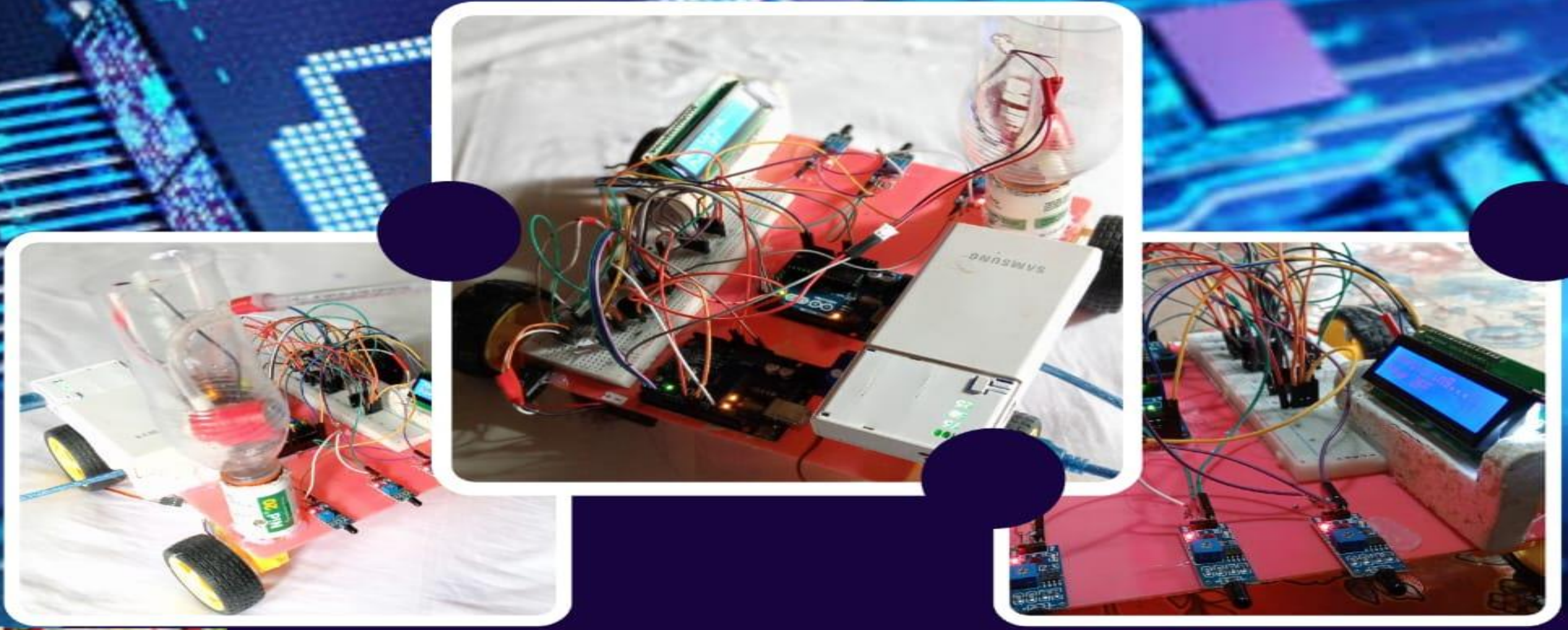
AUTOMATIC FIRE EXTGUINGUISHER ROBOT



Sirajul saliken (Department of physics)

Sirajum monir (Department of computer science and Engineering)

Automatic Fire Extinguisher Robot

Written By

Sirajum Monir
B.Sc Engr.
Department of Computer Science & Engineering
University of Chittagong

