CMP 438 Project 2 - Fire Fighting Robot

Mahamadou Nimaga
Edward Serrano-Rodriguez
Muhaimin Ali

Repo Link:

https://github.com/Shumel-coder/FireFightingRobot/blob/main/FireFightingRobot.ino

Demo Video:

https://youtube.com/shorts/Bu0i_9VdLD8?feature=share

Objective

My group's objective for this project was to build a robot equipped with sensors that can detect a small flame and extinguish it using water or a fan. We built the robot using infrared 3 flame sensor modules, Arduino Uno R3, two motor drivers, a car chassis, two wheels, 2 DC motors, a servo, a submersible DC pump, a 9V battery with an adapter, and wires. We had our robot sense which direction a fire had been detected using 3 sensors on the right, left, and in front. If the right sensor picks up a flame reading the robot will turn right. If the left sensor senses a flame the robot will turn left. If the flame is detected in front of the robot it will move forward, stop, and activate the DC pump while the servo moves the water spray from left to right in a sweeping motion to put out the flame.

Component List

- **1 Arduino Uno R3**
- **2 DC Motors**
- **2 Wheels**

- **1 Car chassis**
- **1 L298N Motor Driver**
- **1 L293D Mini Stepper Motor Drive Shield Module**
- **3 IR Flame Sensor Modules**
- **SG90 Servo Motor**
- **1 DC 3-5V Mini Submersible Water Pump**
- **1 Clear Vinyl tube**
- **Soda Can/Water container**
- **Mini breadboard**
- **Wires**
- **Hot Glue Gun**
- **Electric Tape**
- **2-sided Tape**
- **Matches (For testing the robot)**

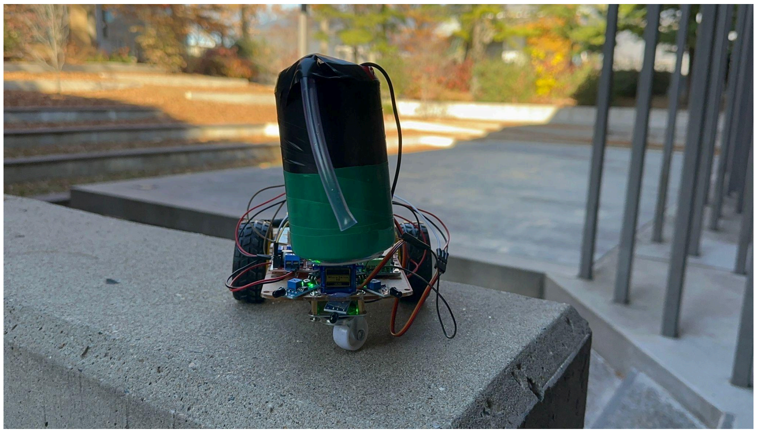
Assembly

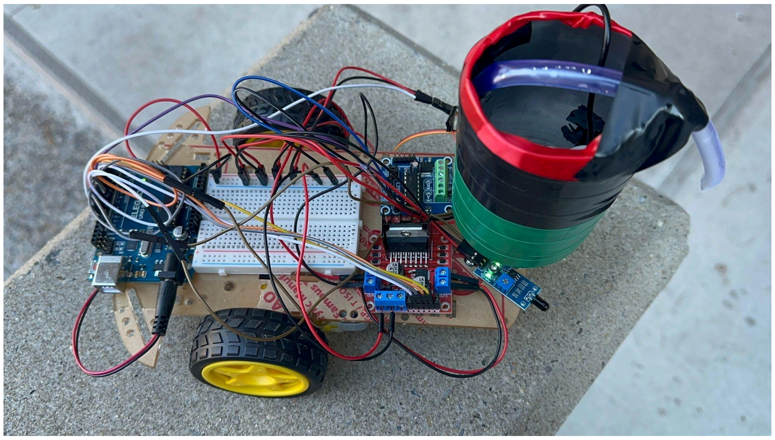
For the assembly of the robot, first, we put the wheels on the car chassis and set up the DC motors with the wheels. Then we continued to mount the rest of the pieces. The two pieces we started with mounting were the IR sensors and servo. For the IR sensors, we used double-sided tape to tape them to each direction, straight, left, and right, in the front of the robot. Then near the nose of the chassis, we hot-glued the servo motor. Since the servo will be moving we wanted to make sure this piece was mounted very securely.

