Graduation thesis

AUTONOMOUS MOBILE ROBOT USING LIDAR SENSOR

Suppervisor: Assoc. Prof. Nguyen Thi Lan Huong

Students: Nguyen Thai Nguyen - 20192241

Table of contents

1. Introduction

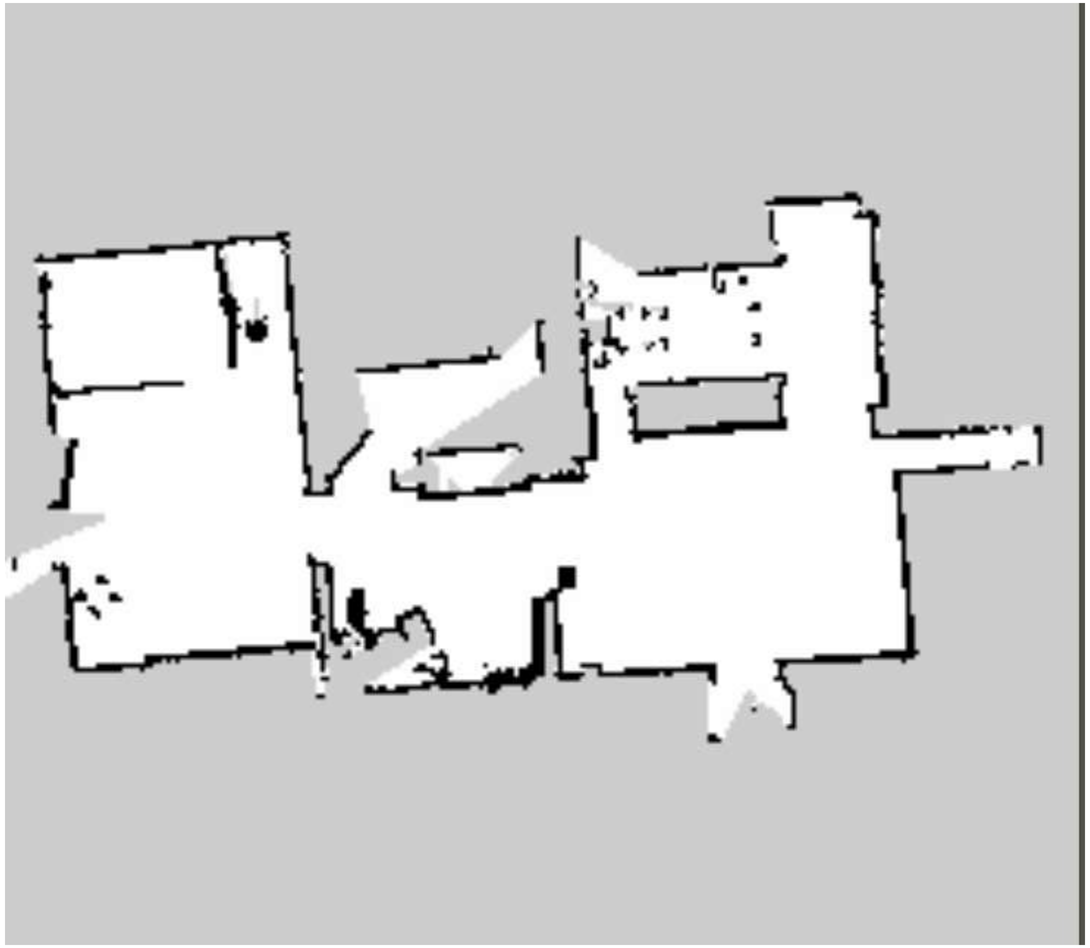
1.1 introduction - SLAM

Simultaneous

Localization

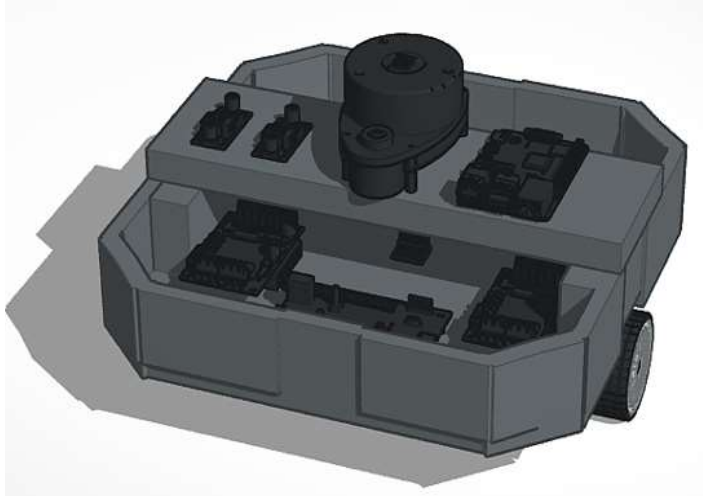
And

Mapping



Figures 1.1. Apply slam to build a map.

1.2 introduction – the goals of project



Figures 1.2. 3d robot model design.

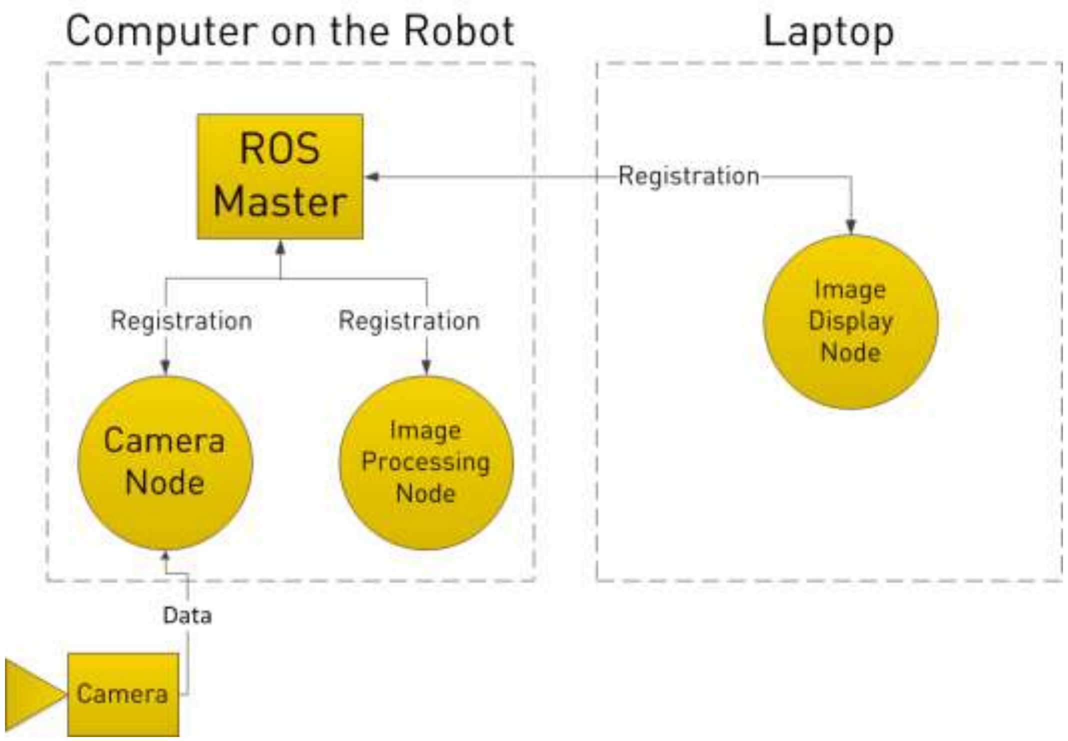
- Simulate and navigate robots on Gazebo and Rviz.
- Robots are capable of building a map in unknown space.
- Robot can find the shortest path to the destination.
- In addition, during movement, the robot can avoid fixed obstacles and react appropriately to avoid moving obstacles.

1.3 Robot operating system (ROS)

ROS definition: The Robot Operating System (ROS) is an open-source framework.