Sirajul Saliken
B.Sc Hons.
Department of Physics
University of Chittagong

Automatic Fire Extinguisher Robot

Components needed

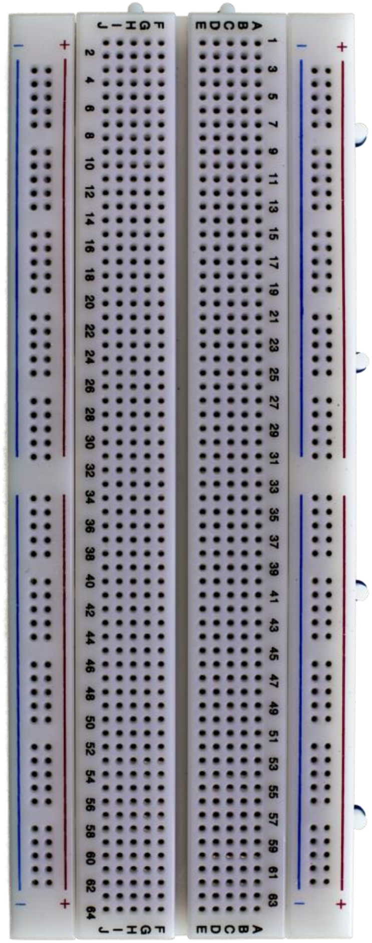
- ❑ Arduino Uno (2)
- ❑ KY026 Flame Sensor(3)
- ❑ IRFZ44N MOSFET(2)
- ❑ 1N4007 Diode(2)
- ❑ BO Motor(2)
- ❑ I2C Modular with 16*2 LCD display(1)
- ❑ 5V pump
- ❑ 2 x 18650 lithium-ion batteries (3.2V each)
- ❑ Breadboard and jumper wires
- ❑ 1k ohm register(2)
- ❑ Water Bottle
- ❑ Piezo buzzer
- ❑ USB Power Bank
- ❑ Micro Servo

Project Description

This project involves building a comprehensive fire detection and response system using an Arduino Uno The system integrates various components, including KY 026 flame sensors, an I 2 C 16 x 2 LCD, an IRFZ 44 N MOSFET, a 5 V pump, BO motors, a piezo buzzer, and a servo motor When a flame is detected by any of the flame sensors, the system performs the following actions Displays a warning message on the I 2 C 16 x 2 LCD Activates the 5 V pump to extinguish the flame Stops the BO motor immediately to prevent further fire spread Produces an audible alert using the piezo buzzer Controls the MO servo to perform a specific action (e g adjusting a nozzle or moving a barrier) The entire system is powered using a combination of the Arduino Uno and an external battery setup (two 18650 lithium ion batteries in series providing 7 4 V) The power of Arduino will provided by the USB power bank

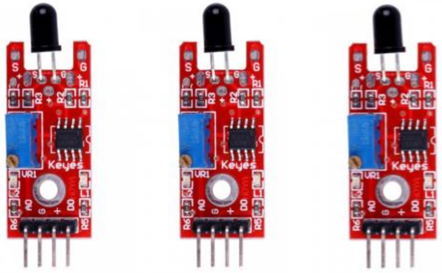
Advantages of the Project

Fire Detection and Response The system provides immediate detection and response to fire, minimizing damage and potential hazard Automated Operation The automated nature of the system ensures quick and efficient response without human intervention Versatility The system can be adapted for various applications, such as industrial safety, home security, and robotics Multi Component Integration The project demonstrates effective integration of multiple components, including sensors, actuators, and display units, enhancing its functionality Scalability Additional sensors or actuators can be easily integrated into the system, making it scalable for larger or more complex environments Educational Value The project offers a comprehensive learning experience for those interested in electronics, programming, and IoT applications Future Use and Enhancements



