

期末專題_B16

一、專題題目：老爺回收智慧循跡自走車

二、成員名單與分工

學號	姓名	軟體分工	硬體分工
1081535	陳怡潔	撰寫顏色判斷，RFID 卡片讀取，超音波測距，物聯網平台進行遠端操控等功能，以及整合整體程式碼	架設 mfrc522 以及超音波測距模組
1083337	王映儒	設計終點以及停車路段自走車功能	組裝車體，安裝寵物小車和相機等設備，以及設計自走車路徑
1083342	黃予庭	設計即時時間播報功能	

三、專題功能說明

首先操作者可利用 RFID 讀卡機逼卡，識別自走車需沿藍線或綠線行走，行走過程中 LED 綠燈亮起，且透過攝像頭擷取畫面，偵測圖片中有色線段位置並調整車頭行走方向。

在行徑過程中運用超音波測距儀，來判斷前方是否有異物阻擋，若有異物則會發出請移除異物之警告音訊，並亮起 LED 紅燈，直到異物被排除自走車才會繼續前進。

此外，路徑中可設置紅線區，當自走車偵測到此區域時，會停下亮起黃燈並廣播現在時刻，隨後播放回收車音樂，提醒民眾該出來倒回收了，當音樂播放結束，自走車自動繼續行走。

另外，藍色與綠色路徑之別為：藍線可利用物聯網平台遠端控制自走車臨時停車可供民眾進行回收，綠線則否。但若操作者仍堅持在綠線要求臨時停車，自走車則會發出禁止臨停之警訊。

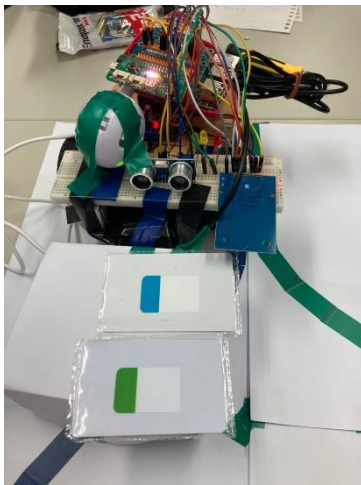
最終，當攝像頭偵測到黑線時，代表指定路線已行駛完畢。這時，自走車停止測距、停車和遠端操控等功能，沿黑線一路加速回到終點回收廠。當到達指定定點後，音箱會提醒操作者按下“q”結束程式。

四、專題操作與執行流程

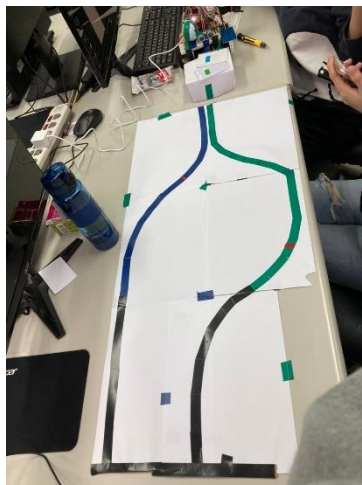
1. 讀卡機逼卡，識別行走路線，如圖一

準備了兩張 RFID 卡，利用讀卡機辨別卡號 ID，來決定自走車應行駛藍線或是綠線。

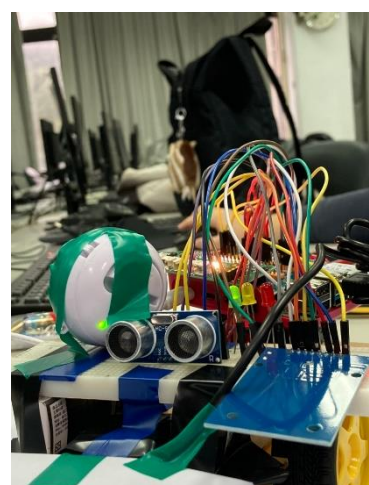
2. 自走車循跡行走，如圖二和圖三
自走車透過攝像頭去擷取畫面，在前方指定三個像素點位置來判別是否為該路徑顏色。若是左像素點為該指定顏色則自走車左轉 0.1 秒；中間像素點為該顏色則自走車向前行走；右像素點為該顏色則往右前行，進而達成自走車循跡行走的功能。
3. 透過超音波測距儀，判斷前方是否有異物，如圖四
若前方 20 公分內有障礙物，則會停止前行亮起紅燈，並發出警訊。
4. 抵達紅線區停車回收，如圖五
當攝像頭讀取到紅線時，則會停止前行亮起黃燈，並播放音樂。
5. 物聯網平台遠端控制臨時停車，如圖六
我們設定藍線為可臨停；綠線為不可臨停線段，透過 ubidots 物聯網平台遠端控制自走車臨時停車。在可臨停線段會停止前行亮起黃燈，並播放音樂；在不可臨停線段則會發出警訊，並繼續行駛。
6. 抵達黑線區加速行駛回指定終點，如圖七
當抵達黑色路段後，程式會停止拿取物聯網平台的雲端數據，如此一來就會大大縮短程式執行一個迴圈的時間，自走車在相同時間內向前、向左或向右的次數會因此而增加，而自走車的速度已回隨之提升，達成加速的效果。
7. 到達終點，如圖八
最終點設計為垂直原先路段的長條黑線，因此當攝像頭辨別指定三個像素點都為黑色時，車子便不再行駛，亮起三個燈泡，並告知操作者結束程式。



