PROJECT NAME:

FIRE FIGHTER ROBOTIC VEHICLE USING AURDINO

PAGE OF CONTENTS

1.Abstract:

The abstract of the fire-fighter robotic car project focuses on the development of an innovative autonomous vehicle designed specifically for firefighting operations. This project aims to create a sophisticated robotic car equipped with advanced sensors, fire suppression systems, and real-time communication capabilities. The robotic car's primary objective is to navigate challenging environments and respond to emergency fire situations effectively. By utilizing cutting-edge technologies, such as AI-driven navigation, thermal imaging, and remote control, the project aims to enhance firefighting efficiency, minimize human risks, and improve overall emergency response strategies. The abstract highlights the interdisciplinary nature of the project, combining robotics, AI, and firefighting expertise to develop a practical and valuable solution for addressing fire-related emergencies.

2.Introduction:

Robot is a machine that looks like a human being and performs various complex tasks. There are many types of robots such as fixed base robot, mobile robot, underwater robot, humanoid robot, space robot and medicine robot etc. In this paper a FIRE EXTINGUISHING ROBOT is proposed. This robot is equipped with a single flame sensor used to sense environmental fire and feed the signals to the microcontroller in order to trigger the pump which sprinkles water in order to extinguish the fire. The project that is being presented is focused on a firefighting robot. Robots are capable of performing tasks in a more efficient, cost-effective, and accurate manner than humans. It has grown in popularity as technology has advanced, making human work simpler. The firefighting robot is programmed to scan for and extinguish fires in affected areas. The consequences of fire cannot be prevented, and they can occur in both young, newly formed forests and mature natural forests. Fire has a focused impact on plant growth because it destroys undesirable vegetation, allowing other species to emerge.

Figure 1

4.Problem definition:

Fire disaster is one of the dangerous problems that can lead to heavy loss both financially and by taking lives. Sometime it becomes difficult for fighters to access the site of a fire because of explosive materials, smoke, and high temperatures. Such situations risk the lives of fire fighters too. In such environments, fire-fighting robots can be useful. This Fire Extinguishing Robot is based on IOT Technology. In Fire Extinguishing Robot, we intend to build a system that could extinguish a small flame by sensing and moving to the location itself. Sometime delay in the arrival of fire fighters leads to numerous consequences. The Fire Extinguishing robot continuously monitors the environment and extinguishes it without delay.

5.Methodology:

The theme of this paper is to automatically sense the environmental fire and extinguish it without human intervention. The methodology is divided into three parts. The first part is on the design structure, followed by hardware description and the finally on the programming design. All these three parts were assembled together and experiments were then performed to build a system that can extinguish the fire that was carried out.

6. Design Structure:

In this section, the prototype of robotic system is presented, in which it consists of IR flame sensors, servo motors, submersible water pump, motor driver, mini breadboard, BO motors, rubber wheels, processor, and communication module for exchanging data between the fire-fighting robot and Arduino software. Fig 2 shows the basic prototype of our firefighting robot. The robot carries four main functions: First, it initializes itself i.e. its sensors gets initializes as the power is supplied. Second, robot sense the surrounding environment (for instance for the level of temperature) and identify the fireplace. Third, robot sends the navigating information and starts to navigate itself towards the fireplace. Fourth, finally the robot starts to extinguish the fire with the help of servo motors and submersible water pump.

