International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

REAL TIME WIRELESS FIRE FIGHTING ROBOT

T. Jaya sriram[1], C. Harshavardhan Reddy[2], G. Sai Chandan[3], M. Rishik Naik[4], V. Praveen[5]

[1] Assistant Professor, Dept. Of Mechanical Engineering, Gurunanak Institute Of Technology, Khanapur, India.
[2,3,4,5] UG Scholars, Dept. Of Mechanical Engineering, Gurunanak Institute Of Technology, Khanapur, India.

---***---

Abstract - The **Real-time Wireless Firefighting Robot** is an innovative autonomous robotic system designed for effective fire response in hazardous environments. This robot integrates advanced technologies such as wireless communication, sensors, and real-time data processing to enhance firefighting capabilities. Equipped with a high-resolution camera, the robot can navigate through smoke-filled areas and transmit live video feeds to the control center. The wireless communication system enables seamless remote operation, allowing firefighters to assess the situation and make informed decisions. The robot's robust design and fire-resistant materials ensure its durability in high-temperature environments. Overall, the Live Wireless Firefighting Robot presents a cutting-edge solution for improving firefighting efficiency and reducing risks to human responders. This robot is equipped with advanced sensors for real-time fire detection and localization. Its wireless connectivity enables remote control and monitoring, allowing firefighters to operate it from a safe distance. The robot's mobility and agility enable it to navigate challenging environments, reaching areas that may be hazardous for humans. With a built-in water or foam spraying system, the robot can effectively suppress fires. Additionally, it incorporates live video streaming to provide a comprehensive view of the situation. The use of wireless technology not only facilitates seamless communication but also ensures quick response times, making this robot a valuable asset in firefighting scenarios, minimizing risks to human lives.

***Key Words*:** Fire Fighting Robot, Mobility, Live video feeds.

1. INTRODUCTION

Real-time firefighting robots represent a significant advancement in emergency response technology, merging robotics with firefighting to enhance efficiency and safety in hazardous situations. These robots are equipped with state-of-the-art sensors, cameras, and actuators, enabling them to navigate through complex environments and assess fire conditions in real-time. Utilizing AI algorithms, they can analyze data on temperature, smoke density, and gas levels to identify the extent and location of the fire accurately. Moreover, these robots can autonomously deploy firefighting mechanisms such as water cannons, foam sprayers, or even chemical suppressants to extinguish flames effectively. Their ability to operate in high-temperature environments and hazardous conditions makes them invaluable assets for firefighters, allowing them to access areas that may be too dangerous for human intervention. By augmenting traditional firefighting techniques with robotic assistance, real-time fire fighting robots have the potential to revolutionize emergency response strategies, minimizing human risk and property damage while maximizing the efficiency of firefighting efforts. Tool to improve manuscript clarity for the reviewers. The final layout of the typeset paper will not match this template layout.

HARDWARE USED

ESP8266 MICROCONTROLLER: A microcontroller is an integrated circuit (IC) device used for controlling other portions of an electronic system, usually via a microprocessor unit (MPU), memory, and some peripherals. These devices are optimized for embedded applications that require both processing functionality and agile, responsive 18 interactions with digital, analog, or electromechanical components. An ESP8266 Wi-Fi module is a SOC microchip mainly used for the development of endpoint IoT (Internet of things) applications. It is referred to as a standalone wireless transceiver, available at a very low price. It is used to enable the internet connection to various applications of embedded systems. The ESP8266 Wi-Fi module comes with a boot ROM of 64 KB, user data RAM of 80 KB, and instruction RAM of 32 KB. It can support 802.11 Wi-Fi network at 2.4

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

GHz along with the features of I2C, SPI, I2C interfacing with DMA, and 10-bit ADC. Interfacing this module with the microcontroller can be done easily through a serial port. An external voltage converter is required only if the operating voltage exceeds 3.5 Volts. It is most widely used in robotics and IoT applications due to its low cost and compact size.

Fig-1: Microcontroller

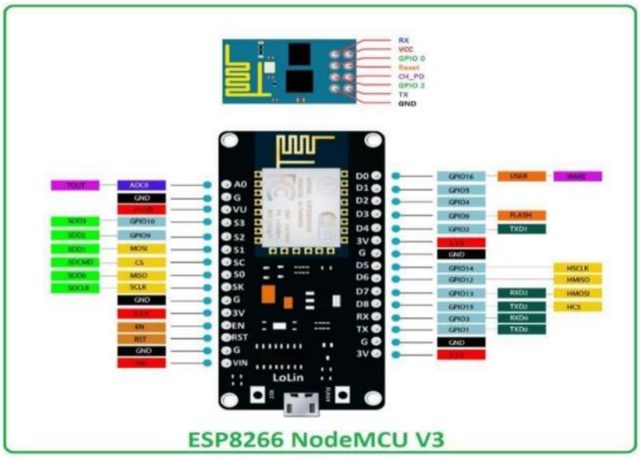


Fig-2: ESP8266 Wi-Fi module

BATTERIES: A lithium-ion or Li-ion battery is a type of rechargeable battery which uses the reversible reduction of lithium ions to store energy. It is the predominant battery type used in portable consumer electronics and electric vehicles. It also sees significant use for grid-scale energy storage and military and aerospace applications. Compared to other rechargeable battery technologies, Li-ion batteries have high energy densities, low self- discharge, and no memory effect (although a small memory effect reported in LFP cells has been traced to poorly made cells) generate CPU interrupts.

