

COLLEGE OF COMPUTING AND INFORMATICS TECHNOLOGY

Semi-Autonomous Fire-Fighting Robot System

By

[CS26-34]

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

A Project Report Submitted to the School of Computing and Informatics Technology
in Partial Fulfillment of the Requirements for the Award of the Degree of
Bachelor of Science in Computer Science
of Makerere University

Supervisor

[Dr.Rose Nakibuule]
School of Computing and Informatics Technology, Makerere University
rnakibuule@cit.ac.ug
, +256-701-116-829

[June, 2026]

GROUP MEMBERSHIP

SN	Reg. Number	Student Number	Name	Email Address	Signature
1	23/U/13097/EVE	2300713097	NAGGAYI DAPHNE PEARL	daphnepearl101@gmail.com	Daphne N
2	23/U/15155/PS	2300715155	NANNOZI RACHEAL	rachealnannozi77@gmail.com	Racheal N
3	23/U/25628/PS	2300725628	GALIWANGO ANANDA	anandagaliwango@gmail.com	Ananda G
4	23/U/14140/EVE	2300714140	NALUNKUMA KETRA	ketrana4@gmail.com	Ketra N

Preliminary pages (Use Roman Numeral Numbering system)

i. Declaration page by group members

ii. Supervisor(s) approval page

iii. Table of content page(s)

iv. List of figures if applicable

v. List of tables if applicable

vi. List of abbreviations if applicable

Main Content pages (Use Hindu Arabic numbering system)

Declaration

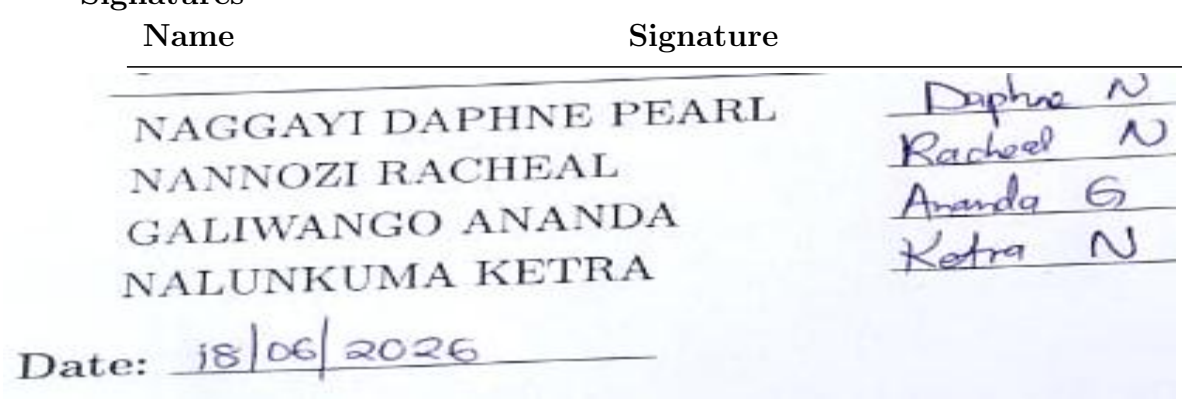
We, the undersigned, declare that the work presented in this Project Report titled *"Semi-Autonomous Firefighting Robot "* is our original work. It has not been submitted, in full or in part, to any other institution or for any other academic award.

We affirm that all sources of information used in this work have been properly acknowledged through citation and referencing. This Project Report was jointly developed by the members of our group, and each member contributed significantly to the research, design, implementation, writing, and review of the work.

Group Members

No.	Student Number	Registration Number	Name	Email
1	23/U/13097/EVE	2300713097	NAGGAYI DAPHNE PEARL	daphnepearl101@gmail.com
2	23/U/15155/PS	2300715155	NANNOZI RACHEAL	rachealnannozi77@gmail.com
3	23/U/25628/PS	2300725628	GALIWANGO ANANDA	anandagaliwango@gmail.com
4	23/U/14140/EVE	2300714140	NALUNKUMA KETRA	ketrana4@gmail.com

Signatures



Supervisor Approval

This project was carried out under the supervision of:

Dr. Rose Nakibuule
School of Computing and Informatics Technology (CIT)
Makerere University
Email: rnakibuule@cit.ac.ug
Tel: +256 701 116 829



Supervisor's Signature Date

This is to certify that this report was prepared under my supervision and is ready for submission.

Contents

1 Introduction **1**

1.1 Background 1

1.2 Problem Statement 2

1.3 Objective 2

1.3.1 Main Objective 2

1.3.2 Specific Objectives 2

1.4 Research Scope 3

1.4.1 Technical Boundaries 3

1.4.2 Functional Boundaries 3

1.5 Significance of the Research 4

1.5.1 Theoretical and Technical Significance 4

1.5.2 Societal and Local Impact (Uganda Context) 4

2 Literature Review **6**

2.1 Core Literature: Firefighting Robots 6

2.2 Fire Detection and IoT-Based Systems 6

2.3 Sensing and Perception in Mobile Robotics 6

2.4 Path Planning and Navigation Algorithms 7

2.5 Learning-Based Navigation and AI Approaches 7

2.6 Embedded Systems and Robot Architecture 7

2.7 Obstacle Avoidance and Reactive Control 8

2.8 Firefighting Robot Integration Systems 8

2.9 Critical Analysis and Research Gap 8

3 Methodology and System Design **9**

3.1 Introduction 9

3.2 Hardware Selection and Justification 10

3.3 Hardware Assembly 13

3.4 System Architecture 13

3.5 User Interface Design 15

4 System Implementation **17**

4.1 System Overview 17

4.2 Hardware Integration 17

4.3 Software Implementation 19

4.3.1 Navigation and Path Planning Implementation 19

4.3.2 Fire Localization and Coordinate Generation 20

4.3.3 Fire Detection and Suppression Algorithm 21

4.4 Operation modes 24
4.4.1 Autonomous Mode 24
4.4.2 Manual mode 24
4.5 Hardware and Software Communication 24
4.6 Obstacle Avoidance 25
4.7 User Interface Implementation 27
4.8 System Integration 27
4.9 Implementation Challenges 28

5 Evaluation and Results **29**
5.1 Introduction 29
5.2 Fire Detection Performance 29
5.3 Fire Localization Accuracy 30
5.4 Evaluation of the A* Navigation Algorithm 30
5.5 Obstacle Avoidance Performance 31
5.6 Response Time Analysis 31
5.7 Fire Suppression Effectiveness 32
5.8 User Interface Performance 32

6 Conclusion and Recommendations **34**
6.1 Conclusion 34
6.2 Recommendations 34
6.3 Final Remarks 35

7 References **36**

List of Figures

1 Figure showing overall project methodology. 10
2 Figure showing how the three main layers coordinate for full functionality of the robot. 15
3 Figure showing the user interface 16
4 Final robot hardware assembly showing the integration of the Raspberry Pi, Arduino Uno, sensors, actuators, camera module, and fire suppression system. 19
5 Fire localization and coordination 21
6 Fire Detection and Suppression Flowchart 23
7 System Communication Flow 25
8 Obstacle Avoidance Flow Chart 26
9 Fire Detection distance vs Response time 29
10 Fire Localization accuracy 30
11 Overall System success Rate 32

List of Tables

1 Hardware Components and Functions 12
2 Fire Detection Results 29
3 Fire Localization Accuracy 30
4 Obstacle Avoidance Success Rate 31
5 Response Time Results 31
6 Fire Suppression Results 32

List of Abbreviations

- SARFF - Semi-Autonomous Robotic Fire Fighter
- ROS - Robot Operating System
- SLAM - Simultaneous Localization and Mapping
- DRL - Deep Reinforcement Learning
- DWA - Dynamic Window Approach
- DDQN - Double Deep Q-Network
- DQN - Deep Q-Network
- PPO - Proximal Policy Optimization
- SAC - Soft Actor-Critic
- AMCL - Adaptive Monte Carlo Localization
- KPI - Key Performance Indicator
- IoT - Internet of Things
- LiDAR - Light Detection and Ranging
- RGB-D - Red Green Blue-Depth
- PWM - Pulse Width Modulation
- FL - Fuzzy Logic

1 Introduction

1.1 Background

Fire outbreaks continue to pose serious threats to human life, property, and the environment worldwide. According to the World Health Organization (WHO), over 180,000 people die annually due to fire-related injuries, while millions suffer from burns and smoke inhalation [1]. Similarly, the National Fire Protection Association (NFPA) reported that in 2023, fires caused property losses worth billions of dollars globally and resulted in numerous firefighter casualties [2]. In Uganda and other developing countries, inadequate firefighting equipment and delayed emergency response times have contributed to increased fire-related losses in both residential and industrial areas [3].

Traditional firefighting relies heavily on human intervention, which exposes firefighters to life-threatening conditions such as extreme heat, toxic smoke, structural collapse, and explosions [4]. Despite improvements in protective gear and safety protocols, firefighters continue to face significant risks when operating in confined or hazardous environments [5]. In addition, delays in fire detection and response often allow fires to spread rapidly, increasing casualties and property damage [3].

The integration of robotic systems in firefighting presents a promising solution to these challenges. Robotic firefighters can operate in environments that are unsafe or inaccessible to humans, thereby reducing injury risks and improving response efficiency [6]. These robots can be equipped with smoke and flame sensors, thermal cameras, and gas detectors to autonomously detect fires and localize their sources. When combined with wireless communication, such systems can receive early fire alerts from environmental sensors installed in buildings, enabling faster response before human firefighters arrive.

To effectively operate in such environments, robotic firefighters must be capable of autonomous navigation and obstacle avoidance. This requires the implementation of intelligent path planning algorithms such as A*, Dijkstra's algorithm, or Rapidly-exploring Random Trees (RRT) to ensure optimal route selection toward the fire source [7]. Additionally, Simultaneous Localization and Mapping (SLAM) enables the robot to build and update maps of unknown environments in real time, which is crucial in smoke-filled or low-visibility conditions [8]. To further enhance safety and reliability, the system must also incorporate obstacle detection and avoidance mechanisms based on ultrasonic, infrared, or LiDAR sensors [9].