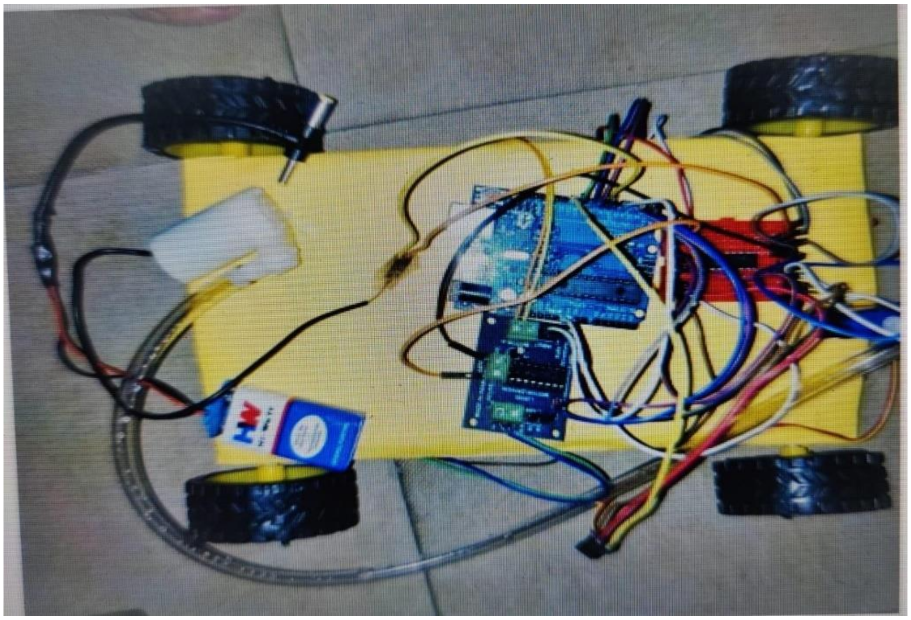


Figure 1. Fire Fighting Robot

6.1 Arduino UNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc and initially released in 2010. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards(shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE(Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes pre-programmed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

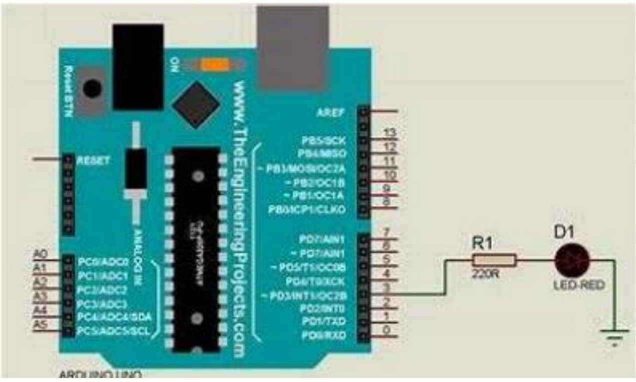


Figure-3

6.2 Fire sensor:

The main brain of this project is the Arduino, but in-order to sense fire we use the **Fire sensor module** (flame sensor) that is shown below.

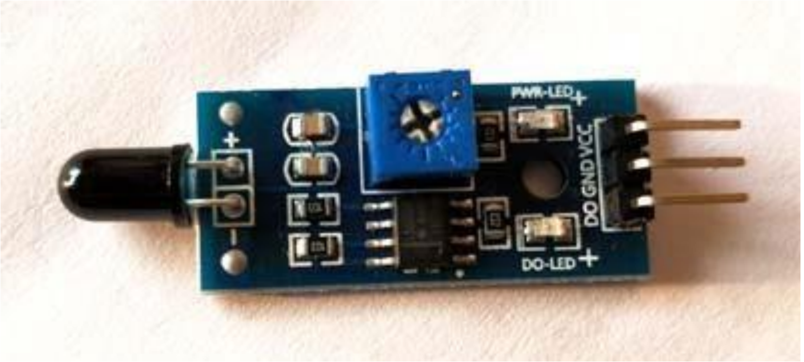
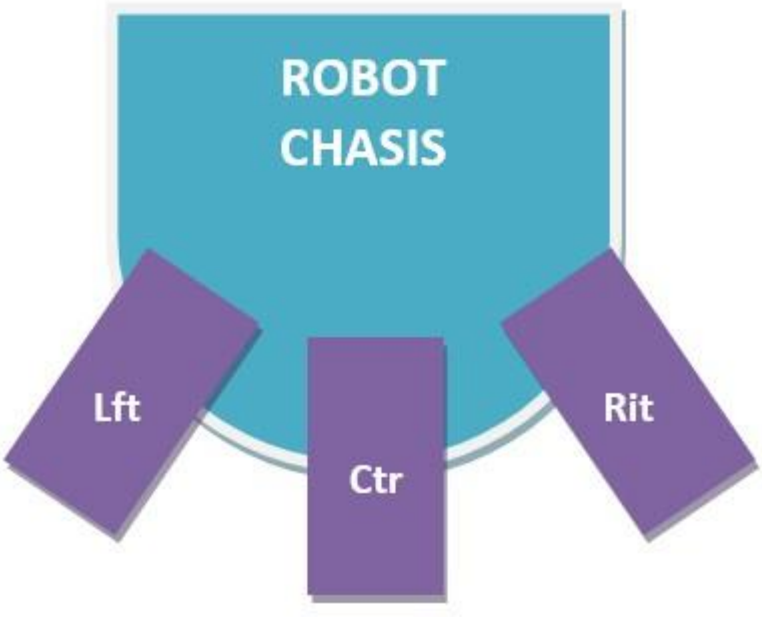