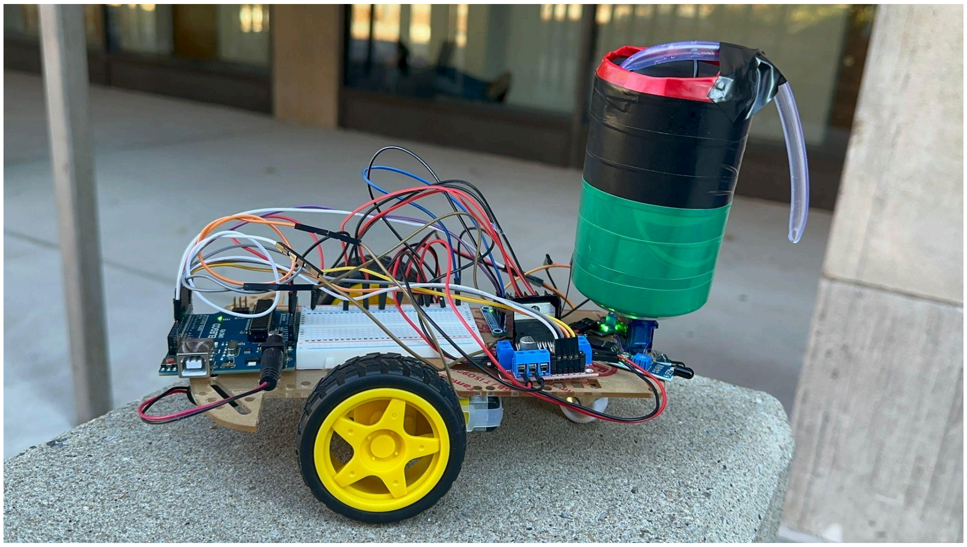
Next, we arranged both the motor drivers, the Arduino board, and the breadboard on the chassis so they could fit and easily be connected. We then double-sided taped these pieces onto the chassis. We also double-sided taped a 9V battery to the back-underside of the chassis and connected the adapter to the battery and Arduino. Afterward, we connected the wires setting up the power and GND from the Arduino to the breadboard. Then the power and GND connections from the IR sensors, servo, and the motor drivers to the breadboard.

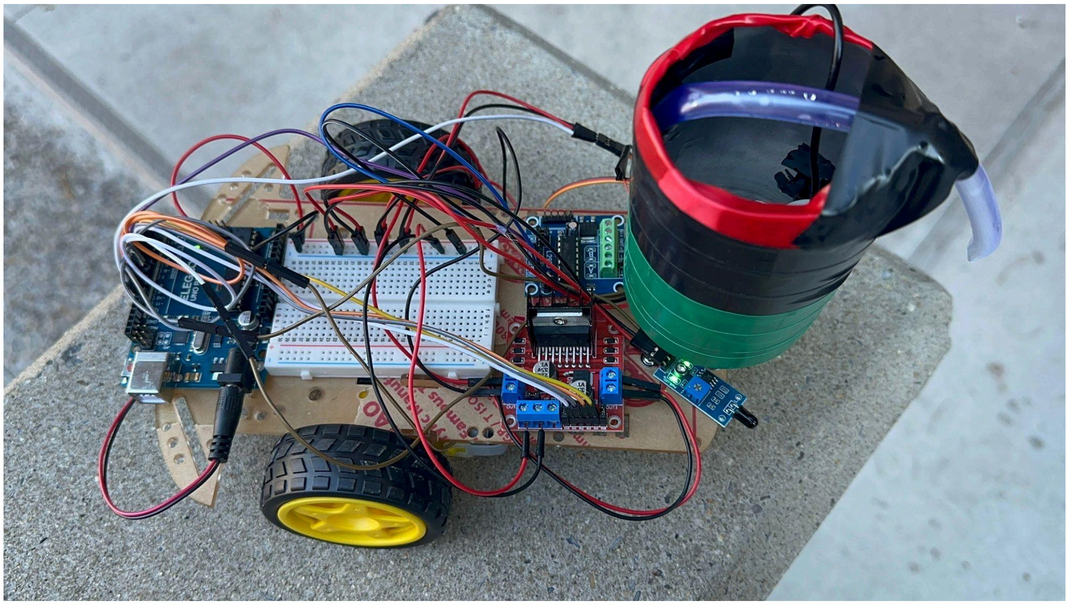
In addition, we connected the L293D motor shield from pin 6 to IN1. Then we connected the motor driver L298N from the Arduino pins 2,3,4,5 to the motor driver pins IN4, IN3, IN2, and IN1 respectively. Next, we connected the wires from the left motor to OUT1 and OUT2. We connected the right motor to OUT3 and OUT4. Then we connected the DO pins from each of the sensors to pins 8,9, and 10 on the Arduino. The forward sensor is pin 8, the left is pin 9, and the right is pin 10. Pin 11 on the Arduino is then connected to the servo motor.

