****

**CSE461: Introduction to Robotics**

**Fire Fighting Robot**

**Semester:** Spring 2024

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Section**  **No** | **Content** | **Page No** |
| 1 | Introduction | 3 |
| 2 | List of Components | 3 |
| 3 | Schematic Diagram | 4 |
| 4 | User Manual | 5 |
| 5 | Code | 7 |
| 6 | Demo Video Link | 10 |
| 7 | Conclusion | 10 |
| 8 | References | 11 |

1. **Introduction**

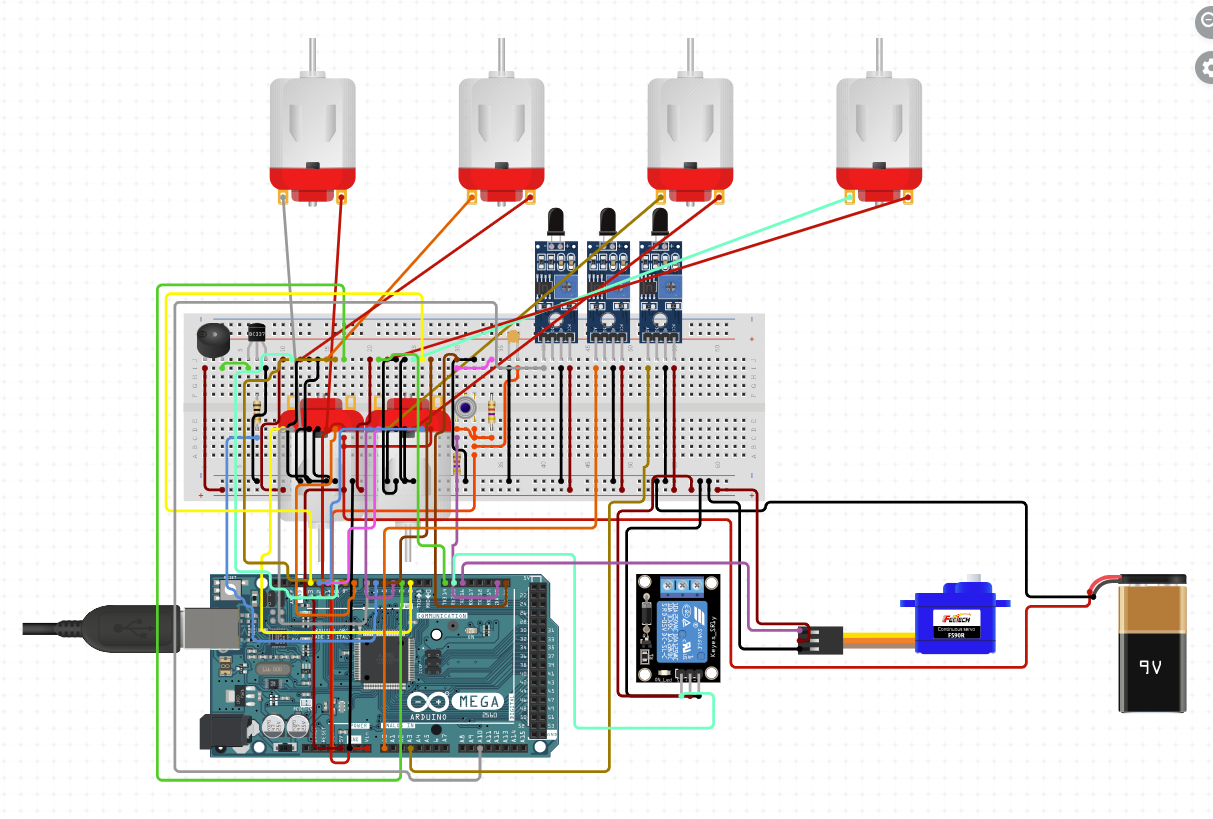
The integration of robotics into emergency response systems significantly boosts both safety and efficiency, particularly in the context of fire emergencies. Our project introduces a Fire Fighting Robot, utilizing the Arduino Uno - a microcontroller celebrated for its adaptability across diverse applications. This robot is specifically designed to autonomously detect and extinguish fires, thereby minimizing human risk and enhancing the speed of emergency responses.

This prototype not only demonstrates the effective incorporation of technology into vital emergency services but also showcases the potential to redefine standards in automated fire safety solutions. As we refine and enhance this model, it holds the promise of providing substantial benefits to public safety. By reducing the need for direct human intervention in hazardous situations, it paves the way for broader adoption and further development of robotics in critical response scenarios, ultimately aiming to safeguard lives and property more efficiently.

