



新北市立中和高級中學

New Taipei Municipal Zhonghe Senior High School

獎狀

1 年 18 班 吳振榮同學

參加 109 學年度中和高中校內科學展覽會

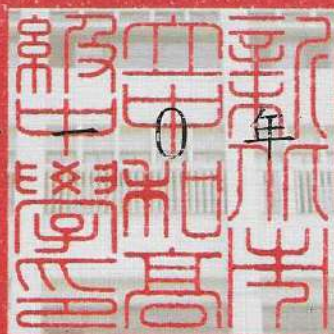
作品名稱：自製 Arduino 控制車，自動停車系統

榮獲 工程學科（一）組 優等獎

特頒獎狀 以資鼓勵

校長 柯雅菱

中華民國 一 〇 年 三 月 十 六 日



中華民國第 60 屆中小學科學展覽會

校內初賽作品說明書

科 別: 工程學(一)科

作品名稱: 自製 Arduino 控制車，自動停車系統

關 鍵 詞: 自動停車、超音波感測、藍芽控制車

班級: 118

作者: 吳振榮、林威廷

指導老師: 華光永

編號:

摘要

人們在進行停車時，常常無法拿捏好停車格的距離，本研究是將現實問題縮小，並進行小型模擬來實驗車子在哪個位置中需要使用多少輪差和在限定空間內進行安全且快速的停車，此方面由超音波感測器進行感測，並傳回到 Arduino UNO 開發版對前後兩台車的車距進行運算，進而做出停車判斷，若此距離可進行停車，則會自動控制車子本體移動到適當的位置進行停車，若無法進行停車，則會用其他方式進行停車。作品使用藍芽與手機進行連接，所以能進一步使用手機控制 Arduino 車子 (此部分為車子到達目的地時的操控)。

壹、研究動機

在日常生活中車子已成為人們的必備工具，但是在停車時是一件大考驗，常常因為沒拿捏好距離而東撞西撞，所以如果能讓車子計算停車的安全距離及角度，並讓車子自行停車，不僅能使車子安全的停入停車格，也可使駕駛省去頻頻觀看後照鏡來進行停車的時間。

貳、研究目的

- 一、學習自行組裝以 Arduino 為基礎的車子。
- 二、探討馬達之轉速及前後轉的狀況所影響車子的停車結果。
- 三、研究超音波感測器對距離的判斷以及運用超音波感測器計算停車格的車距。
- 四、研究運用藍芽如何讓 Arduino 開發版與手機進行訊息的傳送與接收。
- 五、研究基礎電路以及基本 Arduino 基礎語法，並運用基礎電學來控制馬達的正反轉及轉速。

參、研究設備及器材

名稱	規格	數量
Arduino 開發版	UNO	1
超音波感測器	HC-SR04	1
馬達	減速馬達	4
電池盒	鋰電池電池盒	1
電池	鋰電池	4
銅鋅電線	紅、黑	12
馬達驅動板	L298N	1
壓克力板	無	1
筆記型電腦	ASUS WIN10	1
Android 手機	HTC ONE Android 4.1	1
輪子	無	4
杜邦線	公對母	14
藍芽模組	HC-05	1
伺服馬達	SG-90	1
9V 電池	9V	1

肆、研究過程及方法

一、自行組裝 Arduino 車子

(一) 硬體設備配置以及大致功能介紹

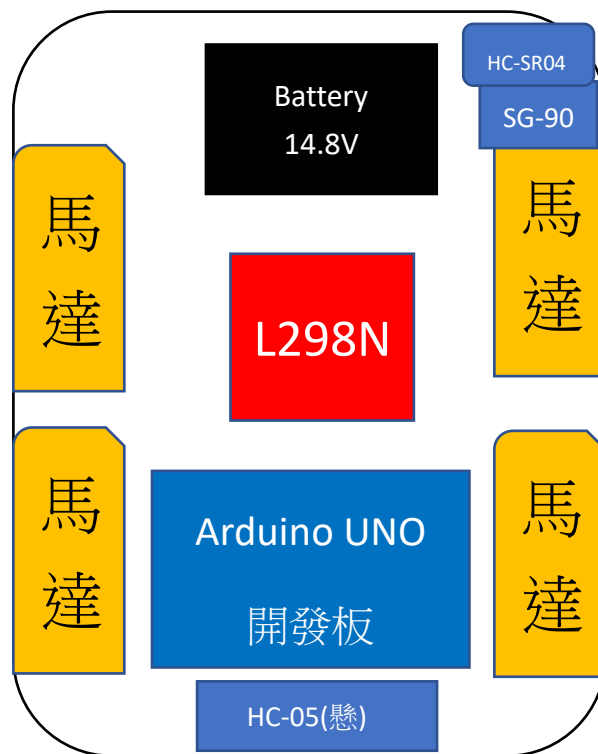


圖 1 俯視設計圖

圖 1 為預期車子硬體設備擺放位置，並以 Arduino UNO 為主要控制器，並傳輸穩定 5 伏特電壓給 HC-05 和 HC-SR04，以及對藍芽接收器做連接，不斷的向客戶端接收資料。當接收到自動停車指令時，便會使超音波感測器進行測量停車格長度，接著讓伺服馬達轉動，使超音波感測器能以其他角度來進行距離的判斷。L298N 馬達驅動模組會給予伺服馬達穩定的 5 伏特電壓，並給予馬達穩定電壓，馬達的正反轉是給予馬達高電壓以及低電壓，若要進行轉速的調整，則會讓 Arduino UNO 開發版傳輸 PWM 進行調速，若要使車子進行轉彎時，則會以左、右兩側的轉速的差別來進行轉彎。馬達驅動模組會接收外部給予的 14.8 伏特電壓，同時能輸出 5 伏特的電壓給伺服馬達。