Lastly, we have to connect the soda can and water pump. To do this we cut a soda can top off. We then taped around it. Next, we put the submersible DC pump in the can and taped it to the inside. Then we hot glued the servo motor arm piece to the underside of the can. We also placed a tube connected to the DC pump that came out and pointed down. We taped this tube to the can. The pump wires can be connected to the A+ and A- connections on the L293D motor shield. Finally, we can place the can on top of the servo motor in such a way that the spout is pointed forward. That concludes the assembly of the robot. The pictures below show the completed robot as well as a picture of the bottom of the soda can showing the hot glued servo arm.



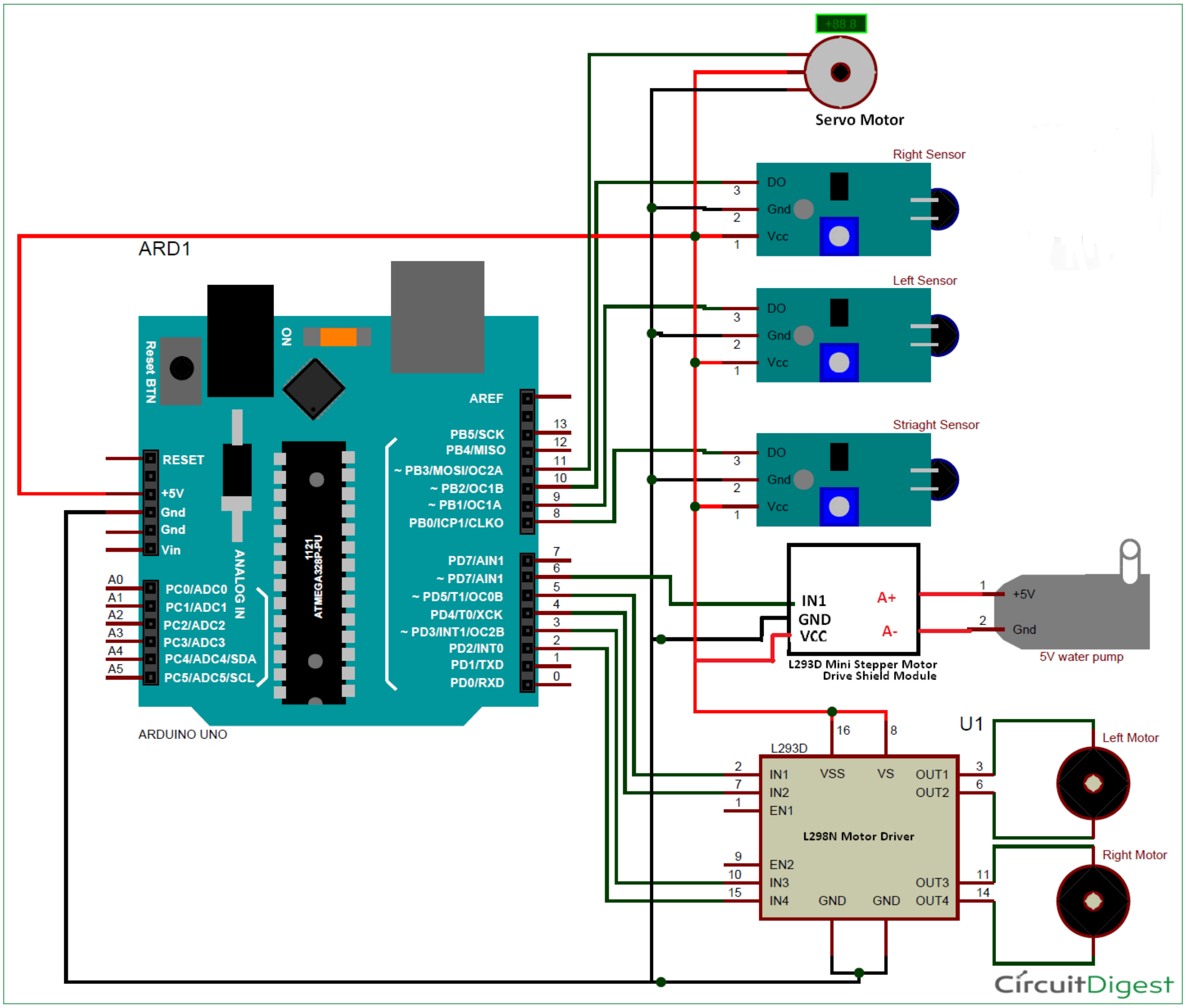






Circuit Diagram

This is a circuit diagram of all the connections required for this robot.



Code and Explanation

First, we include the header file to use the servo motor. Then we set up some variables, my servo for our Servo and a variable to control the position of the servo. We also make a fire variable that will track if a fire is present in the front of the robot that's a boolean.

```
#include <Servo.h>

 Servo myservo;
```

```
  int pos = 0;

  boolean fire = false;
```

Here we define our sensor pins.

```
  /*-------defining Inputs------*/

  int forwardSensor = 8;
  int leftSensor = 9;
  int rightSensor = 10;
