ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

Automated guided vehicle for cargo delivery in factories

ZHENGJIE WANG
(zw65@illinois.edu)
XUHONG HE
(xuhongh2@illinois.edu)
YUYI AO
(yuyiao2@illinois.edu)
QIQIAN FU
(qiqianf2@illinois.edu)

TA: Haoran Cui, Yanbing Yang

April 14, 2025

Abstract

Cargo delivery in large factories faces challenges like low efficiency and safety risks with manual labor. To address these issues, we design an automated guided vehicle (AGV) to reduce labor costs and improve safety level ini this process. The AGV systems include following components: navigation system, obstacle avoidance system, instruction system and mechanical and electrical control.

Contents

1 Introduction

1.1 Problem

Cargo delivery in large factories has long been a problem due to the low efficiency, restricted working period, and high safety risks caused by manual labor. Human-operated systems, such as forklifts, often results into delays, errors, and accidents, as evaluated by injuries linked to forklifts in the U.S[1]. Human-operated vehicles or manual transport methods often result in delays, errors, and inconsistent performance, especially in complex factory surroundings. Workers' working hours are limited and therefore cannot maintain a 24/7 operation. Additionally, heavy machinery, narrow transporting channels, and dynamic obstacles increase the risk of accidents and injuries. To solve these problems, factory owners and researchers want to design a kind of automated guided vehicle that performs better than manual labor. These challenges call for the rising demand of automated guided vehicles (AGVs), which will eliminate human error, reduce labor costs by 20–30% [2], and operate continuously day and night.

1.2 Solution

An automated guided vehicle needs to be designed and assembled. The vehicle needs to deliver cargo within a large factory. The vehicle needs to be equipped with a control/navigation and obstacle avoidance system. This system ensures that the vehicle can move to the ordered destination by itself safely. Moreover, the vehicle needs to lift goods for at least 10 kilograms. This AGV frees workers from moving the goods from point to point, and the only job they need to do is to place the goods in the correct position when the AGV reaches its destination. To accomplish these functions, five parts will be built and assembled on a manual forklift to transform the cart into an AGV. The main system is a Raspberry Pi, serving as a core processor of any signals. Users can access the AGV from a distance, and the order assigned by the user will be transferred to the main system through the instruction subsystem, which contains database to provide the user with the available positions the AGV can go to. Once the AGV is in motion, another two subsystems will start functioning. The navigation subsystem employes laser radar and SLAM reconstruction positioning to navigate the route, and the obstacle avoidance subsystem uses ultrasonic sensor to detect any possible obstacles and give instant signal to avoid them. These subsystems communicate with the main system, and after processing, the Raspberry Pi will send signals to the mechanical and electrical subsystem to adjust the power supply of each motion part, controlling the motion of the cart.

