A PROJECT REPORT ON

Autonomous Fire-Fighting Robot Using Arduino

A Main Project Submitted to Jawaharlal Nehru Technological University, Kakinada in partial fulfillments of Requirements for the Award of Degree of

MASTER OF COMPUTER APPLICATIONS

Submitted By

Mani Surya Bhaskar Lingineni
23KT1F0035

Under the Esteemed Guidance of

V.S.R.K Prasad G M. Tech., (Ph.D).,
Associate Professor

DEPARTMENT OF COMPUTER APPLICATIONS (MCA)
Potti Sriramulu Chalavadi Mallikarjuna Rao College of Engineering and Technology
(AUTONOMUS)
Approved by AICTE
Raghava Reddy Street, Kothapeta,Vijayawada-520 001.
2023-2025

POTTI SRIRAMULU CHALAVADI MALLIKARJUNA RAO
COLLEGE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)
Sponsored by SKPVV Hindu High Schools Committee, Estd. L 1906
Approved by AICTE New Delhi | ISO 9001 : 2015 Certified
Accredited by NAAC A++ | NBA | Permanently Affiliated to JNTU, Kakinada (CC-KT)

DEPARTMENT OF COMPUTER APPLICATIONS (MCA)

CERTIFICATE

This is to certify that the project work entitled "**Autonomous Fire-Fighting Robot Using Arduino**" is a bonafide work carried out by **Mr. Mani Surya Bhaskar Lingineni (23KT1F0035) o**n fulfillment for the award of the degree of **MASTER IN COMPUTER APPLICATIONS** of Jawaharlal Nehru Technological University, Kakinada during the year 2023-2025. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the above degree.

Project Guide **Head of the Department**

External Examiner

ACKNOWLEDGEMENT

I would like to extend my heartfelt thanks to the many individuals who have supported, guided, and encouraged me throughout this project.

I am deeply grateful to my Principal, **Dr. S. SARAVANA KUMAR**, for his inspiring words filled with dedication and discipline, which motivated me throughout my work.

My sincere gratitude goes to the Head of Department, **Dr. A. PATHANJALI SASTRI** , Professor & Head of the Department for his unwavering support through training sessions, which served as a crucial foundation for this project.

A special thanks to my project guide, **Mr. V.S.R.K PRASAD**.G Associate Professor, Guide of our project for his meticulous guidance, valuable corrections, and continuous encouragement. His insights and suggestions were invaluable in shaping the outcome of this project.

Lastly, I would like to thank all those who directly or indirectly contributed to the successful completion of this project.

Submitted by

Mani Surya Bhaskar. L **23KT1F0035**

DECLARATION

I certify that:

a) The work contained in this report is original and has been done by me under the guidance of my supervisor.

b) The work has not been submitted to any other institute for any degree or diploma.

c) I have followed the guidelines provided by the Institute in preparing the report.

d) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources ,I have given due credit to them by citing them in the text of the report and providing their details in the references. Furthermore, I have taken permission from the copyright owners of the sources, whenever necessary.

Submitted by

Mani Surya Bhaskar. L **23KT1F0035**

DATE:

PLACE: VIJAYAWADA

Abstract

This project focuses on the design and implementation of an Arduino-based fire fighting robot prototype integrated with IoT capabilities to detect and suppress fires in hazardous environments. The robot is equipped with flame sensors, motor driver module, dc pump which are all interfaced with an Arduino microcontroller to detect fire. Upon fire detection, the robot automatically navigate towards the source and activates an onboard water pump mechanism to stop the fire.

This system enhances fire safety by reducing response time and minimizing human risk in fire-prone areas. The prototype showcases a cost-effective, scalable, and efficient solution for intelligent fire detection and firefighting in residential and industrial settings.

CONTENTS

1. INTRODUCTION

1.1 BACKGROUND

Fire accidents are among the most devastating disasters, often resulting in the loss of lives, property, and environmental damage. Traditional firefighting methods involve human firefighters entering hazardous areas, putting their safety at great risk. In many cases, intense heat, toxic gases, and structural instability make it extremely difficult and dangerous for humans to approach the fire. With advancements in technology, particularly in robotics and automation, there is an opportunity to develop systems that can assist or even replace humans in such dangerous operations.

Robots have proven to be highly effective in environments that are unsafe for humans, such as search and rescue missions, hazardous material handling, and deep-sea exploration. In the context of firefighting, robots can play a vital role by detecting fires early, navigating through complex environments, and extinguishing fires without risking human lives.

1.2 NEED FOR AUTOMATION

Automation in firefighting is crucial for several reasons:

- **Safety**: Reduces human exposure to dangerous environments.
- **Efficiency**: Robots can act faster and reach areas inaccessible or risky for humans.
- **24/7 Operation**: Automated systems can continuously monitor environments without fatigue.
- **Accuracy**: Sensors and programmed algorithms can quickly detect fire sources, minimizing response time.
- **Cost Reduction**: Although there is an initial investment, long-term automation can reduce the costs associated with human labour and potential damages.

An autonomous fire-fighting robot can navigate through obstacles, detect a fire source using sensors, and take immediate action to extinguish it. Such systems can significantly improve emergency response and minimize the impact of fire incidents

1.3 PURPOSE

The primary purpose of this Fire Fighting Robot is to detect the fires and extinguish it in hazardous environments while minimizing the risk to human life. By integrating IoT technology, this system aims to provide faster, safer and more efficient response to fire emergencies, especially in areas that are difficult to reach for fire fighters to access .

1.4 Scope of the Project

This project aims to design and develop a prototype fire-fighting robot capable of:

- Detecting fire using flame and temperature sensors.
- Navigating towards the fire source while avoiding obstacles.
- Extinguishing the fire using a mounted water or chemical spraying mechanism.
- Operating autonomously with minimal human intervention.
- Being scalable and modifiable for various applications such as homes, offices, factories, and warehouses.

The project will focus on creating a low-cost, efficient robot that demonstrates basic fire-fighting capabilities. Future improvements can include enhanced navigation algorithms, integration with IoT platforms for remote monitoring, and AI-based decision-making for handling multiple fire sources simultaneously.

