PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

Project Report

Project Name: Fire Fighting Robot

Course Code: EEE-212

Course Tittle: Electrical Technology Sessional

Date of Submission: 29 October, 2024

Submitted To:

Prof. Md Naimur Rahman

Chairman

Department of Electrical & Electronics Engineering
Faculty of Computer Science & Engineering

Submitted By:

Name: Md. Al Mamun
ID No: 2102034
Registration No: 10161
Session: 2021-2022
Faculty of Computer Science & Engineering

Name: Md. Senarul Islam
ID No: 2102023
Registration No: 10150
Session: 2021-2022
Faculty of Computer Science & Engineering

Abstract

The Firefighting Robot project aims to develop a robotic system that assists firefighters in reaching difficult areas, reducing the risk to their lives, and lowering theburden on manpower. The robot will be equipped with sensors to detect fire, motors to navigate through obstacles, and a water pump to extinguish fires,It offers a costeffectivesolution to modern firefighting challenges.

List of Tools

1. **Arduino UNO R3 x1**
2. **Flame Sensor x3**
3. **BO Motor x4**
4. **Servo Motor x1**
5. **5V Relay Motor x1**
6. **Voltage Regulator x1**
7. **Mini Bread Board x1**
8. **L298 Driver x1**
9. **Li-ion Battery x2**
10. **Water Pump**
11. **Jumper Wires**
12. **Switch**

Introduction:

The Arduino-based Firefighting Robot project aims to develop an autonomous robot capable of detecting and extinguishing fires in indoor environments1. The robot utilizes Arduino Uno as its central control unit, integrating fire sensors to detect flames, motor drivers to navigate towards the fire, and a water pump to extinguish it21. Additionally, a GSM module is included to send real-time SMS alerts to designated phone numbers, ensuring timely notification of fire incidents1. This project enhances fire safety by automating fire detection and response, minimizing risks, and providing immediate communication to relevant authorities1.

Objectives:

1. **Detect and locate indoor fires automatically.**
2. **To enhance firefighting safety.**
3. **To reduce manpower burden.**
4. **To improve efficiency.**
5. **Navigate towards the source of the fire accurately.**
6. **Extinguish the detected fire accurately.**

Description of Tools

1. ArduinoUNO: A microcontroller board used as the brain of your project, enabling control and interface with various sensors and actuators.

2. FlameSensor:
   A device that detects flames and fire through infrared wavelengths emitted by the flame.

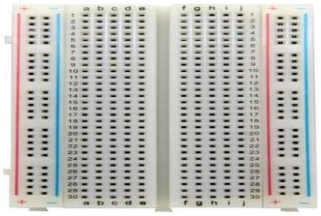
3. BO Motor: Battery Operated DC motor commonly used in low-power applications and robotics.

4. Servo Motor: A motor with feedback control, allowing precise control of angular position.

5. **5V Relay: An electrically operated switch that allows circuits to be controlled by low-power signals.**

6. **Voltage Regulator: A device that maintains a constant voltage level to protect circuits from voltage spikes.**

7. **Mini Breadboard**: A small, reusable board for prototyping electronic circuits without soldering.



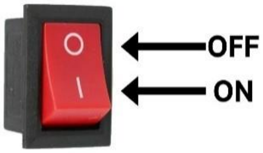
8. **L298 Driver**: A dual H-bridge motor driver that allows you to control the speed and direction of two DC motors.

9. **Li-ion Battery**: A rechargeable battery providing high energy density, often used to power portable electronics.

10. **MiniWater Pump**: A device that moves water using a mechanical action, used in firefighting robots to extinguish fires.

11. **Jumper Wires:** Connecting wires used for making temporary circuits on breadboards or other prototyping setups.

12. **Switch:** A device used to open or close an electrical circuit.



Procedure

Step1:**Gather Components**: Collect Arduino UNO, sensors, motors, water pump, and power supply.

Step-2: **Mount Flame Sensors**: Attach flame sensors at the front of the chassis.

Step-3: **Install Motors**: Fix motors to the chassis and connect to the L298 driver.

Step-4: **Attach Water Pump**: Connect the water pump to the tank and Arduino.

Step-5: **Connect Servo Motor**: Attach the servo motor to control the nozzle direction.

Step-6: **Set Up Power Supply**: Connect Li-ion battery to power the components.

Step-7: **Wire Flame Sensors**: Connect sensors to Arduino's digital pins.

