EC280 - MINI PROJECT IN ELECTRICAL CIRCUITS AND SYSTEMS

ARDUINO BASED FIRE FIGHTING ROBOT
PROJECT REPORT

SUBMITTED BY

PITTA MANOJ - 211EC236

RAVIPATI VARSHITH - 211EC240

UNDER THE GUIDANCE OF

Dr. SUSHIL KUMAR PANDEY

NATIONAL INSTITUTE OF TECHNOLOGY ,KARNATAKA

SURATHKAL,SRINIVASNAGAR,MANGALORE

Table of Contents

Abstract

The fire-fighting robot project is an innovative solution to combat fire accidents by providing a reliable and efficient method for fire extinguishing. The robot is built using an Arduino Uno microcontroller and a flame sensor. It is designed to detect the presence of a fire, and extinguish it using a water pump and a pipe. The project involves designing the hardware, programming the microcontroller, and testing the robot's functionality. This project report aims to provide a comprehensive guide on building the fire-fighting robot and documenting the process involved.

Introduction

Fires can cause significant damage to property and threaten human lives. It is essential to detect and extinguish fires as quickly as possible. Firefighting robots have been developed to address this issue, and this project aims to build one such robot based on the Arduino platform. This robot will have the ability to detect fires, navigate through obstacles, and extinguish the fire using water pumped from a container.

The robot will use various components, including a smart car kit, a flame sensor module, 18650 batteries, an Arduino Uno, an L293 motor driver, an LM2596 DC-DC buck converter module, a mini modular breadboard, a water pump, a MG90S micrometer gear servo, jumper wires, a SIM800L, and a relay module.

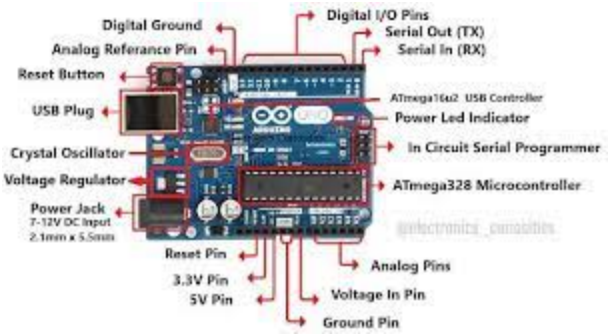
The robot's functionality will be achieved through programming the Arduino Uno to control the various components. The flame sensor will detect the fire's presence. The water pump will then activate, and the robot will move towards the fire, extinguishing it with water. The SIM800L module will enable remote control of the robot using a mobile device.

This project aims to demonstrate the potential of Arduino-based firefighting robots in enhancing fire safety in various settings.

Components Used

- Smart Car Kit
- 1-channel Flame Sensor Module(3X)
- Arduino Uno
- L293 Stepper Motor Driver18650 3.7V battery (2x)
- LM-2596 DC-DC Buck Converter Module
- Mini Modulator Breadboard
- Water Pump (5V)
- Metre Pipe
- Rocker switch
- MG90S Micro Metal Gear Servo
- SIM-800L
- Relay Module

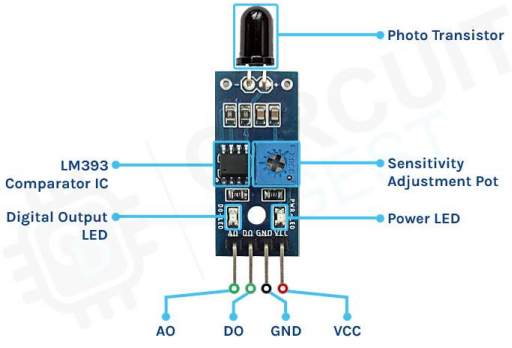
1. Arduino Uno



Arduino Uno is a microcontroller board based on the ATmega328P microcontroller.

In the project, the Arduino Uno can be used as the main controller for the robot. It can receive input signals from the sensors, process them, and then send output signals to control the various components of the robot. It can also communicate with the SIM800L GSM module to receive commands and transmit data wirelessly.

