

# 報告書

電動輪椅車之自動追人控制系統

合作廠商：成光科技股份有限公司

指導老師：丁慶華 教授

撰寫者：吳敦晏 黃思豪 學生

中華民國 107 年 6 月

June, 2018

# 目錄

第 1 章、零件細項	5
第 2 章、作品架構	6
第 3 章、作品成果	7
3.1、模式判斷	7
3.2、GP 控制器(Joystick)	7
3.3、感測裝置	7
3.4、通訊系統	8
3.5、辨識系統	9
3.6、驅動方式	14
3.7、控制系統	14
3.8、安全機制	20
第 4 章、待改善問題與未來解決方向	24
第 5 章、結論	25

# 圖目錄

圖一、電動輪椅車之架構圖.....	6
圖二、硬體架構圖.....	6
圖三、所有感測器擺放位置之示意圖.....	8
圖四、Arduino 與 Zigbee 通訊步驟.....	8
圖五、區域區分方式.....	9
圖六、自走車上 Zigbee 的位置.....	10
圖七、實測 Rssi 值與距離之趨勢圖.....	10
圖八、方向判斷的依據.....	11
圖九、自走車上焦電感測器的位置.....	12
圖十、主流程圖.....	15
圖十一、中斷流程圖(倒車雷達).....	16
圖十二、中斷流程圖(電流感測器).....	16
圖十三、自走車的倒車雷達方向示意圖.....	20
圖十四、實際自走車之倒車雷達位置.....	21
圖十五、自走車右轉避障示意圖.....	22

# 表目錄

表一、電子零件項目 .....	5
表二、模式判斷 .....	7
表三、指令介紹 .....	8
表四、焦電感測器判斷結果 .....	12
表五、最終判斷結果的方式 .....	13
表六、第一階段時的控制細節 .....	17
表七、第二階段時的控制細節 .....	18
表八、第三階段時的控制細節 .....	19
表九、避障控制之細節 .....	22
表十、利用串列通訊(UART)解碼(鮑率：1000) .....	23

# 第 1 章、使用零件細項

此章節將以表格的方式介紹本研究所用到的電子零件，如表一。

表一、電子零件項目

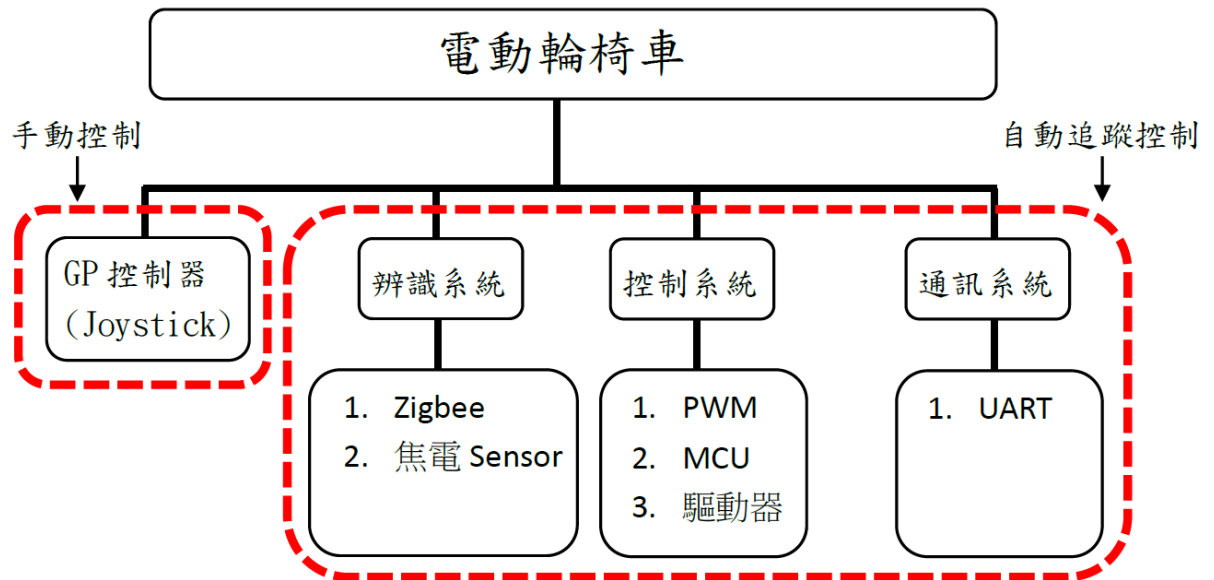
零件名稱	個數
電流感測器	2
焦電感測器	3
Zigbee X2C (TH)	2
Zigbee X2C (SMT)	1
Arduino mega	1
Arduino pro mini	1
倒車雷達模組 (控制器 X1、超聲波 X4)	1
直流電機驅動器 100A 雙路	1
變壓模組	2

