

臺北市立大安高級工業職業學校

電子科

專題報告

語音控制自走車

Voice controlled self-propelled car

學生 組長:林欣翰
組員:施欣妤
組員:林侑賢
組員:蔡葆錡

指導老師:張瑞芬老師

中華民國 108 年 1 月

摘要

隨著科技的日新月異，許多新產品相繼開發，語音控制的應用也越來越多元，我們希望結合語音控制與小型自走車，不只能夠自動避障，還能利用語音辨識聽從指令來判斷前進的方向，自動避障是將超音波感測器偵測到的數據藉由 Arduino IDE 所撰寫的程式來進行判斷並執行避障的功能。

語音辨識自走車可以提供多樣化的應用，如日常用品的搬運。可成為個人的日常生活上的好幫手。透過團隊合作的方式，達到系統整合所需的相關技術，進而應用在其它不同目的的系統上。

關鍵詞：Arduino IDE、超音波感測器、自走車、語音辨識

目錄

摘要.....	I
目錄.....	II
圖目錄.....	III
表目錄.....	錯誤! 尚未定義書籤。
第 1 章 前言.....	1
1-1 專題製作背景及目的.....	1
1-2 預期成果.....	1
第 2 章 理論探討.....	2
2-1 硬體.....	2
2-2 軟體.....	6
第 3 章 專題準備.....	8
3-1 專題架構圖.....	8
3-2 專題流程圖.....	8
第 4 章 專題成果.....	9
4-1 問題發現與解決.....	9
4-2 專題成果.....	9
第 5 章 結論與建議.....	10
5-1 結論.....	10
5-2 建議.....	10
參考文獻.....	10
附錄.....	11

圖目錄

圖 1 ARDUINO UNO 板	3
圖 2 超音波避障模組 HC-SR04	4
圖 3 馬達驅動模組 L9110S	5
圖 4 馬達驅動模組 L9110S 電路圖	5
圖 5 GA12-N20 微型金屬馬達	6
圖 6 ARDUINO IDE 介面	7
圖 7 專題流程圖	8

第1章 前言

1-1 專題製作背景及目的

坊間已經有許多使用不同方式控制的自走車，而我們希望製作出能夠使用聲音來操控並利用超聲波感測自動避障的自走車，應用範圍廣泛，希望能夠有利於未來的科技發展

1-2 預期成果

- 1.超聲波感測自動避障
- 2.語音控制車子行走方向
- 3.OPA 放大電路

第2章 理論探討

2-1 硬體

2-1-1 ARDUINO UNO 板

Arduino 是一個開放原始碼的單晶片微控制器，使用了 Atmel AVR 單晶片，採用了開放原始碼的軟硬體平台，建構於簡易輸出/輸入介面，並且具有使用類似 Java、Processing 的開發環境，也因為其簡單操控的特性，我們選擇使用 UNO 板來配合專題，既可以達到我們的需求，也不會需要太多困難的操作。Arduino Uno (圖 1) 的運作直流電壓為 5V，可由 USB 連接埠、電源輸入 插座及 Vin 接腳提供電源。輸出接腳腳位則分成數位輸出以及類比輸出。數位腳位編號由 D0 至 D13，D 則代表 Digital。其中 D3、D5、D6、D9、D10、D11 的數字編號旁，有波浪符號 (~)，表示這些腳位可以使用 PWM (Pulse Width Modulation) 的方法來模擬出類比訊號。類比腳位編號則由 A0 至 A5，可用來接受類比電壓輸入，但不能輸出類比電壓，如前所述，類比電壓必須透過數位腳位 D3、D5、D6、D9、D10、D11，以 PWM 模擬。ATmega328 內建類比數位轉換器 (Analog-to-digital converter，簡稱 ADC)，預設會將 0V 到 5V 轉換為 0 至 1023 的數值。實際上，A0 至 A5 也可作為數位輸出、輸入腳位使用，此時 A0 至 A5 分別可視為 D14 至 D19。

