PORTFOLIO

LIST OF CONTENTS

introducing
ABOUT ME

An individual who has a deep and abiding passion for robotics.

personal SKILLS

An individual proficient in programming languages such as C++ and Python, well-versed in middleware (ROS 1 & 2), and knowledgeable in Linux commands.

C++

Python

ROS 1

ROS 2

EDUCATION

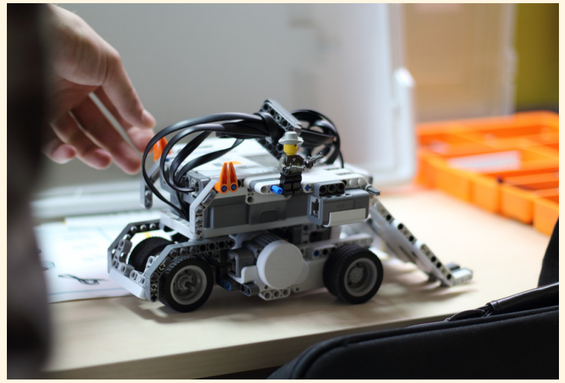
2019-2023

Bachelor of Mechanical Engineering

Nanyang Technological University
Graduated in Robotics and Mechatronics

work
EXPERIENCE

A fresh graduate with internship experience in Robotics and a good collaborator from my past working experience.



Research Intern

Assistant Construction Site Supervisor

project

PORTFOLIO

Numerous robotics-related projects throughout my internship, Final Year Project, and various academic courses.

01 **Project**
Multi-SLAM system

02 **Project**
Drone Object Avoidance System

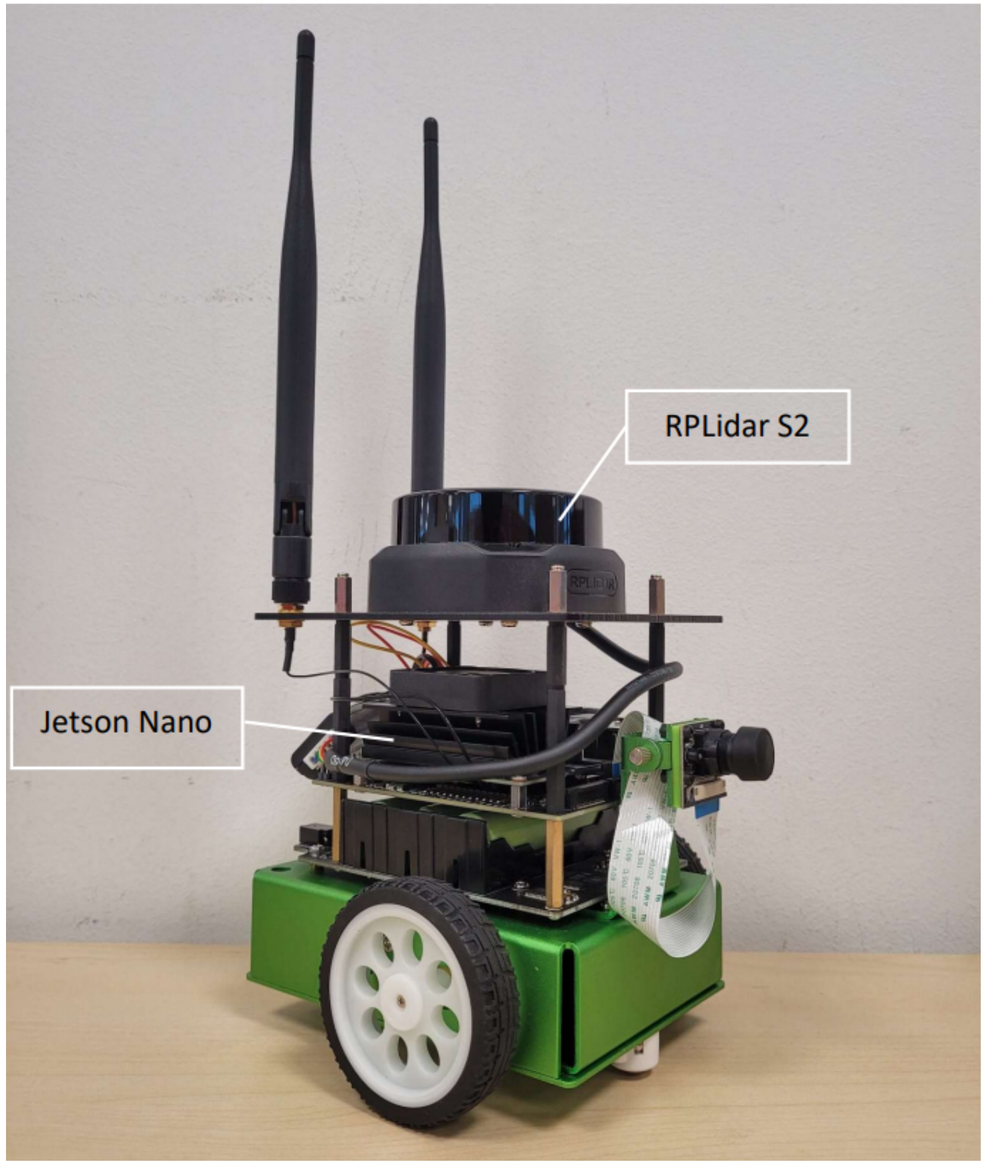
03 **Project**
Autonomous Multi-Robot Navigation and Mapping for Swarm Applications

04 **Project**
Dish Washer Robotic Hand

PROJECT 01

Multi-SLAM system (Internship)