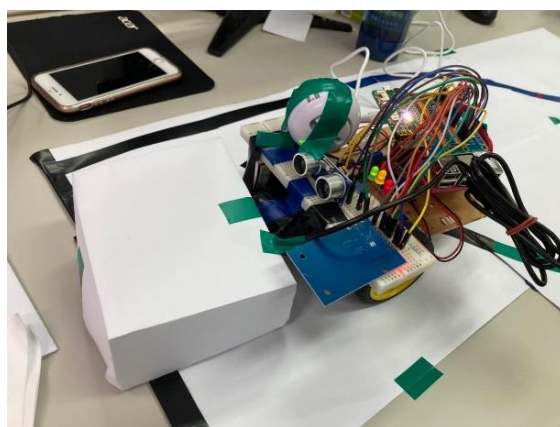
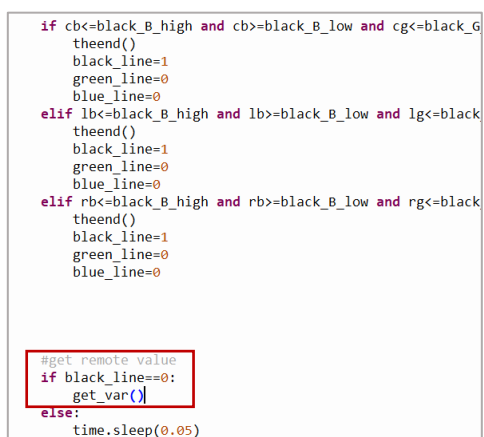
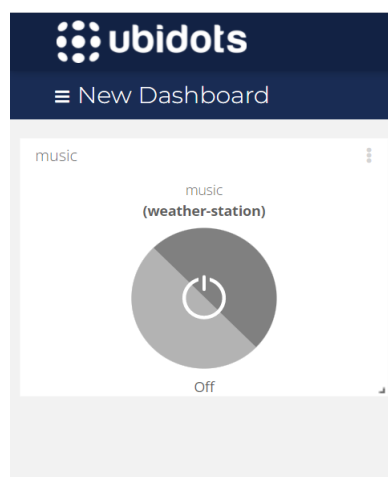
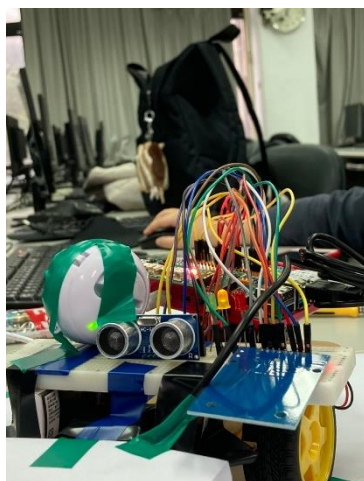
圖一