※在此只列出額外加裝上去的電子零件，不包含自走車固有的零件

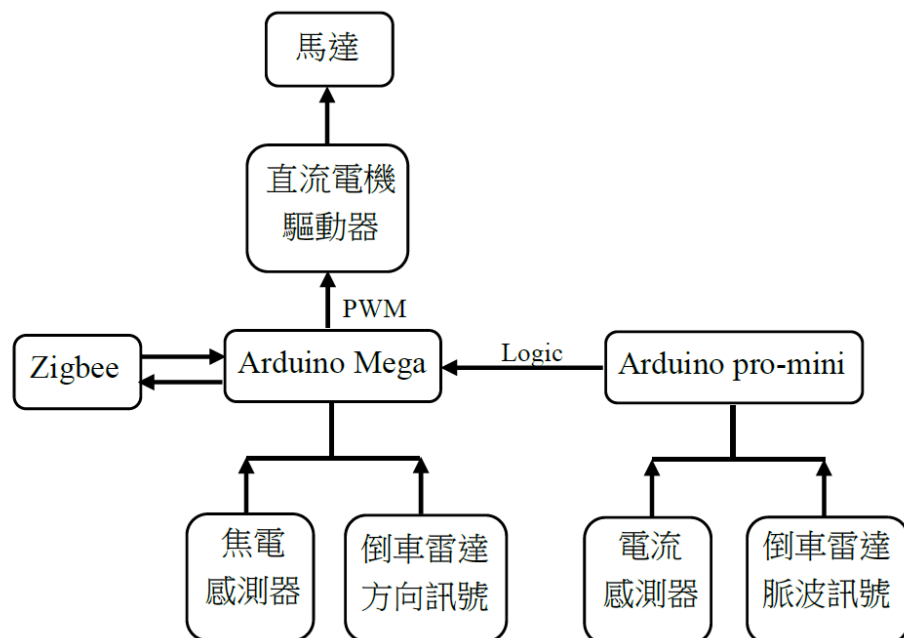
## 第 2 章、作品架構

本研究主要分為三大部份：辨識系統、控制系統、通訊系統。

自走車又分為兩種模式：手動模式與自動追蹤模式，系統架構如圖一，硬體架構如圖二。



圖一、電動輪椅車之架構圖



圖二、硬體架構圖

## 第3章、作品成果

### 3.1 模式判斷

Arduino Mega 會不斷檢查裝置在自走車上的 Zigbee S2C(TH)的第 6 隻腳位是否有脈波的輸入。當配戴在前導者身上的 Zigbee S2C(SMT)通電時，Zigbee S2C(TH)的第 6 隻腳位會有脈波產生，系統則判斷進入自動追蹤模式；反之，則進入手動模式，如表二。

表二、模式判斷

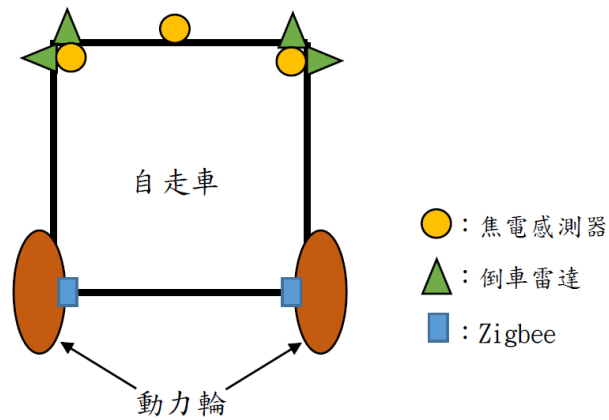
Zigbee S2C(SMT)	Zigbee S2C(TH)腳位 6	系統判斷
通電	有脈波	自動追蹤模式
未通電	無脈波	手動模式

