

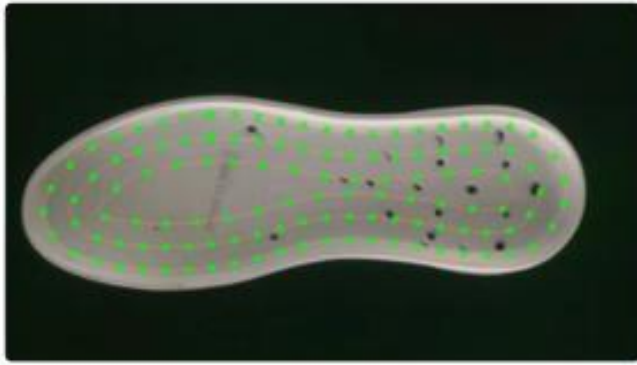
專案及論文簡介

報告製作人：陳佑任

目錄

一、鞋底噴膠機影像辨識.....	1
二、運動內衣點膠機及畫線機影像辨識.....	2
三、深度相機點雲處理.....	3
四、多功能產品檢測機.....	4
五、ROS MoveIt 手臂及路徑模擬.....	5
六、AMR 論文：智能乒乓球撿球機.....	8

(一) 鞋底噴膠機影像辨識



1. 應用場景：

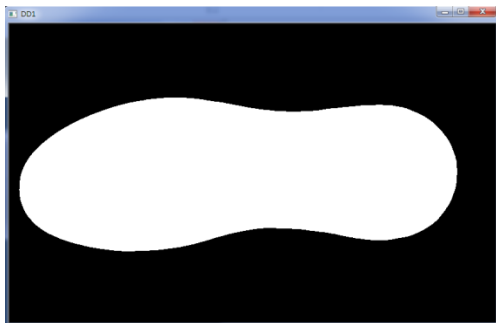
取代塗膠過程會揮發對人體有害之氣體及提高噴塗效率。

2. 困難與痛點：

- (1) 有害人體健康
- (2) 人力需求大

3. 實作流程：

- (1) 使用 OpenCV 將 RGB 彩色圖片轉為「灰階圖片」(有助於提升電腦計算速度)
- (2) 利用「灰階圖片」之色差找出所需的鞋底「特徵」
- (3) 利用「特徵」進行二值化，將所需要的區域改白色，不需要的區域改黑色



- (4) 噴塗路徑：使用形態學工具之腐蝕，將輪廓特徵往內等距縮放獲取輪廓，形成路徑達到均勻噴塗的效果

(二)運動內衣點膠機及畫線機影像辨識



1. 應用場景：

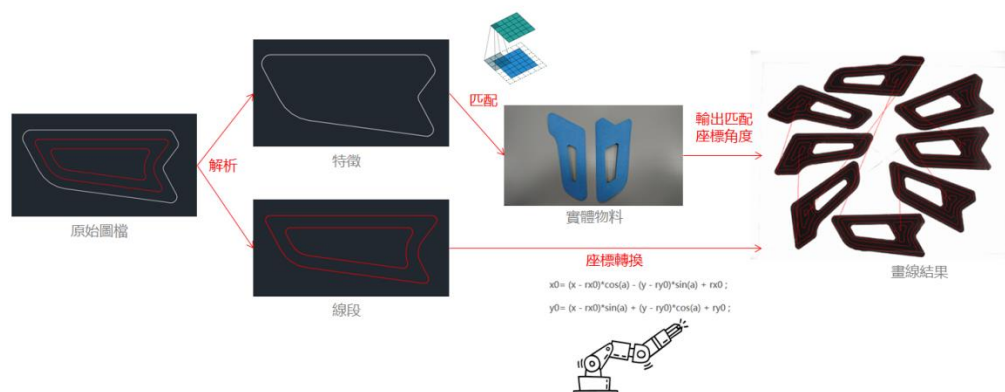
由於布料需要仰賴人工進行貼合，需畫出邊界線給予貼合人員作為貼合依據，而運動內衣多為無車縫內衣，可利用此設備進行辨識定位及點膠作業。