Research into semi-autonomous firefighting robots highlights their potential to reduce firefighter exposure and improve fire suppression efficiency [10]. These systems can perform autonomous pathfinding, target localization, and deploy fire-extinguishing mechanisms such as water, foam, or CO depending on the fire type. When integrated with Internet of Things (IoT)-based fire detection systems, such robots can act as first

responders, containing small fires before human teams arrive [4].

Therefore, developing a semi-autonomous robotic firefighter that combines fire detection, path planning, and obstacle avoidance capabilities is both timely and relevant [11]. Such a system enhances safety, reduces response time, and minimizes property loss. It also represents an important step toward intelligent disaster management systems capable of operating autonomously in dynamic and hazardous environments.

1.2 Problem Statement

Traditional path planning and obstacle avoidance methods such as A* and Dijkstra's algorithm are widely used in robotic navigation; however, they often face limitations in dynamic and cluttered environments such as fire scenes. In such environments, obstacles may change rapidly due to spreading fire, smoke, and structural instability, making static path planning less effective.

In addition, semi-autonomous firefighting robots require real-time decision-making capabilities to navigate safely while responding to fire incidents. Achieving reliable navigation and obstacle avoidance on low-cost embedded systems such as Raspberry Pi and Arduino presents further challenges due to limited processing power and memory constraints.

Therefore, there is a need to design and implement an efficient semi-autonomous firefighting robot that integrates reliable path planning, real-time obstacle detection, and responsive control mechanisms suitable for constrained hardware environments.

1.3 Objective

1.3.1 Main Objective

The main objective of this project is to design and implement a semi-autonomous firefighting robot capable of detecting, localizing, navigating to, and extinguishing fires using a low-cost embedded system.

1.3.2 Specific Objectives

1. To assemble and develop an integrated fire response system capable of detecting fire, estimating its location, providing real-time monitoring and control through a user interface, and suppressing the fire using a water pump mechanism.
   “‘
2. To implement the A* path planning algorithm for efficient and optimal navigation of the robot towards the fire location while avoiding obstacles.

3. To design and implement an ultrasonic-based obstacle detection and avoidance system to ensure safe navigation in dynamic indoor environments.
4. To evaluate the overall system performance based on key metrics such as fire detection accuracy, response time, navigation success rate, obstacle avoidance efficiency, and fire suppression effectiveness in a controlled environment.

1.4 Research Scope

This research focuses on the design and implementation of a semi-autonomous firefighting robot capable of detecting fire, navigating indoor environments, and performing basic fire suppression tasks. The system is developed using a low-cost embedded architecture consisting of a Raspberry Pi for high-level processing and an Arduino for low-level control.

The study involves both software development and physical prototyping. Testing is conducted in a controlled indoor environment to evaluate system performance in fire detection, navigation, obstacle avoidance, and fire suppression.

The scope of the research is limited by the following technical and functional constraints:

1.4.1 Technical Boundaries

- The system is implemented using a hybrid embedded architecture consisting of a Raspberry Pi and Arduino, with communication between the two units handled through serial communication.
- The navigation system is based on the A* path planning algorithm for route optimization and target acquisition.
- The system does not rely on external datasets; instead, sensor data is obtained in real time from onboard sensors such as flame sensors and ultrasonic sensors during testing.
- The study focuses on a rule-based and algorithmic approach to obstacle avoidance and does not include machine learning or reinforcement learning methods.

1.4.2 Functional Boundaries

- The robot operates in indoor, structured environments with static and dynamic obstacles. Outdoor navigation is excluded.
- Fire detection is based on flame sensor input, and fire suppression is limited to a small-scale water pump mechanism suitable for extinguishing minor fires.

- The system operates in a semi-autonomous mode. By default, the robot can function autonomously, performing navigation, fire detection, and suppression without human intervention. However, the user can switch to manual mode at any time to directly control robot movements through the user interface.

- A user interface is included for real-time monitoring, system feedback, and switching between autonomous and manual control modes.

1.5 Significance of the Research

This research provides important theoretical, technical, and societal contributions in the field of low-cost autonomous robotics for disaster response systems, particularly in resource-constrained environments.

1.5.1 Theoretical and Technical Significance

- **Development of a Low-Cost Autonomous Firefighting System:** The project demonstrates the design and implementation of a semi-autonomous firefighting robot capable of fire detection, navigation, and suppression using affordable embedded hardware, contributing to practical robotics solutions in constrained environments.

- **Integration of Hybrid Embedded Architecture:** The research presents a working model of a dual-processor system combining Raspberry Pi for high-level decision-making and Arduino for low-level control, providing a scalable architecture for future embedded robotics systems.

- **Implementation of Real-Time Path Planning and Obstacle Avoidance:** The use of the A* algorithm for navigation and ultrasonic-based obstacle avoidance provides a practical approach to autonomous movement in structured indoor environments.

- **Sensor-Based Fire Detection and Localization:** The study contributes a practical method of fire detection and approximate localization using low-cost sensors, suitable for real-world deployment in early fire response systems.

1.5.2 Societal and Local Impact (Uganda Context)

- **Reduction of Risk to Human Firefighters:** The system acts as a first-response robotic unit capable of entering hazardous environments, reducing direct exposure of firefighters to dangerous conditions such as heat, smoke, and structural instability.

- **Improved Emergency Response Efficiency:** By enabling faster detection, navigation, and suppression of small fires, the system has the potential to reduce response delays and minimize fire spread, thereby lowering property and infrastructure losses.

- **Promotion of Affordable Robotics Solutions:** The use of low-cost components such as Raspberry Pi and Arduino demonstrates that effective robotic systems can be developed and deployed in developing countries without requiring expensive infrastructure.

- **Academic and Educational Value:** The project provides a practical reference for students and researchers interested in embedded systems, robotics, and autonomous navigation, supporting further innovation in the field.

2 Literature Review

2.1 Core Literature: Firefighting Robots

Firefighting robots have been widely researched as a means of reducing human exposure to hazardous environments. Early and modern prototypes demonstrate that mobile robotic systems can be effectively used for fire detection and suppression tasks using low-cost embedded hardware [6, 12].

Most existing systems are built using microcontroller-based architectures such as Arduino combined with simple flame sensors and actuation mechanisms like water pumps. These systems demonstrate that basic autonomous fire suppression is achievable in structured environments, although their operational scope is often limited.

Several implementations integrate ultrasonic sensors for obstacle detection and infrared flame sensors for fire localization. Such designs show that simple sensor fusion can improve navigation and detection performance in indoor environments, although they remain limited in adaptability and scalability [13–16].

Despite these advancements, most systems are tested in controlled environments and lack robust autonomy in complex indoor fire scenarios.

2.2 Fire Detection and IoT-Based Systems

Fire detection is a critical component of firefighting robotics and has been extensively studied in IoT-based systems. Many modern approaches integrate distributed sensors and wireless communication to improve early fire detection and response times [9, 11].

IoT-enabled systems allow remote monitoring and early alerting, which significantly improves response efficiency in smart environments. Drone-based and robotic fire detection systems also utilize lightweight AI models for real-time detection and classification of fire events [17–19].

However, most IoT-based fire detection systems focus on monitoring rather than autonomous navigation and suppression, limiting their effectiveness in active firefighting scenarios.

2.3 Sensing and Perception in Mobile Robotics

Perception systems in mobile robots rely heavily on sensors such as infrared flame detectors, ultrasonic range finders, LiDAR, and camera-based systems. Ultrasonic sensors are widely used for short-range obstacle detection due to their simplicity and affordability [13, 20].

Advanced robotic systems combine multiple sensors to improve environmental awareness. Sensor fusion approaches integrating LiDAR, vision, and range sensors improve robustness in complex environments [7, 21].

However, sensor performance degrades in fire conditions due to smoke, heat distortion, and reflective surfaces, which presents a major challenge for reliable autonomous operation.

2.4 Path Planning and Navigation Algorithms

Path planning is a fundamental component of autonomous robotics. Classical algorithms such as A* and Dijkstra remain widely used due to their efficiency and optimality in static environments [22, 23].

These algorithms compute optimal paths based on cost functions and grid-based representations of the environment. In practical robotic applications, global planning is often combined with local obstacle avoidance systems to handle dynamic changes during navigation.

Research also explores hybrid approaches combining heuristic optimization and reactive navigation strategies to improve adaptability in uncertain environments [20, 24].

Despite their effectiveness, traditional algorithms struggle in highly dynamic environments where real-time adaptation is required.

2.5 Learning-Based Navigation and AI Approaches

Recent advances in robotics explore the use of Deep Reinforcement Learning (DRL) for navigation and decision-making. DRL enables robots to learn optimal navigation policies directly from interaction with the environment [25–27].

These methods offer advantages in complex and dynamic environments where classical algorithms may fail. However, DRL approaches often require high computational resources and are difficult to deploy on low-cost embedded systems.

Although promising, most DRL-based navigation systems are still experimental and rarely applied in real-world firefighting robots.

2.6 Embedded Systems and Robot Architecture

Low-cost robotic systems commonly adopt a hybrid architecture combining a microcontroller and a single-board computer. The Arduino microcontroller handles real-time sensor processing and motor control, while the Raspberry Pi performs higher-level computation and decision-making [10, 28].

This architecture improves modularity, scalability, and system reliability. Communication between components is typically achieved through serial communication protocols, enabling coordinated control of sensing, planning, and actuation.

Such architectures are widely used in educational and prototype-level robotics due to their affordability and flexibility.

2.7 Obstacle Avoidance and Reactive Control

Obstacle avoidance is essential for safe robot navigation in unknown environments. Reactive control methods using ultrasonic sensors are widely adopted in low-cost robotic systems for real-time collision avoidance [20, 24].

These methods allow robots to respond dynamically to unexpected obstacles. However, they may fail in complex scenarios where sensor noise or occlusion occurs.

More advanced approaches combine global planning with local reactive control to improve navigation robustness in dynamic environments.

2.8 Firefighting Robot Integration Systems

Recent research emphasizes the importance of integrating detection, navigation, and suppression into a unified system. Firefighting robots are expected to operate autonomously from fire detection to suppression without continuous human intervention [29, 30].