二、電路及接線

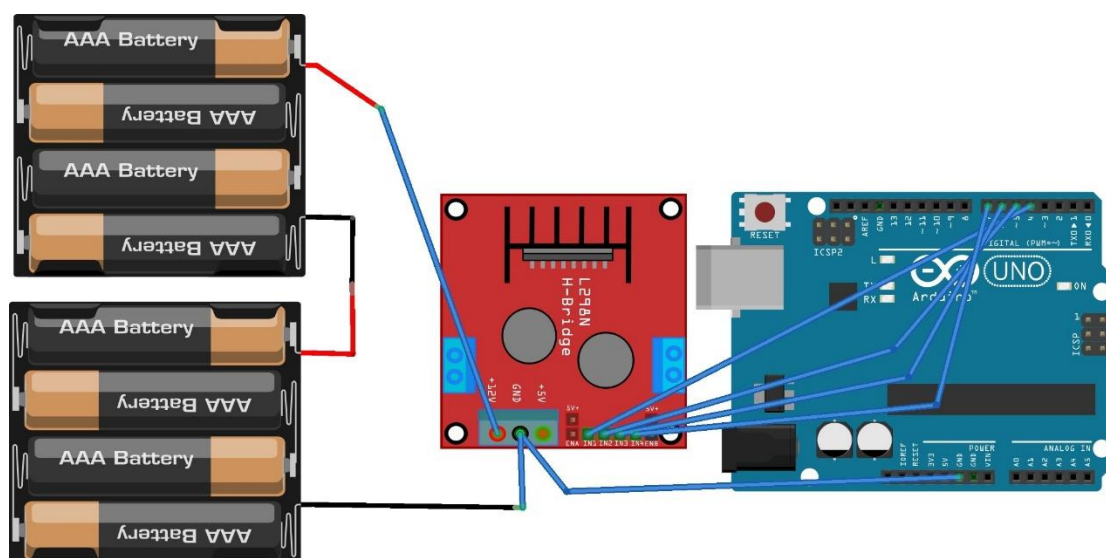


圖 2 L298N 與 Arduino UNO 接線

將 12 伏特的電壓(此處用 12V 來做代替)接入 L298N 並 GND 接回 Arduino UNO 的 GND，並將 L298N 的 In1、In2、In3、In4 接回 Arduino UNO 的 4、5、6、7 這四個孔，其中 5 和 6 接孔為 PWM 輸入，可以輸出類比(analog)訊號，將用來控制轉速，In1 和 In2 為控制左側兩顆馬達，In3 和 In4 為控制右側兩顆馬達。

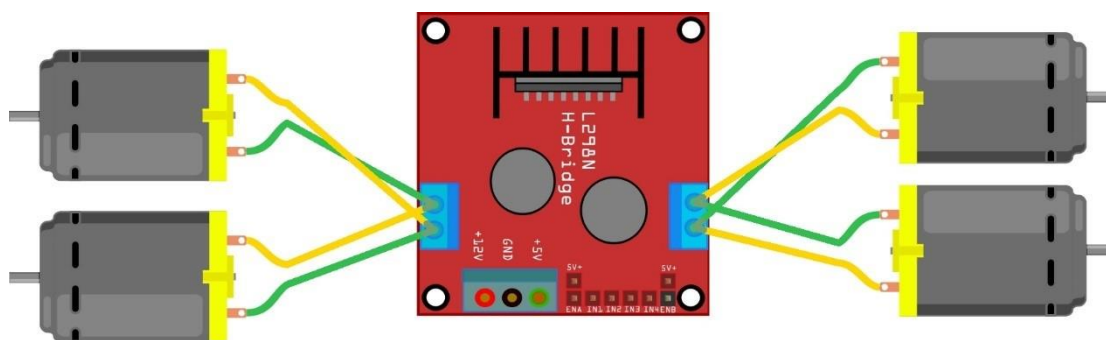


圖 3 L298N 與馬達的接線

將馬達 1 上側的線接到 1 號接口，下側的線接到 2 號接口，將馬達 2 上側的線接到 2 號接口，下側的線接到 1 號接口，將馬達 3 上側的線接到 3 號接口，下側的線接到 4 號接口，將馬達 4 上側的線接到 4 號接口，下側的線接到 3 號接口，1 號及 4 號為數位輸出，輸出高、低電壓，2 號及 3 號接口為類比輸出，可進行 PWM 調速。

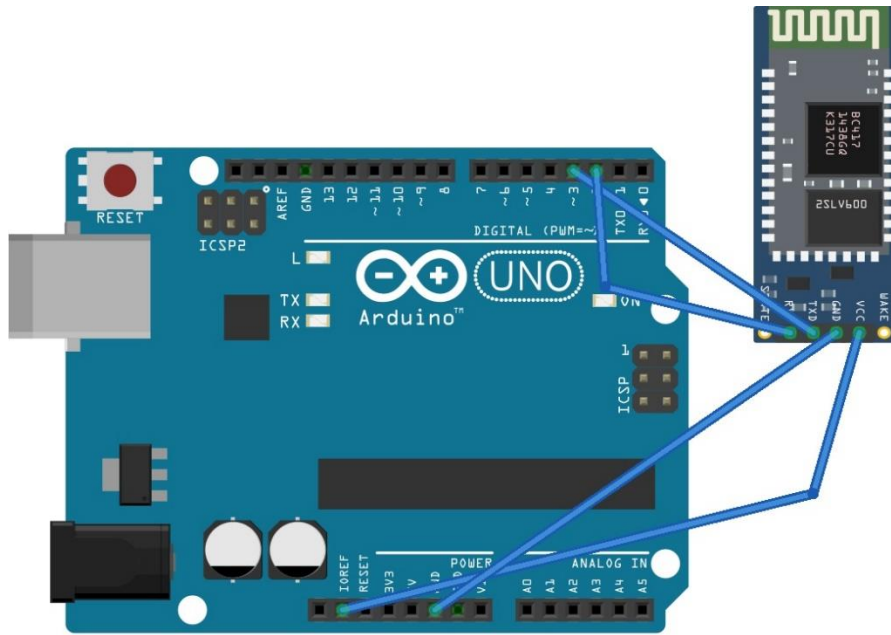


圖 4 藍芽模組與 Arduino UNO 的接線

將 VCC 接到 5V，GND 接到 GND，藍芽模組的傳輸端(TXD)與 3 號接口連接，藍芽模組的接收端(RXD)與 2 號接口連接，藍芽模組為接收手機所傳輸的訊息。

