Arduino Based Fire Fighting Robot

Shubham Sanjay Shinde
Pursuing Third year in B.E. CSE (AI&ML)
Smt. Indira Gandhi College of Engineering
Navi Mumbai, India
2022ca61f@sigce.edu.in

Neha Akhileshchandra Dubey
Pursuing Third year in B.E. CSE (AI&ML)
Smt. Indira Gandhi College of Engineering
Navi Mumbai, India
2022ca14f@sigce.edu.in

Arya Sandip More
Pursuing Third year in B.E. CSE (AI&ML)
Smt. Indira Gandhi College of Engineering
Navi Mumbai, India
2022ca34f@sigce.edu.in

Prof. Manisha Hatkar
Professor, Dept. of AI & ML
Smt. Indira Gandhi College of Engineering
Navi Mumbai, India
manisha.hatkar@sigce.edu.in

***Abstract*— This project showcases the creation of an Arduino-based fire-fighting robot that can detect and put out fires. The robot is fitted with flame sensors, a temperature sensor, and a gas sensor to effectively identify fire, changes in temperature, and smoke levels. When a fire is detected, the system triggers a water pump to extinguish it. Beyond its fire-fighting abilities, the robot includes real-time monitoring and alerting features through the ESP8266 Wi-Fi module and the Blynk IoT platform. This setup enables notifications about fire, temperature, and smoke to be sent to the user's mobile device, allowing for remote monitoring and control. Additionally, the system can detect obstacles using an infrared (IR) sensor, improving its ability to navigate autonomously in dangerous environments. The design of the robot aims to offer a cost-effective and efficient solution for fire detection and control in various settings, including industrial, residential, and commercial spaces.**

Keywords— Fire Fighting robot, flame sensor, temperature and smoke sensor, Arduino with ESP8266 module, flame sensing robot, Arduino robot.

I. INTRODUCTION

Fire hazards present serious threats to lives, property, and the environment. In places like homes, factories, and businesses, quick detection and prompt action during fire emergencies are essential to reduce damage. Traditional fire detection and suppression systems tend to be fixed and depend on human intervention, which can slow down response times and heighten the risk of loss. As the need for smart solutions increases, robotics and IoT (Internet of Things) technologies provide exciting advancements in autonomous fire detection and firefighting.

This project aims to create an Arduino-based fire-fighting robot that can autonomously identify and extinguish fires. The robot is equipped with various sensors, such as flame sensors, a temperature sensor, and a gas sensor, which work together to effectively detect fire, temperature fluctuations, and smoke—essential indicators of fire risks. When a fire is detected, the robot activates a water pump to put it out. Additionally, an ESP8266 Wi-Fi module connects the robot to the Blynk IoT platform, allowing for real-time alerts and monitoring of fire, temperature, and smoke levels through a mobile device.

The goal of this project is to develop an affordable, self-sufficient system that improves fire safety through early detection, rapid response, and remote monitoring. This system is especially beneficial in scenarios where human presence is restricted or poses a risk, such as in industrial settings or dangerous areas.

The main objectives are:

- To design and implement an Arduino-based robot capable of detecting and extinguishing fires.
- To integrate a real-time notification system using the ESP8266 Wi-Fi module and Blynk IoT platform for remote monitoring.
- To enhance the robot's navigational capabilities using an infrared (IR) sensor to avoid obstacles.

II. METHODOLOGY

This section describes the hardware components, sensor integration, and software design involved in creating the Arduino-based fire-fighting robot. The main functions of the robot encompass fire detection, extinguishing, real-time monitoring, and obstacle avoidance. These capabilities are accomplished through a blend of sensors, actuators, and wireless communication modules.

1. Hardware Components

The following hardware components were used to build the robot:

- *Arduino UNO:* The microcontroller that serves as the central processing unit, managing inputs from sensors and controlling the actuators.

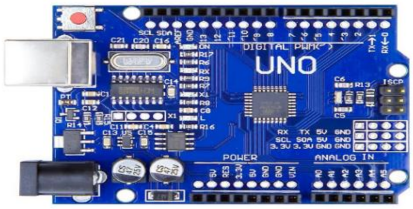

Figure 1: Arduino Uno

- *Flame Sensors*: Three flame sensors are used to detect the presence of fire. They are placed at the front, left, and right of the robot to ensure comprehensive coverage.