1. **List of Components**

The following components were meticulously selected to build and operate our Fire Fighting Robot, ensuring optimal performance and reliability:

1. **Arduino Uno R3**: Microcontroller board that serves as the brain of our robot.
2. **4WD Smart Robot Chassis Kit**: Provides the physical structure and mobility for the robot.
3. **5V Relay Module**: Controls high-voltage water pump using low-voltage signals from Arduino.
4. **Flame Sensor Fire Detection Modules**: Detect flames and trigger appropriate actions.
5. **MLX90614**: Non-contact infrared temperature sensor used for fire detection.
6. **Ultrasonic Sonar Sensor**: Detects obstacles and helps in navigation (will be updated in future upgrades).
7. **L298N H-Bridge Dual Motor Driver**: Drives DC motors connected to the robot's wheels for the robot’s movements.
8. **Servo Motor Micro SG90:** Provides precise angular control for various mechanisms such as directing the water pump.
9. **Submersible 3V Mini DC Water Pump:** Used for dispensing water to extinguish fires.
10. **9V Battery:** A power source for the Arduino and various components.
11. **Battery Holder:** Holds the 9V battery securely.
12. **Breadboards:** Used for prototyping and connecting electronic components.
13. **Breadboard Jumper Wire Sets:** Connect components on the breadboard for testing and prototyping.
14. **Buzzer:** Serves as an indicator of fire with a warning sound.
15. **Schematic Diagram**

****

With the help of particular pin connections, a firefighting robot with numerous sensors and output devices is controlled by this Arduino program. Three flame sensors, linked to pins 8 (Left), 9 (Right), and 10 (Forward), are used by the robot to sense the presence and direction of fire. Four motors are involved in the movement control: the left motors are driven forward and backwards by pins 2 (LM1) and 3 (LM2), while the right motors are similarly controlled by pins 4 (RM1) and 5 (RM2). A servo motor connected to pin 11 can change a water nozzle's angle to distribute water efficiently. It can be swept from 50 to 130 degrees. Upon detecting a fire, the water pump, which is linked to pin 6, is triggered to release water. Finally, we have also kept a Passive Buzzer used for providing audio feedback connected to Pin 7. All the ground and VCC connections were given accordingly. The code uses infrared sensors to detect obstacles or fires. Based on sensor readings, the robot adjusts the motor speeds to navigate around obstacles. If a fire is detected, the robot activates the water pump using the servo motor to extinguish the fire. Additionally, a passive buzzer provides audio feedback during operation. By using separate pin assignments to integrate sensors, motors, and servo control, the robot can effectively navigate towards and suppress fires on its own.

1. **User Manual**

**Initial Setup:**

* Ensure the battery is fully charged and properly installed.
* Fill the water reservoir connected to the submersible pump.
* Position the robot in a starting location conducive to patrolling the designated area

**Operating the Robot:**

* Flame sensors actively scan for signs of fire. If a fire is detected, the robot will navigate towards it.
* Once positioned optimally near the fire, the robot will activate the water pump. The servo motor adjusts the nozzle to effectively target and extinguish the fire.

**Maintenance:**

* **Regular Checks:** check the battery levels, water reservoir, and sensor functionality weekly.
* **Cleaning:** Clean the chassis and sensors to avoid dust accumulation that could impair functionality
* **Storage:** Store the robot in a dry, cool place away from direct sunlight when not in use.

**Safety Precautions:**

* Ensure that water does not come into contact with the circuit to prevent short circuits and potential damage.
* Always supervise the robot during operation to ensure it functions correctly.

**Troubleshooting:**

* **The robot does not start:** Ensure the battery is properly charged and installed.
* **The robot does not detect fire:** Check the alignment and cleanliness of the flame sensor.
* **Water pump not functioning:** Verify that the water reservoir is filled and check the relay and pump connections

1. **Code**

#include <Servo.h> // Included servo.h library

Servo myservo; // Created a servo object

int pos = 0; // pos variable stores servo’s position

boolean fire = false; // boolean variable fire tracks if there's fire

// Defined PINs for each sensor, motor and buzzer

#define Left 8

#define Right 9

#define Forward 10

#define LM1 2

#define LM2 3

#define RM1 4

#define RM2 5

#define pump 6

#define buzzerPin 7

void setup()

{

// Set pin modes as input or output (sensors: inputs, and motors, pump, and buzzer: outputs)

pinMode(Left, INPUT);

pinMode(Right, INPUT);

pinMode(Forward, INPUT);

pinMode(LM1, OUTPUT);

pinMode(LM2, OUTPUT);

pinMode(RM1, OUTPUT);

pinMode(RM2, OUTPUT);

pinMode(pump, OUTPUT);

pinMode(buzzerPin, OUTPUT);

// Attached servo to pin 11 and set initial position

myservo.attach(11);

myservo.write(90);

}

// Function to put off fire

void put\_off\_fire()