Figure 2. Fire Sensor Module

As you can see these sensors have an **IR Receiver (Photodiode)** which is used to detect the fire. How is this possible? When fire burns it emits a small amount of Infra-red light, this light will be received by the IR receiver on the sensor module. Then we use an Op-Amp to check for change in voltage across the IR Receiver, so that if a fire is detected the output pin (DO) will give 0V(LOW) and if the is no fire the output pin will be 5V(HIGH).

So, we place three such sensors in three directions of the robot to sense on which direction the fire is burning.



We detect the direction of the fire we can use the motors to move near the fire by driving our motors through the **L293D module**. When near a fire we have to put it out using water. Using a small container we can carry water, a 5V pump is also placed in the container and the whole container is placed on top of a **servo motor** so that we can control the direction in which the water has to be sprayed. Let’s proceed with the connections now

HARDWARE USED:

ATmega328P microcontroller (Arduino UNO):

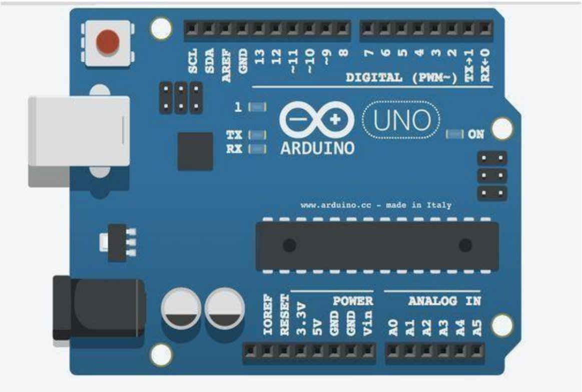


Fig 3: Arduino Uno microcontroller based development board

A Microcontroller is a compact device with a processor, storage and configurable input/output devices on a single integrated circuit. We'll be using the Arduino UNO board, which combines a microcontroller with all of the extras needed to quickly create and debug projects. The ATmega3288based UNO is a microcontroller board

1. 14 Digital input/output pins,
2. 6 Analog inputs,
3. 16MHz Quartz crystal
4. USB connector,
5. Power jack,
6. ICSP header, and
7. Reset button.

Attach it to a computer via USB cable or power it with an AC to DC adapter to get started. The ATmega3288 has 32kb of memory, 2kb of SRAM, and 1kb of EPROM. The Arduino Software can be used to programmed the UNO(IDE). The boot loader on the ATmega3288 on the UNO comes preprogrammed, allowing you to upload new code. It uses the original STK 500 protocol to communicate

Flame Sensors:

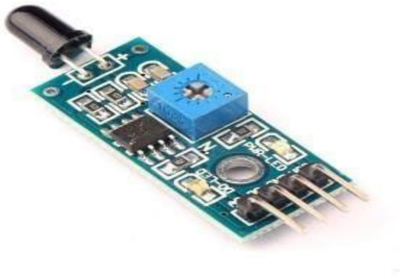


Fig 5: Flame Sensor

A flame sensor is the most sensitive to normal light of any sensor. This sensor senses flame if the light source emits a wave length between 760nm and 1100nm.The detection angle would be 60o and can be achieved from a distance of 100cm. This sensor's output is either an analog or digital signal. The infrared flame flash method is used by this sensor as shown in Figure 4.