Step-8: **Connect Motors to L298**: Wire motors to L298 driver and Arduino.

Step-9: **Wire Water Pump**: Connect water pump to a digital pin on the Arduino.

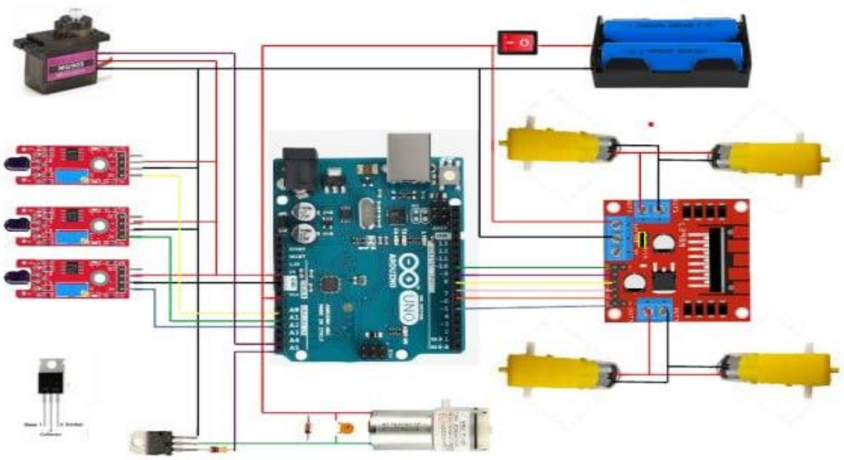
Step-10: **Upload Code**: Program Arduino with fire detection and control logic.

Step11:**Test Robot**: Power on and verify functionality for fire detection and extinguishing .