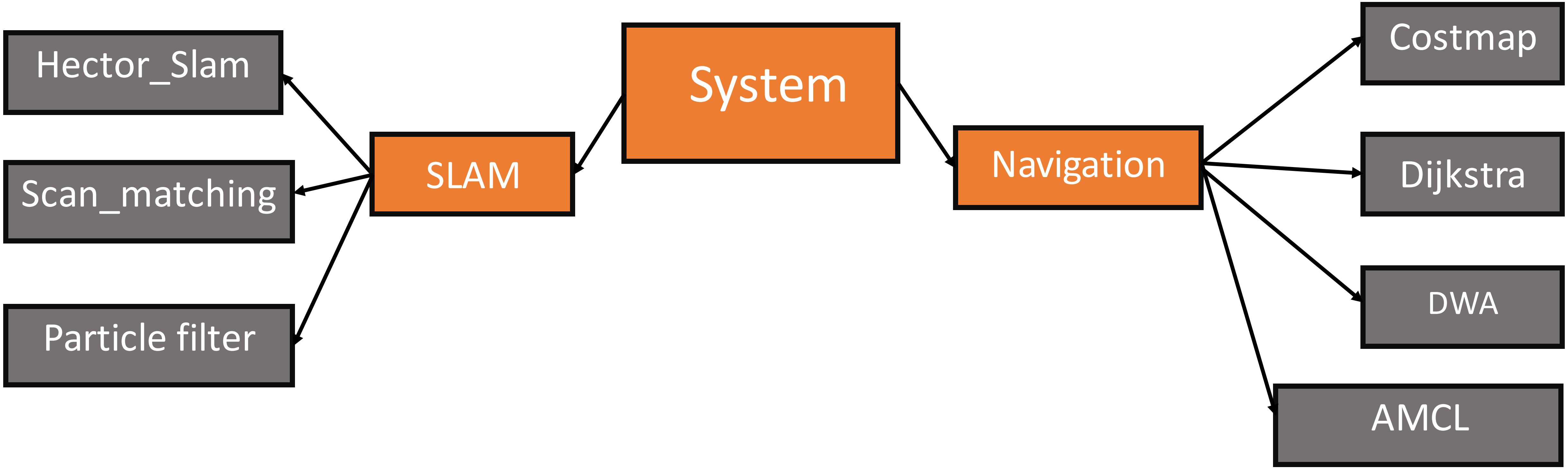
Why ROS was chosen in this system:

- Ros can be divided into open-source packages that can be downloaded and reuse.
- Nodes in ROS do not have to be on the same system.



How ROS work: a ROS system can be divided into nodes and each node must register with ROS master to communicate with other nodes

1.4 introduction- system overview



2. Hardware design

2.1 Hardware design-block diagram

The function of main blocks:

- Source: include 5V supply for raspberry pi, sensor,stm32f4 and 12V block supply for motors.
- Sensors: include Lidar sensor supply the distance of obstacles around robot, and the IMU senso help determine the angle and direction of robot
- Computers: Raspberry pi as intermediary processor between laptop and robot. laptop is used to control and monitor the robot
- Stm32f4 is used to control the robot speed, and continuously calculate the odometry from the IMU, and the number of pulses of the stepper motor, then send it back to the raspberry pi

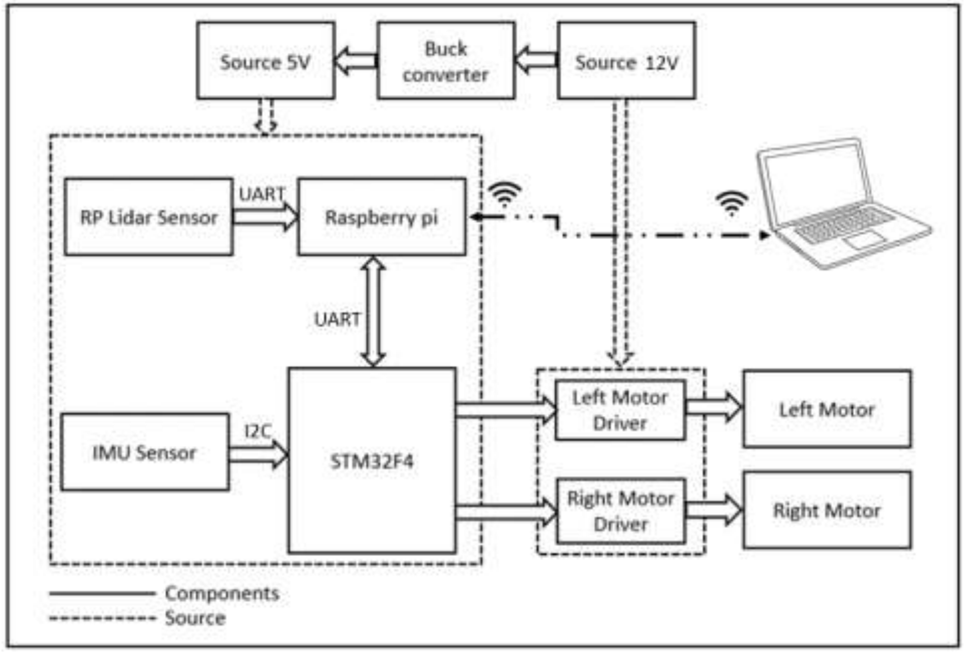


Figure 2.1. block diagram.

2.2 Hardware design – wiring diagram

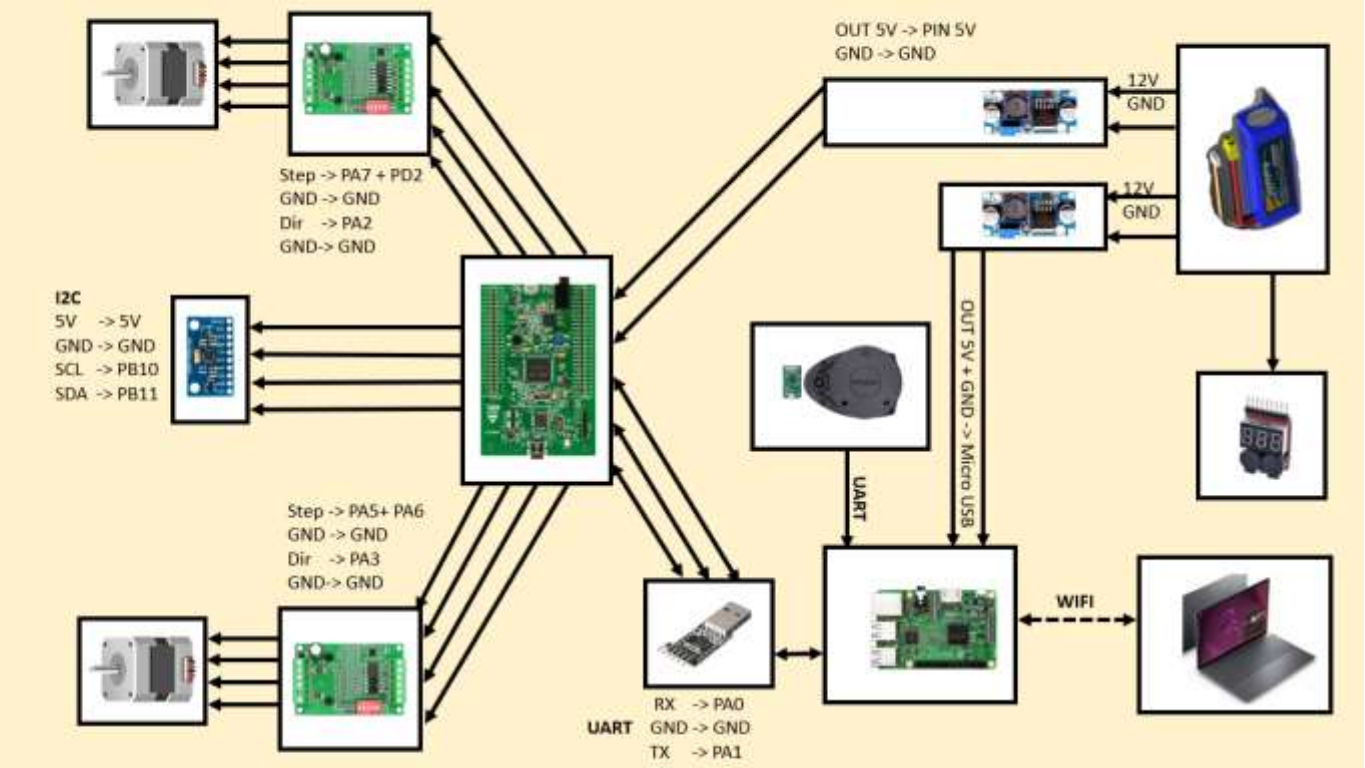


Figure 2.1. wiring diagram.

The connection:

- MPU9250 ~ STM32f4: I2C protocol
- Rplidar A1 ~ Raspberry pi: UART protocol
- STM32f4 ~ Raspberry pi: UART protocol
- Raspberry pi ~ Laptop: Wi-Fi network

2.3 hardware design- PCB design.

