

# 第一章 研究動機與目的

## 1-1 研究動機

近年來，由於工業科技和電子科技越來越發達，很多東西都已經機械化，像是震災或者發生災害時，有些高科技國家，其實已經在研究一些機器人，如何使那些機器人去代替人類到危險的地方先進行救災的動作，就算遇到不測，也只是機器壞損而已。

所以，本文想要靠著自走車搭配 Arduino 程式還有視訊鏡頭，去做到遠端遙控來去探測危險的地方可能有什麼危險存在，進而去由機器發現，然後讓人類想辦法解決，萬一小車遭遇危險也比較不會有生命安全的顧慮。

希望能夠藉著逐一修改 Arduino 程式、簡易而人性化的控制面板、照射角度寬廣的視訊鏡頭以及零件配置整齊的自走車，達到所想要的結果，讓期許不再是期許而是實現。

## 1-2 研究目的

由於科技的發達，現在走進自動化的趨勢，什麼東西都講究快速，便利，且耗損能源低，漸漸地，人們用電腦來取代人力，因此我們便針對此來設定我們的專題題目—自走車。我們利用在學校所學的電子學、電路學，以及伺服控制、Arduino 程式，作為我們設計自走車的基礎。利用電源電路、馬達、外加攝影鏡頭，以及 Arduino 程式來製作自走車。

我們會選擇製作自走車的原因是因為看到許多的國家都有不斷的研究與發表自走機器人例如：美國 NASA 於 1997 年登陸火星探測車旅居者號。覺得將來不論工業與家庭未來一定會以機器人來取代現有的人力，往後工廠會使用機器人來工作使商品消耗與損失降低以及工作效率的提升，而家庭則是利用機器人做家事等等；雖然我們沒有良好的設備及物資去研究自走機器人，但是我們想從研究自走車來更了解機器人的發展，所以選擇了自走車作為本次專題的題目。

## 第二章 自走車主要材料及元件介紹

### 2-1 Arduino 開發板介紹

Arduino 是源自義大利的一個開放源程式碼的硬體專案平台，該平台包括一塊具備簡單 I/O 功能的電路板以及一套程式開發環境軟體，可以用來開發交互產品。

Arduino 也可以開發出與 PC 相連的周邊裝置，能在運行時與 PC 上的軟體進行通信。Arduino 的硬體電路板可以自行焊接組裝，也可以購買已經組裝好的模組，而程式開發環境的軟體則可以從網上免費下載與使用。

因為 Arduino 是一塊基於開放原始碼的 Simple I/O 版，並且具有使用類似 Java、C 語言的開發環境。可以快速使用 Arduino 語言與 Flash 或 Processing 等軟體作出互動作品。

## 2-2 自走車材料表

自走車所需材料如下：

表 2-1 自走車材料表

	名稱	數量
1	小車固定底盤	1
2	小車輪胎	2
3	輔助輪	1
4	馬達	2
5	馬達固定架	2
6	螺絲	4 包
7	連接線	12
8	WIFI 盒	1
9	18650 電池	4
10	USB 下載線	1
11	雲台	1
12	資料光碟片	1
13	小車控制主板	1
14	攝影鏡頭	1
15	9V/2A 電源	1
16	18650 電池盒	1
17	USB 數據線	1



