

撰寫日期：2020/07/06

隊名：阿義紅茶好喝

指導老師：蔡智強/教授

隊長：童凱廷/中興大學電機系二年級

隊員 1：黃柏儒 /機械系二年級    隊員 2：李旭明/機械系二年級    隊員 3：褚懿德/電機系四年級

## 壹、創作敘述

### 一、平台

選擇 NVIDIA Jetson Tx2 與 TI TMS570LS3137 開發版為系統的核心，安裝 linux 作業系統，用 Dronekit 編輯 Ardupilot 程式，使用一條網路線連接兩者，並使用機器人作業系統(Robot Operating Systems, ROS) 作為系統功能模組間以及兩塊開發板間的溝通橋樑。

使用 Ardupilot 是因為其開源且支援多種嵌入式硬件，可在多種平台(例如 Windows、Linux)上運行，而本身提供了很多功能函式庫，可供操控無人載具，也有支援自動駕駛，在各種討論社群網站上，也有許多資源可以運用。

我們選用 NVIDIA Tx2，尺寸僅佔 50mm x 97mm，跟 Tx1 相比 256 核心 NVIDIA Pascal<sup>TM</sup>架構和 8GB 記憶體可以提升兩倍以上的效能，可快速執行現代人工智慧演算法，並可同時執行多個神經網路進行強化學習，而其佔用空間小使能載裝工具的空間較大，功耗僅 7.5 瓦，而本身 NVIDIA Jetpack

是一套完整的 SDK，內含深度學習程式庫、電腦視覺等許多功能。

TI TMS570LS3137 Microcontroller 有 16/32 位元 RISC 快閃微控制器能為針對安全系統設計的高效能汽車級微控制器提供安全架構。核心採用高效率的 1.66DMIPS/MHz。使用運轉時脈最高達 180MHz 的組態下，CPU 可提供最高 298 DMIPS 處理能力，能快速反應並控制機器人。

## 二、整體設計概念及其原因

使用強化學習與深度學習來讓機器人學習躲避障礙物，用 Ardupilot 設定好基本操作流程後(前、後、左、右、右前、右後；左前；左後)，先於 Gazebo 中建置比賽環境，在 ROS 當中使用模擬的比賽環境來讓機器人進行強化學習，經過大量的模擬訓練，可讓行動更加準確，不需要設計過多的演算法去操控機器人行動，而它會自動生成邏輯去做有效率的行動。

## 貳、功能需求

### 一、 智慧賽道 ( 上半場 )：

關卡 1．移動、避障、顏色及物體識別、穿刺功能

關卡 2．移動、顏色及物體識別、夾取功能

關卡 3．移動、避障、顏色及物體識別、推擠功能

### 二、 重量擂台 ( 下半場 )：遙控、移動、格鬥技能、夾取功能

## 參、設計概念與佈置

### 一、 動力轉向結構

為了使機器人機動性最大化，動力部分我們使用麥克納姆輪作為全向輪的輪子。由於麥克納姆輪上之輪轂與輥子夾一角度，所以當輪子前後滾動時也會有橫向的受力。再安排每個輪子依特

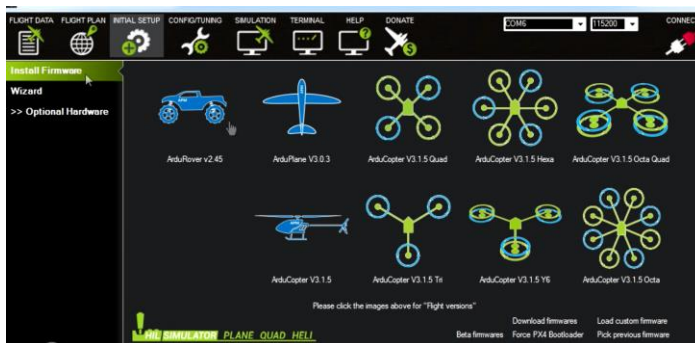
定方向擺置，可使機器人除了前後移動，也可斜向移動或 yaw 軸轉動，使機動性大大提升。為了實現全向移動，我們將在每個輪子裝上獨立之直流馬達，並連結至相對應之馬達驅動器。



