高雄市高英高級工商職業學校

Kao Ying Industrial Commercial Vocational High School

專題製作報告

無線控制自走車

學生姓名：周　顯　宗

楊　嘉　華

謝　明　翰

鍾　浩　威

指導老師：葉　忠　賢　老師

中　華　民　國　105　年 05 月

致 謝

感謝葉忠賢老師，在這高中的三年裡，不只教島我們在電腦上的技術，也教倒了我們許許多多的做人處事道理，方向來邁進，更謝謝老師的指導，讓我們可以重新發現自己長久以來的不足與缺點，並且在學習的過程中，與老師師生之間的感情更加深厚，這樣的感受，對我們小組而言，更是珍貴如珠，在我們不會的電路及沒有見過的零件老師都加以詳細告訴我們其中的功能，例如某些零件的腳位亦或者是程式碼上面出現的錯誤老師都會耐心的告訴我們其中的錯誤到哩，沒有排解出這一些困難這一個專題想必很難完成所以非常的感謝對我們的加以重視希望在未來的專題中還可以繼續幫我們找出困難排除因難。

周顯宗組長、謝明翰組員、楊嘉華組員、鍾浩威組員等 謹上 2016/05

無線控制自走車

摘 要

透過本篇研究 Arduino 微電腦的學習，深入了解 Arduino 使用方法及功能，且經由實際電路板的過程去對 Arduino 運作有更深入的了解。為何想製作無線控制自走車，是因為坊間有很多類似的相關產品，種類繁多，想要藉由製作一個 Aruino 微電腦搭配組合語言城市去執行將超音波感測器達到可以偵測到障礙物並迴避功能因為透過 Arduino 微電腦的輸出入設計及製作無線控制自走車的過程中，可以深入了解，如何透過程式組合語言去設計、控制超音波感測器，小組成員同學預期此設計能先達到對此知識的了解和樂趣，若此階段沒問題了，在會進階地針對無線控制自走車的缺點進行改進，就能使超音波感測來應用在各種方面的上，此為下一階段小組所預計達成的目標；故現行之小組專題製作的目標即是想透過 Arduino 微電腦的學習，去製作一個能方面快速地偵測地形的無線控制自走車，以達到科技與生活相結合的目的。

關鍵詞：Arduino 微電腦、超音波感測器、自走車、無線控制

目 錄

致謝 I

摘要 II

目錄 III

表目錄 IV

圖目錄 V

壹、前言 1

一、製作動機 1

二、製作目的 1

三、製作架構 1

四、製作預期成效 4

貳、理論探討 4

參、專題製作 15

一、設備及器材 15

二、製作方法與步驟 16

三、專題製作 17

肆、製作成果 23

伍、結論與建議 27

一、結論 27

二、建議 27

附錄一 參考文獻 28

無線控制自走車程式碼 29

表目錄

表 2-2-1： 基本電壓效應 8
表 2-2-2： 超音波測距模組 HC-SR04 規格.....9
表 2-2-3： L298P Shield 參數.....11
表 3-1-1： 專題製作使用儀器(軟體)設備一覽表.....14
表 3-3-1： 專題製作計畫書 16
表 3-3-2： 無線控制自走車之材料表 21

圖目錄

圖 1-3-1 超音波避障自走車流程圖 2
圖 2-1-1 HC-06 藍芽模組 4
圖 2-1-2 HC-06 藍芽模組的外觀與接腳 4
圖 2-2-1 超音波傳播過程 8
圖 2-2-2 超音波感測器 HC-SR04(實體) 9
圖 2-2-3 超音波測距模組 HC-SR04 偵測角度範圍 9
圖 2-2-4 L298P 模組 11
圖 2-2-5 L298 電路圖 11
圖 2-2-6 arduino Uno 12
圖 3-2-1 製作方法及步驟 15
圖 3-3-1 Arduino Uno 電路 18
圖 3-3-2 超音波感測器 HC-SR04 電路 19
圖 3-3-3 藍芽 HC-06 電路 20
圖 3-3-4 超音波感測原理作用圖 22
圖 4-1-1 組裝自走車 Part1 23
圖 4-1-2 組裝自走車 Part2 23
圖 4-1-3 藍牙信號配對 23
圖 4-1-4 Word 23
圖 4-1-5 Powerpoint 23
圖 4-1-6 藍牙超音波避障自走車成品圖 23
圖 4-1-7 藍牙超音波避障自走車成品圖(2) 24
圖 4-1-8 成品圖 24
圖 4-1-9 主從配對(一) 25
圖 4-1-10 主從配對(二) 25
圖 4-1-11 控制介面(一) 25
圖 4-1-12 控制介面(二) 25
圖 4-1-13 配對完成 25
圖 4-1-14 行駛中 25

圖 4-1-15 感測障礙物 26

壹、前言

一、製作動機

我們要製作的主題為自動避障自走車，坊間有很多類似的相關產品，種類繁多，像是有用光影控制動作的自走車、或是用溫度控制的、還有聲音控制的、還有會沿著預設的黑線走的循跡自走車之類等等…，而我們選擇是用聲波感應的，近年來，由於科技越來越發達，我們利用這次的專題製作，來探討如何將聲波感應原理應用在車子上，了解如何讓一輛自走車能自動閃避障礙物。

二、製作目的

這次專題報告的內容主要包括超音波感測原理、無線藍芽控制、接收器原理、感測器電路…等各種實際電路製作。我們目的要製作一台使用超音波感測避障並用藍芽控制的自走車，自走車可執行前進、後退、左轉、右轉、停止、避障、循跡等移動功能，自走車控制是由手機ＡＰＰ發送藍芽訊號來與自走車進行連接，電源方面是由外接直流電池（9V）供電，也可以由充電電池供電。

三、製作架構

（一） 製作流程說明

決定專題題目，分析自走車結構，找出需要的材料並購買，利用買來的材料組裝自走車，在將編寫好的程式寫入 Arduino Uno ，如測試失敗便檢查自走車組裝及程式編寫步驟，如無問題便完成。

（二） 製作流程圖

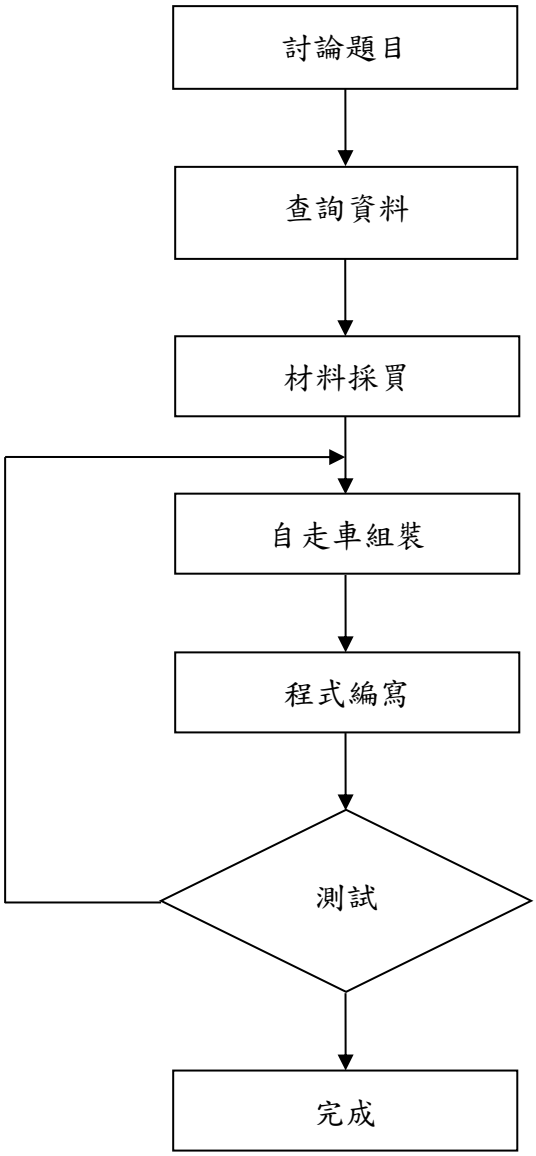


圖 1-3-1 超音波避障自走車流程圖

四、製作預期成效

1. 將一般市售遙控車加以改裝，利用智慧型手機的無線藍牙技術來遙控，捨棄傳統的遙控器來操作遙控車。
2. 透過手機操控，就像使用手機玩遊戲一般，只要動動手指觸控，就可以輕鬆的操控車體。
3. 裝設於車體前端的超音波感測器，讓感測器可以透過距離感測，若遭遇障礙物，可控制遙控車來自動停止並轉向，以避免碰撞障礙物。
4. 在車子行駛時遇到障礙物時，就不會有操控不當的問題發生，可以大幅降低遙控車因為碰撞而毀損的問題。
5. 在危險場所但又不得不進去調查內部情況時，可用無人機來讓操作者在安全的地方也能知道內部情況。

貳、理論探討

電子相關零組件

（一）藍牙 HC-06

市面上的藍牙裝置通常都屬於 slave（從端）設備，像藍牙滑鼠/鍵盤、藍牙 GPS、藍牙遙控玩具...等等。

一般的 Arduino 藍牙遙控和通訊實驗，也都是用「從端」模式，所以 HC-06 模組足以應付大多數需求。HC-06 和 HC-05 的主要差異在於主、從模式的設定，以及設置模組參數的 AT 命令的支援程度。HC-05 通常支援 30 多道 AT 命令，HC-06 只支援少數 AT 命令，但實務上會用到的 AT 命令通常也只有兩、三個。不管是 HC-05、HC-06 對 Arduino 都沒有影響，控制程式都一樣，實際接線也只用到 4 條線：電源、接地、傳送（TxD）和接收（RxD）。