```

Here we define our submersible pump and motor pins.

```
  /*-------defining Outputs------*/
  int pump = 6;
  int leftMotor1 = 2;
  int leftMotor2 = 3;
  int rightMotor1 = 4;
  int rightMotor2 = 5;
```

In the setup section we set the sensors as input, and the motors and pump as outputs. We also begin a serial monitor for debugging. Then we also attach the servo to pin 11 and start it off at 90.

```
void setup()

  {

          pinMode(leftSensor, INPUT);

          pinMode(rightSensor, INPUT);

          pinMode(forwardSensor, INPUT);

          pinMode(leftMotor1, OUTPUT);

          pinMode(leftMotor2, OUTPUT);

          pinMode(rightMotor1, OUTPUT);

          pinMode(rightMotor2, OUTPUT);
```

```
        pinMode(pump, OUTPUT);

        Serial.begin(9600);

        myservo.attach(11);

        myservo.write(90);
  }
```

This function put_off_fire, gets called once a fire has been detected at the front sensor to put out the fire.

```
  void put_off_fire()

  {

        delay(500);
```

These lines are to turn off the motors.

```
        digitalWrite(leftMotor1, HIGH);

        digitalWrite(leftMotor2, HIGH);

        digitalWrite(rightMotor1, HIGH);

        digitalWrite(rightMotor2, HIGH);
```

Then the pump is turned on.

```
        digitalWrite(pump, HIGH);
        delay(500);
```

While the pump is on the following lines are for the servo to sweep left and right.

