[1]

AUTONOMOUS FIREFIGHTING ROBOT USING ARDUINO UNO R3

Edgar Jr B. Villas, Mohammad Naim D. Mariga
Mindanao State University - Iligan Institute of Technology

***Abstract*— Fire hazards in confined spaces pose significant risks, necessitating innovative solutions for efficient detection and suppression. This study presents the design and development of an Autonomous Firefighting Robot (AFR) utilizing an Arduino UNO R3 microcontroller. The robot integrates flame, ultrasonic, and IR sensors for real-time fire detection, obstacle avoidance, and navigation. Central to its operation is an interrupt-driven architecture, enabling the AFR to prioritize critical tasks such as fire detection and obstacle avoidance by immediately responding to high-priority events. Testing revealed that the AFR effectively avoids obstacles and reliably activates its water pump to extinguish fires. However, fire detection is limited to a narrow frontal range due to the binary output of the flame sensor, highlighting a need for improved sensing technology. Despite this limitation, the AFR demonstrated its potential as a low-cost and autonomous solution for fire safety, contributing to the advancement of robotics in emergency response systems.**

***Index Terms*— Arduino UNO R3, Confined Spaces, Interrupt-Driven Architecture, Fire hazards**

I. Introduction

The increasing occurrence of fire hazards necessitates innovative solutions to enhance safety measures in various environments. Traditional fire detection and suppression systems often rely on manual intervention, which can be slow and less effective, especially in small and confined areas. To address these challenges, the current study explores the practical aspects of developing and deploying an Autonomous Firefighting Robot (AFR) that leverages modern technology for efficient fire management.

The proposed AFR uses advanced sensors and microcontrollers to navigate autonomously and detect fires in real-time, utilizing flame detection and water spraying mechanisms. The robot aims to identify and suppress fires promptly to minimize potential damage and enhance safety. Additionally, the robot employs ultrasonic sensors to detect obstacles to ensure smooth operation and navigation within its environment.

Central to the robot's functionality is its interrupt-driven architecture, which enables the system to respond rapidly to critical events, such as fire detection or obstacle avoidance, without being delayed by ongoing routine tasks. Interrupt-driven systems are designed to handle high-priority signals by temporarily suspending non-urgent processes, ensuring that the AFR can prioritize immediate threats and maintain operational efficiency. For instance, upon detecting a fire or encountering an obstacle, the interrupt mechanism triggers a rapid response, activating the necessary subsystems—such as the water pump or motor adjustment—while seamlessly managing other functions. This architecture not only optimizes resource utilization but also enhances the robot's ability to operate in dynamic and hazardous environments with minimal latency.

This study delves into the concept and implementation of the Autonomous Firefighting Robot to address the inefficiencies of traditional fire management systems. By integrating sensor technology, autonomous navigation, and interrupt-driven processing, the AFR promises to offer a dynamic, responsive, and user-centric approach to fire safety. This innovative approach significantly contributes to the development of a smart and secure environment, paving the way for safer and more efficient fire hazard management solutions.

II. Literature Review

Autonomous firefighting systems are increasingly being researched as a more efficient way to manage fire hazards without human intervention. Traditional fire detection and suppression methods often suffer from delays and inefficiencies that make them less effective in preventing damage and ensuring safety. Recent advancements in robotics and sensor technology have paved the way for the development of autonomous systems that can detect fires, navigate complex environments, and suppress flames promptly. Studies have demonstrated the potential of these systems to enhance safety and operational efficiency that highlights the importance of real-time data processing and interrupt-driven responses in achieving optimal performance.

The study of Adilshah et al. (2022) presents an Arduino-based Autonomous Fire Extinguishing Robot that detects and extinguishes fires using three flame sensors and a water pump mechanism. Controlled by an Arduino Uno microcontroller, the robot autonomously navigates towards the fire and activates the extinguisher. It effectively reduces the need for human intervention in firefighting, mitigating the risks associated with fire hazards. The results demonstrate the robot’s efficiency in promptly identifying and suppressing fires, highlighting the potential for automation in enhancing fire safety and response [1].

A study by Aliff et al. (2019) presents QRob, a compact firefighting robot designed to enhance safety during fire extinguishment in hazardous and confined environments. Equipped with ultrasonic and flame sensors, QRob can autonomously detect fire locations and avoid

[1]Manuscript received December 15, 2024. This work was supported in part by the Department of Computer Engineering, College of Engineering at MSU-Iligan Institute of Technology.