(圖一)麥克納姆輪<sup>1</sup>

## 二、 動力控制系統

為了控制馬達轉動，我們使用 Ardupilot 內提供之 Rover 程式碼做修改，控制直流馬達驅動器。而普遍的馬達驅動器，是搭配二輪驅動(只能控制兩個馬達)，為達成四輪驅動，我們會用兩個馬達驅動器並聯。



Ardupilot 介面(圖二)<sup>2</sup>

## 三、 程式設計概念

機器人透過 ROS 啟動 LiDAR 光達來及時建置地圖，然後攝像頭進行影像辨識。藉由感測器來獲得周遭的情報建置 3D 點雲圖，然後使用 ROS 控制機器人進行相對應的功能，例如：爬坡、刺破氣球、積木歸位。

<sup>1</sup> (圖一) 麥克納姆輪，圖原:<https://www.itread01.com/content/1549544428.html>

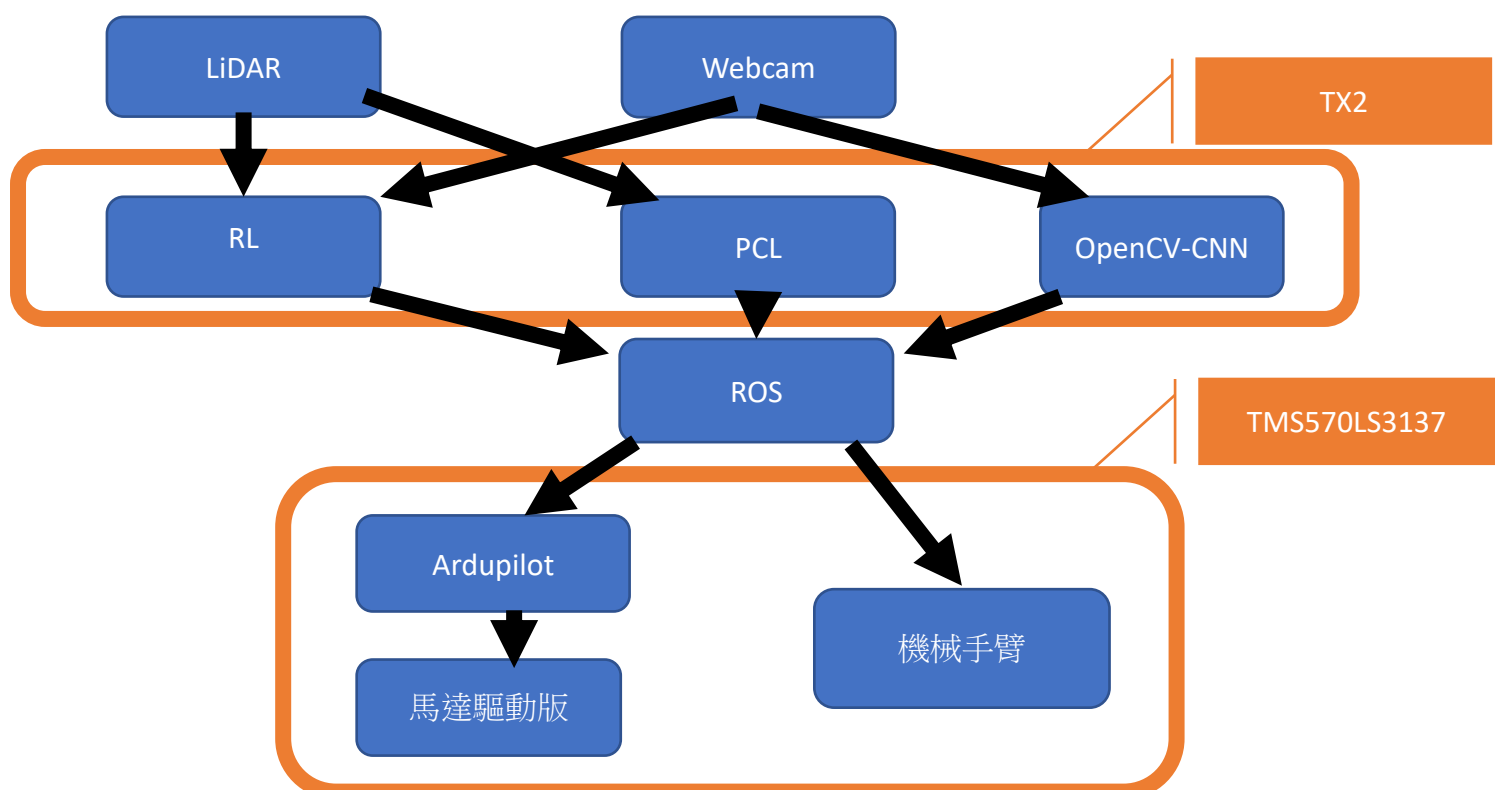
<sup>2</sup> (圖二) Ardupilot 介面，圖原 Ardupilot 程式介面

