**41068 ROBOTICS STUDIO**

PROJECT TITLE

Use an autonomous mobile robot (Turtle Bot) to  
 replicate supply chain activities for a smart factory.

SPRINT NUMBER: 1

DOCUMENT VERSION NUMBER: 1

DATE: 21/08/2024

**TEAM DETAILS**

|  |  |  |
| --- | --- | --- |
| **SID** | **Full Name** | **Email** |
| **14018295** | **Laurentius Setiadharma** | [**Laurentius.V.Setiadharma@student.uts.edu.au**](mailto:Laurentius.V.Setiadharma@student.uts.edu.au) |
| 24544963 | Shing Hin Yuen | shinghin.yuen@student.uts.edu.au |
| 14096301 | Luis Pratama | Luis.Pratama-1@student.uts.edu.au |
| 24620674 | Louis Ferdinand | louis.ferdinand-1@student.uts.edu.au |

Table of Contents

[PROJECT DESCRIPTION 2](#_Toc175077767)

[SPRINT 1 3](#_Toc175077768)

[SLO 1 3](#_Toc175077769)

[SLO 1.1: Communicating with the stakeholder 3](#_Toc175077770)

[SLO 1.2 Identifying and agreeing on the priorities, goals and system requirements 5](#_Toc175077771)

[Priority 5](#_Toc175077772)

[Goals 5](#_Toc175077773)

[System requirements 6](#_Toc175077774)

[SLO 2 7](#_Toc175077775)

[SLO 2.1 Problem Statement 7](#_Toc175077776)

[SLO 2.2 Functional requirements and design parameters of the project 7](#_Toc175077777)

[SLO 2.3 Technical statement 9](#_Toc175077778)

[SLO 2.4 Design Objectives and Evaluation Criteria 10](#_Toc175077779)

[SLO 2.6 Timeline for the Sprint 1 10](#_Toc175077780)

[SLO 3 12](#_Toc175077781)

[SLO 3.1 Configuring the system 12](#_Toc175077782)

[SLO 3.2 Set up a simple indoor office environment in Gazebo 12](#_Toc175077783)

[SLO 3.3 Design and develop a subscriber/publisher 12](#_Toc175077784)

[SLO 4 13](#_Toc175077785)

[SLO 4.1 Time management 13](#_Toc175077786)

[SLO 4.2 Communication skills 14](#_Toc175077787)

# **PROJECT DESCRIPTION**

*(200 words)*

The objective of the Smart Factory project is to use automation in the processing and transportation of goods within a factory's storage facility. This initiative aims to enhance efficiency, minimize the need for manual labor, and optimize overall operations. The project aims to resolve the inefficiencies of manual sorting, storing, and retrieving goods by introducing an autonomous mobile robot system. These inefficiencies now result in delays, errors, and higher operational expenses. The robot will effortlessly integrate itself into the current manufacturing facilities employing sophisticated mapping and localization algorithms to independently explore the surroundings and execute tasks with accuracy.

The aim of this initiative focuses around establishing a cooperative system in which robots and human workers can exist together, resulting in increased productivity while reducing the need for labor-intensive jobs. The project seeks to transform traditional manual procedures by offering a scalable and flexible solution that can be easily adjusted to different industrial layouts and storage setups. The project seeks to improve these essential operations by using automation, resulting in substantial reductions in processing times, improved precision in products handling, and the availability of real-time data for enhanced inventory management.

This report emphasizes the main outcomes of the project, which cover the development of the autonomous system, the incorporation of safety characteristics, and the overall influence on factory operations. The project makes an important contribution to the progress of automation in industrial environments, showcasing the possibilities for innovation in mechatronic engineering.

# **SPRINT 1**

## **SLO 1**

### SLO 1.1: Communicating with the stakeholder

1. **Meeting Appointment email: First Meeting**

**A screenshot of a email

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a video meeting

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. **Meeting Minute: First Meeting**

**Meeting Date:** 21/08/23 – 15.30  
**Stakeholder Name:** Vinu Vihan Maddumage

**Agenda Items:**

* Discuss the current manual goods handling process and its inefficiencies.
* Present the proposed autonomous robotic solution and its objectives.
* Clarify the project's deliverables and expected outcomes.
* Address any constraints or limitations identified during the design phase.
* Establish communication protocols and frequency for future updates.