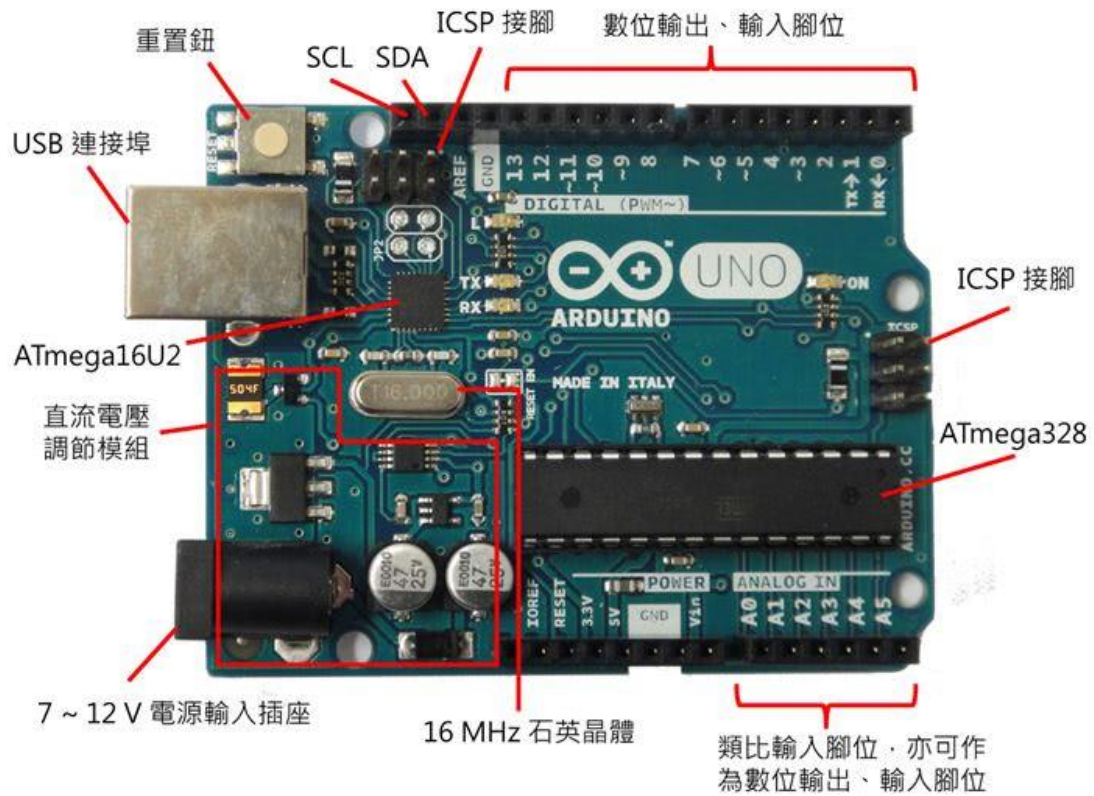


圖 1 ARDUINO UNO 板

參考連結: <https://openhome.cc/Gossip/Books/mBlockArduino1-3and1-4.html>

2-1-2 超音波避障模組 HC-SR04

超音波感測器是由超音波發射器、接收器和控制電路所組成。當它被觸發的時候，會發射一連串 40 kHz 的聲波並且從離它最近的物體接收回音。超音波是人類耳朵無法聽見的聲音，因為它的頻率很高。