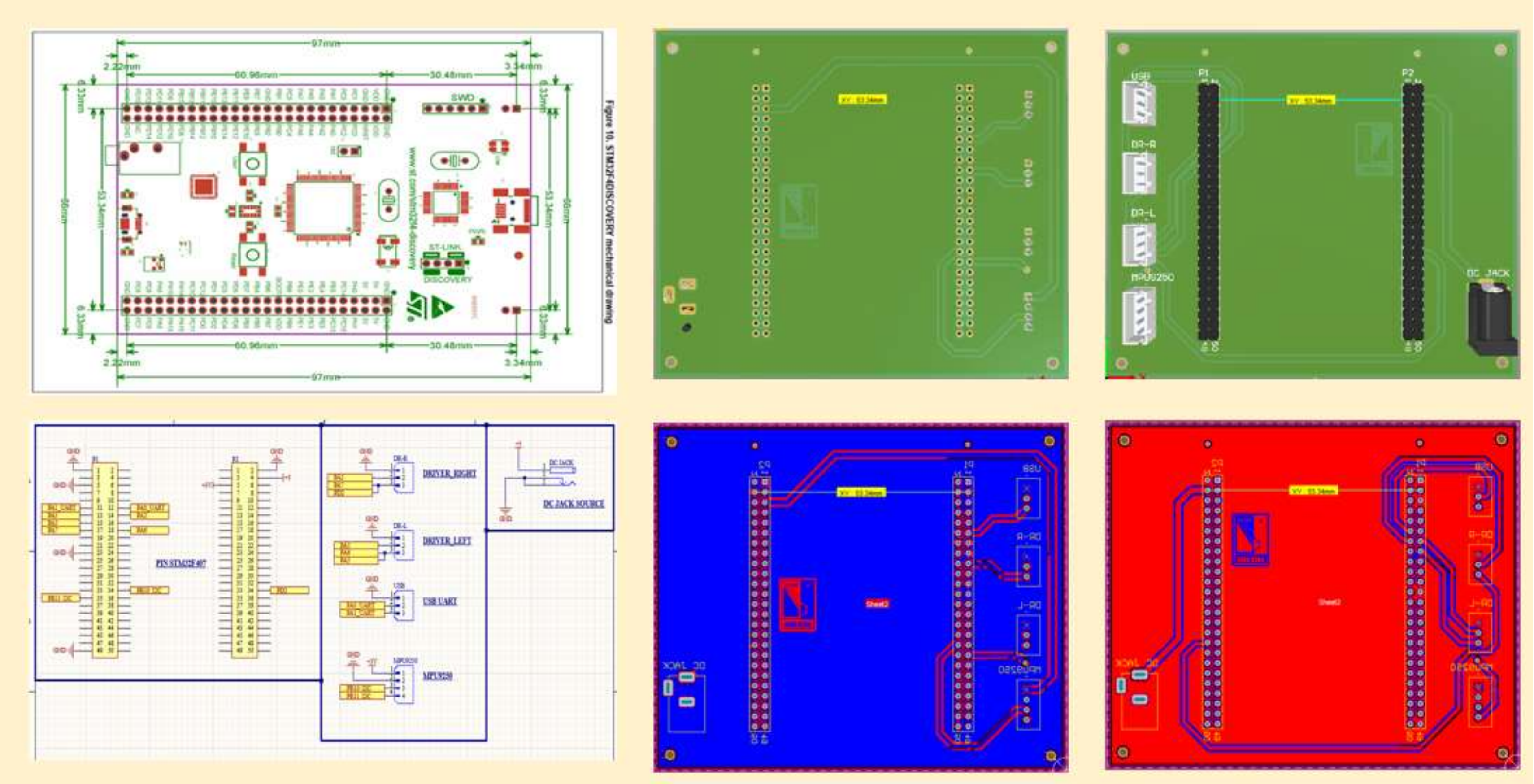


Figure 3.1. PCB design in Altium.

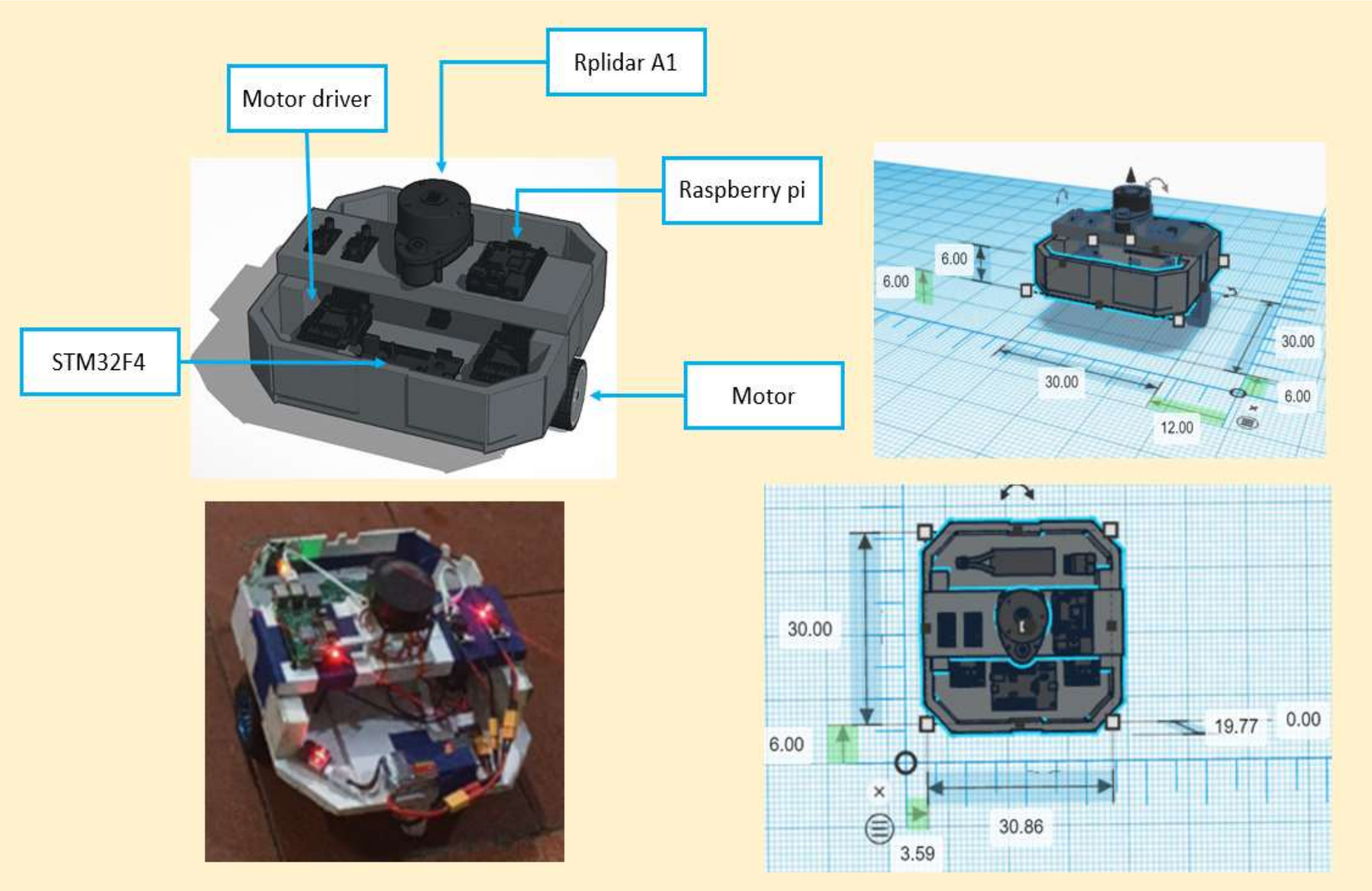
2.4 Hardware design – frame robot

Main components:

Stm32F4, Raspberry pi, Rplidar A1, Motors.

Robot dimensions:

features	value
Diameter	300 mm
Robot height	198 mm
wheel distance	280 mm
wheel diameter	65 mm