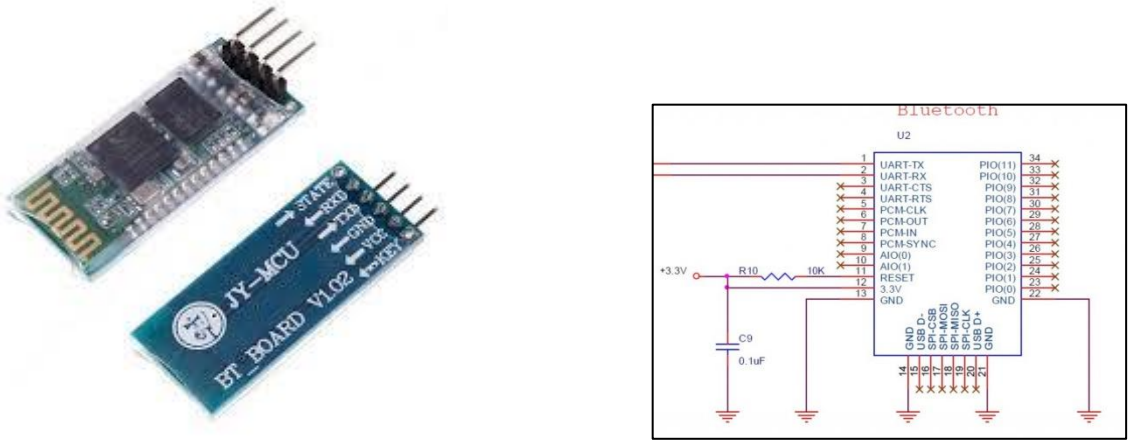


圖 2-1-1 HC06 藍芽模組

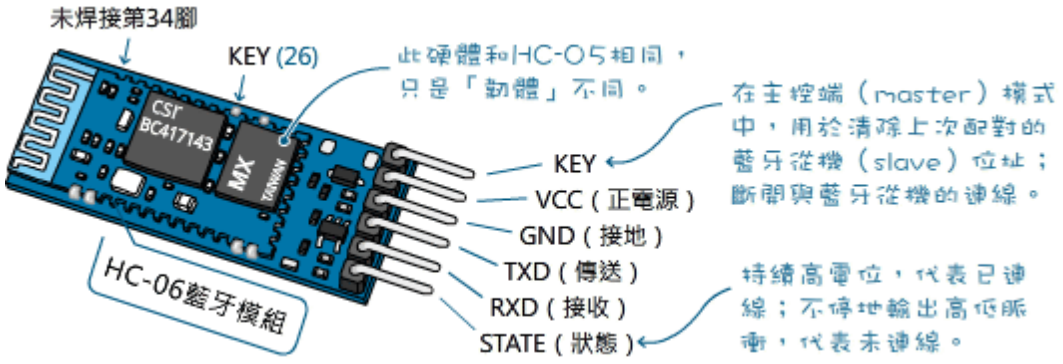


圖 2-1-2 HC-06 藍芽模組的外觀與接腳

藍芽是怎樣的短距無線通訊呢？現在我們從技術層面跟各位談藍芽通訊。

1.藍芽的運作原理-跳頻展頻技術

藍芽技術運作的原理主要是運用跳頻展頻技術(Frequency Hopping Spread Spectrum -- FHSS)方式，使藍芽晶片的兩端，以某一特定形式的窄頻載波同步地在 2.4MHz 頻帶上傳送訊號。詳細來說，此跳頻展頻技術(FHSS)的傳輸技術，是將欲傳輸的信號透過一系列不同的頻率範圍廣播出去，而由傳送裝置先傾聽頻道(Listens Channel)，若偵測出頻道處於閒置狀態時，信號便由此頻道傳送出去；反之，若偵測出頻道為使用中，便使用跳躍程序進行傳送。重要的是，傳送與接收必定要同步切換頻道才可以正常接收資料。藍芽最多可進行 1 對 7 的傳輸，除了資料之外，也可以使用連續變化斜率差量調變(Continuous Variable Slope Delta-Modulation ： CVSD)技術來進行語音傳輸及使用分時多工 (Time-Division Multiple Access ： TDMA)協定技術之通訊協定。在藍芽技術中，無線電的發射功率為 0dBm (Power Class 3)，傳輸範圍約為 10m。

跳頻技術(Frequency-Hopping Spread Spectrum；FHSS)是在 2.45GHz 頻帶以 1MHz 的頻寬將其劃分為 75-81 個無線電頻率通道（Radio Frequency Channel；RFC），並且以使用接收和發送兩端一樣的頻率跳躍模式（Frequency Hopping）來接發訊號及防止資料擷取（跳躍頻率的最大時間間隔為 250ms，也就是每秒跳頻至少 4 次）

分時多工，是利用時槽的概念，將時間切割頻道，即在一段時間內，通話採用某種頻率而在傳送送完封包後，即釋放該頻率給其他需要使用者被運用在 GSM，DCS，DECT，PDC，PHS，PACS 等系統。

藍芽計畫雖是 1998 年開始，但是藍芽的技術根基卻來自 1997 年制訂完成的無線區域網路通訊協定：IEEE-802.11。

藍芽基本上使用的頻帶範圍，是使用 2.45GHz，這個無線電頻帶是全世界共同開放、不受法令限制的頻帶，舉凡工業、科學、醫療（ISM，Industrial/Scientific/Medical）、甚至微波爐等都是使用 2.45GHz 的頻帶。

由於這個頻帶太被廣泛使用了！那麼使用此頻帶進行通訊，絕對是很容易收到干擾的，因此藍芽規格被設計成可跳頻通訊，能夠在一秒鐘內進行 1,600 次的跳頻動作，此這樣的動作避免其他通訊的干擾。

由於每秒 1,600 次的快速跳頻，這也使得藍芽無線收發的資料封包不能太長，否則不能滿足如此頻繁的跳頻次數，所以藍芽短封包、快速跳頻的特性，也使其無線傳輸能抗干擾、更穩定通信。

附註:跳頻技術曾是以往軍用通訊中,為防止無線電通信被監聽而創出。

2.藍芽技術選擇 2.45GHz 的理由

藍芽規格會選擇 2.45GHz 這麼常用的頻帶，然後用快速跳頻與短封包來補強，那為何不一開始就選擇一個少用的頻帶呢？

事實上 2.45GHz 頻帶被廣泛、甚至氾濫使用，但其優點是不用錢、不用額外申請,而且世界通用,事實上無線電頻帶各國的限制與開放程度都不同，有的頻帶在某些國家是開放,有些國家則是軍用,所以為了藍芽能世界通用，避免頻帶的額外付費與額外申請，藍芽仍選擇 2.45GHz，而不用多說，我們還用 900MHz、1,800MHz（1.8GHz）、以及 1,900MHz（1.9GHz）等頻帶，作為行動電話的服務之用。

3.藍芽特性

藍芽技術是一種小範圍的無線電頻率技術,裝置間透過晶片可互相溝通,不必再透過纜線傳輸。目前，低功率的無線傳輸仍以紅外線為主，應用的層面仍侷限在行動電話、PDA、電腦及其相關產品上。紅外線的傳輸的距離短、且受方向限制，用過筆記型電腦的紅外線埠和印表機連線的人可能有過這樣的經驗：努力地調整筆記型電腦和印表機的角度（幾乎是面對面）和位置（一公尺內）後，發現其慢無比，最後還是乖乖的找條線接上電腦，是不是資料的傳遞一定得要經過 銅線呢？

「藍芽」就是著眼於此，而開發出的傳輸技術，它的主要特點如下：

1. 操作頻段：2.4GHz 工業、科學、醫療頻段
2. 可及範圍十公分到一百公尺
3. 最高可支援八個連結埠
4. 可以超越紅外線傳輸埠的限制，需要傳輸端與接收端面對面一直線的傳輸支援同步和非同步傳輸模式，易與 TCP/IP 整合成網路。

如上所述，只要支援藍芽的設備，彼此間就可以互相交換資料，不管是皮包裡的行動電話、桌上的電腦、辦公室那一頭的印表機都可以互相溝通。

「藍芽」計畫在十到一百公尺範圍內建構起專屬個人的無線區域網路，最少可以容納八個小網路，對稱式傳輸時，上下傳速度均為 432Kbps，但採非對稱式時，上傳為 56Kbps、下傳則提高為 721Kbps，未來則可擴充為 2 或 5Mbps，可更為提高電腦及行動電話的通訊能力。另外，藍芽以更高的頻率（2.45GHz）取代紅外線，傳輸速率更高、應用更廣，可以符合未來寬頻網路時代的要求。

此外，跳頻展頻技術(FHHS)在同步且同時的情況下，發射與接收兩端以特定型式的窄頻電波來傳送訊號，為了避免在一特定頻段受其他雜訊干擾，收發兩端傳送資料經過一段極短的時間後，便同時切換到另一個頻段，由於不斷的切換頻段，因此較能減少在一個特定頻道受到的干擾，也不容易被竊聽。跳頻展頻(FHHS) 所展開的訊號，可依特別設計來規避雜訊或重覆的頻道，並且跳頻訊號必須遵守 FCC（Federal Communications Commission）的

要求，使用 75 個以上的跳頻訊號，且跳頻至下一個頻率的最大時間間隔為 400ms，在 IEEE 802・11 中最大時間間隔通常定為 250ms，也就是每秒跳頻 4 次。而且，藍芽所發出的信號需低於 10 pico 瓦，而 GSM 手機啟動時會產生一至三瓦不等的發射功率，相較之下安全 The Federal Communications Commission (FCC) is an independent agency of the United States government, created by Congressional statute ~ 129 ~ 許多。

藍芽技術的精髓，是在於讓使用者可以在公共場所利用行動電話、筆記型電腦或 PDA 等個人行動終端器材取得資訊服務。舉例而言，當消費者進到車站、機場或購物中心等場所時，可以將手上的行動終端設備與這些場所的藍芽接取系統 連結，取得鄰近或遠端的服務。

（二）超音波感測原理

許多測距的工具如紅外線、超音波、電磁波...等，是常見的方法，但是我們最常見到的方法似乎是超音波。如圖 2-2-1 所示，首先透過驅動觸發線路，將聲波在透過發射器將聲波直線發射出去，在其傳播過程之中，遇到阻礙,則回反射或折射,這時候透過聲波接收感測器來接收反射或折射的聲波，進而將發射聲波的時間與接收的時間相減後，透過聲波的速度(一般就是音速),則可以反求得距離,再透過資料輸出線路將距離資料用脈衝(Pulses)的方式輸出