E. J. B. Villas is with the Department of Computer Engineering, MSU-Iligan Institute of Technology, Iligan City, Philippines (e-mail: edgar.villas@g.msuiit.edu.ph).

M. N. D. Mariga is with the Department of Computer Engineering, MSU-Iligan Institute of Technology, Iligan City, Philippines (e-mail: mohammadnaim.mariga@g.msuiit.edu.ph).

obstacles, maintaining a safe distance of 40 cm from flames. The robot allows operators to monitor and control firefighting operations remotely via a smartphone-connected camera. Overall, QRob demonstrates significant potential in improving firefighter safety and efficiency in challenging situations, successfully achieving its design objectives [2].

Dhiman et al. (2022) developed an autonomous firefighting robot using deep learning and machine vision to enhance fire detection and suppression. The robot employs convolutional neural networks (CNNs) for accurate fire identification and is equipped with sensors for navigation and obstacle avoidance. This integration ensures efficient and timely responses to fire hazards, reducing risks to human firefighters. The study demonstrated the robot's effectiveness in controlled environments, highlighting the potential of advanced robotics and AI in improving fire safety and emergency response systems [3].

Oyelami et al. (2024) introduced a low-cost multi-sensing firefighting robot featuring a dual-unit system comprising a static sensing station and a mobile robot. The mobile unit, controlled by an Arduino Mega microcontroller, utilized flame sensors with a 1-meter detection range, an ultrasonic sensor for obstacle avoidance, and a water pump for extinguishing fires. Its static unit was equipped with flame sensors and a transmitter to relay fire location data to the robot. This design aimed to enhance autonomy and extend the sensing range compared to earlier models that relied heavily on manual or remote control systems [4].

The study by Baynes et al. (2003) explores the performance and energy consumption of embedded real-time operating systems (RTOS) using a simulation environment called SimBed. This environment allows the testing of RTOSs on a simulated microcontroller with real-time stimuli. A key focus of the study is the critical role of interrupts in managing real-time task execution. The paper demonstrates how interrupt-driven systems prioritize high-priority tasks immediately, ensuring timely response to critical events such as external stimuli or system-level interrupts. Comparisons among preemptive multitasking RTOSs, cooperative multitasking RTOSs, and minimal task schedulers revealed that efficient interrupt handling directly influences task timing accuracy, system reliability, and energy consumption. Preemptive systems like μC/OS-II outperformed simpler schedulers under high system load due to their ability to quickly handle interrupts and context switches. These findings validate the effectiveness of an interrupt-driven architecture, making it a cornerstone for achieving real-time responsiveness and efficient resource management in autonomous systems such as the AFR [5].

III. Methodology

Research Approach

After reviewing existing studies and analyzing the needs for autonomous firefighting in small areas, the researchers developed a structured methodology to guide the design and implementation of the Autonomous Firefighting Robot. This approach outlines the step-by-step processes, components, and algorithms that ensure the robot's effective operation in detecting and extinguishing fires autonomously.

System Design

The Autonomous Firefighting Robot employs a combination of hardware and software components to achieve its firefighting and navigation functionalities. Below is a breakdown of the key system elements:

Hardware Components

1. Power Supply: The robot is powered by a rechargeable 9V battery pack, supplying the necessary voltage to operate the motors, sensors, and fire-extinguishing components.
2. Sensors:
    a. Flame Sensor: Detects the presence of a fire by sensing infrared radiation. This data is transmitted to the Arduino UNO microcontroller for processing.
    b. Ultrasonic and IR Sensor: Detects obstacles to help the robot navigate its environment safely and avoid collisions while searching for fire hazards.
3. Microcontroller: An Arduino UNO microcontroller serves as the central processing unit, receiving and interpreting sensor data, executing interrupt-driven commands, and controlling the robot's motors and extinguishing mechanism.
4. Motor Driver and Chassis: The robot uses a motor driver to regulate movement and a chassis to house all components, designed for balance and ease of navigation in small, enclosed spaces.
5. Water Pump and Fire-Extinguishing Mechanism: Upon detecting fire, the robot activates the water pump, which sprays water to extinguish the fire.