2. 1-Channel Flame Sensor Module



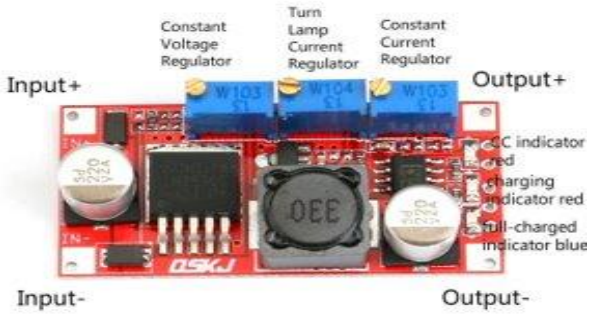
The 1 channel flame sensor module is used to detect the presence of fire or flames. When a flame is detected, the module will output a HIGH signal,

indicating that a flame is present. Conversely, when there is no flame detected, the module will output a LOW signal. The Arduino can then activate the water pump to extinguish the fire.

3. L293 DC Motor Driver

In this project, the L293D is used to control the four DC motors that are responsible for moving the robot towards the flame.

4. LM2596 DC-DC Buck Converter Module



The LM2596 DC-DC buck converter module is a popular module used to step-down voltage i.e used to convert a higher voltage source to a lower voltage that is suitable for use in our project. The two 18650 batteries used to power the robot provide a total voltage of 7.4V, which can be used to step down the voltage to 5V and use it for the SIM800L module.

5. MG90S Micro Metal Gear Servo

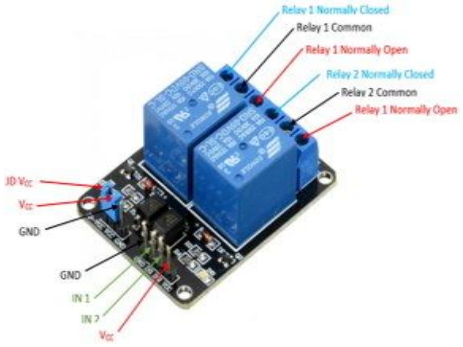
In this project, the MG90S servo motor can be used to control the movement of the water pump. The servo motor can be connected to a linkage or arm that is attached to the water pump's on/off switch.

6. SIM800L

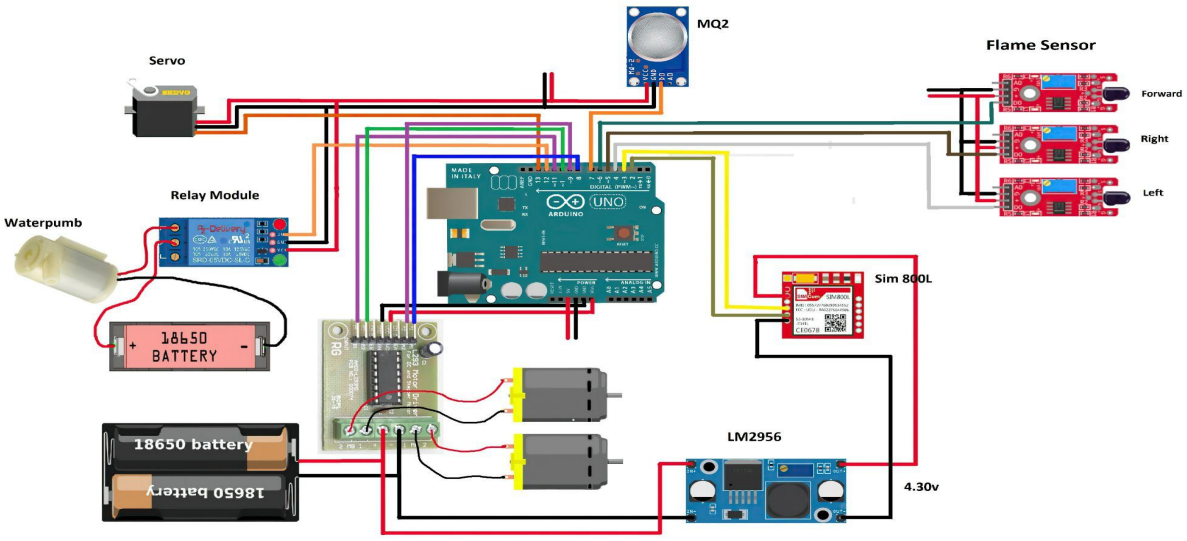
The SIM800L is a small, low-power GSM/GPRS module that enables wireless communication in various applications. In this project, the SIM800L module can be used to send SMS messages to the mobile phone when a fire has been detected and extinguished.