讓載具搭載光達 (Lidar)來透過其掃描取得與週遭障礙物的距離，並且透過里程計之里程計算、PCL (Point Cloud Library)建立 3D 地圖以及 AMCL (Adaptive Monte Carlo Localization) 之室內定位來取得載具目前的所在位置與想要到達的目的地，並將上述感測到的資料提供給機器人判斷。

攝像頭擷取的圖像經過 opencv-CNN 處理，能將圖像中的區域預選(Region Proposal)以邊界框(Bounding Box)的形式做分類並給予標籤，達到物體辨識的效果，給予的標籤映射到點雲(Point Cloud)上建立起 3D 語意地圖，能夠提取目標物體所在的座標供機器人判斷。

DDPG 採用 Actor-Critic 架構以及 DQN 的概念(Replay Buffer 和 Target Network)，Actor 部分是採用 DPG 確定性策略的方式，輸出連續的動作，Critic 部分使用 TD-error 來評價 Actor 做得好壞。而 DDPG 相較於 DQN 能夠在連續動作上更有效的學習，並且訓練時所需的收斂時間較短。

透過 ROS 社群層節點間的發佈(Publish)與訂閱(Subscribe)整合各個功能的資訊，並且下達命令給 TMS570LS3137 控制馬達驅動版與機械手臂。



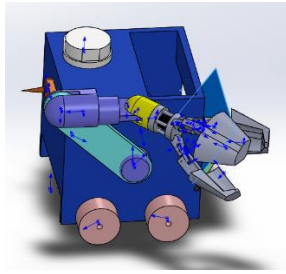
第一關，首先判斷最先辨識的兩個氣球遠近(或是只有一個)，先輸入最近氣球的座標作為目的地，讓機器人的動到座標的正前方，再用 OpenCV 判斷氣球顏色，若是需要戳破的顏色，舉起手臂，車子往前刺破氣球，然後向後回到原位，放下手臂以免移動時誤破其他氣球，接著輸入第二顆氣球位置作為目的地，重複上述動作直到戳破兩個氣球或判斷過三個氣球後，然後進入第二關。

第二關，利用攝像頭判斷積木及盒子的顏色及座標，行動到積木前方，再進行夾取的動作，抬起爪子後藉由紅外線接收器判斷是否已夾起積木，若沒有夾起，則找尋積木位置並移動到前方進行夾取，若有夾起，則移動至盒子中心上方 5cm 並放開爪子，退後至盒子前方判斷積木是否掉出來，若是掉出來則重複抓取動作，若沒有則進行下一個積木的抓取，若是視野內沒有積木存在則移動至兩個盒子的中心點的後方位置並再偵測以免積木掉落至後方，若沒有則進入下一關。