However, most existing systems lack full integration and are often limited to partial implementations such as either detection or navigation, but not both in a complete pipeline.

2.9 Critical Analysis and Research Gap

Despite significant progress in robotics and fire safety systems, several gaps remain in the literature:

- Lack of fully integrated systems combining fire detection, navigation, obstacle avoidance, and suppression.
- Limited deployment of low-cost embedded architectures for real-time autonomous firefighting.
- Insufficient testing of robotic systems in realistic indoor fire conditions with smoke and dynamic obstacles.
- Limited practical implementation of advanced AI methods in resource-constrained environments.
- Lack of user-interactive semi-autonomous systems with manual override capability.

Therefore, there is a need for a cost-effective semi-autonomous firefighting robot that integrates flame detection, ultrasonic sensing, A* path planning, and a user interface for real-time control and monitoring within a unified embedded architecture.

3 Methodology and System Design

3.1 Introduction

This chapter presents the methodology used in the design, development, implementation, and evaluation of the Semi-Autonomous Fire-Fighting Mobile Robot. The project adopted a system integration approach that combines hardware and software components to achieve autonomous fire detection, navigation, obstacle avoidance, and fire suppression.

The system consists of a Raspberry Pi running the Robot Operating System (ROS), an Arduino Uno for low-level hardware control, a camera module for fire localization, a flame detection system, an ultrasonic sensor mounted on a servo motor for obstacle detection, and a relay-controlled water pump for fire suppression.

The robot supports both autonomous and manual modes of operation. In autonomous mode, the robot independently detects and navigates toward a fire source before activating the suppression mechanism. In manual mode, an operator can remotely monitor and control the robot through a graphical user interface. Figure 1 illustrates the overall methodology adopted for the development and implementation of the proposed system.

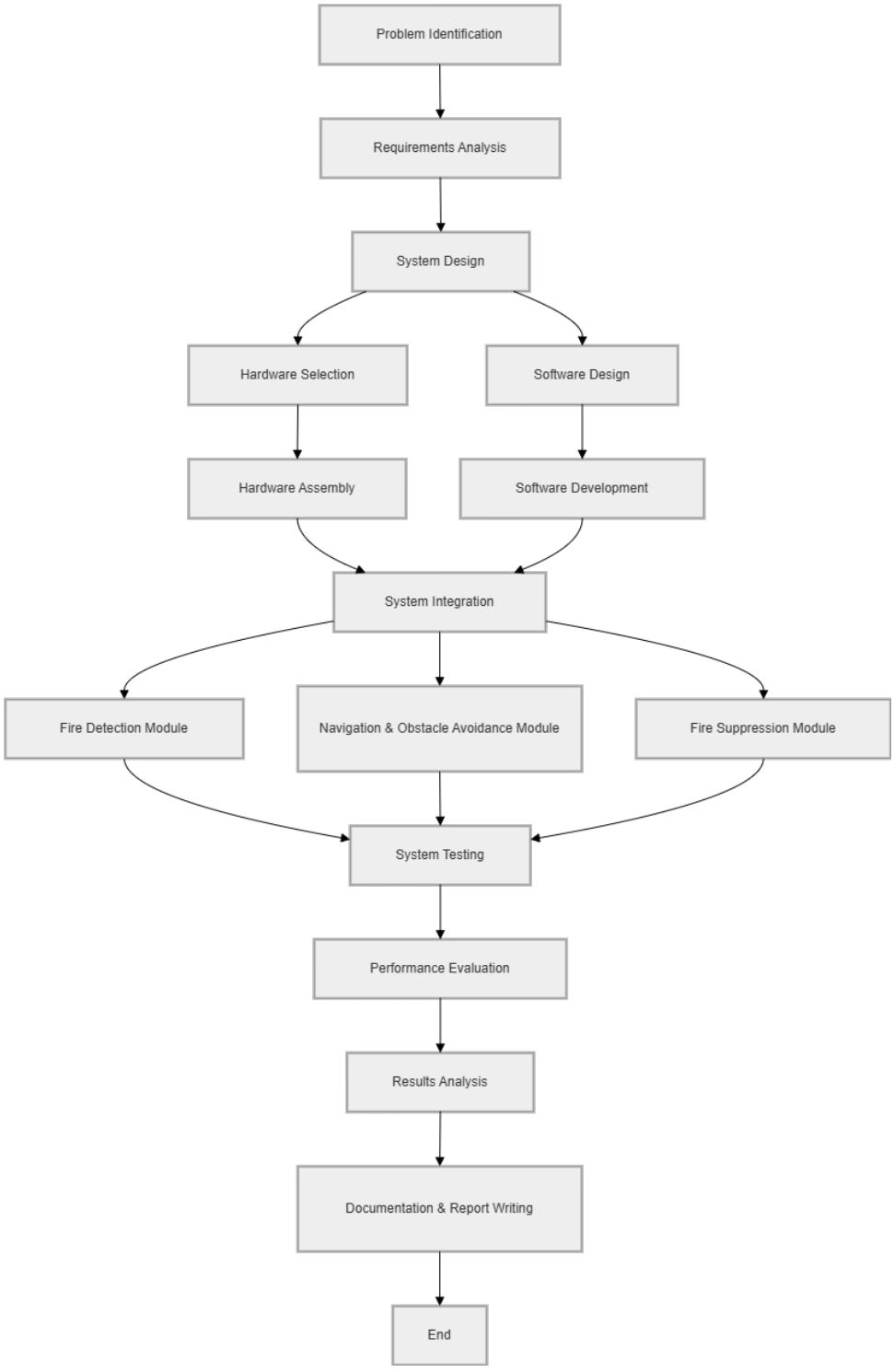


Figure 1: Figure showing overall project methodology.

3.2 Hardware Selection and Justification

The hardware components used in the development of the autonomous fire-fighting robot were selected based on their functionality, compatibility, reliability, cost-effectiveness, and suitability for the project requirements. The system required components capable of supporting autonomous navigation, fire detection, wireless communication, and fire

suppression. Each component plays a specific role in ensuring the efficient operation of the robot. Table 1 presents the selected hardware components and their respective functions within the system.

Table 1: Hardware Components and Functions

Hardware Component	**Purpose**	**Justification**
Raspberry Pi	Primary processing unit for image processing, navigation, and system coordination	Selected for its computational power, ROS support, and ability to interface with sensors and microcontrollers
Arduino Uno	Handles low-level control tasks such as motor control, ultrasonic sensing, servo actuation, and relay switching	Chosen for its reliability, ease of programming, and compatibility with hardware modules
Camera Module	Captures environmental visuals and supports fire detection and live streaming to the user interface	Provides real-time visual feedback and assists in identifying fire locations
Flame Detection Module	Detects presence of fire and generates coordinate-related data for navigation	Enables early fire detection and supports autonomous response
Ultrasonic Sensor	Measures distance to obstacles for navigation and collision avoidance	Provides accurate real-time distance sensing for safe movement
Servo Motor	Rotates ultrasonic sensor for multi-directional scanning	Allows wider environmental coverage and improved obstacle detection
Relay Module	Controls switching of the water pump while providing electrical isolation from Arduino	Ensures safe control of high-power components using low-power signals
Water Pump	Acts as the fire suppression system by spraying water on detected fires	Provides direct physical response to extinguish fires
Motor Driver	Controls speed and direction of DC motors	Enables precise movement control of the robot
Power Supply	Supplies electrical power to all system components	Ensures stable and sufficient power for continuous system operation

3.3 Hardware Assembly

The hardware components were mounted on a mobile robotic chassis. The Raspberry Pi and Arduino Uno were securely installed at the center of the chassis to ensure stability and balanced weight distribution.

The camera and flame detection modules were positioned at the front of the robot to maximize visibility and fire detection capability. The ultrasonic sensor was mounted on a servo motor to allow environmental scanning in multiple directions.

The relay module and water pump were connected near the water reservoir to simplify wiring and improve maintenance.

3.4 System Architecture

The Semi-Autonomous Fire-Fighting Mobile Robot was implemented using a layered architecture consisting of three main layers: Perception, Processing, and Control. Each layer performs specific functions and communicates with the others to achieve autonomous fire detection, navigation, obstacle avoidance, and fire suppression.

1. **Perception Layer**

   This layer is responsible for environmental sensing and data acquisition. The following tools and techniques were used:

   - ESP32-CAM module for real-time image acquisition and wireless transmission of video frames.
   - Flame sensor module for detecting the presence and direction of fire.
   - HC-SR04 ultrasonic sensor for obstacle detection and distance measurement.
   - Sensor data acquisition techniques for continuous monitoring of the environment.

   Sensor readings and camera data are transmitted to the Raspberry Pi for processing.

2. **Processing Layer**

   The Raspberry Pi serves as the main processing unit. The following tools and methods were used:

   - Raspberry Pi 4 running Raspberry Pi OS.
   - Python programming language for implementing navigation, fire localization, and system control algorithms.
   - OpenCV library for image processing and fire localization from camera frames.

- A* (A-Star) path planning algorithm for autonomous navigation towards the detected fire location.
- Obstacle avoidance algorithm based on ultrasonic sensor measurements.
- Serial communication (UART) for exchanging commands with the Arduino Uno.
- Bluetooth communication for manual control and GUI updates.

The Raspberry Pi receives sensor information, determines the fire coordinates, computes the optimal path to the fire source, and sends movement commands to the Arduino.

3. **Control Layer**

   The Arduino Uno performs low-level hardware control using the following techniques:

   - Embedded C/C++ programming using the Arduino IDE.
   - Motor control algorithms for robot movement.
   - PWM (Pulse Width Modulation) for speed control of DC motors.
   - Servo motor control for ultrasonic sensor scanning.
   - Relay-based switching for activating and deactivating the water pump.

   The Arduino receives movement and actuation commands from the Raspberry Pi through UART serial communication and executes the corresponding actions. “‘

Communication Between Layers

The three layers communicate through defined interfaces:

- The Perception Layer sends sensor readings and camera data to the Processing Layer.
- The Processing Layer analyzes the incoming data and generates navigation and fire-suppression decisions.
- Commands are transmitted from the Processing Layer to the Control Layer using UART serial communication between the Raspberry Pi and Arduino Uno.
- Feedback from sensors is continuously returned to the Processing Layer, enabling real-time obstacle avoidance and navigation updates.