3. Software design

3.1 Software design

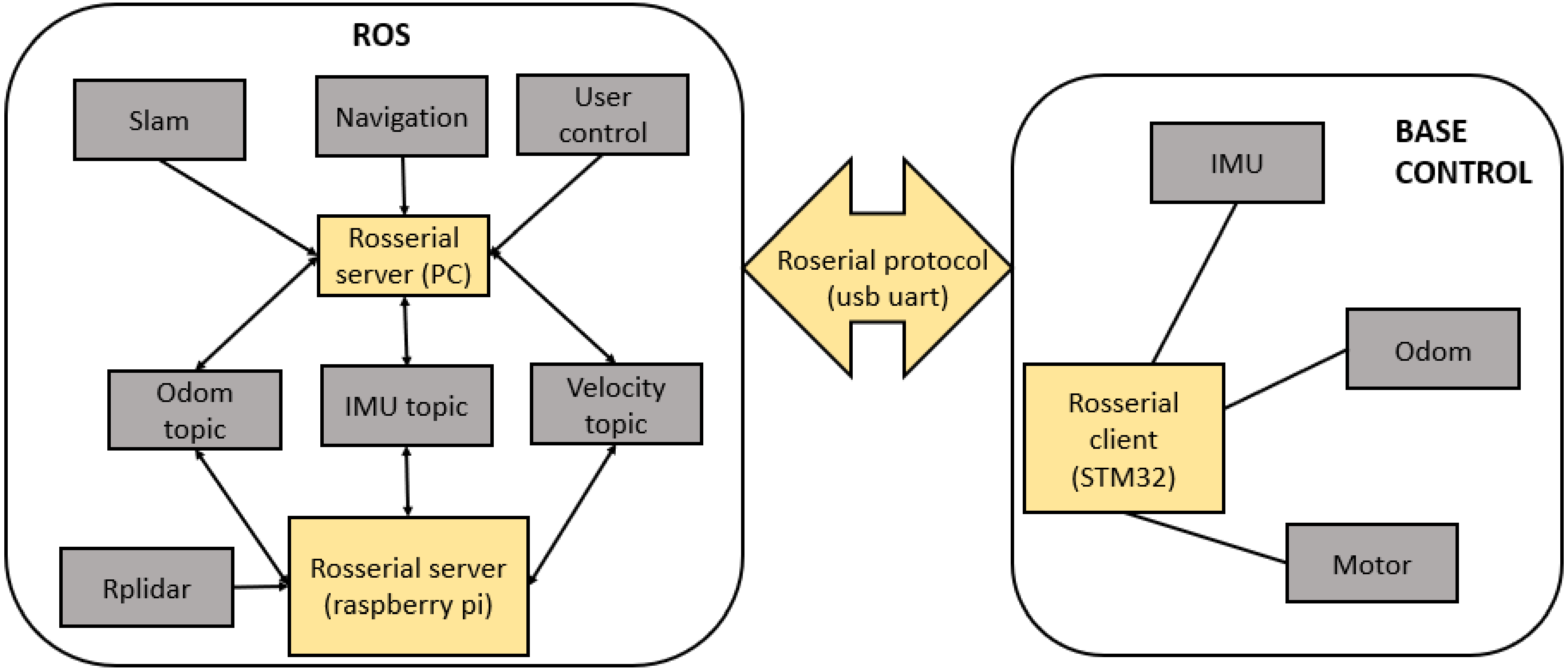


Figure 3.1. software structure of all system.

3.2 software design - flowchart

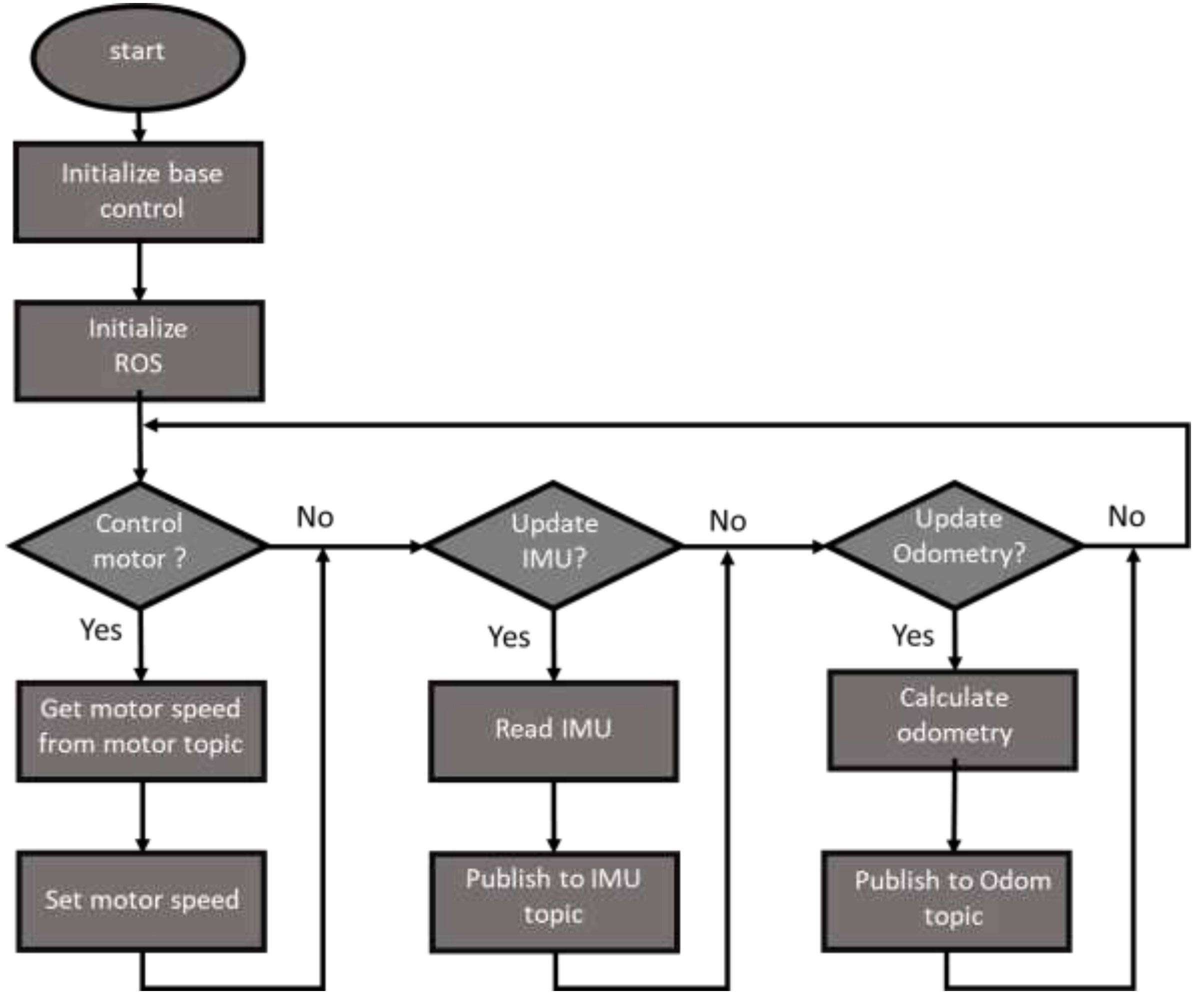


Figure 3.2. flowchart of base_control.

4. Result.

4.1 result – build map

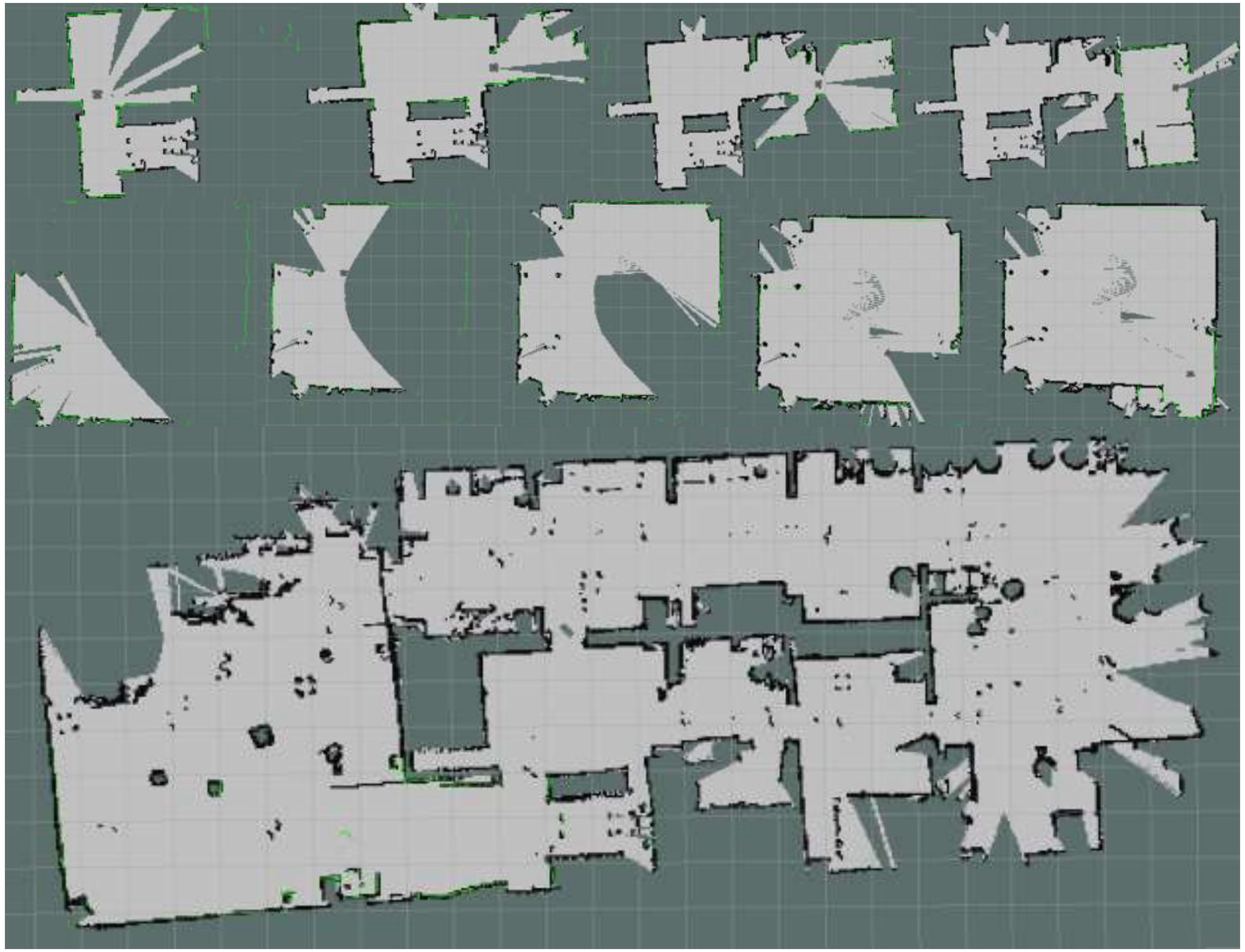


Figure 4.1. the process build map.