如下圖(圖 2)所示，超音波測量距離的方法，是測量聲音在感測器與物體之間往返經過的時間：

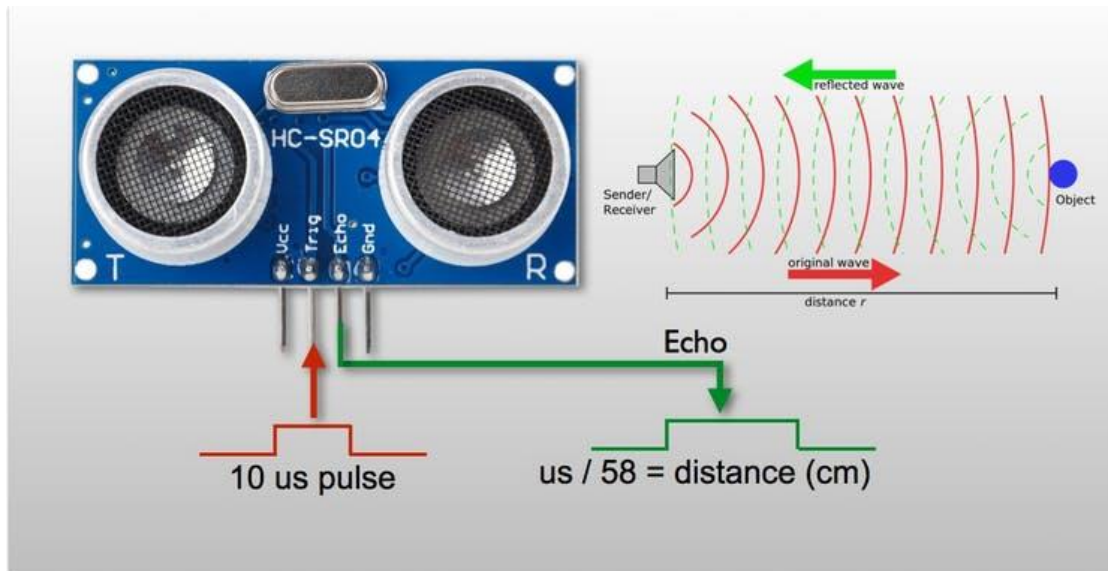


圖 2 超音波避障模組 HC-SR04

參考連結: <https://www.hackster.io/powerberry/like-a-bat-with-hc-sr04-829486>

2-1-3 馬達驅動模組 L9110S

L9110S 直流電機驅動板，板載 2 個 L9110S 馬達控制晶片，模組可以同時驅動 2 個直流電機或 1 個 4 線 2 相式步進電機，模組輸入電壓：2.5-12V，每通道具有 800mA 連續電流輸出能力，

模組描述：

型號：L9110S

- 1 雙 L9110S 晶片的馬達驅動
- 2 模組供電電壓：2.5-12V
- 3 適合的馬達範圍：馬達工作電壓 2.5v-12V 之間，最大工作電流 0.8A
- 4 可以同時驅動 2 個直流馬達，或者 1 個 4 線 2 相式步進馬達
- 5 PCB 板尺寸：2.8cm*2.1cm
- 6 固定安裝孔，直徑：3mm

L9110S 直流電機/步進電機驅動板



圖 3 馬達驅動模組 L9110S

參考連結：

<https://sites.google.com/site/wenyutransformer/02jin-jie-pian/12kong-zhi-zi-zou-che-19110s>