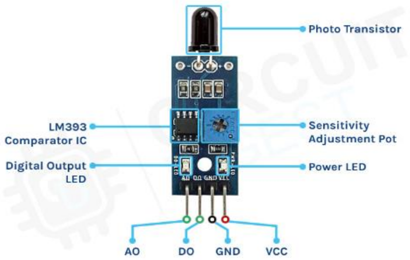

Figure 2: Flame Sensor

- *Temperature Sensor (LM35):* This sensor measures the ambient temperature to detect significant temperature rises that could indicate fire.

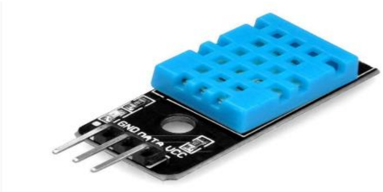

Figure 3: Temperature Sensor

- *Gas Sensor (MQ-2):* The gas sensor detects smoke and harmful gases like CO2, indicating fire-related combustion.

Figure 4: Smoke Sensor

- *Water Pump:* The pump is activated to spray water when a fire is detected. It is controlled via a TIP122 transistor for switching based on sensor data.

Figure 5: Water Supply Motor Pump

- *Infrared (IR) Sensor:* The IR sensor is used for obstacle detection, enabling the robot to avoid obstacles while navigating.

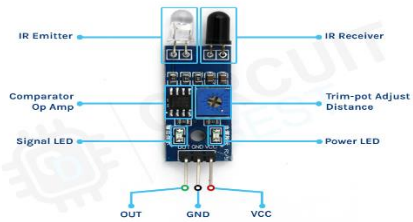

Figure 6: IR Sensor

- *Servo Motor:* A servo motor is integrated to rotate the water outlet pipe. The servo can rotate up to 90 degrees to direct the water spray towards the detected fire source, enhancing the effectiveness of the extinguishing process.

Figure 7: Servo Motor

- *ESP8266 Wi-Fi Module:* This module enables wireless communication, sending real-time fire, temperature, and smoke notifications to the user's mobile device through the Blynk IoT platform.

Figure 8: ESP8266 Wi-Fi Module

- *Motor Driver (L298N):* Controls the movement of the robot by managing the two DC motors responsible for forward, backward, and turning motions.

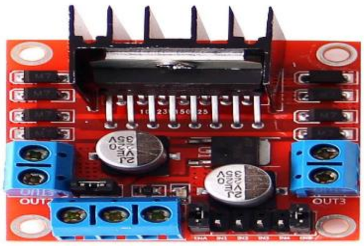

Figure 9: L298N Motor Driver

- *Power Supply:* A 12V battery powers the robot.

Figure 10: 2 x (3.5v) battery

2. System Design and Integration

The design of the robot integrates all the sensors and modules to work together to achieve real-time fire detection, reporting, and extinguishing. The system is divided into three key operations:

a) *Fire Detection and Extinguishing:*

Each flame sensor keeps a constant watch over its designated area for any signs of fire. When a sensor detects a flame, the robot promptly activates the water pump to put it out. The sensors communicate using digital signals (HIGH/LOW) to show whether a flame is present. The water pump is managed by a TIP122 transistor, which functions as a switch to control the water flow.
The temperature sensor (LM35) measures the surrounding temperature, and if it goes above a certain limit (like 50°C), it strengthens the decision to trigger the firefighting system. At the same time, the MQ-2 gas sensor checks the air for smoke or flammable gases. If it detects any, the gas sensor sends a signal to the Arduino, providing additional confirmation that a fire is occurring.

b) *Real-Time Monitoring and Notification:*

The ESP8266 Wi-Fi module connects to the Arduino, allowing for wireless communication. This module transmits data from the flame, temperature, and gas sensors to the Blynk IoT platform. The Blynk app is set up on a mobile device to receive notifications, showing real-time sensor readings like:

- Fire detected (based on flame sensors)
- Temperature readings (from LM35)
- Smoke detected (from MQ-2)

When the flame sensors detect fire, the user receives an immediate alert on their mobile device via the Blynk app, allowing for remote monitoring even when not in proximity to the robot.

c) *Navigation and Obstacle Avoidance*

The robot is built to move on its own. An IR sensor located at the front detects any obstacles in its way and sends a signal to the Arduino to either stop or change direction to prevent collisions. The motor driver (L298N) manages the two DC motors that drive the wheels, allowing for smooth movement in various directions.

3. Software Implementation

The Arduino is programmed using the Arduino IDE. The logic flow is divided into:

- *Sensor Monitoring*:
  Continuous polling of flame, temperature, and gas sensors.