4.1 result – Evaluate the built map.

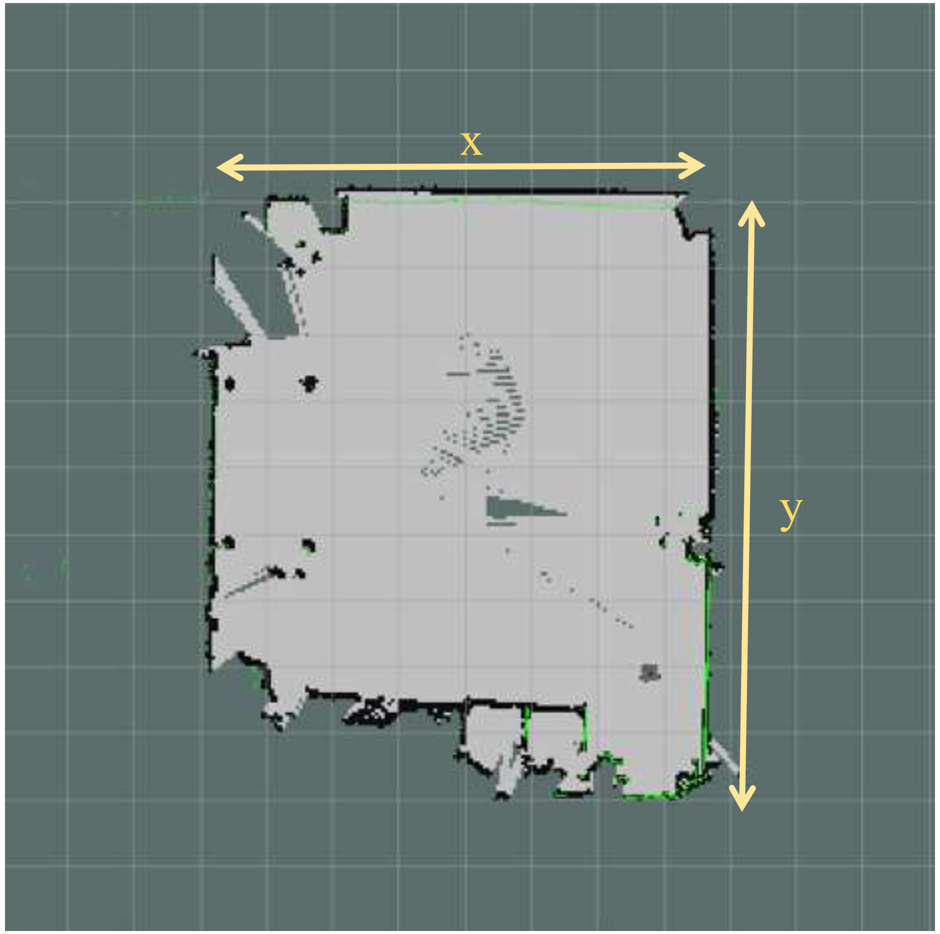


Figure 4.2. the process build map in a large yard.

	Reality (m)	SLAM (m)	error (m)
x	7.63	7.57	0.06
y	8.87	9.00	0.13

4.1 Result – Evaluate the built map.

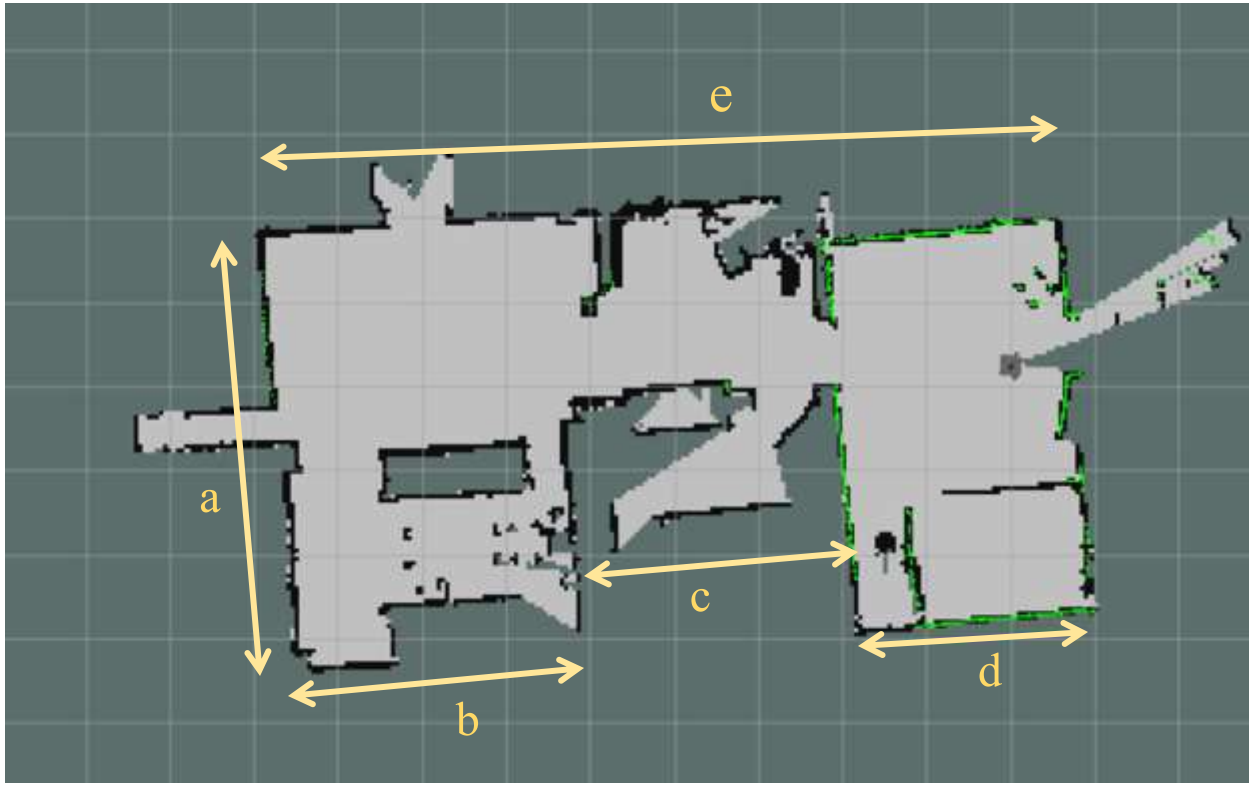


Figure 4.3. the process build map in home.