7. Relay Module

The relay module is a device that allows you to control high voltage and high current circuits using a low voltage signal from a microcontroller such as the Arduino Uno. In this project, we can use the relay module to control the water pump that is used to spray water on the fire.



Schematic Diagram



Assembly

1. Smart Car Kit

- Attach the motors to the chassis.
- Connect the motors to the battery box.
- Attach the wheels to the motors.
- Test the motor connections.

2. Motors

- Take two motors which correspond to the front and back of the left side of the robot and connect their joins in a cross connection. This will represent the left motors. Repeat the same for the right motors.

3. L293 Motor Driver

- The power supply's positive wire should be connected to the VCC pin on the driver, while the negative wire should be connected to the GND pin. This will provide 7.4V and supply 1A to the driver.
- The motor driver has two channels (M1 and M2), each of which can drive one motor. Connect the left side motors to M1 channel and right side motors to M2 channel.
- Connect the driver's control pins (IN1, IN2, IN3 and IN4) to four other digital pins on the Arduino (Pin8, Pin9, Pin10, Pin11).
- Connect the 5V and OUT pins to the Vin and GND pins of the Arduino. This will provide 5V and supply 1A to the Arduino.

4. LM2596 DC-DC Buck Converter Module

- Adjust the output voltage of the LM2596 by turning the potentiometer on the module to the desired voltage (4.3V). Use a multimeter to measure the output voltage to ensure that it is correct.
- Connect the positive and negative wires from the input power source to the IN+ and IN- pins on the module.

- Connect the positive and negative wires from the output of the module to the Vcc and GND pins on the SIM800L.

5. Flame Sensor

- Solder the 3 GND pins of the three sensors used to provide a common GND. Repeat the same for the 3 Vcc pins.
- Connect the D0 pin of each of the modules to digital input pins of the Arduino(Pin4, Pin5, Pin6).

6. Water Pump

- Connect the positive(red) wire to the Normally Closed slot in the relay module and the negative(black) wire to the negative terminal of the power supply.