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

        myservo.write(pos);

        delay(10);
        }

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

```
        myservo.write(pos);

        delay(10);
        }
```

Then the pump is turned off and myservo returns to its starting position.

```
        digitalWrite(pump, LOW);

        myservo.write(90);
```

The fire variable is also turned false since the fire is put out.

```
        fire = false;
 }
```

In our loop, we first start the servo at 90 degrees so pointing forward. Then we write that if no sensor detects a flame, stop the motors. The IR sensors return 1 if no flame is detected, and 0 if a flame is detected. Then we have a branch that if the front sensor senses fire, moves forward and turns the fire variable to true. If the right sensor or left senses fire the robot should move in the appropriate direction, so we have those if branches as well. We then add a delay so the robot moves at a better pace. We then have a while loop that says if the fire variable is true, such as when the front sensor senses a flame, to call the put_off_fire function to put out the flame. If the other senses sense flame the robot moves in that direction so the flame is brought to the front of the robot where the function to put out the flame can be called. We also add in our Serial debugging statements to print out which sensor picks up a flame reading in each direction branch.

```
void loop()

 {

        myservo.write(90);
```

```
    if (digitalRead(leftSensor) == 1 && digitalRead(rightSensor) == 1 && digitalRead(forwardSensor) == 1)  //If Fire not detected all sensors are one

    {

    //Do not move the robot

    digitalWrite(leftMotor1, LOW);

    digitalWrite(leftMotor2, LOW);

    digitalWrite(rightMotor1, LOW);

    digitalWrite(rightMotor2, LOW);

    }



    else if (digitalRead(forwardSensor) == 0)  //If Fire is straight ahead

    {

    //Move the robot forward

    digitalWrite(leftMotor1, HIGH);

    digitalWrite(leftMotor2, LOW);

    digitalWrite(rightMotor1, HIGH);

    digitalWrite(rightMotor2, LOW);

    fire = true;
    Serial.println("forward");

    }



    else if (digitalRead(leftSensor) == 0)  //If Fire is to the left
```

```
{

//Move the robot left

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, HIGH);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, HIGH);

Serial.println("left");

}



else if (digitalRead(rightSensor) == 0)  //If Fire is to the right

{

//Move the robot right

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, HIGH);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

Serial.println("right");
}



delay(300);  //Slow down the speed of robot



while (fire == true)

{
        put_off_fire();
        }
 }
```

Challenges Faced

Building the fire-fighting robot came with its challenges. Wiring everything together was harder than we expected as we had to ensure all the connections were secure and correct. It took a lot of trial and error. Getting the flame sensors to react properly was another challenge, sometimes it did not pick up the flame, or it would shift slightly and lose track of it. For the can, we have the rocker arm for the servo taped to the bottom of the can. However, we realized we needed the linear rocker arm to be more sturdy at the bottom of the can. The weight of the water in the soda was also causing issues, with the robot leaning forward at times and tipping over. Fitting all the electronic components on the chassis without things getting in each other’s way was tricky and took some creative thinking.

There were even moments during testing when the robot moved away from the flame instead of toward it. Since our project involved fire, we had to test the robot outdoors. As we were outside troubleshooting our robot, the cold weather was not on our side since we had wooden matches. The cold winds sometimes put our wooden matches out too soon for the three flame sensors to detect them. Apart from the sensors' difficulty picking up the flame, since the robot was tested during the daytime, it also picked up the light from the sun. From picking the light from the daytime, the left flame sensor would just flicker along with the middle sensor, and the right sensor would just stay turned on.

Conclusion

In conclusion, we learned that this project was going to involve much wiring as this involved a mobile robot along with the mechanism of detecting fire and extinguishing with a water pump. The inclusion of the water we figured out as a group that we needed to make sure that the water pump would pump the water out without making contact with the hardware components. Since the project involved fire, we had to choose between regular lighters or wooden matches. In the end, we decided to use wooden matches as these were easily decomposable; unlike the lighter, it could damage the spark wheel if exposed to water. As challenging as it was to build and troubleshoot the fire-extinguishing robot, it was satisfying to have the robot function the way it was supposed to.

Thankfully we already had a semi-pre-built robot car with the motors and connections installed. It took a lot of testing but we built a robot that could detect a flame, move toward it, and put it out. Observing the final product in action made all the hard work worth it. This project taught us about being proactive together, and we're proud of what we accomplished.