Fig-3: Batteries

L298D MOTOR DRIVER*:* L298 is a high power version of L293 motor driver IC. The L298D is a popular motor driver IC .As the name suggests it is mainly used to drive motors L298N Dual H Bridge Motor Driver is a motor controller breakout board which is typically used for controlling speed and direction of motors. It can also be used to control the brightness of certain lighting projects such as high powered LED arrays. An H-bridge is a circuit that can drive a current in either polarity and be controlled by pulse width modulation.

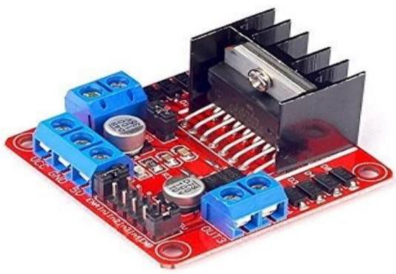


Fig-4: L298D Motor driver

L-293 MOTOR DRIVER: The L293D is a popular integrated circuit (IC) commonly used as a motor driver. It is specifically designed to control the direction and speed of small DC motors, making it useful in various robotics and electronics projects. Here's a description of the L293D motor driver. Motor Control: The L293D IC has two sets of H-bridge motor driver circuits, which can independently control the direction and speed of two DC motors. It can drive motors with voltages ranging from 4.5V to 36V and currents up to 600mA (1.2A peak). H-Bridge Configuration: Each motor driver circuit in the L293D consists of four transistors arranged in an H-bridge configuration. The H-bridge allows bidirectional control of the motor, enabling it to rotate in forward or

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

reverse directions. Enable Pins: The L293D has separate enable (EN) pins for each motor channel. By applying a logic high signal to the EN pin, the corresponding motor channel is enabled, allowing motor operation. When the EN pin is pulled low, the motor is disabled, and it stops. Logic Inputs: The L293D has two control inputs (IN1 and IN2) for each motor channel. These inputs are used to control the motor's direction. By applying specific logical combinations to the IN1and IN2 pins, the motor can be set to rotate in different directions (forward, reverse, or stop).

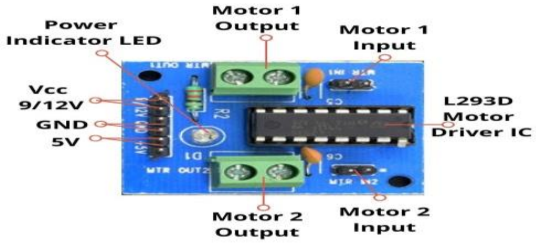


Fig-5: L-293D

DC MOTORS: A DC motor in simple words is a device that converts direct current (electrical energy) into mechanical energy. It’s of vital importance for the industry today. A DC motor is designed to run on DC electric power. Two examples of pure DC designs are Michael Faraday's homo-polar motor (which is uncommon), and the ball bearing motor, which is (so far) a novelty. By far the most common DC motor types are the brushed and brushless types, which use internal and external commutation respectively to create an oscillating AC current from the DC source— so they are not purely DC machines in a strict sense. Which operate in the ratings of 12v DC 0.6A.

Fig-6: DC Motor

CONNECTING WIRES: Connecting wires refers to the process of joining two or more electrical conductors to establish a continuous electrical path. This connection allows the flow of electric current between different components or devices. The purpose of connecting wires can vary depending on the context, and it can involve various techniques and components to ensure a secure and reliable electrical connection.

Fig-7: Connecting Wires

SERVO MOTORS Servo motors are high-precision superstars of the robot world. Unlike regular motors that simply spin, servos can be told to move to a specific angle and hold it there. This magic trick relies on a combination of parts: a regular DC motor, a gear system, and a position sensor (often a potentiometer). The control circuit receives an electrical signal indicating the desired position for the shaft. This signal is converted into a pulse with a specific width. The wider the pulse, the further the motor turns. The position sensor keeps track of the actual shaft angle and feeds that information back to the control circuit. If there's a mismatch, the circuit sends a new pulse to the motor, fine-tuning its movement until it reaches the target position. This continuous feedback loop ensures the servo hits the right spot and stays there.