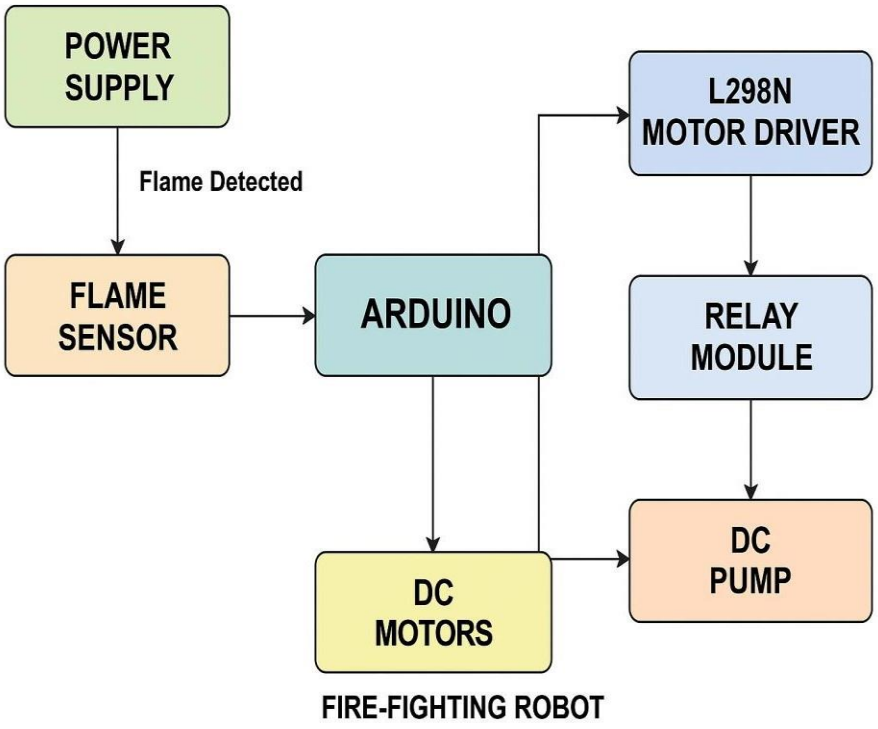
2. OBJECTIVES

The primary objectives of the Fire Fighting Robot project are:

- **Fire Detection**:
  To accurately detect the presence of a fire using appropriate sensors such as flame, smoke, or temperature sensors.
- **Autonomous Navigation**:
  To enable the robot to autonomously move towards the source of the fire while avoiding obstacles in its path.
- **Fire Extinguishing**:
  To develop a mechanism (such as a water sprayer, CO_2 extinguisher, or chemical system) that can effectively extinguish small fires upon detection.
- **Continuous Monitoring**:
  To allow the robot to continuously scan its surroundings for potential fire threats during operation.
- **Safety and Reliability**:
  To ensure that the robot operates safely, minimizing risks to human life and itself while functioning in harsh or risky environments.
- **Compact and Cost-Effective Design**:
  To build a lightweight, compact, and economically viable robot suitable for practical deployment in homes, offices, and industries.
- **Expandability for Future Enhancements**:
  To design the system with flexibility for future upgrades like wireless control, advanced pathfinding algorithms, or AI-based decision-making.

3. SYSTEM OVERVIEW

3.1 BLOCK DIAGRAM



3.2 SYSTEM ARCHITECTURE

The system architecture of the fire fighting robot is designed to perform fire detection and extinguishing tasks autonomously by integrating various hardware and software components. At the core of this system is a **microcontroller** (such as Arduino or Raspberry Pi), which acts as the central processing unit and coordinates all robot operations.

To detect fire, the robot is equipped with **flame sensors**, which identify the presence and direction of a flame, and **temperature sensors**, which confirm abnormal heat levels to ensure reliable fire detection. The robot also uses **ultrasonic sensors** to detect obstacles in its path, enabling it to navigate through the environment safely and efficiently.

The microcontroller receives inputs from all the sensors and processes them in real time. Based on the sensor data, it sends control signals to a **motor driver module (e.g., L298N)**, which then drives the **DC motors** responsible for the robot's movement. This allows the robot to approach the fire source while avoiding any obstacles detected by the ultrasonic sensors.

Once the robot reaches the fire, the microcontroller activates an **extinguishing mechanism**—either a **fan**, which blows out the flames, or a **water pump**, which sprays water to douse the fire. The choice of extinguishing method depends on the design and use case of the robot.

All components are powered by a **battery pack**, typically rechargeable, ensuring mobility and uninterrupted operation. The components are mounted on a mobile **chassis** with wheels, forming the physical body of the robot.

4. REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

1. Arduino UNO
2. Fire sensor or Flame sensor (3 Nos)
3. L298N motor Driver module
4. Mini DC Submersible Pump
5. Relay module
6. Small Breadboard
7. Robot chassis with motors (2) and wheels (4)
8. Asmall bottle

1. Arduino UNO

Arduino Uno Technical Specifications

Microcontroller	ATmega328P-8 bit family microcontroller
Operating voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provides PWM output)
DC Current on I/O Pins	40mA
DC Current on 3.3V Pin	50mA
Flash Memory	32KB (0.5KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

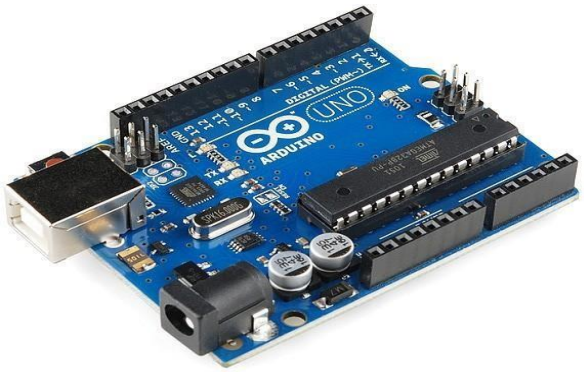


Fig : 1

Arduino UNO pin configuration

Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
I/O Pins	D0 – D13	Can be used as input or output pins
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data
PWM	3,5,6,9,11	Provides 8-bit PWM output
External Interrupts	2,3	To trigger an interrupt
SPI	10(SS),11(MOSI),12(MISO) and 13 (SCK)	Used for SPI communication
Inbuilt LED	13	To turn on inbuilt LED
TWI	A4(SDA), A5(SCA)	Used for TWI communication
AREF	AREF	To provide reference voltage for input voltage

2. Fire or Flame sensor

IR Infrared Flame Module Pin Diagram

3 pin-Flame Sensor Module

Pin Number	Pin Name	Description
1	VCC	+5V
2	GND	Ground (-)
3	OUT	Digital Output (0 or 1)

Working Principle

This sensor/detector can be built with an electronic circuit using a receiver like electromagnetic radiation. This sensor uses the infrared flame flash method, which allows the sensor to work through a coating of oil, dust, water vapor, otherwise ice.

Different Types:

Flame-sensors are classified into four types

- IR single frequency
- IR multi-spectrum
- UV flame detectors
- UV/ IR flame detectors

Features & Specifications

The features of this sensor include the following.

- Photosensitivity is high
- Response time is fast
- Simple to use
- Sensitivity is adjustable
- Detection angle is 600,
- It is responsive to the flame range.
- Accuracy can be adjustable
- Operating voltage of this sensor is 3.3V to 5V
- Analog voltage o/p's and digital switch o/p's
- The PCB size is 3cm X 1.6cm
- Power indicator & digital switch o/p indicator
- If the flame intensity is lighter within 0.8m then the flame test can be activated, if the flame intensity is high, then the detection of distance will be improved.