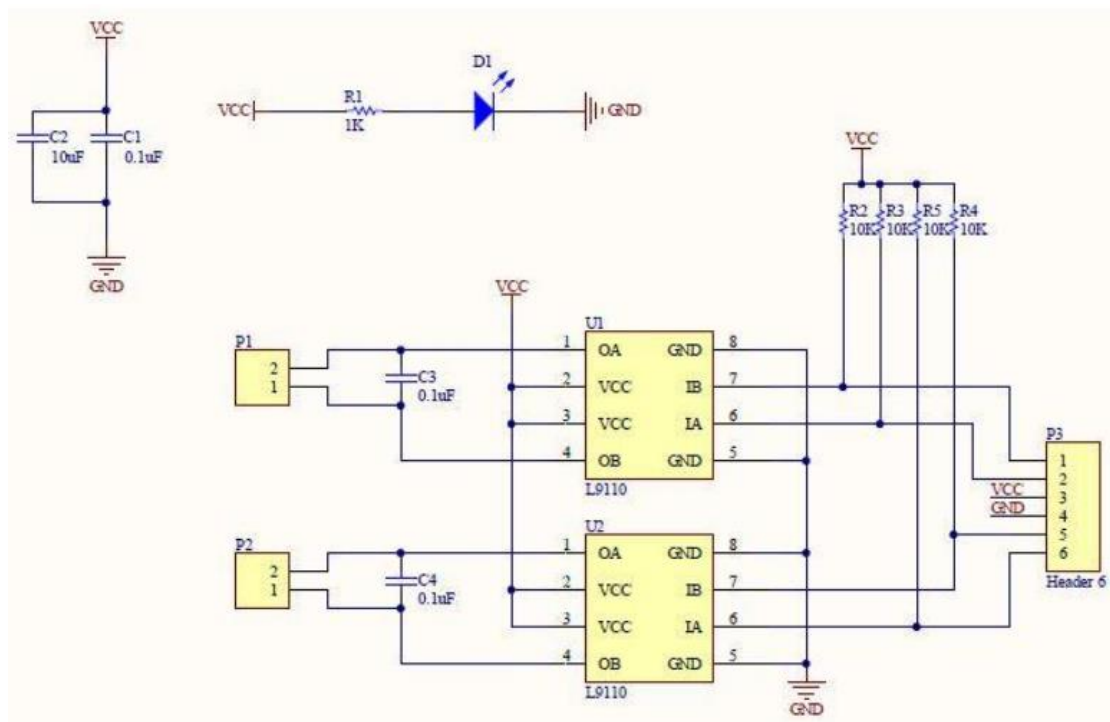


圖 4 馬達驅動模組 L9110S 電路圖

參考連結: <http://pizgchen.blogspot.com/2016/05/19110s.html>

2-1-4 GA12-N20 微型金屬馬達

產品型號 : GA12-N20

直 徑 : 12mm

軸 徑 : 3mm

軸 長 : 10mm

標準電壓: 6V (電壓:3V~12V)

應用輸入電壓: 3~12V 之間



圖 5 GA12-N20 微型金屬馬達

2-2 軟體

Arduino 是以 AVR-GCC 和其他一些開源軟體為基礎，以 Java 編寫的跨平台應用軟體，軟體無需安裝，只要到官方網站下載並完成解壓縮後即可開始使用。

Arduino Software IDE 使用與 C 語言和 C++相仿的程式語言，並且提供了包含常見的輸入/輸出函式的函式庫。程式會包含兩個函式： `setup()`：在程式執行開始時會執行一次，用於初始化設定。 `loop()`：直到 Arduino 硬體關閉前會重複執行函式內的程式碼。此專題所使用的 Arduino 版本為 1.8.5，其介面如圖 12，白色部分為程式撰寫部分，黑色部分為偵錯顯示內容。

