Misr University for Science & Technology

College of Information Technology

جامعة مصر للعلوم والتكنولوجيا

كلية تكنولوجيا المعلومات

Fire Fighting Robot: Project Report

Prepared By

Student Name Student ID

Abdulrahmn Mohamed
200038291

Under Supervision

Name of Doctor: Dr. Mohammed Abdelrahman Abdelrahman Marey

Submitted to:

T.A. Ahmed Lotfi

1. Project introduction

This document details the design, development, and implementation of an autonomous "Fire Fighter Robot." The primary objective of this project is to create a mobile robotic system capable of automatically detecting a fire, navigating to its source, and extinguishing it. The system is built upon the Arduino platform, utilizing a 4-wheel drive chassis, a multi-sensor array for detection, and a 2-axis servo-controlled arm for precise extinguishing.

2. Problem Statement (Definition)

Fires in residential, industrial, and high-risk environments pose a significant and immediate threat to human life, property, and the environment. The critical period between ignition and the arrival of human firefighters is when a small, controllable fire can escalate into an uncontrollable blaze. Furthermore, first responders are often required to enter these hazardous, smoke-filled, and structurally compromised environments, placing their lives at great risk. There is a clear need for an automated system that can provide an immediate, on-site response to mitigate a fire's spread and reduce the danger to human personnel.

3. The Proposed solution

The proposed solution is an autonomous robot prototype. Its operation is based on the components listed in the wiring table and is divided into the following steps:

1. Detection & Navigation: The robot uses three chassis-mounted flame sensors (Left: A0, Front: A1, Right: A2) to continuously scan its environment. The system compares the analog readings from these sensors to determine the fire's direction and navigates (moves forward, turns left/right) to center the flame.

2. Targeting: Once the robot is in close proximity (determined by the strength of the front sensor A1), it stops navigating. It then activates its 2-axis robotic arm, which is controlled by two servo motors: ARM_BASE (D12) and ARM_SERVO (D11).

3. Verification & Extinguishing: A fourth flame sensor (Arm_FLAME SENSOR) is mounted on the arm itself and connected to pin A3. As the arm sweeps the area, this sensor provides precise confirmation of the fire's location. When this sensor detects the fire, it signals the Arduino to activate the PUMP (connected to pin D2) via a relay module, extinguishing the fire only where it is precisely detected.

4. Pros and Cons (advantage and disadvantage)

Pros (Advantages):

- **Human Safety:** It removes the need for firefighters to enter the immediate danger zone, reducing the risk of injury or death.
- **Rapid Response:** The robot can be stationed in a high-risk area 24/7, providing an immediate response seconds after ignition.
- **Precision Extinguishing:** The use of a 2-servo arm with its own dedicated sensor (A3) prevents water waste and only targets the confirmed flame, reducing collateral water damage.
- **Autonomy:** The robot operates without human intervention after activation, based on its sensor inputs.

Cons (Disadvantages):

- **Sensor Limitations:** Flame sensors are line-of-sight and can be falsely triggered by other strong IR sources (like direct sunlight). They also cannot see through heavy smoke.
- **Navigation Limitations:** The current design (based on the provided wiring table) does not include obstacle avoidance sensors (like ultrasonic), so it assumes a clear path.
- **Mobility:** The 4-wheel chassis is limited to flat, indoor surfaces and cannot navigate stairs or rubble.
- **Scale:** This is a small-scale prototype with a limited water supply and battery, suitable only for small, incipient-stage fires.