Applications

These sensors are used in several dangerous situations which includes

- Hydrogen stations
- Industrial heating
- Fire detection
- Fire alarm
- Fire fighting robot
- Drying systems
- Industrial gas turbines

- Domestic heating systems
- Gas-powered cooking devices

3 . L298N Motor Driver module

Working of L298N motor driver

The **L298N** is a dual H-bridge motor driver module that allows you to control the **speed** and **direction** of **two DC motors** or one **stepper motor**. It acts as a bridge between the microcontroller (e.g., Arduino) and the motors, providing the necessary current and voltage.

Input 1 = HIGH (5V)	Output 1 = HIGH	Motor 1 rotates in clock wise direction
Input 2 = LOW (0V)	Output 2 = LOW	
Input 3 = HIGH (5V)	Output 1 = HIGH	Motor 2 rotates in clock wise direction
Input 4 = LOW (0V)	Output 2 = LOW	
Input 1 = LOW (0V)	Output 1 = LOW	Motor 1 rotates in Anti-clock wise direction
Input 2 = HIGH (5V)	Output 2 = HIGH	
Input 3 = LOW (0V)	Output 1 = LOW	Motor 2 rotates in Anti-clock wise direction
Input 4 = HIGH (5V)	Output 2 = HIGH	

Applications

- Used to drive high current Motors using Digital Circuits
- Can be used to drive stepper motors
- High current LED's can be driven
- Relay Driver module (Latching Relay is possible)

4. Mini DC Submersible Pump

Specifications:

- Input Voltage: 2.5V to 6V DC
- Working Current - 130-250mA
- Power 0.4 - 1.5W
- max Lift - 40-110 cm
- Flow Rate - 80-120 L/H
- Outside Dia of Water Outlet - 7.5mm

- Inside Dia of Water Outlet - 4.7mm
- Diameter - 24mm
- Length - 45mm
- Height - 33mm
- Wire Length - 15-20cm
- Material - Engineering Plastic

Working of Mini DC Submersible Pump

A mini submersible water pump is a centrifugal water pump, which means that it uses a motor to power an impeller that is designed to rotate and push water outwards. The motor is located in a waterproof seal and closely connected to the body of the water pump which it powers.

4. **Relay Module**

Working Principle

When the **IN pin** receives a LOW signal (0V), the optocoupler activates the relay, closing the **NO (Normally Open)** contact and allowing current to flow through the external load (like a bulb, fan, etc.).

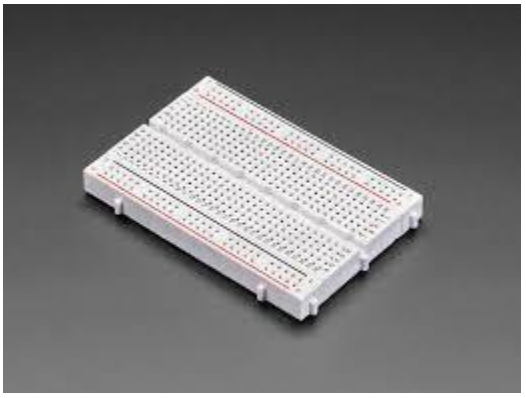
Specifications

Operating Voltage	5V DC
Trigger Voltage	0 – 5v (LOW- level trigger)
Control Signal	Digital
Input Pins	VCC, GND, IN (signal)
Indicator LED	Shows relay ON/OFF state

Applications

- Home Automation Systems
- Automative Projects
- IoT Applications
- Power Management

5. Small Breadboard



Working of small Breadboard

A breadboard is used to build and test circuits quickly before finalizing any circuit design. The breadboard has many holes into which circuit components like ICs and resistors can be inserted.

The bread board has strips of metal which run underneath the board and connect the holes on the top of the board.

The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally while the remaining holes are connected vertically . Because the solderless breadboard does not require soldering, it is reusable.

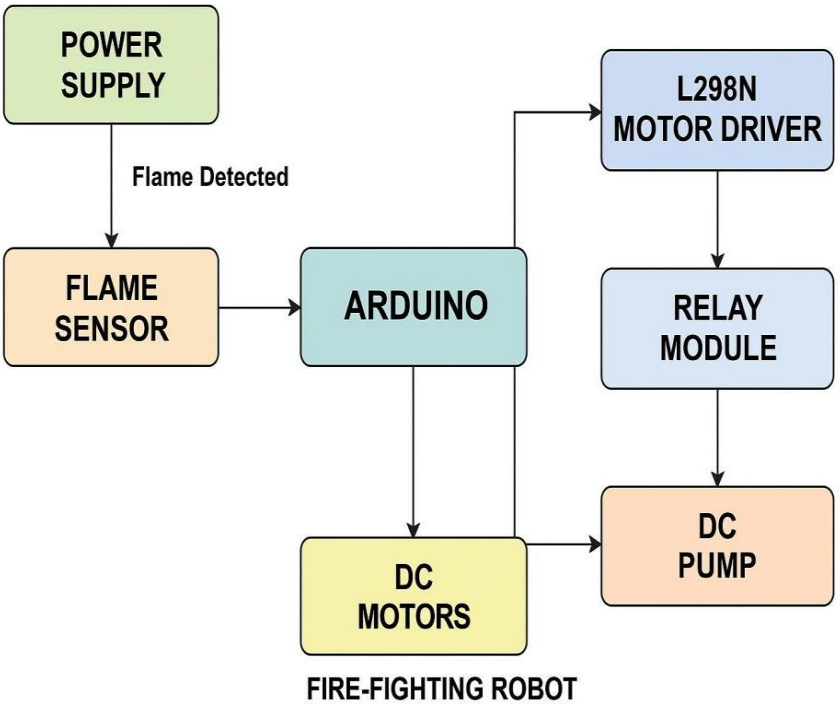
This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property.