- *Fire Detection Algorithm:*
  If any flame sensor detects fire (digital input goes HIGH), the system checks the temperature and gas sensor for additional validation.
  Once confirmed, the water pump is activated using the following control logic:

  if (Flame Detected == TRUE) AND (Temp > Threshold) OR (Smoke Detected == TRUE)
  ➔ activate water pump

- *Communication with Blynk:*
  The ESP8266 Wi-Fi module connects to the local Wi-Fi network and communicates with the Blynk cloud server. Data from sensors is sent to the Blynk app, allowing the user to receive real-time updates.

- *Obstacle Avoidance:*
  The IR sensor constantly monitors the robot's path. If an obstacle is detected within a certain distance, the following algorithm ensures safe navigation:

  if (Obstacle Detected == LOW)
  ➔ stop motors

4. *Flowchart*

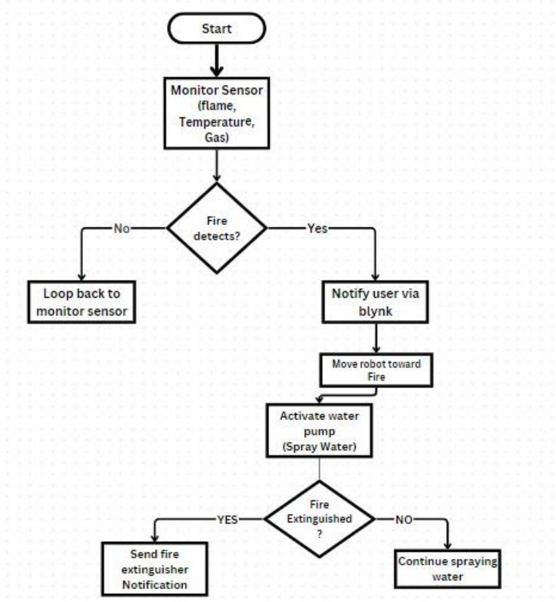

Figure 11: Flowchart

5. *Model*

The below image is the final hardware model of Arduino based Fire Fighting Robot.

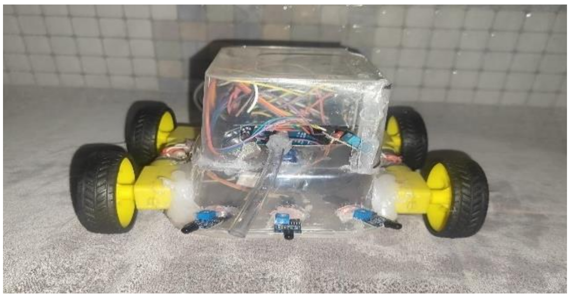

Figure 12: Robot Front View

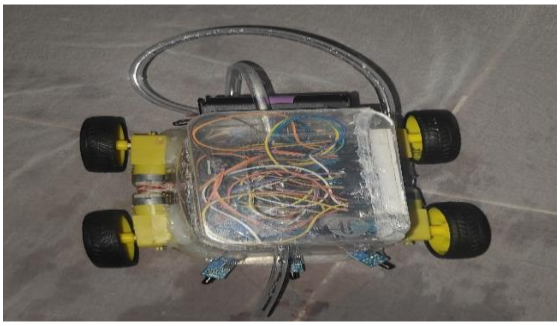

Figure 13: Robot Top View

6. *Connection Diagram*

The Connection of the system is illustrated in Figure, which shows the connections between the Arduino, sensors, actuators, and the ESP8266 Wi-Fi module (add a diagram if necessary)

