

元 智 大 學

電機通訊學院英語學士班

畢業專題製作成果報告

全方位自走車

Omni bearing Autonomous Mobile Manipulator

組 員：1093704 劉彥狄

1093711 周裕庭

1093717 陳崧凱

1093739 劉姿妤

指導教授：林柏江 助理教授

中 華 民 國 一 一 二 年 九 月

一、	目錄	
一、	目錄.....	2
二、	摘要.....	3
三、	研究動機與目的	3
四、	研究方法及流程	3
	(一) 硬體設定	3
	(二) ROS2 環境設置.....	5
五、	研究結果	8
六、	結論與未來發展	9
七、	參考文獻	9
八、	組員分工方式	9

二、 摘要

隨著科技日新月異的進步以及自動駕駛觀念的引進，許多的汽車產業紛紛在汽車中加入自動駕駛的技術並廣泛的應用，減輕了駕駛人在高速公路上的壓力並增加其安全性。研究中將使用機器人作業系統並搭配自走車的概念將程式導入工具車中，並使工具車能夠成功完成定位、地圖建構、導航以及利用機械手臂與攝影機辨識電梯按鈕並按下按鈕等功能。

三、 研究動機與目的

為了使工具機有智慧的完成定位、地圖建構以及導航等功能，此研究將使用 Ubuntu 20.04[1]以及搭配 ROS(Robot Operating System)2 Foxy[2]的環境並參考 TurtleBot3[3]自走車競賽概念去改造，最後再利用工具機實現定位、地圖建構以及導航的功能，進而完成工具機自駕。

本次研究裡，我們使用 NVIDIA Jetson AGX Orin[4]的微型處理器加裝 ROS2 套件並完成 SLAM 以及導航的功能，此次使用 cartographer 套件包來完成 SLAM 的功能，加上 RoboSense Helios[5]光學雷達掃描周遭環境來建立起地圖，以此來達到工具機自駕的功能。搭配 ZED2 景深攝影機[6]安裝 YOLOv7[7]套件去進行輔助辨識，辨識環境障礙物以及電梯按鈕，利用達明機器人手臂[8]搭配 NVIDIA 處理器，使工具機能在判斷後自行按下電梯按鈕到達我們預設的樓層，來達到不用人為操控的一系列任務。

四、 研究方法及流程

(一) 硬體設定

1. NVIDIA Jetson AGX Orin

NVIDIA Jetson AGX Orin 是一個高效率的微型處理器，如 Fig.1(a)所示。搭載 2048 核心的 NVIDIA 的 GPU 與 12 核心的 ARM 架構之 CPU，且其 CPU 的處理速度可達 1.3 GHz。此台處理器也搭載 64GB 的 DDR5 的 RAM，也同時配置 GPIO 的腳位、DisplayPort、WIFI 與 Type-C 孔等等。運行的作業系統為 Ubuntu 20.04，以使得此處理器能流暢的運行 ROS，並能有效處理深度學習的任務。

2. ZED2 景深攝影機

ZED2 景深攝影機在座深度學習的物體辨識時，是一個相當好用的工具，如 Fig.1(b)所示。此次即須其進行電梯按鈕與環境障礙物的輔助辨識。其景深幀率可為 100HZ、廣度最高可達 20 公尺，其輸出的格式甚至可達 2.2K。綜合前幾項此攝影機之硬體參數，可以完整的完成影像辨識的工作。

3. RoboSense Helios 光學雷達

此項工具車最大的任務，即是需要能夠在建築物內自由行走，並不受人控制。為了達到前項目標，即需要此 RoboSense Helios 光學雷達協助掃描工具車的周遭環境，並建立起一個地圖且規劃出能供工具車自由行走的

路徑，如 Fig.1(c)所示。此雷達能掃描的區域廣度達 150 公尺，且可以 360 度進行旋轉。其盲點區域皆小於 20 公分，且準確度皆介於正負 3 公分。有此工具的搭載，使得環境構建的工作變得相當容易。

4. 達明機器人機械手臂

透過景深攝影機辨識出電梯按鍵後，工具車需要去按壓電梯的按鈕才有辦法上下樓，此時，需要靠車上所搭載的達明機械手臂[8]來完成這項任務，如 Fig.1(d)所示。其手臂能觸及範圍達 1300 毫米，且其附載的能力可達 14 公斤。其控制箱內所搭載之系統也能完美的與 NVIDIA 微型電腦做配合，順利的完成一系列自行行走的任務。

