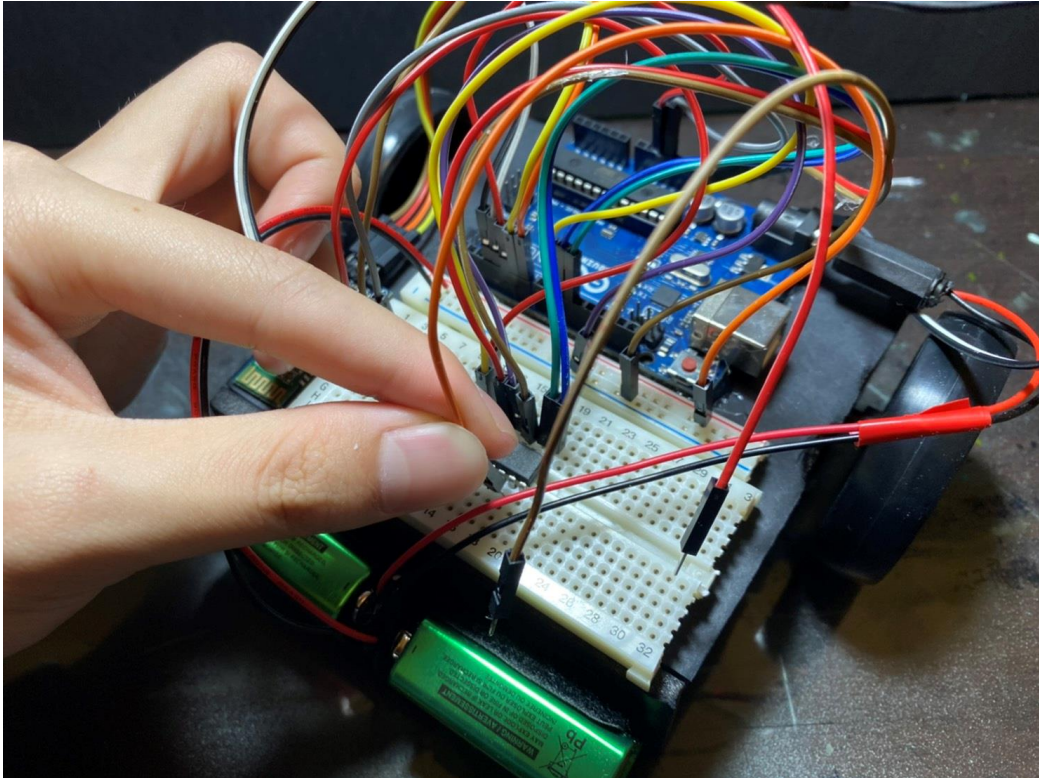
(圖二) 黏貼 1



(圖三) 黏貼 2

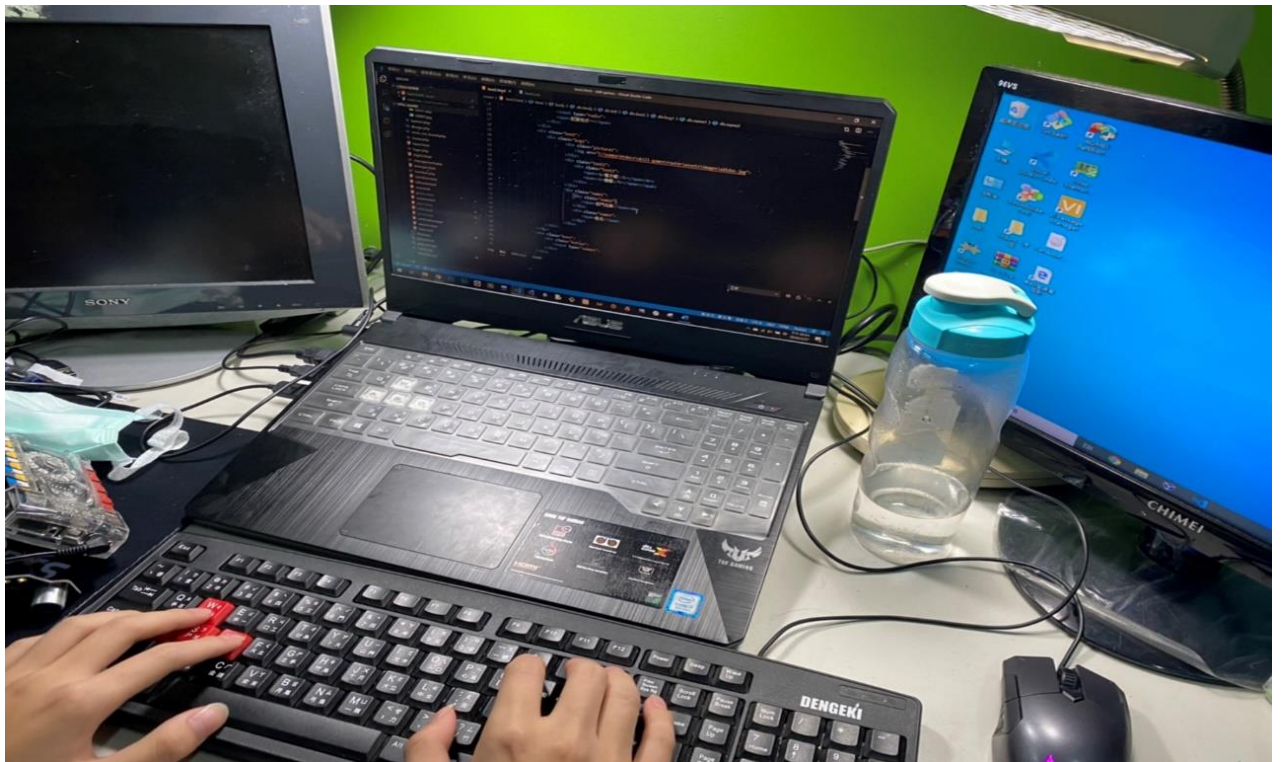


(圖四) 組裝機構

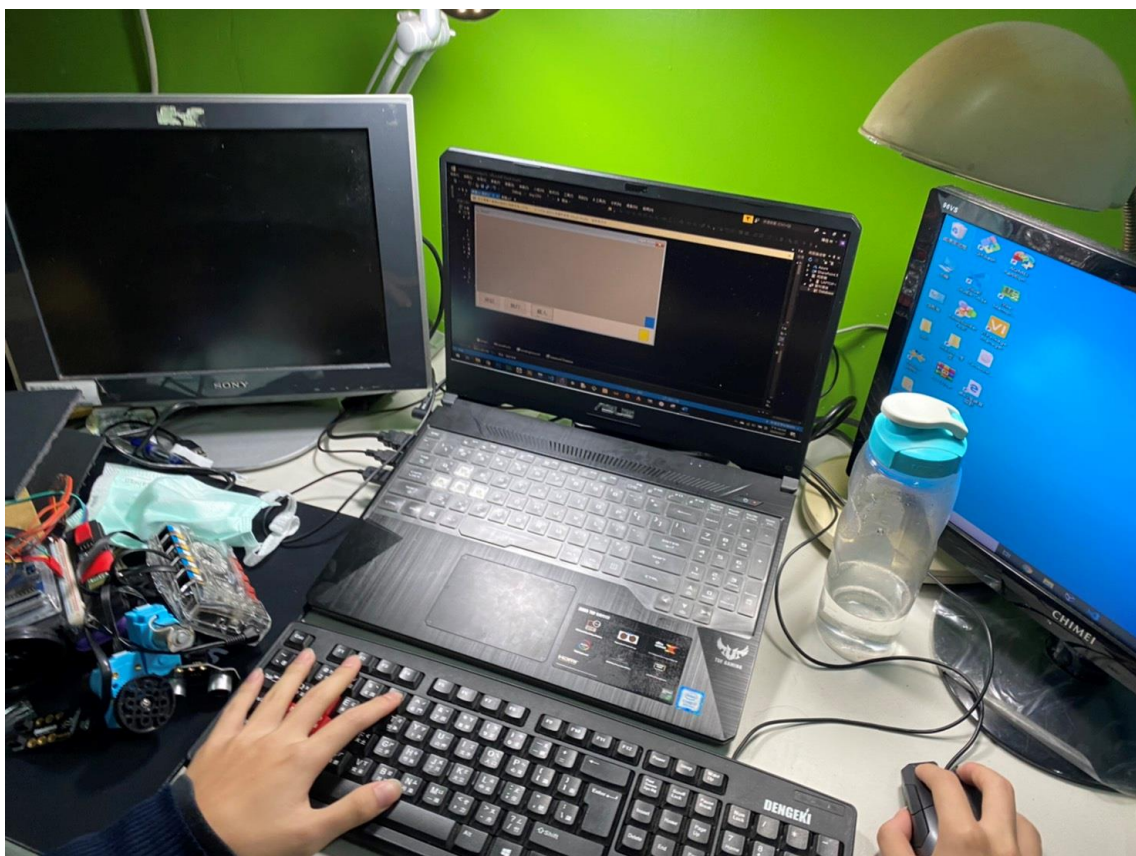


(圖五) 電路接線

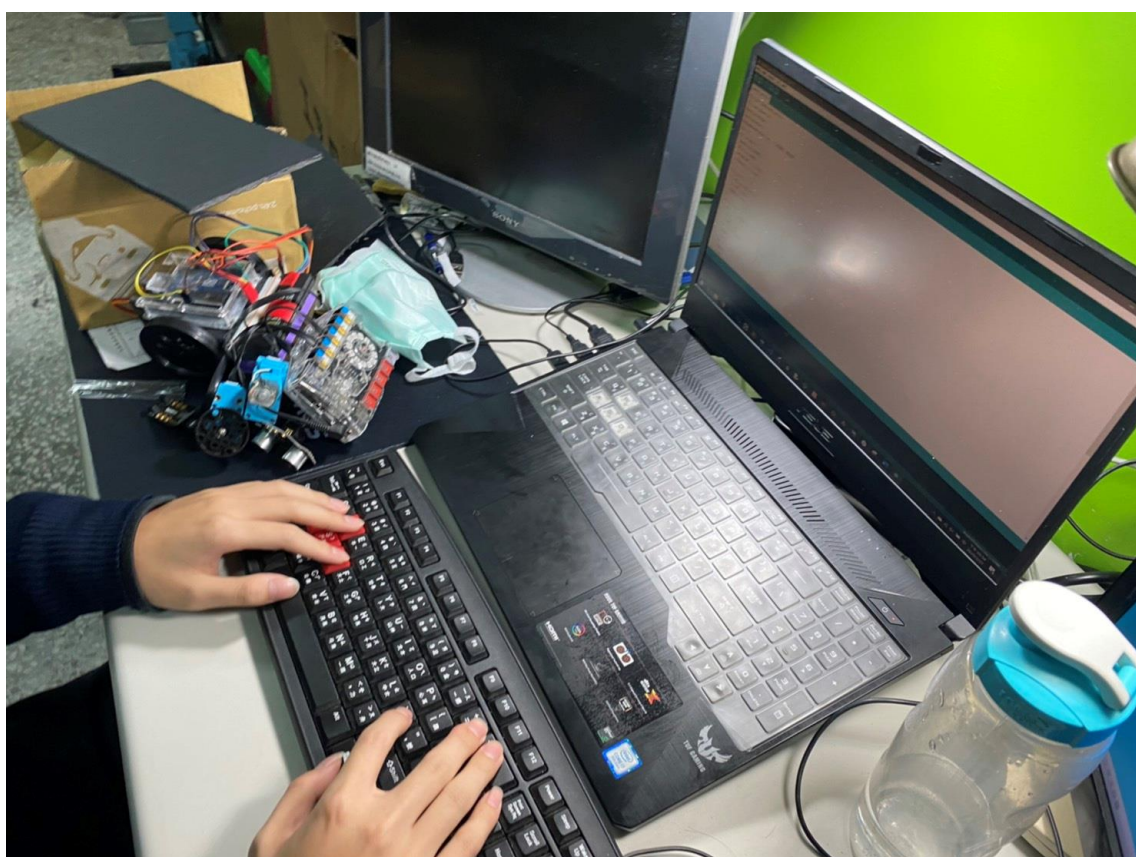
二、軟體製作過程



(圖六) PHP 程式碼





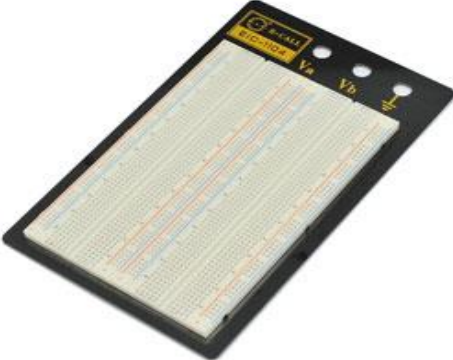

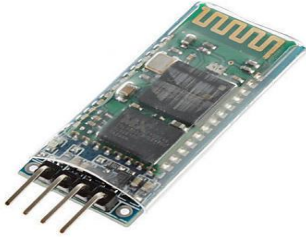

(圖七) 撰寫 C#程式碼



(圖八) 撰寫 Arduino 程式碼

三、硬體設備

表 2 設備

設備	
Arduino uno 板	電腦
	
麵包板	MG996R
	
HC-06	手機
	

四、程式撰寫

```
{
    public partial class Form1 : Form
    {
        //初始值
        int a=0,x1,y1,d=1,f;
        string x, y;
        int [,] b = new int[2, 2];
        int[] c = new int[4];
        public Form1()
        {
            InitializeComponent();
        }
        private void timer1_Tick(object sender, EventArgs e)
        {
            //回家時的運算
            if (f == 0)
            {
                for (int i = c[1]; i <= c[3]; i += 10)
                {
                    pictureBox3.Location = new Point(c[0], i);
                    Thread.Sleep(100);
                    pictureBox3.Visible = true;
                    pictureBox3.BackColor = Color.Yellow;
                }
            }
            else
            {
                for (int i = c[1]; i <= c[3]; i += 10)
                {