圖 2-1 車體模板

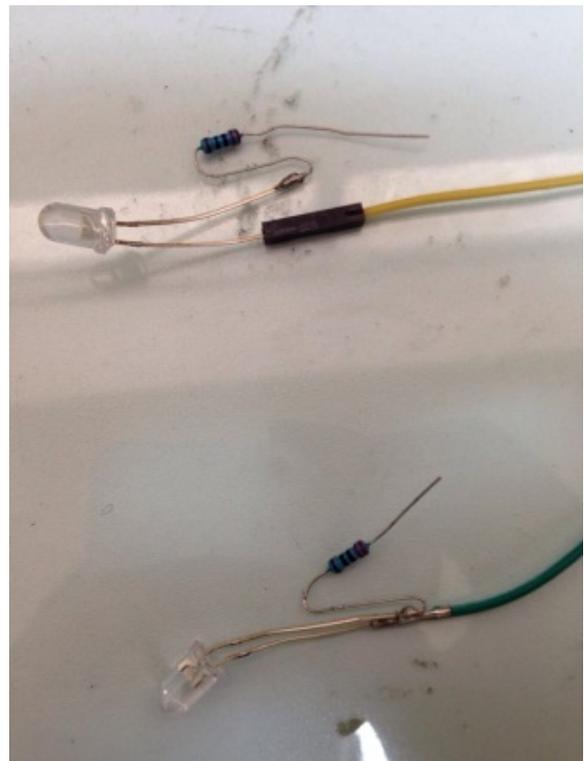


圖 2-2 LED 燈



圖 2-3 車頭燈罩



圖 2-4 WiFi 接收盒

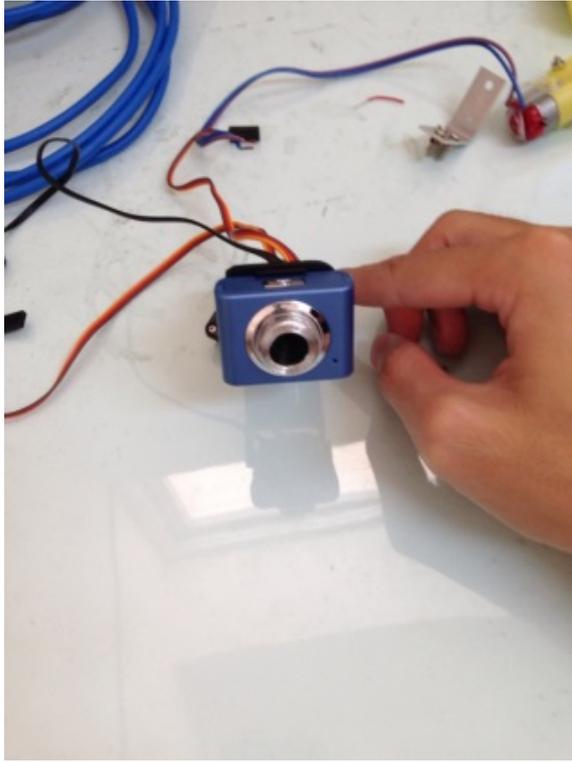


圖 2-5 視訊鏡頭

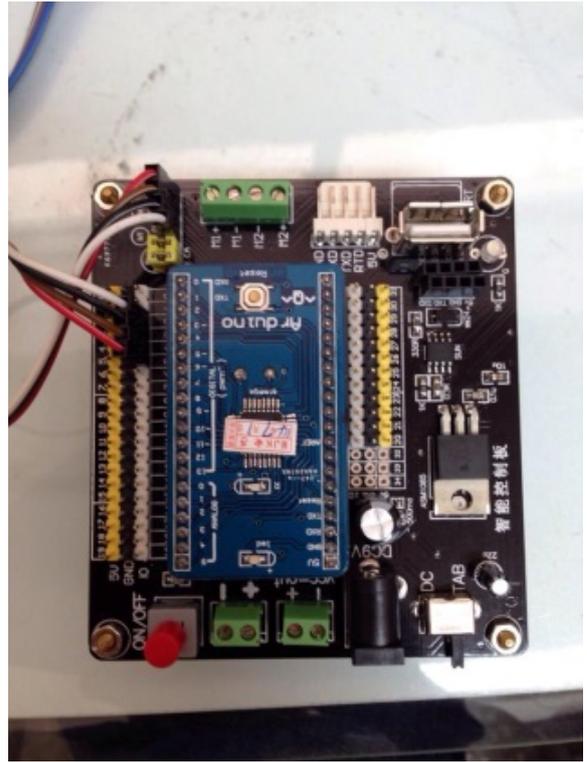


圖 2-6 程式燒錄模組



圖 2-7 輪胎

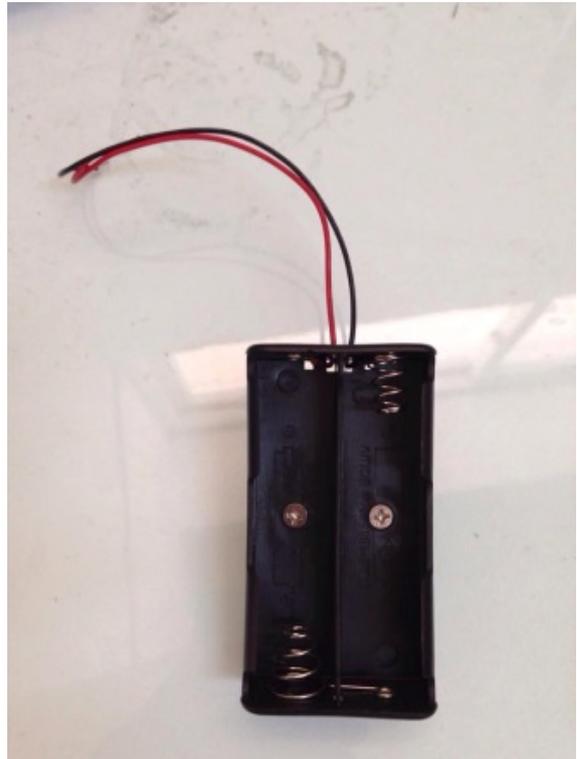


圖 2-8 電池盒



圖 2-9 輔助輪軸

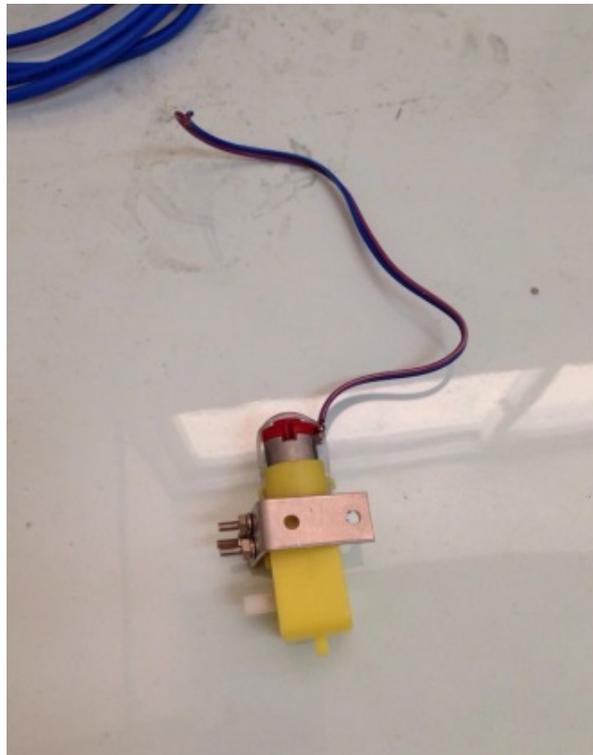


圖 2-10 馬達



圖 2-11 鑽孔機

## 第三章 自走車組裝及測試運作

### 3-1 自走車零件組裝步驟

圖 3-1 為車體木板，因為買來時上面是完全乾淨的一塊薄木板，所以要先將所有零件位置做設計然後把位置圖大概佈置出來，如圖 3-2，再來將所有螺絲鎖點進行鑽孔，如圖 3-3，鑽孔完之後就大概完成了車體木板的鑽孔設計了，如圖 3-4。