A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

7. Robot chassis with wheels

8.Connecting Wires

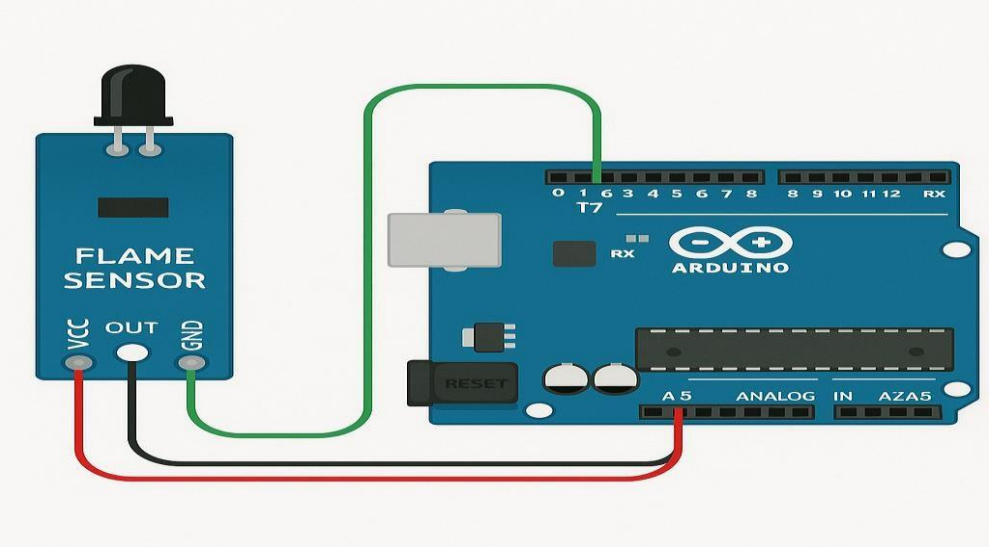
HARDWARE LAYOUT



The hardware layout of the fire-fighting robot is centered around the Arduino Uno microcontroller, which serves as the main control unit for all connected components. A flame

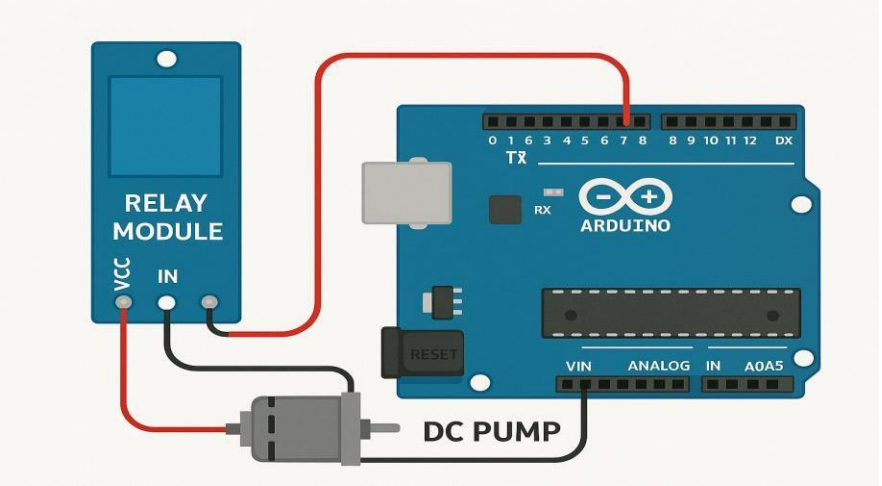
sensor is interfaced with the Arduino to detect the presence of fire by sensing infrared radiation. When a fire is detected, the Arduino processes the signal and takes action accordingly. For movement, two DC motors are connected via the L298N motor driver module, allowing the robot to navigate toward the fire. The direction and speed of these motors are controlled by the Arduino through digital and PWM signals. A relay module is also connected to the Arduino, acting as a switch that controls a DC water pump. When fire is detected, the Arduino activates the relay, turning on the pump to spray water or an extinguishing agent through a mounted nozzle. All components are powered through a regulated battery pack, and the layout ensures efficient routing of power and signal connections for smooth operation. This arrangement allows the robot to autonomously patrol an area, detect fire, move toward it, and extinguish it effectively.

FLAME SENSOR CIRCUIT CONNECTION



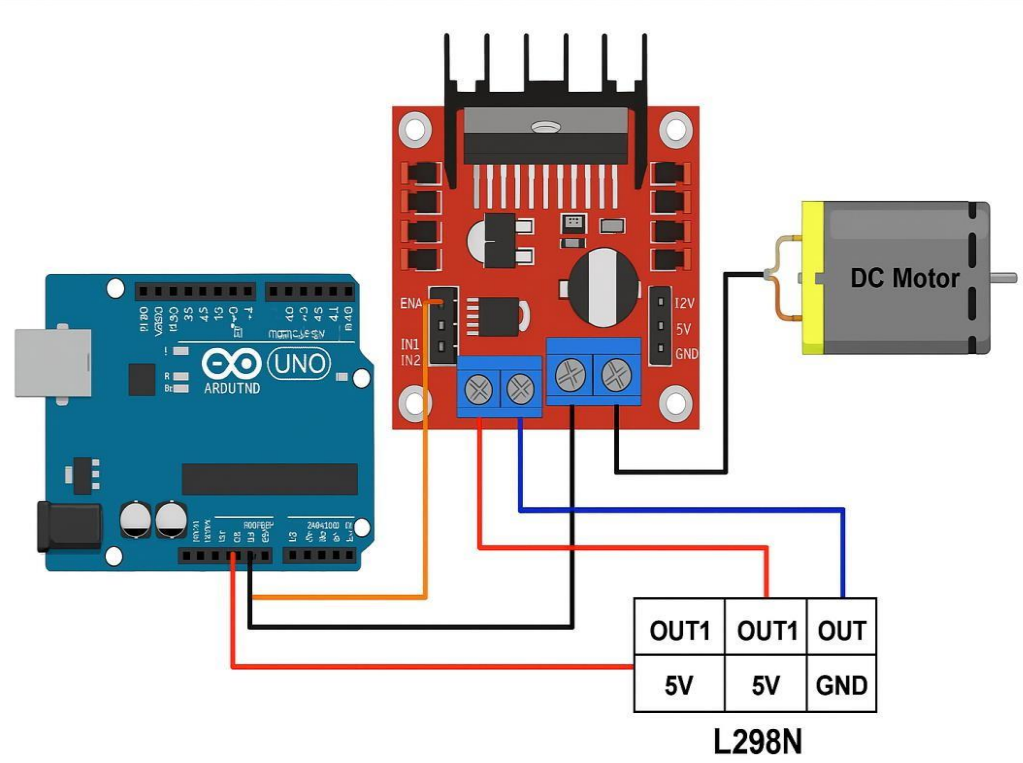
The 3-pin flame sensor module is connected to the Arduino Uno for detecting the presence of fire through infrared radiation. The **VCC pin** of the sensor is connected to the Arduino's **5V pin** to supply power, the **GND pin** is connected to **GND** on the Arduino, and the **OUT pin** is connected to one of the Arduino's digital input pins (e.g., D7). When the flame sensor detects a flame, it sends a LOW signal to the Arduino, allowing the microcontroller to initiate actions such as stopping the robot or activating a fire-extinguishing pump.

INTERFACING OF RELAY MODULE WITH DC PUMP



The relay module acts as an electrically operated switch that allows the Arduino Uno to control high-power devices like the DC pump. In this setup, the **VCC** pin of the relay is connected to the Arduino's **5V**, the **GND** pin to **GND**, and the **IN** pin is connected to a digital output pin (e.g., D8). The DC pump is connected in series with the relay's Normally Open (NO) terminal and a power supply. When the Arduino sends a HIGH signal to the relay's IN pin, the relay closes the circuit, activating the pump to spray water and extinguish fire. This setup enables safe and effective control of the pump through low-power signals.