1.3 Visual Aid

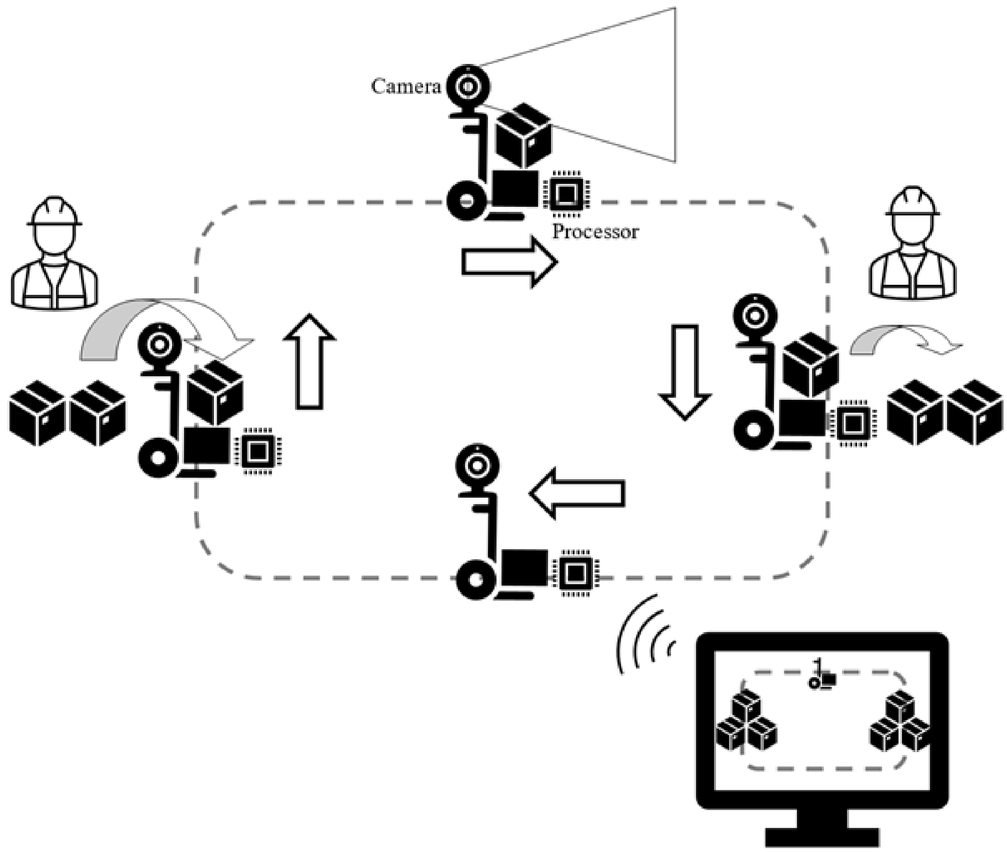

Figure 1: AGV System Overview

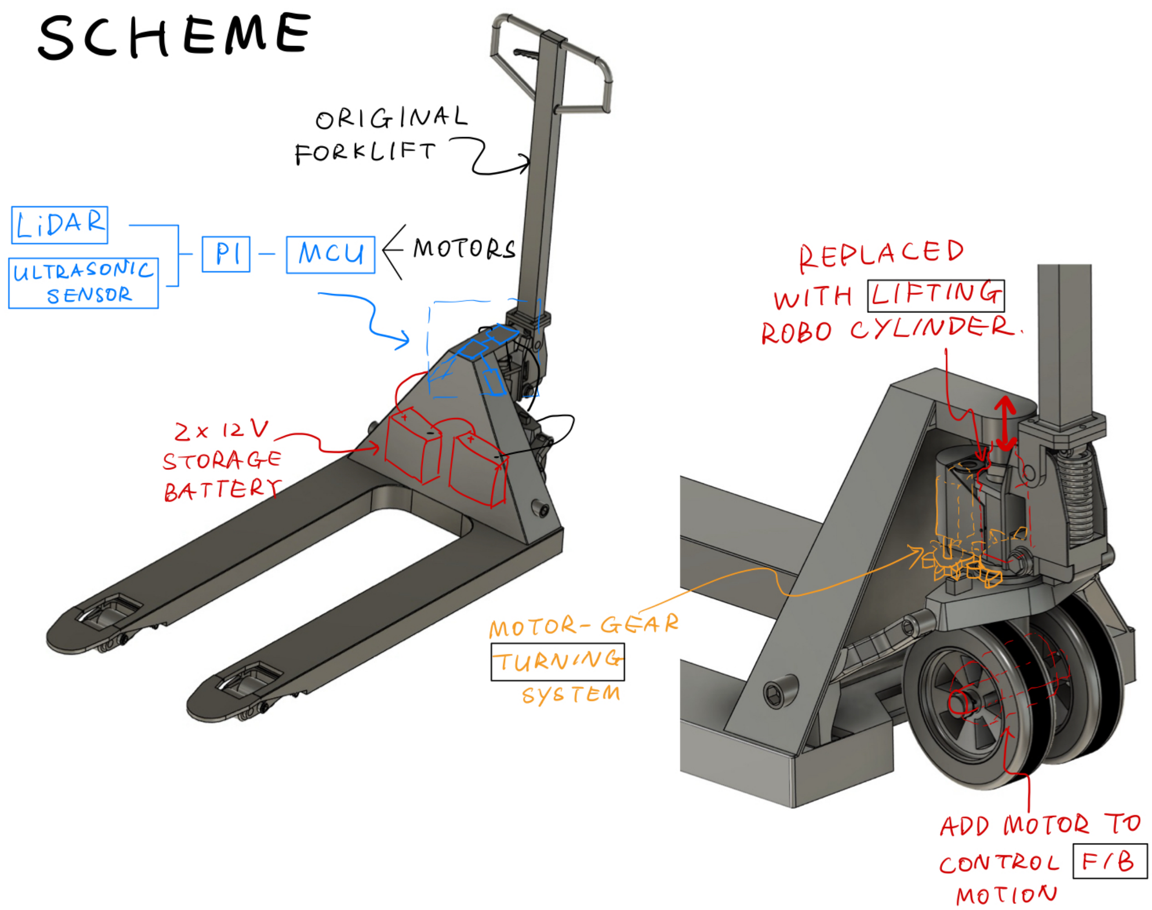

Figure 2: AGV Scheme

1.4 High-level requirements list

- The AGV should move along the direction line on the ground from the start point to the destination.
- When handling cargo, the AGV must properly align its fork before lifting or lowering the load.
- While moving, the AGV must continuously scan for obstacles and stop immediately if one is detected.