Fig 4.2 shows the flame sensor. This sensor is able to detect a flame by sensing light wavelength between 760 –1100 nanometers. The test distance depends on the flame size and sensitivity settings.

The detection angle is 60 degrees, so the flame does not have to be right in front of the sensor.

There are two sensor outputs

i.Digital – sending either zero for nothing detected or one for a positive detection

ii. Analog – sending values in a range representing the flame probability/size/distance; must be connected to a PWM capable input

It has four pins,

1. Voltage Supply (Vcc),

2. Ground pin (GND),

3. Analog output (Aout), and

4. Digital output (Do)

Servo Motors:

Fig 7 shows the Servo Motors. Servo Motors are electronic devices that are mainly used for providing specific velocity and acceleration

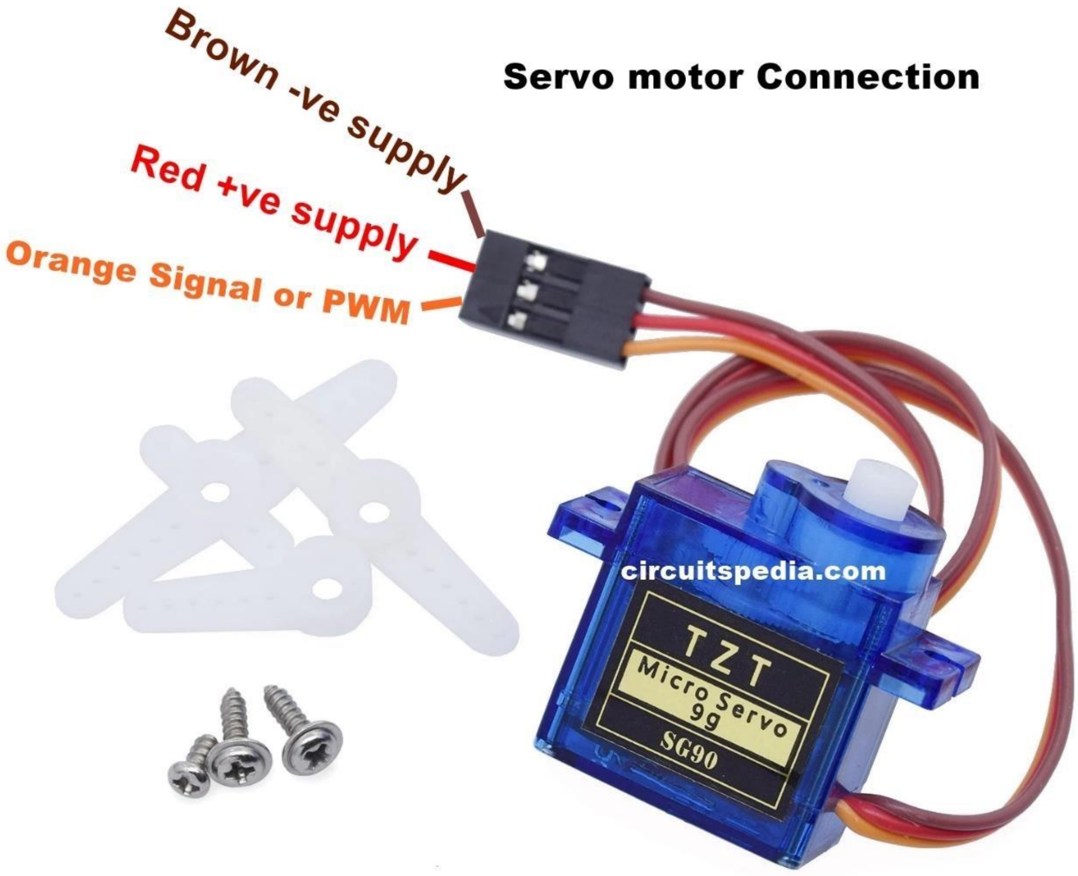


Figure 6. Servo Motor

Submersible Water Pump:

Fig 7. Shows the Submersible Water Pump. Submersible Water Pump is ideal for making automatic watering system using Arduino. The water pump is an important part of the robot as it will pump water to extinguish the fire.