7. MG90S Micro Metal Gear Servo

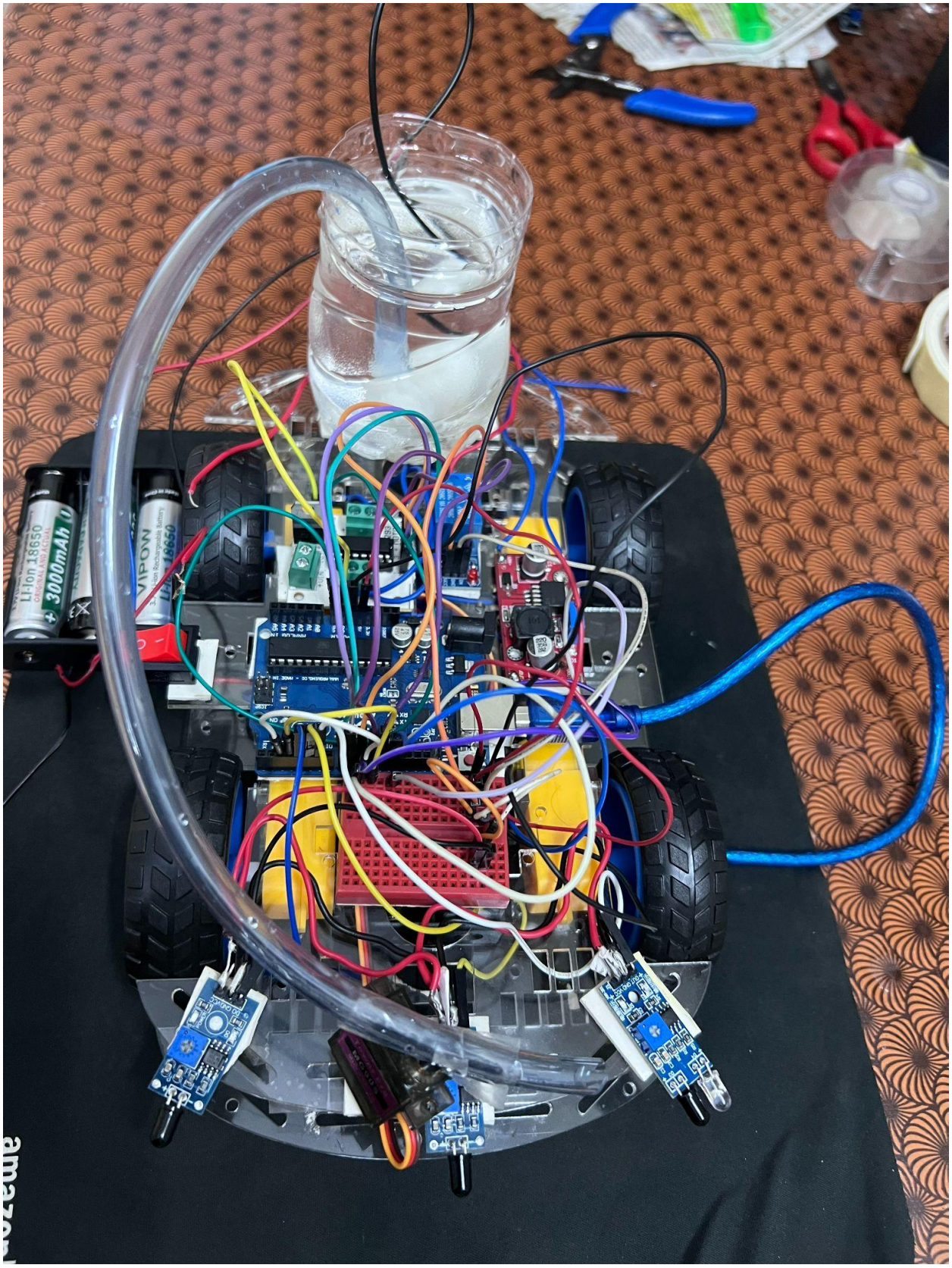
- Connect the black(or brown), red and yellow(orange) wires to the GND, 5V and a PWM pin on the Arduino(Pin13).

8. Relay Module

- Connect the Vcc and GND and connect the IN pin to the Arduino(Pin12).
- Connect the positive terminal of the power supply to the Common contact slot of the relay module.

9. SIM800L

- Connect the VCC and GND and connect the RXD and TXD pins to the Arduino(Pin2, Pin3).


3000mAh
VIPOW

Working

There are three sensors which cover around 70-90 degrees in front of the robot. The left sensor activates the right wheels, the right sensor activates the left wheels and the forward sensor activates all the wheels.

When we bring a flame source to the forward sensor, it detects the flame and then sends signals to the Arduino which will then turn on all the motors and then the robot moves straight towards its source. If the left or right sensor detects the flame, then the right or left wheels will rotate towards the flame source until the flame source comes in front of the forward sensor and then all the motors will be turned on and then the robot moves towards the flame source. The motors are turned on by sending signals through the pins connected to the L293 Motor driver which drives the individual channels for the left and right wheels.

Once the robot reaches the flame source, it will stop and then will turn on the Relay module. The relay module will then activate the water pump connected to the container holding the water. The pipe connected to the container will then draw water whose other end is connected to the shaft of the Gear Servo which rotates to spray water on the flame source. The relay will remain turned on until the flame source is extinguished and then gets turned off.

When the flame source is detected, the SIM800L module sends an SMS to the mobile number provided in the Arduino saying “Fire Detected!!” and will again send a message “Fire Extinguished” once the flame has been extinguished. Alternatively, we can make a phone call to the mentioned mobile number also. The SIM800L requires an operating voltage of 3.3V to 4.4V, hence the 7.4V power supply provided is step-down to 4.3V using the LM2596 DC-DC Buck Converter Module.