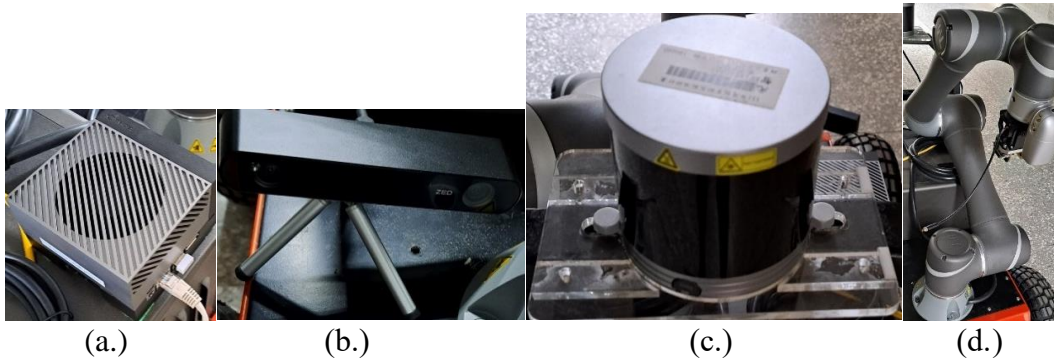


Fig.1 車體主要零組件 (a). NVIDIA Jetson AGX Orin (b). ZED2 景深攝影機 (c). RoboSense 光學雷達 (d). 達明機器人機械手臂

5. 車體組合

前述四項控制器與套件，將分別裝載至工具車的車體上。而光學雷達、達明機械手臂皆是透過內網的方式，來與 NVIDIA 微型電腦來做整合與搭配。為達到前項需求，特在車上加裝一個 HUB，並將 NVIDIA 微型電腦、達明機械手臂與光學雷達用有線網路來做連接。分別將 NVIDIA 微型電腦的 IP 設為 192.168.1.102、光學雷達設為 192.168.1.200 和機械手臂設為 192.168.1.130，形成一個區域內網。接著，透過 NVIDIA 微型電腦上所搭載的 ROS Foxy 去控制達明機械手臂。NVIDIA 微型電腦(/tf)、達明機械手臂(/tm_driver_node)、RoboSense Helios 光學雷達(/rslidar_sdk)與 ZED 2 景深攝影機(/zed2i)彼此間，是由 NVIDIA 微型電腦發出一個 topic，而其餘的零組件為個別的 node，並適時擔任 listener 與 publisher 的角色。最後，再將車體的馬達控制板接到 NVIDIA 微型電腦的 GPIO 腳位，使得 NVIDIA 微型電腦能夠控制車體的馬達，整體自走工具車如 Fig.2 所示。



Fig.2 自走車整體結構(a).左 (b).右 (c).後 (d).前

(二) ROS2 環境設置

1. Talker 節點與 Listener 節點

此次的機器人研究是在 ROS2 環境中開發的，ROS2 不同於 ROS1 的傳輸控制協定與使用者資料包協定(TCPROS/UDPROS)，ROS2 是使用資料分發服務(DDS)通訊協定去運作，DDS 是一個機器對機器的即時通訊中介協定，廣泛的應用於網路、雲端和國防等領域中。ROS2 在機器的通訊機制因為是使用 DDS 協定，讓節點與節點之間只需要設定一個 Domain ID 就能夠即時通訊，但 ROS1 的通訊機制是需要一個主機去連接每個節點，並且都要讓那些節點連接到主機的 IP 位置，相較於 ROS2，ROS1 很容易因為主機故障導致整個系統故障，但 ROS2 的通訊機制因為沒有一個主要的主機，所以當其中一個節點故障時，並不會造成整個系統故障。而在這次研究中，機器人被開發於 ROS2 Foxy 的平台中，首先，在環境設置的部分，參考了 ROS2 以及 Turtlebot3 的官方網站[2][3]，並下載有關 Foxy 的套件包以及 Turtlebot3 的相關導航、地圖建構以及模擬環境 Gazebo 的相關套件包，最後為了讓各個節點能夠成功互相通信，將 Domain ID 設置為 30。