	Reality (m)	SLAM (m)	error (m)
a	5.24	5.30	0.06
b	3.42	3.50	0.08
c	3.20	3.25	0.05
d	2.96	2.87	0.09
e	9.72	9.50	0.12

- **Evaluate**: The map is built with dimensions close to reality with an error of no more than 0.13m

4.2 Result – Navigation.

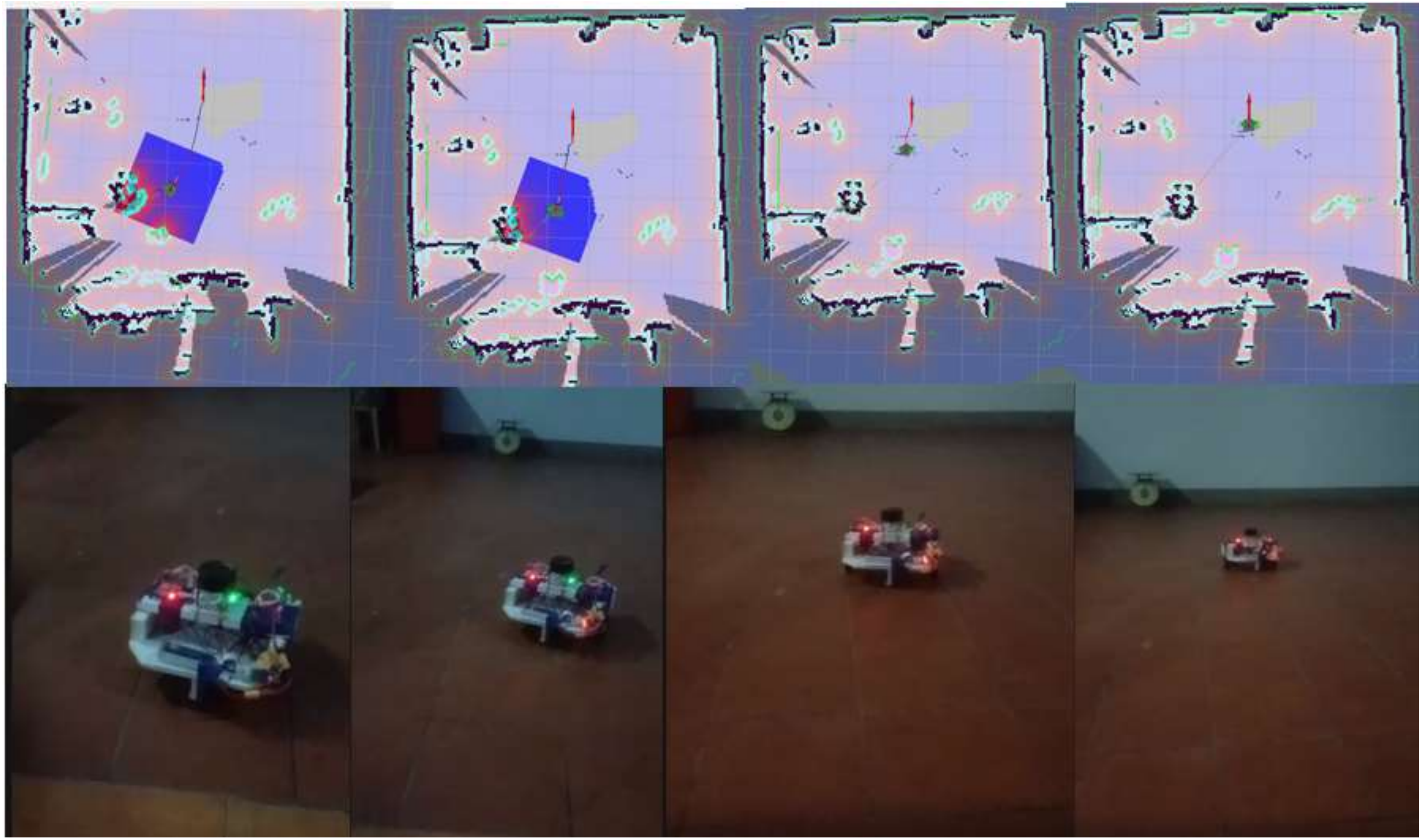


Figure 4.4. Robot go to the destination.

4.2 Result – Navigation.

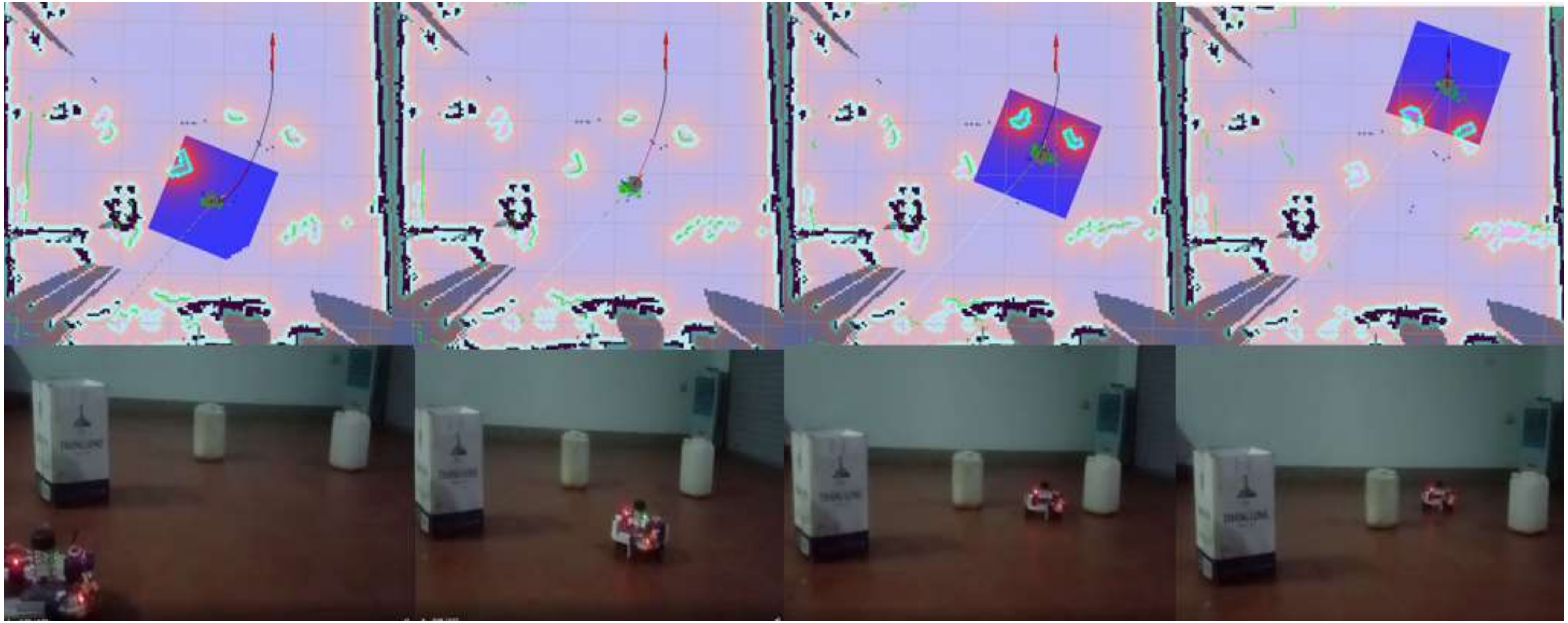


Figure 4.5. the process build map.

4.2 Result – Navigation.

Evaluate: The robot was able to reach its destination, avoiding fixed and newly appearing obstacles relatively accurately. However, when moving on slippery surfaces, when it reached the end of the path, the robot still had difficulty moving. find destination