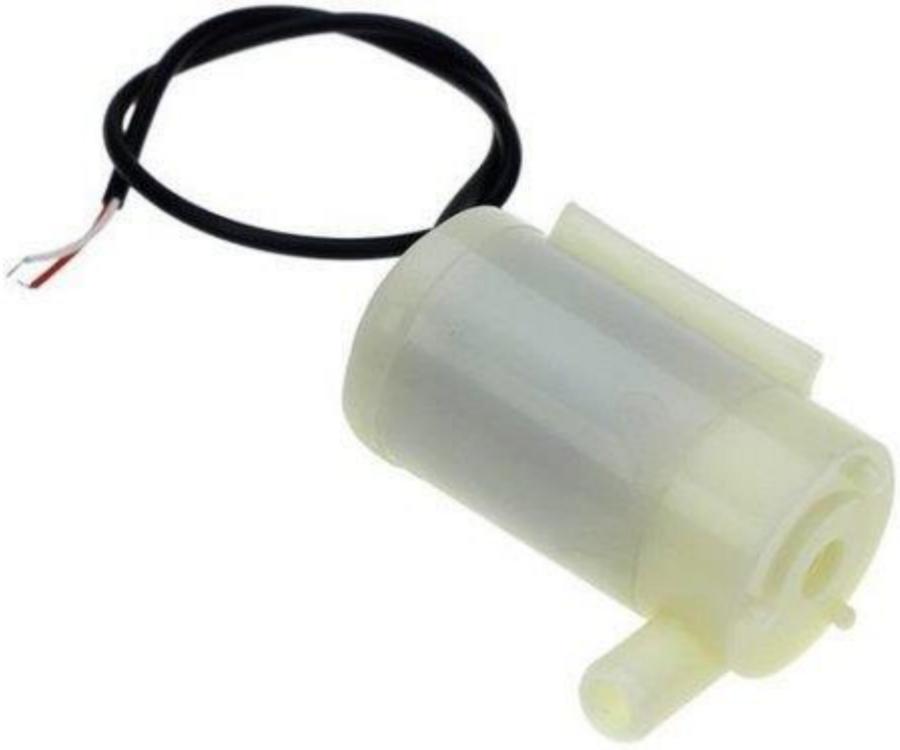


Figure 7. Submersible Water Pump

Motor Drivers:

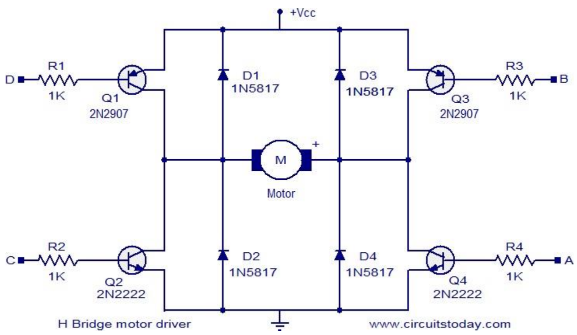


Fig 8: H Bridge circuit

Motor drivers are used to describe the direction of movement of the robot. It is used to give high voltage and high current as an output to run the motors which are used in the project for the movement of the robot. Fig 5 is the circuit of the H Bridge which is used for the motor driving in the IC L293D and also provides the bidirectional motor control.

7.Conclusion:

The proposed system is an independent automated structure for the urban indoor small farming without the usage of soil or manual watering. With the integration of some technological advanced instruments, this mechanization will not only encourage supply of fresh produce of certain amount to the people not even being involved actively in gardening but also monitor the system status with the web service of the parameter data.