**Discussion Notes:**

* The stakeholder emphasized the need to reduce manual labour and increase the efficiency of the storage facility.
* Emphasizing the significance of early environmental mapping and limitation definition in the project. An important focus was placed on the robot's ability to manage objects of different dimensions and masses.
* The stakeholder provided feedback on the need for a scalable solution that could adapt to different factory layouts.
* There was a discussion about the deliverables, including the importance of real-time data tracking and the expected format for progress reports (simulations, videos, and written reports).
* Safety features such as emergency stop mechanisms and human-awareness systems were discussed, with a focus on ensuring safe operation in a human-robot collaborative environment.

**Decided/Action Items:**

* The team will prioritize developing the robot's environment mapping and navigation capabilities.
* Initial simulations will focus on simple scenarios to identify any constraints and refine the design.
* Progress will be reported through a combination of simulations, videos, and written reports, with an emphasis on real-time data tracking, prioritizing video simulations over written reports.
* The team will initially focus on developing a minimum viable solution, and if there is enough time and resources, they will consider adding further features as stretch goals.
* Future meetings will be scheduled after key milestones (e.g., after the first sprint) to review progress and adjust plans as necessary.

### SLO 1.2 Identifying and agreeing on the priorities, goals and system requirements

### 1. Priority

* **Automation:** Fully automate the sorting, storing, and retrieval processes to minimize human involvement, reduce errors, and increase operational consistency.
* **Efficiency:** Design a cost-effective and time-efficient system that significantly outperforms manual labour, optimizing both productivity and operational costs.
* **Flexibility:** Ensure that the solution is adaptable to a variety of warehouse layouts, good types, and environments, and that it can provide reliable performance in a range of situations.
* **Safety:** Incorporate robust safety measures to promote safe collaboration between the autonomous system and workers in factories, thereby ensuring that the system operates without causing any risks and enabling a collaborative work environment.

### Goals

* **Develop an autonomous mobile robot that can sort and store items according to their status as raw materials, work-in-progress, processed, or finished goods within a smart factory simulation.**
* **Integrate customizable mapping and localization capabilities that enable the system to adjust to a variety of warehouse layouts, thereby guaranteeing operability and adaptability in a variety of environments.**
* **Integrate a robust sensor system within the simulation that enables accurate item classification, efficient navigation, and safe collaboration between the autonomous system and human workers.**
* Create algorithms that optimize goods handling, path planning, and real-time inventory management, thereby minimizing the need for human involvement and maximizing operational efficiency.

### System requirements

* **Navigation System:** Develop a robust mapping and path-planning system that allows the robot to accurately navigate the warehouse environment, adapting to different layouts and ensuring efficient movement between locations.
* **Obstacle Avoidance:** Implement obstacle detection and avoidance algorithms to ensure safe navigation around objects, workers, and other potential hazards in the warehouse.
* **Sorting System:** Design and integrate a sorting mechanism that enables the robot to differentiate between raw materials, work-in-progress, processed, and finished goods, ensuring correct storage and retrieval.
* **Communication Interface:** Establish a reliable communication interface for real-time monitoring and control, allowing for seamless interaction between the robot and human operators.
* **Language and Software Versions:** Use C++ as the programming language, with the system running on Ubuntu 22.04 and ROS2-Humble, and **Gazebo** as the simulation environment, providing a stable and flexible platform for robot development, deployment, and testing.
* **Hardware:**
* **Robotic Platform:** Utilize the TurtleBot for navigation and mobility within the warehouse environment.
* **Sensors:**
  + **LiDAR:** For precise mapping and obstacle detection.
  + **RGBD Camera:** For depth perception and identifying different types of goods.

## **SLO 2**

### SLO 2.1 Problem Statement

*"The current process of handling and transporting goods within the factory's storage facility is inefficient and heavily reliant on manual labour. The manual sorting, storing, and retrieval of goods are prone to errors, leading to delays and increased operational costs. Furthermore, the existing system is not optimized for scalability or robustness, making it unsuitable for large-scale industrial implementation. To overcome these limitations, an autonomous mobile robot will be introduced to streamline the process, enhance reliability, and optimize the overall efficiency of the factory's operations."*

### SLO 2.2 Functional requirements and design parameters of the project

**Overview:** Based on the client’s description, we have drafted a simulated warehouse environment with three key areas: an entry point (Area A) for the robot to receive manufactured goods, storage shelves (Area B) located on the side of the warehouse, and multiple output locations (Area C) leading to different departments.