### 3.2 GP 控制器(Joystick)

此部分是廠商所給予的控制器，非本研究自行製作。

### 3.3 感測裝置

本作品運用 3 種感測器：Zigbee、焦電感測器、倒車雷達，擺放位置如圖三。

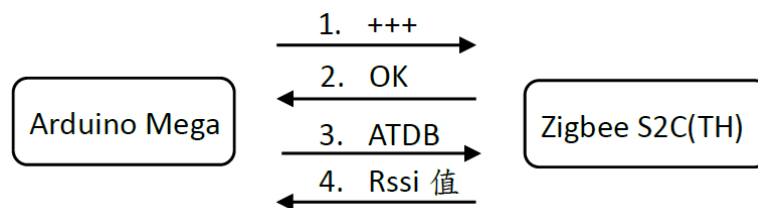


圖三、所有感測器擺放位置之示意圖

### 3.4 通訊系統

Zigbee 有兩種模式：API、AT command，我們使用 AT command 模式，此模式可以立即得到我們所要的資訊，步驟如圖四，指令意義如表三。

此部份的通訊方式需使用串列通訊(UART)。



圖四、Arduino 與 Zigbee 通訊步驟

表三、指令介紹

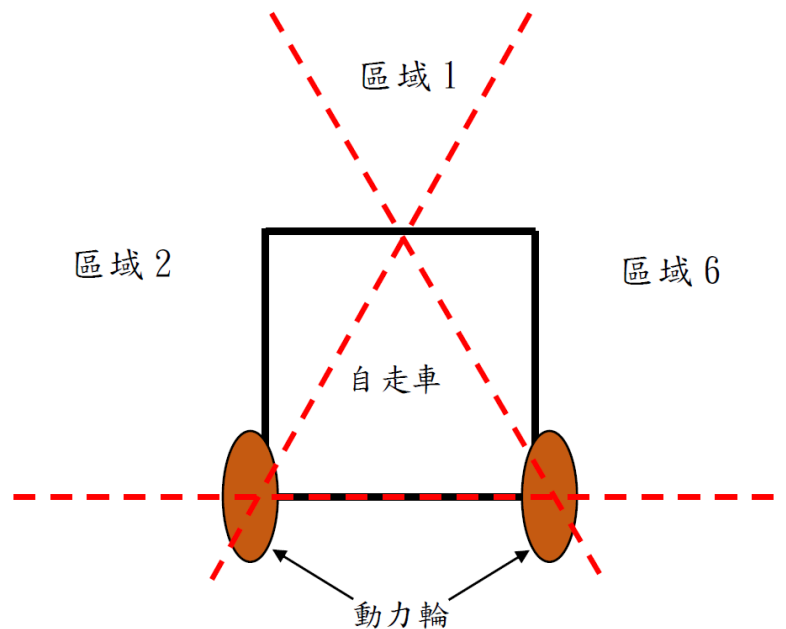
指令名稱	指令意義
+++	要求進入 AT Command 模式
ATDB	要求 Rssi 值



即使 Zigbee 並未與另一個 Zigbee 進行連線，ATDB 指令仍將會暫存最新的 rssi 值，不會將其值變為 0。因為上述的問題，所以每當由自動追蹤模式轉變成手動模式時，系統將會對 Zigbee 進行 reset 的動作。

### 3.5 辨識系統

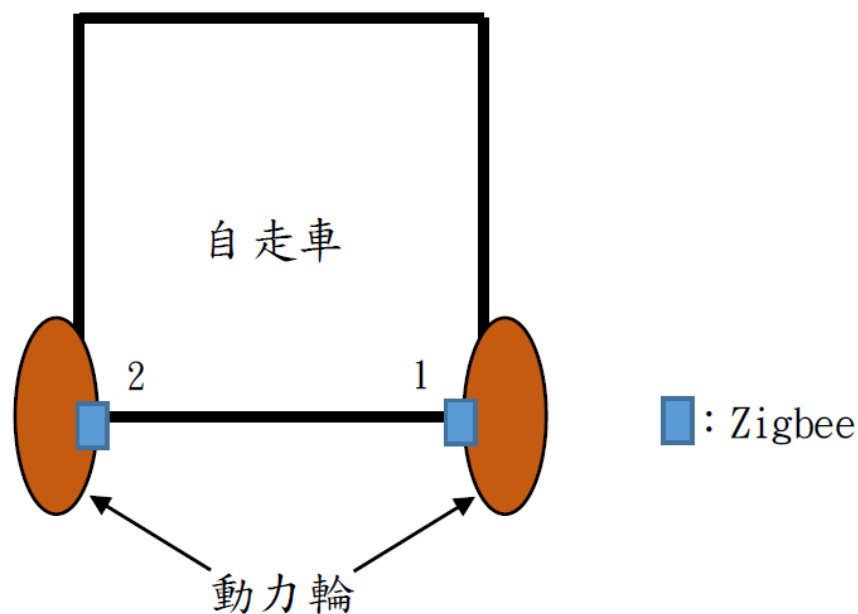
本研究運用兩種感測器： Zigbee、焦電感測器，進行前導者的追蹤。我們將以自走車為平面中心，區分出 3 個區域，如圖五。以下將詳細解釋兩種感測器的判斷方式與最終的判斷結果。