Bread board

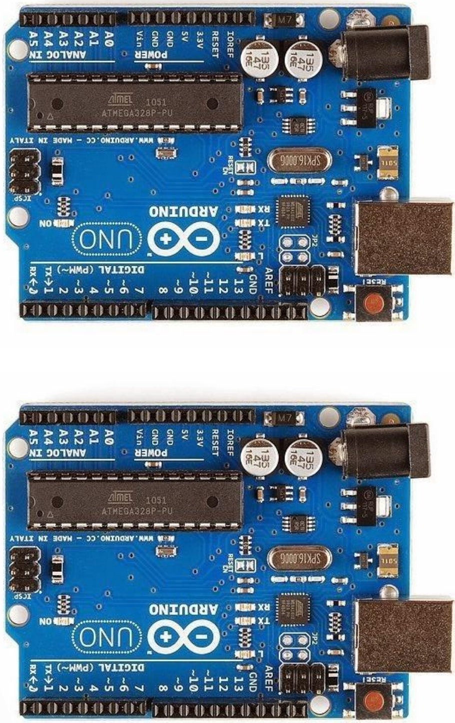
Micro Servo



KY-26 Flame Sensor

IRFZ44N Mosfet

16*2 LCD display



Arduino Uno

1k ohm register

IN4007 Diode

BO motor

18650 lithium-ion batteries

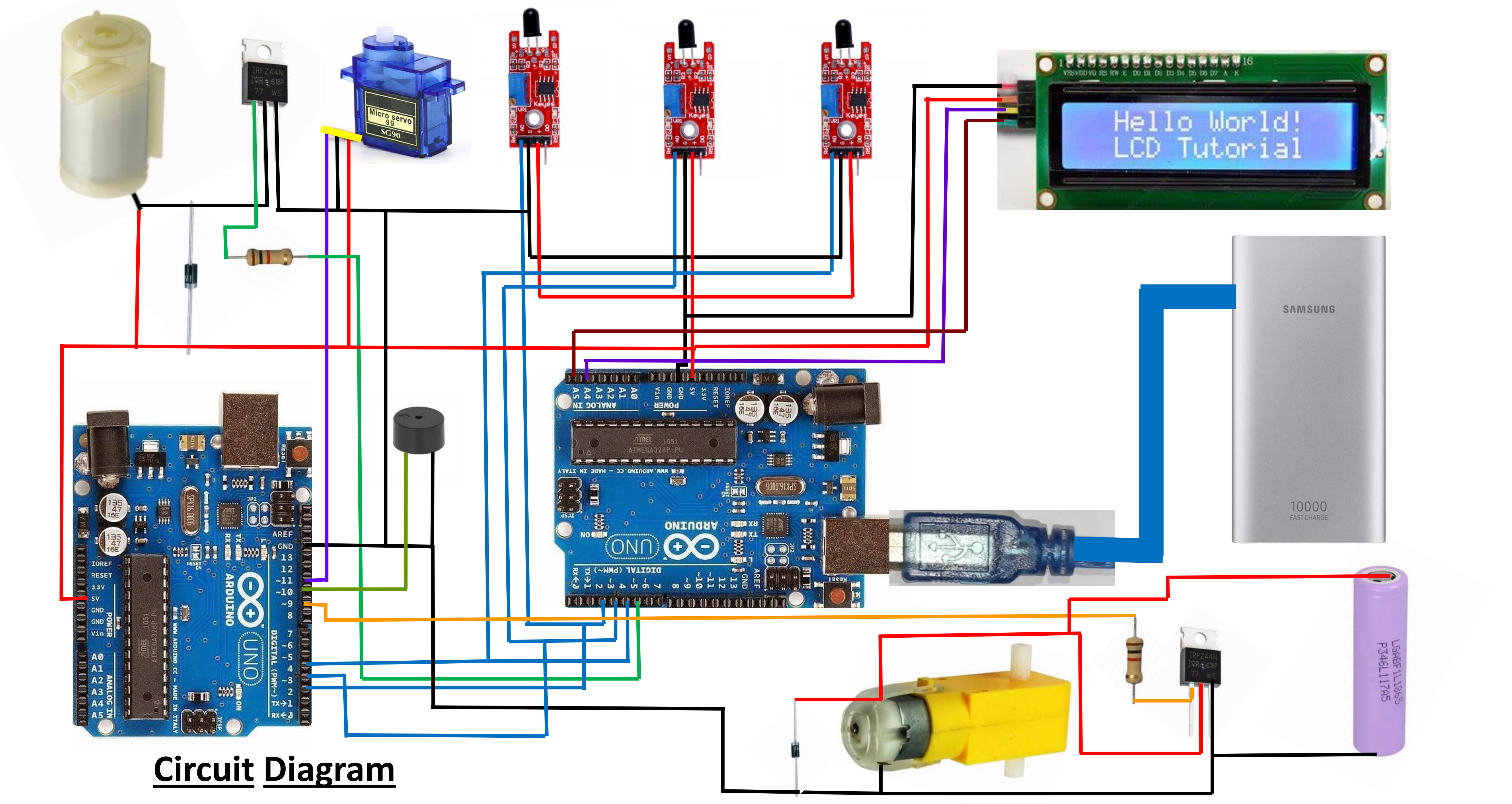
5V Pump

Piezo Buzzer

USB power bank

Jumper wires

Project Material



Circuit Diagram

Process of creating this project

Connections:

IRFZ44N MOSFET

Source (S): Connect to GND on the Arduino.
Drain (D): Connect to the negative terminal of the pump.
Gate (G): Connect to a digital pin on the Arduino (e.g., pin 5) through a 1k ohm resistor.

Pump:

Positive Terminal: Connect to the 5V pin on the Arduino.
Negative Terminal: Connect to the Drain of the MOSFET.

1N4007 Diode

Place across the pump terminals.
Cathode (marked end): Connect to the positive terminal of the pump.
Anode: Connect to the negative terminal of the pump.

KY026 Flame Sensors:

Connect each KY026 flame sensor:
VCC to 5V on the Arduino.
GND to GND on the Arduino.
Digital output (DO) to different digital pins on the Arduino (e.g., pins 2, 3, and 4).

I2C 16x2 LCD:

Connect the LCD as previously described:
GND to GND on the Arduino.
VCC to 5V on the Arduino.
SDA to A4 on the Arduino.
SCL to A5 on the Arduino.

Now for the second Arduino:

Battery Holder:

Place the two 18650 batteries in the holder, ensuring the correct orientation (positive and negative terminals).
When connected in series, the combined output will be 7.4V (3.7V + 3.7V).