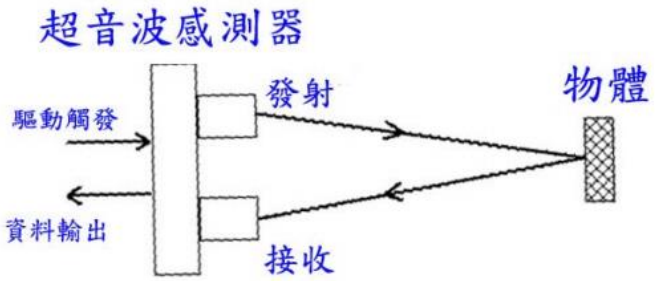
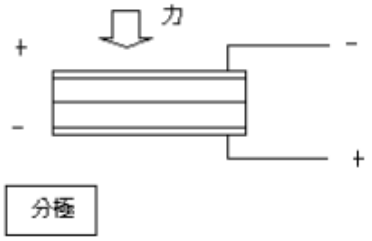
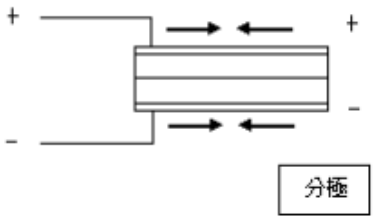


圖 2-2-1 超音波傳播過程

超音波屬於一種非接觸型的感測元件。超音波感測器分為發射器和接收器，兩者同樣具有「壓電效應」，對超音波發射器而言稱為「壓電效應」，主要是將電氣信號轉換成壓力的信號去壓縮周圍空氣；而這些受壓縮的空氣將會壓縮接收器上的壓電材料，並產生「壓電效應」，進而將壓力信號轉換成電的信號。

表 2-2-1 基本電壓效應

	壓電效應	壓電逆效應
說明圖		
特性	壓電振盪元件的分極方向，由力與元件端子的出力極性來決定。	壓電振盪元件的分極方向和輸入電壓的極性有關，而振盪元件的伸縮方向由外加電壓來決定。
用途	(1)壓電振盪元件 (2)超音波偵測器	(1)壓電振盪用 (2)高頻用振盪元件 (3)超音波發生器

本專題使用 HC-SR04，是非常普遍用於超音波測距的模組，其圖 2-2-2、圖 2-2-3、表 2-2-2 為 HC-SR04 相關資料。

圖 2-2-2 超音波感測器 HC-SR04(實體)

表 2-2-2 超音波測距模組 HC-SR04 規格

電氣特性	規格
電壓	5v
電流	15mA Typ. 50mA Max
工作頻率	40kHz
可量測最長距離	4.5m
可量測最短距離	2cm
驅動訊號	10uS TTL level pulse
回應脈波	Input TTL lever sigal and the range in pro-portion
尺寸	45*20*15mm

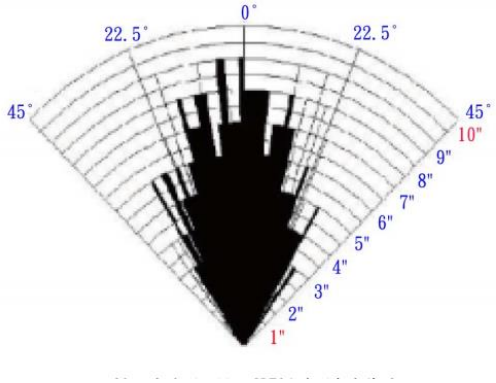


圖 2-2-3 超音波測距模組 HC-SR04 偵測角度範圍

由於超音波穿透性與準確性，所以超音波在非破壞檢測技術的測量應用上，運用非常廣泛，已下為超音波常見的應用與特性。

可使用超音波檢查之用途特性

1. 確定物料內部或表面缺疵、中斷、不均一等等。
2. 測定物料厚度。

使用超音波之特性

1. 可穿透甚長〈20 呎〉的物料內部距離。
2. 行程為一有界範疇。
3. 物料中有夾層、中斷、或為兩種不同性質的物料界限時，即行折回。
4. 由兩種不同性質物料界限折回時，聲波振動形態及生變化。
5. 靈敏、準確。

超音波檢測法的優缺點：

一、優點：

1. 可使用於多數材料
2. 可作深入內部之探傷
3. 測厚及測定物性
4. 檢測速度較快並準確
5. 輕便，並可保有永久紀錄

二、缺點：

1. 裝備昂貴
2. 操作、判讀較難，需要經驗豐富人員
3. 易受料件形態、結構、表面粗糙度之限制

298P Shield：

298P Shield 直流電機驅動器採用 LGS 公司優秀大功率電機專用區動芯片 L298P，可直接驅動 2 個直流電機，驅動電流達 2A，電機輸出端採用 8 隻高速蕭特基二極體作為保護。該電路線路布線合理、均採用貼元件片，疊層設計可直接插接到 Arduino 上。

L298P Shield 直流電機驅動器具有 PWM 調速馬達。電機供電可使用 Arduino VIN 輸入或驅動器上的接線柱輸入，使用跳線切換。

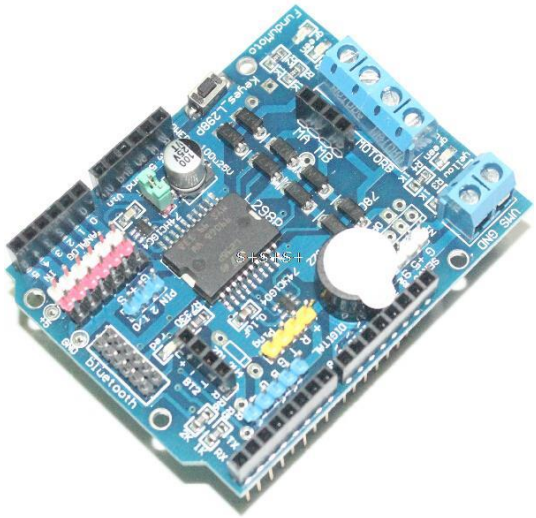


圖 2-2-4 L298P 模組

表 2-2-3 L298P Shield 參數

邏輯部分輸入電壓 VD	5v
驅動步分輸入電壓 VS	VIN 輸入 6.5~12v，PWRIN 輸入 4.8~24v
邏輯部分工作電流 Iss	≦36mA
驅動部分工作電流 Io	≦2A
最大消耗功率	25w（T=75℃）
控制信號輸入電平	高電平：2.3v≦Vin≦5v 低電平：-0.3≦Vin≦1.5v
工作溫度	-25℃～+130℃
驅動型式	雙路大功率 H 橋驅動

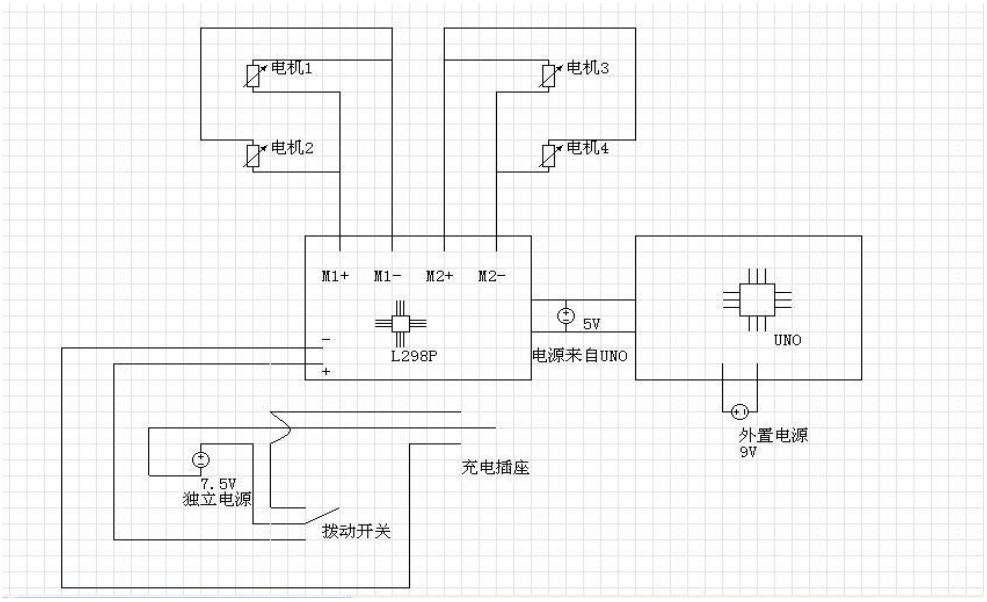


圖 2-2-5 L298P 電路圖

Arduino

『Arduino® 』,是一個開放原始碼的單晶片控制器,它使用了 Atmel AVR 單晶片(Atmel_Corporation, 2013)，採用了基於開放原始碼的軟硬體平台，構建於開放原始碼 Simple I/O 介面版，並且具有使用類似 Java，C 語言的 Processing4/Wiring 開發環境(B. F. a. C. Reas，2013; C. Reas & Fry, 2007, 2010)。Processing 由 MIT 媒體實驗室美學與計算小組(Aesthetics & Computation Group)的 Ben Fry(http：//benfry.com/)和 Casey Reas 發明，Processing 已經有許多的 Open Source 的社群所提倡，對資訊科技的發展是一個非常大的貢獻。

Arduino Uno 硬體介紹

使用 ATMega 8U2 來當作 USB-對序列通訊，並多了一組 ICSP 給 MEGA8U2 使用：未來使用者可以自行撰寫內部的程式~也因為捨棄 FTDI

USB 晶片~Arduino 開發板需要多一顆穩壓 IC 來提供 3.3V 的電源。

控制器核心：ATmega328

控制電壓：5V

建議輸入電(recommended)：7-12 V

最大輸入電壓 (limits)：6-20 V

數位 I/O Pins：14 (of which 6 provide PWM output)