The robot is tasked with three major functions:

1. Identifying the boxes, whether they are at the entry point or on the shelves.
2. Transporting boxes from the entry point to the shelves.
3. Retrieving boxes from the shelves and delivering them to their designated departments.

A black and white drawing of a house

Description automatically generated

Figure 1. Warehouse environment

**Functionality Requirement:**

1. **Environment Creation**

* **Design Parameter:** Simulated Environment - Develop a simulated warehouse environment with a defined layout, including an entry point, storage shelves, and multiple output locations.
* **Solution:** Virtual Warehouse - By utilizing Gazebo, a realistic warehouse environment can be simulated, accurately representing the operational zones and physical limitations within the factory.

1. **Sorting System**

* **Design Parameter:** Sorting Mechanism - Design a sorting system that can identify, categorize, and move different types of goods (raw materials, work-in-progress, processed, and finished goods).
* **Solution:** TurtleBot with Sensors - Utilize the TurtleBot equipped with appropriate sensors to recognize and sort goods into designated areas within the warehouse.

1. **Efficient Navigation**

* **Design Parameter:** Path Planning - Implement a navigation system that enables the robot to efficiently travel between different areas of the warehouse.
* **Solution:** SLAM and Obstacle Avoidance - Use Simultaneous Localization and Mapping (SLAM) combined with obstacle detection algorithms to allow the robot to navigate the warehouse safely and efficiently.

1. **Communication and Control**

* **Design Parameter:** Real-time Monitoring - Develop a system for real-time monitoring and control of the robot's operations within the warehouse.
* **Solution:** ROS2 and C++ - Utilize ROS2 and C++ to ease communication between the central control system and the robot, which allows the execution of commands and the exchange of data in real time.

1. **Safety and Reliability**

* **Design Parameter:** Safety Measures - Incorporate safety features to ensure that the robot operates without posing a risk to human workers.
* **Solution:** Emergency Stop and Redundancy - To guarantee the robot's safety and reliability, it is necessary to implement an emergency stop mechanism and redundancy in critical systems such as navigation and communication.

### SLO 2.3 Technical statement

**Reason for simulation robot and environment:**

Through simulating both the robot and the environment, the expenses required to construct the system can be minimized, and the potential hazards associated with testing the robot can be reduced. By using gazebo, we can effectively simulate both the environment and robot with a high degree of accuracy, eliminating the need for the actual physical components.

**Justify the choice of sensors:**

* **LiDAR (Light Detection and Ranging)**

LiDAR sensors are required for real-time mapping and obstacle detection, which allows the robot to generate precise 3D maps of its surroundings. This is critical for safe navigation and efficient path planning in a dynamic and potentially crowded environment such as a warehouse, where detecting obstacles requires accuracy and reliability.

* **RGB-D Camera (Red, Green, Blue-Depth)**

An RGB-D camera captures both color images and depth information, allowing the robot to recognize and categorize objects based on their appearance and spatial positioning. This capability is critical for tasks like sorting and storing goods, which require the robot to distinguish between different types of items and handle them precisely.

**Justify the choice of platform:**

Ubuntu 22.04 LTS, ROS2 (Humble), and Gazebo were chosen for their robust ability to support complex autonomous robotics tasks. Ubuntu is a stable operating system, whereas ROS2 provides advanced real-time communication and modularity, which are essential for handling complex navigation and sensor integration tasks. Gazebo enables high-fidelity simulations, allowing for thorough testing of the robot in an actual factory environment. This combination ensures a dependable and adaptable development platform, ideal for optimizing supply chain operations in a smart factory.