BO Motor:

Positive Terminal: Connect to the positive terminal of the battery holder (7.4V).
Negative Terminal: Connect to the Source(C) of the IRFZ44N Mosfet

IRFZ44N Mosfet:

Connect to the Source(C) to the negative terminal of BO Motor

Drain: Connect to a digital pin on the Arduino (e.g., pin 9) through a 1k ohm resistor.
Gate: Connect to the GND of the battery holder.

Diode (1N4007):

Place the diode across the motor terminals to protect against back EMF.
Cathode (marked end): Connect to the positive terminal of the motor.
Anode: Connect to the negative terminal of the motor.

KY-26 Flame Sensors:

VCC: Connect to the 5V pin on the Arduino.
GND: Connect to the GND pin on the Arduino.
Digital Output (DO): Connect each flame sensor's DO to different digital pins on the Arduino (e.g., pins 2, 3, and 4).

Arduino Power:

Connect the Arduino Uno to your computer via a USB cable. This will power the Arduino.

GND Connection:

Connect the GND of the battery holder to the GND of the Arduino to ensure a common ground.

Piezo Buzzer Connections: Connect the positive terminal of the piezo buzzer to a digital pin on the Arduino (e.g., pin 10).
Connect the negative terminal of the piezo buzzer to the GND on the Arduino.

Mo Servo:

VCC (Red Wire): Connect to the 5V pin on the Arduino.
GND (Brown or Black Wire): Connect to the GND pin on the Arduino.
Signal (Orange or Yellow Wire): Connect to a digital 11 pin on the Arduino .