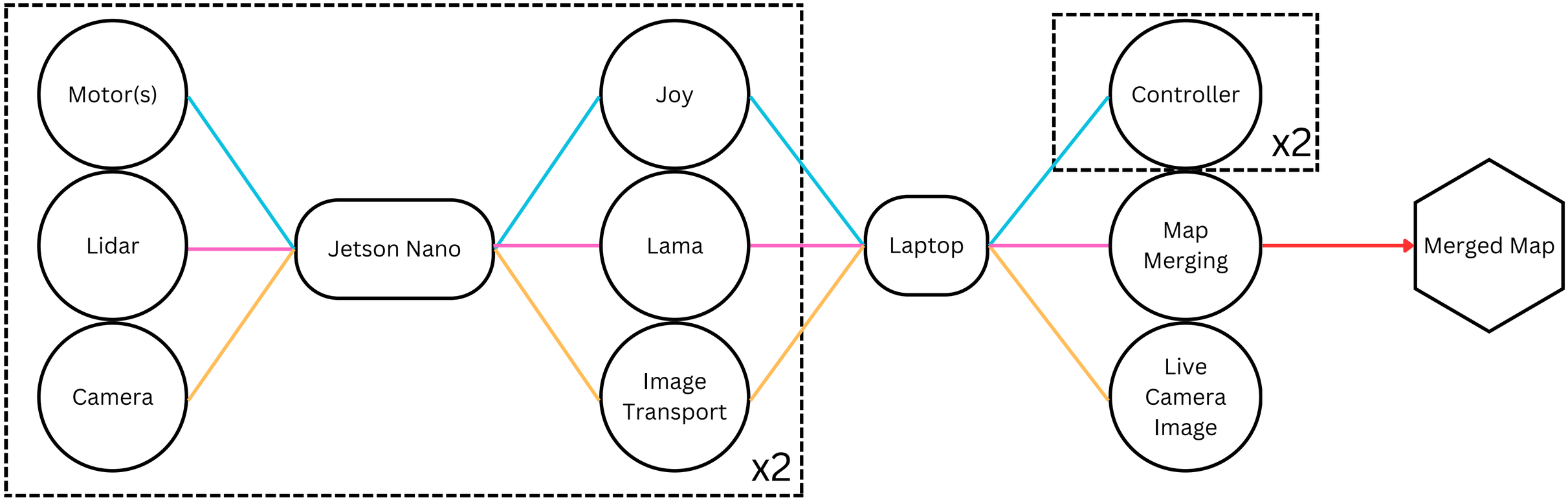
- Constructed two identical robots, each equipped with **Jetson Nano** and **RPLidar S2**.
- Implemented control mechanisms using open-source ROS packages like **joy** and **image_transport**, while integrating the **lama_ros** package for SLAM algorithm functionality.
- Reconstructed **map merging algorithm** from academic paper.



PROJECT 01

Multi-SLAM system (Internship)

Work Flow of the system

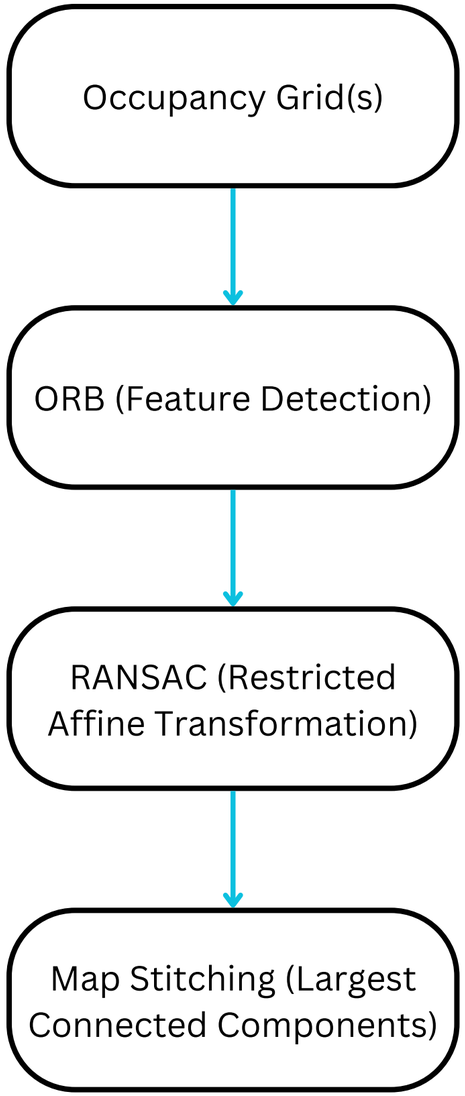


PROJECT 01

Multi-SLAM system (Internship)