類比輸入 Pins：6 組 DC Current per I/O Pin：40 mA

DC Current for 3.3V Pin：50 mA

Flash Memory：32 KB (of which 0.5 KB used by bootloader)

SRAM：2 KB

EEPROM：1 KB

Clock Speed：16 MHz

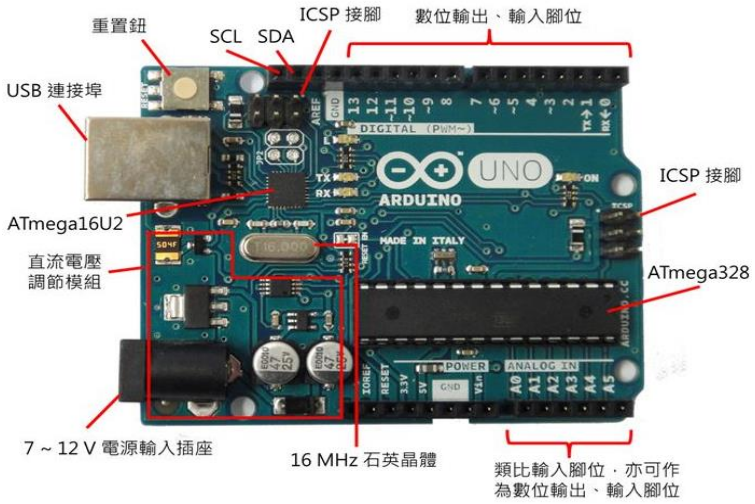


圖 2-2-6 arduino Uno

Arduino 特色

- 開放原始碼的電路圖設計，程式開發介面
- http：//www.arduino.cc/免費下載，也可依需求自己修改!!
- Arduino 可使用 ISCP 線上燒入器，自我將新的 IC 晶片燒入「bootloader(http：//arduino.cc/en/Hacking/Bootloader?from=Main.Bootloader)。
- 可依據官方電路圖(http：//www.arduino.cc/)，簡化 Arduino 模組，完 成獨立運作的微處理控制
- 感測器可簡單連接各式各樣的電子元件(紅外線,超音波,熱敏電阻，光敏電阻,伺服馬達,...等)
- 支援多樣的互動程式
- 使用低價格的微處理控制器(ATMEGA8-16)
- USB 介面，不需外接電源。另外有提供 9VDC 輸入
- 應用方面，利用 Arduino，突破以往只能使用滑鼠，鍵盤，CCD 等輸入的

裝置的互動內容，可以更簡單地達成單人或多人遊戲互動。

參、專題製作

一、設備及材料

表 3-1-1 專題製作使用儀器(軟體)設備一覽表

儀器（軟體） 設 備 名 稱	應 用 說 明
pc 電腦	編譯程式碼
手機	與 Arduino 的藍芽裝置作連結並傳送訊息
Word	書面報告
Power point	上台報告之簡報
程式編輯器	編好之軟體除錯即轉燒錄檔
Arduino	編譯程式碼
電源供應器	提供電源
三用電表	量測電路
藍芽 HC-06	主從配對
藍芽 APP	配對軟體

二、製作方法與製作步驟

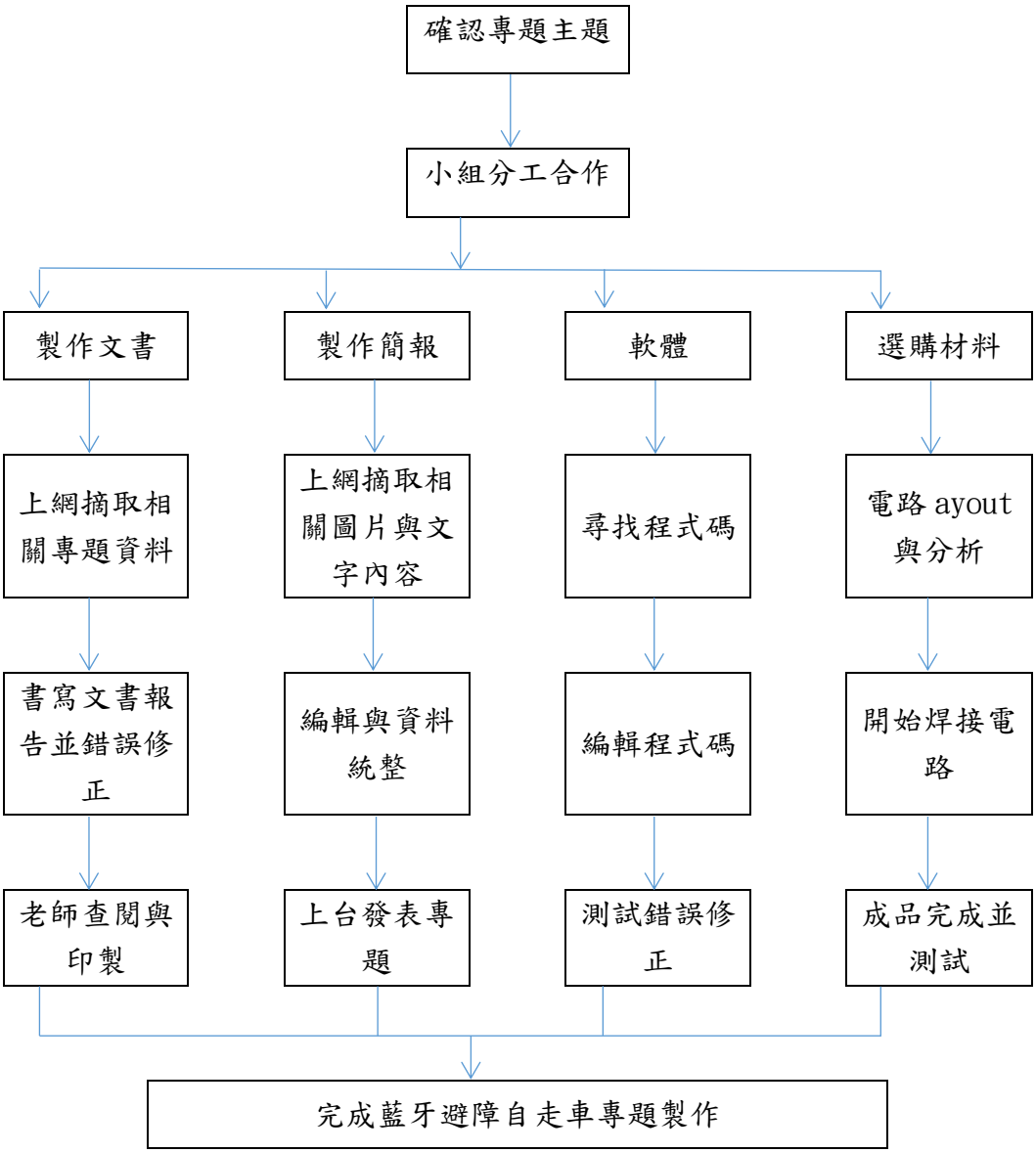


圖 3-2-1 製作方法及步驟

三、專題製作

表 3-3-1 專題製作計畫書

<table>
<tr><td colspan="2">專題型別</td><td colspan="2">□ 個人型專題 ▌團隊型專題</td></tr>
<tr><td colspan="2">專題性質</td><td colspan="2">Arduino 控制研究</td></tr>
<tr><td colspan="2">科別/年級</td><td colspan="2">資訊科三年級</td></tr>
<tr><td rowspan="2">專題名稱</td><td>中文名稱</td><td colspan="2">無線控制自走車</td></tr>
<tr><td>英文名稱</td><td colspan="2">Wireless control self-propelled vehicle</td></tr>
<tr><td colspan="2">專題製作簡述</td><td colspan="2">透過 Arduino 微電腦學習控制硬體了解其控制的使用、以及運用 Arduino 設計一個超音波感測的專題，我們選擇了解運用超音波感測的方式使無形聲波接觸到物體得以確認該有此物體、應用在生活中、使生活更加便利，學到新知識，達到確認物體所在處的效果，並能用在探測上。</td></tr>
<tr><td colspan="2">指導老師姓名</td><td colspan="2">葉忠賢　老師</td></tr>
<tr><td colspan="2" rowspan="2">參與同學姓名</td><td>周顯宗組長</td><td>楊嘉華組員</td></tr>
<tr><td>鍾浩威組員</td><td>謝明翰組員</td></tr>
<tr><td colspan="2">專題執行日期</td><td colspan="2">１０４ 年 ９ 月 １ 日至 １０５ 年 ５ 月 ３１ 日</td></tr>
</table>

超音波感測器

透過這個專題製作，去了解如何運用 Arduino 在超音波自動控制上，當在使用超音波感測器時，能進行附近感測，且經由透過程式碼，來更改超音波距離，使超音波感測的表示方法：剛開始超音波感測器送電時，當使用者按下按鈕其自動模式，自走車會自動位移如果遇到障礙物則會自動感測進行避障。

本專題的內容由 Arduino 為基礎做輸出入架構，其中功能、設定超音波感測設定電源開關啟動時，首先我們專題所用的手機要連線至藍芽做配對。透過手機的操作發送其指令，讓自走車移動並使用超音波感測避障。