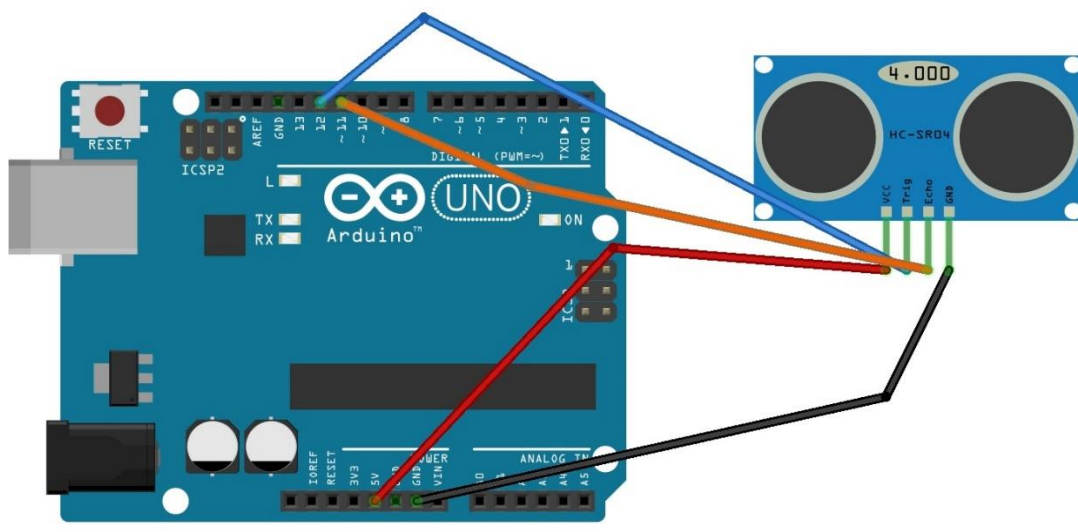


圖 5 超音波感測器與 Arduino UNO 的接線

將 VCC 接到 5V，GND 接到 GND，超音波感測模組的控制端(trig)接到 12 號接口，接收端(echo)接到 11 號接口，此模組藉由超音波接收的電位時間差之值加以做計算，即可得到物體與超音波感測模組之距離，超音波感測模組所感測的有效距離 2 到 400 公分，且超音波感測模組的準確率與距離成反比，所以當進行自動停車時，我們直接假設停車格的距離與車子本體為 15 公分以內，以達成最佳效果。

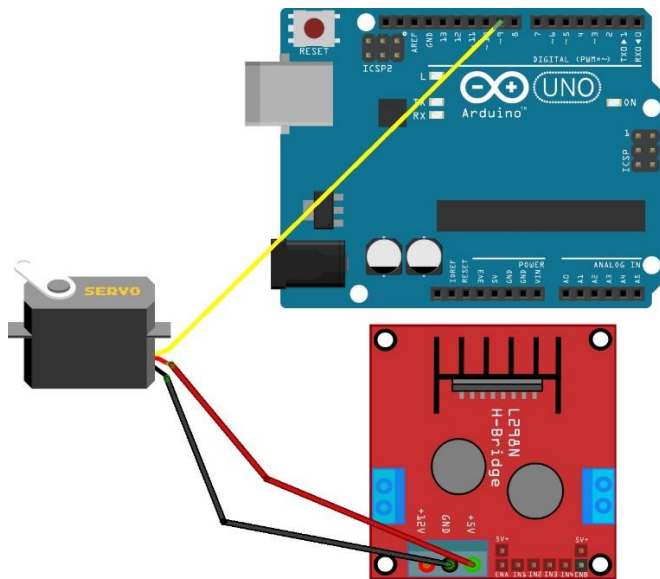


圖 6 伺服馬達與 Arduino UNO 的接線

將伺服馬達的紅線接到 5V，黑線接到 GND，訊號線接到 9 號接口，此伺服馬達的作用為藉由 SG-90 的運轉，帶動超音波感測模組(將把伺服馬達與超音波感測模組鎖在一起)，即可感測到更多面向的障礙物，使自動停車更加順利。

三、手機與藍芽連接之程式設計(使用 App inventor)



圖 6

為給予藍芽初始化，在還未與藍芽連接時，其他按鈕接不可被按，所以此時只有 bluetooth list 為 true，其他按鍵為 false。

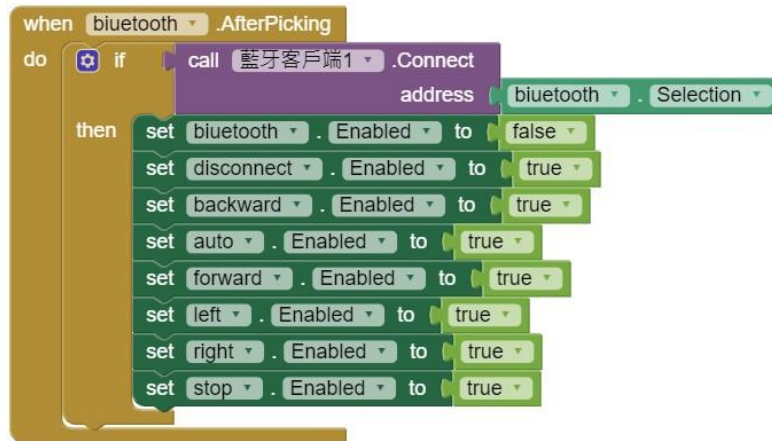


圖 7

當已經連接到藍芽時，將 bluetooth list 變成 false，將不可被點擊，把其他按鍵變成 true，即可被點擊。



圖 8

傳送資料給 bluetooth，當向右時會傳送 r 這個字元給藍芽客戶端，後退時會傳送 b 這個字元給藍芽客戶端，向左時會傳送 l 這個字元給藍芽客戶端，前進時會傳 f 這個字元給藍芽客戶端，停止時會傳送 s 這個字元給藍芽客戶端(上述接個字元給藍芽客戶端)。

