ASSUMPTION COLLEGE SAMUTPRAKARN

TATAYO BUS

Project report in computer

RESCUE ROBOT SYSTEM

Submitted By :

Veerakit Jenjadsap M.5/2 No.7
Patcharapon Sukcharoenthananon M.5/2 No.17
Pinyaphat Kongngern M.5/2 No.41
Wimonluck Yuthaisong M.5/2 No.45

Submitted To :

Master Ronie Molina
Master Atiphan Thoongsook

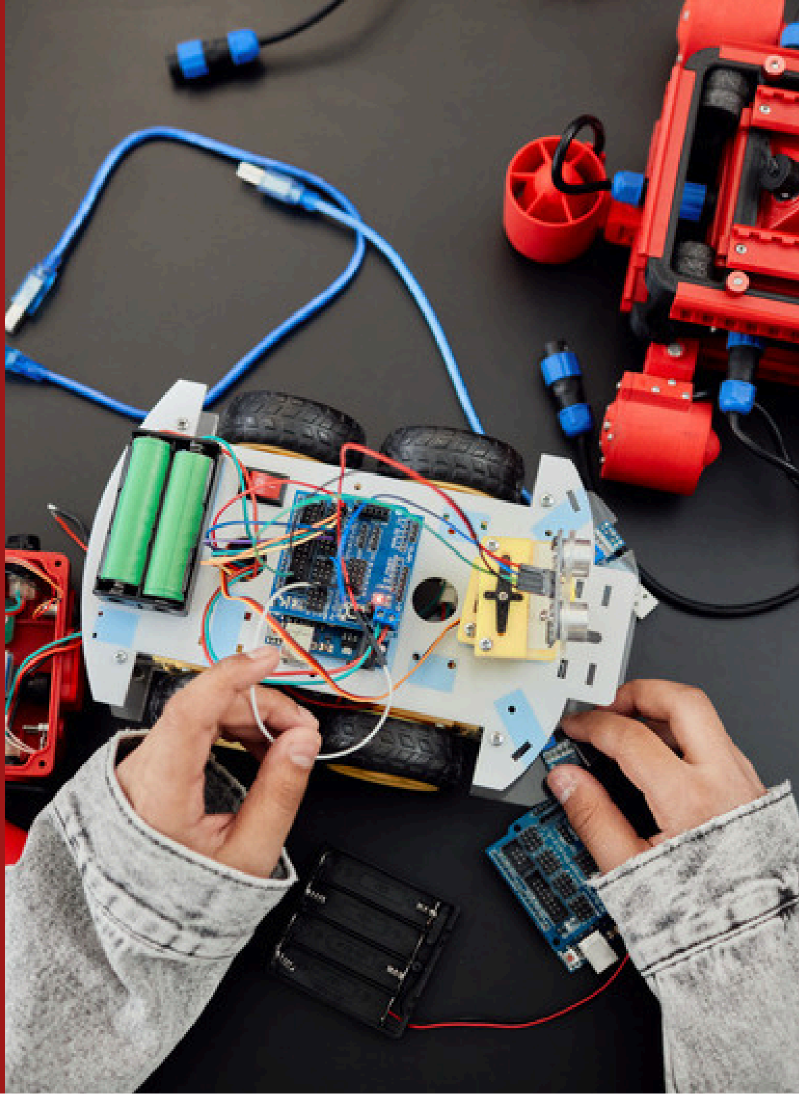
Table of Contents

PAGE | | CONTENT

INTRODUCTION

Tatayo Bus is the name for our group project of Rescue Robot System designed to detect a gas and alert passengers quickly. Equipped with gas sensors, system can identify potentials fire and trigger alarms. When a gas detected, the robot activates an alert via buzzer and open the doors automatically for passengers. It also have a switches to open the car roof for emergency exits. With optional mobility, it a significant step forward in improving fire safety and security of public transportation.