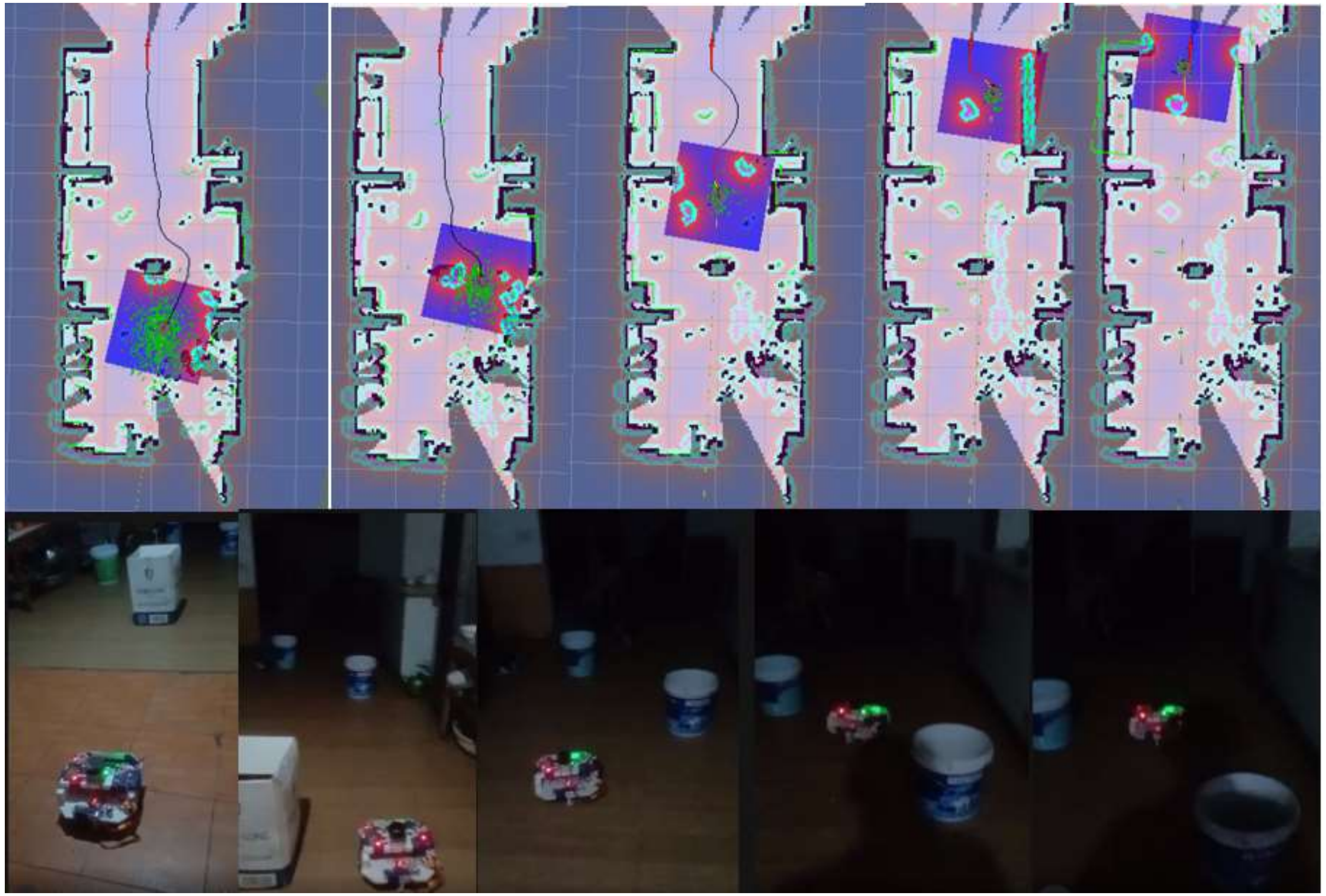
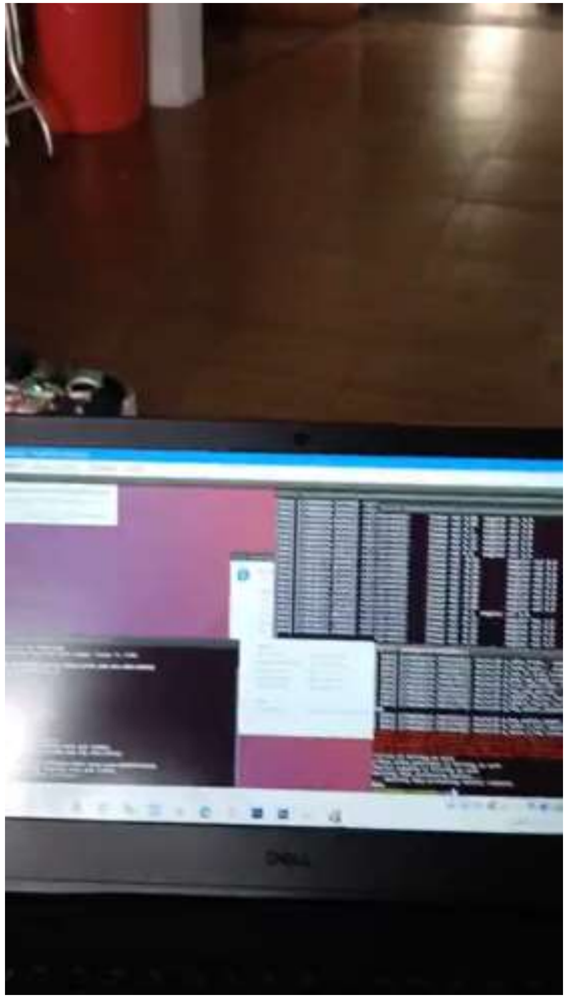


Figure 4.6. robot go to the destination with new obstacles appear.

4.3 Result – video demo



video1.video demo build map.

video1. video demo navigation.

5. Conclusion and development direction.

5.1 Conclusion

Achieved:

- The robot's information gathering and mapping process is quite accurate compared to reality.
- During movement, the robot's ability to avoid obstacles is quite accurate. When encountering static or moving obstacles, the robot can react appropriately to avoid those obstacles.

Restrict:

- Robots cannot avoid obstacles lower than LIDAR
- Does not work well on slippery surfaces.

5.2 Development direction.

To perfect an optimal robot that operates well and stably in many different environmental conditions, from there commercializing the robot model requires further development:

- Improve the hardware design to be sturdy, beautiful, and resistant so that the robot can transport goods and supplies to users.
- Integrate an additional camera to be able to receive electricity from objects and the surrounding environment.
- Integrate other functions to perform functions such as robot vacuum cleaners and delivery robots.

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !

Appendix- Hector slam

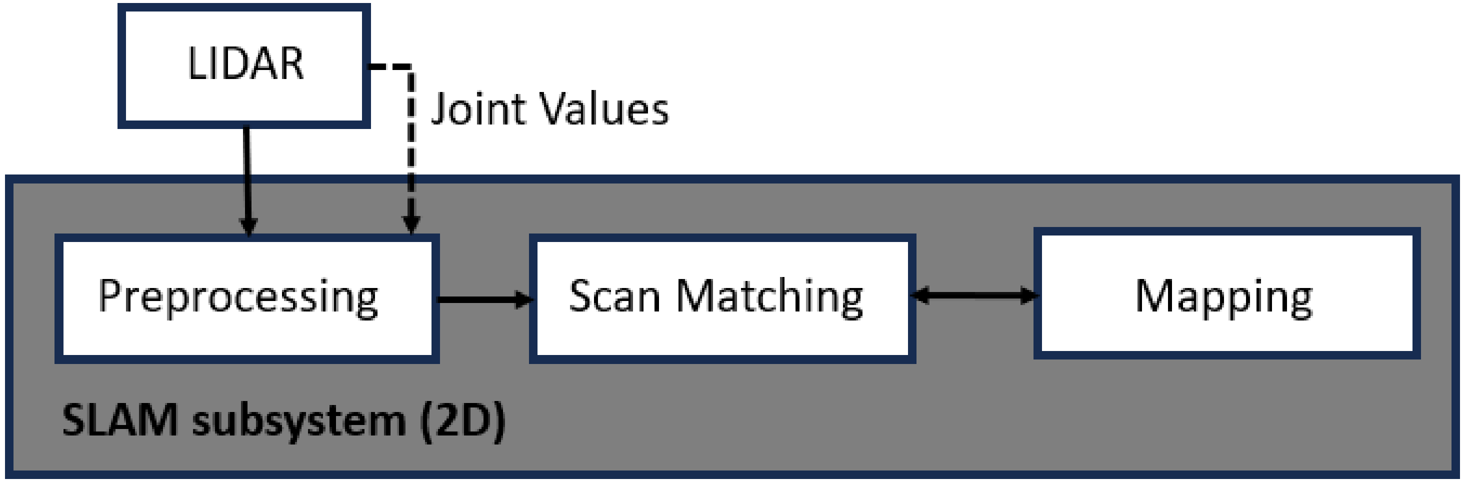


Figure 1. Map building process.

- **Step 1**: The LIDAR sensor will scan the surrounding environment.
- **Step 2**: The preprocessing process will use the Hector SLAM algorithm
- **Step 3**: The Scan Matching process will optimize the alignment of drawn points and newly scanned points
- **Step 4:** Let the robot move around and complete the map.