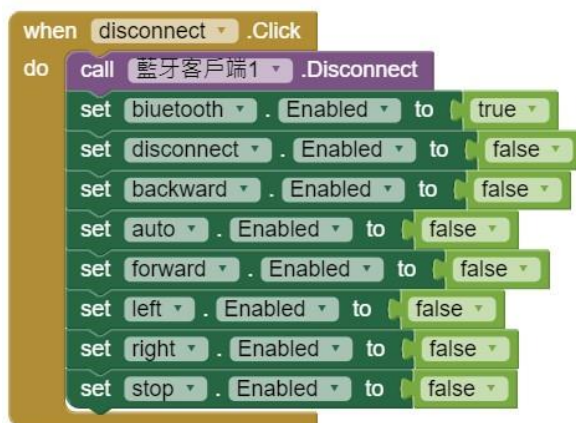


圖 9

當使用者要結束使用，則可點選 disconnect，點選後將會斷開使用者與藍芽的連線，並將其他按鍵變成 false(將不可被點選)，bluetooth list 改為 true(可以被點選)。

四、Arduino 程式設計

(一)由於採用 HC-05 以及 SG-90，所以要先引用 SoftwareSerial 和 Servo 函式庫。

```
#include <SoftwareSerial.h> // 引用「軟體序列埠」程式庫
#include<Servo.h>
```

圖 10

(二)做基本變數宣告。

```
SoftwareSerial BT(2, 3); // 設定軟體序列埠(接收腳, 傳送腳)
Servo myservo;
char data; // 接收序列埠值的變數
const int In1 = 4;
const int In2 = 5;
const int In3 = 6;
const int In4 = 7;
const int trigPin = 12; //Trig Pin
const int echoPin = 11; //Echo Pin
int finish = 0, finish_back = 0;
int angle, extent;
long duration, cm;
unsigned long start;
bool park = true, servo = true, pd = true; //parking detect
```

圖 11

(三)先預設輸出及輸入的腳位，以及與藍芽接收器和 Arduino UNO 與電腦連接的鮑率。

```
void setup() {
  myservo.attach(9);
  myservo.write(0);
  Serial.begin(9600);
  BT.begin(9600); // connect BT, rate = 9600
  pinMode(In1, OUTPUT);
  pinMode(In2, OUTPUT);
  pinMode(In3, OUTPUT);
  pinMode(In4, OUTPUT);
  pinMode(trigPin, OUTPUT); // declared I/O
  pinMode(echoPin, INPUT);
  pinMode(13, OUTPUT);
}
```

圖 12

(四)Arduino 與 L298N 的關係 (正反轉及馬達轉速)

L298N 介面(輸出及輸入位置)		控制的馬達
In1	OUT 1	In1 和 In2 控制左側兩顆馬達，當 In1 為低電壓時，為正轉，In2 為 PWM 輸出，可調整轉速。
In2	OUT 2	
In3	OUT 3	In3 和 In4 控制右側兩顆馬達，當 In4 為低電壓時，為正轉，In3 為 PWM 輸出，可調整轉速。
In4	OUT 4	

表(一)

	In1	In2(PWM)	In3(PWM)	In4
前進	LOW	100 數值越大轉速越快	100 數值越大轉速越快	LOW
後退	HIGH	150 數值越大轉速越慢	150 數值越大轉速越慢	HIGH
左轉	LOW	0 數值越大轉速越快	90 數值越大轉速越快	LOW
右轉	LOW	90 數值越大轉速越快	0 數值越大轉速越快	LOW
停止	LOW	0	0	LOW

表(二)

表(一)為 L298N 輸出位置。表(二)為控制正反轉及轉速的數值，經由測驗發現當前進時，PWM(註 1)數值越大，轉速越快，然而後退時，PWM 數值越小，轉速越快，因為在後退時 In1 和 In2 為 HIGH，而馬達是藉由高低電壓差來決定正反轉，所以當其中一個接口為 HIGH，另一個應為 LOW，所以若要進行調速，PWM 的值必須越小(越接近 LOW)，轉速才能快。且 PWM 的數值大約應設在 45~255(隨著電池的電量的不同值會有所改變)，因為值太小，電池無法供應四顆馬達運轉，所以此時的馬達將無法運轉，值過大時將無差異，因為 PWM 輸出最大值為 255。

註(一)：Arduino 的輸出有數位輸出及類比輸出，數位輸出只會輸出 HIGH 不然就是 LOW 兩種電壓，而類比輸出會輸出 0~255，不同的數值會產生不同的電量，此稱為 PWM 輸出。

(五)前進、後退、向左、向右、停止的程式碼(數值皆可調整)。

```
void forward() {
    digitalWrite(In1, LOW);
    digitalWrite(In4, LOW);
    analogWrite(In2, 100);
    analogWrite(In3, 100);
}
void backward() {
    digitalWrite(In1, HIGH);
    digitalWrite(In4, HIGH);
    analogWrite(In2, 150);
    analogWrite(In3, 150);
}
void turnleft() {
    digitalWrite(In1, LOW);
    digitalWrite(In4, LOW);
    analogWrite(In2, 0);
    analogWrite(In3, 90);
}
void turnright() {
    digitalWrite(In1, LOW);
    digitalWrite(In4, LOW);
    analogWrite(In2, 90);
    analogWrite(In3, 0);
}
void stopping() {
    digitalWrite(In1, LOW);
    digitalWrite(In4, LOW);
    analogWrite(In2, 0);
    analogWrite(In3, 0);
}
```

圖 13

(六)超音波感測模組執行程式。

```
void detect() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH); //give Trig HIGH, continue 10 microsecond
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    pinMode(echoPin, INPUT); //read echo electric
    duration = pulseIn(echoPin, HIGH); // time of receiving HIGH time
    cm = (duration/2) / 29.1; //time to cm
}
```

圖 14

藉由超音波的發送以及接收，並運用 pulseIn() 函式計算高電壓的時間，將高電壓的時間設為 x，由於超音波的發射有發射及接收，所以要先將 x 除以 2(去及回)，再將此數值除以 29.1(註 2)，即可算出物體與車身的距離。

註 2: 由於音速的公式為 $c = 331.5 + 0.607 * (\text{溫度})$ ，此時我們假設室溫為 20 度 C，所以可得目前音速為 343.64m/sec，且超音波感測模組在進行測量時都是以微秒在進行的，加上我們測量地距離比較小，所以我們將單位縮小成公分，因此可得此刻音速大約為 0.03436cm/ microsecond，最終可以計算出音速行走一公分大約為 29.1 微秒。