圖 3-1 車體模板



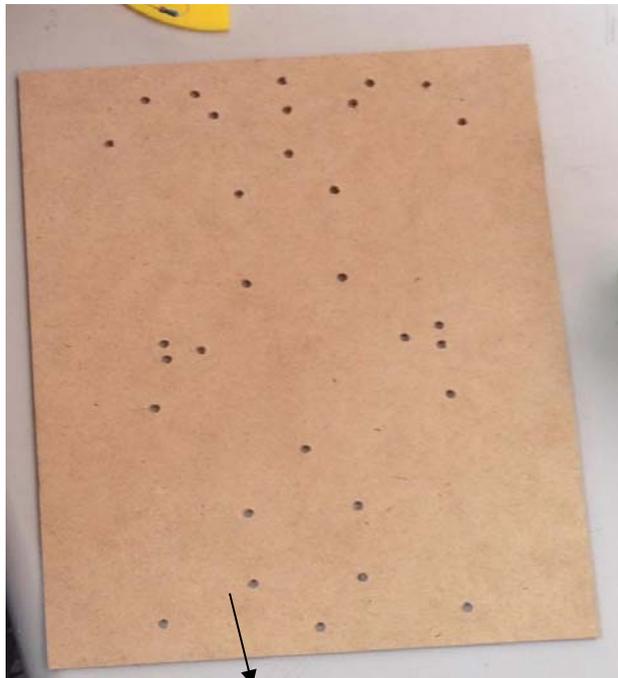
鉛筆畫線規劃的預設位置

圖 3-2 已規劃位置車體模板



正在鑽孔中的樣子

圖 3-3 鑽孔中車體模板



已鑽孔完成品

圖 3-4 已完成鑽孔的車體模板

把鑽孔好的車體模板依「已規畫好」的位置依序裝上、LED 燈、LED 車頭燈燈罩、WIFI 接收器、視訊鏡頭、程式燒錄模組、電池盒、輔助輪，如圖 3-5~圖 3-8，如果沒有按照位置將鎖點鎖上可能會出現卡線造成零件無法依預期使用的情況，當東西都安裝好之後，離小車完成品就不遠了，如圖 3-9。