藍芽超音波避障自走車理論操作：

(1)透過超音波感測的運用，了解其感測器的偵測範圍。

(2)透過 Arduino 輸出入功能。該如何運作;程式如何進行更改方法。

硬體電路圖：Arduino 與超音波感測、藍芽電路圖：

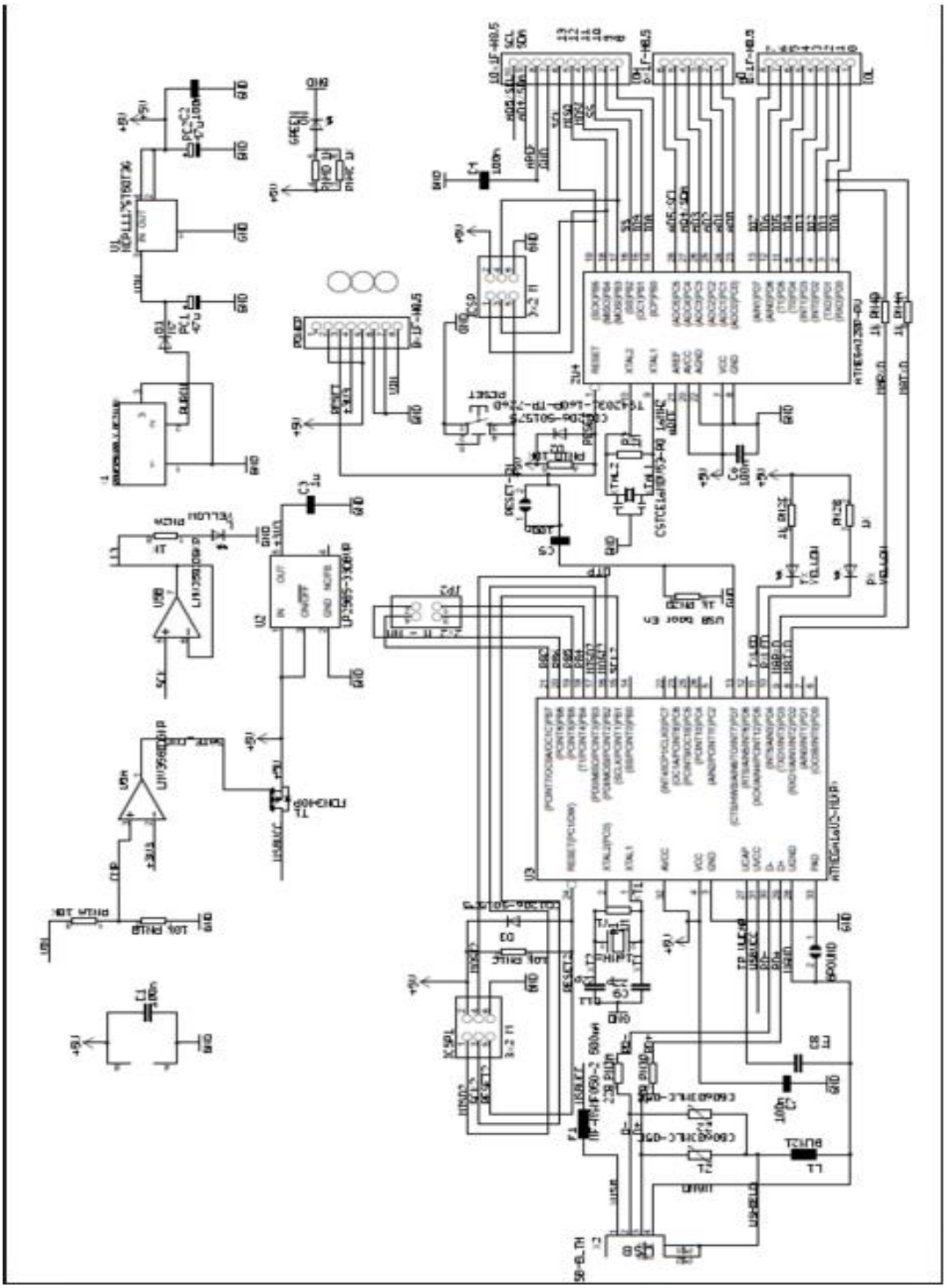


圖 3-3-1 Arduino Uno 電路

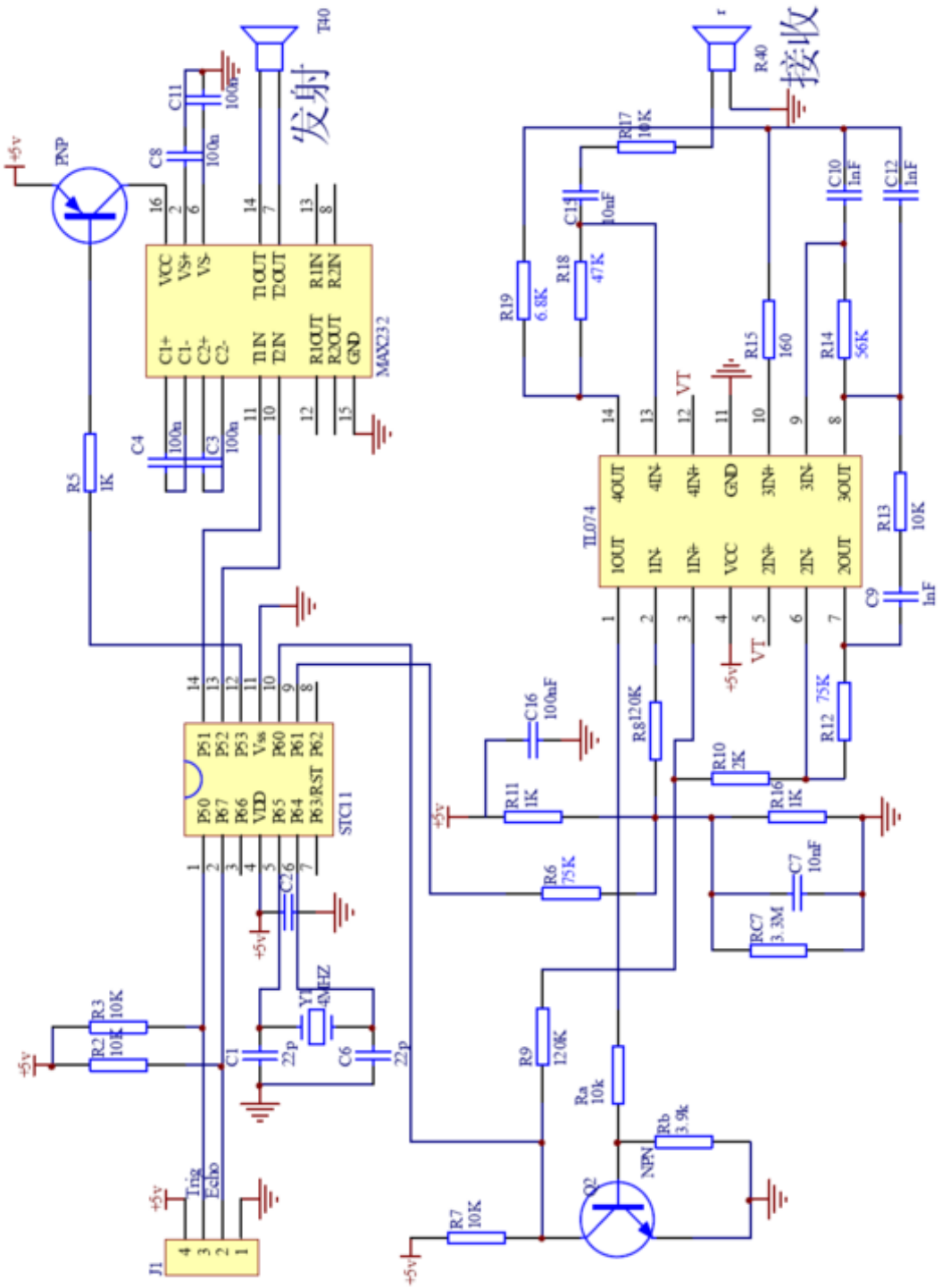


圖 3-3-2 超音波感測器 HC-SR04 電路

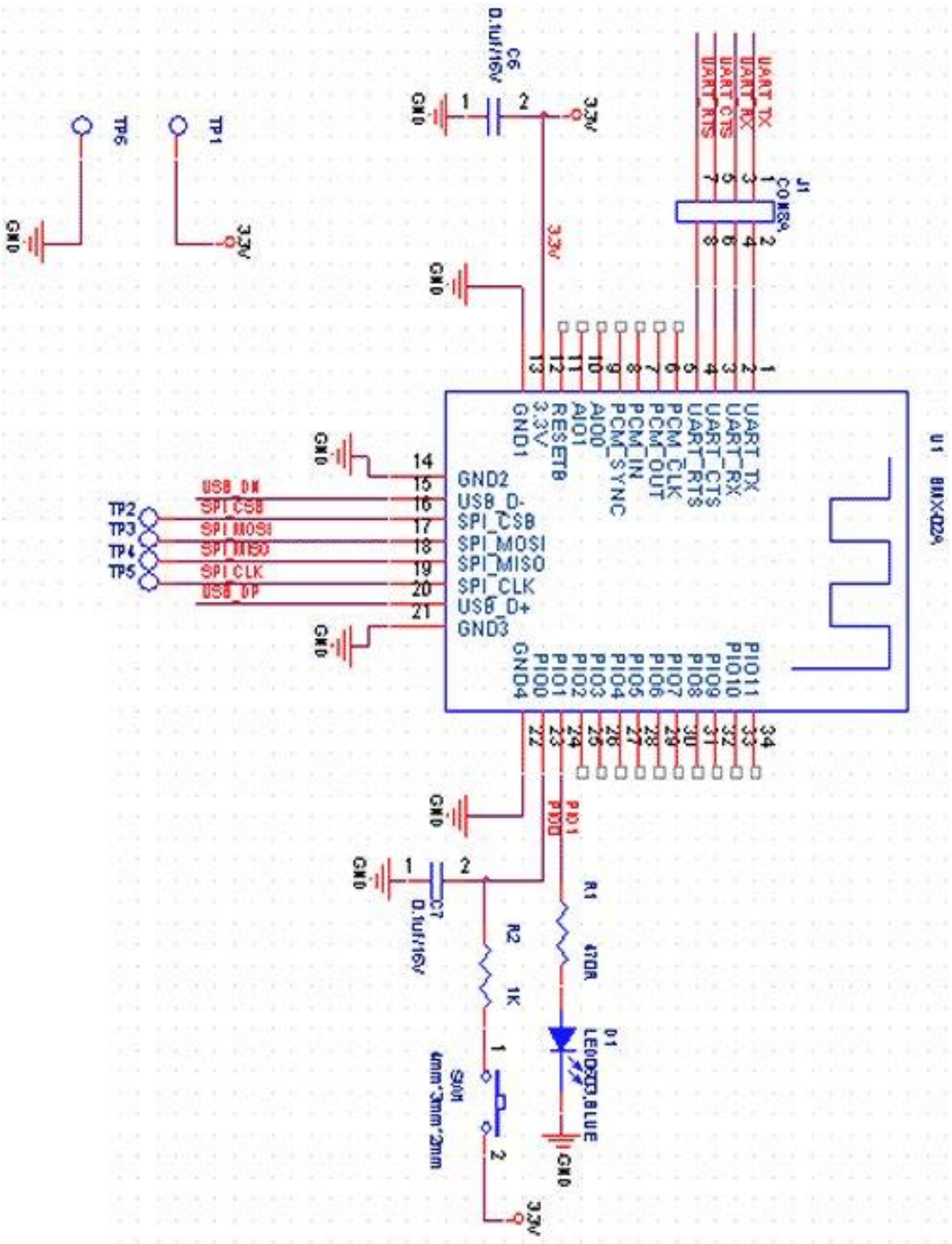


圖 3-3-3 藍芽 HC-06 電路

表 3-3-2 無線控制自走車之材料表

編號	零件名稱	數量
1	藍芽模組	1
2	超音波感測器	1
3	Arduino Uno 模組	1
4	L298P Shield 模組	1
5	Arduino 循跡感測模組	1
6	直流馬達	1
7	杜邦線	17
8	伺服馬達	1
9	電壓表	1
10	輪胎	4
11	底板	1
12	9V 電池	1
13	撥桿開關	1
14	輕觸開關	1
15	Bh-9vpc	1

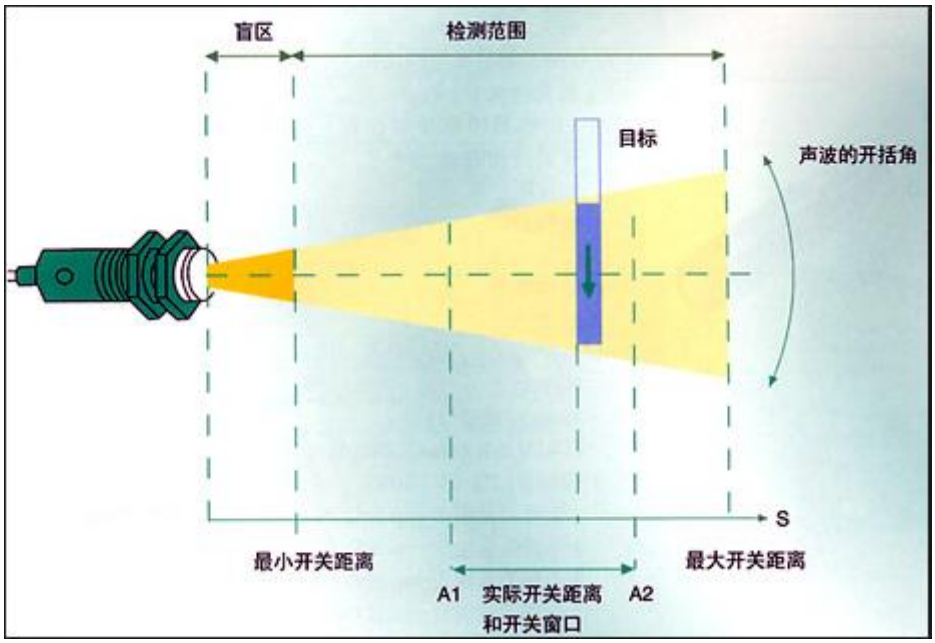


圖 3-3-4 超音波感測原理作用圖

這次的專題由顯宗來去分配專題工作，雖然第一次製作專題，但我跟我的組員，盡力去了解超音波感測器的功能及原理，來完成這次的專題，為何要了解超音波感測，因為要讓我們的自走車擁有自動避障的功能來防止撞到障礙物，為了讓自走車能夠順利地行走在安全的路上，才會在自走車上面設置超音波感測器，進而探測行駛路上的障礙物。