2 Design

2.1 Block Diagram

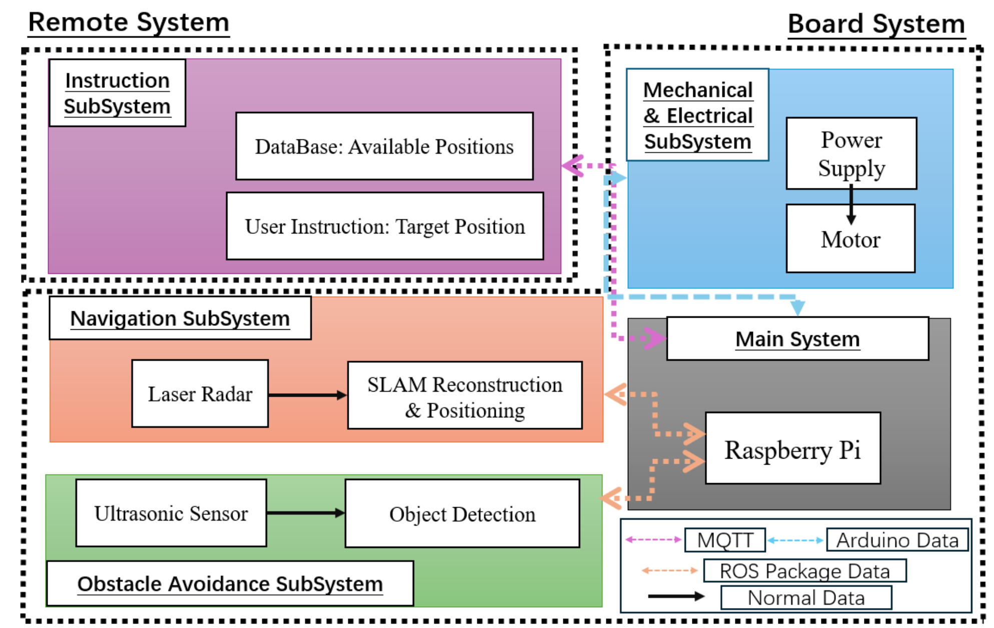

Figure 3: AGV System Overview

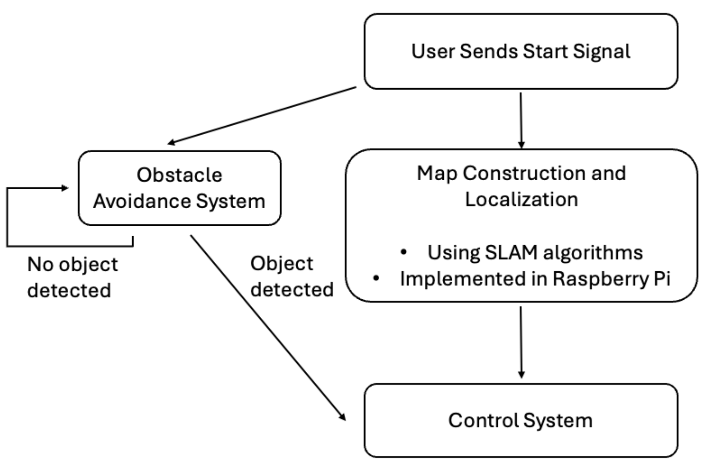

Figure 4: AGV System Block Diagram

2.2 Subsystem Overview

The AGV consists of four modular interconnected subsystems: Navigation SubSystem, Obstacle Avoidance SubSystem, Instruction SubSystem and Mechanical Electrical SubSystem. Each subsystem is essential to ensuring the AGV operates efficiently, safely, and autonomously within the factory environment. Together, these subsystems work in harmony to create an AGV capable of reliable and efficient cargo transport.

2.2.1 Navigation SubSystem

The SLAM (Simultaneous Localization and Mapping) [3] algorithm is responsible for constructing a real-time map of the factory environment while simultaneously tracking the vehicle's position within it. It integrates data from sensor (we use laser rader here) to estimate the vehicle's position and update the environmental map. This subsystem provides critical spatial scene for path planning. By maintaining accurate localization, it ensures the forklift operates autonomously in the factory scenarios.