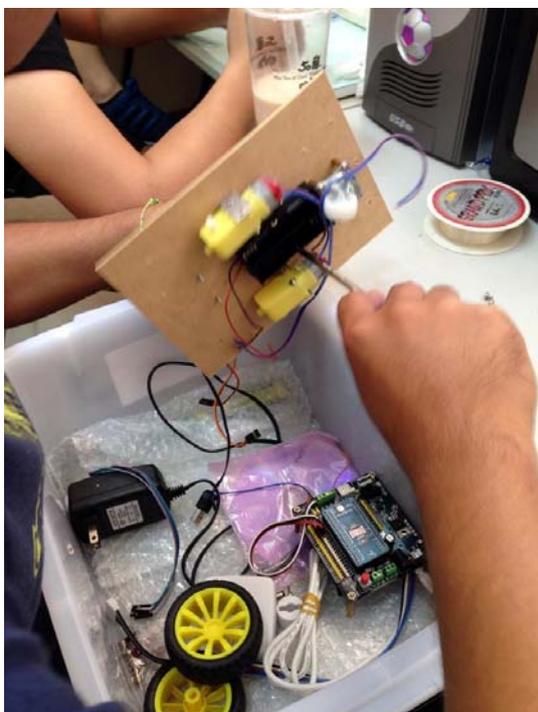


圖 3-5 裝上馬達及電池盒

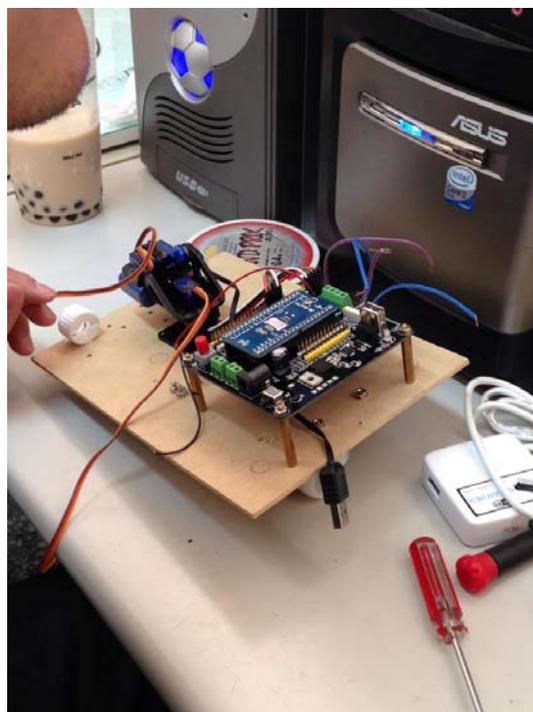


圖 3-6 裝上視訊鏡頭

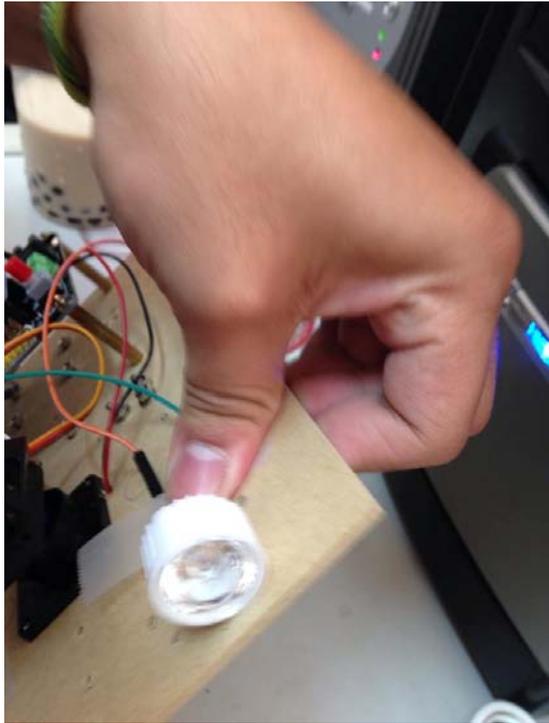


圖 3-7 裝上車頭燈

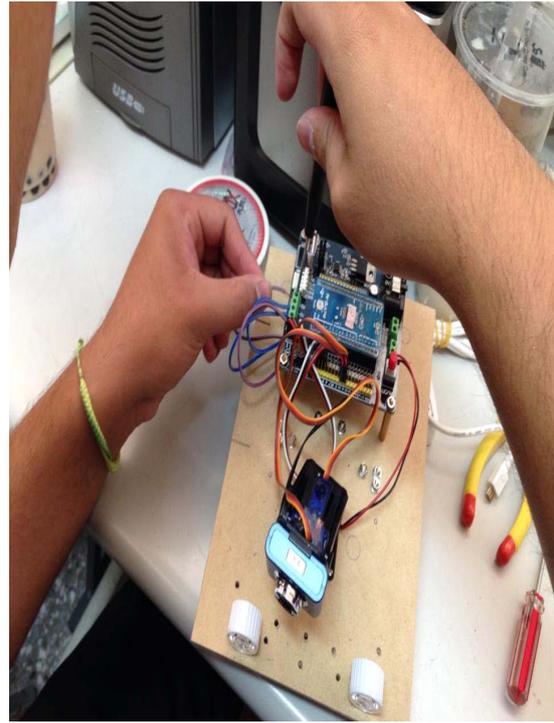