(七)自動停車

<pre>int autoparking() { digitalWrite(In1, LOW); digitalWrite(In4, LOW); analogWrite(In2, 50); analogWrite(In3, 42); while(park == true) { detect(); if(cm >= 16) { start = millis(); while(cm >= 16) { detect(); } finish = millis() - start; detect(); } } }</pre>	<pre>if(finish < 1562) { Serial.println(finish); myservo.write(100); digitalWrite(In1, HIGH); digitalWrite(In4, HIGH); analogWrite(In2, 190); analogWrite(In3, 190); delay(300); digitalWrite(In1, HIGH); digitalWrite(In4, LOW); analogWrite(In2, 70); analogWrite(In3, 50); delay(430); digitalWrite(In1, HIGH); digitalWrite(In4, HIGH); analogWrite(In2, 175); analogWrite(In3, 175); delay(200); digitalWrite(In1, LOW); digitalWrite(In4, LOW); analogWrite(In2, 0); analogWrite(In3, 0); myservo.write(0); park = false; return 0; }</pre>
--	--

圖 15

圖 16

圖 15 為給予車子初始速度，由於物理元件的關係，左側及右側輪子無法保持相同轉速，所以我們將 In2 的轉速調快以確保車子能直線行駛，並在車子行駛中持續使用超音波感測模組進行測量距離，測量直到距離大於 16 公分，此時會把這個時間做紀錄，並等到距離再次小於 16 公分時做時間紀錄，將 millis() 函式在背景執行的時間減去一開始記錄的時間即可得到經過停車格的時間。

我們設計的自動停車分為倒車入庫以及路邊停車，判斷標準為車子行經停車格所經過的時間，若行經時間大於 1.562 秒則進行路邊停車，若行經時間小於 1.562 秒則進行倒車入庫，會採取此方式的原因為車子無法使用路邊停車的方式停入停車格較小的地方，所以此時直接讓車子將此情況是為要停入車庫，實行倒車入庫。以上數據取得方式如下：

首先測量車子行駛之速率，車子行駛 196.4 公分的時間是以平均來做計算，實施 6 次，實施六次實驗結果若下：

第一次	6.85 秒
第二次	6.43 秒
第三次	6.60 秒
第四次	6.59 秒
第五次	6.49 秒
第六次	6.73 秒

平均:6.615 秒

由速率=距離除以時間可知 213 公分/6.615 秒=32.199cm/sec。接著測量車子的長度為 25 公分(車子實體如圖 17)，並做多次實驗後發現路邊停車的安全距離為車子前後 12.65 公分，進行加總後得出停車格為 50.3 公分時為路邊停車最短安全距離。由時間=距離除以速率可知 50.3 公分除以 32.199cm/sec=1.562 秒。並由此數字做為倒車入庫及路邊停車的標準值。

圖 16 的類比輸出數值為經多次實驗的微調所得出的最佳數值，經由輪差可以使車子做出轉彎的動作，在運用高、低電壓的差別做出前進及後退的效果。



圖 17

<pre> Serial.println(finish); myservo.write(100); digitalWrite(In1,HIGH); digitalWrite(In4,HIGH); analogWrite(In2,190); analogWrite(In3,190); delay(300); digitalWrite(In1,HIGH); digitalWrite(In4,LOW); analogWrite(In2, 85); analogWrite(In3, 65); delay(480); digitalWrite(In1,HIGH); digitalWrite(In4,HIGH); analogWrite(In2,170); analogWrite(In3,170); delay(450); detect(); </pre>	<pre> while(k != 5){ digitalWrite(In1,LOW);//right digitalWrite(In4,LOW); analogWrite(In2, 120); analogWrite(In3, 50); delay(460); digitalWrite(In1,HIGH);//back digitalWrite(In4,HIGH); analogWrite(In2, 170); analogWrite(In3, 170); delay(450); k++; } </pre>	<pre> digitalWrite(In1,LOW); digitalWrite(In4,LOW); analogWrite(In2, 60); analogWrite(In3, 60); delay(500); digitalWrite(In1,LOW); digitalWrite(In4,LOW); analogWrite(In2, 0); analogWrite(In3, 0); myservo.write(0); park = false; return 0; </pre>
---	--	--

圖 18

圖 19

圖 20

圖 18、圖 19、圖 20 為進行路邊停車的程式，伺服馬達為進行除錯較方便的工具，對於執行結果無關。程式的最後將 park 改變成 false，宣布自動停車結束。

伍、研究結果

本研究結果將分為 3 部分說明，第一部分探討進行停車前的前置作業，第二部分為倒車入庫，第三部分為路邊停車。

一、停車前的前置作業

我們將車子的設定成初始位置皆在前一個障礙物前，如圖 22，並使車子維持一定的速度向前行，直到到達第二個障礙物，將會進行自動停車。

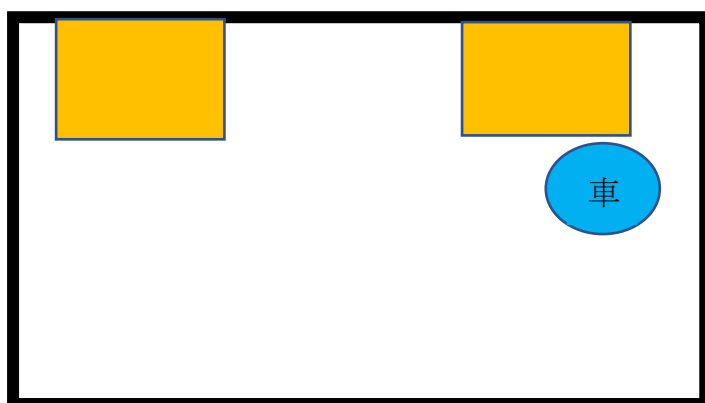


圖 22

二、倒車入庫

當行經距離小於 50.3 公分時進行倒車入庫，參考圖 23。倒車入庫的流程如下：

第一步	往後 0.3 秒。
第二步	進行退後轉彎，轉彎的方向為向右後方轉，所以此時左側馬達轉速較快，持續 0.48 秒。
第三步	往後 0.25 秒。
第四步	停止馬達。
第五步	結束倒車入庫。