2. 困難與痛點：

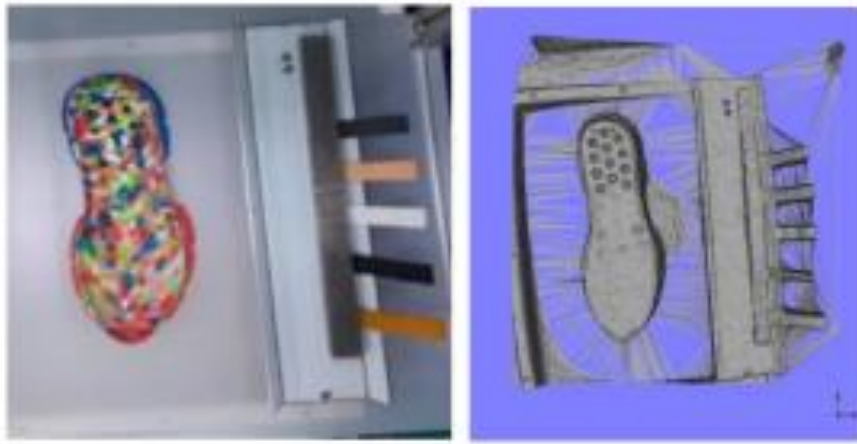
目前此設備為泛用型，有助於提升生產效率及精度。

3. 實作流程：

- (1)布料生產過程中皆會使用 CAD 進行生產，利用 CAD 檔輪廓建立模板
- (2)加工完成之布料，可隨意擺放至工作平台
- (3)程式進行匹配對象與目標來達到定位的功能
- (4)取得匹配中心座標、角度及縮放值進行三角函數的座標轉換
- (5)相機座標與機器手座標的空間轉換關係，最終控制機械手到達目標位置



(三)深度相機點雲處理



1. 目標：

找出鞋邊牆輪廓

2. 實作流程：

(1)點雲解析每個逗點分隔內的信息為 XYZ 值

(不同雷射及深度相機設備回傳數值會略有不同，但原理相同)

```
207 206 205 378.00,206 206 381.00,206 207 383.00,:
208 207 204 376.00,207 205 379.00,207 206 382.00,:
209 208 204 377.00,208 205 380.00,208 206 382.00,:
210 209 204 377.00,209 205 380.00,209 206 383.00,:
211 210 204 378.00,210 205 381.00,210 206 384.00,:
212 211 203 376.00,211 204 379.00,211 205 382.00,:
213 212 203 377.00,212 204 380.00,212 205 383.00,:
214 213 203 377.00,213 204 381.00,213 205 384.00,:
215 214 203 378.00,214 204 382.00,214 205 385.00,:
216 215 203 379.00,215 204 383.00,215 205 386.00,:
217 216 203 380.00,216 204 384.00,216 205 387.00,:
```

(2)利用 Z 值將數據進行分割

(3)將解析完成之數值進行點雲可視化

(4)進行數值的行掃描比對找出每行最高的兩個點保存鞋邊牆數據

(已知鞋邊牆為最高點)

(5)利用找出之鞋邊牆數據來進行手臂姿態演算的參考依據

補充：

另一種影像處理處理方式

將 X、Y 做為圖片 pixel 大小、Z 軸進行歸一化成 0~255 之間之灰階圖片進行影像處理，獲得感興趣區域後，將原始數值及二值化數值整合取出數值特徵

(四) 多功能產品檢測機



1. 應用場景：

完整記錄產品檢測過程，讓廠商提高管理效率與消費者可以快速追朔相關資料。

2. 困難與痛點：

- (1) 檢測設備一般需針對多項項目進行檢測，轉換多台設備分別執行檢測，整體效率不佳
- (2) 需要大量人工進行判斷且判斷標準、責任歸屬難以釐清

3. 實作流程：

- (1) 品檢人員針對識別證工號進行掃描
- (2) 針對型號直接讀取產品數據庫獲得公差與產品規格
- (3) 設備通信(每個設備皆有不同的通信協定)，一次性檢測
- (4) 觸發多台檢測設備信號並獲取所有檢測數值，利用數據庫之公差規範進行程式邏輯撰寫
- (5) 程式自動判斷產品是否有符合規範，並上傳數據庫將品檢履歷進行保存