第三關，判斷地形的上升為道路並定位道路頂點中央為目的地，讓訓練好的機器人自動前行，接著定位後方板子的後方正中間為目的地，讓訓練好的機器人自動前行，接著把鏡頭朝向板子的正對面，辨識字母 T 與 L，設定字母 T 與 L 座標的中點前方為目的地，讓機器人前行，然後舉起手臂，機器人前行，後退到上個設定的目的地並判斷字母 T 與 L 中間的木板是否掉落，若無掉落則重複前進、後退、判斷的動作，若是掉落，則不做事。

#### 四、外型結構設計理念

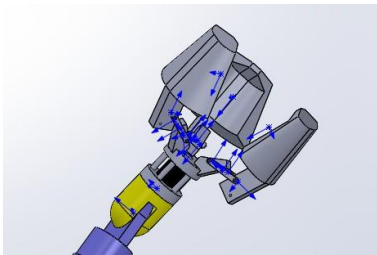
外觀構想主要有車體、光達、攝像頭、機械手臂、載物台、氣球爆破裝置。我們將開發板、馬達驅動器、馬達、電池、線路.....都包裝在車體內並使用多重隔層使之散熱及減輕重量，而車體骨架採用角鋼使結構有高可塑性以及調整性。



(圖三)車體設計圖<sup>3</sup>

由於光達需要能觀測車體周圍 360 度周遭地形，我們將光達架設於車體頂端，並將攝像機以嵌入式的方式設置於車體前方以減少體積。

機械手臂的部分，為避免遮擋住光達與攝像機的視角，我們將其設置於車體左側，並以三個關節使其可伸縮，在收納以及夾取物體更為方便。爪頭的部分，考慮到有正方形之積木與圓形之撞球，我們將會使用三爪(對球型物體有利)且類似爪斗車之爪頭(對方形物體、小型物體有利)，使物體更不容易從旁滑落。另外手臂上也會安裝針刺作為氣球爆破裝置。



(圖四)爪頭設計圖<sup>4</sup>



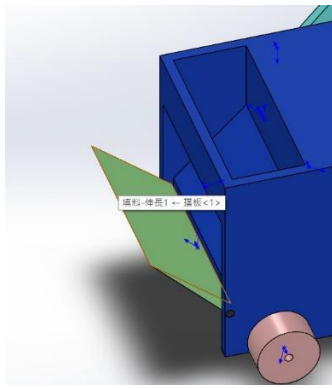
(圖五)爪斗車<sup>5</sup>

載物台考量到下半場的寶藏爭奪賽，如是僅用機械手臂一次搬運一個物體，效率太低，我們在車體後方裝置一載物台。而載物台由三個物件組成:斜坡、閘門、推物裝置。斜坡斜度可提供物體滾出的動力，而斜坡出口高於集寶區之高(5cm)。閘門部份我們會使用齒輪連接金屬齒輪減速馬達閘門不因物體撞擊而開。最後因為是使用斜坡載物台的關係，積木和撞球應會堆於集寶區邊緣，可能造成物體滾落失敗，故我們加上推物裝置使物體夠順利進入集寶區。

<sup>3</sup> (圖三)車體設計圖，使用程式:SolidWork

<sup>4</sup> (圖四)爪頭設計圖，使用程式:SolidWork

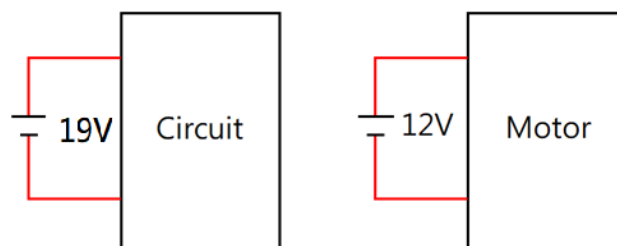
<sup>5</sup> (圖五)爪斗車，圖源: <http://www.yeandu.com.tw/Company01.aspx>