Water Bottle:

Take a bottle and cut the back of it, not upside down. Then attach the Mo servo to the bottle using super glue on the back.Now take the 5v motor attatch a pipe with it and put it to the bottle. The pump will sink in the water and the pipe of the motor attatch it to the Mo servo so whenever the sensor ditect a signal the pump and mo servo works together and spray the water on the flame thus extinguish the flame. MO SERVO works like a robotic arm in this project

First Arduino Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
const int flameSensor1 = 2;
const int flameSensor2 = 3;
const int flameSensor3 = 4;
const int motorPin = 5;
void setup() {
  lcd.begin(16, 2);
  lcd.backlight();
  pinMode(flameSensor1, INPUT);
  pinMode(flameSensor2, INPUT);
  pinMode(flameSensor3, INPUT);
  pinMode(motorPin, OUTPUT);
  lcd.setCursor(0, 0);
  lcd.print("Monitoring...");
  digitalWrite(motorPin, LOW);
  Serial.begin(9600);
}
```

```
void loop() {
  int sensorValue1 = digitalRead(flameSensor1);
  int sensorValue2 = digitalRead(flameSensor2);
  int sensorValue3 = digitalRead(flameSensor3);
  Serial.print("Sensor 1: ");
  Serial.print(sensorValue1);
  Serial.print(" Sensor 2: ");
  Serial.print(sensorValue2);
  Serial.print(" Sensor 3: ");
  Serial.println(sensorValue3);
  if (sensorValue1 == LOW || sensorValue2 == LOW ||
sensorValue3 == LOW) {
    digitalWrite(motorPin, HIGH);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Flame Detected!");
    lcd.setCursor(0, 1);
    lcd.print("Pump ON");
  }
else {
    digitalWrite(motorPin, LOW);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Monitoring...");
    lcd.setCursor(0, 1);
    lcd.print("Pump OFF");
  }