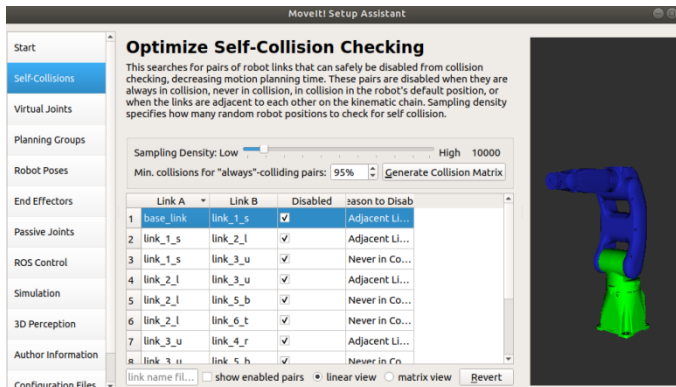
(五) ROS MoveIt 手臂及路徑模擬

1. 應用場景：

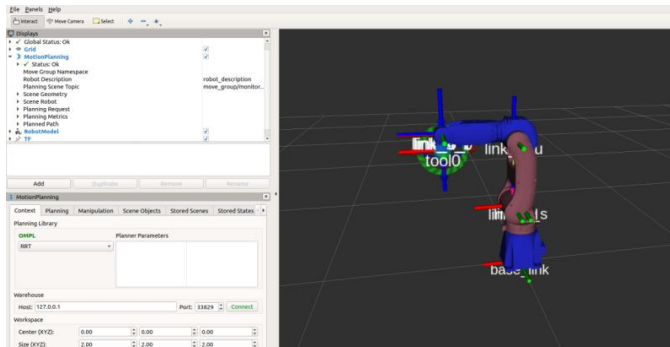
協助同仁運用噴膠手臂路徑及姿態演算法撰寫並提供可視化模擬工具

2. 實作流程：

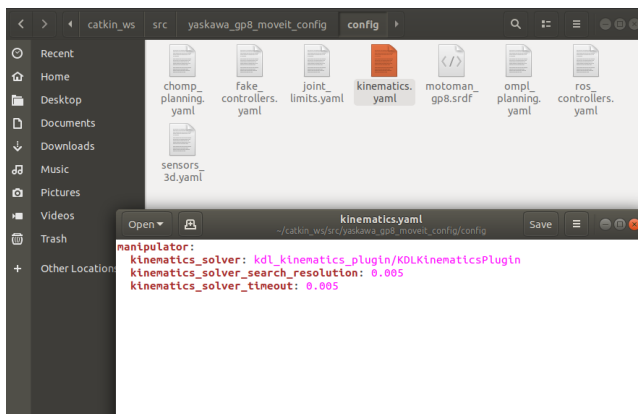
(1) 獲取/製作 URDF 文件(ROS 讀取 3D 模型格式)，並且設定手臂運算參數