{

// Stop the motors

digitalWrite(LM1, HIGH);

digitalWrite(LM2, HIGH);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, HIGH);

// Activate the water pump to extinguish the fire

digitalWrite(pump, HIGH);

delay(1000); //Wait for 1 second while water is sprayed

// Sweep the servo arm back and forth to simulate extinguishing the fire

for (pos = 50; pos <= 130; pos += 1) {

myservo.write(pos);

delay(10);

}

for (pos = 130; pos >= 50; pos -= 1) {

myservo.write(pos);

delay(10);

}

// Turn off the water pump and reset the servo arm to its neutral position

digitalWrite(pump,LOW);

myservo.write(90);

fire=false; // Reset fire status flag

}

// Function to make the buzzer beep

void beep()

{

tone(buzzerPin, 1000); // Play a tone of frequency 1000 Hz

delay(1000);

tone(buzzerPin, 700); // Play a tone of frequency 700 Hz

delay(1000); // We can adjust the delay for desired beep duration

noTone(buzzerPin); // Stop the tone

delay(100); // We can adjust the delay for a desired interval between beeps

}

void loop()

{

myservo.write(90); // Center the servo

// Check sensor readings to determine movement

if (digitalRead(Left) == 1 && digitalRead(Right) == 1 && digitalRead(Forward) == 1)

{

// If all sensors detect no fire, stop the robot

digitalWrite(LM1, HIGH);

digitalWrite(LM2, HIGH);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, HIGH);

}

else if (digitalRead(Forward) == 0)

{

// If the front sensor detects a fire, move forward and set fire flag

digitalWrite(LM1, HIGH);

digitalWrite(LM2, LOW);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, LOW);

fire = true;

}

else if (digitalRead(Left) == 0)

{

// If the left sensor detects an obstacle, turn left and beep

digitalWrite(LM1, LOW);

digitalWrite(LM2, LOW);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, LOW);

beep();

}

else if (digitalRead(Right) == 0)

{

// If the right sensor detects fire, turn right and beep

digitalWrite(LM1, HIGH);

digitalWrite(LM2, LOW);

digitalWrite(RM1, LOW);

digitalWrite(RM2, LOW);

beep();

}

// If the fire flag is set, beep and put off the fire

if (fire == true)

{

beep();

put\_off\_fire();

}

delay(100); // We can adjust this value to adjust the distance between movements

}

1. **Demo Video Link**

[CSE461 Project Group2: Autonomous Fire Fighting Robot](https://www.youtube.com/watch?v=B0pN9eqLqX4)

1. **Conclusion**

In summary, the incorporation of robotics, demonstrated by our Fire Fighting Robot that makes use of the Arduino Uno microcontroller, is a revolutionary advancement in emergency response systems, especially when it comes to reducing the severity of fire incidents. This prototype greatly increases the effectiveness of emergency responses while also reducing human risk through autonomous detection and extinguishing capabilities. This model has the ability to redefine norms in automated fire safety solutions, and this is becoming more and more clear as we continue to improve and polish it. Our Fire Fighting Robot creates the foundation for wider use and continued growth of robots in critical response scenarios by minimizing the need for direct human intervention in dangerous situations. In the end, our project has the potential to significantly improve public safety by more effectively protecting people and their property than in the past. The cooperation of human creativity and technology becomes more and more important as we manage the complexity of contemporary emergency management, and our Fire Fighting Robot is a testament to the ability of innovation to serve human safety and well-being.

In the future, we intend to make this model more robust by adding in more accurate controls and cutting-edge sensing technologies to make it even more precise, opening the door to even more dependable and effective autonomous firefighting systems. Incorporating obstacle avoidance algorithms will also greatly improve the robot's capacity to safely navigate challenging terrain. Furthermore, the incorporation of heat-resistant parts and waterproofing will allow the robot to be able to tolerate higher temperatures and unfavourable conditions, guaranteeing its operation in difficult fire scenarios. These improvements will not only make the robot more efficient at fighting fires but also make it more resilient and long-lasting, which will improve the ability of autonomous emergency response systems to better safeguard people and property. Through this project, we hope to inspire many more people to work on and go even further in creating a more polished product.

1. **References**
2. [**https://www.youtube.com/watch?v=4CpuhDSBuIU**](https://www.youtube.com/watch?v=4CpuhDSBuIU)
3. [**https://www.youtube.com/watch?v=nMdZ2xcrKRk&t=80s**](https://www.youtube.com/watch?v=nMdZ2xcrKRk&t=80s)
4. [**Fire Fighting Robot using Arduino by hobby project (youtube.com)**](https://www.youtube.com/watch?v=yiTJZJmxDmA&ab_channel=HobbyProject)
5. All components were purchased from: [**https://techshopbd.com/**](https://techshopbd.com/)