圖五、區域區分方式

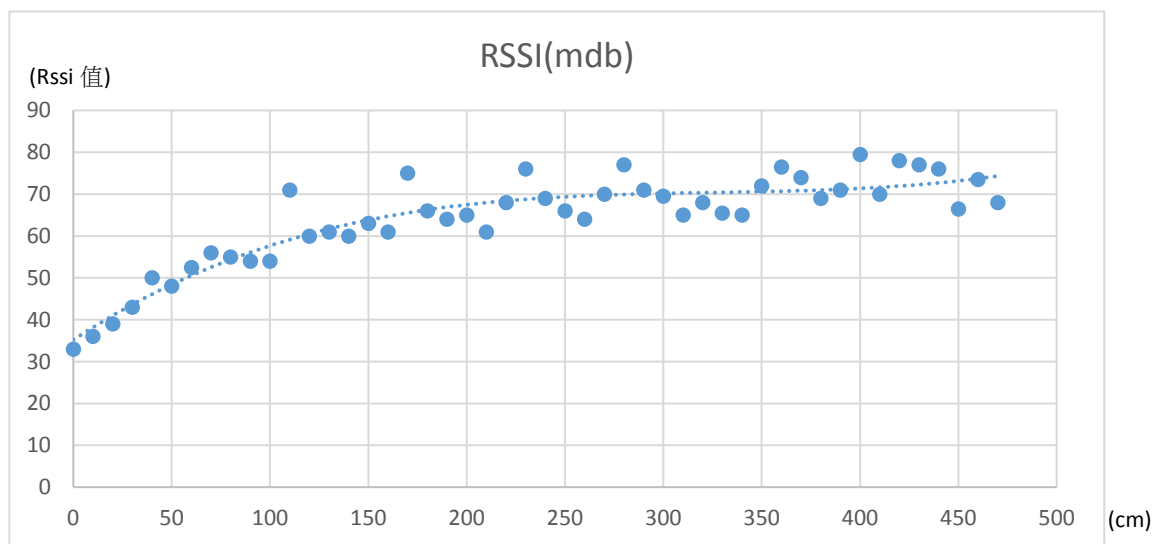
#### ●Zigbee：

當 2 個或 2 個以上的 Zigbee 連線時，皆可得到每個連線裝置之間的訊號強度指標 Rssi 值(Received Signal Strength Indicator)。本研究總共使用 3 個 Zigbee，1 個裝戴在前導者身上，2 個裝置在自走車，裝置位置如圖六。



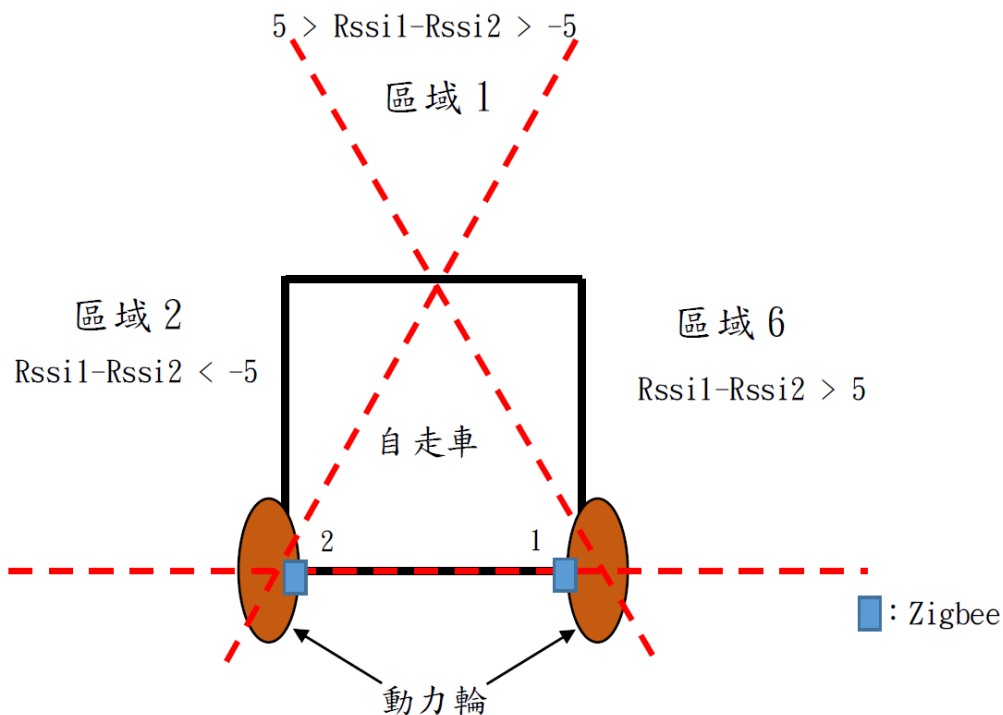
圖六、自走車上 Zigbee 的位置

訊號強度指標 Rssi 值(Received Signal Strength Indicator)跟距離有大致的趨勢，距離越近，強度越強；反之，越弱。然而，其值與距離並非具有線性關係，容易受到環境週遭的影響，如圖七。



圖七、實測 Rssi 值與距離之趨勢圖

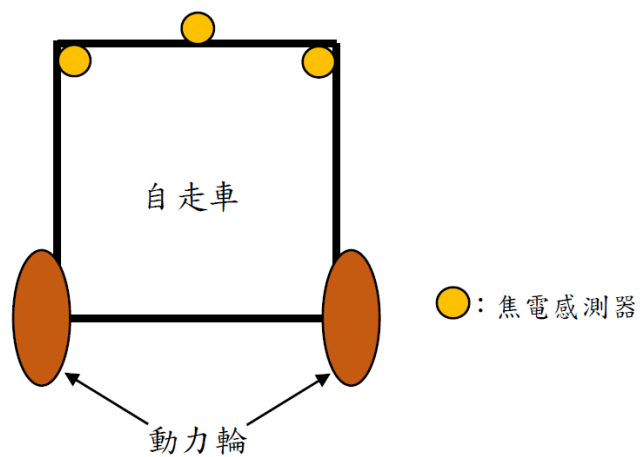
雖然準確度不高，但依然可以用於方向的判斷，而本研究判斷的方式是利用兩個 Zigbee 的 Rssi 值進行大小比較，判斷方式如圖八。