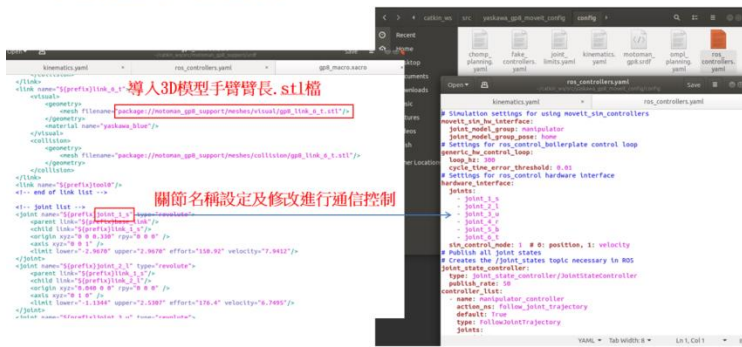
(2) 開啟並設定完成後，導出文件手臂模型



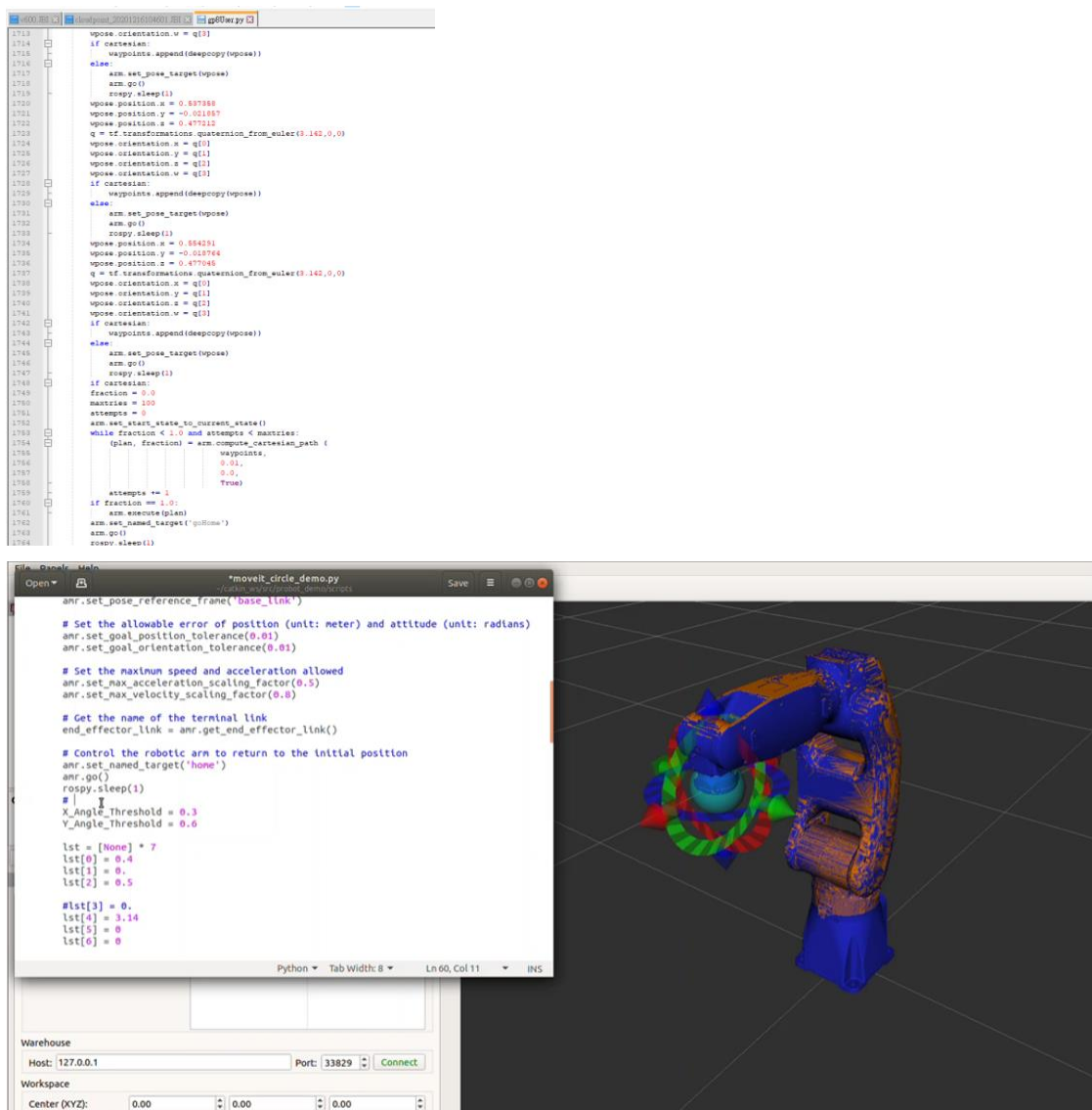
(3) 設定手臂運動學求解器 默認為 kdl



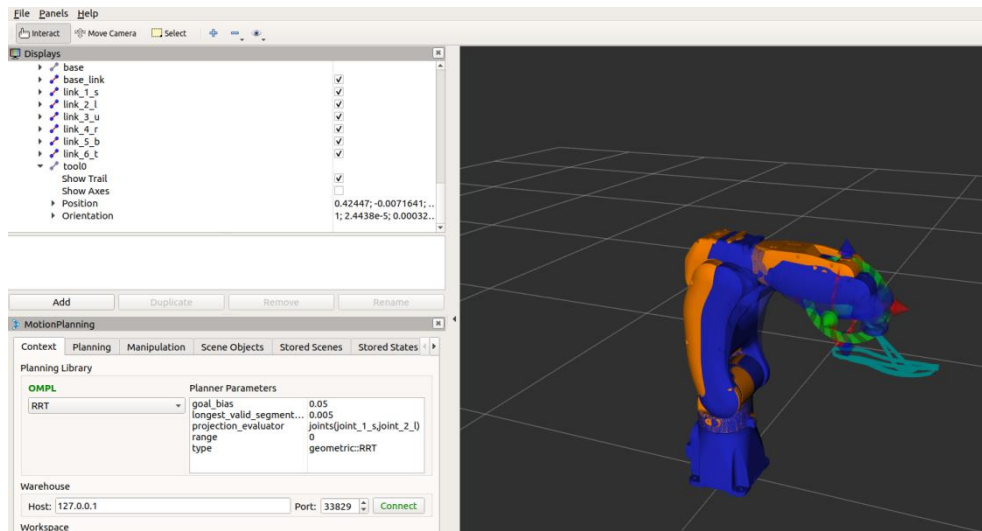
(4) 3D 圖檔及模擬環境手臂關節通信設定參數



(5) 使用 Python 將點雲座標位移路徑及姿態程式撰寫



(6)可視化驗證姿態及軌跡路徑



(7)手臂最末端笛卡爾座標系 時間位移之座標、速度、加速度數值進行保存

```
name.txt
positions: [7.780042902243647e-05, 0.10217761338697205, -0.402390038101842, 6.630393631819093e-05, -0.9853080022791457, 4.347945216850753e-05]
velocities: [-2.4139346175153654e-05, 0.34146452794011484, 0.4116055020111498, -3.129656628706405e-06, -0.07014097317969926, 2.975425377781807e-05]
accelerations: [-3.403160875542848e-05, 0.0622686911850303, 0.9239294289338565, -1.104995777959683e-05, -0.2616607273011753, 5.446294788915729e-05]
effort: []
time_from_start:
  secs: 1
  nsecs: 918005300
-
positions: [7.61313325316681e-05, 0.2073031235256083, -0.37160504693318055, 6.614015450484e-05, -0.9908074832032661, 4.566747166532049e-05]
velocities: [-2.613825276481937e-05, 0.38451301181396724, 0.4738517261150281, -3.92394779036825e-06, -0.08933871265800418, 3.323046326700369e-05]
accelerations: [-2.233693104845181e-05, 0.5612917413491778, 0.84986622467553, -1.108512682591988e-05, -0.2885744814798491, 4.445110594986845e-05]
effort: []
time_from_start:
  secs: 1
  nsecs: 986850406
-
positions: [7.445899733739884e-05, 0.2324590134704126, -0.3404603800000995, 6.587210550093113e-05, -0.9969562600815878, 4.782680890292357e-05]
velocities: [-2.7373751209526154e-05, 0.41896613715816866, 0.5274657349131338, -4.683110130852236e-06, -0.10049959600349474, 3.592838832829149e-05]