8.Circuit Diagram:

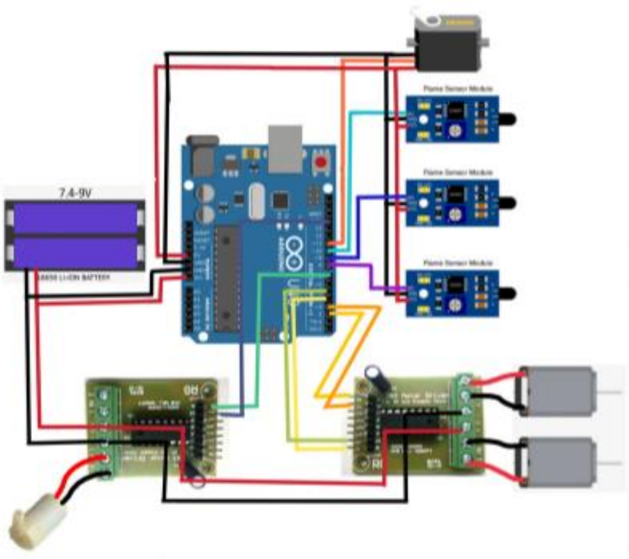


Figure -5

9.References:

https://www.bing.com/ck/a?!&&p=117777326ae75550JmltdHM9MTY2OTg1
MjgwMCZpZ3VpZD0xNTcyMjFkYy1mOGNiLTYxM2EtMzQyYy0zMGNhZ
jk2NjYwMGQmaW5zaWQ9NTIwNw&ptn=3&hsh=3&fclid=157221dc-f8cb613a-342c-

30caf966600d&psq=automatic+plant+watering+system+using+arduino&u=a1a
HR0cHM6Ly9jcmVhdGUuYXJkdWluby5jYy9wcm9qZWN0aHViL25lZXRpd
Ghha3VyL2F1dG9tYXRpYy1wbGFudC13YXRlcmluZy1zeXN0ZW0tdXNpb
mctYXJkdWluby11bm8tODc2NGJh&ntb=1 (**viewed on date**:12-04-2023)

10.Code:

```
int m1=3;
int m2=2;
int m3=4;
int m4=5;
int f1=A0;
int f2=A1;
int buz=6;
int pump=7;
int i=0;
void setup() {
  // put your setup code here, to run once:
pinMode(m1,OUTPUT);
pinMode(m2,OUTPUT);
pinMode(m3,OUTPUT);
pinMode(m4,OUTPUT);
pinMode(buz,OUTPUT);
pinMode(pump,OUTPUT);
pinMode(f1,INPUT);
pinMode(f2,INPUT);
Serial.begin(9600);
digitalWrite(pump,0);
digitalWrite(buz,0);
}

void loop() {
  // put your main code here, to run repeatedly:
   int f1val=digitalRead(f1);
   int f2val=digitalRead(f2);
if(f1val==0)
{
    Serial.println("Fire Detected,0");
    digitalWrite(buz,1);
    digitalWrite(m1,1);
    digitalWrite(m2,0);
    digitalWrite(m3,1);
    digitalWrite(m4,0);
    delay(4000);
    digitalWrite(m1,0);
    digitalWrite(m2,1);
    digitalWrite(m3,1);
    digitalWrite(m4,0);
    delay(1000);
    digitalWrite(m1,0);
    digitalWrite(m2,0);
    digitalWrite(m3,0);
  digitalWrite(m4,0);
    digitalWrite(pump,1);
    delay(4000);
    digitalWrite(pump,0);
    digitalWrite(buz,0);
  }

  if(f2val==0)
 {
    Serial.println("Fire Detected,0");
    digitalWrite(buz,1);
    digitalWrite(m1,1);
    digitalWrite(m2,0);
    digitalWrite(m3,1);
    digitalWrite(m4,0);
    delay(4000);
    digitalWrite(m1,1);
    digitalWrite(m2,0);
    digitalWrite(m3,0);
    digitalWrite(m4,1);
    delay(1000);
    digitalWrite(m1,0);
    digitalWrite(m2,0);
    digitalWrite(m3,0);
    digitalWrite(m4,0);
    digitalWrite(pump,1);
    delay(4000);
  digitalWrite(pump,0);

  digitalWrite(buz,0);

 }



delay(100);
```