小組分工的配置：

顯宗負責找資料，及製作文書軟體內容，過程中亦有上網查詢相關專題資料，選擇要如何去製作專題，嘉華、明漢、浩威都對這次專題有參與配合，浩威是要上網查有關專題相關的資料，負責把相關的資料和圖片下載下來，再去製作簡報的作業，如有發生困難時可以找小組討論或者找老師諮詢，解決你所不懂的事情。

嘉華和明漢，明漢是選購材料、模組並去組裝自走車，並檢測模組是否正常，再把相關程式與電路燒錄製光碟片，嘉華負責了解如何運用 Arduino 去控制程式，讓模組能夠偵測運行，而去分析程式，並編輯程式。

肆、製作成果

我們小組決定題目是藍芽超音波避障自走車，開始有去思考專題功能並測試功能及利用專題光碟片去欣賞影片，把它的功能運用在專題中;這整個流程中，我們小組都用相機及相關電腦設備將之拍照下來，經將這些資料完整處理過後，我們呈現在這個專題報告之中，如下所示：

圖 4-1-1 組裝自走車 Part1

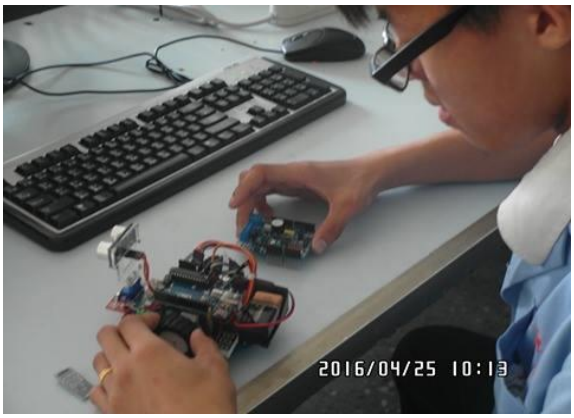


圖 4-1-2 組裝自走車 Part2

圖 4-1-3 藍牙信號配對

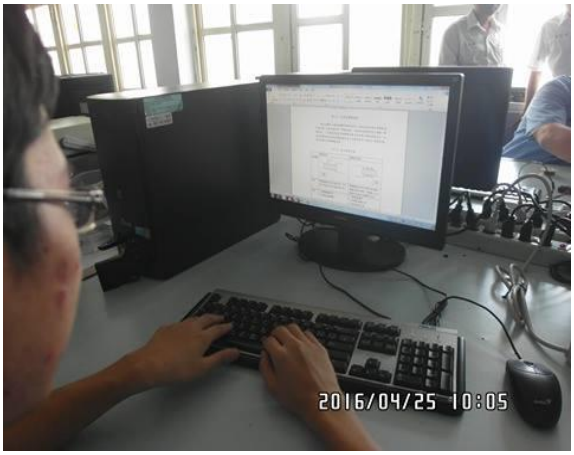


圖 4-1-4 Word

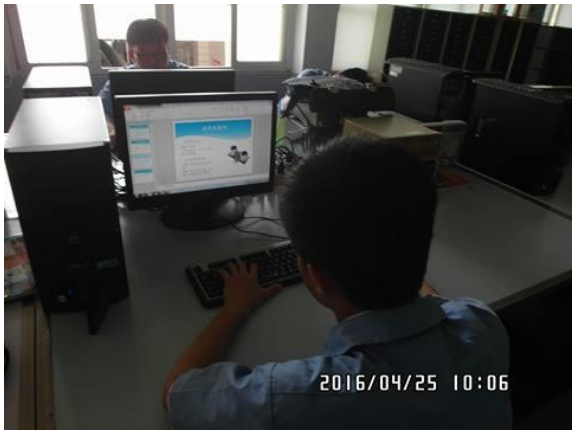


圖 4-1-5 Powerpoint

圖 4-1-6 藍牙超音波避障自走車成品圖

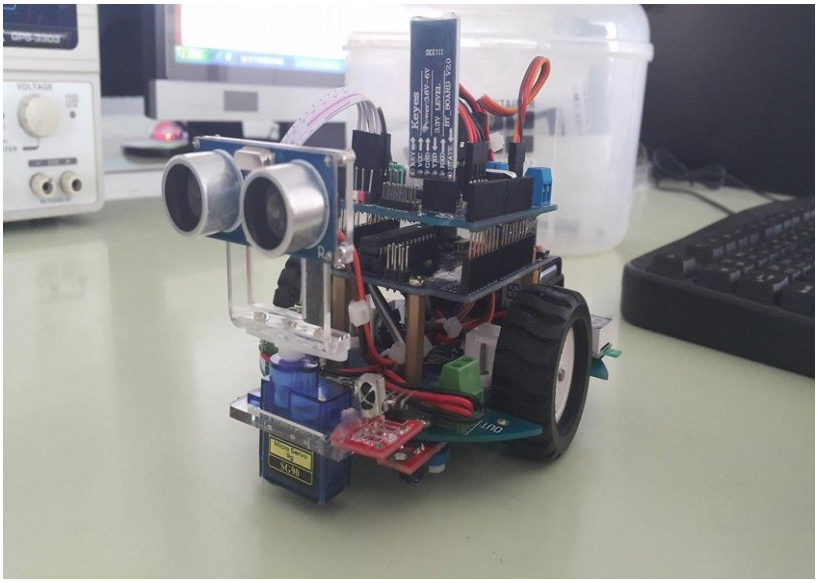


圖 4-1-7 藍牙超音波避障自走車成品圖(2)