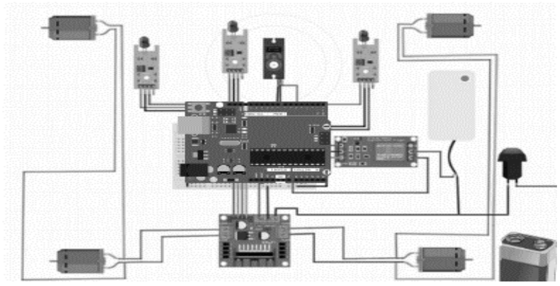

Figure 14: Circuit wiring of Arduino

III. PROGRAMMING

The Robot is trained using Arduino programming. Below is the code for functional part of the robot.

```
void loop() {
  int flame1 = analogRead(flameSensor1);
  int flame2 = analogRead(flameSensor2);
  int flame3 = analogRead(flameSensor3);
  int temp = analogRead(tempSensor);
  int gas = analogRead(gasSensor);
  int obstacleDetected = digitalRead(irSensor);

  String data = String(flame1) + " " + String(flame2) + " " +
  String(flame3) + " " + String(temp) + " " + String(gas) + " IR: "
  + String(obstacleDetected);
  Serial.println(data);

  bool nearFlame = (flame1 < 100 || flame2 < 100 || flame3 < 100);

  // Obstacle avoidance logic
  if (obstacleDetected == LOW) { // Obstacle detected
    stopMotors();  // Stop if obstacle detected
    wasMovingForward = false; // Reset flag
    wasTurning = false;
  } else if (nearFlame) {
     if (flame1 < 100 && flame2 >= 100 && flame3 >= 100) {
      turnLeft(); // Turn left if flame detected on the left
    } else if (flame1 >= 100 && flame2 >= 100 && flame3 < 100) {
      turnRight(); // Turn right if flame detected on the right
    } else if ((flame1 < 100 && flame3 < 100) || flame2 < 100) {
      moveForward(); // Move forward towards the flame
    }

  } else {
    if (wasMovingForward) {
      stopMotors();
      spreadWater();
      wasMovingForward = false;
    }
    else {stopMotors(); // Stop if no flame is detected
    wasMovingForward = false; // Reset the forward movement flag
    wasTurning = false;}
  }

  delay(100);
}
```

As mentioned in Methodology, Servo motor is used to rotate the water outlet pipe. Below is the code for its function:

```
void spreadWater() {
  digitalWrite(waterPumpPin, HIGH); // Turn on the water pump
  for (int angle = 60; angle <= 120; angle += 5) {
    waterServo.write(angle);
    delay(100);
  }
  for (int angle = 120; angle >= 60; angle -= 5) {
    waterServo.write(angle);
    delay(100);
  }
  for (int angle = 60; angle <= 120; angle += 5) {
    waterServo.write(angle);
    delay(100);
  }
  for (int angle = 120; angle >= 60; angle -= 5) {
    waterServo.write(angle);
    delay(100);
  }
  digitalWrite(waterPumpPin, LOW); // Turn off the water pump
}
```

IV. RESULT

This section outlines the data gathered from tests conducted on the Arduino-based fire-fighting robot. The evaluation focused on the robot's capabilities to detect fire, measure temperature, sense smoke, extinguish flames, and send real-time notifications using the ESP8266 Wi-Fi module and the Blynk IoT platform. The results are examined in relation to detection accuracy, response time, and system reliability across various scenarios.

1. *Fire Detection and Extinguishing*

The flame sensors were placed at three positions on the robot: front, left, and right. Multiple fire sources were introduced at varying distances and orientations to test the robot's detection accuracy and response.

Figure 15: Robot detecting flame

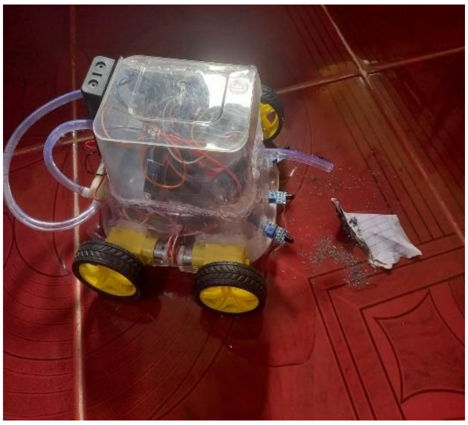

Figure 16: Robot extinguished the flame

2. *Temperature and Smoke Detection*

The temperature sensor (LM35) and gas sensor (MQ-2) were tested by exposing the robot to rising temperatures and smoke, simulating a fire environment.

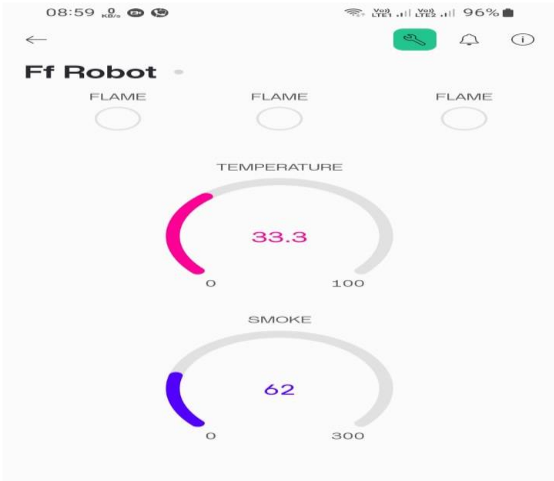

Figure 17: Blynk Platform

3. *Real-Time Notification and Monitoring*

The ESP8266 Wi-Fi module underwent testing to confirm its ability to send real-time notifications to the Blynk IoT platform. The evaluation focused on measuring the time it took to receive notifications on a mobile device when fire, temperature, or smoke was detected.