圖八、方向判斷的依據

### ● 焦電感測器：

此感測器會接收生命體所發出的紅外線的变化量，用以辨識前導者的方向，增加系統判斷的可信度。若焦電感測器有偵測到前導者，則會給出一高電位給予 Arduino Mega；反之，則為低電位。本研究總共使用 3 顆焦電感測器，如圖九。判斷結果，如表四。



圖九、自走車上焦電感測器的位置

表四、焦電感測器判斷結果

焦電 1	焦電 2	焦電 3	判斷結果(前導者所在區域)
L	L	H	6
L	H	H	1
L	H	L	1
H	H	L	1
H	L	L	2
L	L	L	Zigbee 判斷結果
H	H	H	Zigbee 判斷結果

## ●最終判斷結果：

此最終判斷結果的目的是為了合併兩個感測器的判斷結果。其中，焦電感測器的判斷結果優先於 Zigbee 的判斷結果，因為焦電感測器的可確性相對較高。判斷方式將以表格的方式闡述，如表五。

表五、最終判斷結果的方式

焦電感測器的判斷結果	Zigbee 感測器的判斷結果	系統最終的判斷結果
區域 6	區域 6	區域 6
	區域 1、區域 2	區域 1
區域 1	區域 1	區域 1
	區域 2、區域 6	區域 1
區域 2	區域 2	區域 2
	區域 1、區域 6	區域 1
x	Zigbee 判斷結果	Zigbee 判斷結果
x	Zigbee 判斷結果	Zigbee 判斷結果

X：3 顆焦電感測器皆是同一個準位的情況

### 3.6 驅動方式

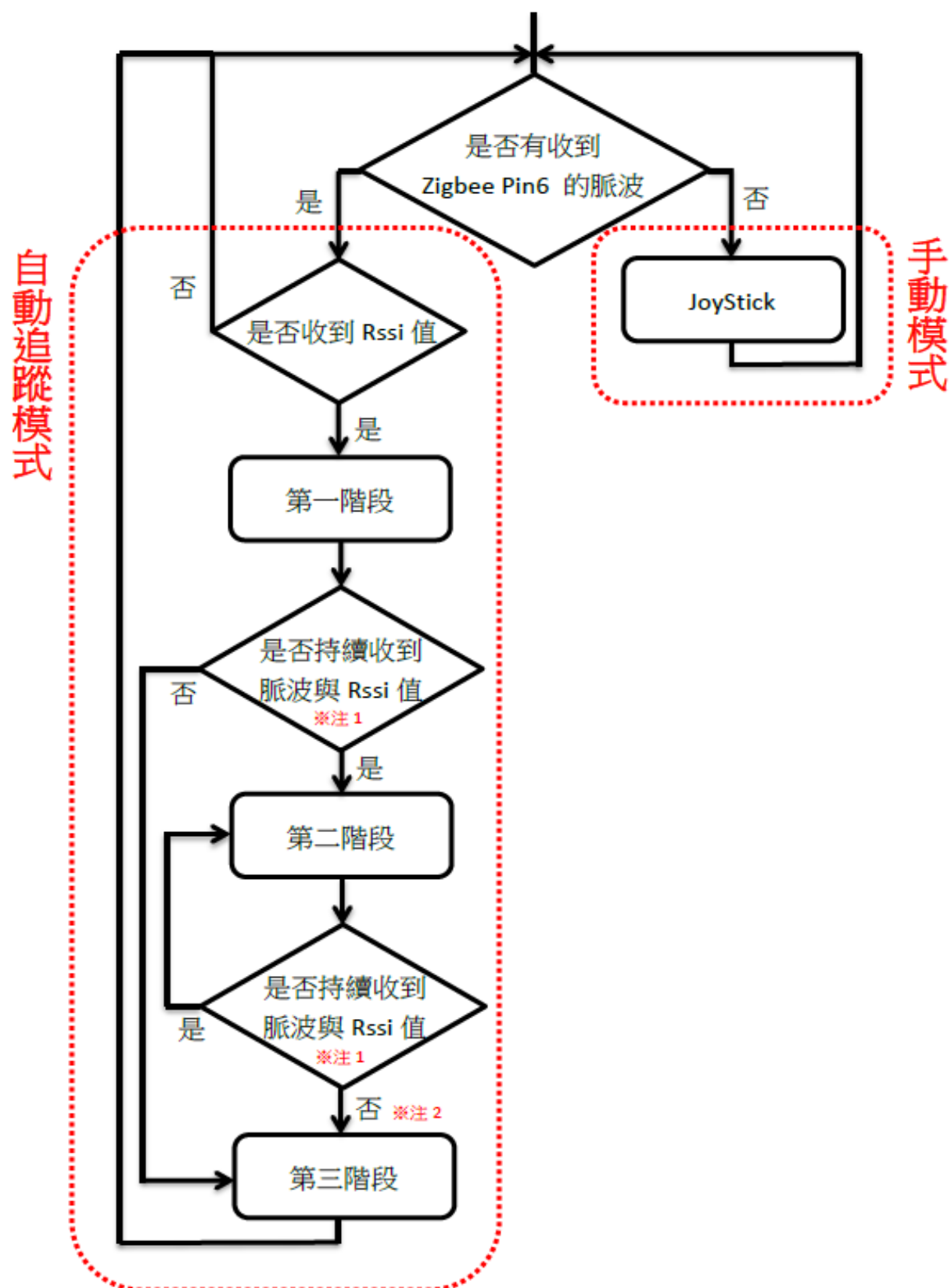
本作品利用 Arduino Mega 給予驅動電路正轉或反轉的訊號，以及透過 PWM 進行速度的調控。其中，PWM 的脈波頻率需要超過人類所能聽見的頻率以上，所以我們將頻率設定為 31k HZ，人類將聽不到馬達驅動時的噪音。