OBJECTIVES



- Detect fire using gas sensors.
- Trigger an alert sounds and visual alarm upon detection
- Open the doors automatically upon detection
- Use switches to open the car roof.
- Test the motor and switches to make sure there is no problem.
- Test and optimize for accuracy and reliability in fire detection.

KEY COMPONETS

Sensors

- Gas Sensor: detect toxic or explosive gasses and measure gas concentration. When a gas is detected, it will convert the concentration of various gases into electrical signals.

Actuators

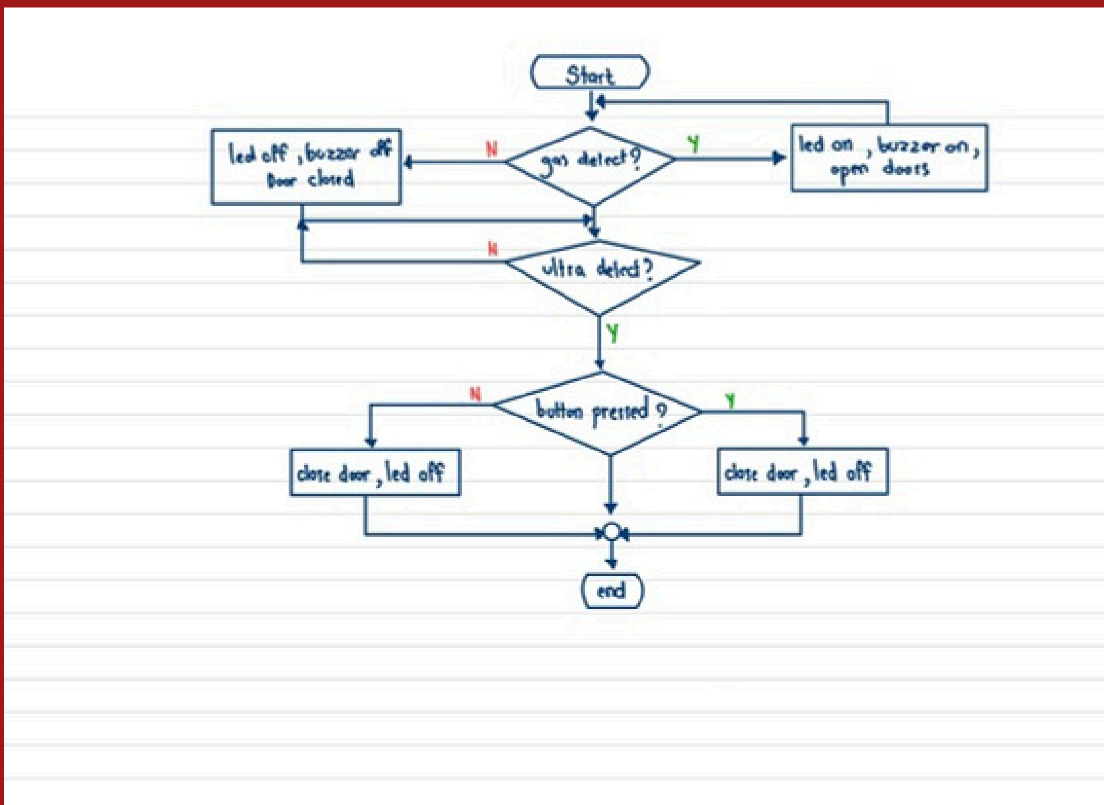
- Buzzer: Produces an audible alarm to alert occupants of potential fire danger when activated.
- LED: Visual alert system that flashes red when a fire is detected to increase visibility in low-light conditions.
- Motor: An automatic door system that opens when a fire is detected to increase more exits.
- Switch: Control an emergency exit, When it is pressed down it will activate to open the car roof.

Control Unit

- ESP32 microcontroller: Acts as the brain of the system, processing data from sensors,controlling the display, and managing alerts. It's programmed to activate the alarm based on sensors reading.