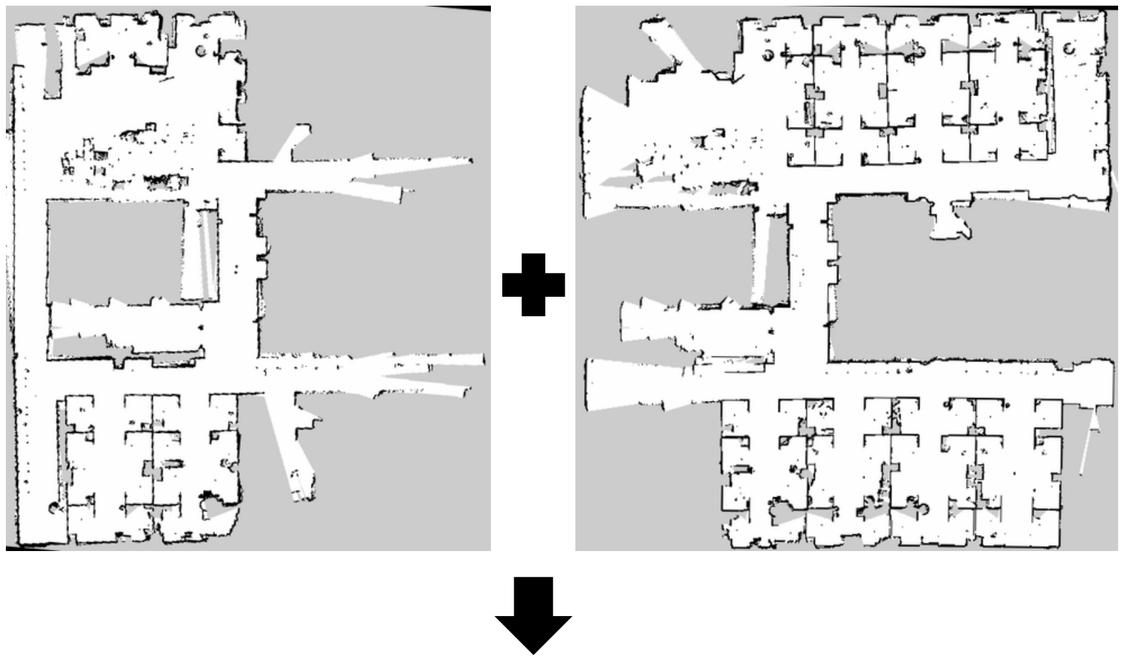
1. Map merging method for **2D SLAM (Occupancy Grid)**
2. Packages used for map merging algorithm
   - Tf2
   - Opencv2

Work Flow of Map Merging Algorithm



PROJECT 01

Multi-SLAM system (Internship)



Comparison with single SLAM system using the same algorithm (Lama ROS)

- Reduced mapping time
- Few overlapping areas → deceptive merged map
- The final quality is affected by individual maps (noises stacked up)

PROJECT 02

Drone Object Avoidance System (Internship)

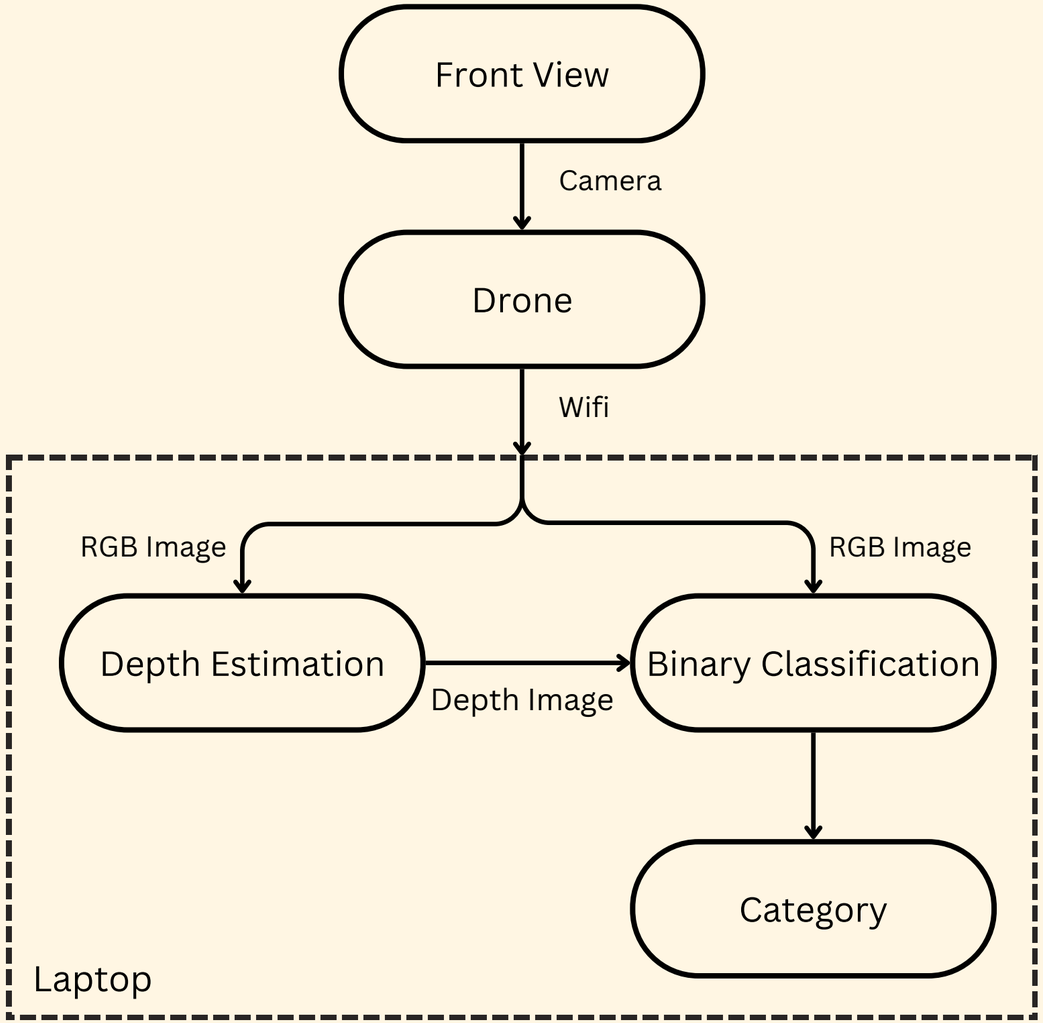
- Developed **deep learning model** for obstacle avoidance.
- **Outsource drone** from ShenZhen DJI.
- Utilized libraries such as **DJITelloPy**, **TensorFlow Keras**, **OpenCV**, **Pygame** and others.