Sources of Error

1. **Flame Sensor Accuracy:** The flame sensors may have limitations in accurately detecting flames, especially if there are environmental factors like strong ambient light or interference from other heat sources. False positives or false negatives in flame detection can occur.
2. **Sensor Placement:** Improper placement of the flame sensors may result in suboptimal detection coverage. If the sensors are not positioned correctly, some flames may go undetected, leading to delayed response or incomplete fire suppression.
3. **Obstacle Detection:** The obstacle detection mechanism, such as the ultrasonic sensor, may not detect all obstacles accurately. This can result in collision or difficulty navigating through complex environments, affecting the robot's ability to reach the fire location.
4. **Servo Accuracy:** The accuracy and precision of the MG90S servo motor may have limitations. Small variations in servo movement or mechanical constraints may impact the desired orientation of the water spray nozzle, potentially affecting the effectiveness of fire suppression.
5. **Water Supply and Pressure:** The water pump's performance, water supply, and pressure can vary, affecting the water spray distance and coverage. Inadequate water supply or low pressure may limit the robot's ability to effectively extinguish the fire.
6. **Power Supply:** Issues related to the power supply, such as insufficient battery capacity or voltage fluctuations, can impact the overall performance and runtime of the robot. It is important to ensure a stable and reliable power source to avoid unexpected shutdowns or malfunctions.
7. **Communication Interference:** If communication modules, such as the SIM800L module, are used for remote control or monitoring, interference or signal loss can

occur, affecting the reliability of communication between the robot and the control center.

8. **Mechanical and Electrical Connections:** Loose or unreliable connections between components, such as jumper wires or motor connections, can lead to intermittent or faulty operation. It is crucial to ensure secure and proper connections to minimize potential errors.
9. **Environmental Factors:** External factors like temperature, humidity, or physical obstructions can influence the performance of the robot and its components. Extreme temperatures or exposure to water or dust may impact the reliability and longevity of the system.

Arduino Code

```
#include <Servo.h>             //include servo.h library
#include <SoftwareSerial.h>  //include SoftwareSerial.h library

Servo myservo;

int pos = 0;
boolean fire = false;

const String PHONE = "+91**********"; //use your number with country code

#define rxPin 2
#define txPin 3
SoftwareSerial sim800L(rxPin,txPin);
#define Left 4          // left sensor
#define Right 5         // right sensor
#define Forward 6       //front sensor
#define GAS_SENSOR 7    //Gas sensor
```

```
#define LM1 8           // left motor
#define LM2 9           // left motor
#define RM1 10          // right motor
#define RM2 11          // right motor
#define pump 12         //water pump
void setup()
{
  Serial.begin(115200);
  sim800L.begin(9600);
  sim800L.println("AT");
  delay(1000);
  sim800L.println("AT+CMGF=1");
  delay(1000);
  pinMode(Left, INPUT);
  pinMode(Right, INPUT);
  pinMode(Forward, INPUT);
  pinMode(GAS_SENSOR, INPUT);
  pinMode(LM1, OUTPUT);
  pinMode(LM2, OUTPUT);
  pinMode(RM1, OUTPUT);
  pinMode(RM2, OUTPUT);
  pinMode(pump, OUTPUT);
  myservo.attach(13);
  myservo.write(90);
  while(sim800L.available())
  {
      Serial.println(sim800L.readString());
  }
}
void put_off_fire()
```

```
{
  digitalWrite(LM1, HIGH);
  digitalWrite(LM2, HIGH);
  digitalWrite(RM1, HIGH);
  digitalWrite(RM2, HIGH);
  digitalWrite(pump,HIGH);
  delay(500);

  for (pos = 50; pos <= 110; pos += 1)
  {
    myservo.write(pos);
    delay(10);
  }
  for (pos = 110; pos >= 50; pos -= 1)
  {
    myservo.write(pos);
    delay(10);
  }
  digitalWrite(pump,LOW);
  myservo.write(90);
  fire=false;
}
void loop()
{
  myservo.write(90); //Sweep_Servo();
  if (digitalRead(Left) ==1 && digitalRead(Right)==1 &&
digitalRead(Forward) ==1)
  {
    delay(500);
    digitalWrite(LM1, HIGH);
```

```
  digitalWrite(LM2, HIGH);
  digitalWrite(RM1, HIGH);
  digitalWrite(RM2, HIGH);
}
else if (digitalRead(Forward) ==0)
{
  digitalWrite(LM1, HIGH);
  digitalWrite(LM2, LOW);
  digitalWrite(RM1, HIGH);
  digitalWrite(RM2, LOW);
  fire = true;
}
else if (digitalRead(Left) ==0)
{
  digitalWrite(LM1, HIGH);
  digitalWrite(LM2, LOW);
  digitalWrite(RM1, HIGH);
  digitalWrite(RM2, HIGH);
}
else if (digitalRead(Right) ==0)
{
  digitalWrite(LM1, HIGH);
  digitalWrite(LM2, HIGH);
  digitalWrite(RM1, HIGH);
  digitalWrite(RM2, LOW);
}
delay(400);//change this value to change the distance
if(digitalRead(GAS_SENSOR)== 0)
{
  Serial.println("Gas is Detected.");
```

```
    send_sms();
  }
  while (fire == true)
  {
    put_off_fire();
    Serial.println("Fire Detected.");
    make_call();
  }
}
void make_call()
{
    Serial.println("calling....");
    sim800L.println("ATD"+PHONE+";");
    delay(20000); //20 sec delay
    sim800L.println("ATH");
    delay(1000); //1 sec delay
}
void send_sms()
{
    Serial.println("sending sms....");
    delay(50);
    sim800L.print("AT+CMGF=1\r");
    delay(1000);
    sim800L.print("AT+CMGS=\""+PHONE+"\"\r");
    delay(1000);
    sim800L.print("Gas Detected");
    delay(100);
    sim800L.write(0x1A);
    delay(5000);
}
```