accelerations: [-1.6832416797074850e-05, 0.5368702439272021, 0.8612847681337287, -1.2577711917539395e-05, -0.32441452260209425, 4.122775904953904e-05]
effort: []
time_from_start:
  secs: 2
  nsecs: 49087836
-
positions: [7.285847012620359e-05, 0.2574352878803841, -0.3086976317011385, 6.550163158486283e-05, -1.0037427338681595, 4.996038301337193e-05]
velocities: [-2.8123540308458332e-05, 0.4479367168692245, 0.5756407177493346, -5.407803080844625e-06, -0.12770399964777696, 3.810586628048085e-05]
accelerations: [-8.838651612537339e-06, 0.4633589368566637, 0.8043028934936498, -1.250542819632228e-05, -0.34094395545735095, 3.388678407255086e-05]
effort: []
time_from_start:
  secs: 2
  nsecs: 106501210
-
positions: [7.132521494911062e-05, 0.20234194966811804, -0.2763757250320128, 6.52705327699675e-05, -1.011579778487458, 5.207085718306087e-05]
velocities: [-2.8485848474140261e-05, 0.47232205672593064, 0.6191123665137004, -6.087961202347386e-06, -0.14679030789718858, 3.985406197627764e-05]
accelerations: [-4.40122312228703e-06, 0.4352576545411299, 0.7988962339383724, -1.2582742437418084e-05, -0.36363857843454134, 3.0497318450577885e-05]
effort: []
time_from_start:
  secs: 2
  nsecs: 100546489
-
positions: [6.985507089985787e-05, 0.30720921365182297, -0.24347067105490539, 6.494056030680180e-05, -1.0191957685436674, 5.4160691345198926e-05]
velocities: [-2.8573502529852765e-05, 0.4933723220326104, 0.6591353838160117, -6.721607114480548e-06, -0.16576305985177775, 4.12847492758266e-05]
accelerations: [1.1254488786071464e-06, 0.3799129963637341, 0.7523067117033263, -1.198133827688777e-05, -0.3723937146864108, 2.4864068146257918e-05]
effort: []
..
```