                    pictureBox3.Location = new Point(c[2], i);
                    Thread.Sleep(100);
                    pictureBox3.Visible = true;
                    pictureBox3.BackColor = Color.Yellow;
                }
            }
        }
    }
}

////////////////////////////////////
if (f==0)
{
    for(int i=c[0];i<=685;i+=10)
    {
        pictureBox3.Location = new Point(i,c[3]);
        Thread.Sleep(100);
        pictureBox3.Visible = true;
        pictureBox3.BackColor = Color.Yellow;
    }
}
else
{
    for (int i = c[2]; i <= 685; i += 10)
    {
        pictureBox3.Location = new Point(i, c[3]);
        Thread.Sleep(100);
        pictureBox3.Visible = true;
        pictureBox3.BackColor = Color.Yellow;
    }
}
for (int i = c[3]; i <= 280; i += 10)
{
    pictureBox3.Location = new Point(685,i);
    Thread.Sleep(100);
    pictureBox3.Visible = true;
    pictureBox3.BackColor = Color.Yellow;
}

////////////////////////////////////
timer1.Enabled = false; //關閉timer
button4.Enabled = false; //關閉button
MessageBox.Show("執行完成", "通知", MessageBoxButtons.OK, MessageBoxIcon.Information); //顯示完成
System.Windows.Forms.Application.Exit(); //退出form
```

```

}
public static DialogResult InputBox(string title, string promptText, ref string value)
{
    Form form = new Form();