Figure 2 illustrates the interaction and communication among the three architectural layers.

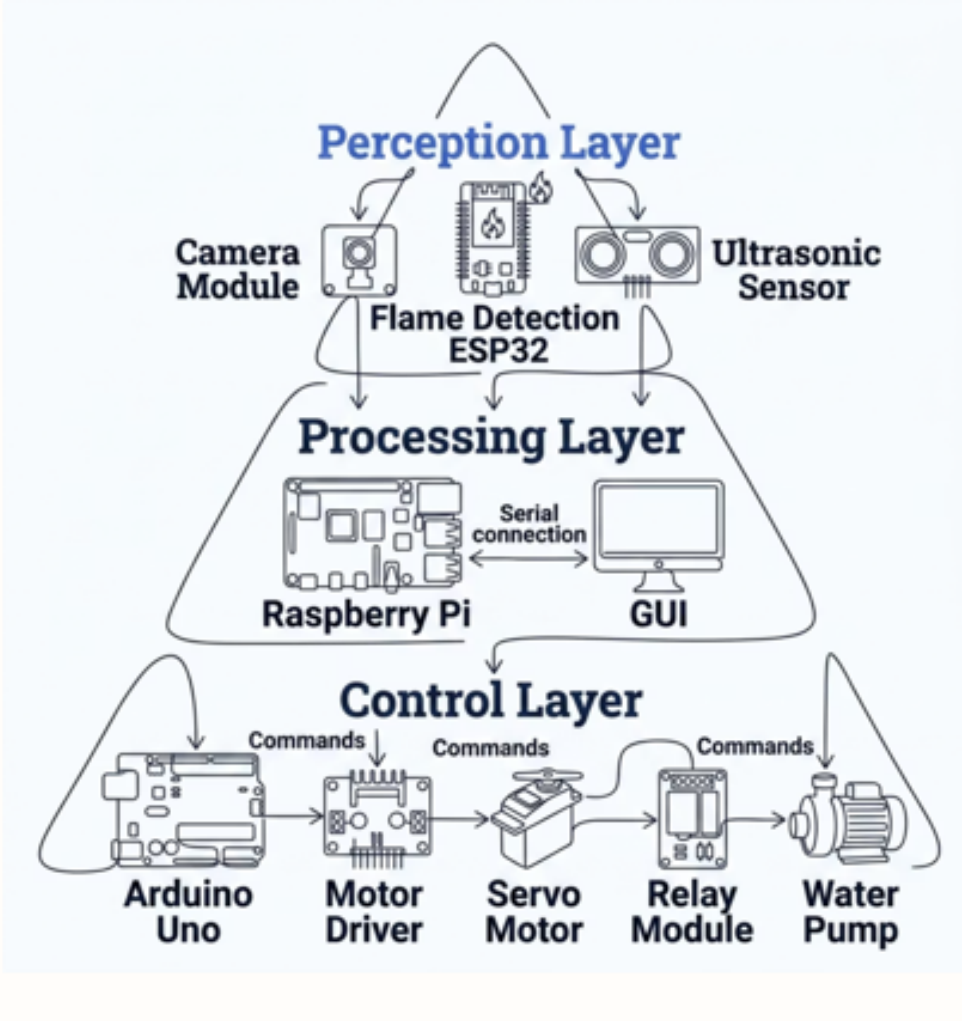


Figure 2: Figure showing how the three main layers coordinate for full functionality of the robot.

3.5 User Interface Design

A web-based graphical user interface (GUI) was developed using HTML, CSS, and JavaScript to provide operators with real-time monitoring and control of the robot. The interface was designed to be accessible through a web browser and to provide an intuitive means of interacting with the system.

The front-end was implemented using HTML for page structure, CSS for styling and layout, and JavaScript for dynamic interaction and communication with the robot system. The interface communicates with the Raspberry Pi, which acts as the central processing unit and server for system data exchange.

The GUI was designed using a client-server architecture, where the Raspberry Pi processes sensor data and transmits status information to the user interface. User commands

issued through the interface are sent back to the Raspberry Pi for execution.

The interface provides:

- Live video streaming from the onboard camera.
- Display of robot position.
- Display of fire coordinates.
- Environment map visualization.
- Manual movement controls.
- Autonomous mode activation.
- System status monitoring.
- Fire suppression controls.

The interface serves as the primary communication platform between the user and the robot, enabling both autonomous supervision and manual intervention when required.

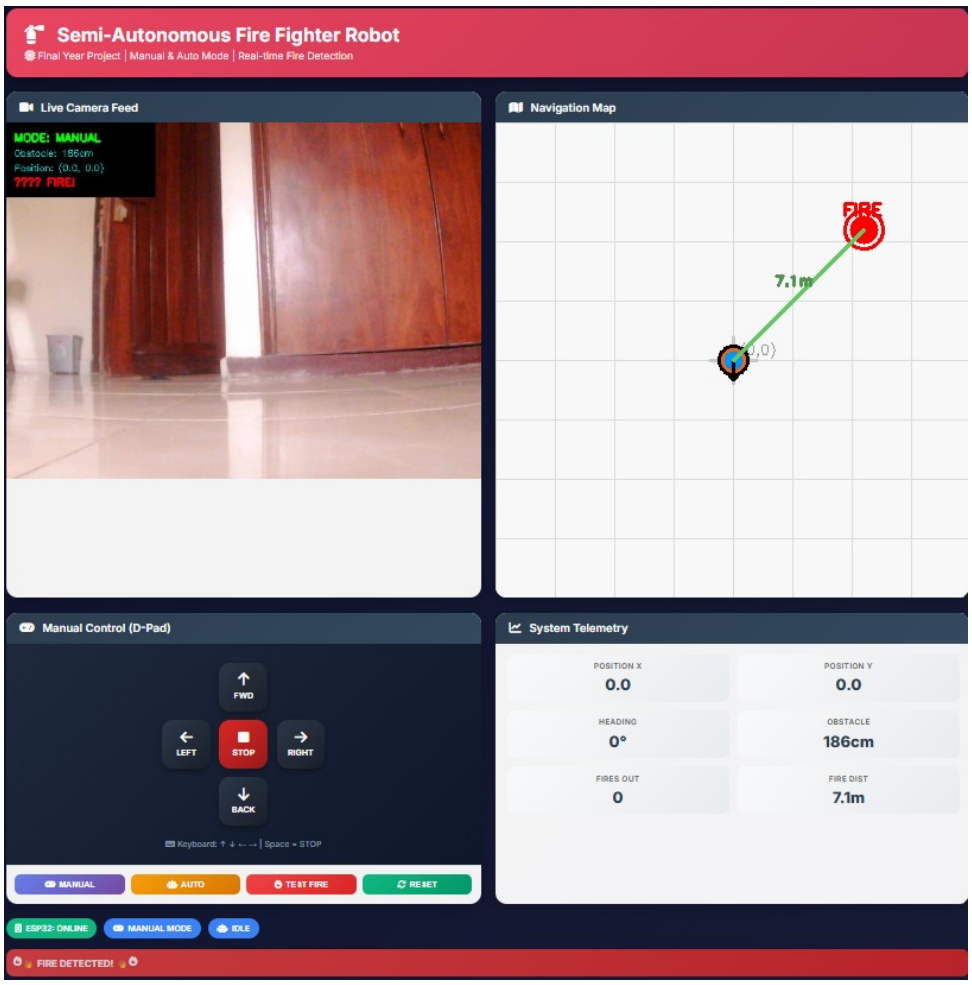


Figure 3: Figure showing the user interface

4 System Implementation

4.1 System Overview

This chapter presents the implementation of the semi-autonomous fire-fighting robot. The system integrates multiple hardware and software components to achieve fire detection, localization, autonomous navigation, and fire suppression.

The system follows a distributed architecture consisting of three main processing units: the Raspberry Pi, Arduino Uno, and ESP32. The Raspberry Pi acts as the central processing unit responsible for high-level decision-making, image processing, and navigation control. The Arduino Uno handles low-level hardware control such as motor driving, ultrasonic sensing, servo actuation, and relay switching. The ESP32 is responsible for fire detection using a flame sensor and generating coordinate data representing the fire location.

Communication between ESP32 and Raspberry Pi is achieved using Bluetooth, while communication between Raspberry Pi and Arduino Uno is handled through serial communication.

4.2 Hardware Integration

The Semi-Autonomous Fire-Fighting Mobile Robot was developed through the integration of multiple hardware modules, each performing specialized functions while communicating through wired and wireless interfaces. The integration strategy was designed to ensure reliable sensing, processing, navigation, and fire suppression operations.

The Raspberry Pi 4 served as the central processing unit and coordinated communication among all system components. Due to its higher computational capability, the Raspberry Pi was responsible for executing navigation algorithms, processing sensor information, managing camera operations, and coordinating system-wide decision-making.

The ESP32-based flame detection subsystem was integrated to perform fire detection and coordinate generation. Communication between the ESP32 and Raspberry Pi was implemented using Bluetooth technology, enabling wireless transmission of fire coordinates from the detection unit to the robot. This wireless approach eliminated the need for additional cabling and allowed flexible placement of the flame detection module within the environment.

The Raspberry Pi communicated with the Arduino Uno through Universal Asynchronous Receiver Transmitter (UART) serial communication. Once fire coordinates were received from the ESP32, the Raspberry Pi processed the information and generated navigation commands. These commands were transmitted via the serial interface to the Arduino Uno, which acted as the low-level hardware controller.

The Arduino Uno was integrated with the motor driver, ultrasonic sensor, servo motor,

and relay module. The motor driver received Pulse Width Modulation (PWM) signals from the Arduino to control the speed and direction of the DC motors. The ultrasonic sensor continuously provided obstacle distance measurements to the Arduino, while the servo motor enabled sensor scanning during obstacle avoidance operations.