  delay(500);
}
```

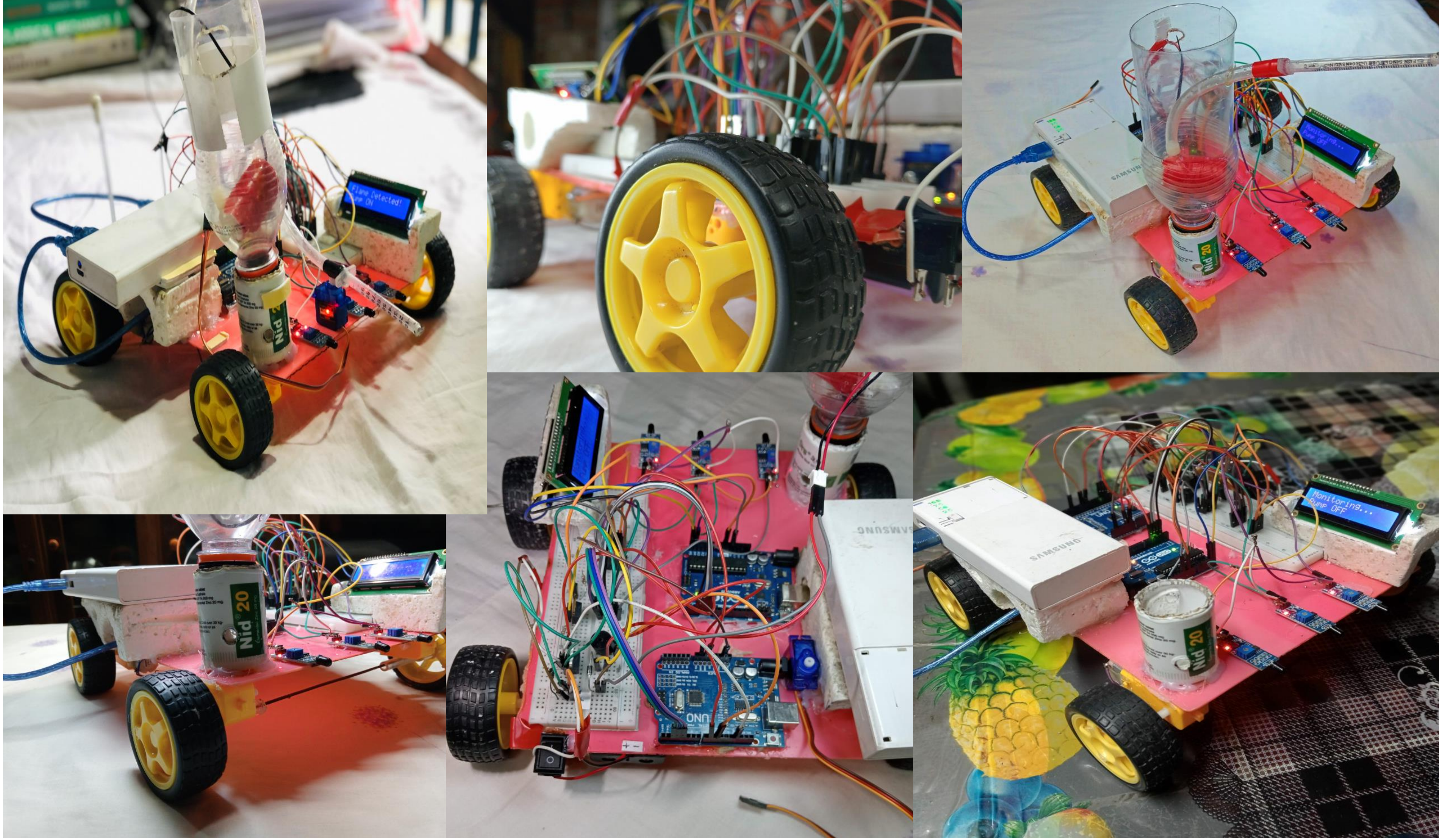
Second Arduino Code

```
#include <Servo.h>
Servo myServo;
const int motorPin = 9;
const int flameSensor1 = 2;
const int flameSensor2 = 3;
const int flameSensor3 = 4;
const int buzzerPin = 10;
const int servoPin = 11;
void setup() {
  pinMode(motorPin, OUTPUT);
  pinMode(flameSensor1, INPUT);
  pinMode(flameSensor2, INPUT);
  pinMode(flameSensor3, INPUT);
  pinMode(buzzerPin, OUTPUT);
  myServo.attach(servoPin);
  Serial.begin(9600);
}
void loop() {
  int sensorValue1 = digitalRead(flameSensor1);
  int sensorValue2 = digitalRead(flameSensor2);
  int sensorValue3 = digitalRead(flameSensor3);
Serial.print("Sensor 1: ");
  Serial.print(sensorValue1);
  Serial.print(" Sensor 2: ");
  Serial.print(sensorValue2);
  Serial.print(" Sensor 3: ");
  Serial.println(sensorValue3);
  if (sensorValue1 == LOW || sensorValue2
== LOW || sensorValue3 == LOW) {
    digitalWrite(motorPin, LOW);
    digitalWrite(buzzerPin, HIGH);
    Serial.println("Flame detected! Motor
stopped and buzzer on.");
    for (int i = 0; i < 2; i++) {
      myServo.write(45);
      delay(1000);
      myServo.write(0);
      delay(2000);
    }
  }
else {
    digitalWrite(motorPin, HIGH);
    digitalWrite(buzzerPin, LOW);
    Serial.println("No flame detected.");
  }

  delay(100);
}
```

Future Use and Enhancements Remote

Monitoring and Control Integrating IoT capabilities to allow remote monitoring and control via a smartphone or web application Data Logging and Analysis Adding a data logging feature to track sensor readings over time, helping in analyzing patterns and improving system efficiency Advanced Sensors Incorporating additional types of sensors (e g smoke sensors, temperature sensors) to enhance fire detection accuracy and reliability Battery Management System (Implementing a BMS to monitor and manage the battery health, ensuring long term reliability and safety Wireless Communication Adding wireless communication modules (e g Wi Fi, Bluetooth) for better connectivity and integration with other smart systems Enhanced User Interface Developing a more sophisticated user interface for easier interaction and real time status updates Power Optimization Optimizing the power consumption of the system to prolong battery life, especially in remote or off grid applications Integration with Sprinkler Systems Connecting the system to existing sprinkler systems for enhanced fire suppression capabilities in larger facilities Improved Actuators Using more powerful and precise actuators to handle specific tasks during fire incidents, such as closing doors or activating extinguishing agents It can widely use in fire service to prevent fire with out using human


Flame Detected!
Pump ON
Monitoring...
Pump OFF
Nid 20
SAMSUNG
UNO