圖 3-8 裝上程式燒錄模組

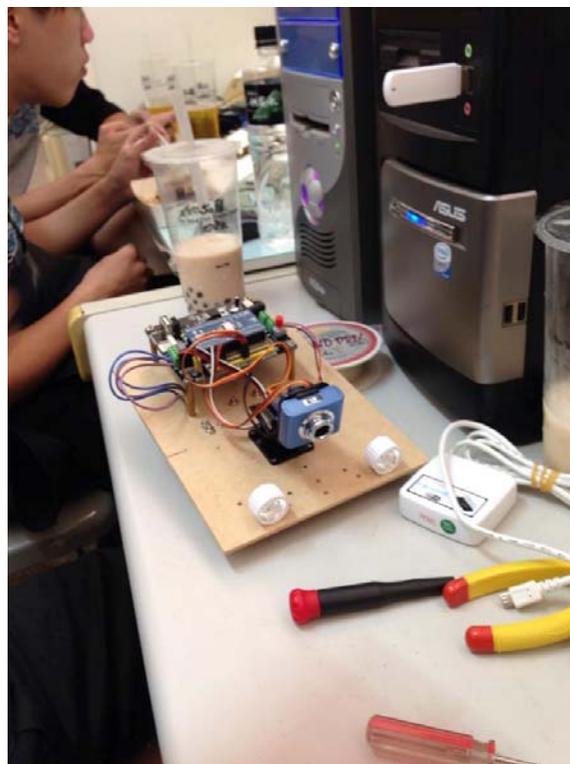


圖 3-9 組裝完的完成圖 ↑

當小車外型進度已經完成到圖 3-9，就可以開始進行輪胎位置擺放的規劃，如圖 3-10。

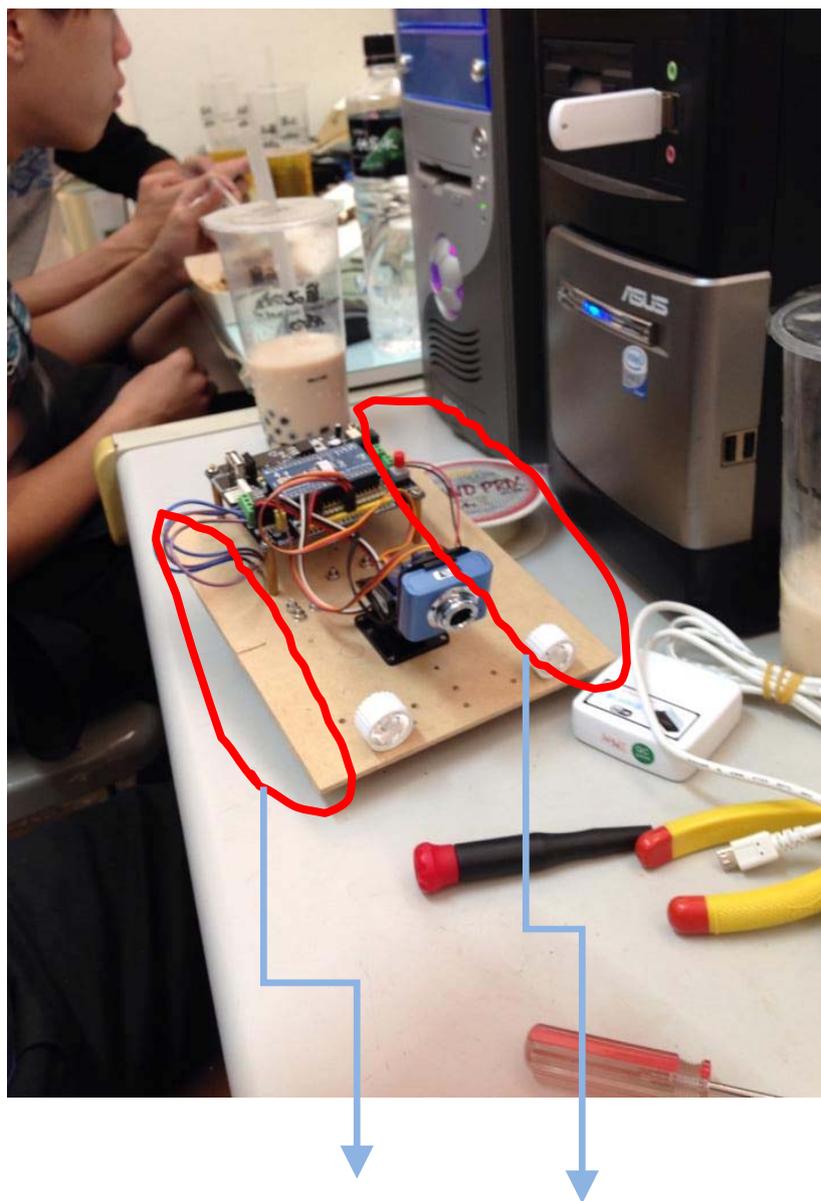
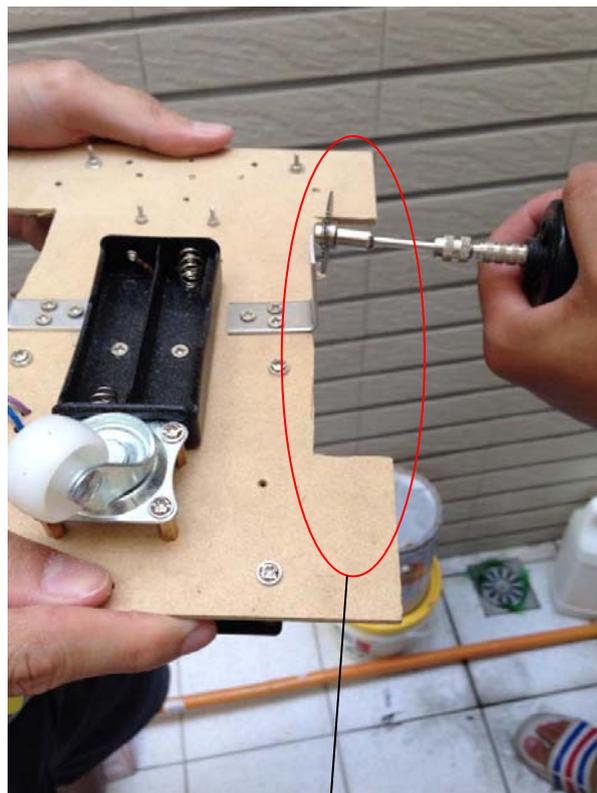