SYSTEM ARCHITECTURE

SYSTEM ARCHITECTURE DIAGRAM

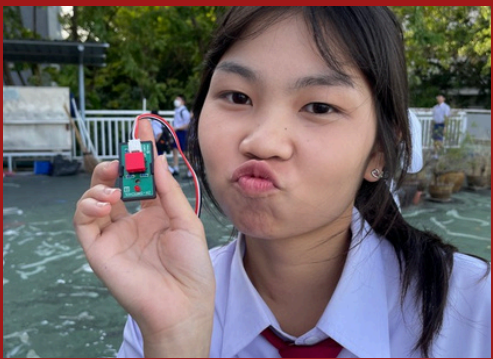


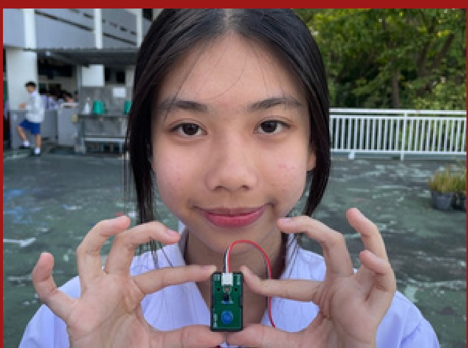
DESCRIPTION OF SYSTEM FLOW

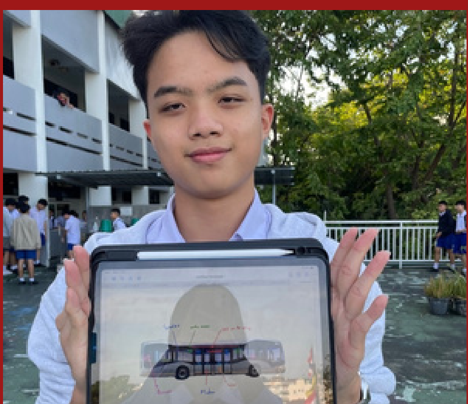
Describe these following stages of the diagram

- Input stage :   Detect Gas and have buzzer and LED
- Processing Stage : when have gas everything will activate.
- Output Stage : the door will automatically open and buzzer and LED will turn on.

METHODOLOGY







Collecting the components

Our group has gathered all the equipment needed for this project. The ESP32 microcontroller has been chosen as the main processing unit due to its ease of use and flexibility. It can seamlessly work with the sensors we have selected.

The sensors include gas detectors for detecting gas, as well as LEDs and a buzzer to send signals to a door. These signals will then be sent to the motor to open the bus doors as required, allowing passengers to evacuate safely in case of a fire.

Planning the Layout

The team carefully planned the layout to ensure functionality, efficiency, and clarity. The ESP32 microcontroller was placed centrally on the breadboard to simplify wiring and maintain tidy connections.

Sensors were strategically placed in various locations inside the bus to maximize their efficiency. The wiring was also neatly arranged to maintain a clean and organized appearance. Smoke detectors were installed on the ceiling at two points to detect smoke, while three sprinklers were mounted on the ceiling to extinguish fires immediately.

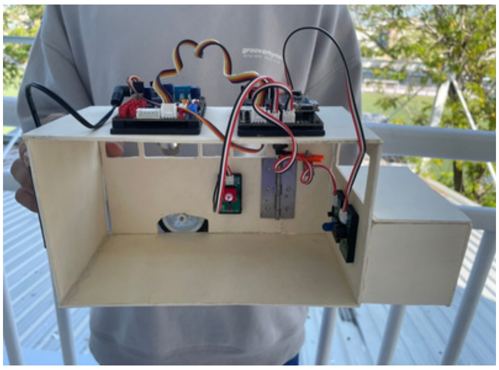
LED indicators were included to light up when smoke is detected, and a buzzer activates instantly upon detecting smoke. The system sends a signal to the sound sensor, which triggers the motor to open the bus doors immediately.

Under normal circumstances, a switch is available to manually open and close the doors as usual.