    Label label = new Label();
    TextBox textBox = new TextBox();
    Button buttonOk = new Button();
    Button buttonCancel = new Button();
    ///////////////////////////////////////////////////
    form.Text = title;
    label.Text = promptText;
    textBox.Text = value;
    ///////////////////////////////////////////////////
    buttonOk.Text = "OK";
    buttonCancel.Text = "Cancel";
    buttonOk.DialogResult = DialogResult.OK;
    buttonCancel.DialogResult = DialogResult.Cancel;
    ///////////////////////////////////////////////////
    label.SetBounds(9, 20, 372, 13);
    textBox.SetBounds(12, 36, 372, 20);
    buttonOk.SetBounds(228, 72, 75, 23);
    buttonCancel.SetBounds(309, 72, 75, 23);
    ///////////////////////////////////////////////////
    label.AutoSize = true;
    textBox.Anchor = textBox.Anchor | AnchorStyles.Right;
    buttonOk.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;
    buttonCancel.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;
    ///////////////////////////////////////////////////
    form.ClientSize = new Size(396, 107);
    form.Controls.AddRange(new Control[] { label, textBox, buttonOk, buttonCancel });
    form.ClientSize = new Size(Math.Max(300, label.Right + 10), form.ClientSize.Height);
    form.FormBorderStyle = FormBorderStyle.FixedDialog;
    form.StartPosition = FormStartPosition.CenterScreen;
    form.MinimizeBox = false;

```

```

    form.MaximizeBox = false;
    form.AcceptButton = buttonOk;
    form.CancelButton = buttonCancel;
    ///////////////////////////////////////////////////
    DialogResult dialogResult = form.ShowDialog();
    value = textBox.Text;
    return dialogResult;
}
private void button5_Click(object sender, EventArgs e)
{
    string value1="";
    if(InputBox("輸入X", "輸入黑板的寬:", ref value1) == DialogResult.OK);
    x1 = Convert.ToInt32(value1);
    ///////////////////////////////////////////////////
    string value2 = "";
    if (InputBox("輸入Y", "輸入黑板的長:", ref value2) == DialogResult.OK);
    y1 = Convert.ToInt32(value2);
    ///////////////////////////////////////////////////
    MessageBox.Show("輸入完成");
    button5.Enabled = false; //關閉button
}
private void pictureBox1_Click(object sender, EventArgs e)
{
    //畫方塊
    a = a + 1;

    if(a==1)
    {
        x = Convert.ToString(System.Windows.Forms.Cursor.Position.X);
        y = Convert.ToString(System.Windows.Forms.Cursor.Position.Y);
        b[0, 0] = Convert.ToInt32(x)-404;
        b[0, 1] = Convert.ToInt32(y)-224;
    }
    else