For fire suppression, the Arduino controlled a relay module connected to the water pump. The relay provided electrical isolation between the low-voltage control circuitry and the higher-power pump circuit, ensuring safe operation. Upon receiving a fire suppression command from the Raspberry Pi, the Arduino activated the relay, which switched on the water pump to extinguish the detected fire.

The camera module was directly connected to the Raspberry Pi and provided real-time visual data. The Raspberry Pi processed and streamed the video feed to the graphical user interface, allowing users to monitor robot movement and system operation remotely.

Power integration was achieved using separate power distribution paths. A regulated power supply was provided to the Raspberry Pi and control electronics, while the DC motors and water pump were powered through a dedicated high-current supply line. This approach minimized voltage fluctuations and prevented performance degradation caused by the high current demands of the motors and pump.

The resulting hardware architecture enabled seamless communication between sensing, processing, and actuation subsystems. Bluetooth communication facilitated wireless fire coordinate transmission between the ESP32 and Raspberry Pi, while UART serial communication enabled reliable command exchange between the Raspberry Pi and Arduino Uno. Together, these communication mechanisms allowed coordinated operation of all hardware components within the robot system.

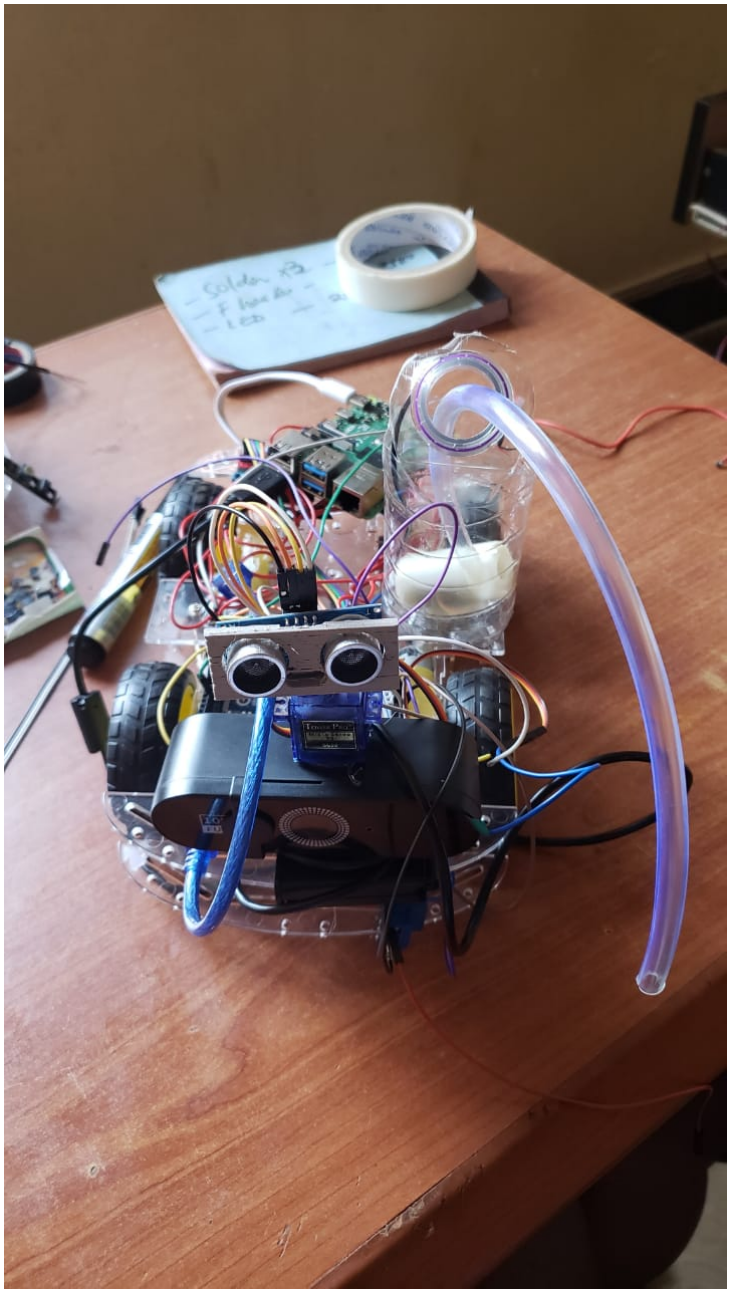


Figure 4: Final robot hardware assembly showing the integration of the Raspberry Pi, Arduino Uno, sensors, actuators, camera module, and fire suppression system.

4.3 Software Implementation

4.3.1 Navigation and Path Planning Implementation

The navigation subsystem was implemented on the Raspberry Pi using the A* (A-Star) search algorithm. A* is a graph-based search algorithm that identifies the shortest path between a starting position and a target location while minimizing traversal cost. The algorithm combines the actual distance traveled from the start node with a heuristic estimate of the remaining distance to the goal node [31].

Upon receiving fire coordinates from the fire detection module, the Raspberry Pi generates a path from the robot's current position (start node) to the target fire location (goal node). The algorithm evaluates available paths and selects the route with the lowest estimated total cost.

The environment is represented as a grid-based map, where each cell corresponds to either a traversable location or an obstacle. During navigation, obstacle information obtained from the ultrasonic sensor is used to update the map and prevent the robot from traversing blocked paths. The robot continuously updates its understanding of the environment as it moves.

When an obstacle is detected during movement, the navigation map is updated and the A* algorithm is executed again to generate an alternative path. This dynamic replanning capability enables the robot to navigate efficiently toward the fire source while avoiding collisions.

The A* algorithm was selected because it provides efficient path planning, guarantees an optimal path when an admissible heuristic is used, and is computationally suitable for implementation on the Raspberry Pi. This ensures that the robot can navigate toward the fire location while minimizing travel distance even in dynamic environments containing unexpected obstacles [22].

The A* evaluation function is given by:

$$f(n) = g(n) + h(n) \tag{1}$$

where:

- $g(n)$ is the actual cost from the start node to node n,
- $h(n)$ is the heuristic estimate of the cost from node n to the goal node,
- $f(n)$ is the estimated total cost of the path through node n.

4.3.2 Fire Localization and Coordinate Generation

Fire localization is achieved using the camera module in combination with an ESP32-based flame detection system.

A coordinate reference system is established within the operating environment, where the center of the room is defined as the origin (0,0). This provides a consistent frame of reference for mapping fire locations.

The ESP32 is integrated with the flame sensor and is responsible for detecting fire presence. When a fire is detected, the system determines the relative position of the flame detection module within the environment and generates corresponding coordinate data. These coordinates are embedded within the ESP32 system logic and represent the estimated location of the fire source relative to the reference origin.

Once a fire event is confirmed, the ESP32 transmits the generated coordinates to the Raspberry Pi via a communication interface (e.g., Wi-Fi, serial, or MQTT depending on implementation). The Raspberry Pi then uses these coordinates as the target destination for navigation and path planning. The coordinate output enables the robot to localize the fire source, compute the movement trajectory, and initiate autonomous navigation toward the detected fire.

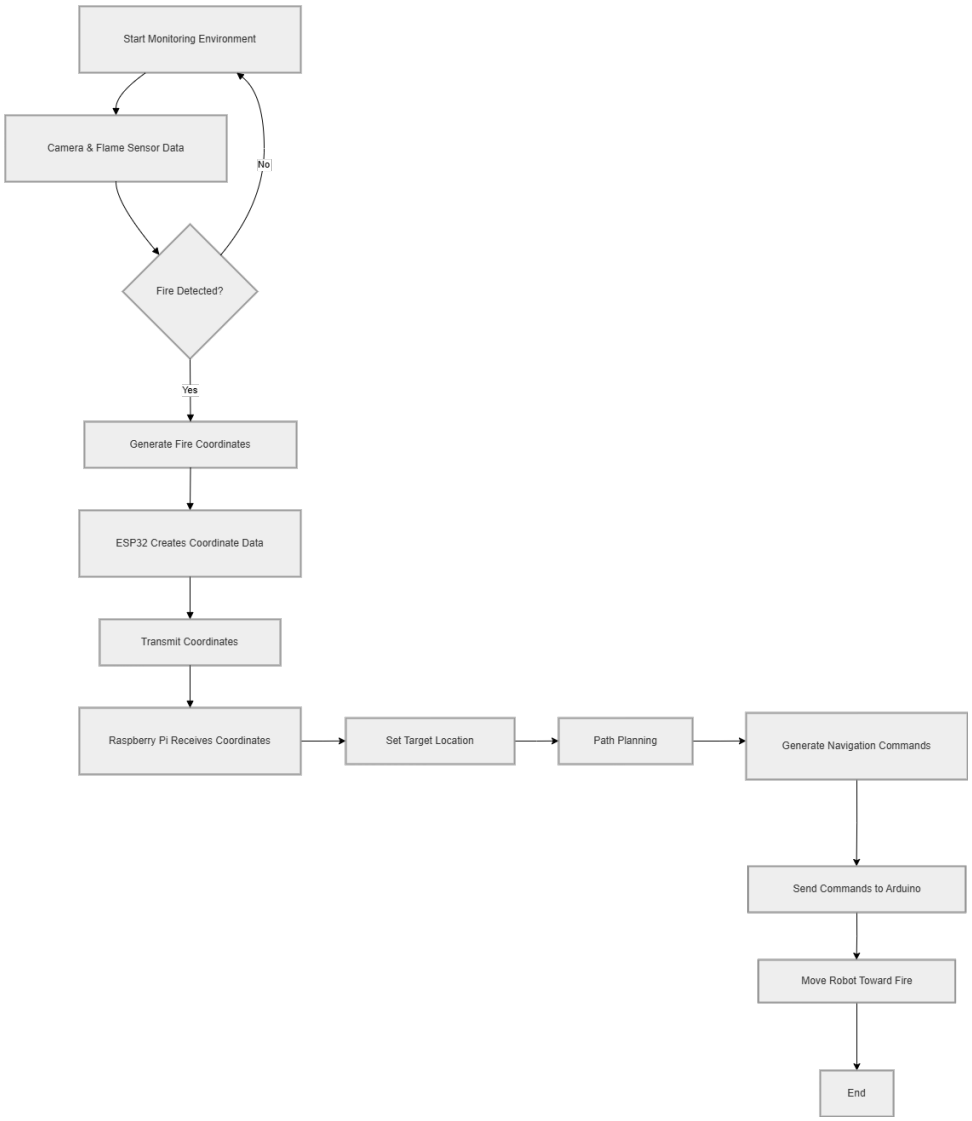


Figure 5: Fire localization and coordination

4.3.3 Fire Detection and Suppression Algorithm

The fire detection process operates continuously throughout robot operation.