L298N DRIVER MODULE CONNECTION



The L298N motor driver module is used to control the direction and speed of DC motors in the fire-fighting robot. In this setup, the **ENA** pin (Enable A) is connected to a PWM-enabled pin on the Arduino (e.g., D9) to regulate speed. The **IN1** and **IN2** pins are connected to Arduino digital pins (e.g., D4 and D5) to control the direction of the motor. The **OUT1** and **OUT2** terminals connect to the motor terminals. Power is supplied to the L298N through its **12V** and **GND** pins, and the onboard 5V regulator can power the Arduino if required. This configuration allows the Arduino to fully control the motor's motion based on sensor input or programmed logic.

4.2 Software Requirements:

Arduino IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board.

Refer to the Getting Started page for Installation instructions.

It's hardware products are licensed under a CC BY-SA license, while software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL)permitting the manufacture of Arduino boards and software distribution by anyone.

Arduino boards are available commercially from the official website or through authorized distributors. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits.

The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the Arduino language, inspired by the Processing language and used with a modified version of the Processing IDE.

In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command line tool developed in Go. aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators.

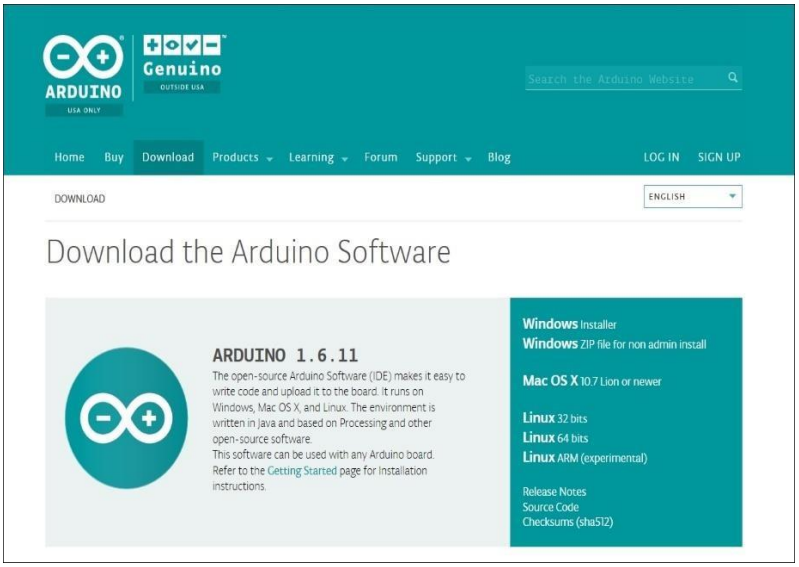
Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors. The Arduino integrated development environment (IDE) is a cross-platform application (for Microsoft Windows, macOS, and Linux) that is written in the Java programming language.

It originated from the IDE for the languages Processing and Wiring. It includes a 8 code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board.

It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2 .The Arduino IDE supports the languages C and C++ using special rules of code structuring.

The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main () into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.



INTRODUCTION TO C++

Arduino is a popular open-source platform for building electronic projects, and it uses a simplified version of C++ for programming. With Arduino, you can interact with sensors, motors, LEDs, and other hardware components through C++ code. This guide provides an overview of the key concepts in C++ as applied to Arduino development.

1.Setting Up Arduino Environment

To get started with Arduino programming:

1. Download and install the **Arduino IDE** from the official Arduino website.
2. Connect your Arduino board to your computer via USB.
3. Open the Arduino IDE and select the correct board and port under **Tools**.

2.Basic Structure of Arduino Code

An Arduino program consists of two main functions:

- setup(): This function is called once when the Arduino is powered on or reset. It's used to initialize variables, set pin modes, and perform setup operations.
- loop(): This function is called repeatedly in a continuous loop. It's used for the main program logic, such as reading sensors, controlling motors, or handling events.

```
void setup() {
  // Initialize the pin connection setup here
}

void loop() {
  // looping the code and apply the logic
}
```

3. Variables and Data Types

Arduino uses standard C++ data types, such as int, float, char, and boolean, but there are also some Arduino-specific types for hardware interfacing, such as byte and word.

- int for integer values (e.g., pin numbers, loop counters)
- float for decimal values (e.g., sensor readings)
- boolean for true/false values

```
int ledPin = 13;  // Pin number for the LED
float temperature = 22.5; // Temperature reading
bool ledState = false; // LED on/off state
```

4.Digital and Analog I/O

Arduino boards have **digital** and **analog** input/output pins. C++ allows you to control and read these pins in your program.

Digital Pins: Used for simple on/off signals (HIGH or LOW).

- pinMode(pin, mode): Sets the mode of a pin (INPUT, OUTPUT, INPUT_PULLUP).
- digitalWrite(pin, value): Write HIGH (1) or LOW (0) to a digital pin.
- digitalRead(pin): Read the state of a digital pin (HIGH or LOW).

```
int ledPin = 13; // Digital pin 13
void setup() {
  pinMode(ledPin, OUTPUT);  // Set pin 13 as an output
}

void loop() {
  digitalWrite(ledPin, HIGH);  // Turn the LED on
  delay(1000);  // Wait for 1 second
  digitalWrite(ledPin, LOW);  // Turn the LED off
  delay(1000);  // Wait for 1 second
}
```

Analog Pins: Used for continuous values (e.g., sensor readings).

- analogRead(pin): Reads an analog value from a pin (0 to 1023).
- analogWrite(pin, value): Writes an analog value (PWM) to a pin (0 to 255).

```
int sensorPin = A0;  // Analog pin A0
int sensorValue = 0;

void setup() {
  Serial.begin(9600);  // Start serial communication
}

void loop() {
  sensorValue = analogRead(sensorPin);  // Read the sensor value
  Serial.println(sensorValue); // Print the value to the Serial Monitor
  delay(1000);  // Wait for 1 second
}
```

5.Functions

Functions in Arduino work similarly to standard C++ functions. You can create functions to modularize your code and perform specific tasks.

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  blinkLED();
  delay(1000);
}

void blinkLED() {
  digitalWrite(13, HIGH); // Turn LED on
  delay(500);             // Wait for 500 ms
  digitalWrite(13, LOW);  // Turn LED off
}
```

6. Loops and Conditionals

You can use loops (for, while) and conditional statements (if, else) for more complex logic.

```
int buttonPin = 2; // Button connected to pin 2
    int buttonState = 0;

    void setup() {
     pinMode(buttonPin, INPUT);
     pinMode(13, OUTPUT);
    }

    void loop() {
     buttonState = digitalRead(buttonPin);
     if (buttonState == HIGH) {
       digitalWrite(13, HIGH);  // Turn LED on
     } else {
       digitalWrite(13, LOW);  // Turn LED off
     }
    }
```

7. Communication Protocols

Arduino supports various communication protocols like **Serial Communication**, **I2C**, and **SPI**.

- **Serial Communication**: Used to communicate with a computer or another microcontroller.
  - Serial.begin(baudrate): Initializes serial communication.
  - Serial.print(data): Sends data to the Serial Monitor.