Placing and Finalizing Prototype

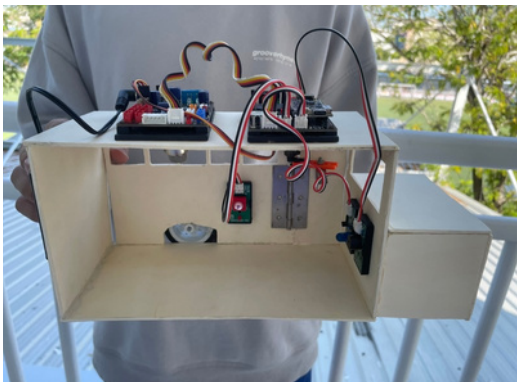
During weeks six and seven, we concentrated on putting the bus prototype together and affixing different parts to the paper. For writing code to test the model's functionality and testing the program, this procedure is essential. The ESP 32 microcontroller was first installed at the back of the bus model to act as the main hub for connecting all other devices throughout the assembly process.

To improve the bus's performance, sensors have been installed in the middle of the model, specifically smoke sensor to detect smoke inside the vehicle, and switches for opening the front and rear doors have already been installed.



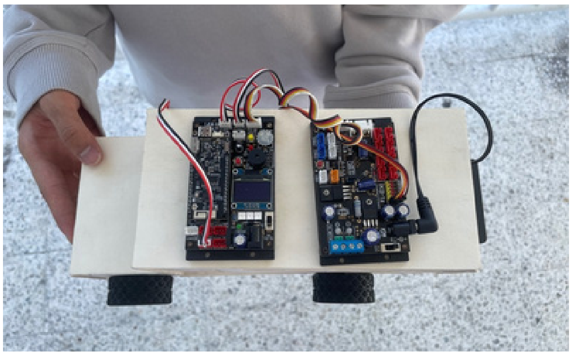
Inside the bus, on the left-hand side when facing forward, there are two switches installed near the door. These switches are used to open and close the door under normal circumstances. On the ceiling, we have installed a smoke detector to ensure it can detect smoke throughout the entire vehicle, which is why it is placed in this position. On the right side, there is an LED light that serves as a signal during emergencies. Slightly to the left, there is a circuit board installed on the ceiling as well, to ensure the entire system operates efficiently.

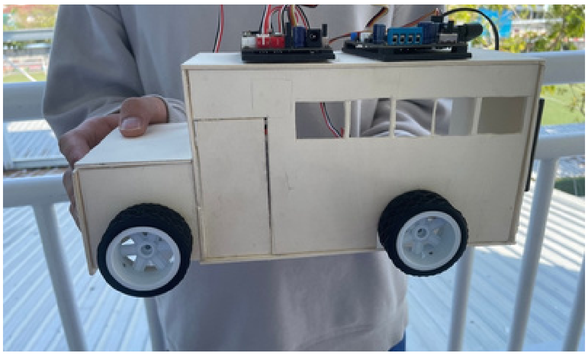
For this project, we used several durable materials that we believe will not easily break, such as wooden boards, hot glue, paper, and others, to ensure strength and stability. These materials were chosen to securely support the installation of circuit boards and various sensors, ensuring long-term usability. Additionally, we organized the sensors neatly to maintain a clean and orderly appearance.