When a fire is detected, the following steps are executed:

1. The fire coordinates are generated.

2. The Raspberry Pi calculates the required navigation path.

3. The robot moves toward the target location.
4. Obstacle avoidance remains active during navigation.
5. Upon reaching the fire source, the Raspberry Pi sends a suppression command.
6. The Arduino activates the relay module.
7. The water pump is switched on.
8. Water is sprayed until the fire is extinguished.
9. The water pump is turned off.

Figure 11 shows the Fire Detection and Suppression Flowchart.

Figure: Fire Detection and Suppression Flowchart

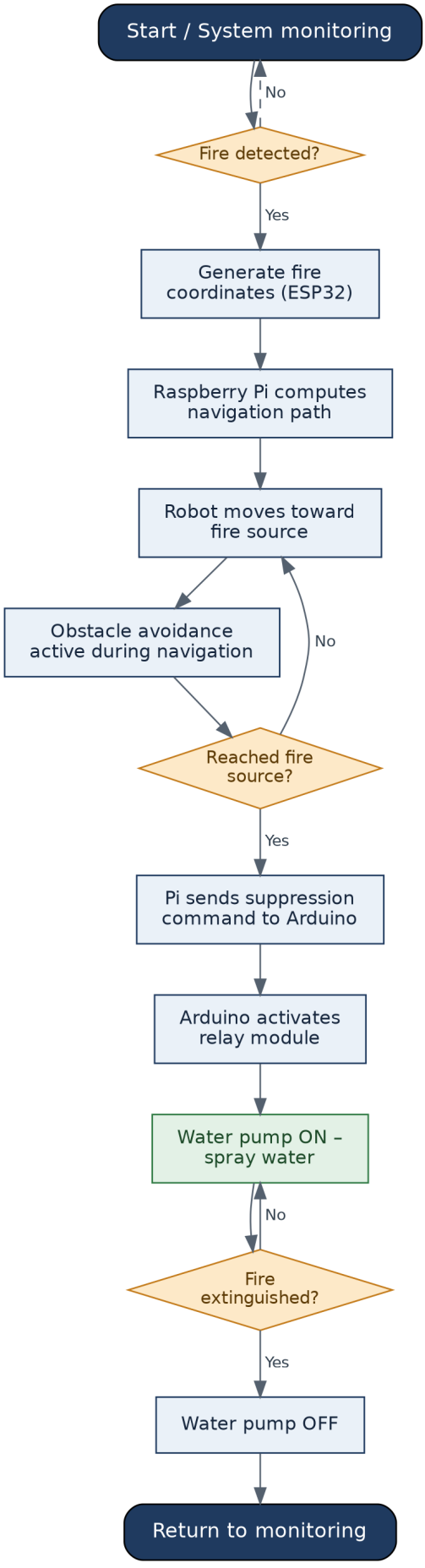


Figure 6: Fire Detection and Suppression Flowchart

4.4 Operation modes

4.4.1 Autonomous Mode

In autonomous mode, the robot independently performs fire detection, localization, navigation, obstacle avoidance and fire suppression.

Once a fire is detected and coordinates are generated, the Raspberry Pi computes navigation commands and continuously guides the robot toward the target location. Upon reaching the fire source, the relay module activates the water pump automatically.

4.4.2 Manual mode

In manual mode, the operator controls robot movement using the graphical user interface.

The live video feed allows remote observation of the environment while directional controls enable movement commands to be issued manually. Fire suppression can also be activated by the operator when necessary.

4.5 Hardware and Software Communication

Communication within the system follows a hierarchical architecture.

The flame detection module and camera provide fire localization information to the Raspberry Pi. The Raspberry Pi processes the information and generates movement commands based on the robot's current position and target coordinates.

The Raspberry Pi communicates with the Arduino through serial communication. The Arduino executes motor commands, controls the servo-mounted ultrasonic sensor, collects obstacle information, and manages the relay module connected to the water pump.

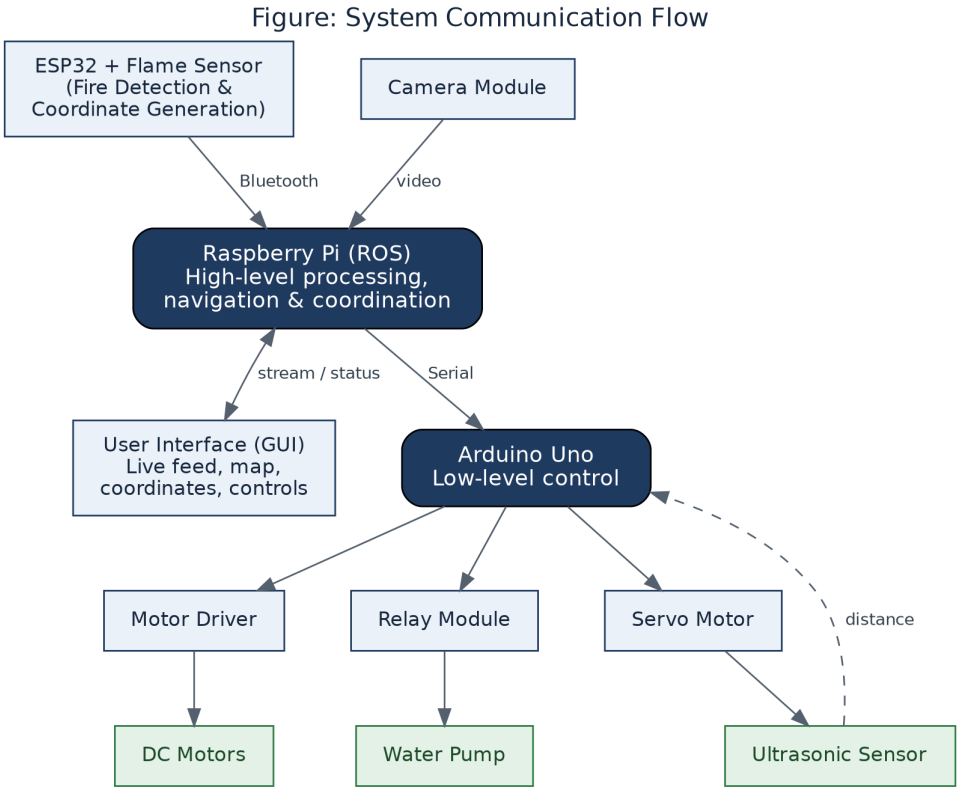


Figure 7: System Communication Flow

4.6 Obstacle Avoidance

Obstacle avoidance is achieved using an ultrasonic sensor mounted on a servo motor.

The servo rotates the sensor to scan three directions: front, left, and right.

- Front
- Left
- Right

The sensor scans only the front, left, and right directions because these are the directions relevant to the robot's navigation decisions. Since the robot is designed to move forward toward the fire source, obstacles located behind the robot do not influence path selection. Limiting the scan to three directions reduces processing time and enables faster obstacle avoidance decisions.

The obstacle avoidance process is as follows:

1. Measure the distance directly ahead.
2. If the path is clear, continue moving forward.
3. If an obstacle is detected, stop movement.

4. Rotate the sensor to the left and record the distance.
5. Rotate the sensor to the right and record the distance.
6. Compare both measurements.
7. Select the direction with the largest available space.
8. Turn toward the selected direction.
9. Resume forward movement.