Estimated Cost of the Project

Sl.no	Component	Cost
1 .	Smart Car Kit	800
2.	1-channel Flame Sensor Module(3X)	150
3.	Arduino Uno	300
4.	L293 Motor Driver	300
5.	18650 3.7V battery (2x)	150
6.	LM-2596 DC-DC Buck Converter Module	60
7.	Mini Modulator Breadboard	30
8.	Water Pump (5V) ,Metre Pipe,Rocker switch	50
9.	MG90S Micro Metal Gear Servo	200
10.	SIM-800L	300
11.	Relay Module	50
	Total Cost	**2400**

Conclusion

The fire-fighting robot project based on Arduino Uno demonstrates the potential for utilizing robotics and automation to address fire hazards. The integration of various components, such as flame sensors, a motor driver, a servo motor, and a water pump,

allows the robot to detect flames, mobilize towards the fire, and extinguish it using a controlled water spray.

The project emphasizes the significance of automation and robotics in fire safety, offering a potential solution for rapid response and intervention in fire emergencies. By combining the principles of sensing, control, and actuation, the fire-fighting robot demonstrates the capability to address fire hazards in a more efficient and safer manner.

However, it is essential to continuously test and refine the system to enhance its performance, reliability, and safety. Implementing additional safety features, integrating communication modules, and considering real-world fire scenarios can further enhance the effectiveness of the fire-fighting robot.

Overall, this project showcases the application of technology to address critical challenges and contribute to fire safety, highlighting the potential for further advancements in the field of robotics and automation in emergency response systems.

References

[1] "Arduino," Wikipedia, Wikimedia Foundation, Apr. 26, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Arduino

[2] "Flame Sensor," Wikipedia, Wikimedia Foundation, Apr. 13, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Flame_sensor

[3] "LM2596," Wikipedia, Wikimedia Foundation, Feb. 14, 2023. [Online]. Available: https://en.wikipedia.org/wiki/LM2596

[4] "L293," Wikipedia, Wikimedia Foundation, Jan. 23, 2023. [Online]. Available: https://en.wikipedia.org/wiki/L293

[5] "MG90S." Wikipedia, Wikimedia Foundation, 16 Mar. 2022, https://en.wikipedia.org/wiki/MG90S

[6] "Relay." Wikipedia, Wikimedia Foundation, 10 Apr. 2023, https://en.wikipedia.org/wiki/Relay

[7] "SIM800." Wikipedia, Wikimedia Foundation, 3 Apr. 2023, https://en.wikipedia.org/wiki/SIM800