首先，使用 ROS2 Foxy 做一個 Talker 節點與 Listener 節點的測試，確保 ROS2 Foxy 的基礎通信能夠執行。在測試當中，Talker 節點會以特定的頻率發布 Hello World 給 Listener 節點，假如 Foxy 有被成功的安裝，並有成功的設置，Listener 節點則會訂閱 talker 節點發出的主題並發出聽到 Hello World 的消息到主題上，測試結果如 Fig. 3，可以發現到 Talker 節點與 Listener 節點之間的通信狀態。

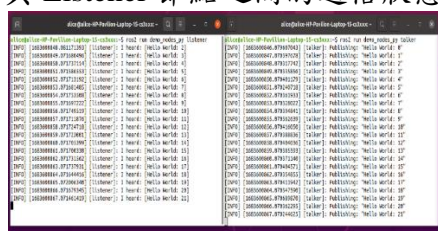


Fig. 3 Talker 節點與 Listener 節點通訊的測試結果

2. Gazebo 環境測試

完成節點通訊後，接著測試 Gazebo 的環境能否正確的執行。在 Gazebo 環境測試中，使用的是 Turtlebot3 中的 Waffle 機器人，並用 Turtlebot3 內建的六角型世界當作預設環境。而在匯入世界之前，必須先使用設置環境變量的指令 export，將環境設置為每次終端機開啟都為 waffle 機器人的指令並寫入.bashrc 檔案中，使其永久生效。寫完環境變量的指令後，接著匯入內建的世界，如 Fig.4，最後將控制 Waffle 機器人的指令寫入並執行，可以成功的讓機器人移動，並顯示出速度以及角度，如 Fig.5。而在此環境中，最後希望能夠將它安裝至研究需要用的 NVIDIA Jetson AGX Orin[4]的微型電腦中，並與自走車連接。

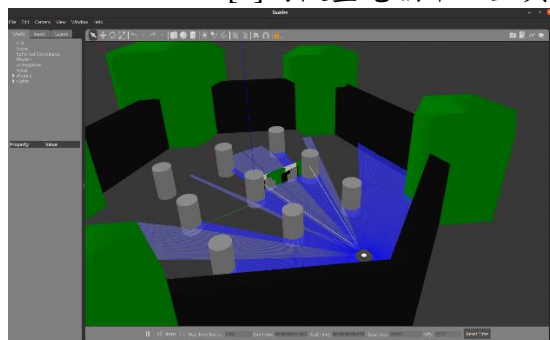


Fig.4 在 Gazebo 中匯入世界的結果

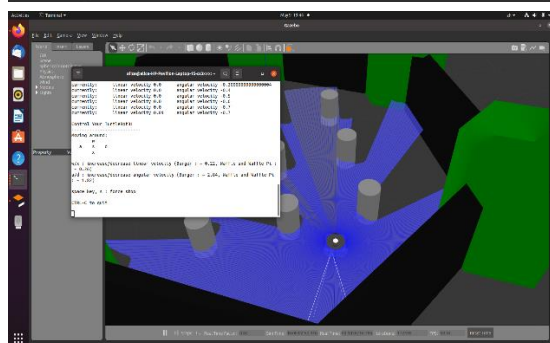


Fig.5 使用鍵盤操控機器人移動方向

3. 自走車 SLAM 與導航

在此研究中將參考 Turtlebot3 的官方網頁去實現自走車 SLAM 以及導航的功能。SLAM 是一項透過感測器估計當前位置加上建造地圖的技術，其中的定位功能又分為視覺定位以及雷達定位，視覺定位是利用機器人的攝影機鏡頭去進行周遭物件的收集，而雷達定位主要是用光學雷達掃描周圍的環境並進行地圖的建構。這次使用的是 cartographer 套件包來完成 SLAM 的功能，cartographer 是一個在跨 2D 以及 3D 的平台與雷達配置的 SLAM 系統，讓開發者能夠將此系統應用於開發自走車上面，完成 SLAM 並得到地圖以及自走車目前姿態的相關資訊後，再接著執行讓自走車自動導航的程式。以 Turtlebot3 的房間地圖為例，首先，將預設為 waffle 機器人的指令寫進.bashrc 檔裡面，使得每次在執行套件包的程式時不用特別在執行一項設定機器人類型的指令，接著使用 ROS2 的指令打開 Gazebo 環境，接著執行 SLAM 的程式進行地圖建構並打開 Rviz 以及開啟內建的鏡