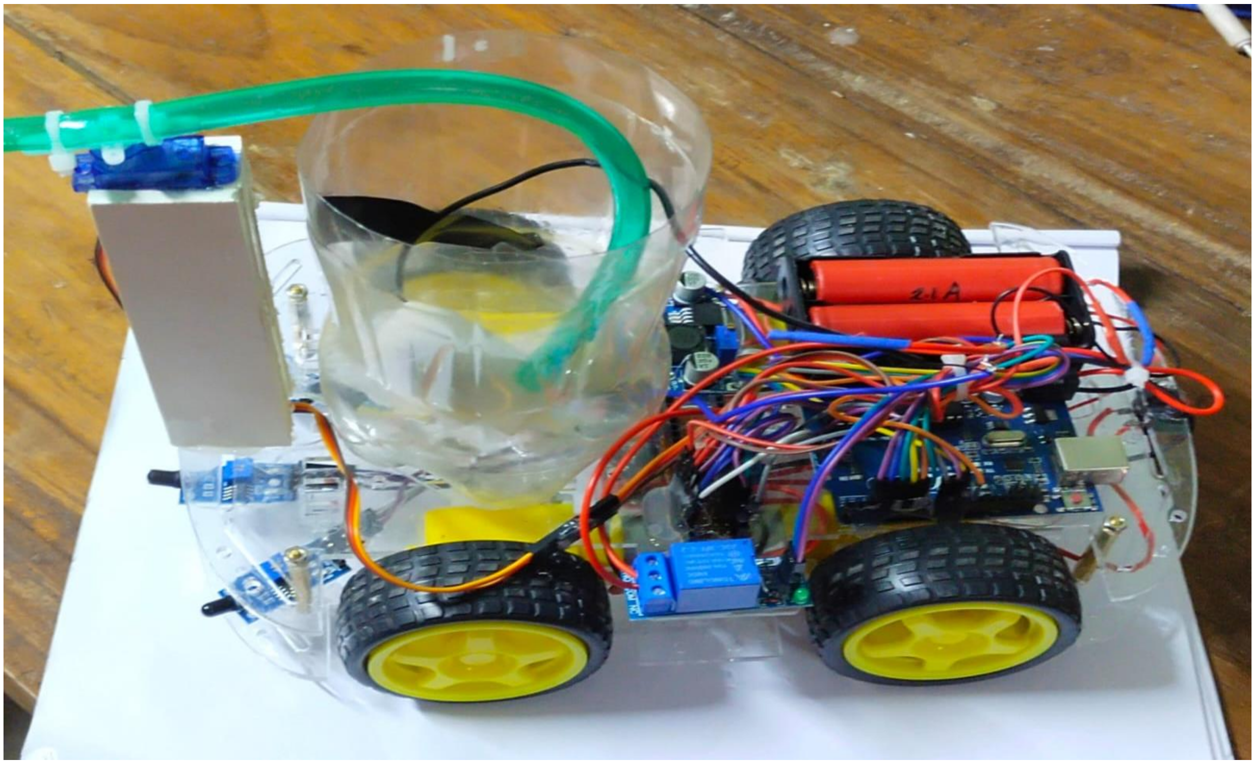
Circuit Diagram

FIRE FIGHTER ROBOT

- **Circuit Diagram**



After finishing the Project Structure



Coding;

```
#define enA 10//Enable1 L298 Pin enA
#define in1 9 //Motor1 L298 Pin in1
#define in2 8 //Motor1 L298 Pin in2
#define in3 7 //Motor2 L298 Pin in3
#define in4 6 //Motor2 L298 Pin in4
#define enB 5 //Enable2 L298 Pin enB
#define ir_R A0
#define ir_F A1
#define ir_L A2
#define servo A4
#define pump A5
int Speed = 160; // Write The Duty Cycle 0 to 255 Enable for Motor Speed
int s1, s2, s3;
void setup(){ // put your setup code here, to run once
Serial.begin(9600); // start serial communication at 9600bps
pinMode(ir_R, INPUT);// declare fire sensor pin as input
pinMode(ir_F, INPUT);// declare fire sensor pin as input
pinMode(ir_L, INPUT);// declare fire sensor pin as input
pinMode(enA, OUTPUT); // declare as output for L298 Pin enA
pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4
pinMode(enB, OUTPUT); // declare as output for L298 Pin enB
pinMode(servo, OUTPUT);
pinMode(pump, OUTPUT);
for (int angle = 90; angle <= 140; angle += 5) {
```

```
servoPulse(servo, angle); }
for (int angle = 140; angle >= 40; angle -= 5) {
servoPulse(servo, angle); }
for (int angle = 40; angle <= 95; angle += 5) {
servoPulse(servo, angle); }
analogWrite(enA, Speed); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed
analogWrite(enB, Speed); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed

delay(500);
}
void loop(){
s1 = analogRead(ir_R);
s2 = analogRead(ir_F);
s3 = analogRead(ir_L);

//==========================================================================
// Auto Control
//==========================================================================
Serial.print(s1);
Serial.print("\t");
Serial.print(s2);
Serial.print("\t");
Serial.println(s3);
delay(50);
if(s1<250){
Stop();
digitalWrite(pump, 1);
for(int angle = 90; angle >= 40; angle -= 3){
servoPulse(servo, angle);
}
```

```
for(int angle = 40; angle <= 90; angle += 3){

servoPulse(servo, angle);

}

}

else if(s2<350){

Stop();

digitalWrite(pump, 1);

for(int angle = 90; angle <= 140; angle += 3){

servoPulse(servo, angle);

}

for(int angle = 140; angle >= 40; angle -= 3){

servoPulse(servo, angle);

}

for(int angle = 40; angle <= 90; angle += 3){

servoPulse(servo, angle);

}

}

else if(s3<250){

Stop();

digitalWrite(pump, 1);

for(int angle = 90; angle <= 140; angle += 3){

servoPulse(servo, angle);

}

for(int angle = 140; angle >= 90; angle -= 3){

servoPulse(servo, angle);

}

}

else if(s1>=251 && s1<=700){

digitalWrite(pump, 0);
```

```
backword();

delay(100);

turnRight();

delay(200);

}

else if(s2>=251 && s2<=800){

digitalWrite(pump, 0);

forword();

}

else if(s3>=251 && s3<=700){

digitalWrite(pump, 0);

backword();

delay(100);

turnLeft();

delay(200);

}else{

digitalWrite(pump, 0);

Stop();

}

delay(10);

}

void servoPulse (int pin, int angle){

int pwm = (angle*11) + 500; // Convert angle to microseconds

digitalWrite(pin, HIGH);

delayMicroseconds(pwm);

digitalWrite(pin, LOW);

delay(50); // Refresh cycle of servo

}

void forword(){ //forword
```

```
digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, HIGH); //Left Motor forword Pin

}

void backword(){ //backword

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}

void turnRight(){ //turnRight

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, HIGH); //Left Motor forword Pin

}

void turnLeft(){ //turnLeft

digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}

void Stop(){ //stop

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}
```