圖 3-10 紅線部分為輪胎位置規劃的兩側

規畫好之後就可以開始準備切割木板，如圖 3-11，使輪胎可以安裝在木板兩側，完成小車的動力裝置。



紅線部分為量測完輪胎距離後切割完成的部分

圖 3-11

之後依序安裝上馬達跟輪胎，就完成囉，如圖 12~圖 14

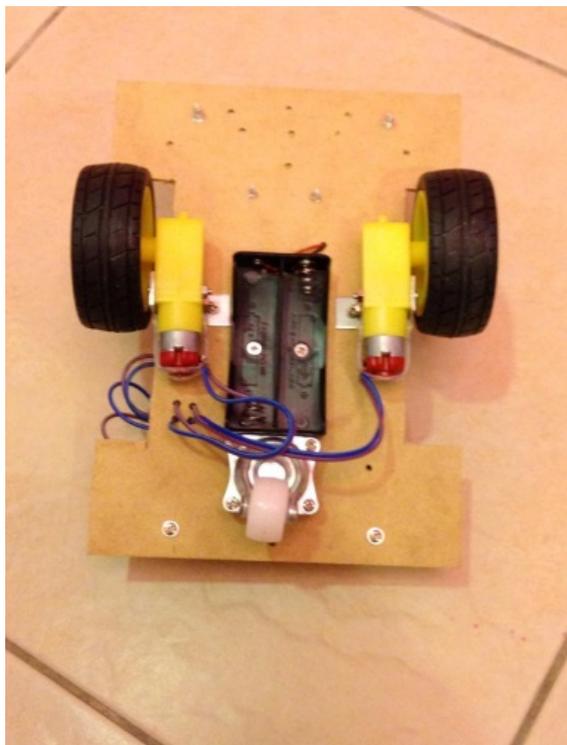


圖 3-12 背面完成圖

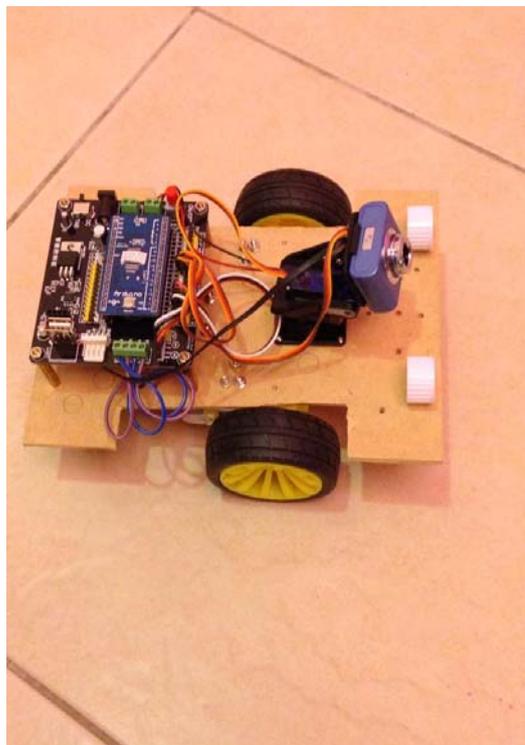


圖 3-13 側面完成圖

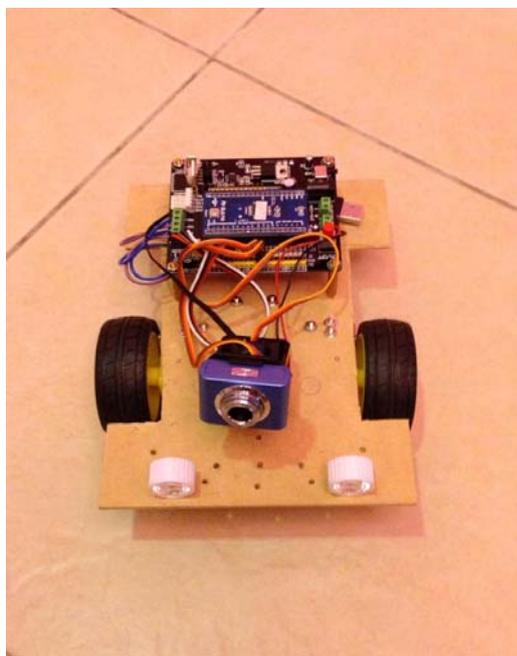


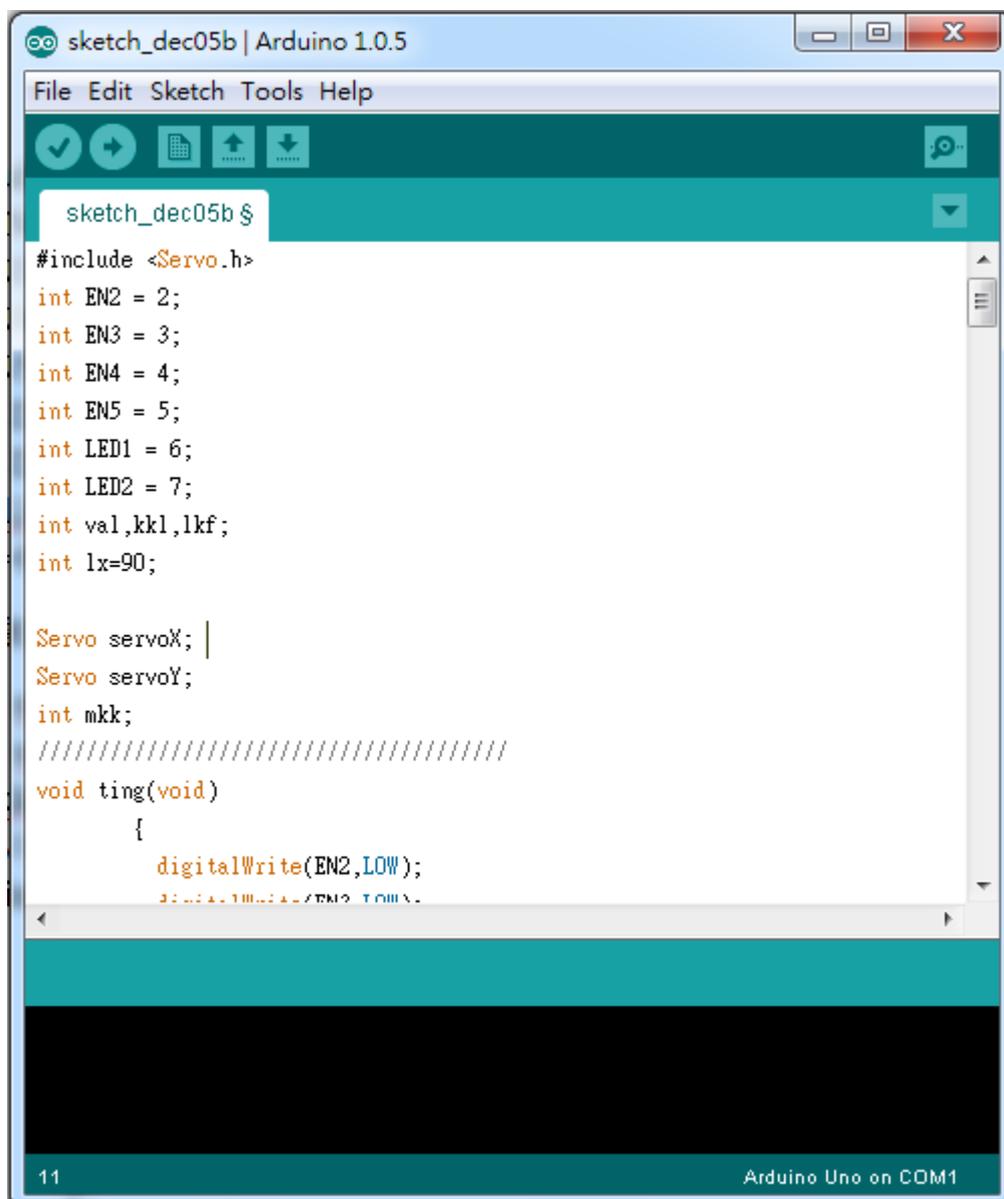
圖 3-14 正面成品圖

## 第三章 自走車組裝及測試運作

### 3-2 程式碼測試及運作

#### 3-2-1 Arduino 燒錄軟體

這個程式是用於把 Arduino 程式碼燒入晶片裡，並做適當的程式修改及偵錯。



```
sketch_dec05b | Arduino 1.0.5
File Edit Sketch Tools Help
sketch_dec05b $
#include <Servo.h>
int EN2 = 2;
int EN3 = 3;
int EN4 = 4;
int EN5 = 5;
int LED1 = 6;
int LED2 = 7;
int val,kkl,lkf;
int lx=90;

Servo servoX; |
Servo servoY;
int mkk;
////////////////////////////////////
void ting(void)
{
    digitalWrite(EN2,LOW);
    digitalWrite(EN3,LOW);
    digitalWrite(EN4,LOW);
    digitalWrite(EN5,LOW);
}

11 Arduino Uno on COM1
```