### 3.7 控制系統

本研究中的 Arduino Mega 為主控制器，負責控制模式切換、接收 Rssi 值、輸出 PWM…等主要功能；而 Arduino pro-mini 為輔助控制器，只進行防碰撞的安全機制。Arduino pro-mini 將會以邏輯電位給予 Arduino Mega 進行中斷，立即躲避障礙物，主流程圖如圖十，中斷流程圖如圖十一與圖十二。

此自走車的動力來源是透過 2 顆裝在後輪的 24V DC 馬達。由於前輪是輔助輪，不具有動力，所以其左右轉是依賴後輪的差速進行轉彎。當自走車進入自動追蹤模式且兩個 Rssi 值皆不為零時，自走車會依據最終判斷的結果，進行左轉或右轉或直走等 3 個方向。

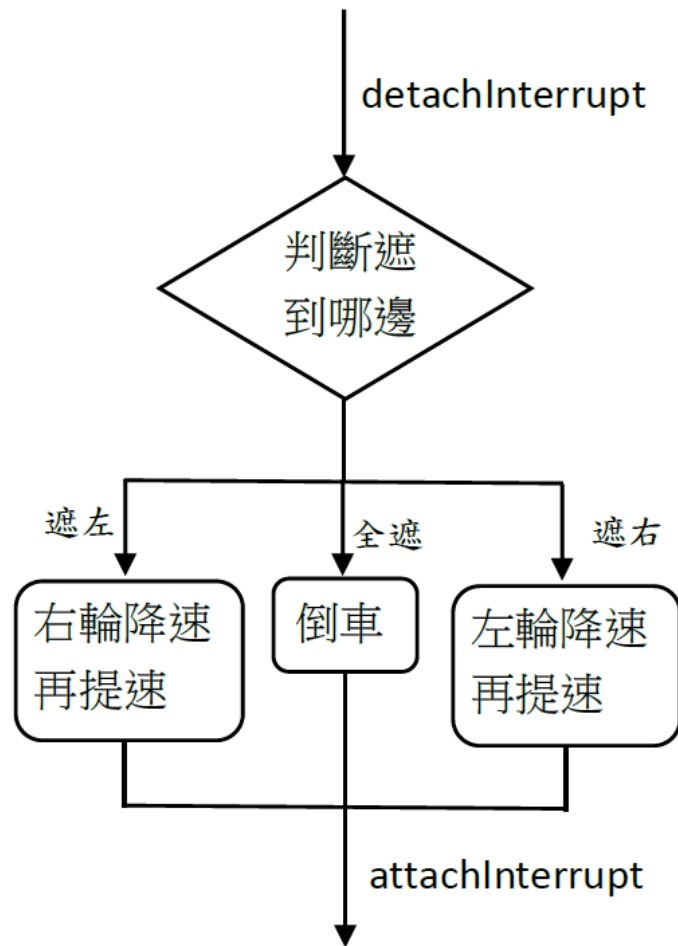
我們將自走車分類為 3 種行走階段：第一階段、第二階段與第三階段。



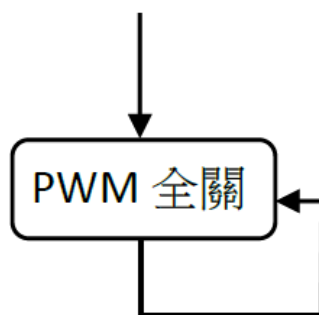
※注 1:此脈波為 Zigbee Pin6 的脈波

※注 2:若未收到脈波或 Rssi 值為 0 則為否

圖十、主流程圖



圖十一、中斷流程圖(倒車雷達)



圖十二、中斷流程圖(電流感測器)