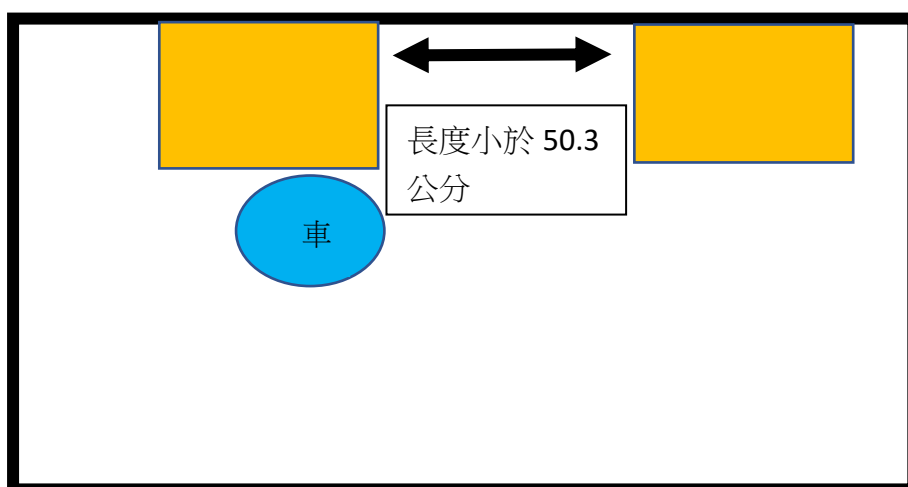


圖 23

三、路邊停車

當行經距離大於 50.3 公分時進行路邊停車，參考圖 24。路邊停車的流程如下：

第一步	往後 0.3 秒。
第二步	進行退後轉彎，轉彎的方向為向右後方轉，所以此時左側馬達轉速較快，持續 0.48 秒，此處與倒車入庫類似，但類比輸出數值不太一樣。
第三步	往後 0.45 秒。
第四步	重複第五步及第六步四次。
第五步	進行轉彎，轉彎的方現為項由前方轉，此時左側馬達轉速較快，持續 0.46 秒。
第六步	後退 0.45 秒。
第七步	往前 0.5 秒，做最終位置的微調。
第八步	停止馬達。
第九步	結束路邊停車。

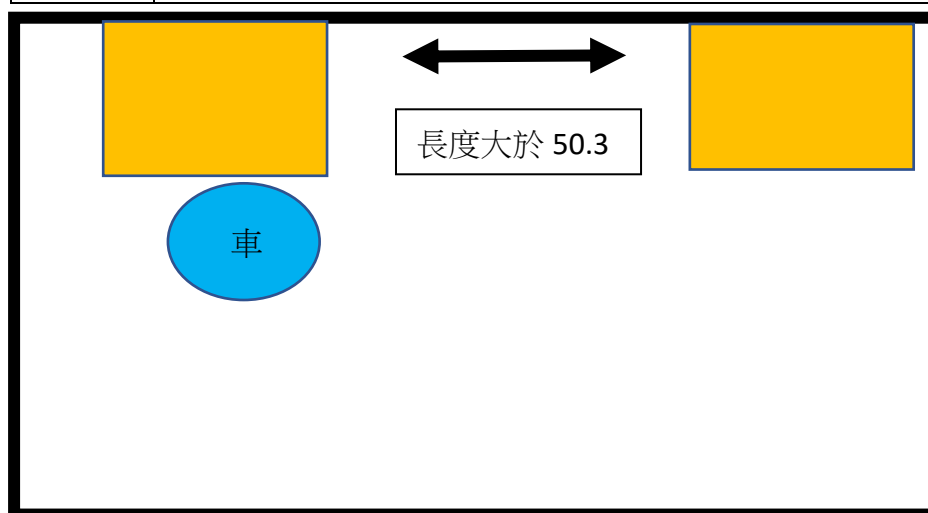


圖 24

陸、討論

此作品為我們完成的第一個作品，從一開始慢慢熟悉 Arduino 程式，一直到完成 Arduino 開發版與各項模組的接線，經過多次除錯、數據微調，這過程中收穫良多，以下為我們對這次作品提出的討論：

一、Q:『本實驗的停車結果是否會有停車不整齊的情況(無法與停車格對齊)?』

A:『有的，由於本實驗的停車路徑較為死板，所以當進行停車時，車子停止角度會影響最終停車結果，還有我們所購買的物理元件價格較為低廉，所以物理元件的損壞速度較快，因此也會造成誤差。』

二、Q:『為何不做多面向的超音波感測，多做幾個超音波感測不是能使停車路徑更加靈活?』

A:『原本有想過說利用伺服馬達的旋轉帶動超音波感測，但是礙於技術的限制，我們無法利用此方法得知車子的實際位置，只能做更多面向的偵測而已，所以幫助不大。』

三、Q:『此作品是否還有其他的限制或哪部分無法突破?』

A:『此停車系統會礙於停車格大小，若停車格太小有可能撞到其他障礙物，所以我們是在假設障礙物的寬是大於車子本體的寬許多的情況進行實驗。』

柒、 結論

進行停車為許多駕駛的困擾之處，若以本實驗結果再加以做其他方面的調整，並把物體放大，再考慮現實生活中會遇到的狀況，並把此系統普及化，我想這將會解決許多人的困擾。

此作品能計算停車格的大小，並做加以判斷，給予較活躍的停車方式，且此作品運用超音波感測模組也是本作品的一大特點之一，藉由超音波感測模組的感測，使整個停車系統更加完善、安全。