(圖六)匣門設計圖<sup>6</sup>

## 五、電力與控制系統

為避免因使用單電源可能造成的供電問題（如供電不穩、馬達干擾或燒毀電路等），將馬達與其它電路分別使用兩組電源，直接提供所需電壓。



(圖七)Circuit：開發板、攝像頭、光達；Motor：馬達驅動器、馬達<sup>7</sup>

## 肆、創意與技術

### 一、動力優化

機器人以麥卡納姆全向輪作為移動工具，我們認為全向輪在第一關的避障能力會比一般輪子或是履帶來的機靈，再轉向的時候不會有會撞到東西的風險。

### 二、3D 點雲圖

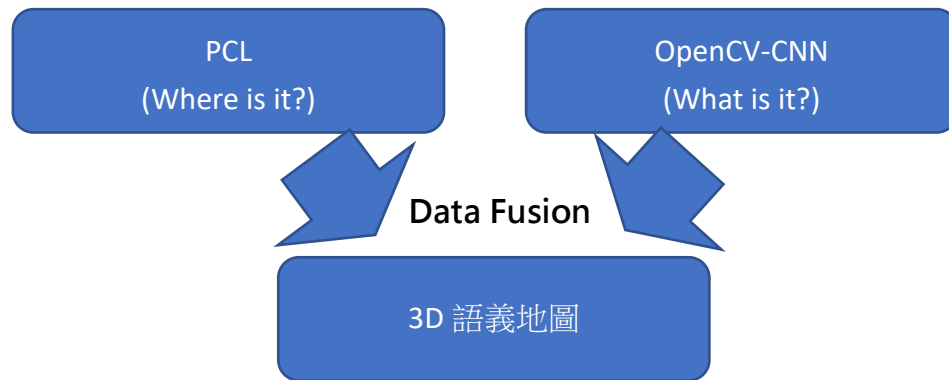
點雲資料庫 PCL 中提供許多點雲處理上的最新算法，包括過濾、特徵估計、表面重建、配

<sup>6</sup> (圖六)匣門設計圖，使用程式:SolidWork

<sup>7</sup> (圖七)Circuit：開發板、攝像頭、光達；Motor：馬達驅動器、馬達

準、模型擬合和切割。幫助我們在 2D 影像與深度資訊做結合

我們使用 ORB-SLAM2 的 RGB-D 影像地圖進行追蹤與稀疏建圖，且將深度資訊投影到相應密集點雲圖，基於影像處理的技術，進行場景感知與辨識，將 2D 影像資訊與深度資訊作結合，及 Opencv-CNN 物體偵測得到語義資訊，並搭配 PCL 點雲模組建造 3D 語意地圖。