圖 3-2-1 Arduino 燒錄軟體

### 3-2-2 Arduino 程式碼

```
#include <Servo.h>

int EN2 = 2;

int EN3 = 3;

int EN4 = 4;

int EN5 = 5;

int LED1 = 6;

int LED2 = 7;

int val, kkl, lkf;

int lx=90;

Servo servoX;

Servo servoY;

int mkk;

////////////////////////////////////

void ting(void)

{

    digitalWrite(EN2, LOW);
```

```
    digitalWrite(EN3, LOW);  
    digitalWrite(EN4, LOW);  
    digitalWrite(EN5, LOW);  
}
```

```
void qian(void)
```

```
{  
    digitalWrite(EN2, LOW);  
    digitalWrite(EN3, HIGH);  
    digitalWrite(EN4, LOW);  
    digitalWrite(EN5, HIGH);  
}
```

```
void hou(void)
```

```
{  
    digitalWrite(EN2, HIGH);  
    digitalWrite(EN3, LOW);  
    digitalWrite(EN4, HIGH);  
    digitalWrite(EN5, LOW);  
}
```

```
void zuo(void)
{
    digitalWrite(EN2, LOW);
    digitalWrite(EN3, HIGH);
    digitalWrite(EN4, HIGH);
    digitalWrite(EN5, LOW);
}
```

```
void you(void)
{
    digitalWrite(EN2, HIGH);
    digitalWrite(EN3, LOW);
    digitalWrite(EN4, LOW);
    digitalWrite(EN5, HIGH);
}
```

```
void servo_left()
{
    int servotemp=servoX.read();
    servotemp-=2;
```

```
if(servotemp<170&&servotemp>10)

servoX.write(servotemp);

else if (servotemp<=10) servoX.write(10);

else servoX.write(170);

}
```

```
void servo_right()

{

int servotemp=servoX.read();

servotemp+=2;

if(servotemp<170&&servotemp>10)

servoX.write(servotemp);

else if (servotemp<=10) servoX.write(10);

else servoX.write(170);

}
```

```
void servo_up()

{

int servotempl=servoY.read();
```

```

servotemp1+=2;

if(servotemp1<170&&servotemp1>10)

servoY.write(servotemp1);

else if (servotemp1<=10) servoY.write(10);

else servoY.write(170);

}

void servo_down()

{

int servotemp1=servoY.read();

servotemp1-=2;

if(servotemp1<170&&servotemp1>10)

servoY.write(servotemp1);

else if (servotemp1<=10) servoY.write(10);

else servoY.write(170);

}

void setup()

{

servoX.attach(10);

servoY.attach(11);