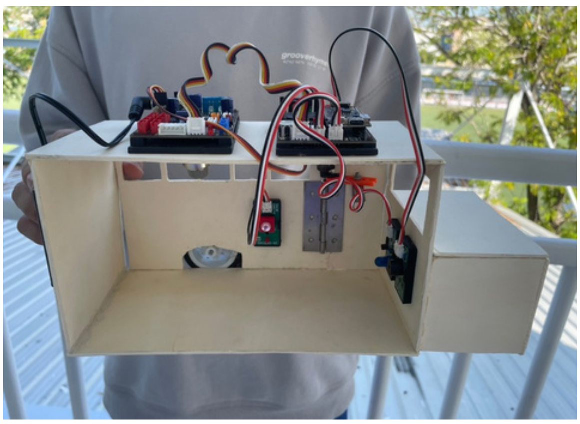


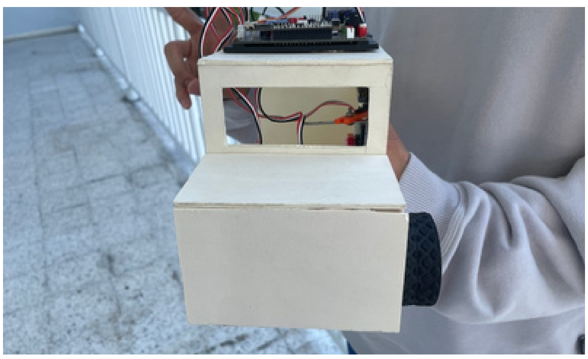
Once the assembly process was underway, adjustments were made to improve the layout of the ESP32 and various sensors. These components were repositioned slightly to ensure there were no obstructions and that everything was properly aligned for optimal connectivity and accurate detection.

By the end of Week 7, the prototype was fully assembled and completed. All components were correctly positioned and ready for the next steps of programming and testing. This milestone marked a significant step toward achieving a functional and visually appealing product that meets both technical and design specifications.









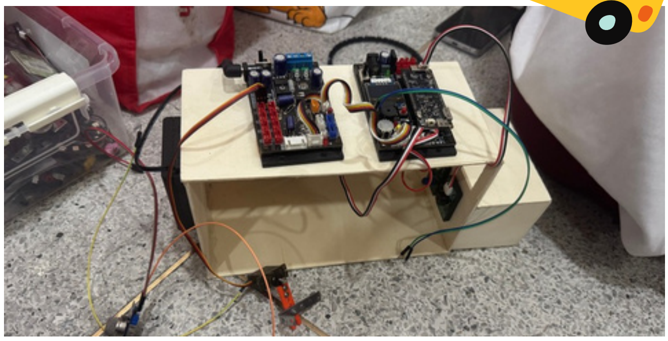
PROGRAMMING, INITIAL TESTING, AND SYSTEM INTEGRATION

PROGRAMMING AND INITIAL TESTING

During Week 10, our main task was to write and debug the initial code for the components of our project. We started by reviewing the system requirements and identifying the code needed to control the sensors and actuators.

The ESP32
microcontroller served as the main brain of our system, and we programmed it to manage data from the ultrasonic sensor, soil moisture sensor, temperature sensor, and servo motor. Each component required specific code to function properly. For example, the ultrasonic sensor was programmed to measure distance, while the servo motor needed precise control for movement. We used the Arduino IDE to upload the code and debug it step by step. Some errors occurred during testing, such as incorrect sensor readings and delays in the servo motor's response. We fixed these issues by adjusting the code, ensuring the right libraries were included, and checking our wiring
connections.

By the end of the week, we had successfully uploaded working code for each component and tested them individually to confirm they were functioning as expected. The next step was to combine these components into a complete system during Week 11.



In Week 11, we focused on integrating all the components we programmed in Week 10 into a fully functional system.

The ESP32 served as the central hub of the system, connecting the sensors and actuators to ensure they could work together. We connected the ultrasonic sensor (which was about to be added), gas sensor, LED, buzzer, and servo motor to the microcontroller through the breadboard, ensuring stable connections.