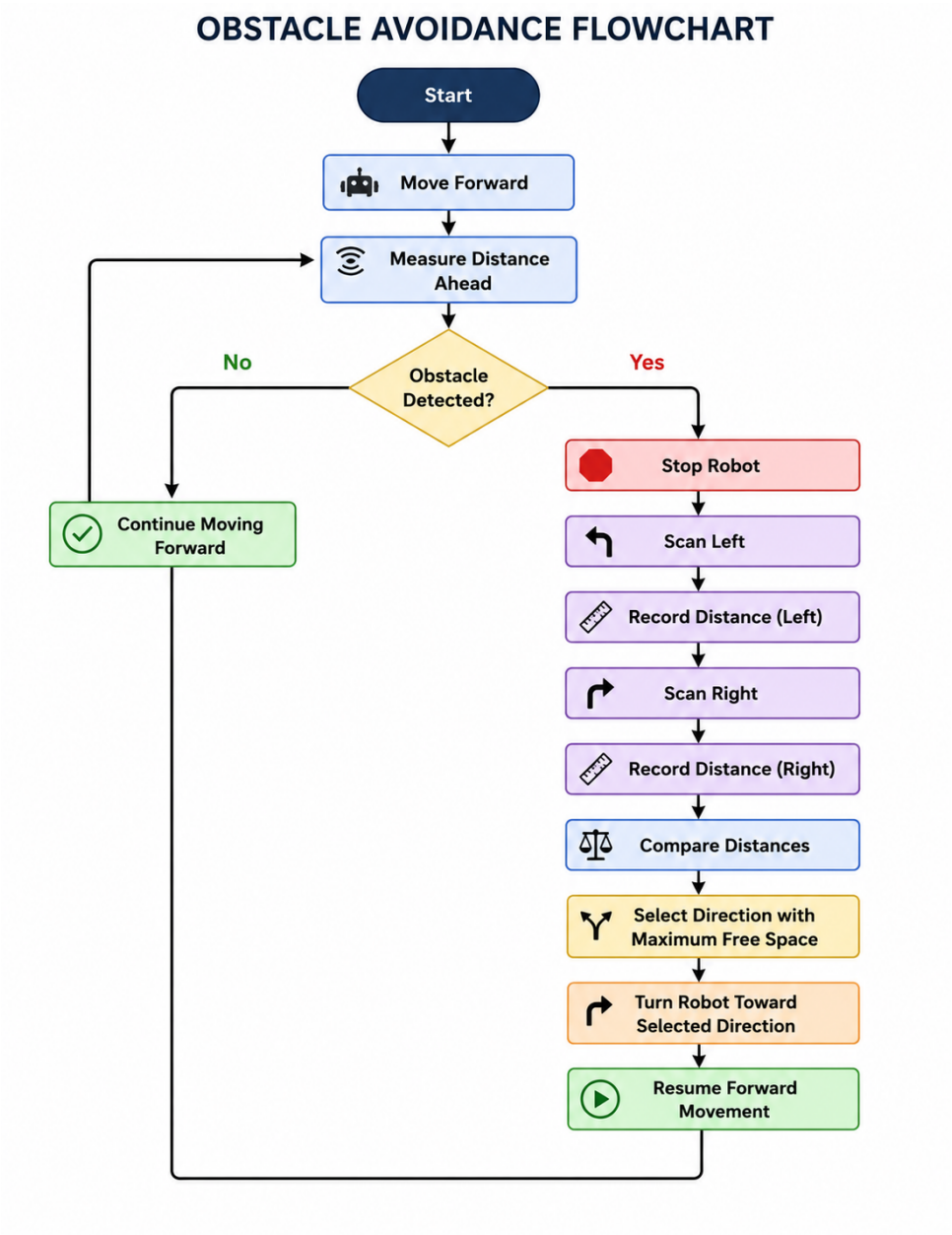


Figure 8: Obstacle Avoidance Flow Chart

4.7 User Interface Implementation

A graphical user interface (GUI) was developed to allow real-time monitoring and control of the system.

The interface includes:

- Live camera feed from the robot
- Display of robot position on a map
- Display of fire coordinates received from ESP32
- Manual control options for movement and pump control
- System status indicators (idle, moving, fire detected, suppressing fire)

The interface communicates with the Raspberry Pi to display real-time system updates.

4.8 System Integration

The system integrates all components into a unified semi-autonomous fire-fighting platform.

The complete data flow is:

1. ESP32 detects fire using flame sensor
2. ESP32 generates coordinate data
3. Coordinates are transmitted to Raspberry Pi via Bluetooth
4. Raspberry Pi processes coordinates and determines navigation path
5. Raspberry Pi sends movement commands to Arduino Uno via serial communication
6. Arduino controls motors and ultrasonic scanning
7. Robot navigates toward fire location
8. Upon arrival, Arduino activates relay module
9. Water pump is triggered to extinguish fire
10. Camera provides continuous feedback to the user interface

4.9 Implementation Challenges

- Power supply distribution issues due to high motor and pump current demand
- Flame sensor noise caused by environmental lighting conditions
- Bluetooth communication delays between ESP32 and Raspberry Pi
- Motor instability requiring calibration and tuning
- System integration complexity across multiple controllers

5 Evaluation and Results

5.1 Introduction

This section presents the results obtained from testing the autonomous fire-fighting robot system. The evaluation was conducted in a controlled indoor environment to assess system performance in fire detection, localization, navigation,path planning, obstacle avoidance and fire suppression. The results are analyzed using quantitative metrics and qualitative observations.

5.2 Fire Detection Performance

The system was tested under different distances and environmental conditions to evaluate flame detection reliability.

Table 2: Fire Detection Results

Distance from Sensor (m)	Detection Status	Response Time (s)
2	Detected	1.2
5	Detected	1.4
10	Detected	2
15	Weak Detection	2.5
24	Not Detected	–

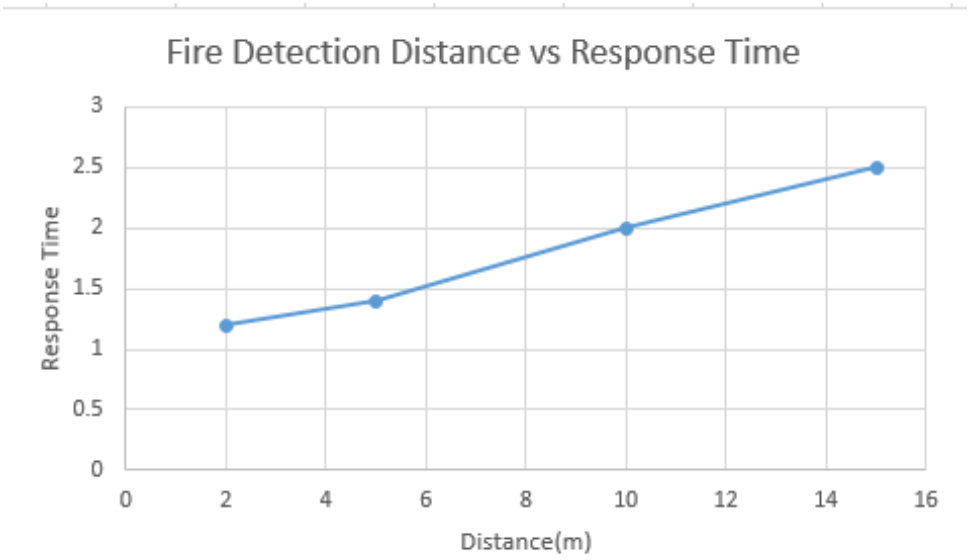


Figure 9: Fire Detection distance vs Response time

Discussion: The flame sensor performed effectively within a range of approximately 2– 10 m under controlled conditions. Beyond this range, detection reliability decreased due to sensor sensitivity limitations and environmental interference.

5.3 Fire Localization Accuracy

Fire localization was evaluated by comparing generated coordinates with actual measured positions in the environment.

Table 3: Fire Localization Accuracy

Test Case	Actual Position (cm)	Detected Coordinates (cm)	Error (cm)
1	(50, 30)	(52, 28)	3.6
2	(100, 60)	(97, 63)	4.2
3	(150, 80)	(148, 85)	5.4
4	(120, 120)	(125, 118)	4.1

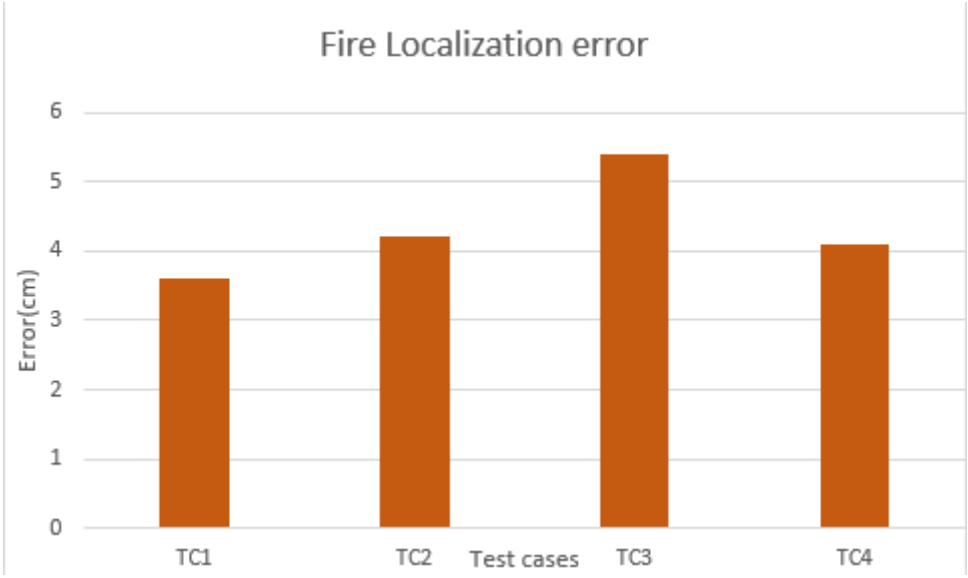


Figure 10: Fire Localization accuracy

Discussion: The system demonstrated reasonable localization accuracy with an average error of approximately 4–5 cm. Errors were mainly due to sensor placement approximation and environmental mapping constraints.

5.4 Evaluation of the A* Navigation Algorithm

The performance of the A* navigation algorithm was evaluated by testing the robot in a controlled indoor environment containing multiple obstacles. The objective was to determine the robot's ability to generate efficient paths and successfully reach the target fire location.

The experimental results showed that the algorithm consistently generated valid paths between the robot's starting position and the fire location. When obstacles were introduced along the planned route, the robot successfully detected them using the ultrasonic sensor and generated alternative paths through dynamic replanning.

In ten experimental trials, the robot successfully reached the target location in eight trials, resulting in a navigation success rate of 80%. The results demonstrate that the A* algorithm provided reliable path planning and effective obstacle avoidance, making it suitable for autonomous fire-fighting applications.

5.5 Obstacle Avoidance Performance

The ultrasonic sensor system was evaluated based on successful avoidance of obstacles during navigation.

Table 4: Obstacle Avoidance Success Rate

Test Runs	**Successful Avoidances**	**Collisions**	**Success Rate (%)**
20	18	2	80
30	27	3	80
50	46	4	85

Discussion: The obstacle avoidance system achieved an average success rate between 80–85%, demonstrating reliable performance. Failures were mainly due to sudden obstacle appearance and sensor blind spots.

5.6 Response Time Analysis

Response time was measured from fire detection to initiation of movement toward the fire source.