```

```

#include <SoftwareSerial.h>
SoftwareSerial BT(6,7); // 接收腳, 傳送腳
#define pwm 200
#define pwn 230
void setup()
{
    Serial.begin(9600);
    BT.begin(9600);
    //馬達一
    pinMode(10,OUTPUT); // M1Pin1
    pinMode(9,OUTPUT); // M1Pin2
    pinMode(11,OUTPUT); // E1Pin
    //馬達二
    pinMode(5,OUTPUT); // M2Pin1
    pinMode(4,OUTPUT); // M2Pin2
    pinMode(3,OUTPUT); // E2Pin
    motorstop();
}
char c;
void loop()
{
    c = BT.read();
    switch (c) {
        case 'u': //Forward
            UP();
            break;
    }
}

```



```

    case 'l': //Turn Left
        Left();
        break;
    case 'r': //Turn Right
        Right();
        break;
    default:
        Serial.println("no");
        break;
}
}

void UP()
{
    analogWrite(11, pwn);
    analogWrite(3, pwn);
    digitalWrite(10, HIGH);=
    digitalWrite(9, LOW);
    digitalWrite(5, LOW);
    digitalWrite(4, HIGH);
    delay(850);
    motorstop();
}

void Right()
{
    analogWrite(11, pwn);
    analogWrite(3, pwn);
    digitalWrite(10, HIGH);

```

表 3 材料

一、Arduino

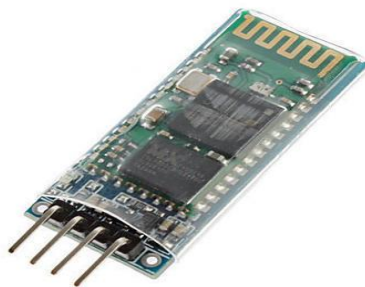
Arduino 是 ATmega328P 的微控制器開發板。有 14 個數字輸出入腳(其中 6 個可用作 PWM 輸出)。透過 USB 連接供電為 5V，透過電源輸入插座供電最多為 12V，通常不建議低於 7V，可能會有供電不足的問題。



(圖二) Arduino UNO

二、藍芽模組 HC06

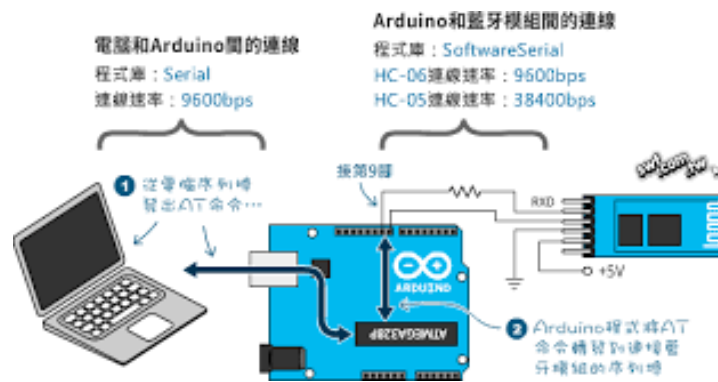
HC-06，採用 BC417143 晶片，支援藍牙 2.1+EDR 規範，有 4 個接角分別是 VCC、GND、TXD、RXD（如圖四）。所支持的波特率有：4800、9600、19200、38400、57600、115200、23400、460800、921600、1382400。傳輸的距離為 10M。



(圖三) 藍芽模組 HC06

(表一) HC-06 接角對應表

接角	功能	接上 UNO
VCC	電源	VCC
GND	接地	GND
TXD	傳送	D8
RXD	接收	D9



(圖四) 連接示意圖

藍芽的運作原理是在 2.45 GHz 的頻帶上傳輸，可以傳輸數位資料和聲音。每個藍芽裝置都有 IEEE 802 標準所制定的 48-bit 地址。傳輸範圍最遠在 10 公尺。藍芽技術傳輸量大，每秒鐘可達 1MB，可以設定加密保護，每分鐘變換的頻率約一千六百次，因而不好截收。