頭，可以同時看到機器人所看到的視角，如 Fig.6 與 Fig.7。

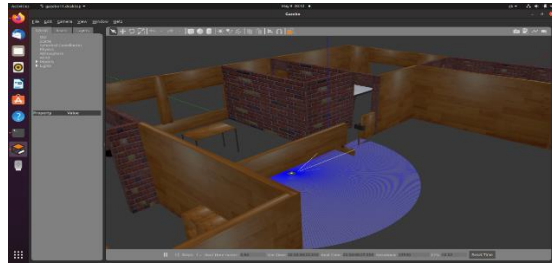


Fig.6 以內建的房間地圖為例的機器人SLAM過程

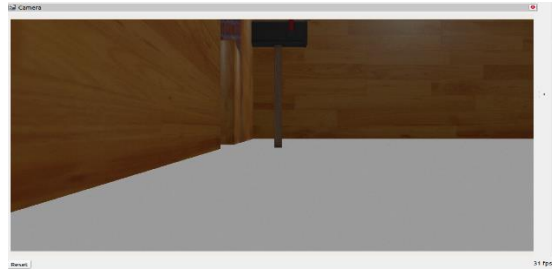


Fig.7 Rviz 內建相機看到的視角

研究中，機器人的 SLAM 是運用 Teleop 指令並使用鍵盤控制方向與速度，最後，為了方便之後導航功能的執行，所以在讓機器人導航之前，會先打一行程式將建構出來的地圖儲存起來，也就是 Fig.8 方便之後執行導航時能夠直接將地圖匯入。

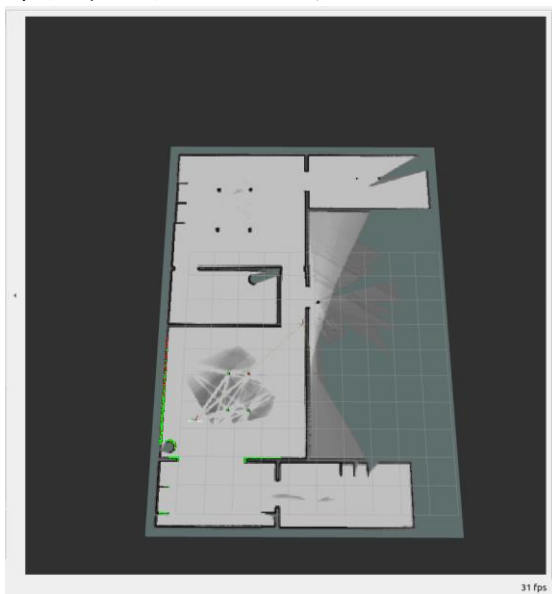


Fig.8 機器人利用 SLAM 建構出來的地圖

建構以及儲存完地圖後，接著就是使機器人能夠導航到目標中。以 Turtlebot3 內建的房間地圖為例，在 SLAM 完後接著執行導航的程式碼，並開啟 Rviz，方便開發者對照 Gazebo 世界以及 Rviz 可視化地圖，而當導航的環境設置完成時，接著會手動點選 Rviz 上方的 Navigation2 goal 來設置機器人所要到達的目的地。此時，機器人會根據剛才 SLAM 好的地圖進行路線規劃並且到達指定的位置，如 Fig.9，而在研究中希望能夠將以上的 SLAM 跟導航的模擬與我們的自走車做結合並將 ROS2 系統安裝至