### 三、深度學習與強化學習優化機器人自控能力

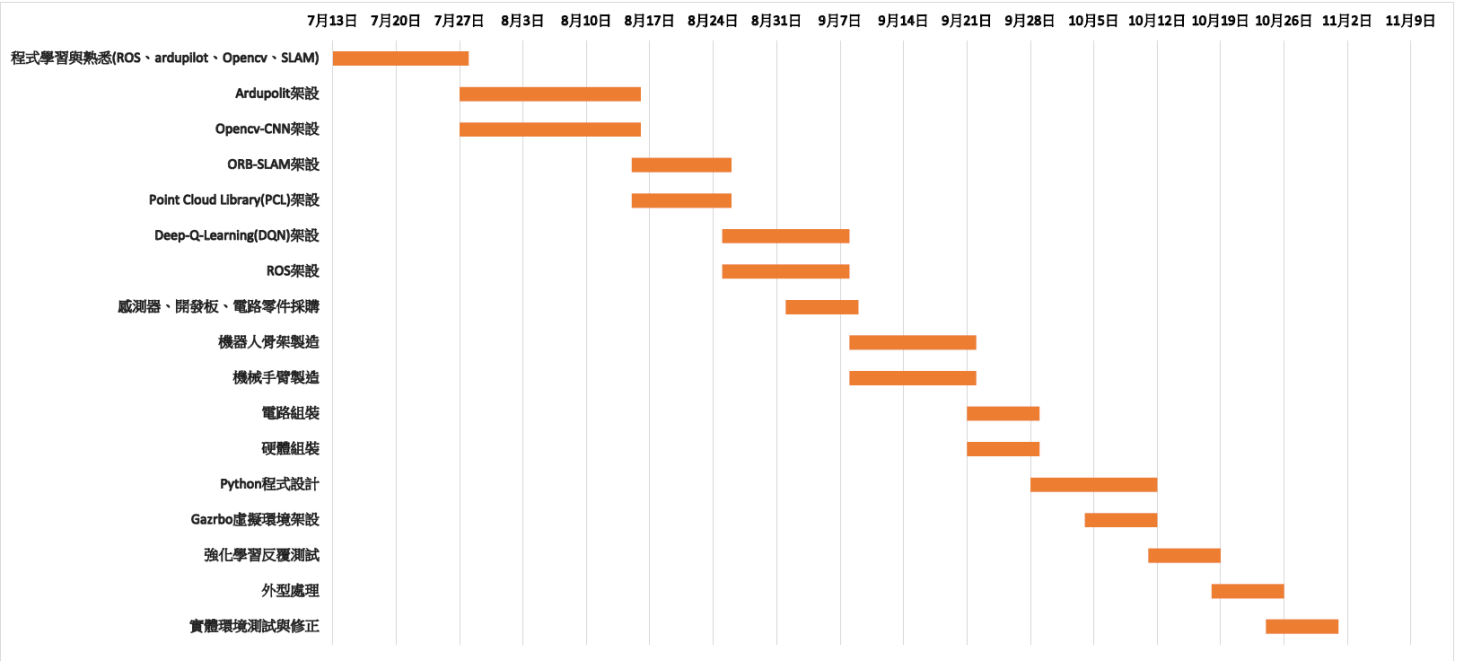
是最重要的部分，我們選擇最近發展的機器人深度強化學習演算法(Deep Reinforcement Learning, DRL)·結合了深度學習 (Deep Learning, DL) 與強化學習 (Reinforcement Learning, RL) ，透過不斷的嘗試，並根據環境中獲得的獎勵 (Reward) 進行評估，最後讓機器人學會如何避障，只需要輸入目的座標，不需要操控行動。



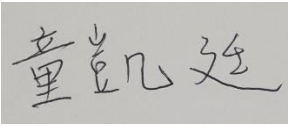
## 伍、預算規劃

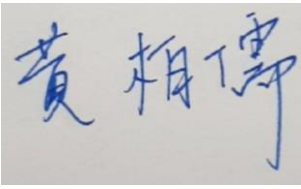
名稱	數量	小計	備註
光達	1	6700	360 度雷射測距儀
攝像機	1	699	羅技 C270 HD 網路攝影機
開發板	1	19990	NVIDIA Jetson TX2
開發板	1	1,062	TI TMS570LS3137
直流馬達(for 麥輪)	4	3528	MOD0020 210RPM 6V 直流馬達
齒輪減速馬達(for 閘門)	1	746	MOA0039 30:1 微型金屬齒輪減速馬達 HPCB
麥克納姆輪	4	2120	14144L 鋁萬向輪直徑 60mm
馬達驅動器	2	1266	ADIO-L298N 馬達驅動器
5V 鋰電池	3	600	兩組備用
9V 鋰電池	3	600	兩組備用
機械手臂材料	1	2000	
角鋼骨架		2000	
線路		500	
載物台材料		400	
其他		1000	
合計		43211	

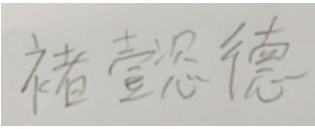
陸、工作時程表

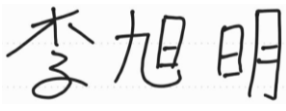


參賽人員簽名

童凱廷: 

黃柏儒: 

褚懿德: 

李旭明: 

指導教授簽名: 