三、舵型馬達 MG996R

舵機是種位置伺服的驅動器，目前在模型，遙控機器人中使用較為普遍。

控制信號由接收機的通道進入信號晶片，獲得直流偏壓。內部有基準電路，產生週期 20ms、寬度 1.5ms 的基準信號，將獲得的直流偏置電壓和電位器做電壓比較，獲得電壓差輸出。電壓差的正負輸出到驅動晶片決定電機的轉向。當電機轉速一定時，通過減速齒輪帶動電位器旋轉，使電壓差為 0。



(圖五) 舵型馬達 MG996R

四、Visual C#

為物件導向的高階程式語言，為 .NET 眾多程式語言之一。其語法與 Java, Javascript, C++ 相似。C# 由 C/C++ 延伸而來，以 .NET Framework 作為根基。C# 可以開發各種程式，是種通用語言，從複雜大型的作業系統到小型嵌入式系統都可做使用。



(圖六) Visual C

五、PHP

PHP 是種開源的電腦指令碼語言，適用於網路的開發。PHP 的語法，像是 C 語言、Java、Perl。PHP 的應用範圍相當廣泛。一般來說 PHP 大多執行在網頁伺服器上，透過執行來產生使用者瀏覽的網頁。PHP 可以在大多多的伺服器、作業系統上去做執行。



(圖七) PHP

伍、研究結果

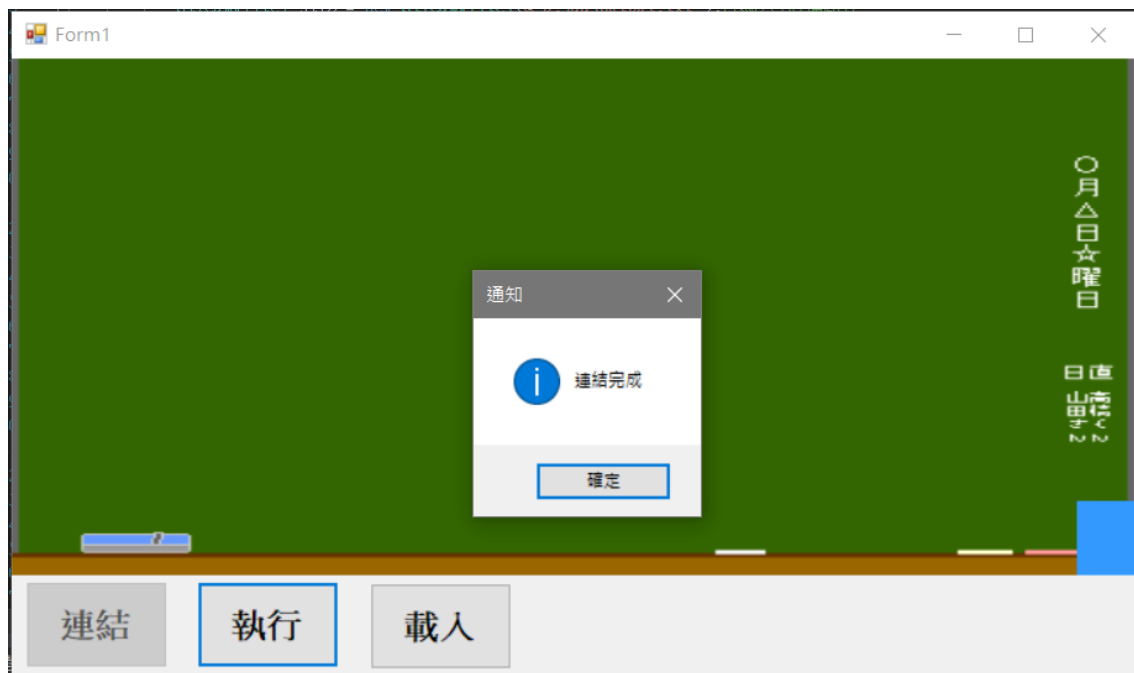
伍、研究結果

手機取得照片，照片利用網路的方式傳到電腦。電腦讀取完，按下連接，載入照片到 PictureBox，並且電腦藍芽連接 Arduino。在 PictureBox 上點擊兩下，畫出需要擦拭的正方形。此時按下執行，電腦便會開始計算，並且同時傳到 Arduino。