5.Tools And Software Used

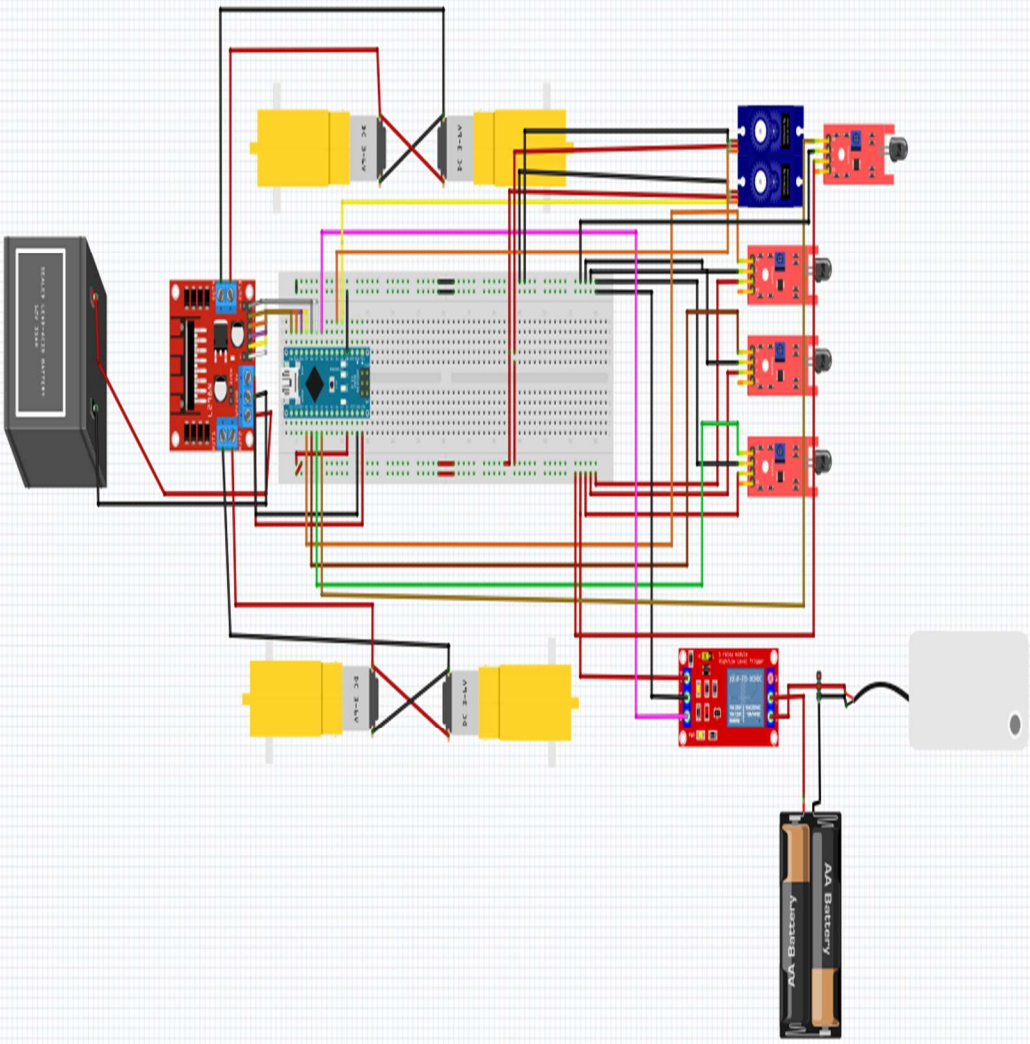
Software Tools:

- **Arduino IDE:** Used for writing, compiling, and uploading the C++ code to the microcontroller.
- **Fritzing:** The software likely used to create the circuit schematic diagram.
- **C++ (Arduino):** The programming language used for the robot's logic.
- **Servo.h Library:** The standard Arduino library required to control the ARM_SERVO and ARM_BASE.

Hardware Tools

- **Arduino Nano** (Microcontroller)
- **L298N Motor Driver**
- **4x DC Motors** & Chassis
- **4x Flame Sensors** (3 for navigation, 1 for arm)
- **2x Servo Motors** (for Arm and Base)
- **5V Relay Module**
- **DC Water Pump**
- **Breadboard, Jumper Wires, and multiple Power Supplies**

6 Circuit design



7. hardware Connections

FIRE FIHGTER ROBOT			
part	pin part	pin	wier color
L298	ENA	5	WHITE
	ENB	6	GRAY
	IN1	7	BROWN
	IN2	8	PURPLE
	IN3	9	ORANGE
	IN4	10	BLUE
L_FLAME SENSOR	GND	GND	GRAY
	5V	5V	WHITE
	A	A0	PURPLE
R_FLAME SENSOR	GND	GND	GREN
	5V	5V	BLUE
	A	A1	YELLOW
R_FLAME SENSOR	GND	GND	BLACK
	5V	5V	RED
	A	A2	ORANGE
Arm_FLAME SENSOR	GND	GND	BLACK
	5V	5V	RED
	A	A3	PURPLE
PUMP	DC-	GND	BLACK
	IN	D2	BROWN
ARM_SERVO	GND	GND	RED
	5V	5V	BLACK
	IN	D11	BROWN
ARM_BASE	GND	GND	GRAY
	5V	5V	WHITE
	IN	D12	PURPLE