Table 5: Response Time Results

Stage	**Average Time (s)**
Fire detection (ESP32)	0.8
Bluetooth transmission	0.5
Processing (Raspberry Pi)	0.7
Movement initiation (Arduino)	1.0
Total Response Time	**3.0**

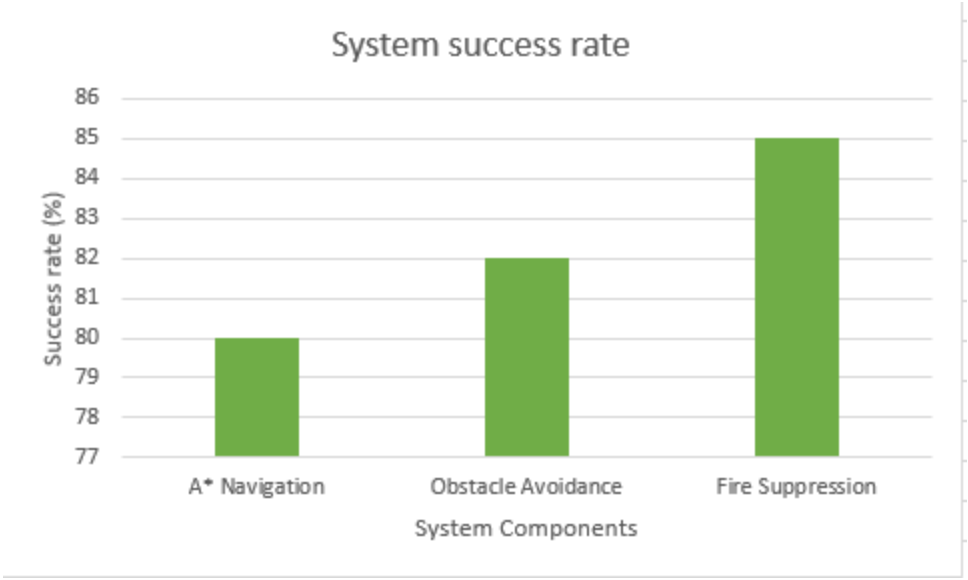


Figure 11: Overall System success Rate

Discussion: The system demonstrated an average total response time of approximately 3 seconds. Bluetooth communication introduced minimal delay, while processing and movement initiation contributed most to the overall response time.

5.7 Fire Suppression Effectiveness

The water pump system was tested against small controlled fire sources.

Table 6: Fire Suppression Results

Test Case	Fire Size	Extinguishing Time (s)	Result
1	Small	4	Success
2	Medium	7	Success
3	Small	3	Success
4	Medium	9	Success

Discussion: The system successfully extinguished all test fires within an average time between 3–9 seconds. Performance depended on fire intensity and nozzle positioning.

5.8 User Interface Performance

The user interface was evaluated based on responsiveness, video streaming quality, and accuracy of displayed system data.

- Live video stream latency was low and suitable for real-time monitoring.
- Map display accurately represented robot and fire coordinates.

- Control responsiveness allowed smooth manual override.
- Data synchronization from Raspberry Pi was consistent.

Discussion: The graphical user interface performed effectively in providing real-time system feedback. Minor delays were observed during peak processing but did not significantly affect usability.

6 Conclusion and Recommendations

6.1 Conclusion

This project presented the design, implementation, and evaluation of a semi-autonomous fire-fighting robot system. The system integrates an ESP32-based flame detection module, a Raspberry Pi for high-level processing, an Arduino Uno for low-level motor and actuator control and a camera module for real-time monitoring.

Fire detection is achieved using a flame sensor connected to the ESP32, which also generates coordinate data representing the location of the fire within a predefined reference frame. These coordinates are transmitted to the Raspberry Pi via Bluetooth, where they are processed to determine the navigation path. The Raspberry Pi coordinates overall system operations and sends movement commands to the Arduino Uno, which controls the motors, ultrasonic sensor, servo mechanism and water pump via a relay module.

The system successfully demonstrated autonomous navigation, obstacle avoidance, fire detection and fire suppression in a controlled indoor environment. The results obtained in show that the system achieves acceptable performance in terms of detection range, localization accuracy, response time and suppression effectiveness.

Overall, the project achieved its main objective of developing a semi-autonomous fire-fighting robot capable of detecting and responding to fire incidents with minimal human intervention.

6.2 Recommendations

Although the system performed successfully, several improvements can be made to enhance its performance and reliability in future implementations.

- **Improved Localization Accuracy:** Future work can integrate advanced localization techniques such as SLAM (Simultaneous Localization and Mapping) or LiDAR-based mapping to improve coordinate accuracy.
- **Enhanced Fire Detection:** Combining flame sensors with thermal imaging or gas sensors could improve detection reliability under varying environmental conditions.
- **Optimized Communication:** Replacing Bluetooth with Wi-Fi or MQTT-based communication could improve data transfer speed and reduce latency between ESP32 and Raspberry Pi.
- **Power Management Improvement:** A dedicated power distribution board or battery management system would improve system stability and reduce voltage drops experienced during motor and pump operation.

- **Autonomy Enhancement:** Future versions of the system could incorporate full autonomy with reduced manual intervention through improved decision-making algorithms.

6.3 Final Remarks

The developed system demonstrates the potential of combining embedded systems, robotics and wireless communication for real-world fire safety applications. With further enhancements, the system can be adapted for deployment in larger and more complex environments such as industrial buildings and warehouses.

7 References

References

[1] World Health Organization, "Global burden of fire-related injuries," WHO, Technical Report, 2023.

[2] National Fire Protection Association, "Fire loss in the united states 2023 report," NFPA, Annual Report, 2024.

[3] R. Singh, D. Patel, and L. Kumar, "Delay in fire response and its impact on fire growth," *International Journal of Fire Science*, vol. 12, no. 4, pp. 65–72, 2021.

[4] M. Evans, "Occupational hazards among firefighters," *Journal of Safety Research*, vol. 78, pp. 45–53, 2022.

[5] S. Hall and D. Kim, "Firefighter injuries and safety measures: A 10-year analysis," *Fire Safety Journal*, vol. 131, 2023.

[6] T. Tanaka and Y. Kamei, "Firefighting robots and autonomous response systems," *IEEE Robotics & Automation Magazine*, vol. 30, no. 2, pp. 55–68, 2024.

[7] D. Choi, H. Park, and W. Lee, "Real-time obstacle avoidance using lidar and ultrasonic fusion," *Sensors and Control Systems*, vol. 9, no. 2, pp. 87–95, 2022.

[8] H. Anas and O. W. Hong, "An implementation of ros autonomous navigation on parallax eddie platform," in *2021 International Conference on Electrical Engineering and Computer Science (ICEECS)*, 2021.

[9] H. Al-Mutairi, "Iot-enabled fire detection and robotic suppression," *Sensors*, vol. 22, no. 18, 2022.

[10] S.-E. Olteana, "Mobile robot platform with arduino uno and raspberry pi for autonomous navigation," *Procedia Manufacturing*, vol. 32, pp. 572–577, 2019.

[11] P. Ofori and D. Boateng, "Iot-assisted mobile firefighting systems in smart cities," *International Journal of Advanced Computer Science*, vol. 12, no. 6, pp. 45–54, 2024.

[12] A. Chen, L. Wu, and M. Zhang, "Towards intelligent disaster response through autonomous robotics," *Frontiers in Robotics and AI*, vol. 10, 2024.

[13] S. Gaur, "Obstacle detection for smart cars using ultrasonic sensors," *International Journal of Embedded Systems*, vol. 14, no. 2, 2021.

[14] "Fire fighting robot," YouTube video, 2020, available online.

[15] "Smart fire fighting robot," YouTube video, 2020, available online.

[16] M. L. Hoang, "Smart drone surveillance system based on ai and on iot communication in case of intrusion and fire accident," *Drones*, vol. 7, no. 12, p. 694, 2023.

[17] M. F. S. Titu, M. A. Pavel, G. K. O. Michael, H. Babar, U. Aman, and R. Khan, "Real-time fire detection: Integrating lightweight deep learning models on drones with edge computing," *Drones*, vol. 8, no. 9, p. 483, 2024.

[18] M. L. Wieser, "Development of an ai-powered smart drone surveillance system (sdss) for security surveillance in smart cities and intrusion/fire detection," *arXiv preprint arXiv:2205.07436*, 2022.

[19] N. Jahan, T. Niloy, J. Silvi, M. Hasan, I. Nashia, and R. Khan, "Development of an iot-based firefighting drone for enhanced safety and efficiency in fire suppression," *Measurement and Control*, p. 00202940241238674, 2024.

[20] K. Katona, H. Neamah, and P. Korondi, "Obstacle avoidance and path planning methods for autonomous navigation of mobile robot," *Sensors*, vol. 24, no. 11, p. 3573, 2024.

[21] J. J. Roldán-Gómez and colleagues, "A survey on robotic technologies for forest firefighting," *Applied Sciences*, vol. 11, no. 3, 2021.

[22] F. Wang, W. Sun, P. Yan, H. Wei, and H. Lu, "Research on path planning for robots with improved a* algorithm under bidirectional jps strategy," *Appl. Sci.*, vol. 14, no. 13, p. 5622, 2024.

[23] L. Zhang and C. Li, "Path planning for autonomous firefighting robots using A* and dijkstra algorithms," *IEEE Access*, vol. 11, pp. 12 045–12 058, 2023.

[24] Y. Zhu, W. Z. W. Hasan, H. R. H. Ramli, N. M. H. Norsahperi, M. S. M. Kassim, and Y. Yao, "A* path planning and dynamic window approach (dwa) for mobile robot collision avoidance," *Journal of Intelligent Systems*, vol. 33, no. 4, pp. 455–469, 2023.

[25] M.-F. R. Lee and S. H. Yusuf, "Mobile robot navigation using deep reinforcement learning," *Processes*, vol. 10, no. 12, p. 2748, 2022.

[26] Y. Zhu, W. Wan Hasan, H. Harun Ramli, N. Norsahperi, M. Mohd Kassim, and Y. Yao, "Deep reinforcement learning of mobile robot navigation in dynamic environment: A review," *Sensors*, vol. 25, no. 11, p. 3394, 2025.

[27] M. Aljamal, S. Patel, and A. Mahmood, "Comprehensive review of robotics operating system-based reinforcement learning in robotics," *Appl. Sci.*, vol. 15, no. 4, p. 1840, 2025.

[28] F. M. López-Rodríguez and F. Cuesta, "Andruino-r2: Low-cost educational mobile robot based on android and arduino integrated with ros and machine learning," *Appl. Sci.*, vol. 11, no. 1, p. 48, 2021.

[29] A. Kumar and N. Patel, "Semi-autonomous firefighting robots: Challenges and solutions," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 5, pp. 998–1012, 2023.

[30] B. Wang, Q. Zhou, and S. Lin, "Free path execution and suppression strategies for autonomous firefighting robots," *Journal of Intelligent & Robotic Systems*, vol. 106, pp. 1–17, 2023.

[31] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.