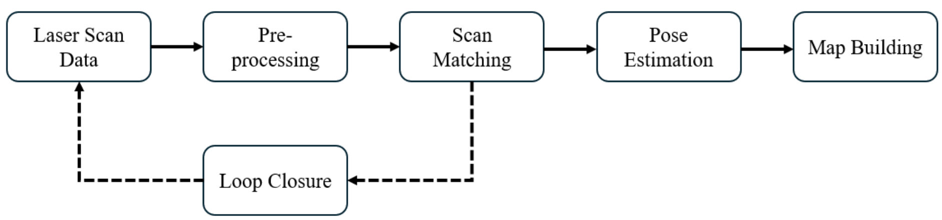

Figure 5: SLAM Scene Construction Block Diagram

2.2.2 Obstacle Avoidance System

The obstacle avoidance system ensures the AGV can move safely through the factory environment by detecting and avoiding both static and dynamic obstacles. Using ultrasonic sensors[4], the system continuously emits sound waves and measures their reflections to identify if there's an obstacle or not and determine their distance from the AGV. The system then sends signal to the AGV to stop until the obstacle disappears.

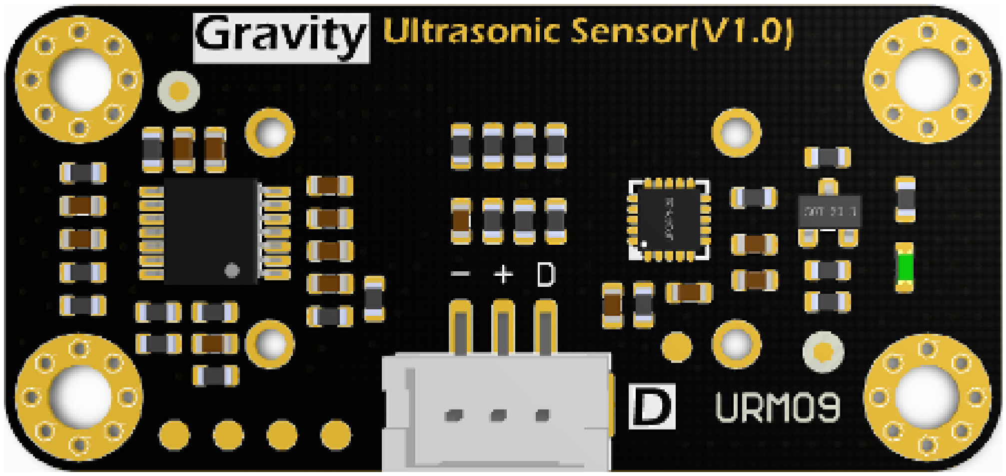

Figure 6: Ultrasonic sensor

2.2.3 Instruction SubSystem

The Instruction subsystem is to enable communication between the AGV and the control center in computer. Message Queuing Telemetry Transport (MQTT) [5] is used as the communication protocol. MQTT broker is the server that receives all messages from the clients and sends the messages to the appropriate clients. Whhen AGV is carrying a cargo, it will request the control center and will receive the information and move to the correct cargo shelf.