Software and Control System

1. Interrupt-Driven Architecture: The system employs an interrupt-driven design to ensure efficient response times. Critical events—such as fire detection and obstacle detection that prioritize immediate action over routine operations.
2. Algorithm:
    a. Initialization: The robot initializes by powering up its sensors and components, preparing for autonomous operation.
    b. Fire Detection and Extinguishing:
        i. The flame sensor continuously scans for fire while it's in normal operation. Upon detection on its front side, an interrupt is triggered, activating the water pump and directing the robot to

approach the fire for extinguishing.

c. Obstacle Avoidance:
   i. The ultrasonic sensor detects obstacles in the robot's path. When an obstacle is detected, an interrupt is triggered, causing the robot to adjust its direction to avoid collision.

d. Return to Normal Operation: After handling each interrupt, the robot returns to its normal patrol and monitoring state, continuing to scan for fires and obstacles.

Block Diagram of the System

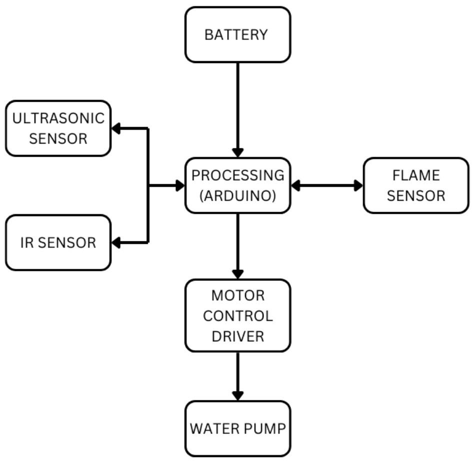

Figure 1: Block Diagram

The block diagram below outlines the system's core functions and interactions between components:

1. Battery: Distributes power to all components.
2. Flame Sensor: Detects fires and sends data to the microcontroller.
3. Ultrasonic and IR Sensor: Detects obstacles and sends data to the microcontroller.
4. Arduino Microcontroller: Executes the overall functionality of the system.
5. Motor Driver: Drives the robot's motors based on the microcontroller's commands.
6. Water Pump: Activates upon fire detection to extinguish flames.

IV. Results and Discussion

The project aims to develop an Autonomous Firefighting Robot (AFR) capable of detecting and extinguishing fires in confined spaces. It uses a combination of sensors and an interrupt-driven algorithm to navigate autonomously, avoid obstacles, and respond to fire hazards effectively. The AFR is equipped with flame sensors for fire detection, ultrasonic and IR sensors for obstacle avoidance, and a water pump for fire suppression. All components, including the Arduino UNO microcontroller, are integrated to enable seamless operation as shown in Figure 2.

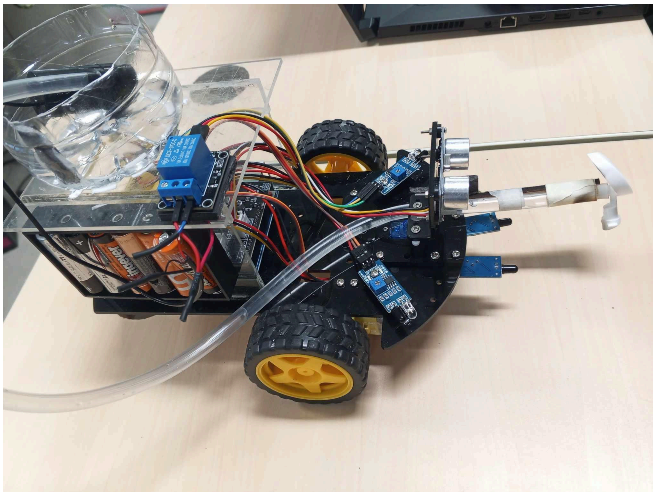

Figure 2: Prototype Implementation

The testing and evaluation of the Autonomous Firefighting Robot (AFR) focused on assessing its performance in obstacle avoidance, fire detection, and fire suppression. Below are the findings based on the conducted tests:

1. Obstacle avoidance

The robot demonstrated a high level of efficiency in navigating its environment and avoiding obstacles. Using ultrasonic and IR sensors, it effectively detected obstacles within its path and adjusted its direction accordingly to ensure smooth movement. This functionality proved reliable across various test scenarios, highlighting the effectiveness of the interrupt-driven architecture for real-time decision-making. The key parameters of the robot's obstacle detection and navigation performance are summarized in Table 1.

Table 1: Obstacle Detection and Navigation Performance

Parameters	**Details**
Sensors Used	Ultrasonic and IR sensors
Obstacle Detection Distance	Maintained a safe distance of 12 cm
Success Rate	95% success rate in avoiding static obstacle
Power Levels	- Forward Motion: 30%
	- Turning: 40%
Navigation System Reliability	Smooth movement and reliable performance across test scenarios

2. Fire Detection

The fire detection mechanism showed significant limitations in its functionality. The robot was only able to detect fires directly in front of it due to the flame sensor's binary output (1 or 0). The complexity of the sensor's readings made it difficult to reliably identify fires from varying angles, impacting the robot's autonomous operation and requiring precise placement of the fire for successful detection. The key performance metrics of the fire detection system are summarized in Table 2.