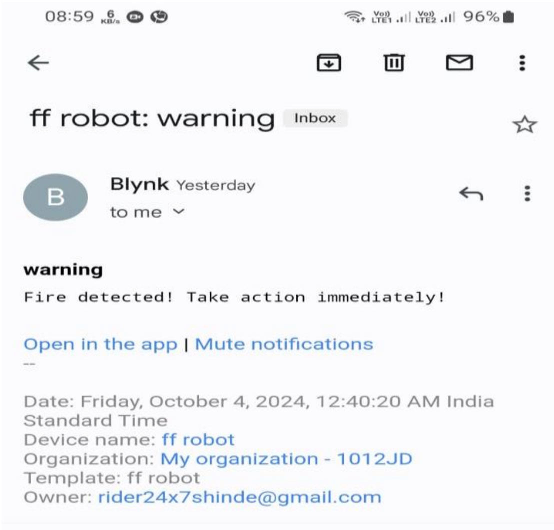

Figure 18: Email Notification

4. *Obstacle Avoidance*

The robot's obstacle avoidance capabilities were tested in a controlled environment with various obstacles. The IR sensor was able to detect obstacles and prevent collisions.

5. *System Reliability and Limitations*

The robot performed well in both detection and extinguishing tasks, but some limitations were noted.

- **Limited Water Supply**: The water pump has a finite capacity, which may not be sufficient for larger fires**.**

- **Range of Detection**: The flame sensors are effective within a range of **2 meters**. Extending the range would enhance the robot's applicability in larger environments.

- **Obstacle Avoidance in Complex Environments**: While the IR sensor was effective in simple environments, more advanced sensors such as ultrasonic sensors could be considered for more complex obstacle detection.

V. DISCUSSION

Q `11 1`qThis paper explores the design and implementation of a firefighting robot powered by Arduino, which can autonomously detect fires, spray water, and send mobile notifications to users. The findings indicate that this Arduino-based firefighting robot effectively addresses fire incidents. By integrating flame, temperature, and smoke sensors with the real-time monitoring features of the ESP8266 Wi-Fi module, the system provides reliable fire detection. Its swift response in extinguishing fires and alerting users underscores its practical application in situations that demand quick action.

VI. CONCLUSION

In this project, we created a fire-fighting robot powered by Arduino that can efficiently detect and extinguish fires with its built-in water spraying system. Utilizing flame sensors, a temperature sensor, and a gas sensor, the robot offers a thorough method for fire detection and safety. Its capability to send real-time alerts to a mobile device through the ESP8266 Wi-Fi module and the Blynk IoT platform improves its functionality, enabling users to keep an eye on fire risks from a distance.

The results showed that the robot was able to detect fires effectively, measure temperature accurately, and sense smoke, all while responding quickly to both detection and extinguishing tasks. Additionally, its obstacle avoidance capability allows for safe navigation, making the robot appropriate for a range of settings, from industrial sites to residential neighbourhoods.

This project highlights the potential of merging robotics and IoT technologies to enhance fire safety. Future improvements might involve extending the detection range, increasing the water supply, and incorporating more sophisticated sensors to further enhance the robot's efficiency. This solution not only acts as a valuable fire-fighting tool but also demonstrates the rising significance of automation and smart technologies in safety applications.

VII. REFERENCES

1. Juin-Ling Tseng and Chia-Wei Chu (2017). A blog on Interaction Design in Virtual Reality Game Using Arduino Sensors
2. Arduino Project Hub (2018). Web Content on Arduino and flame sensor interaction.
3. Dr. Antonio Carlos Bento (2018). Research paper on IoT: NodeMCU 12e X Arduino Uno, Results of an experimental and comparative survey. IJARCSMS, 2321-7782.
4. T. Akilan, Satyam Chaudhary et al., "Surveillance Robot in Hazardous Place Using IoT Technology", *2nd IEEE International Conference on Advances in Computing Communication Control and*

Networking [ICAC3N-20], 01 March, 2021, ISBN 978-1-7281-8338-1.

5. Sanmit Chatterjee (2023). A blog on What are the Key Pros and Cons of the Arduino Programming Language?
6. Vasudhendra Badami (2016). A web content on Arduino programming for beginners-1.