The communication is not only in charge of the information flow between users and AGV, but also in charge of the information between different modules in the system.

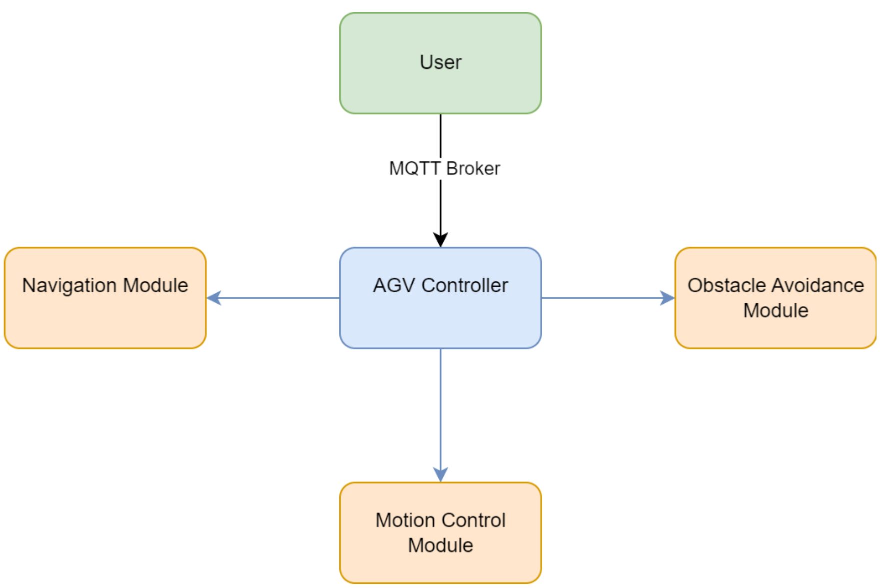

Figure 7: AGV Communication System

2.2.4 Mechanical Electrical SubSystem

The mechanical and electrical subsystem incorporates three electrical motors, controlling the motion of lifting, turning, and accelerating. The three motors are connected to a STM32 microcontroller, sharing the signal from the main system. Digital and analog signals are transferred and assigned to each of these motors, controlling the motion of the AGV to accomplish the order given by the user or the algorithm in the processor.

2.3 Subsystem Requirements and verification

2.3.1 Navigation SubSystem

The navigation subsystem must generate a real-time map with an accuracy of ±10 cm and maintain localization drift under 2% over a 10-meter trajectory. It must process data from laser radar with a refresh rate of at least 10 Hz to ensure efficient responsiveness and store the data locally. The subsystem will provide continuous position updates and map data for route planning. If the system fails to maintain accurate localization and map updates, the vehicle would be unable to navigate autonomously within the factory.

Requirement	Verification Method
The SLAM system must be able to build a map of a 5m × 5m indoor environment within 5 minutes, with mapping accuracy within ±10cm and localization accuracy within ±10cm at 5 test points.	A. Place the robot in a known 5m × 5m indoor area with measured reference points. B. Start the SLAM system and allow it to build the map. C. Export the map and compare key landmarks to ground truth. D. Move the robot to 5 reference points and compare estimated position with ground truth. E. Ensure mapping error is 10cm and localization error 10cm.
The robot must complete autonomous navigation to several random target points on the pre-built map with a 90% success rate, and must detect and avoid static obstacles with 90% success rate.	A. Set 5 navigation goals and let the robot follow planned paths; log success/failure and arrival accuracy for 20 total runs. B. Place static obstacles along the path in at least 10 trials. C. Verify that successful arrivals 90% and obstacle avoidance success 90% in all applicable trials.
Translate high-level velocity commands into motor control actions	The turning action is time-based (e.g., turning the wheels for 1.5 seconds achieves a certain angular change), so there must exist a known mapping between angular velocity and turning time. Do research and find the mapping out.