```

```
servoX.write(90);

servoY.write(90);

pinMode(EN2, OUTPUT);

pinMode(EN3, OUTPUT);

pinMode(EN4, OUTPUT);

pinMode(EN5, OUTPUT);

pinMode(LED1, OUTPUT);

pinMode(LED2, OUTPUT);

  for(kk1=0;kk1<50;kk1++)
  {

    digitalWrite(LED1, HIGH);

    digitalWrite(LED2, HIGH);

    delay(500);

    digitalWrite(LED1, LOW);

    digitalWrite(LED2, LOW);

    delay(500);

  }

Serial.begin(9600);

lkf=0;
```

```
}  
  
void loop()  
  
{  
  
  if(Serial.available())  
  
  {  
  
    lkf = Serial.read();  
  
    switch(lkf)  
  
    {  
  
    case 'a' :  
  
      qian();  
  
      servoX.write(90);  
  
      lkf=0;  
  
      break;  
  
    case 'b' :  
  
      hou();  
  
      lkf=0;  
  
      break;  
  
    case 'c' :  
  
      zuo();
```

```
    lkf=0;

    break;

case 'd' :

    you();

    lkf=0;

    break;

case 'e' :

    ting();

    lkf=0;

    break;

case 'j' :

    servo_left();

    lkf=0;

    break;

case 'l' :

    servo_right();

    lkf=0;

    break;

case 'k' :
```

```
servo_up();

lkf=0;

break;

case 'i' :

    servo_down();

    lkf=0;

    break;

case 'n' :

    digitalWrite(LED1, HIGH);

    digitalWrite(LED2, HIGH);

    lkf=0;

    break;

case 'm' :

    digitalWrite(LED1, LOW);

    digitalWrite(LED2, LOW);

    lkf=0;

    break;

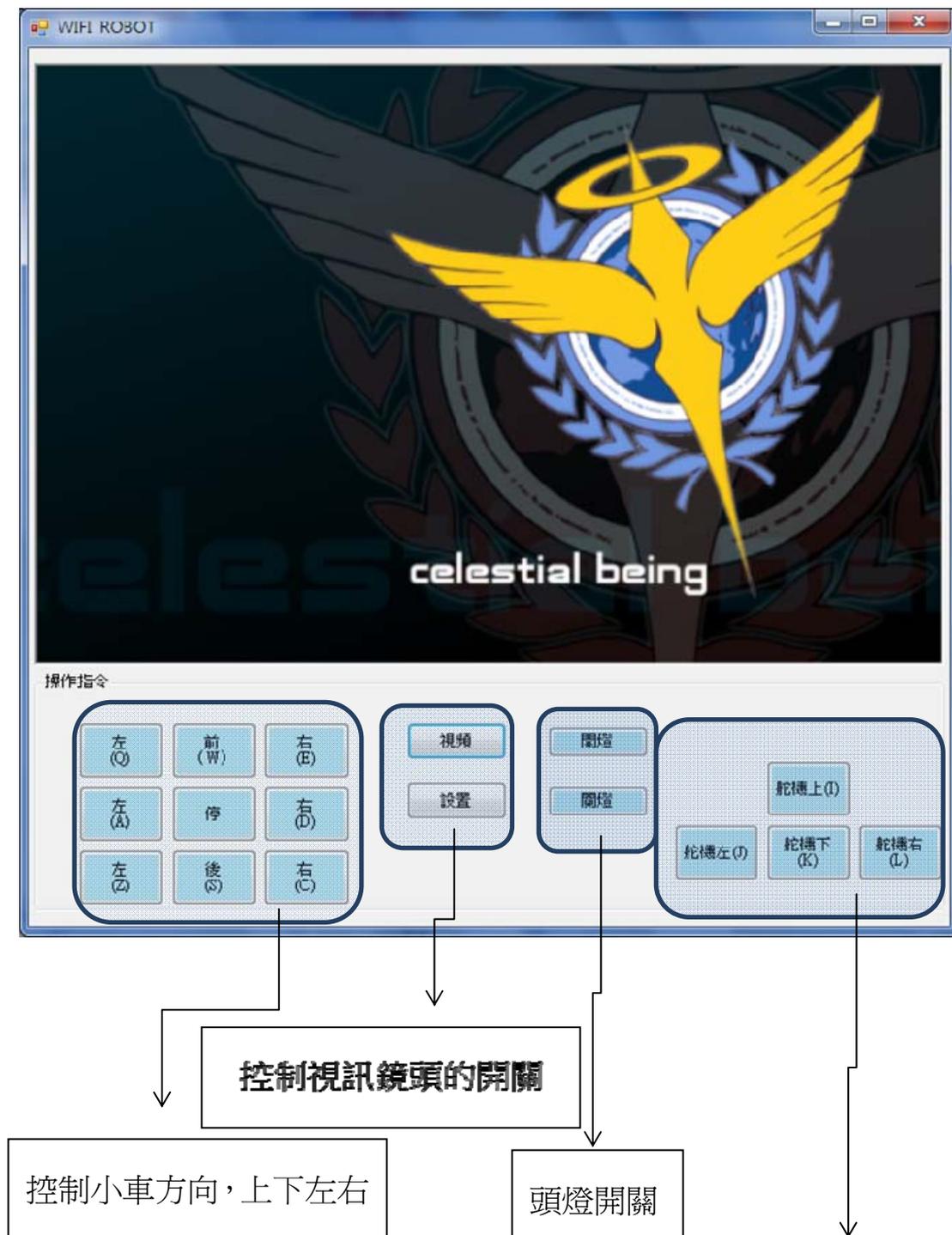
}

}
```

}

### 3-2-3 連接電腦的小車控制面板

這是用 Visual Studio 2010，來進行編寫及介面設計。



電腦控制程式碼

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace WifiVideo

{

    static class Program

    {

        /// <summary>

        /// </summary>

        [STAThread]

        static void Main()

        {

            Application.EnableVisualStyles();

            Application.SetCompatibleTextRenderingDefault(false);

            Application.Run(new Form1());

        }

    }

}
```

}

}

## 第四章 結論

### 4-1 結論與心得

這次專題自走車讓我們學習到很多以前沒有學習到的東西，我們會選擇製作自走車的原因是因為看到許多的國家都有不斷的研究與發表自走機器人，覺得將來不論工業與家庭未來一定會以機器人來取代現有的人力，往後工廠會使用機器人來工作使商品消耗與損失降低以及工作效率的提升，而家庭則是利用機器人做家事等等；雖然我們沒有良好的設備及物資去研究自走機器人，但是我們想從研究自走車來更了解機器人的發展，所以選擇了自走車作為本次專題的題目。

## 參考文獻

- [1] 益眾科技 <http://www.icci.com.tw/web/Home?FP=1064/>
- [2] 自走車網站 <http://s07368.myweb.hinet.net/>
- [3] 自走車專題製作範例  
<http://povc.org/wiki/zi-zou-che-zhuan-ti-%E8%A3%BDzuo-%E7%AF%84li.htm/>
- [4] 專題製作工坊 <http://yawlin.myweb.hinet.net/>
- [5] 自走車所需材料  
[http://www.hyivs.tnc.edu.tw/pic/line\\_tracer/line\\_tracer.html/](http://www.hyivs.tnc.edu.tw/pic/line_tracer/line_tracer.html/)
- [6] GACHIA 循跡自走車競賽  
<http://www.makefaire.com.tw/gachia-24490363213325836208365543147836093.html/>
- [7] 作者：趙英傑，超圖解 Arduino 互動設計入門，出版社：旗標，  
出版日期：2013/04/03。
- [8] 作者：Arduino 原創者 Massimo Banzi，踏進互動科技世界 -  
使用 Arduino，台灣 arduino.tw 林義翔譯，2009。

- [9] 作者：Simon Monk，書名：給邪惡天才的 30 個 Arduino 專題，  
出版社：馥林文化，譯者：謝瑩霖、蔡睿烝。
- [10] 書名：Arduino 輕鬆入門：範例分析與實作設計，出版社：博  
碩文化，出版日期：2014 年 2 月 19 日。
- [11] 作者：楊明豐，書名：Arduino 最佳入門與應用：打造互動設  
計輕鬆學，出版社：基峰資訊，出版日：2014/1/17。