製作作品時遇到許多問題，但是經由不斷的嘗試、不斷的研究，造就出此作品。

捌、 參考資料及其他

一、學會 C 語言 從不懂，到玩上手，完整學習 C 語言程式設計和 Arduino 的 C 語言 陳會安 著

二、Arduino 互動科技 <http://163.22.166.134/arduino/model/1298n.html>

三、Arduino 筆記 <https://atceiling.blogspot.com/p/arduino.html>

摘要

人們在進行停車時，常常無法拿捏好停車格的距離，本研究是將現實問題縮小，並進行小型模擬來實驗車子在哪個位置中需要使用多少輪差和在限定空間內進行安全且快速的停車，此方面由超音波感測器進行感測，並做出判斷，若此距離可進行停車，則會自動控制車子本體移動到適當的位置進行停車。

主要器材: Arduino 開發版、超音波感測器、馬達、電池盒、馬達驅動板、電池、輪子、杜邦線、壓克力板、藍芽模組。

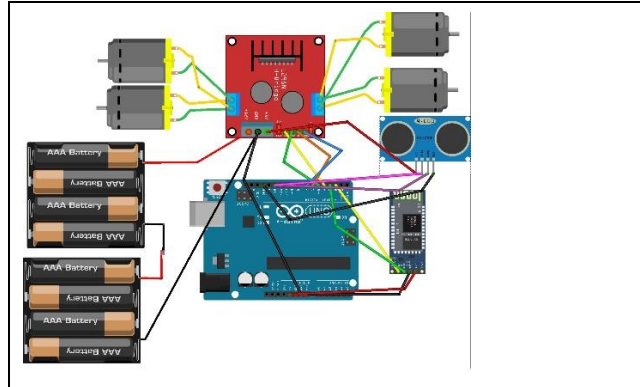
研究過程及方法

(一) 硬體設備配置

一、預估版面



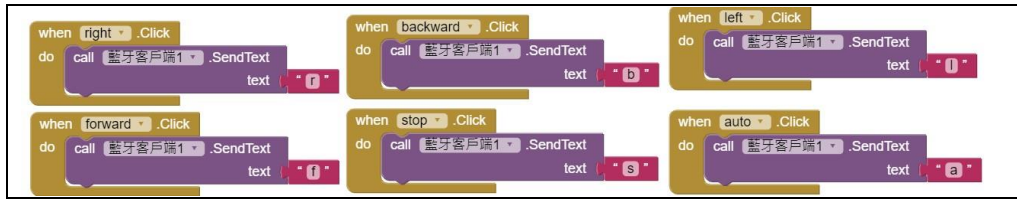
二、



電路及接線

此為此車的總接線圖，L298N 會給予四顆馬達電力，此處的超音波感測器可以接 Arduino 的 5V 也可以接 L298N 的 5V，我們利用四顆鋰電池驅動 L298N。

三、手機與藍芽連接之程式設計(使用 App inventor)



傳送資料給 bluetooth，當向右時會傳送 r 這個字元給藍芽客戶端，後退時會傳送 b 這個字元給藍芽客戶端，向左時會傳送 l 這個字元給藍芽客戶端，前進時會傳 f 這個字元給藍芽客戶端，停止時會傳送 s 這個字元給藍芽客戶端(上述每個字元給藍芽客戶端)。

四、Arduino 程式設計

(一)做基本變數宣告。

```
SoftwareSerial BT(2, 3); // 設定軟體序列埠 (接收腳, 傳送腳)
Servo myservo;
char data; // 接收序列埠值的變數
const int In1 = 4;
const int In2 = 5;
const int In3 = 6;
const int In4 = 7;
const int trigPin = 12; //Trig Pin
const int echoPin = 11; //Echo Pin
int finish = 0, finish_back = 0;
int angle, extent;
long duration, cm;
unsigned long start;
bool park = true, servo = true, pd = true; //parking detect
```

(二)Arduino 與 L298N 的關係 (正反轉及馬達轉速)

經由測驗發現當前進時，PWM 數值越大，轉速越快，然而後退時，PWM 數值越小，轉速越快，因為在後退時 In1 和 In2 為 HIGH，而馬達是藉由高低電壓差來決定正反轉，所以當其中一個接口為 HIGH，另一個應為 LOW，所以若要進行調速，PWM 的值必須越小(越接近 LOW)，轉速才能快。PWM 的數值不能太小，因為值太小，電池無法供應四顆馬達運轉，所以此時的馬達將無法運轉。

(二)Arduino 與 L298N 的關係 (正反轉及馬達轉速)

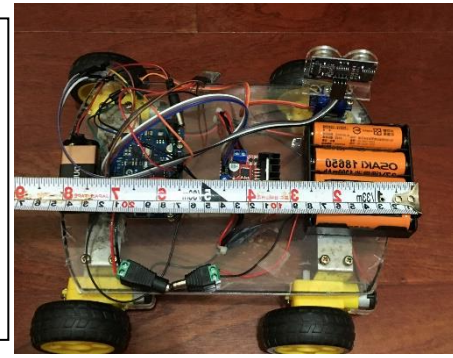
	In1	In2(PWM)	In3(PWM)	In4
前進	LOW	100 數值越大轉速越快	100 數值越大轉速越快	LOW
後退	HIGH	150 數值越大轉速越慢	150 數值越大轉速越慢	HIGH
左轉	LOW	0 數值越大轉速越快	90 數值越大轉速越快	LOW
右轉	LOW	90 數值越大轉速越快	0 數值越大轉速越快	LOW
停止	LOW	0	0	LOW

L298N 介面(輸出及輸入位置)		控制的馬達
In1	OUT 1	In1 和 In2 控制左側兩顆馬達，當 In1 為低電壓時，為正轉，In2 為 PWM 輸出，可調整轉速。
In2	OUT 2	
In3	OUT 3	In3 和 In4 控制右側兩顆馬達，當 In4 為低電壓時，為正轉，In3 為 PWM 輸出，可調整轉速。
In4	OUT 4	

(三)超音波感測模組執行程式。

```
void detect() {  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(5);  
    digitalWrite(trigPin, HIGH); //give  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
    pinMode(echoPin, INPUT); //read echo  
    duration = pulseIn(echoPin, HIGH);  
    cm = (duration/2) / 29.1; //time to  
}
```

藉由超音波的發送以及接收，並運用 pulseIn() 函式計算高電壓的時間，將高電壓的時間設為 x，由於超音波的發射有發射及接收，所以要先將 x 除以 2(去及回)，再將此數值除以 29.1，即可算出物體與車身的距離。