PROJECT 02

Drone Object Avoidance System (Internship)

- **Two** sub-algorithm
  a. Depth estimate (**Fastdepth** / **Midas**)
  b. Binary image classification (**CNN**)

Work Flow of the system



PROJECT 02

Drone Object Avoidance System (Internship)

- CNN with **sigmoid activation** as last layer
- **BinaryCrossentropy** as lost function
- **Adam** optimizer
- 200 epochs
- Confident threshold - 0.7

Layer Summary (Binary Classification)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 256, 256, 16)	592
leaky_re_lu (LeakyReLU)	(None, 256, 256, 16)	0
max_pooling2d (MaxPooling2D)	(None, 128, 128, 16)	0
conv2d_1 (Conv2D)	(None, 128, 128, 32)	4640
leaky_re_lu_1 (LeakyReLU)	(None, 128, 128, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_2 (Conv2D)	(None, 64, 64, 64)	18496
leaky_re_lu_2 (LeakyReLU)	(None, 64, 64, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 64)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 64)	0
dense (Dense)	(None, 128)	8320
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 128)	16512
leaky_re_lu_3 (LeakyReLU)	(None, 128)	0
dense_2 (Dense)	(None, 1)	129

PROJECT 02

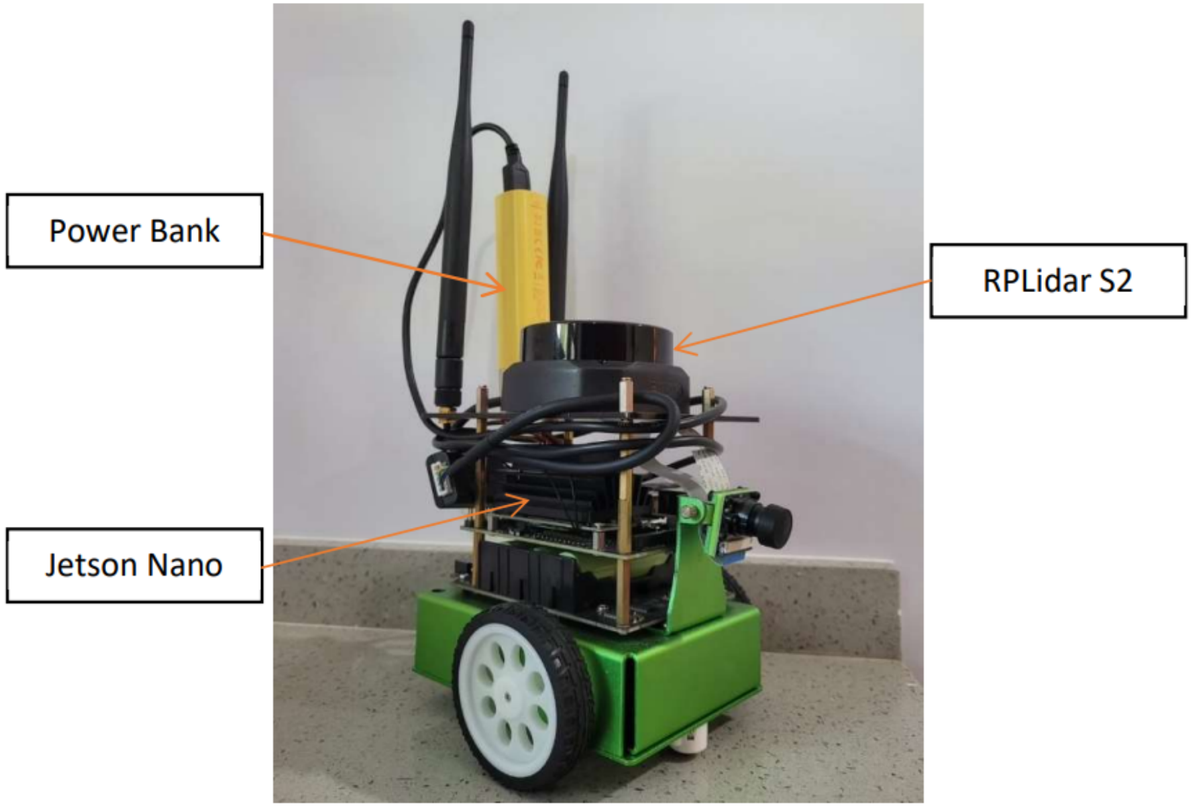
Drone Object Avoidance System (Internship)

- **Three input** configurations (RGB, Depth, Combined [RGB + Depth])