### SLO 2.4 Design Objectives and Evaluation Criteria

|  |  |
| --- | --- |
| **Design Objective** | **Evaluation Criteria** |
| The system has to demonstrate great operating ability in order to ensure precise movement. | Measure the deviation between the intended path and the actual movement path. The measured error of these distances must be less than 5%. |
| The system has to have the capability to simultaneously perform translation and rotation. | An arching path will be given to the system and the actual path will be compared with the intended arching trajectory. The system must maintain the correct arching motion with minimal deviation. |
| The system must precisely stop at the specified target or objective. | Measure the positional error at the stopping point and calculate the difference between the intended stop location and the actual stop location. The positional error must be less than 10%. |
| The system must possess the capability to detect obstacles and either avoid them or come to a stop. | Obstacles will be placed around the warehouse. The system must decelerate or stop within 1 meter of the obstacle or successfully navigate around it without collision. |
| The system has to have a way to identify and categorize the item. | The system’s item identification and sorting accuracy will be tested by comparing the system’s output against a predefined set of categorized items. The system must correctly identify and sort 100% of the items into their respective categories (3 types of manufactured items. |
| The system is required to execute the sorting operation within a predetermined time frame. | Measure the time taken for the system to complete sorting tasks from Entry Point (A) to Shelves (B). The system must perform the sorting operation within the predetermined time frame, with a maximum allowable duration of N seconds. |
| The system must be able to be stopped immediately when the emergency stop is pressed | The emergency stop button will be pressed when the system is operating, and the system must halt all operations. The system must cease all operations instantly with no noticeable delay. |
| The system should have the capability to effectively transmit information and provide updates to both the user and supervisor. | Assess the system’s communication by measuring the latency in transmitting data and updates. The system must ensure that all communications are transmitted with a delay of less than X milliseconds. |
| The system must be able to keep track of the good’s status | Test the system’s ability to track goods by comparing the recorded status of goods with actual goods' status at different stages. The system must maintain accurate records, with discrepancies less than Y%. |

### SLO 2.6 Timeline for the Sprint 1

|  |  |
| --- | --- |
| **Date** | **Description** |
| 5 August 2024 | Team is formed |
| 6 August 2024 | A Topic is selected |
| 12 August 2024 | An email is sent to the Client |
| 16 August 2024 | The meeting with client |
| 18 August 2024 | The meeting minute is completed |
| 19 August 2024 | Discussion of SLO 1 and 2 |
| 20 August 2024 | Completion of SLO 1 and 2 |

## **SLO 3**

### SLO 3.1 Configuring the system

For this subject, a Linux-based operating system (OS) is required, specifically Ubuntu 22.04 LTS. This OS is necessary to access ROS2, which facilitates easier development on robots. However, setting it up has posed several challenges. As a Windows OS user, I initially tried using Azure and a Virtual Machine to access Ubuntu, but both options proved unreliable due to latency issues. Ultimately, I opted to use a dual-boot system. While it presented some difficulties during the initial setup, it was worth the effort, as it offers excellent performance and has significantly increased my work efficiency.

### SLO 3.2 Set up a simple indoor office environment in Gazebo

When setting up the TurtleBot environment in Gazebo, I encountered several challenges. The first issue arose when the packages were not cloned properly. Despite multiple attempts, one of the packages downloaded very slowly, and the cloning process repeatedly failed. I even tried downloading the repository manually, but this approach proved ineffective due to missing elements. Eventually, I restarted all connections, and the package cloned successfully. Another challenge occurred when building the TurtleBot package; initially, it kept indicating that it could not find the necessary packages. To solve this problem, I tried various approaches, including deleting the packages and cloning them again. Ultimately, I discovered that cleaning up the ros2\_ws directory (deleting all folders except src) and then rebuilding it resolved the issue and the TurtleBot and its environment inside Gazebo was displayed correctly.

### SLO 3.3 Design and develop a subscriber/publisher

To publish every nth point of the LaserScan data, I first create a copy of the LaserScan message inside the callback and clear out its ranges data. Then, I select a value for n and iterate through the original LaserScan data, selecting every nth data point and appending it to the ranges vector of the modified scan. Afterward, I adjust the angle\_increment for the modified scan to reflect the reduced number of points. Finally, the modified LaserScan is published to the /modified\_scan topic. Refer to figure 2, for the flow chart of the algorithm.

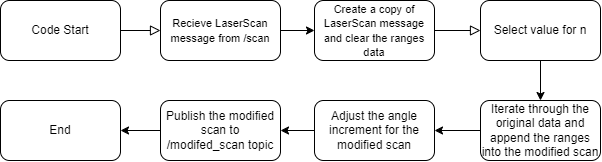


Figure 2. Flowchart for Sprint 1 code algorithm

## **SLO 4**

### SLO 4.1 Time management

A screenshot of a computer

Description automatically generated

### SLO 4.2 Communication skills

A black and white screen with white text

Description automatically generated

A screenshot of a video chat

Description automatically generated