圖二



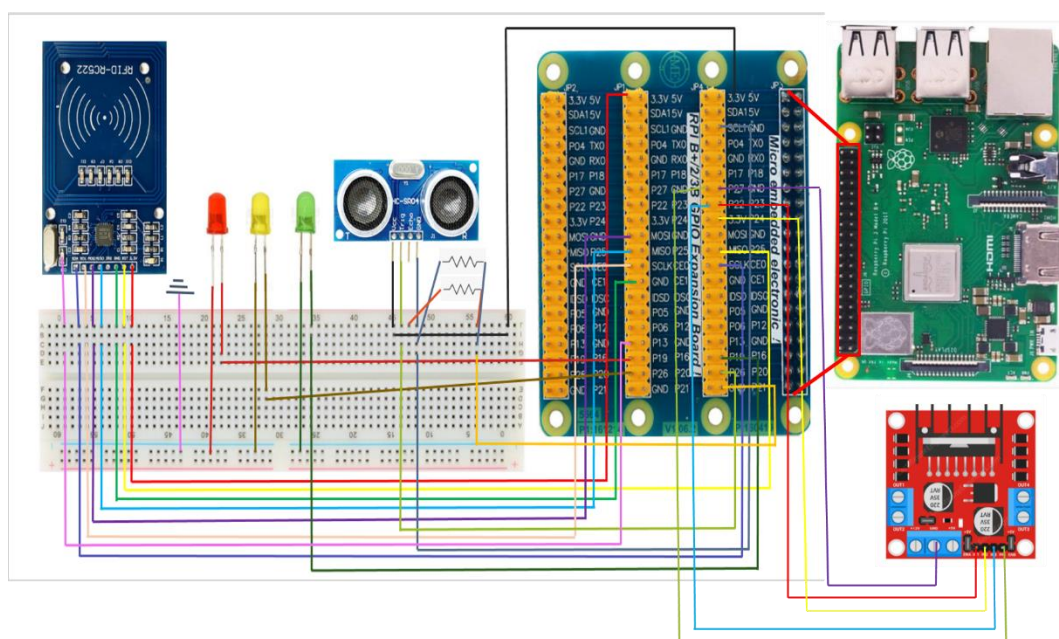
圖三



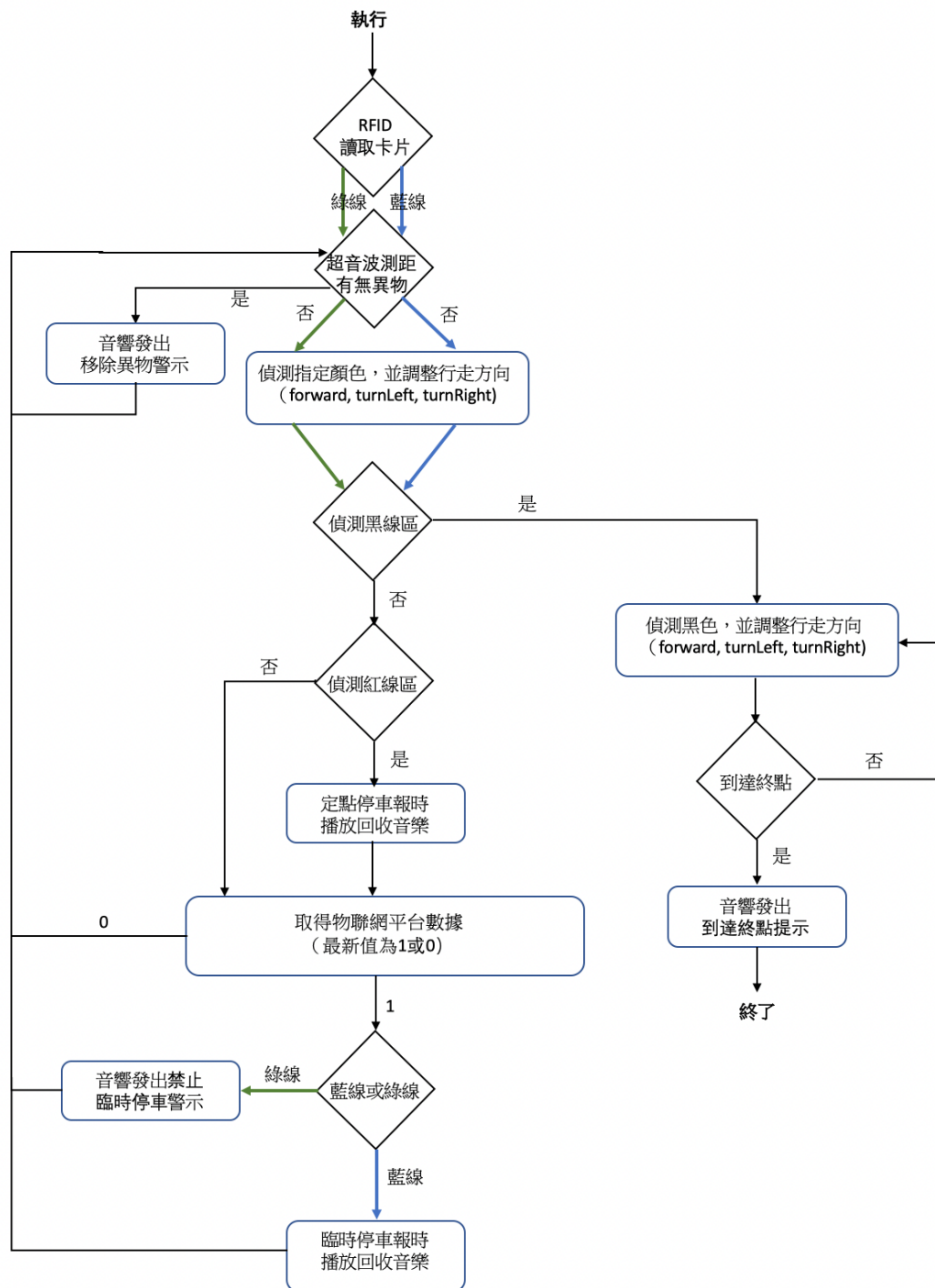
圖七

圖八

五、硬體電路示意圖



六、軟體程式執行流程圖



七、參考的課程實驗網路資源

課程 PPT：

1. Lec 03 - GPIO
2. Lec 06 - Networks&Cloud
3. Lec 07 - Camera

4. Lec 09 - Autonomous car

5. Lec 10 - Smart speaker

網路資源：

1. [RFID 模組 \(https://micro.rohm.com/tw/deviceplus/connect/integrate-rfid-module-raspberry-pi/\)](https://micro.rohm.com/tw/deviceplus/connect/integrate-rfid-module-raspberry-pi/)
2. [超音波測距儀 \(https://blog.everlearn.tw/%e7%95%b6-python-%e9%81%87%e4%b8%8a-raspberry-pi/raspberry-pi-3-model-b-%e8%88%87-hc-sr04-%e8%b6%85%e9%9f%b3%e6%b3%a2%e6%84%9f%e6%b8%ac%e5%99%a8%e4%b9%8b%e6%87%89%e7%94%a8\)](https://blog.everlearn.tw/%e7%95%b6-python-%e9%81%87%e4%b8%8a-raspberry-pi/raspberry-pi-3-model-b-%e8%88%87-hc-sr04-%e8%b6%85%e9%9f%b3%e6%b3%a2%e6%84%9f%e6%b8%ac%e5%99%a8%e4%b9%8b%e6%87%89%e7%94%a8)