2.3.2 Obstacle Avoidance Subsystem

The Obstacle Avoidance Subsystem is responsible for detecting and avoiding both static and dynamic obstacles to ensure safe AGV operation. It continuously scans the surroundings using multiple ultrasonic sensors, which emit high-frequency sound waves and measure the time taken for echoes to return, allowing precise distance estimation. This data is processed to identify obstacles and determine their location relative to the AGV. Requirements 1: The system must detect obstacles within a minimum range of 3 meters Requirements 2: The Obstacle Avoidance Subsystem interfaces with the Mechanical Electrical SubSystem by transmitting stop signals when an obstacle is detected within a 50 cm range.

Requirement	Verification Method
1. Ultrasonic Sensor can operate properly. 2. Ultrasonic Sensor can communicate with control system properly.	1. Place known obstacles at distances of 10, 20, ..., 100 cm and compare sensor readings. 2. Connect the ultrasonic sensor to GPIO pins of the Raspberry Pi and verify signal and echo reception.

2.3.3 Instruction SubSystem

The Instruction subsystem ensure the communication process between the AGV and the control center. When a cargo need to be transfered to final desitnation, the control center will send the Instruction to AGV to ensure it move to start point and then move to the final destination and the correct cargo shelf. MQTT is used as the communication protocol between the AGV and the control center, which is a lightweight and low consumption protocol. Its Quality of Service ensures reliable message delivery, making it suitable for AGV system.

Requirement	Verification Method
The communication system should enable the information flow between vehicle and the central control. This module makes use of MQTT broker to transfer information.	1. Input command in the central control system on computer, the vehicle should receive the start signal and print the information out, which demonstrate its ability in communication. 2. The main control logic of the vehicle should receive information from SLAM algorithm and send control signal to control unit. This can be verify by checking if the values of those information is correctly printed out in the main control logic.

2.3.4 Mechanical Electrical SubSystem

The lifting motion's electrical motor, as well as the lifting plate's strength, should be able to provide a force that can at least lift goods of 10 kilograms. The turning part is designed to make a 90-degree turn within a 3 meters radius. The accelerating motor should provide a minimum operating speed of 1m /s when unloaded, and 0.5 m/s when fully loaded.

Requirement	Verification Method
The motion motors should be controled by the main control system, where the signals from the upper computer (Rasberry Pi) reach at. The respond of the control system should be fast once the signal is received.	A STC89C52RC MCU is mounted on the AGV. It receives signal from the upper computer through serial communication, and then a 8-digits binary number will be received and used to determine which switch should be turned on. Forward/backward motion is controlloed by a small steering engine, which uses PWM signal to set the rotation angles. Turning and lifting motions are controlled by a relay that serves as a switch to enable or disable the respective motion.
The automated guided vechicle shown br powered by a stable power source that supports both the motion of motors and power supply of MCUs.	Two stroage batteries are connected in series to provided a 24 V DC voltage for both motors and MCUs. The large capacity of the storage battery ensures the constant and smooth function for a long time.

2.4 Tolerance Analysis

A critical risk in using ultrasonic sensors for obstacle detection is measurement accuracy and response time, particularly in a dynamic factory environment where reflective surfaces, environmental noise, and sensor blind spots could affect performance. The system relies on three ultrasonic sensors positioned at the front, front-left, and front-right of the AGV to detect obstacles in multiple directions. This configuration allows the AGV to identify obstacles in its path and react accordingly by stopping when an object is detected within a 50 cm range. However, ultrasonic sensors may suffer from signal reflection and absorption issues, especially in environments with metallic surfaces or irregularly shaped objects. Reflections from highly smooth surfaces can cause false detections, while soft materials may absorb the sound waves, leading to undetected obstacles. Additionally, cross-interference between the three ultrasonic sensors could result in inaccurate readings, especially if they operate at similar frequencies.

To mitigate these risks:

- The sensors should operate at slightly different frequencies or incorporate sequential triggering to minimize interference.
- Signal filtering techniques, such as median filtering, should be implemented to discard outlier readings caused by reflections.
- The system should integrate redundant checks where an obstacle is only confirmed if detected by at least two consecutive readings within 50 ms to reduce false posi-

tives.

3 Cost and Schedule

3.1 Cost analysis

3.1.1 Labor Cost

For this project, we assume each team member is compensated at a rate of **$35/hour**, which aligns with typical internship or early career salaries for ECE graduates at the University of Illinois.