Table 2: Fire Detection System Performance

Parameters	Details
Optimal Detection Range	10–20 cm
True Positive Rate	92%
Response Time	1 - 2 secs
Detection Range	30–70 cm (direct line of sight)
Threshold Value	20 ADC units

Within the optimal range of 10–20 cm, the fire detection system achieved a 92% true positive rate, with response times averaging between 1-2 seconds. Although the flame sensors operated reliably within a 30–70 cm range when in direct line of sight, these limitations in angular coverage highlight the need for enhanced sensor configurations to improve fire detection capabilities in dynamic environments.

3. Fire Suppression

Despite the challenges in fire detection, the robot successfully activated the water pump and extinguished fires upon detecting them. This indicates the reliability of the fire suppression system, which performed effectively during testing. The integration of the water pump and its response to flame sensor triggers demonstrates the potential of the AFR in addressing fire hazards in confined spaces. The key performance metrics of the fire suppression system are summarized in Table 3.

Table 3: Fire Suppression System Performance

Parameters	Details
Average Response Time	2.5 seconds
Optimal Fighting Distance	15-20 mm
Water Usage per Engagement	30 ml
System Reliability	Successfully extinguished fires upon detection

Once fires were detected, the suppression system took an average of 2.5 seconds to extinguish fires at the optimal fighting distance of 15–20 mm, using approximately 30 ml of water per engagement. These results highlight the system's effectiveness in mitigating fire hazards and its suitability for confined spaces where precision and reliability are critical.

Table 4: Summary of the Findings

Functionality	Performance	Remarks
Obstacle Avoidance	Highly Effective	The robot efficiently detected and avoided obstacles using ultrasonic and IR sensors.
Fire Detection	Limited	The flame sensor could only detect fires directly in front due to its binary output (1 or 0).
Fire Suppression	Successful	The robot reliably activated the water pump and extinguished fires once detected.

These results indicate that while the AFR successfully demonstrates core functionalities in fire detection and suppression, there are opportunities for improvement in the fire detection system. The robot's reliable obstacle avoidance and effective fire suppression capabilities show promise for practical applications in confined spaces, despite the current limitations in fire detection range and accuracy.

V. Conclusion

The development and testing of the Autonomous Firefighting Robot (AFR) demonstrated its potential as an innovative solution for fire safety in confined spaces. The robot showcased exceptional performance in obstacle avoidance, effectively navigating its environment and avoiding collisions. This highlights the robustness of its interrupt-driven architecture and the reliability of the ultrasonic and IR sensors in ensuring smooth and autonomous navigation.

The fire suppression mechanism also proved to be highly effective, with the robot successfully activating its water pump and extinguishing fires upon detection. This indicates that the integration of the water pump with the flame sensor is functional and responsive under controlled conditions.

However, the AFR exhibited notable limitations in fire detection. The flame sensor's binary output (1 or 0) restricted its ability to detect fires beyond a narrow frontal range, requiring precise fire placement for successful detection. This limitation impacts the robot's overall autonomy and highlights the need for more advanced sensing technologies, such as multi-directional or higher-resolution flame sensors in future iterations.

The power management system demonstrated reliable performance, with the robot maintaining operational capability for approximately 2 hours during patrol mode. This duration provides adequate time for fire detection and suppression tasks in practical applications, though future versions could benefit from extended battery life for longer deployment periods.

Testing revealed that the motor control system, with its calibrated speed adjustments, provided stable movement and reliable navigation. The implementation of different speed settings for various operations (forward, backward, and turning) proved effective in maintaining control during both routine patrol and fire-fighting scenarios.

For future development, several improvements could enhance the robot's capabilities:

1. Implementation of analog flame sensors for better fire direction detection.
2. Development of an adjustable water spray system for more efficient fire suppression.
3. Integration of advanced navigation algorithms for complex environments.
4. Addition of wireless communication capabilities for remote monitoring.
5. Enhancement of the power system for extended operation time.

These findings suggest that while the current AFR prototype successfully demonstrates the viability of autonomous fire-fighting systems, there are significant potentials for further optimization and enhancement. The project establishes a foundation for future research in autonomous fire safety systems, particularly for applications in confined spaces where human intervention may be challenging or dangerous.