RGB

Depth

Combined

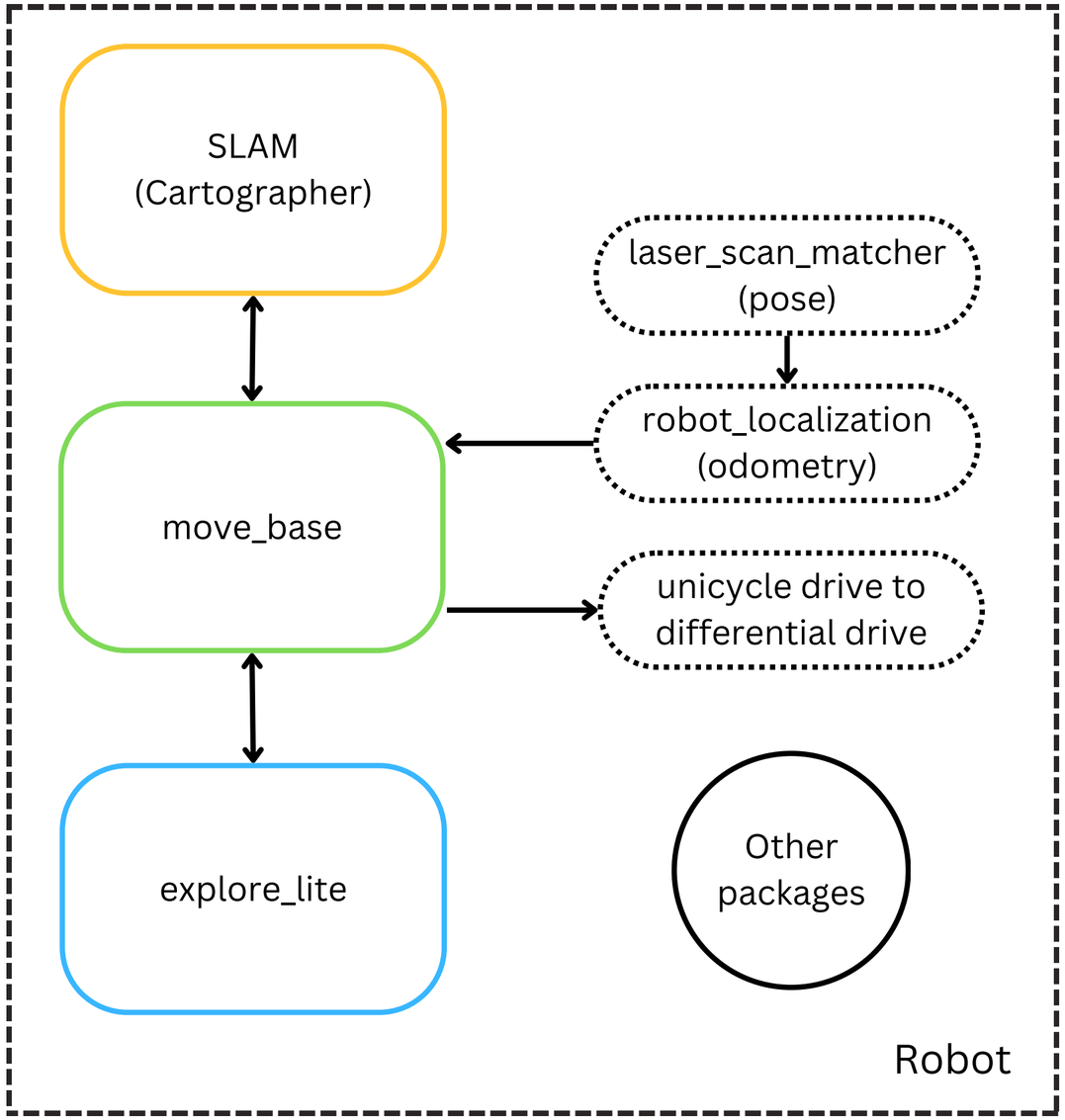


PROJECT 03

Autonomous Multi-Robot Navigation and Mapping for Swarm Applications (Final Year Project)

- **Industry attached FYP** with Temasek Lab (Internship organisation).
- **Robots inherited** from multi-SLAM system (internship).
- Utilized ROS packages such as **move_base**, **cartographer**, **laser_scan_matcher**, **robot localization**, **multirobot_map_merge** and self-written packages.
- SWARM for **map exploration** purposes.

Simplified ROS packages graph



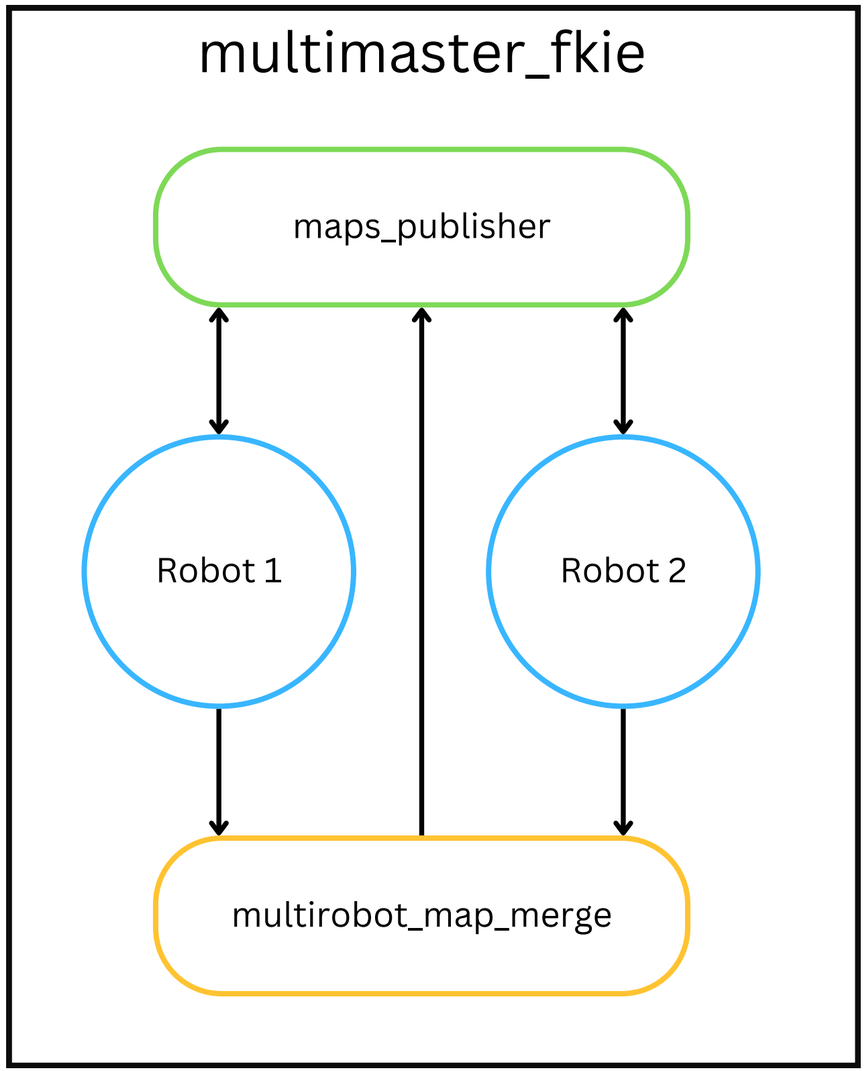
PROJECT 03

Autonomous Multi-Robot Navigation and Mapping for Swarm Applications (Final Year Project)