(六)AMR 論文：智能乒乓球撿球機

應用場景：

設計的智能乒乓球撿球機器人是為了解決撿乒乓球工作繁雜的問題而設計。這個機器人系統作為智能乒乓球訓練場景的一個子模塊，期望能夠進行一定量的乒乓球撿球工作，能夠實現自主識別和定位；同時使用各種傳感器，保證在撿球的過程中自動尋路並且不影響運動員的訓練，讓整個訓練場景體現更多的智能化。

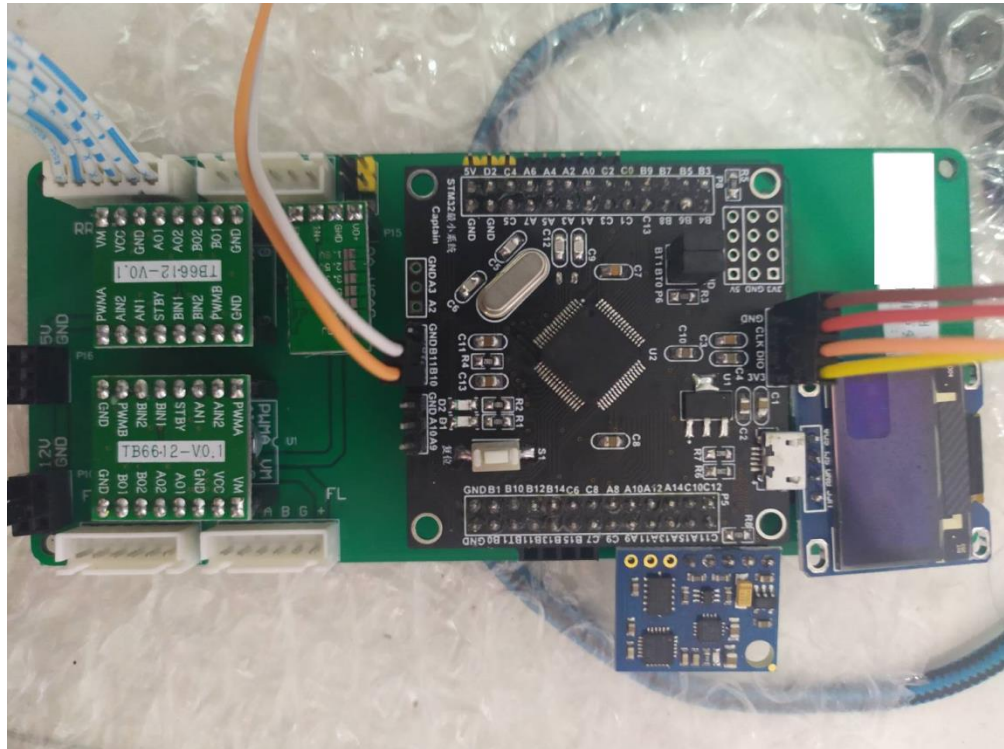
導航機器人進行拆解研究：



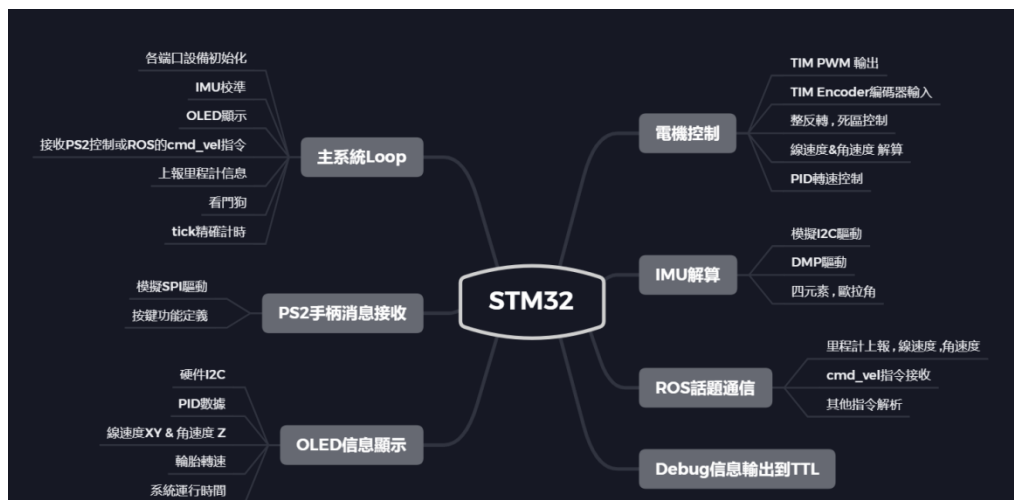
下位機使用 stm32 韌體搭配相關感測器獲取 ROS 所需感測器參數

1. 設定馬達分辨率
2. PWM PID RPM 電機算法整合
3. 差速兩輪運動學控制
4. 並算出機器人移動速度上報給 ROS 系統