Each member works **15 hours/week** over **14 weeks**, and a multiplier of **2.5** is applied to account for meetings, documentation, testing, and project management.

$$\text{Cost}_{\text{individual}} = 35\ \$/\text{hr} \times 15\ \text{hrs/week} \times 14\ \text{weeks} \times 2.5 = \textbf{\$18,375}$$

There are four members in the team, so:

$$\text{Total Labor Cost} = 18,375 \times 4 = \textbf{\$73,500}$$

Table 1: Personnel Cost Table

Name	**$/Hour**	**Hours/Week**	**Weeks #**	**Multiplier**	**Cost ($)**
Xuhong	35	15	14	2.5	18375
Qiqian	35	15	14	2.5	18375
Yuyi	35	15	14	2.5	18375
Zhengjie	35	15	14	2.5	18375
Total	35	60	14	2.5	**73500**

3.1.2 Manufacturing Prototype Costs

The following parts are used in our project and are critical to implementing the AGV control system, SLAM-based localization communication system and sensor feedback systems:

Table 2: Equipment Cost Table

Name	Description	Quantity	Price	Cost
RPLIDAR-C1	The laser radar that can be used for SLAM algorithm	1	430¥	430¥
URM09-Trig	The ultrasonic sensors that can be used for obstacle detection	3	60.0¥	180¥
STC89C52 MCU development board	The MCU used in the main control system	2	80.0¥	160.0¥
DS3230 steering engine	The engine used to control the forward/backward motion	1	108.0¥	108.0¥

3.2 Schedule

We have a time-table showing when each step of design will be completed by week. It also include information about how the tasks will be shared between the team members.

Weekly Schedule: Xuhong He and Qiqian Fu

Date	Xuhong He	Qiqian Fu
2/24/2025	Study ROS and set up the executing environment	Study the concept of ROS
3/3/2025	Study SLAM algorithm and corresponding package	Study deep learning
3/10/2025	Set up the workspace of the whole project	Study concept of cloud point and relevant models
3/17/2025	Start programming on scene construction using SLAM	Testing how well the LiDAR perform using Point Pillar model
3/24/2025	Finish programming on scene construction part on the simulator using SLAM.	Thinking of other ways to implement obstacle avoidance
3/31/2025	Finish programming on vehicle localization part on the simulator using SLAM.	Study how ultrasonic sensor works
4/7/2025	Finish programming on route planning part on the simulator using SLAM.	Writing the code for using ultrasonic sensor to test distance

4/14/2025	Use Lidar to test the scene construction functionality in the real scenario.	Test ultrasonic sensor on Raspberry Pi
4/21/2025	Test the functionality of localization in the real scenario.	Study how to run ROS code
4/28/2025	Integrate the whole SLAM system into the main function.	Integrate the obstacle avoidance logit to the whole system
5/5/2025	Test AGV system in real scenario	Mapping the relationship between angular velocity and turning time
5/12/2025	Prepare final demo	Prepare final demo

Weekly Schedule: Yuyi Ao and Zhengjie Wang

Date	**Yuyi Ao**	**Zhengjie Wang**
2/24/2025	Study basic knowledge about ROS	Studying the mechanism of original forklift
3/3/2025	Set up workspace on Raspberry Pi	Set forklift 3D model in Autodesk Fusion
3/10/2025	Test MQTT system on computer	Prepare for motion connection remodel
3/17/2025	Connect Raspberry Pi to computer with ssh and start testing MQTT.	Set motion connection remodel to the forklift and test
3/24/2025	Finish programming on server code and vehicle side code on computer for communication system	Adjust motion connection remodel to the forklift and test
3/31/2025	Testing communication between the vehicle and computer	Install the MCU to the forklift and test basic functions
4/7/2025	Coding the main logic function framework of AGV.	Set the main control program to the MCU
4/14/2025	Coding for the function for information communication inside AGV: Gathering movement information in main logic function from SLAM algorithm.	Adjust the main control program of the MCU, with the adjustment of connection remodel part
4/21/2025	Coding for the function: Sending control signal in main logic function to control unit.	Combine the MCU with upper computer to test the control
4/28/2025	Integrate the whole communication system into the main function.	Integrate the whole communication system into the main function.
5/5/2025	Test AGV system in real scenario	Test AGV system in real scenario
5/12/2025	Prepare final demo	Prepare final demo

4 Ethics and Safety

The development and deployment of AGVs for industrial use may raise ethical and safety considerations.