NVIDIA Jetson AGX Orin[4]的微型電腦中，並連接至目前研究中的全方位自走車中。

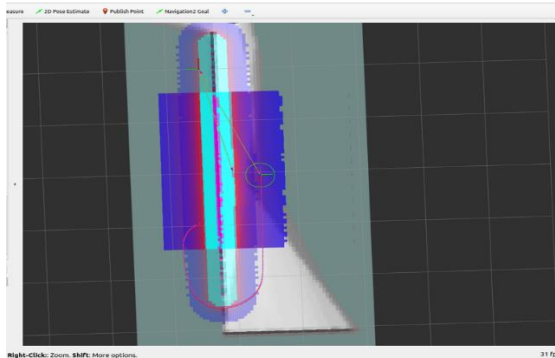


Fig.9 機器人導航模擬圖

4. 機械手臂按電梯按鈕

當工具車走到電梯門口後，ZED 景深攝影機拍攝的畫面若透過 YOLOv7 分析出電梯的按鈕後，將會傳送訊息給達明機械手臂來按電梯的按鈕。當偵測到按鈕順利變色時，即會再傳送一個訊號給機械手臂，通知其回復原本手臂縮回的狀態。再使得車體能夠進到電梯的車廂內，並再透過攝影機所拍攝出的畫面辨識欲前往的樓層的按鈕，並使手臂去按壓該按鈕，如 Fig.10。



Fig.10 機械手臂按壓電梯按鈕

五、 研究結果

本研究在自走車上安裝了 NVIDIA Jetson AGX Orin 微處理器，並同時整合了 ZED2 景深攝影機、RoboSense Helios 光學雷達還有 ROS2 套件，透過這些不同功用與目地的感測器，以得到自走車附近的各項數據與資料，包括視覺影像、深度信息和 3D 點陣圖，實現同時定位和建圖（SLAM）以及導航的功能，實時更新地圖，以應對動態環境中的變化。

首先，系統的高精度定位和建圖功能使自走車能夠在環境中準確地定位自身位置並建立地圖。這對於自走車的自主導航至關重要，能夠避免碰撞和碰壁，並確保安全運行。

系統的多感測模組，使得自走車能夠獲得更全面的環境信息。視覺影像、

深度信息和雷達數據的結合提供了更豐富的感知視角，使自走車能夠容易地理解和分析周圍環境，完成障礙物辨識、路徑規劃和障礙物避免等任務。

此外，系統的環境辨識和電梯按鈕辨識功能使得自走車能夠感知與識別環境中的重要目標。這提供了自走車更高的便利性，使之能夠更好地在不同的環境與樓層進行移動。

六、 結論與未來發展

在未來，我們希望能繼續優化感測器的各項參數，以更精確的辨識出周圍的地景與物體。且以符合應用在不同的環境與光線中，同時也需要再學習與辨識更多不同的物件，擴展應用範圍與能力。

七、 參考文獻

- [1] Ubuntu 20.04. (2018). Ubuntu 20.04.6 LTS (Focal Fossa). [Online] Available: <http://www.releases.ubuntu.com/20.04/>, Accessed on: May. 2023.
- [2] ROS2 Foxy. (2023). ROS 2 Documentation. [Online] Available: <https://docs.ros.org/en/foxy/index.html>, Accessed on: May. 2023.
- [3] TurtleBot3. (2023). [Online] Available: <https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>, Accessed on: May. 2023.
- [4] NVIDIA Corporation. (2023). NVIDIA Jetson AGX Orin Developer Kit User Guide. [Online] Available: <https://developer.nvidia.com/embedded/learn/jetson-agx-orin-devkit-user-guide/index.html>, Accessed on: May. 2023.
- [5] RoboSense. (2023). RoboSense RS-Helios. [Online] Available: <https://www.robosense.ai/en/rslidar/RS-Helios>, Accessed on: May. 2023
- [6] Stereolabs Inc. (2023). Meet ZED 2. [Online] Available: <https://www.stereolabs.com/zed-2/>, Accessed on: May. 2023
- [7] L. Lia, “YOLOv7 on Windows (2). [Online] Available: https://www.larrysprognotes.com/YOLOv7_2/, Jul. 2022 (accessed Apr. 2023).
- [8] TECHMAN ROBOT INC. (2019). 達明機器人機械手臂 TM12. [Online] Available: <https://www.tm-robot.com/zh-hant/tm12/>, Accessed on: May. 2023

八、 組員分工方式

學號	姓名	主要工作	貢獻比例
1093704	劉彥狄	整理所有實驗資料並彙整成報告	20%
1093711	周裕庭	整理所有實驗資料並彙整成報告	20%
1093717	陳崧凱	負責硬體串接、ROS 系統全部整合	30%
1093739	劉姿妤	ROS 自走程式庫撰寫、影像辨識模型建立	30%