```
void setup() {
  Serial.begin(9600);  // Start serial communication at 9600 baud
}

void loop() {
  Serial.println("Hello, Arduino!");
  delay(1000);  // Wait for 1 second
}
```

- **I2C**: A communication protocol used for connecting multiple devices with two wires (SDA, SCL).
- **SPI**: A communication protocol used for high-speed data transfer between microcontrollers and peripherals.

8. Libraries

Arduino provides numerous libraries to make it easier to interface with external hardware such as sensors, motors, displays, etc. Libraries can be included in your sketches using #include.

```
#include <Wire.h>  // I2C library
#include <LiquidCrystal_I2C.h> // LCD library

LiquidCrystal_I2C lcd(0x27, 16, 2); // Create LCD object

void setup() {
  lcd.begin(16, 2); // Initialize LCD
  lcd.print("Hello, Arduino!");
}

void loop() {
  // Display some text
}
```

9. Advanced Topics

- **Interrupts**: Used for handling external events (like a button press) immediately, rather than waiting for the next loop.
- **Timers**: Useful for precise timing in your program, such as creating delays without using delay().
- **PWM (Pulse Width Modulation)**: Used to control motors or dim lights by simulating analog output on digital pins.

4.3 DEVELOPMENT TOOLS

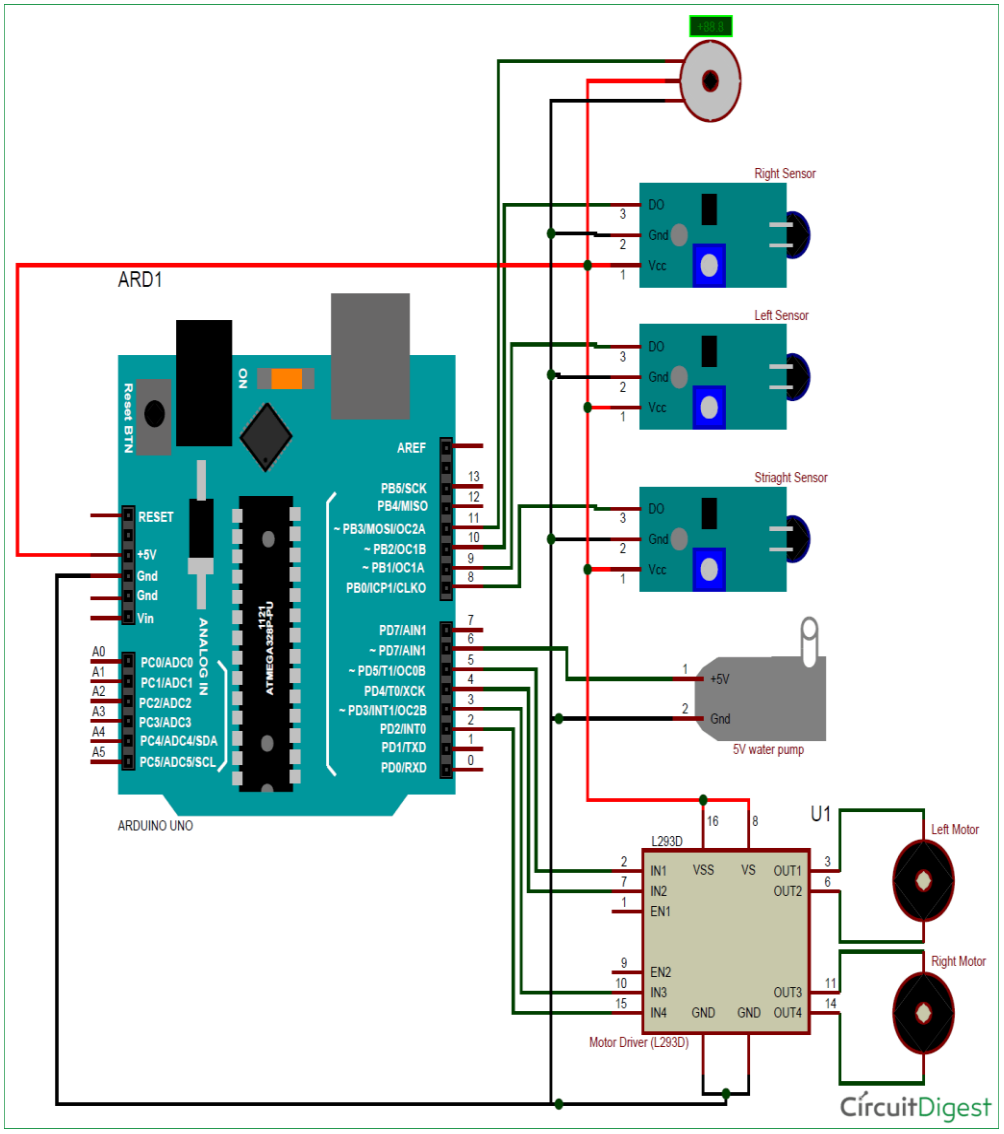
To successfully build and test the fire fighting robot, several development tools and equipment are required. A **soldering kit** is essential for creating strong and permanent electrical connections between components.

A **multimeter** is used to test voltage levels and continuity in circuits, ensuring correct wiring and operation.

Screwdriver sets are needed for assembling the mechanical parts of the robot, while a **hot glue gun** or **mini drill** is helpful for mounting sensors and modules securely onto the chassis. The programming and debugging of the robot are carried out on a **laptop or personal computer**, using the appropriate IDEs and libraries.

A **USB cable** is used to upload the code to the microcontroller and to establish serial communication for monitoring real-time sensor data. These tools together form the physical and software environment required for building, testing, and optimizing the robot's functionality.

CIRCUIT DIAGRAM AND WIRING



5. IMPLEMENTATION

As you can see these sensors have an IR Receiver (Photodiode) which is used to detect the fire. How is this possible? When fire burns it emits a small amount of Infra-red light, this light will be received by the IR receiver on the sensor module. Then we use an Op-Amp to check for change in voltage across the IR Receiver, so that if a fire is detected the output pin (DO) will give 0V(LOW) and if the is no fire the output pin will be 5V(HIGH).

So, we place three such sensors in three directions of the robot to sense on which direction the fire is burning.

We detect the direction of the fire we can use the motors to move near the fire by driving our motors through the L298N Motor Driver Module. When near a fire we have to put it out using water. Using a small container we can carry water, a 5V pump is also placed in the container .

Based on the robotic chassis that you are using you might not be able to use the same type of container that I am using. In that case use your own creativity to set up the pumping system. However the code will remain same. I used a small can (cool drinks bottle) to set the pump inside it and poured water inside it. I then assembled the whole can on top of a servo motor to control the direction of water. My robot looks something like this after assembly.

Programming Arduino:

Once you are ready with your hardware, you can upload the Arduino code for some action. The complete program is given at the end of this page. However I have further explained few important bits and pieces here.

As we know the fire sensor will output a HIGH when there is no fire and will output a LOW when there is fire. So we have to keep checking these sensor if any fire has occurred. If no fire is there we ask the motors to remain moving by making all the pins high as shown below

We have implemented this code functionality with functions or methods …