4.1 Ethical Considerations

4.1.1 Public Welfare and Safety

Human safety has a higher importance than AGVs, especially in shared workspaces. This is most important during the whole development process, and should follow the 7.8 IEEE Code of Ethics, and ACM Code of Ethics and Professional Conduct [6][7]. A breach could lead to accidents or harm due to system failures or inadequate collision avoidance. To avoid this, it is possible to implement safety mechanisms (e.g., LiDAR, emergency stop systems) and evaluate them through limit tests like accelerated scenario-based evaluation.

4.1.2 Data Privacy

AGVs' motion is partly based on collecting operational data (e.g., worker movements), and may have risks in violating privacy if mishandled. A possible solution can be aligning with the client and employer best interests principle [8].Encrypt sensitive data and comply with GDPR or equivalent regulations.

4.2 Safety Standards and Regulatory Compliance

During the development, it is significant to note the restrictions published by the local government. Some relative regulations are listed here. ISO 3691-4 specifies AGV safety requirements, including obstacle detection and emergency braking[9]. The National Electrical Safety Code (NESC) ensures safe electrical infrastructure for AGVs, particularly in high-risk environments [10]. ISO 34502 outlines scenario-based safety evaluations for autonomous systems[11].

4.3 Potential Safety Risks

Potential safety risks in this AGV development are mostly about the risk of physical harm, system failures, and environmental hazards. For instance, in the deployment of an autonomous mobile device, there is a significant risk of collisions with humans, particularly vulnerable people such as children and the olds, leading to injuries. Similarly, in industrial environments, the use of AGVs may also cause collision risks due to sensor failures or algorithmic errors. Additionally, in high-temperature factory environments, such as those involving heat-treating furnaces, the lack of proper protective gear like heat-resistant gloves can lead to burns or other injuries.

4.4 Implementation Verification

All safety systems undergo validation through accelerated scenario testing that simulates edge cases including sensor failures, high-traffic environments, and extreme operating conditions. Compliance documentation includes ISO standard checklists, encryption audit logs, and training records that collectively demonstrate adherence to the highest safety standards throughout the product lifecycle. This comprehensive safety framework not only protects users and developers but also establishes a model for ethical AGV deployment in complex industrial settings.

References

[1] OSHA. "Forklift Accident Statistics." (2023), [Online]. Available: https://www.safetymanualosha.com/forklift-fatalities/.

[2] Deloitte. "The Future of Smart Manufacturing." (2021), [Online]. Available: https://www.deloitte.com/global/en/Industries/industrial-construction/perspectives/Digital-transformation-in-the-fom.html.

[3] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, and L. Carlone, "A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 1234–1250, 2023. DOI: 10.1109/TRO.2023.10075065. [Online]. Available: https://ieeexplore.ieee.org/document/10075065.

[4] Atanas Dimitrov Dimitar Minchev, "Ultrasonic Sensor Explorer," in *Proceedings of the IEEE Conference on Sensors*, IEEE, May 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7542987.

[5] OASIS. "MQTT Version 5.0." (2019), [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.

[6] ACM. "ACM Code of Ethics." (2018), [Online]. Available: https://www.acm.org/code-of-ethics.

[7] IEEE. "IEEE Code of Ethics." (2020), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

[8] ACM. "Software Engineering Code of Ethics." (1997), [Online]. Available: https://ethics.acm.org/code-of-ethics/software-engineering-code/.

[9] ISO. "ISO Standard 83545." (2023), [Online]. Available: https://www.iso.org/standard/83545.html.

[10] IEEE. "IEEE NESC Standards." (2025), [Online]. Available: https://standards.ieee.org/products-programs/nesc/.

[11] GlobalSpec. "Standard 34502." (2022), [Online]. Available: https://standards.globalspec.com/std/14572778/34502.