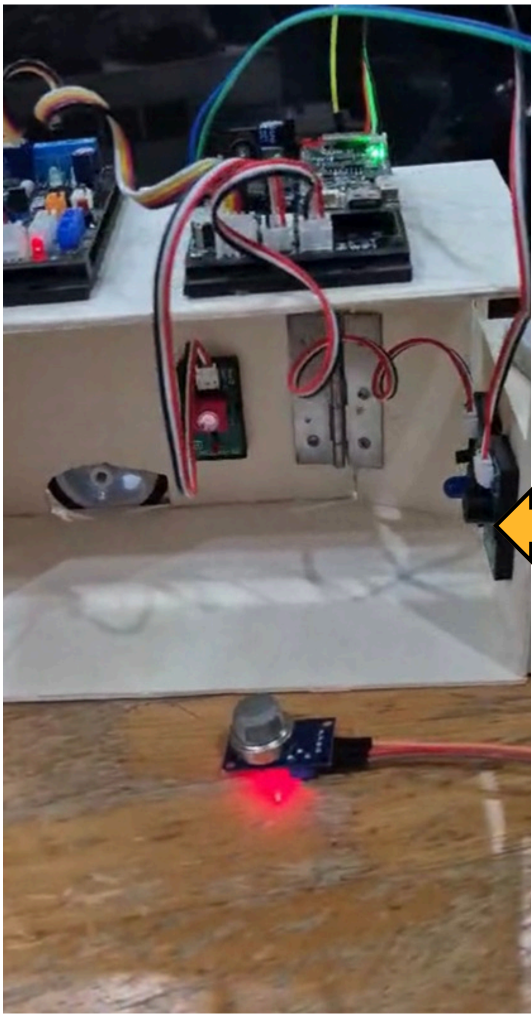
One of the main challenges we faced during integration was managing the power supply. Some components, such as the servo motor, gas detection system, LED, buzzer, and push button, caused slight instability in the systemâ€™s operation while running. We resolved this issue by using an external power supply to balance the load and ensure that each component received adequate power. Additionally, we reorganized the wiring to keep everything neat and reduce the risk of tangled wires.

After addressing these challenges, we successfully tested the integrated system. The results met our expectations, with the ESP32 efficiently processing data from the sensors and controlling the actuators. By the end of Week 11, our system was stable and ready for advanced testing in the following week.

Week 10

Week 11

PROGRAMMING, INITIAL TESTING, AND SYSTEM INTEGRATION



PROTOTYPE REFINEMENT AND OPTIMIZATION

In Week 13, we focused on improving our prototype to enhance its structure, stability, and usability.

We reorganized the wiring on the breadboard to create a cleaner layout and reduce the risk of tangled wires. Additionally, we adjusted the positioning of the sensors to improve accuracy and efficiency. For example, we repositioned the gas sensor and considered adding an ultrasonic sensor to the system.

We also reinforced the prototype structure to ensure stability during operation. Every component was securely installed to provide clear outputs, and in the future, we may further organize the wiring to minimize the risk of disconnections. These improvements made the prototype more user-friendly and reliable.

After these refinements, we tested the updated system again. The results showed greater stability, faster response times, and more accurate readings across all components.