APPENDIX

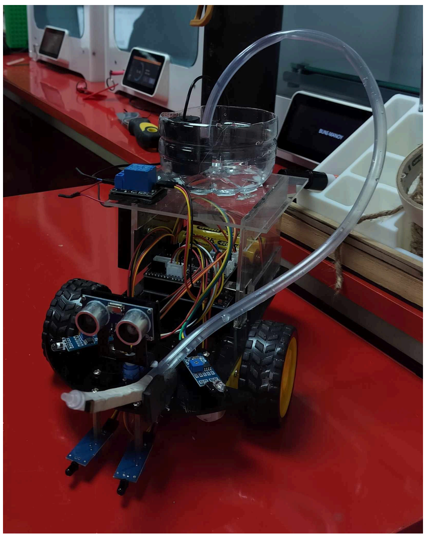

Figure 3: (Prototype Picture)

Figure 4: (LAFVIN Smart Robot Car 2WD Chassis Kit Upgraded V2.0 for Arduino Robot STEM /Graphical Programming Robot Car) - this is the original robot with the added sensors for the firefighting mechanism.

```
#include "motor.h"
#include "ultrasonic.h"
#include "pump.h"

#define APPROACH_SPEED 0.3
#define SAFE_DISTANCE 12

int8_t getFireDirection(void) {
    uint16_t min_value = 1023;
    int8_t direction = -1;

    if(sensor_values[FIRE_SENSOR_RIGHT] < FIRE_THRESHOLD &&
        sensor_values[FIRE_SENSOR_RIGHT] < sensor_values[FIRE_SENSOR_LEFT]) {
        direction = FIRE_SENSOR_RIGHT;
    }
    else if(sensor_values[FIRE_SENSOR_LEFT] < FIRE_THRESHOLD &&
            sensor_values[FIRE_SENSOR_LEFT] < sensor_values[FIRE_SENSOR_RIGHT]) {
        direction = FIRE_SENSOR_LEFT;
    }

    return direction;
}

void adjustRobotDirection(int8_t fireDirection) {
    stopMotors();

    switch(fireDirection) {
        case FIRE_SENSOR_RIGHT:
            // Small adjustment for right sensor
            setMotorSpeed(MOTOR_LEFT, 0.2, FORWARD);
            setMotorSpeed(MOTOR_RIGHT, 0.1, BACKWARD);
            _delay_ms(100);  // Short adjustment time
            break;

        case FIRE_SENSOR_LEFT:
            // Small adjustment for left sensor
            setMotorSpeed(MOTOR_LEFT, 0.1, BACKWARD);
            setMotorSpeed(MOTOR_RIGHT, 0.2, FORWARD);
            _delay_ms(100);  // Short adjustment time
            break;
    }
    stopMotors();
}

void handleFireDetection(void) {
    stopMotors();
    setPumpState(PUMP_ON);

    int8_t fireDirection = getFireDirection();
    if(fireDirection == -1) {
        setPumpState(PUMP_OFF);
        return;
    }

    adjustRobotDirection(fireDirection);

    // Stay in position and fight fire
    uint8_t fireOutCount = 0;
    while(fireOutCount < 100) {  // Check for 1 second of no fire
        if(getFireDirection() == -1) {
            fireOutCount++;
        } else {
            fireOutCount = 0;
            // Small readjustment if fire moves
            int8_t newDirection = getFireDirection();
            if(newDirection != fireDirection && newDirection != -1) {
                adjustRobotDirection(newDirection);
                fireDirection = newDirection;
            }
        }
        _delay_ms(10);
    }

    setPumpState(PUMP_OFF);
}

void checkAndAvoidObstacle(void) {
    uint16_t front_distance = measureDistance();
    uint8_t left_blocked = sensor_values[IR_SENSOR_LEFT] < IR_THRESHOLD;
    uint8_t right_blocked = sensor_values[IR_SENSOR_RIGHT] < IR_THRESHOLD;

    if(front_distance < SAFE_DISTANCE) {
        stopMotors();
        moveBackward();
        _delay_ms(500);

        moveServo(SERVO_LEFT);
```

```c
        _delay_ms(200);
        uint16_t leftDistance = measureDistance();

        moveServo(SERVO_RIGHT);
        _delay_ms(200);
        uint16_t rightDistance = measureDistance();

        moveServo(SERVO_CENTER);

        if(left_blocked && right_blocked) {
            turnLeft();
            turnLeft();
        }
        else if(left_blocked || leftDistance < SAFE_DISTANCE) {
            turnRight();
        }
        else if(right_blocked || rightDistance < SAFE_DISTANCE) {
            turnLeft();
        }
        else if(leftDistance > rightDistance) {
            turnLeft();
        }
        else {
            turnRight();
        }
    }

    else if(left_blocked || right_blocked) {
        if(left_blocked && right_blocked) {
            moveBackward();
            _delay_ms(300);
            turnLeft();
            turnLeft();
        }
        else if(left_blocked) {
            setMotorSpeed(MOTOR_LEFT, 0.3, FORWARD);
            setMotorSpeed(MOTOR_RIGHT, 0.1, FORWARD);
        }
        else if(right_blocked) {
            setMotorSpeed(MOTOR_LEFT, 0.1, FORWARD);
            setMotorSpeed(MOTOR_RIGHT, 0.3, FORWARD);
        }
    }
    else {
        moveForward();
    }
}

int main(void) {
    cli();

    setupMotors();
    setupTimer1();
    setupServo();
    setupUltrasonic();
    setupSensors();
    setupPump();

    sei();

    setPumpState(PUMP_OFF);
    moveServo(SERVO_CENTER);
    _delay_ms(1000);

    while(1) {
        if(fire_detected) {
            handleFireDetection();
            fire_detected = 0;
            obstacle_detected = 0;
        } else {
            checkAndAvoidObstacle();
        }

        _delay_ms(10);
    }

    return 0;
}
```