```
 if (flame_detected_Center == HIGH &&flame_detected_Left == HIGH
&&flame_detected_Right == HIGH)
  {
     Serial.println("No Fire detected! moving robot..");
     moveForward();

  }
```

```
//function for moveForward()

void moveForward()
{

  analogWrite(ENA , speed);
  analogWrite(ENB , speed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
```

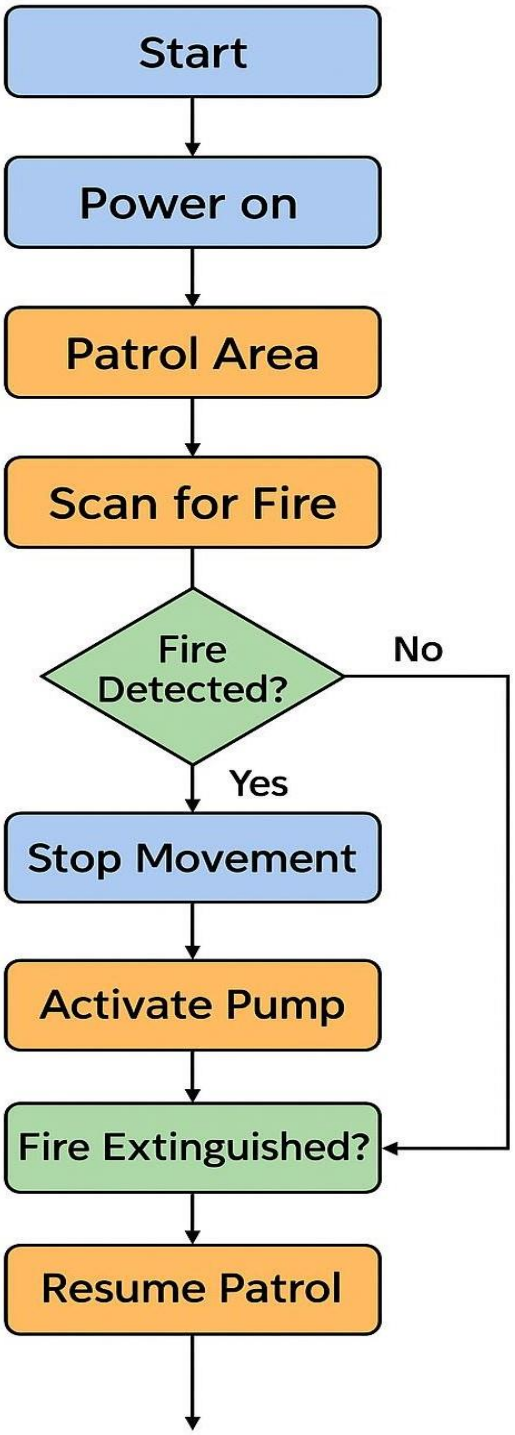
Similarly, if there is any fire we can ask the robot to move in that direction by rotating the respective motor. Once it reaches the fire the left and right sensor will not detect the fire as it would be standing straight ahead of the fire .

```
 if(flame_detected_Center == LOW &&flame_detected_Left == HIGH
&&flame_detected_Right == HIGH)  //if fire is straight ahead
  {
    Serial.println("fire detected. stop robot...");

    stopRobot();

    digitalWrite(relay,HIGH);
    delay(5000);
    digitalWrite(relay,LOW);
  }

//function for stopRobot()

void stopRobot()
{
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
```

Integration of Hardware and Software

The successful operation of the fire-fighting robot relies on the seamless integration of both hardware components and software logic. The hardware—comprising the Arduino Uno, flame sensor, L298N motor driver, relay module, DC motors, and water pump—is connected through proper wiring and power supply channels. The software, developed using the Arduino IDE, consists of code that reads sensor data, controls motor movements, and activates the pump via the relay. The flame sensor continuously monitors for fire; upon detection, the Arduino processes the input and executes a series of commands to stop the robot and activate the water pump. Simultaneously, the L298N module interprets motor control signals from the code to guide the robot's movement. This tight coupling between the program logic and hardware execution ensures that the robot functions autonomously, responding to environmental stimuli in real time to detect and extinguish fires efficiently.

FLOW CHART


Start
Power on
Patrol Area
Scan for Fire
Fire Detected?
No
Yes
Stop Movement
Activate Pump
Fire Extinguished?
Resume Patrol

SOURCE CODE

```
/* FIRE FIGHTING ROBOT CODE */

#define FLAME_SENSOR_1 2  //PIN 2
#define FLAME_SENSOR_2 8  //PIN 8
#define FLAME_SENSOR_3 12 //PIN 12
#define ENA 9
#define IN1 3
#define IN2 4
#define IN3 5
#define IN4 6
#define ENB 10
#define relay 13

int speed = 105;

void setup()
{
   pinMode(FLAME_SENSOR_1, INPUT);
   pinMode(FLAME_SENSOR_2, INPUT);
   pinMode(FLAME_SENSOR_3, INPUT);
   pinMode(IN1, OUTPUT);
   pinMode(IN2, OUTPUT);
   pinMode(IN3, OUTPUT);
   pinMode(IN4, OUTPUT);
   pinMode(relay,OUTPUT);
   pinMode(ENA, OUTPUT);
   pinMode(ENB, OUTPUT);
   digitalWrite(relay,LOW);

   Serial.begin(9600);

}
```

```
void loop()
{
   int flame_detected_Center = digitalRead(FLAME_SENSOR_1);
   int flame_detected_Left = digitalRead(FLAME_SENSOR_2);
   int flame_detected_Right = digitalRead(FLAME_SENSOR_3);


  if (flame_detected_Center == HIGH &&flame_detected_Left == HIGH
&&flame_detected_Right == HIGH)
  {
      Serial.println("No Fire detected! moving robot..");
      moveForward();


   }
   if(flame_detected_Center == LOW &&flame_detected_Left == HIGH
&&flame_detected_Right == HIGH)
   {
      Serial.println("fire detected. stop robot...");

      stopRobot();

      digitalWrite(relay,HIGH);
      delay(5000);
      digitalWrite(relay,LOW);
   }
   if(flame_detected_Left == LOW &&flame_detected_Center == HIGH
&&flame_detected_Right == HIGH)
   {
      Serial.println("fire  detected.  Moving  left...");
      stopRobot();
      delay(1000);
      moveLeft();
      delay(1000);
      stopRobot();
      delay(5000);
      digitalWrite(relay,HIGH);
      delay(5000);
      digitalWrite(relay,LOW);
      moveRight();
```