以下內容將分別詳細闡述每個階段的意涵：

● 第一階段：

此代表自走車尚未移動時(停留在原地)的狀況。若自走車已有最終判斷結果，此時自走車將開始行走，控制細節如表六。

表六、第一階段時的控制細節

最終判斷結果(方向)	行走方式	控制細節
區域 6(右邊)	原地 → 右轉 → 直走	左輪先提速到定值，右輪再提速到相同的定值
區域 1(直走)	原地 → 直走	左右輪一起提速到相同定值
區域 2(左邊)	原地 → 左轉 → 直走	右輪先提速到定值，左輪再提速到相同的定值

## ● 第二階段：

當自走車又繼續接收到最終判斷結果時，其控制細節如表七。當系統沒有接收到 Zigbee(TH)第 6 腳位的脈波或是有任一 Rssi 值為零時，將跳脫此階段；否則，系統就是不斷地停留在此階段。

表七、第二階段時的控制細節

最終判斷結果(方向)	行走方式	控制細節
區域 6(右邊)	直走 → 右轉 → 直走	右輪先降速到速度為零，再提速到與左輪相同速度
區域 1(直走)	保持直走	左右輪保持定值
區域 2(左邊)	直走 → 左轉 → 直走	左輪先降速到速度為零，再提速到與右輪相同速度

### ●第三階段：

若 Arduino Mega 未收到 Zigbee(TH)第 6 腳位的脈波，自走車則切換置手動模式；若有任一 Rssi 值為零時，則判斷有異常，但系統仍保持在自動追蹤模式。只要有以上狀況，則會跳脫第二階段並進入此階段；其目的為使自走車停止，其控制細節如表八。

表八、第三階段時的控制細節

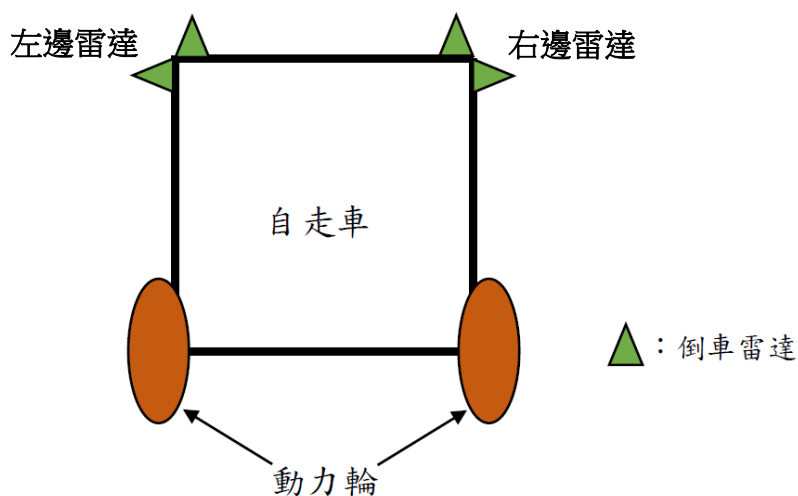
行走狀況	控制細節
正在右轉	兩輪一起降速直到速度為零
直走	兩輪一起降速直到速度為零
正再左轉	兩輪一起降速直到速度為零