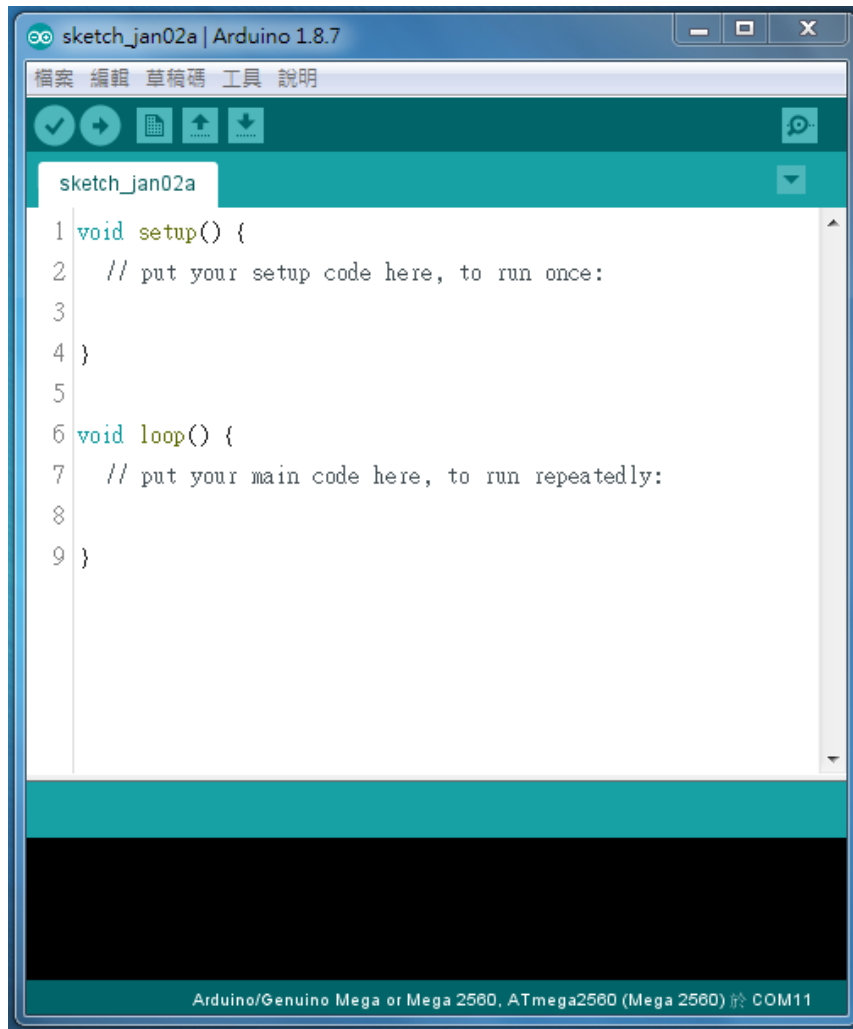
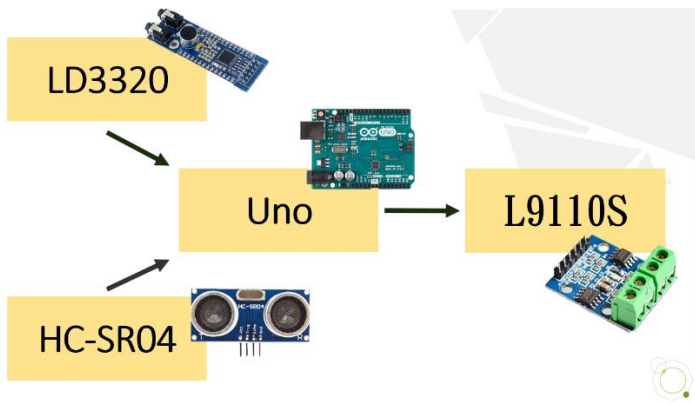