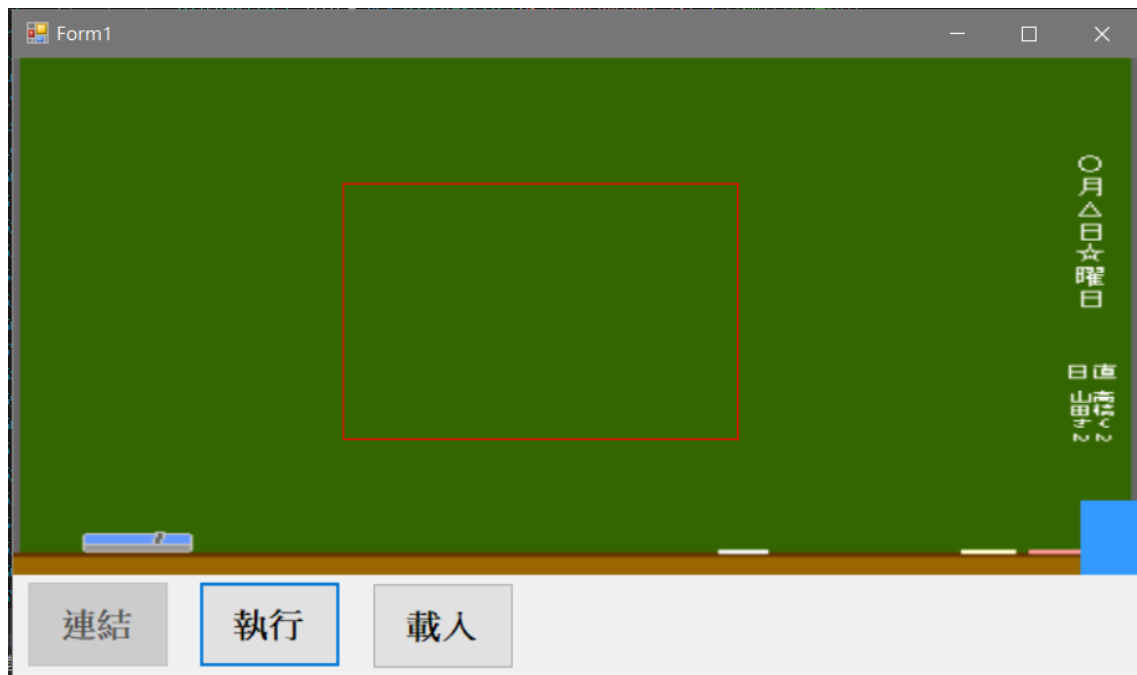
Arduino 便利用讀取到的值，來控制陀型馬達 MG996R，來完成擦拭的動作。載入的按鈕，是因為每個黑板的長寬不盡相同，便可輸入長和寬，來避免運算錯誤的問題。