Code Snippet 1 - main.c code for the implementation of the autonomous fire-fighting robot

```c
ISR(ADC_vect) {
    sensor_values[adc_channel] = ADC;

    // Check for fire (only on fire sensors)
    if((adc_channel == FIRE_SENSOR_RIGHT || adc_channel == FIRE_SENSOR_LEFT) &&
        ADC < FIRE_THRESHOLD) {
        fire_detected = 1;
    }

    // Check for obstacles (only on IR sensors)
    if((adc_channel == IR_SENSOR_RIGHT || adc_channel == IR_SENSOR_LEFT) &&
        ADC < IR_THRESHOLD) {
        obstacle_detected = 1;
    }

    // Move to next channel
    adc_channel = (adc_channel + 1) % 4;
    ADMUX = (ADMUX & 0xF0) | adc_channel;

    // Start next conversion
    ADCSRA |= (1 << ADSC);
}
```

Code Snippet 2 - ultrasonic.c code for the implementation of the interrupt for the autonomous fire-fighting robot

Acknowledgment

We would like to express our deepest gratitude to **Prof. Stephen Haim** for his invaluable guidance, encouragement, and support throughout the development of this project. His insights and expertise have been instrumental in shaping the design and implementation of the Autonomous Firefighting Robot. This work would not have been possible without his mentorship and dedication to fostering innovation and practical learning. Thank you for inspiring us to push the boundaries of our capabilities.

References

[1] P. M. R. Adilshah, S. R. Kukde, U. F. Shaikh, and J. B. Choudhari, “Autonomous Fire Fighting Robot Using Arduino”, IJRESM, vol. 5, no. 4, pp. 194–197, May 2022, Accessed: Oct. 28, 2024. [Online]. Available: https://journal.ijresm.com/index.php/ijresm/article/view/2002

[2] [2] Aliff, M., Samsiah, N., Yusof, M., & Zainal, A. (2019). Development of fire fighting robot (QROB). International Journal of Advanced Computer Science and Applications, 10(1). https://doi.org/10.14569/ijacsa.2019.0100118

[3] [3] Dhiman, A., Shah, N., Adhikari, P. et al. Firefighting robot with deep learning and machine vision. Neural Comput & Applic 34, 2831–2839 (2022). https://doi.org/10.1007/s00521-021-06537-y

[4] A. T. Oyelami, J. A. Oyadokun, O. A. Akintunlaji, and G. C. Ihenacho, “Low-cost multi-sensing fire-fighting robot with obstacle avoidance mechanism,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 13, no. 4, pp. 373–379, Dec. 2024. http://doi.org/10.11591/ijra.v13i4.pp373-379

[5] K. Baynes et al., "The Performance and Energy Consumption of Embedded Real-Time Operating Systems," *IEEE Transactions on Computers*, vol. 52, no. 11, pp. 1454–1467, Nov. 2003. [Online]. Available: https://doi.org/10.1109/TC.2003.1244943