Appendix- Navigation

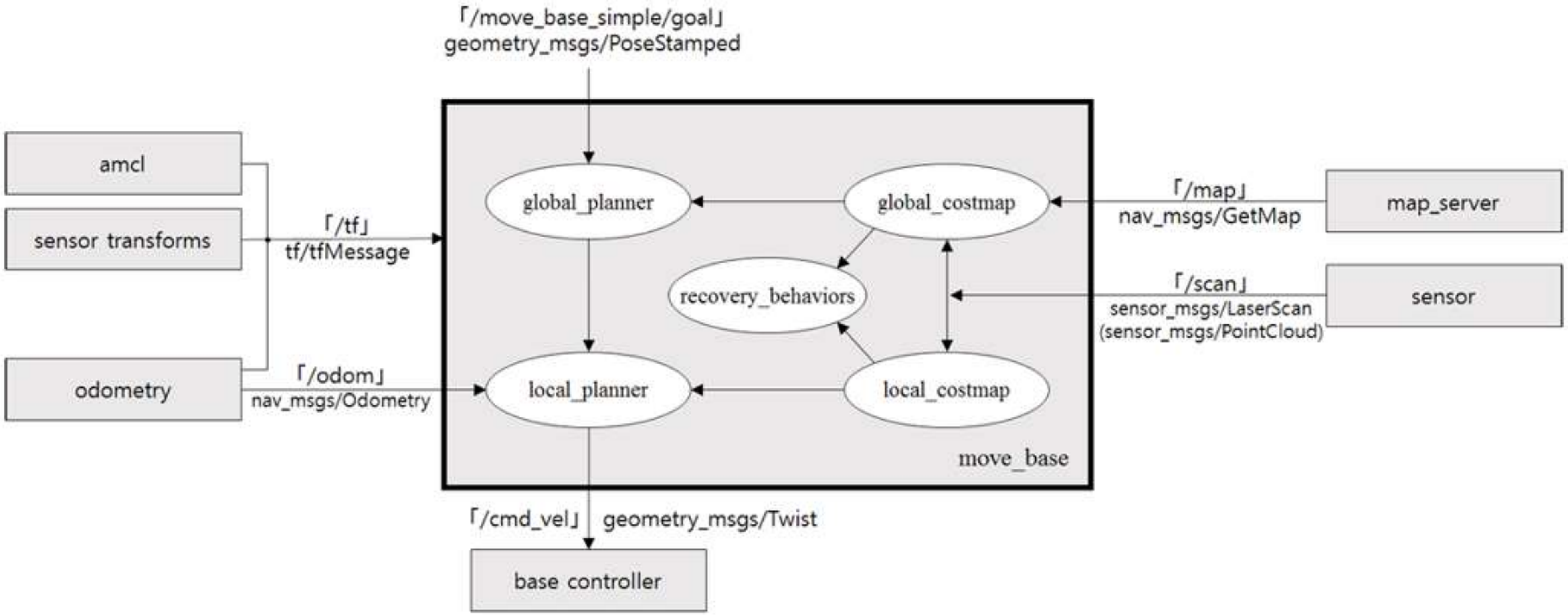


Figure 2. Relationship between essential nodes and topics on the navigation packages configuration.

Appendix- Dijkstra algorithm.

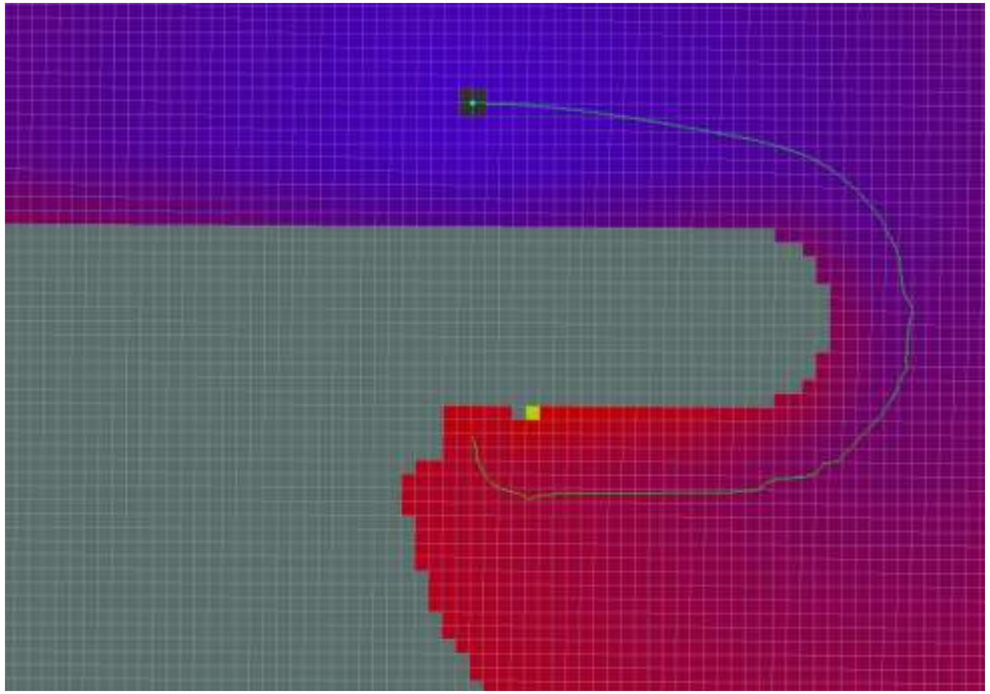


Figure 3. Path planning using Dijkstra's algorithm.

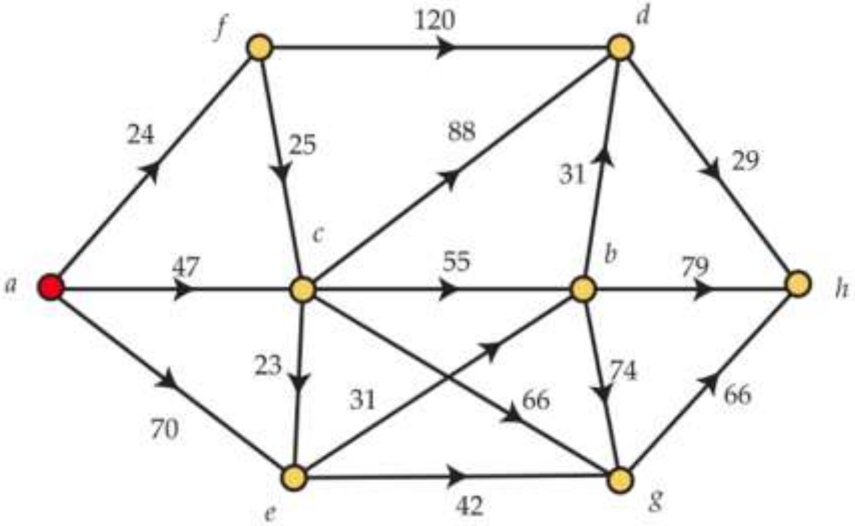


Figure 4. Dijkstra Algorithm.

Appendix- AMCL Algorithm and DWA Algorithm.

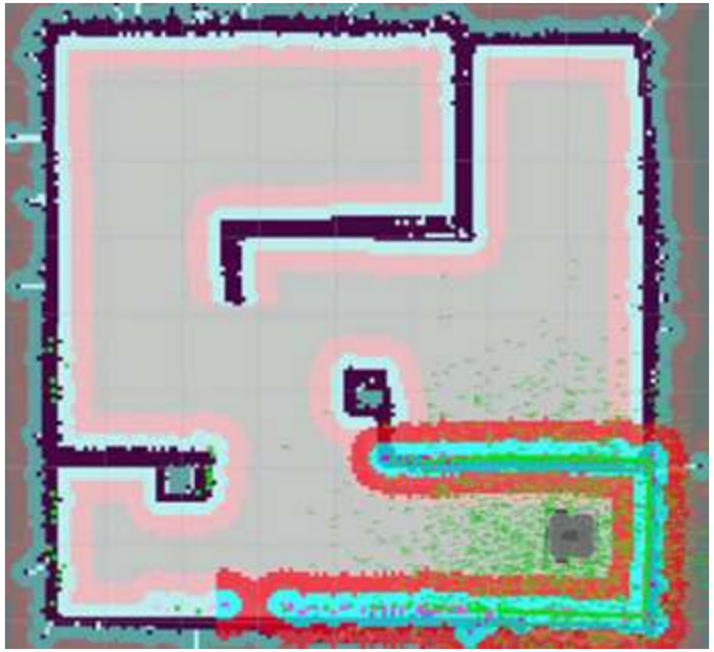


Figure 5. AMCL process for robot pose estimation

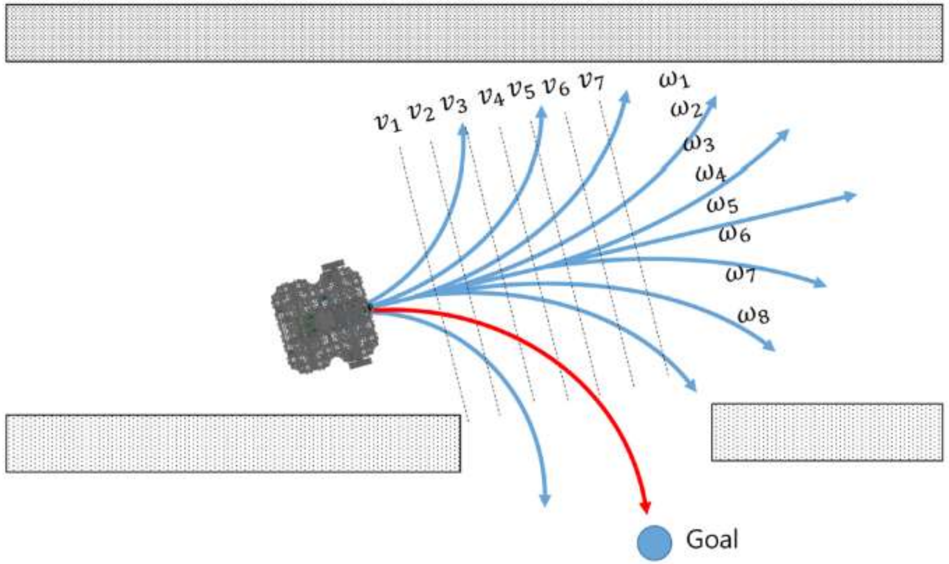


Figure 6. DWA algorithm.