8. Code

The code

```cpp
#include <Servo.h>

//        البنات
#define ENA 5
#define ENB 6
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 10

#define FLAME_LEFT A0
#define FLAME_FRONT A1
#define FLAME_RIGHT A2
#define FLAME_PUMP A3

#define PUMP 2
#define SERVO_ARM 11
#define SERVO_BASE 12

Servo servoArm;
Servo servoBase;


#define FIRE_NEAR_THRESHOLD 80
#define FIRE_DETECT_THRESHOLD 400
#define FIRE_PUMP_THRESHOLD 300

void setup() {
    Serial.begin(9600);


    pinMode(ENA, OUTPUT);
    pinMode(ENB, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);


    pinMode(FLAME_LEFT, INPUT);
    pinMode(FLAME_FRONT, INPUT);
    pinMode(FLAME_RIGHT, INPUT);
    pinMode(FLAME_PUMP, INPUT);


    pinMode(PUMP, OUTPUT);
    digitalWrite(PUMP, LOW);
    pinMode(LED_RED, OUTPUT);
    pinMode(LED_BLUE, OUTPUT);
```

```
50
51
52      servoArm.attach(SERVO_ARM);
53      servoBase.attach(SERVO_BASE);
54      servoArm.write(90);
55      servoBase.write(90);
56
57      Serial.println(" Fire Fighting Robot Ready ");
58  }
59
60  void loop() {
61      int leftVal = analogRead(FLAME_LEFT);
62      int frontVal = analogRead(FLAME_FRONT);
63      int rightVal = analogRead(FLAME_RIGHT);
64      int pumpVal = analogRead(FLAME_PUMP);
65
66
67      if (leftVal < FIRE_DETECT_THRESHOLD || frontVal <                    ↵
          FIRE_DETECT_THRESHOLD || rightVal < FIRE_DETECT_THRESHOLD) {
68          moveTowardFlame(leftVal, frontVal, rightVal);
69          digitalWrite(LED_BLUE, HIGH);
70          digitalWrite(LED_RED, LOW);
71      }
72
73
74      if (frontVal < FIRE_NEAR_THRESHOLD) {
75          stopMoving();
76          digitalWrite(LED_BLUE, LOW);
77          digitalWrite(LED_RED, HIGH);
78
79
80          servoArm.write(130);
81          delay(300);
82
83          // القاعدة تتحرك يمين ويسار
84          for (int pos = 60; pos <= 120; pos += 5) {
85              servoBase.write(pos);
86              delay(60);
87          }
88          for (int pos = 120; pos >= 60; pos -= 5) {
89              servoBase.write(pos);
90              delay(60);
91          }
92
93          // يرجع الذراع للوضع الطبيعي
94          servoArm.write(90);
95          servoBase.write(90);
96      }
97
```

```
 98
 99      if (pumpVal < FIRE_PUMP_THRESHOLD) {
100          digitalWrite(PUMP, HIGH);
101      }
102      else {
103          digitalWrite(PUMP, LOW);
104      }
105
106      delay(200);
107  }
108
109
110  void moveForward() {
111      digitalWrite(IN1, LOW);
112      digitalWrite(IN2, HIGH);
113      digitalWrite(IN3, HIGH);
114      digitalWrite(IN4, LOW);
115      analogWrite(ENA, 150);
116      analogWrite(ENB, 150);
117  }
118
119  void turnRight() {
120      digitalWrite(IN1, LOW);
121      digitalWrite(IN2, HIGH);
122      digitalWrite(IN3, LOW);
123      digitalWrite(IN4, LOW);
124      analogWrite(ENA, 150);
125      analogWrite(ENB, 150);
126  }
127
128  void turnLeft() {
129      digitalWrite(IN1, LOW);
130      digitalWrite(IN2, LOW);
131      digitalWrite(IN3, HIGH);
132      digitalWrite(IN4, LOW);
133      analogWrite(ENA, 150);
134      analogWrite(ENB, 150);
135  }
136
137  void stopMoving() {
138      digitalWrite(IN1, LOW);
139      digitalWrite(IN2, LOW);
140      digitalWrite(IN3, LOW);
141      digitalWrite(IN4, LOW);
142  }
143
144  void moveTowardFlame(int leftVal, int frontVal, int rightVal) {
145      if (frontVal <= leftVal && frontVal <= rightVal) {
146          moveForward();
```

```
147     }
148     else if (leftVal < rightVal) {
149         turnLeft();
150     }
151     else {
152         turnRight();
153     }
154 }
155
```