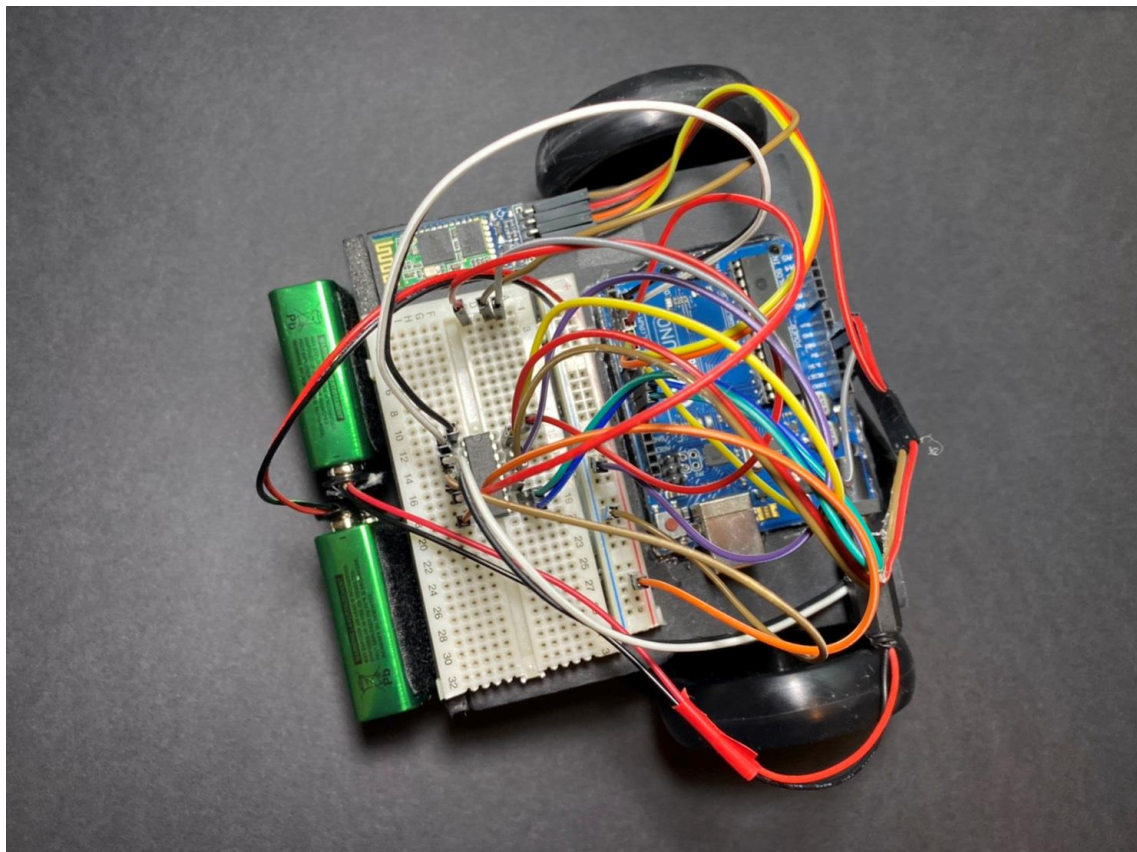
(圖八) C#視窗



(圖九) 載入照片



(圖十) 點擊 PictureBox 兩下 畫出正方形



(圖十一) 按下執行 車子移動

陸、討論

一、 如何去控制直流馬達

如果用 UNO 控制，會使得馬達變成類比，不能去控制連續的開關，這樣會很像伺服馬達，然而這不是我們需要的。因此我們使用了 L293D 這顆 IC，他能控制馬達變成類比。不在侷限在有限的角度控制。

二、編寫程式碼 遇到問題

因為我們用到了跨語言的傳輸和資料庫的讀取，所以不斷上網尋找和摸索，花費了不少的時間和心思。最後發現 C#有 serialport，可以做傳輸上的橋樑，也能幫助 UNO 和 C#的藍芽控制。

三、無法在黑板上移動

黑板後面有鋼板，能夠使得磁鐵吸附在黑板上。因此我們在車子的下部，抓上了三顆，來讓車子吸附在黑板上。磁鐵磁力太強會使得車子無法移動，因為磁力大於了馬力。而磁力太小則會讓車子從黑板上掉落。因此我們換了很多磁鐵，來找尋最適合的磁鐵。

四、HC-06 無法連接藍芽

我們發現無法傳輸資料到 arduino，所以我們檢查了線路，發現線路輸入腳和輸出腳接反了，導致資料無法傳輸。換了正確的線路後，還是無法傳輸，於是我們上網尋找答案，發現鮑率會影響輸出解果，因此我們把鮑率調成 9600bps 才成功傳輸。

五、IC 供電不足

我們本來利用 arduino 上的 5V，去供電給 IC，我們發現 5V，無法讓馬達座移動，因為電壓太小，因此我們外接了 9V，才成功讓馬達順利移動

柒、結論

在這次的專題中，我們用了三年所學到的知識和經驗。不再侷限書上的文字，而是利用自己的想法和所學去實作。或許會遇到很多問題，但慢慢地去嘗試，便能找到答案。甚至我們還自學了 C#、PHP 等等的程式語言，這個專題的完成使我們感到許許多多的成就感。

在未來希望能將此專題加入粉塵吸取裝置，來減少粉塵接觸到人體的機會。例如老師每天都會使用到黑板，因此多多少少會吸許到粉筆灰，而這些污染物也會對呼吸道有一些問題，

捌、參考資料及其他

一、HC-06 的設定

<http://gsyan888.blogspot.com/2014/03/arduino-hc-06-at-command.html>

二、L293D 的馬達控制

<https://atceiling.blogspot.com/2019/08/arduino55l293d.html>

三、C#連接 arduino

<https://zhung.com.tw/article/%E5%8F%96%E5%BE%97%E6%89%80%E6%9C%89%E9%80%A3%E6%8E%A5%E5%88%B0%E9%9B%BB%E8%85%A6%E4%B8%8A%E7%9A%84serial-port%E8%A9%B3%E7%B4%B0%E8%B3%87%E8%A8%8A/>