```
    delay(1000);
    stopRobot();
    delay(500);


  }

  if(flame_detected_Right == LOW &&flame_detected_Center == HIGH
&&flame_detected_Left == HIGH)
  {
    Serial.println("fire detected. Moving right...");
     stopRobot();
    delay(1000);
    moveRight();
    delay(2000);
    stopRobot();
    delay(5000);
    digitalWrite(relay,HIGH);
    delay(5000);
    digitalWrite(relay,LOW);
    moveLeft();
    delay(2000);
    stopRobot();
    delay(500);

  }
  delay(300); // Short delay for stability
}

void moveForward()
{

  analogWrite(ENA , speed);
  analogWrite(ENB , speed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
```

```
void moveBackward()
{

  analogWrite(ENA , speed);
  analogWrite(ENB , speed);
  digitalWrite(IN1, LOW);


  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void stopRobot()
{
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}

void moveLeft()
{
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void moveRight()
{
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4,LOW);
}
```

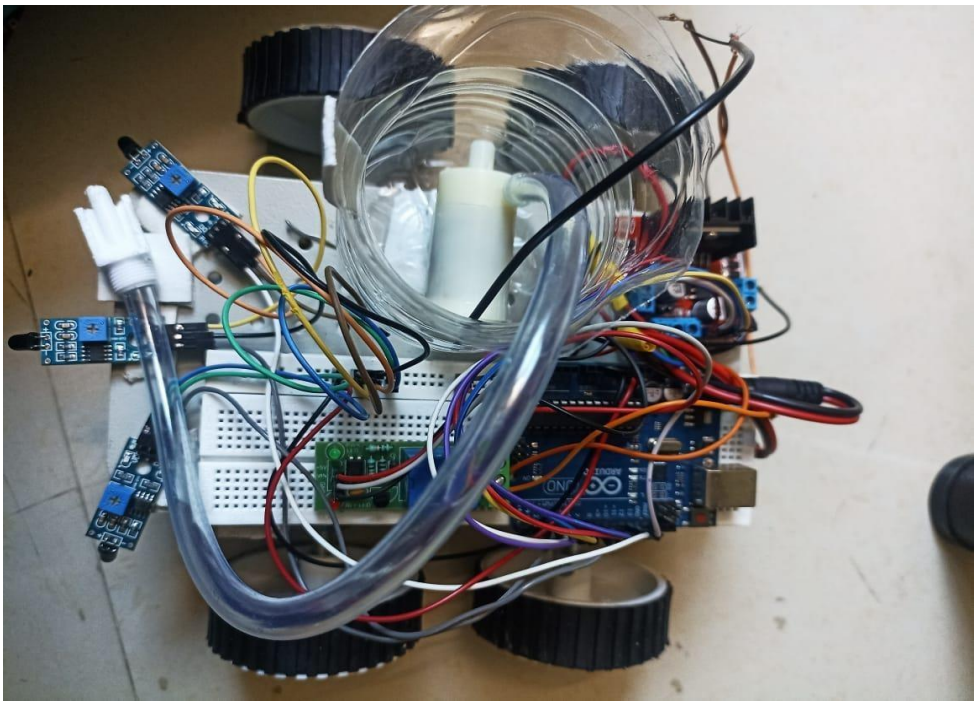
6. TESTING

The robot was tested in a controlled indoor environment. It successfully detected small flames from a lighter/candle and navigated accurately to the source. The fire extinguishing system was effective in putting off the fire within a few seconds.

7. RESULT

In this project, an autonomous Firefighting Robot has been implemented which is capable of detecting flames and extinguishing them successfully. This robot can move forward, move left & right flawlessly. The motors and Arduino code work together to control the movement of the robot. If any of the flame sensors are triggered, then the motor driver module will start to rotate & move the robot to the danger point upon receiving a signal about the danger environment & start to pump the water with the help of relay module. This process will be continued until the fire has been extinguished completely.

After successfully building the project, the simulation was run and the desired output was obtained. Proper snapshots of the results were attached. Thus, an autonomous firefighting robot has been built to achieve the objectives of this project successfully.



8. CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

The Fire Fighting Robot is effective enough to fight against fire on a small scale. It can sense fire flame better at darker places. It is made as a preventer robot. Because it can detect fire instantly and can extinguish it before spreading. This multisensory based robot may be a solution to all fire hazards. Various sensors like flame have been incorporated in this robot. If the fire is detected, a water spraying mechanism is triggered to extinguish the fire. With enough funding and scope, this design of robot can also fight against large fire with larger reserving capacity and an improved sensing unit can provide even an earlier detection of fire at all circumstances.

As a conclusion, the project entitled "**Fire Fighting Robot**" has achieved its aim and objective successfully.

8.2 FUTURE SCOPE

Future work can include the transformation of the experimental robot prototype into a practical robot, which requires improvement in its overall performance. Face detection system can develop for a fire-fighting robot to rescue human beings caught in fire. The face detection system alerts the presence of human beings caught in fire to facilitate their rescue.

The ultrasonic sensor can also be attached to detect any object around the robot to avoid any collision with other objects. Wireless remote-control idea can be introduced to this system so that human can control the mechanism of the robot by their own needs. and its performance can be enhanced by interfacing it with a wireless higher Resolution zooming Camera so that the person controlling it can view the operation of the robot remotely on a screen.

9.BIBILIOGRAPHY

[1]. Automated robot fire extinguisher-2011 8th International Ubiquitous Robots and Ambient Intelligence Conference (URAI)

[2]. Development and implementation of dual mode fire extinguishing robot based on arduino microcontrollers: 2017 IEEE International Conference on Intelligent Control, Optimization and Signal Processing Techniques (INCOS) (INCOS) (INCOS)

[3]. Cease Fire: The Fire Fighting Robot: 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)

[4]. Fire extinguisher robot based on multiple sensors based on DTMF, bluetooth and GSM technology with multiple modes of operation: 2016 International Computational Intelligence Workshop 2016 (IWCI)

[5]. Intelligent Fire Fighting Robot(2010) Kristi Kosasih, E. Merry Sartika, M. Jimmy Hasugin, dan Mulidy Electric Motorering Department ,Maranatha Christian University J1.Prof.Drg. Sumantri Suria.

Plagiarism Checker X - Report

Originality Assessment

35%

Overall Similarity

Date: May 9, 2025 (10:37 AM)
Matches: 2109 / 6072 words
Sources: 42

Remarks: High similarity detected, please make the necessary changes to improve the writing.

Verify Report:
Scan this QR Code