八、開發最耗時的部份與原因

我們一致認為判斷線條顏色及位置是自走車能夠沿著設計路線行走這部分最為困難。

首先在判斷顏色的部分，因為在不同的環境可能會有不同的外來光照，如此就容易導致先前針對某一環境調適好的 BGR 值域無法適用，必須再耐心反複調整。

再來，由於我們是利用攝像頭擷取影像進行路徑判別，所以每次自走車移動的幅度若過大則會造成攝像頭無法在跟蹤路徑，以至於路徑偏移等問題。但若是單次讓自走車行走幅度過小，又會發生自走車無法前進的窘境。於是調整單次行走的幅度大小亦成為本期末專題的大難題。不僅要讓自走車走在該走的位置上，又要讓其可靠自身的馬達流暢前行。

此二個困難之處是造成我們在開發自走車循跡行走功能最耗時之原因。而最終我們幫攝像機上方加上遮蔽物，盡可能減少外來光害使影像判別失誤，來解決不同場地 BGR 值域相異問題；而透過更換最新電池和設定最大馬達扭力，來提升自走車行走流暢度，但過程中自走車還是需要一些人為幫助才可順利前行。

九、展示完整專題流程之影片

1. [綠線路徑展示](https://drive.google.com/file/d/1iHbgE7BhOogBMD05aesq9NW11m7E7JIo/view?usp=sharing)
(https://drive.google.com/file/d/1iHbgE7BhOogBMD05aesq9NW11m7E7JIo/view?usp=sharing)
2. [藍線路徑展示](https://drive.google.com/file/d/1ENs3ucQwAOWbB5dM1LXoz0-QcCmVhPUz/view?usp=sharing)
(https://drive.google.com/file/d/1ENs3ucQwAOWbB5dM1LXoz0-QcCmVhPUz/view?usp=sharing)