圖 6 ARDUINO IDE 介面

第3章 專題準備

3-1 專題架構圖



3-2 專題流程圖

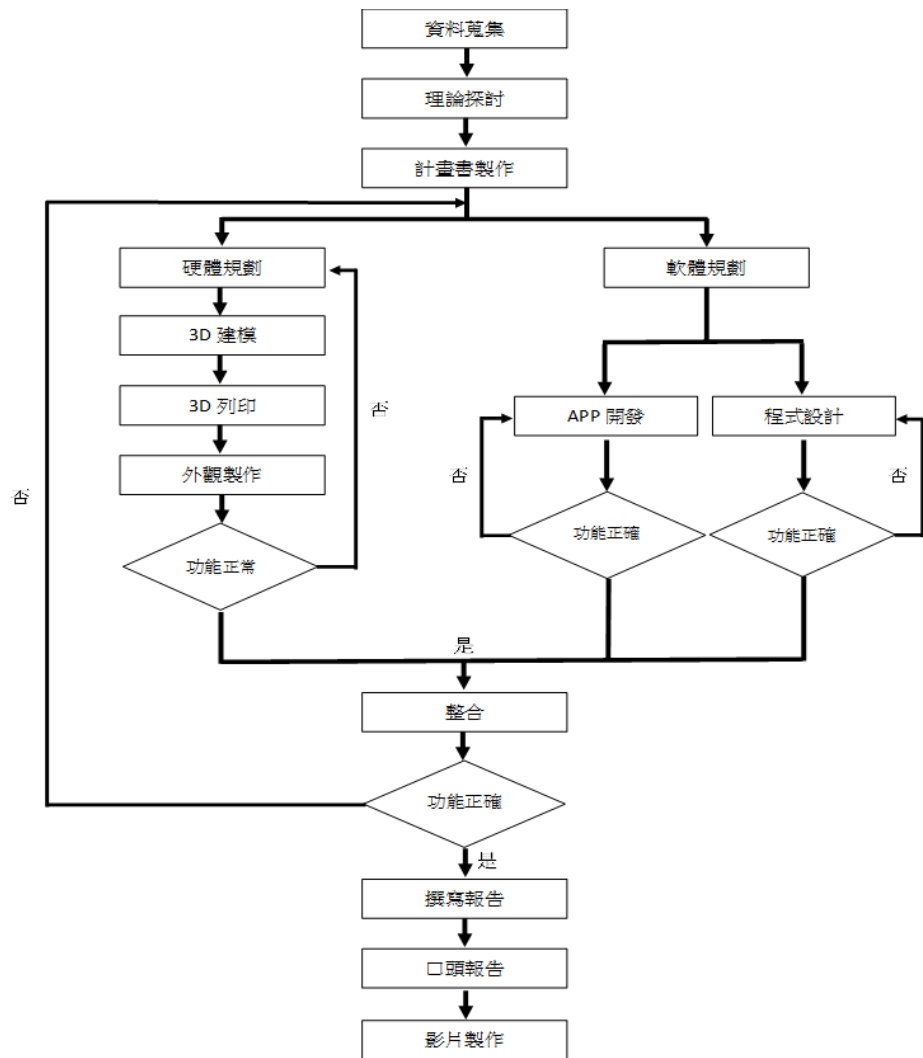


圖 7 專題流程圖

第4章 專題成果

4-1 問題發現與解決

4-1-1 指向性實驗成果不理想

我們本來打算利用麥克風的指向性製作語音辨識來判斷發聲點的位置並自動朝發聲位置前進，嘗試過各種隔音方法也詢問過專業人士的建議(亞哥音響老闆)但成果不理想，因此我們利用 LD3320 模組改為語音控制的方式接收指令並依照指令進行移動，例如:前進、後退、左轉、右轉。

4-1-2 地面摩擦力過大導致輪胎空轉

輪胎與地面摩擦力過大的關係，導致輪胎轉速不同，造成空轉的情形，試著改變場地材質，製作摩擦力較小的平面。

4-2 專題成果

- 1.超聲波感測避障
- 2.語音控制車子行走方向
- 3.OPA 放大電路

第5章 結論與建議

5-1 結論

經過實驗與討論，我們終於成功製作出語音控制自走車，有了超音波避障和語音控制使自走車的應用更加的多元與完整，這次專題始我們瞭解到成功的關鍵不只是學術上的學習成果，團隊間的默契與配合度更是重要，希望之後可以將本次專題進行更深入的研究，對未來的科技發展有所貢獻。

5-2 建議

只要找更好的隔音方法製作出指向性麥克風也許就能成功製作出語音辨識的自走車，之後可以朝這個方向繼續研究。

參考文獻

- 一、黃文玉 (2016)。阿玉 Transformer+Scratch 研究區。2019 年 3 月 13 日，取自 <https://reurl.cc/YERlo>
- 二、Pizg Chen (2016)。L9110S 直流馬達驅動模塊。2019 年 3 月 13 日，取自 <http://pizgchen.blogspot.com/2016/05/l9110s.html>
- 三、Cesare A. Perani (2016)。How to use HC SR04 to measure distance。2019 年 3 月 13 日，取自 <https://www.hackster.io/powerberry/like-a-bat-with-hc-sr04-829486>
- 四、Arduino Uno 功能簡介(2015)。2019 年 3 月 13 日，取自 <https://openhome.cc/Gossip/Books/mBlockArduino1-3and1-4.html>