Each robot is equipped **THREE** main packages :

1. **explore_lite** : guides the Jetbot
2. **move_base** : provides navigation commands
3. **cartographer** : generates a map

SWARM ROS packages graph



PROJECT 03

Autonomous Multi-Robot Navigation and Mapping for Swarm Applications (Final Year Project)

SWARM added **THREE** nodes on previous foundation

1. **multirobot_map_merge** : map merging
2. **maps_publisher** : publish cost map as original map or map full of obstacles
3. **multimaster_fkie** : distributed system

PROJECT 03

Autonomous Multi-Robot Navigation and Mapping for Swarm Applications (Final Year Project)

Modification of open-source code

- Explore_lite is configured to recognize only three conditions (**unknown**, **obstacles**, **free**) by analyzing the occupancy grid map.
- The cartographer produces a **map** containing **varied values**.
- Additional steps of **sorting** and **categorize** is added to the algorithm.

Maps Generated

- Single robot autonomous exploration takes 166 seconds.
- Swarm exploration takes 127 seconds, a **23.5% time saved**.
- Time needed for swarm exploration depends on map merging algorithm, feature detection and overlapping area.

Limitations

- **High computing power** to process odometry and path planning.
- The potential for **premature map merging**.

PROJECT 03

Autonomous Multi-Robot Navigation and Mapping for Swarm Applications (Final Year Project)

Short Story of this Final Year Project

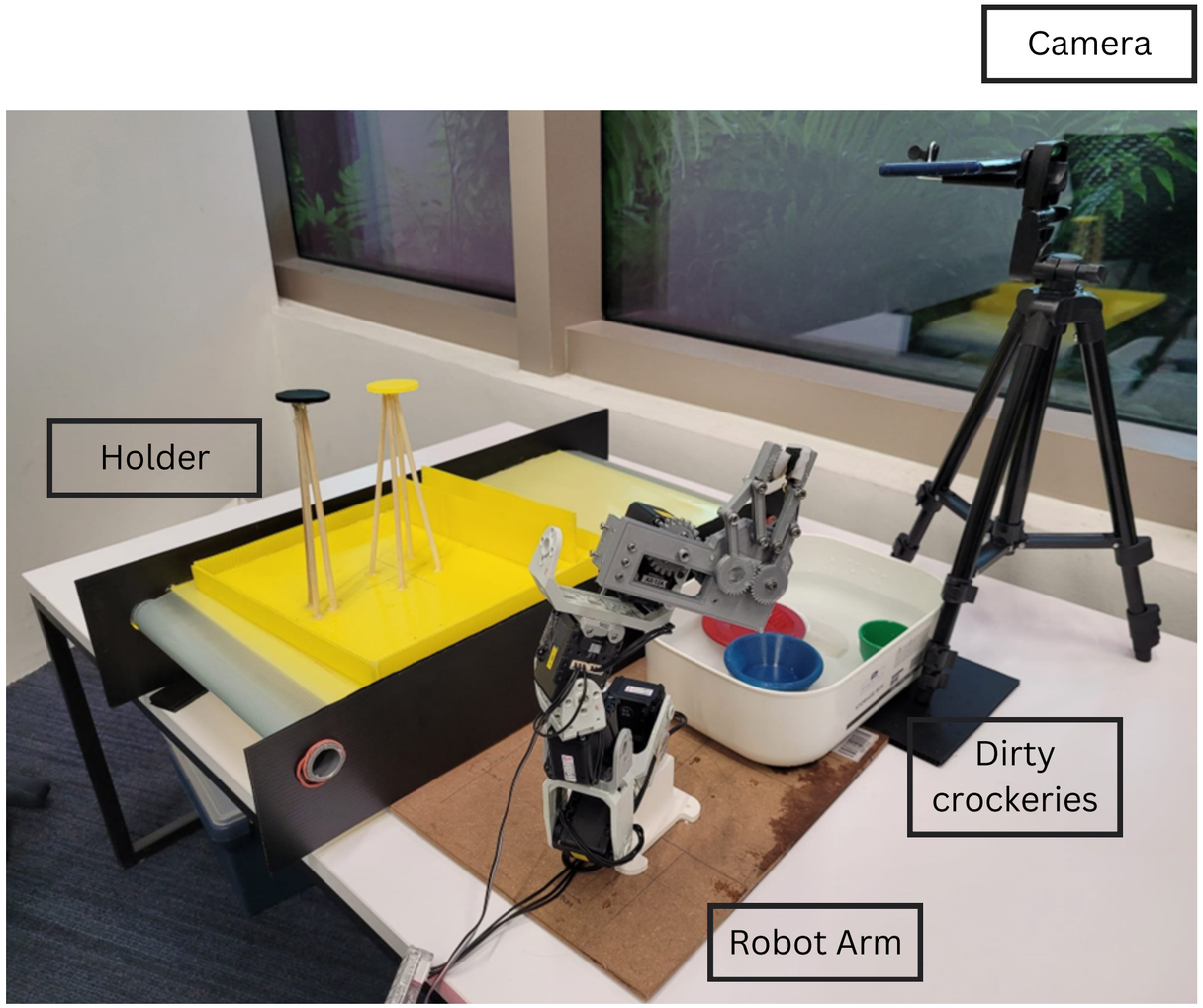
- Initially, the intention was to built a SWARM for **3D map exploration.**
- The chosen sensor for this purpose was the **Zed 2i stereo camera**.
- However, this plan faced challenges as the **computing power** of the Jetson Nano proved **insufficient**, leading to an inability to achieve the 3D map exploration.
- Consequently, a **shift** was made to a 2D map exploration approach utilizing the **RPLidar S2**.