操作功能說明：

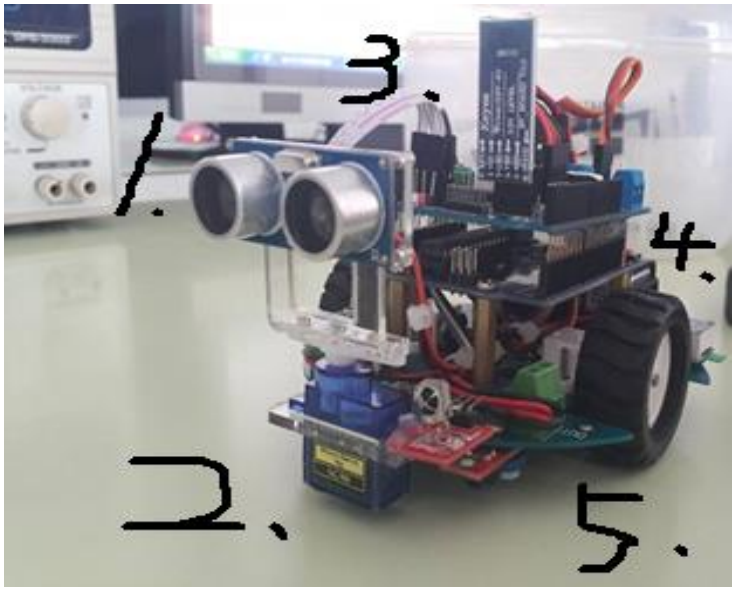


圖 4-1-8 成品圖

1.超音波感測器：感測障礙物

2.伺服馬達：增加感測範圍

3.藍芽 HC-06:主從配對

4.Arduino：微控制器

5.直流馬達：以一定角度逐步轉動的馬達

操作步驟：

圖 4-1-9 主從配對(一)

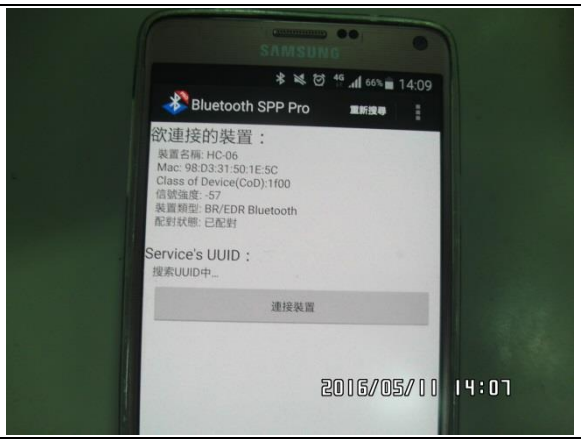


圖 4-1-10 主從配對(二)

圖 4-1-11 控制介面(一)

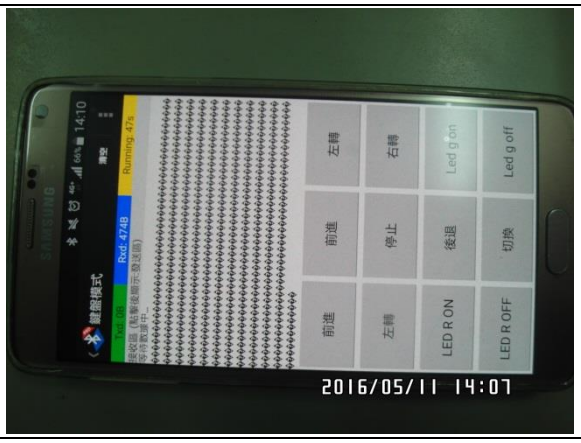


圖 4-1-12 控制介面(二)

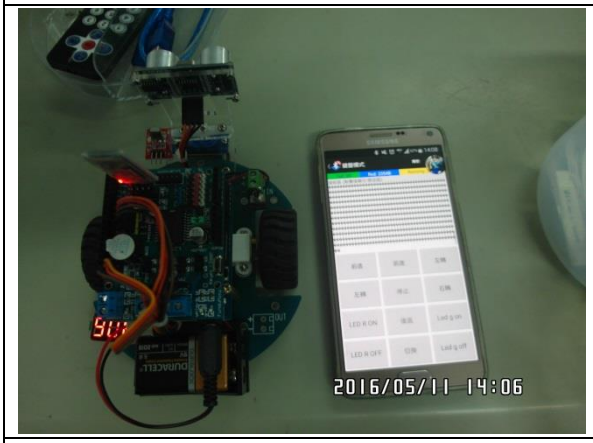


圖 4-1-13 配對完成

圖 4-1-14 行駛中

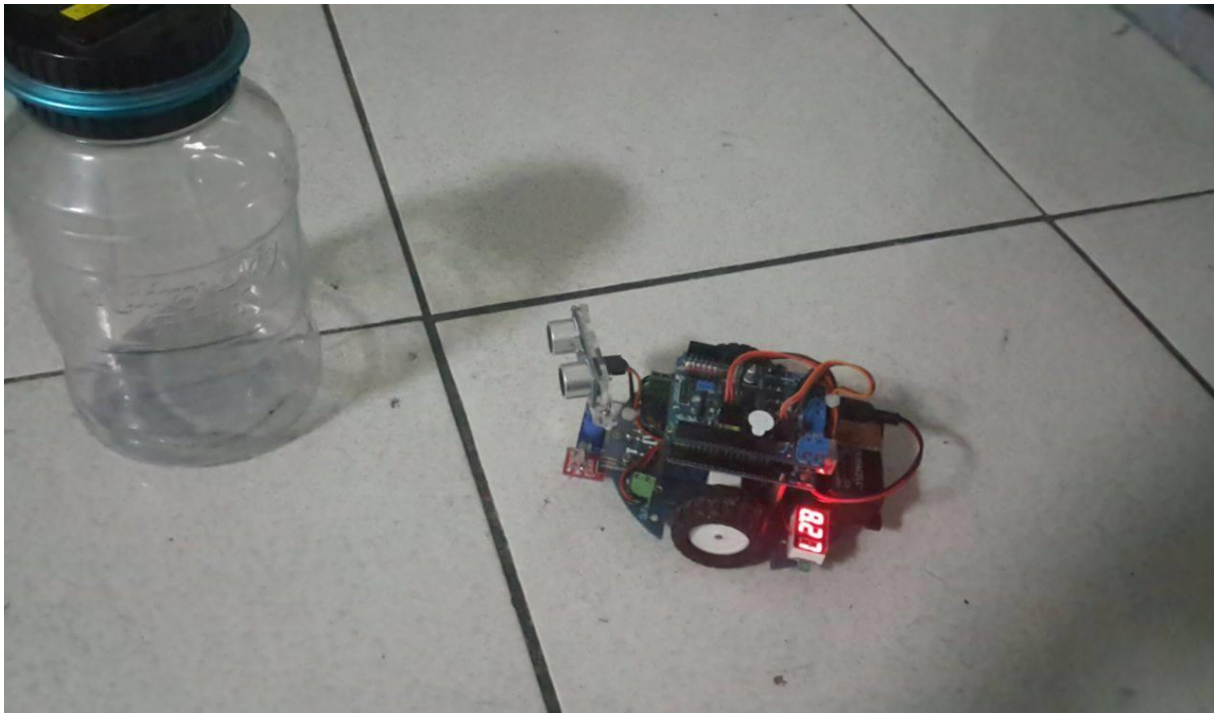


圖 4-1-15 感測障礙物

目前為止，我們所製作的電路實作部分，面對我們所遭遇的困難以及問題，都已盡力完成，緊接著我們的下一步驟即是將我們開始製作外面包裝的成品製作，目的就是要我們瞭解如何運用 Arduino 去控制程式使電路達到感測週遭的障礙物，並迴避至安全路線。

伍、結論與建議

本次單元使我們將對整體我們小組來說是一個完整專題製作的學習過程，能對 Arduino 微電腦控制電路運行有深入瞭解。

(一)結論

本專題涵蓋了電子電路、感測器、機構等知識。從一開始的構思找資料、麵 包板實驗到洗板子測試。其中遇到許多挫折與問題，如感測器的位置放那最適合， 電力系統的選擇，還好經過多次的測試，找出問題所在並解決使得此專題得以完成。

至於其他的功能如蜂鳴器，手控等其實都沒遇到多大的問題。唯一的問題是當結合馬達和感測器的程式時遇到了一些 BUG 的問題。不過經過慢慢的除錯之後就 沒什麼問題了。

自走車包含電源電路、感測電路、馬達電路、Arduino 電路、等四大電路部分，動作原理是藉由超音波模組來偵測有無障礙物的反射訊號。超音波能在惡劣之環境中，但費成本較高，且因為原理之不同，其感測範圍不如紅外線感測器來得大。為了彌補超音波之缺點，可用紅外線不易對人體溫度與周圍環境溫度產生的電荷現象變化的特性，提出紅外線感測器，以不同類型的感測器，進行交互感測來提 升整體的感測良率。

(二)建議

自走車防撞設計對人類生活上都有很大幫助，本論文加上晶片控制，讓自走車的自主性控制得以更靈敏，可讓自走車自動避障，且自動校正繼續循跡行駛，讓自走車可以在行駛中自我避開危險，真正落實於實務系統中，若能引入汽車行駛中，相信對汽車的操控性安全駕駛有相當的助益。汽車業的多功能性一直提升，從傳統的機械變速一直到現在的電子式控制變速，顯示科技的進步與改善；但是汽車行駛的安全性一直是重要課題，汽車在高速行駛下只要稍不留神，會造成相當大的傷亡，因此若能將本研究觀念落實於汽車行駛控制中，對駕駛或其他乘客有相當的保障，且再加入定速器的功能，對目前的交通事故會有相當大的改善。