附錄

程式碼

```
1 #include <ld3320.h>
2 VoiceRecognition Voice;
3 int analogPinA = 10; //L9110S
4 int analogPinA1 = 3;
5 int analogPinB = 5;
6 int analogPinBl = 6;
7 int trigPin = 7; //Trig
8 int echoPin = 8; //Echo
9 long duration, cm;
10 void setup()
11 {
12   Serial.begin(9600);
13   pinMode(trigPin, OUTPUT);
14   pinMode(echoPin, INPUT);
15   pinMode(analogPinA, OUTPUT);
16   pinMode(analogPinA1, OUTPUT);
17   pinMode(analogPinB, OUTPUT);
18   pinMode(analogPinBl, OUTPUT);
19   Serial.print("start!");
20
21   Voice.init(); //Voice函示庫內部設定
22   Voice.noiseTime(0x20);
23   Voice.micVol(0x30);
24
25   Voice.addCommand("jin", 1); //前進
26   Voice.addCommand("fa", 2); //後退
27   Voice.addCommand("guan", 3); //左轉
28   Voice.addCommand("kai", 4); //右轉
29   Voice.addCommand("deng", 5); //停止
30   Voice.addCommand("wu", 6); //迴轉
31   Voice.start();
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77 void carforward() {
78   analogWrite(analogPinA, 255);
79   analogWrite(analogPinA1, 0);
80   analogWrite(analogPinB, 255);
81   analogWrite(analogPinBl, 0);
82 }
83 void carReversing() {
84   analogWrite(analogPinA, 0);
85   analogWrite(analogPinA1, 255);
86   analogWrite(analogPinB, 0);
87   analogWrite(analogPinBl, 255);
88 }
89 void carstop() {
90   analogWrite(analogPinA, 0);
91   analogWrite(analogPinA1, 0);
92   analogWrite(analogPinB, 0);
93   analogWrite(analogPinBl, 0);
94 }
95 void turnright() {
96   analogWrite(analogPinA, 255);
97   analogWrite(analogPinA1, 0);
98   analogWrite(analogPinB, 0);
99   analogWrite(analogPinBl, 255);
100 }
101 void turnleft() {
102   analogWrite(analogPinA, 0);
103   analogWrite(analogPinA1, 255);
104   analogWrite(analogPinB, 255);
105   analogWrite(analogPinBl, 0);
106 }
127 void towardleftgo() {
128   analogWrite(analogPinA, 150);
129   analogWrite(analogPinA1, 0);
130   analogWrite(analogPinB, 255);
131   analogWrite(analogPinBl, 0);
132 }
```

```

107 void turnright90() {
108   for (int i = 0; i < 251; i++) //控制轉向時間
109   {
110     Serial.println(i);
111     if (i < 250)
112       turnright();
113     else
114       carstop();
115   }
116 }
117 void turnleft90() {
118   for (int i = 0; i < 251; i++) //控制轉向時間
119   {
120     Serial.println(i);
121     if (i < 250)
122       turnleft();
123     else
124       carstop();

```

```

133 void detectiondistance() {
134   digitalWrite(trigPin, LOW);
135   delayMicroseconds(5);
136   digitalWrite(trigPin, HIGH); // 給 Trig 高電位，持續 10 微秒
137   delayMicroseconds(10);
138   digitalRead(echoPin); // 讀取 echoPin 的數值
139   duration = pulseIn(echoPin, HIGH); // 收到高電位時的時間
140   cm = (duration / 2) / 29.1; // 將時間換算成距離 cm
141   Serial.print("Distance : "); //將字串顯示於Serial Port
142   Serial.print(cm);
143   Serial.print("cm");
144   Serial.println();
145 }

```

```

33 void loop() {
34   switch (Voice.read())
35   {
36     case 1: //前進
37       Serial.println("前進");
38       while (Voice.read() == 1)
39       {
40         detectiondistance();
41         if (cm < 20) {
42           Serial.println("turnleftgo")
43           carstop();
44           delay(1000);
45           turnright90(); //
46           towardleftgo();
47         } else {
48           carforward();
49           Serial.println("carforward")

```

```

50     }
51   }
52   break;
53   case 2: //後退
54     Serial.println("後退");
55     carReversing();
56     break;
57   case 3: //左轉
58     Serial.println("左轉");
59     turnleft90();
60     break;
61   case 4: //右轉
62     Serial.println("右轉");
63     turnright90();
64     break;
65   case 5: //停止
66     Serial.println("停止");
67     carstop();
68     break;
69   case 6: //迴轉
70     Serial.println("迴轉");
71     turnright();
72     break;
73   default:
74     break;
75 }
76 }

```