(四)自動停車

```
int autoparking() {  
    digitalWrite(In1, LOW);  
    digitalWrite(In4, LOW);  
    analogWrite(In2, 50);  
    analogWrite(In3, 42);  
    while (park == true) {  
        detect();  
        if (cm >= 16) {  
            start = millis();  
            while (cm >= 16) {  
                detect();  
            }  
            finish = millis() - start;  
            detect();  
        }  
    }  
}
```

```
if (finish < 1562) {  
    Serial.println(finish);  
    myservo.write(100);  
    digitalWrite(In1, HIGH);  
    digitalWrite(In4, HIGH);  
    analogWrite(In2, 190);  
    analogWrite(In3, 190);  
    delay(300);  
    digitalWrite(In1, HIGH);  
    digitalWrite(In4, LOW);  
    analogWrite(In2, 65);  
    analogWrite(In3, 50);  
    delay(480);  
    digitalWrite(In1, HIGH);  
    digitalWrite(In4, HIGH);  
    analogWrite(In2, 170);  
    analogWrite(In3, 170);  
    delay(250);  
    digitalWrite(In1, LOW);  
    digitalWrite(In4, LOW);  
    analogWrite(In2, 0);  
    analogWrite(In3, 0);  
    myservo.write(0);  
    park = false;  
    return 0;  
}
```

我們設計的自動停車分為倒車入庫以及路邊停車，判斷標準為車子行經停車格所經過的時間，若行經時間大於 1.562 秒則進行路邊停車，若行經時間小於 1.562 秒則進行倒車入庫，會採取此方式的原因為車子無法使用路邊停車的方式停入停車格較小的地方，所以此時直接讓車子將此情況是為要停入車庫，實行倒車入庫。

```
Serial.println(finish);  
myservo.write(100);  
digitalWrite(In1, HIGH);  
digitalWrite(In4, HIGH);  
analogWrite(In2, 190);  
analogWrite(In3, 190);  
delay(300);  
digitalWrite(In1, HIGH);  
digitalWrite(In4, LOW);  
analogWrite(In2, 85);  
analogWrite(In3, 65);  
delay(480);  
digitalWrite(In1, HIGH);  
digitalWrite(In4, HIGH);  
analogWrite(In2, 170);  
analogWrite(In3, 170);  
delay(450);  
detect();  
int k = 0;  
while (k != 4) {  
    digitalWrite(In1, LOW); //right  
    digitalWrite(In4, LOW);  
    analogWrite(In2, 125);  
    analogWrite(In3, 50);  
    delay(460);  
    digitalWrite(In1, HIGH); //back  
    digitalWrite(In4, HIGH);  
    analogWrite(In2, 170);  
    analogWrite(In3, 170);  
    delay(450);  
    k++;  
}  
digitalWrite(In1, LOW);  
digitalWrite(In4, LOW);  
analogWrite(In2, 60);  
analogWrite(In3, 60);  
delay(500);  
digitalWrite(In1, LOW);  
digitalWrite(In4, LOW);  
analogWrite(In2, 0);  
analogWrite(In3, 0);  
myservo.write(0);  
park = false;  
return 0;
```

由速率=距離除以時間可知 213 公分/6.615 秒=32.199cm/sec。接著測量車子的長度為 25 公分，並做多次實驗後發現路邊停車的安全距離為車子前後 12.65 公分，進行加總後得出停車格為 50.3 公分時為路邊停車最短安全距離。由時間=距離除以速率可知 50.3 公分除以 32.199cm/sec=1.562 秒。並由此數字做為倒車入庫及路邊停車的標準值。

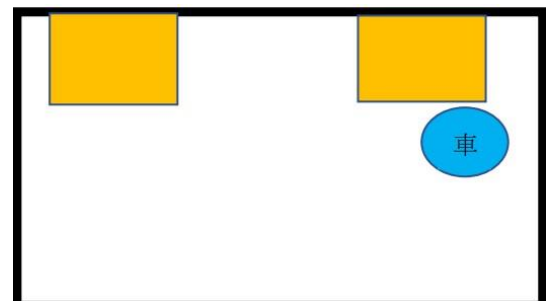
藉由兩障礙物中的距離作出路邊停車或是倒車入庫的判斷。

結論

進行停車為許多駕駛的困擾之處，若以本實驗結果再加以做其他方面的調整，並把物體放大，再考慮現實生活中會遇到的狀況，並把此系統普及化，我想這將會解決許多人的困擾。

此作品能計算停車格的大小，並做加以判斷，給予較活躍的停車方式，且此作品運用超音波感測模組也是本作品的一大特點之一，藉由超音波感測模組的感測，使整個停車系統更加完善、安全。

製作作品時遇到許多問題，但是經由不斷的嘗試、不斷的研究，造就出此作品。



參加過程心得

從一開始下定決心要報名，到最後完成作品並送去比賽，這過程中學到的不只有 Arduino 的應用，還學習到團隊的合作以及不輕易放棄的意志。Arduino 是一個令我非常佩服的產品，運用短短幾行的程式碼輸入到 Arduino 的晶片裡，竟可以做出令人出乎意料的事來。

在剛開始做車子時我就發現光要利用馬達驅動模組驅動一個馬達就十分困難了，但經由不斷的接線測試以及程式的修改，終於完成馬達的驅動，但困難卻一件一件的迎我而來，該如何利用藍芽連接控制？超音波感測模組感測時應該設在幾公分處為正常判斷區域？停車格的距離要設幾公分才不會在進行停車時產生錯誤？這些問題是除了需要程式語法的觀念，還要經由多次測試，以及參數的更改才有辦法得出的結果，然而這當中最快樂的不是完成作品，其實是學習到程式的撰寫、Arduino 的接線、電流的輸出及輸入，以及培養耐心，因為做出作品只是暫時的，而學習到的知識卻是可受用一輩子。這過程中我從一位連控制 LED 燈都有困難的人到學會如何遠端傳送及接收資料，在這樣一步一腳印的學習中，我最終得到一個結論，學習不單單只是為了證明自己的能力，更重要的是要找尋到自己的興趣和獲得可受用一輩子的知識。