### 3.8 安全機制

本節將探討自走車的兩種安全狀況：防碰撞與已碰撞。以下將分別並詳細探討在任一狀況時，系統將會如何降低危險性與風險。

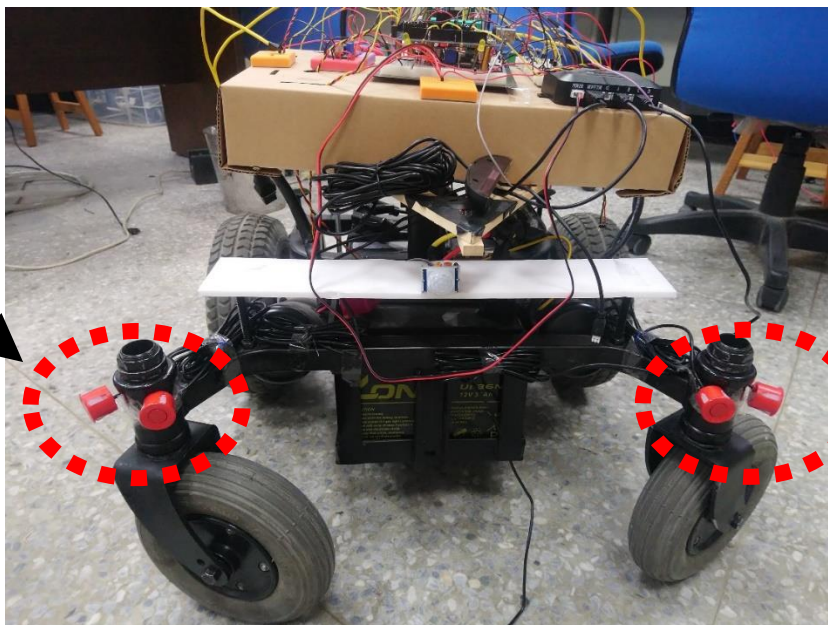
#### ●防碰撞之狀況：

本研究使用市售的倒車雷達並將其裝置在自走車的輔助輪旁，共 4 顆雷達並將其區分為兩類：左邊雷達與右邊雷達，如圖十三與圖十四。



圖十三、自走車的倒車雷達方向示意圖

倒車雷達



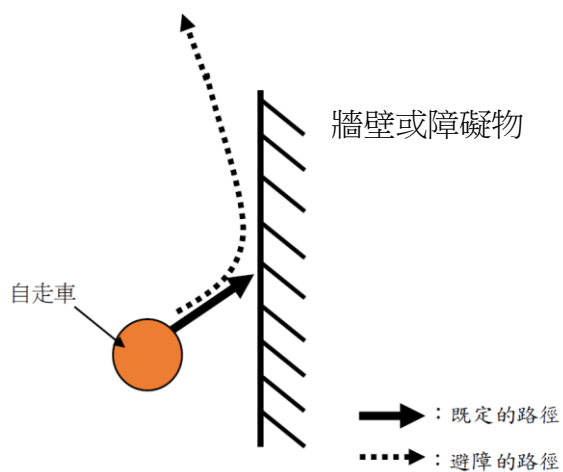
倒車雷達

圖十四、實際自走車之倒車雷達位置

若障礙物與倒車雷達距離不到 50 公分時，則自走車將會開始進行避障。其中，系統又區分為直走時避障與轉彎時避障，避障方式如表九、圖十五。

表九、避障控制之細節

障礙物出現時的狀況	遮蔽哪顆雷達	避障控制
正在直走	左邊雷達	右輪速度降至 0，再提速至與左輪一樣
	右邊雷達	左輪速度降至 0，再提速至與右輪一樣
	皆遮蔽	左右輪倒車 5 秒
正在右轉	左邊雷達	繼續右轉
	右邊雷達	同一時刻，左輪降速，右輪提速，再將左輪提速
	皆遮蔽	左右輪倒車 5 秒
正在左轉	左邊雷達	同一時刻，右輪降速，左輪提速，再將右輪提速
	右邊雷達	繼續左轉
	皆遮蔽	左右輪倒車 5 秒



圖十五、自走車右轉避障示意圖

本研究擷取倒車雷達控制器的兩條訊號線，如圖。一條給予脈波，當距離越近，脈波頻率越高；另外一條則給予方向並利用串列通訊(UART)解碼，不過此訊號並非串列通訊，故此做法可信度較不高，表十。

表十、利用串列通訊(UART)解碼(鮑率：1000)

左邊雷達遮蔽狀況	右邊雷達遮蔽狀況	獲得的十六進位碼
無遮蔽	無遮蔽	FD
無遮蔽	完全遮蔽	FB 或 F9
完全遮蔽	無遮蔽	FD 或 FC
遠距離	近距離	FA 或 FB
近距離	遠距離	FA 或 F8
完全遮蔽	完全遮蔽	FA 或 FB

## ●已碰撞之狀況：

當系統因突發狀況或倒車雷達控制器故障等不可預期之因素，導致自走車撞到障礙物時，此時電流感測器將檢測到自走車馬達的電流超過設定電流值，Arduino Mega 將停止給予 PWM 並進入無限停止狀態，等待重新開機。

## 第 4 章、待改善問題與未來解決方向

此章節將已條列式的方式詳細闡述待改善的問題。

### 1. 倒車雷達方向判斷的問題：

本作品利用有根據的實驗法，將串列通訊(UART)應用於倒車雷達方向訊號的解碼。

雖然可以得到有規律的十六進位碼，但因重複性太高，導致其辨別方向很容易與實際不符合。此部分將常識利用 Arduino 超聲波雷達模組進行替換。

### 2. 自走車在自動追蹤模式不會停下來：

為了讓自走車行走時具有連續性，故我們讓其不斷辨別方向，而未定義何時算是追蹤到目標物。此部份將嘗試擷取焦電感測器的類比訊號，不只用來辨別目標物方向，更能辨別與目標物的距離。



## 第 5 章、結論

無論是業界或學界，利用訊號強度指標 Rssi 值(Received Signal Strength Indicator)進行定位早已發展十幾年，而絕大多數都是做為室內定位之功用。本研究欲利用 Rssi 值做為自走車定位之功用，具有相當創新性。

因 Rssi 值與距離並非線性關係且尚未進行非線性處理，故只依賴 Rssi 值進行定位的判斷，造成自走車的路徑出現嚴重的錯誤。為了補足 Rssi 值的不可靠性，本研究將原本的三維定位轉變成二維定位的方式，只用來判斷前導者的方向，而不知道真實的距離。加入焦電感測器將可提高方向判斷的可靠性，但因只加入 3 顆焦電感測器，只能區分為 3 個方向：左邊、前方、右邊，使得方向的判斷不夠細膩；再加上因為不知道與前導者之間的實際距離，造成自走車每次的旋轉角度都是固定的。

雖然系統已經多次嘗試進行優化且自走車的路徑規劃已比之前進步許多，但因上述問題，所以造成其路徑規劃仍然是機械式的方式。若解決上述問題，相信將可使系統更完善。