附錄一 參 考 文 獻

1.趙英傑、2014，超圖解 Arduino 互動設計入門(第二版)，旗標

2.陳明熒、2016，Arduino 語音互動專題製作與應用，松崗

3.鐘國家、侯安桑、廖忠興，2011，感測器原理與應用實習(第二版)，全華

4.施士文，2015，Arduino 微電腦應用實習(含 AMA 中級先進微控制器應用認證術科試題)、台科大

5.金鈍，許光辰，2002，藍芽技術，五南

6.旗威科技有限公司，2016，無刷直流 BLDC 馬達控制實務： 使用 Atmel SAM C21 ARM Cortex：M0+ 控制核心，旗標

7.楊明豐，2016，Arduino 自走車最佳入門與應用：打造輪型機器人輕鬆學 ，碁峰資訊

8.位明先，2015，Arduino 微電腦專題製作-邁向 AMA Essentials Level 先進微控制器應用認證 ，台科大

9.鄒宏基，2014，自動控制系統 ，國家

10.徐國凱、韓志敏、張濤，2007，自動控制原理，清大出版社

無線控制自走車程式碼

```
#include <Servo.h>
//Arduino 蓝牙遥控可编程智能车程序：
//*******************************
int pinLB=13;         // 定義 8 腳位 左後
int pinLF=11;         // 定義 9 腳位 左前
int pinRB=12;        // 定義 10 腳位 右後
int pinRF=10;        // 定義 11 腳位 右前

int inputPin = 7;   // 定義超音波信號接收腳位
int outputPin = 8;   // 定義超音波信號發射腳位

int Fspeedd = 0;          // 前速
int Rspeedd = 0;          // 右速
int Lspeedd = 0;          // 左速
int directionn = 0;     // 前=8 後=2 左=4 右=6

Servo myservo;              // 設 myservo
int delay_time = 250; // 伺服馬達轉向後的穩定時間

int Fgo = 8;               // 前進
int Rgo = 6;               // 右轉
int Lgo = 4;               // 左轉
int Bgo = 2;               // 倒車


void setup()
{
  Serial.begin(9600);
  pinMode(pinLB,OUTPUT); // 腳位 8 (PWM)
  pinMode(pinLF,OUTPUT); // 腳位 9 (PWM)
  pinMode(pinRB,OUTPUT); // 腳位 10 (PWM)
  pinMode(pinRF,OUTPUT); // 腳位 11 (PWM)

  pinMode(inputPin, INPUT);      // 定義超音波輸入腳位
  pinMode(outputPin, OUTPUT);   // 定義超音波輸出腳位
```

```
  myservo.attach(9);       // 定義伺服馬達輸出第 9 腳位(PWM)
}

void advance(int a)// 前進
{
      digitalWrite(pinRB,HIGH);   //使馬達（右後）動作
      digitalWrite(pinRF,HIGH);
      digitalWrite(pinLB,HIGH);   //使馬達（左後）動作
      digitalWrite(pinLF,HIGH);
      delay(a * 100);
}

void left(int c) //   左轉
{
      digitalWrite(pinRB,HIGH);
      digitalWrite(pinRF,LOW);
      digitalWrite(pinLB,HIGH);
      digitalWrite(pinLF,HIGH);
      delay(c * 100);
}
void right(int b) //   右轉
{
      digitalWrite(pinRB,HIGH);
      digitalWrite(pinRF,HIGH);
      digitalWrite(pinLB,HIGH);
      digitalWrite(pinLF,LOW);
      delay(b * 100);
}
void stopp(int f) //停止
{
      digitalWrite(pinRB,HIGH);
      digitalWrite(pinRF,LOW);
      digitalWrite(pinLB,HIGH);
      digitalWrite(pinLF,LOW);
      delay(f * 100);
}
void back(int g) //後退
{
```

```
        digitalWrite(pinRB,LOW);   //使馬達（右後）動作
        digitalWrite(pinRF,HIGH);
        digitalWrite(pinLB,LOW);   //使馬達（左後）動作
        digitalWrite(pinLF,HIGH);
        delay(g * 100);
}
void turnR(int d)             //右轉(雙輪)
      {
        digitalWrite(pinRB,LOW);   //使馬達（右後）動作
        digitalWrite(pinRF,HIGH);
        digitalWrite(pinLB,HIGH);
        digitalWrite(pinLF,HIGH);   //使馬達（左前）動作
        delay(d * 100);
      }
void turnL(int e)             //左轉(雙輪)
      {
        digitalWrite(pinRB,HIGH);
        digitalWrite(pinRF,HIGH);    //使馬達（右前）動作
        digitalWrite(pinLB,LOW);    //使馬達（左後）動作
        digitalWrite(pinLF,HIGH);
        delay(e * 100);
      }
void detection()              //測量 3 個角度(0.90.179)
      {
        int delay_time = 250;    // 伺服馬達轉向後的穩定時間
        ask_pin_F();                 // 讀取前方距離

       if(Fspeedd < 10)             // 假如前方距離小於 10 公分
        {
        stopp(1);                    // 清除輸出資料
        back(2);                     // 後退 0.2 秒
        }

        if(Fspeedd < 25)             // 假如前方距離小於 25 公分
        {
          stopp(1);                  // 清除輸出資料
          ask_pin_L();               // 讀取左方距離
          delay(delay_time);         // 等待伺服馬達穩定
```

```
        ask_pin_R();                  // 讀取右方距離
        delay(delay_time);          // 等待伺服馬達穩定

        if(Lspeedd > Rspeedd)     //假如 左邊距離大於右邊距離
        {
         directionn = Rgo;          //向右走
        }

        if(Lspeedd <= Rspeedd)     //假如 左邊距離小於或等於右邊距離
        {
         directionn = Lgo;          //向左走
        }

        if (Lspeedd < 10 && Rspeedd < 10)     //假如 左邊距離和右邊距離皆小於 10 公
分
        {
         directionn = Bgo;          //向後走
        }
      }
      else                              //加如前方不小於(大於)25 公分
      {
        directionn = Fgo;             //向前走
      }

    }
void ask_pin_F()     // 量出前方距離
    {
      myservo.write(90);
      digitalWrite(outputPin, LOW);     // 讓超聲波發射低電壓 2μs
      delayMicroseconds(2);
      digitalWrite(outputPin, HIGH);   // 讓超聲波發射高電壓 10μs，這裡至少是 10μs
      delayMicroseconds(10);
      digitalWrite(outputPin, LOW);      // 維持超聲波發射低電壓
      float Fdistance = pulseIn(inputPin, HIGH);   // 讀差相差時間
      Fdistance= Fdistance/58;            // 將時間轉為距離距离（單位：公分）
       Serial.println(Fdistance);
      Fspeedd = Fdistance;                    // 將距離 讀入 Fspeedd(前速)
    }
```

```
 void ask_pin_L()     // 量出左邊距離
     {
        myservo.write(5);
        delay(delay_time);
        digitalWrite(outputPin, LOW);     // 讓超聲波發射低電壓 2μs
        delayMicroseconds(2);
        digitalWrite(outputPin, HIGH);   // 讓超聲波發射高電壓 10μs，這裡至少是 10μs
        delayMicroseconds(10);
        digitalWrite(outputPin, LOW);      // 維持超聲波發射低電壓
        float Ldistance = pulseIn(inputPin, HIGH);   // 讀差相差時間
        Ldistance= Ldistance/58;            // 將時間轉為距離距离（單位：公分）
         Serial.println(Ldistance);
        Lspeedd = Ldistance;                      // 將距離 讀入 Lspeedd(左速)
     }
void ask_pin_R()     // 量出右邊距離
     {
        myservo.write(177);
        delay(delay_time);
        digitalWrite(outputPin, LOW);     // 讓超聲波發射低電壓 2μs
        delayMicroseconds(2);
        digitalWrite(outputPin, HIGH);   // 讓超聲波發射高電壓 10μs，這裡至少是 10μs
        delayMicroseconds(10);
        digitalWrite(outputPin, LOW);      // 維持超聲波發射低電壓
        float Rdistance = pulseIn(inputPin, HIGH);   // 讀差相差時間
        Rdistance= Rdistance/58;            // 將時間轉為距離距离（單位：公分）
        Serial.println(Rdistance);
        Rspeedd = Rdistance;                      // 將距離 讀入 Rspeedd(右速)
     }

void loop()
{
   char val = Serial.read();
   Serial.write(val);
   if (-1 != val) {
      if ('W' == val)
      advance(10);
      else if ('D' ==val)
      right(6);
```

```
else if ('A' == val)
left(6);
else if ('S' == val)
back(10);
else if ('Q' == val)
stopp(10);
else if ('C' == val)
    while('C' == val)
    {
        myservo.write(90); //讓伺服馬達回歸 預備位置 準備下一次的測量
        detection(); //測量角度 並且判斷要往哪一方向移動
        if(directionn == 8) //假如 directionn(方向) = 8(前進)
        {
          char val = Serial.read();
          Serial.write(val);
          if ('C' == val)
          {
            digitalWrite(pinRB,HIGH);
            digitalWrite(pinRF,LOW);
            digitalWrite(pinLB,HIGH);
            digitalWrite(pinLF,LOW);
           break;
         }
          advance(1); // 正常前進
        }
       if(directionn == 2) //假如 directionn(方向) = 2(倒車)
      {
          char val = Serial.read();
          Serial.write(val);
          if ('C' == val)
          {
            digitalWrite(pinRB,HIGH);
            digitalWrite(pinRF,LOW);
            digitalWrite(pinLB,HIGH);
            digitalWrite(pinLF,LOW);
           break;
         }
          back(8); // 倒退(車)
```

```
            turnL(3); //些微向左方移動(防止卡在死巷裡)
          }
           if(directionn == 6) //假如 directionn(方向) = 6(右轉)
         {
            char val = Serial.read();
            Serial.write(val);
            if ('C' == val)
            {
              digitalWrite(pinRB,HIGH);
              digitalWrite(pinRF,LOW);
              digitalWrite(pinLB,HIGH);
              digitalWrite(pinLF,LOW);
             break;
           }
             back(1);
             turnR(6); //  右轉
         }
           if(directionn == 4) //假如 directionn(方向) = 4(左轉)
         {
            char val = Serial.read();
            Serial.write(val);
            if ('C' == val)
            {
              digitalWrite(pinRB,HIGH);
              digitalWrite(pinRF,LOW);
              digitalWrite(pinLB,HIGH);
              digitalWrite(pinLF,LOW);
             break;
           }
              back(1);
              turnL(6); //  左轉
          }
           }
    delay(100);
  }
else
{
  //stop();
```

```
        delay(100);
    }
}
```