元件進程式撰寫及感測器通信差速兩輪運動學演算：

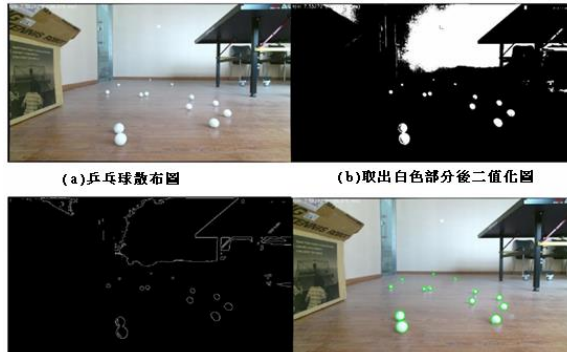


韌體架構圖：

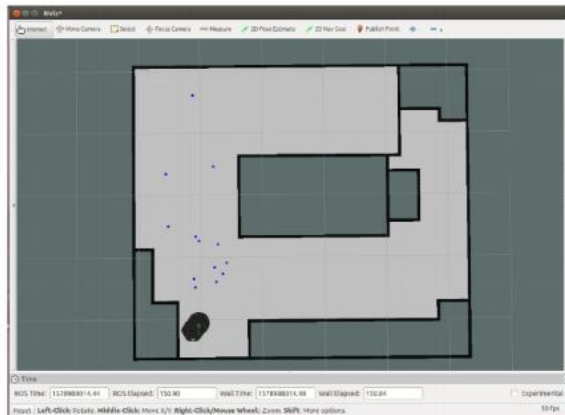


上位機使用樹梅派及 ROS 系統

1. 使用 LiDAR 建立地圖邊界
2. 彩色相機:使用影像處理獲取乒乓球中心



3. 深度相機:獲取乒乓球中心之深度數值



4. 建立位移目標點位並規劃撿球順序
5. 遇到障礙物自動避障並重新規劃位移路徑

論文進度:

碩士論文目前正在進行中，主要分成「上位機」及「下位機」進行開發

(1)下位機:除了 OLED 信息顯示外，其他項目皆已實現並利用 PS2 手柄模擬 ROS 信號。

(2)上位機: 開始與下位機通訊，目前仍持續進行中。