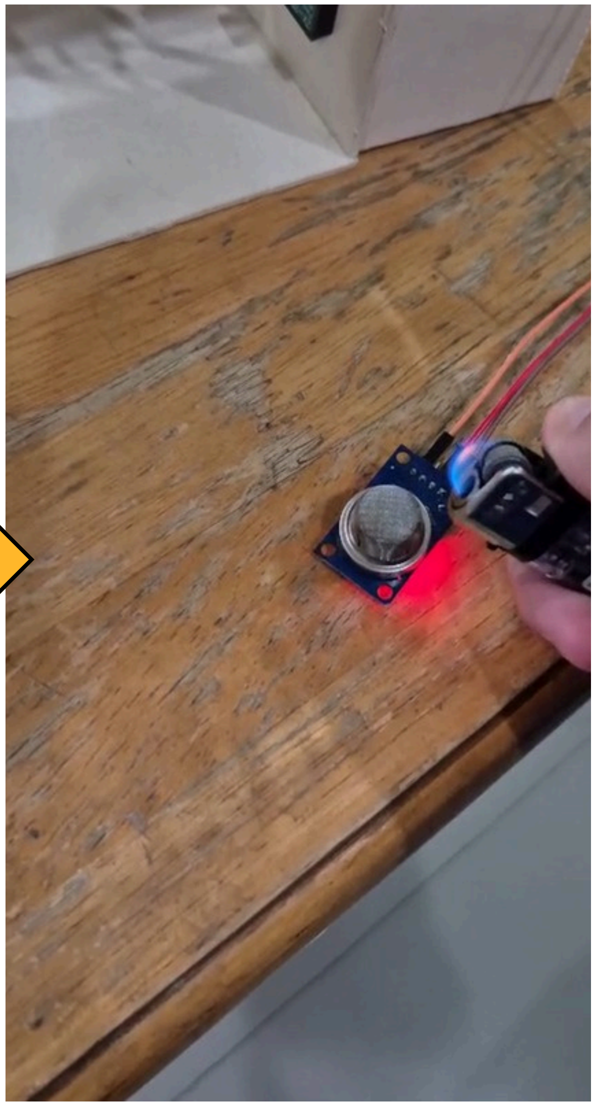
ADVANCED TESTING AND TROUBLESHOOTING

In Week 12, we conducted advanced testing to ensure that our integrated system functioned correctly under different conditions.

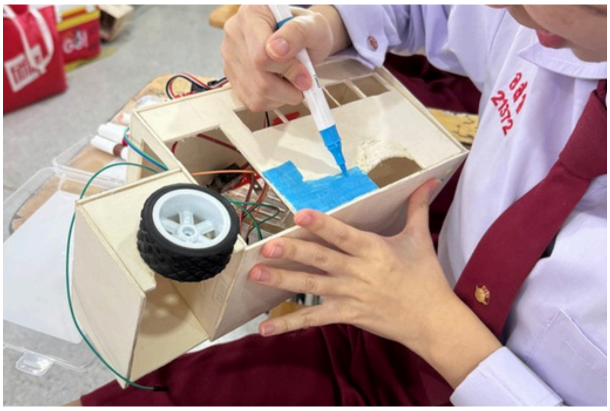
We tested each sensor and actuator to verify their consistency, even in changing environments. The gas sensor was tested by detecting gas from a lighter to ensure accurate detection at various distances.

One of the challenges we encountered was inconsistent sensor data, especially when the power supply fluctuated. We resolved this issue by securing the wiring connections and using a more stable external power source. Additionally, we noticed delays in the servo motorâ€™s response when handling larger loads. To fix this, we optimized the code to enhance the motorâ€™s efficiency and reduce response time.

By the end of the week, our system became more stable and reliable, ready for further improvements in Week 13.



CONCLUSION



Progress made in Phase 1 and Phase 2 has significantly advanced our project, from initial planning and prototyping to system integration and testing.

In Phase 1, we focused on identifying key components, defining objectives, and building a preliminary prototype. This phase laid the foundation for further development by ensuring that all necessary components were selected and prepared.

In Phase 2, we focused on programming, testing, and system integration to produce a stable and functional prototype. Through advanced testing and troubleshooting, we addressed key issues such as power supply inconsistency and sensor accuracy issues. Improvements were made to improve the structure, stability, and usability of the prototype.a

Overall, the project has demonstrated its core functionality and successfully achieved the initial objectives. The progress made so far provides a solid foundation for further improvements to improve performance, usability, and sensing capabilities.

Further Development :

While the project has met its primary objectives, there are several areas where further improvements can be made to enhance the system's functionality and efficiency:

Power Optimization

In the future, we plan to incorporate solar panels to replace the use of batteries or electricity, which will increase energy efficiency and reduce pollution.

User Interface Development

We will develop an online application that can send alerts through the app in case of danger.

Additional Sensors

We will add sensors for flame detection and moisture detection to improve accuracy.

Prototype Design

We will improve and create functional, movable wheels and may use steel for the new construction.

Real-World Testing

In the future, we will conduct trials in real-world locations, such as on public roads or by borrowing spaces from driving practice fields, to ensure the system performs at its best before actual deployment.

Error Handling and Alerts

In the future, we will continuously make improvements and fix errors by utilizing AI for calculations and notifications. We will also develop an app to consistently send alerts to passengers.