Knowledge learned due to this unique experience

- Implementation of **pointcloud2** and **3D SLAM** algorithm.
- **Fusion of sensor data**, combining IMU and odometry information through Extended Kalman Filter (EKF).

PROJECT 04

Dish Washer Robotic Hand



- Constructed a robotic arm equipped with **Dynamixel motors** and devised a system for picking up and sorting pre-identified crockeries.
- Implemented crockeries recognition using a **camera.**
- Utilized the OpenCV **Hough Circle Transform** for accurate crockeries (circle) identification.
- Employed **pixel colour** analysis to distinguish between different types of crockeries such as cups, plates, and bowls.
- Integrated ROS **MoveIt** for efficient pick-and-place operations.

PROJECT 04

Dish Washer Robotic Hand

My Contribution in this group project

- Crockeries detection using **OpenCV**
- **Hough Circle Transform** and **pixel identification**

Code

https://github.com/cheesern/crockeries_detection.git

Competition

PORTFOLIO

Participated in a university-exclusive robotics competition organized specifically for my cohort

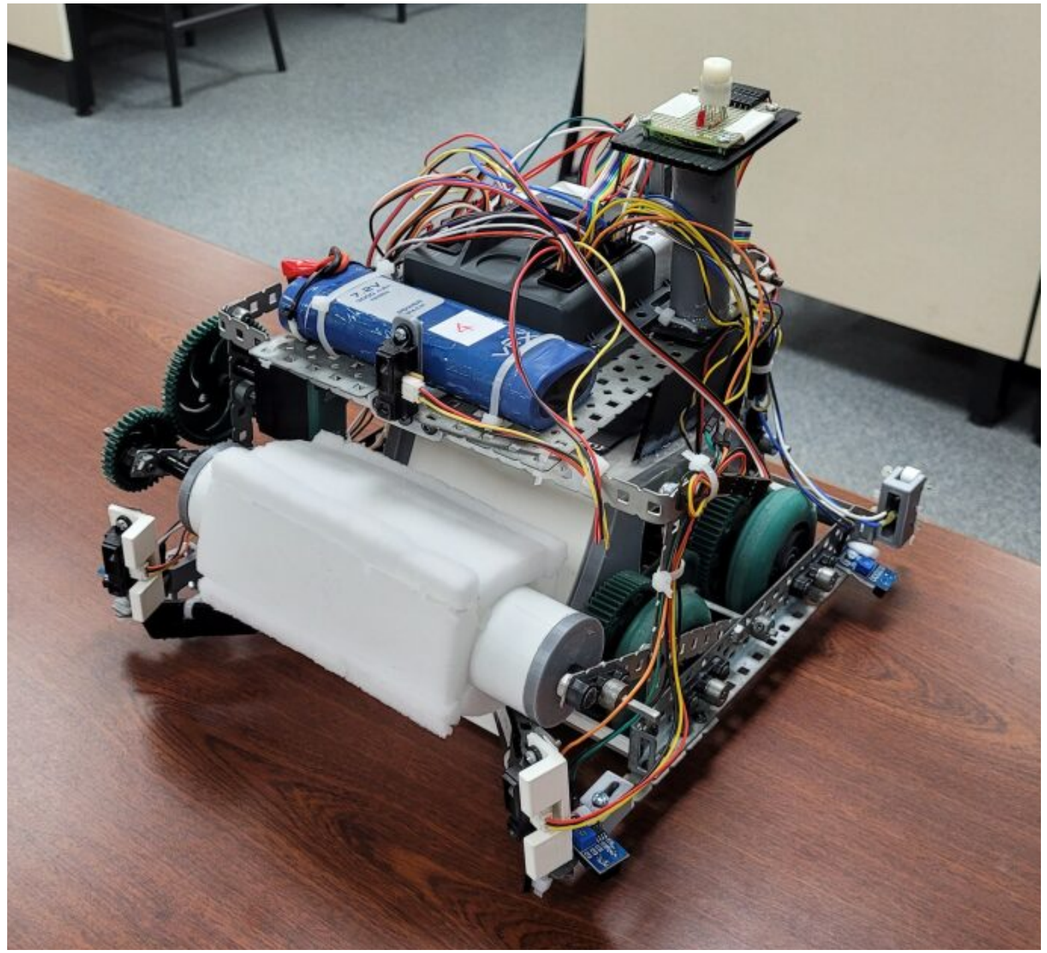
01

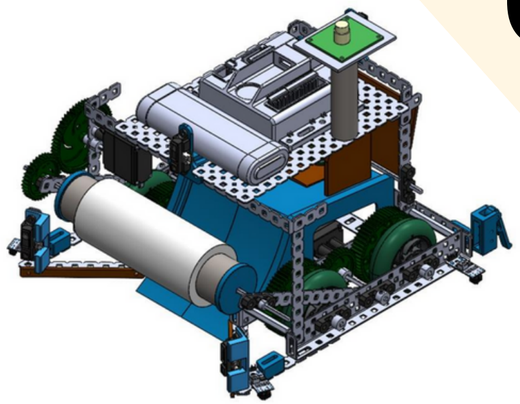
Competition

Small Autonomous Ball Collector Vehicle

01 Competition

Small Autonomous Ball Collector Vehicle



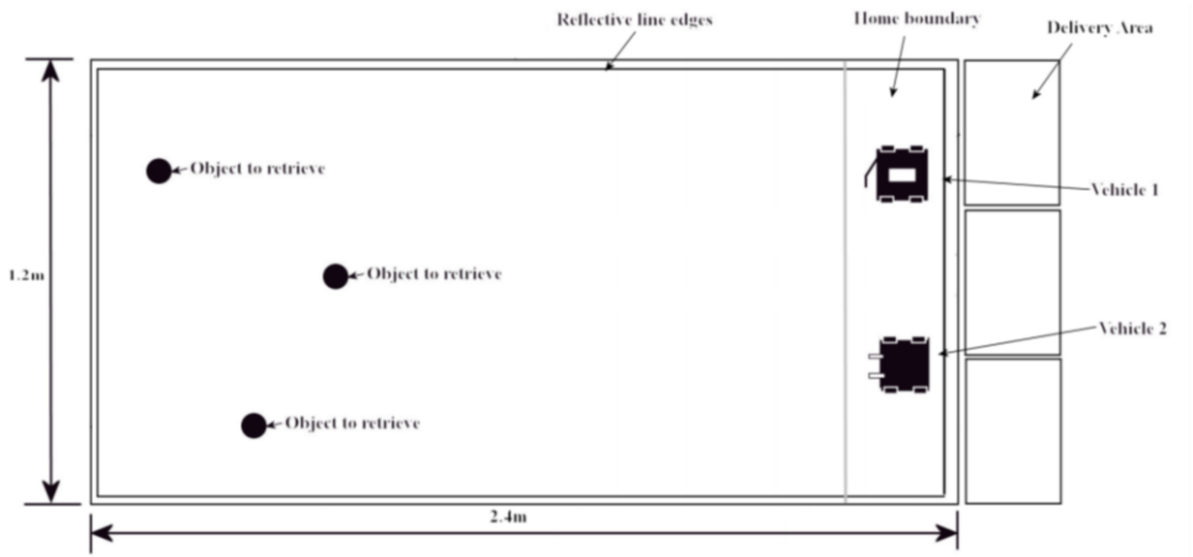


- Engineered a compact autonomous vehicle entirely from the ground up, incorporating a **VEX controller** and **various sensors**.
- Collaborated with a team of 7 members to develop an autonomous vehicle with functionalities including **ball collection**, **edge detection**, and **object avoidance**.
- Achieved **1st place** in an autonomous ball collection competition, showcasing the vehicle's innovative navigation system.

01 Competition

Small Autonomous Ball Collector Vehicle

Competition Stage

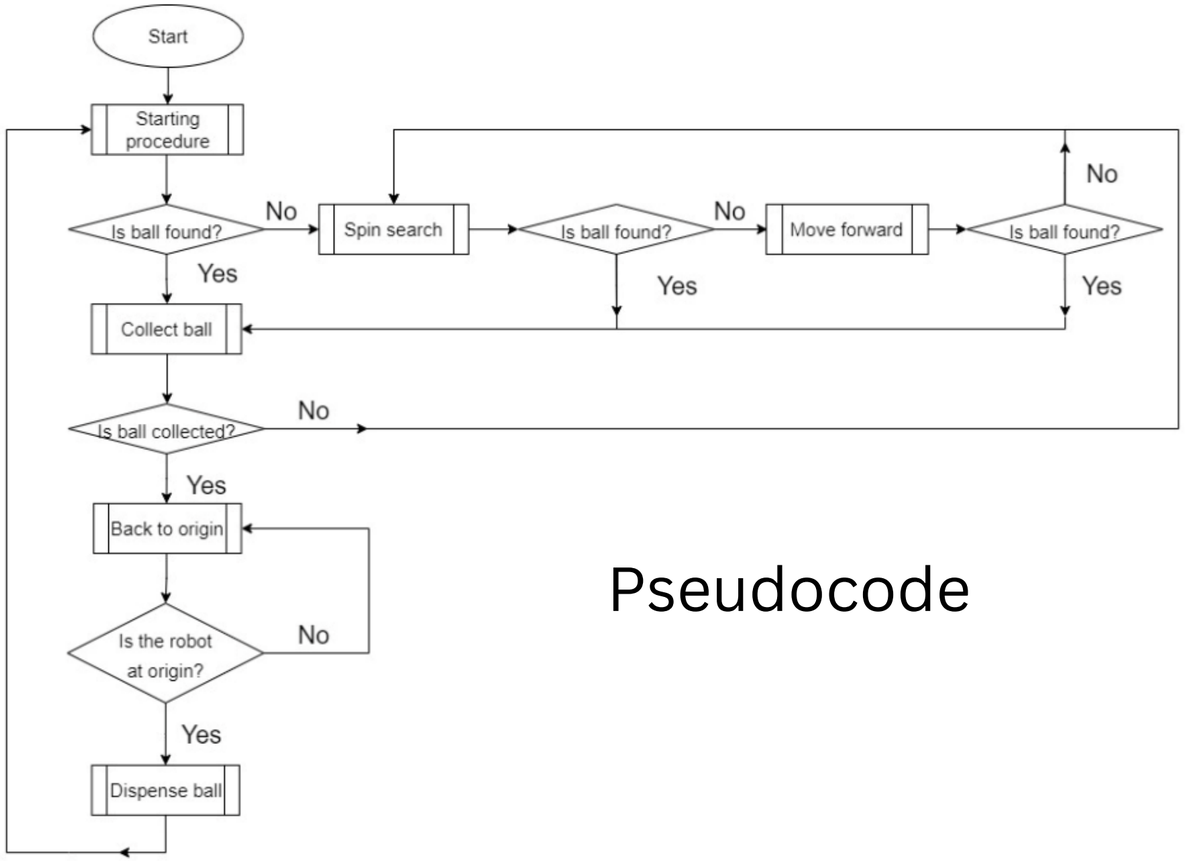


My Contribution

- Hardware design
- Sensors calibration

Vehicle Codes
https://github.com/Khoo395/Tennis_Ball_Collector.git

Full Report
https://github.com/cheesern/ball-collector/tree/main



Pseudocode

THANKS FOR WATCHING

Have a nice day!