Fig-8: Servo Motors

SOFTWARE USED

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a Motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analogy input/output (I/O) pins that may be interfaced to various expansion boards.

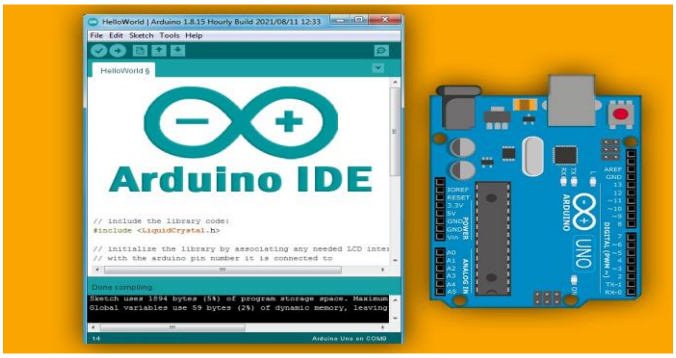


Fig 9: Arduino and board

Sample code Turn an LED on and off every second.

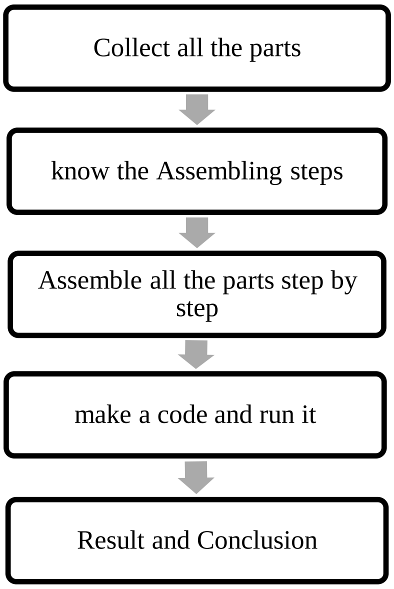
```
https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

Fig-10: Arduino basic code.

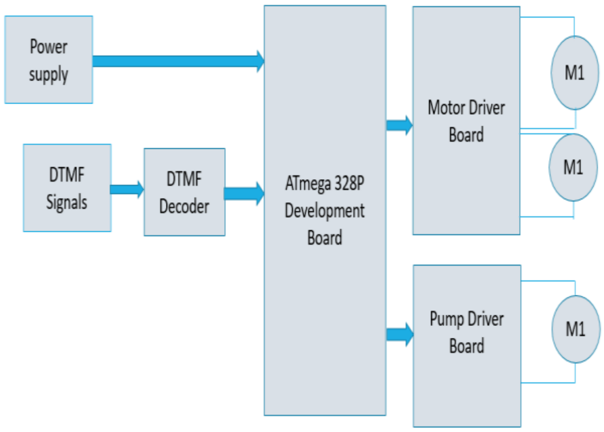
2. METHODOLOGY



Flow chart-1: Methodology

EXPLANATION

- Group all the parts Requires.
- Know how to Assemble.
- Step by Step assemble all the parts to form the final product.
- Create a code in Arduino and run it for our application.
- Run The Code, after satisfy our required usage install the code in ESP8266.
- Control the Robot with the phone and Run the Robot.



Flow Chart-2: Overall Process chart

PROBLEM STATEMENT

Firefighters face immense danger entering burning buildings, with limited visibility due to smoke and potential for hazardous debris. Current firefighting technology often relies on manual control, which can be inefficient in navigating complex environments. Therefore, there is a critical need for more advanced firefighting robots that can enter hazardous zones, operate autonomously, and adapt to changing fire conditions. Firefighters face a constant battle against dangerous and unpredictable environments. Traditional firefighting methods rely heavily on human intervention, which puts firefighters at immense risk. Here's the problem: Current firefighting technology often falls short. Existing firefighting robots may struggle in navigating hazardous terrain due to debris or tight spaces. Additionally, these robots might have limited functionalities, often focusing solely on water delivery. This lack of adaptability and single-function approach hinders effective firefighting, especially when dealing with dynamic situations and complex fire locations. In essence, we need a new generation of firefighting robots that are more versatile, can navigate challenging environments, and offer a wider range of firefighting capabilities to improve response times and firefighter safety.

3. RESULT AND DISCUSSION

RUNNING THE AURDINO CODE:

```
#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

#include<Servo.h>


const char *ssid = "Fire";

const char *password = "123456789";

int s1=70,s2=90;

Servo servo1;

ESP8266WebServer server(80);

String page =""; //For the Web Server

String page2="";  //For updating Status of robot

int sensor=10;

String mstr;

void setup() {

page="<center><h1>chitti the robot</h1><body><p><a href=\"F\"><button>forward</button></a></p><p><center><a href=\"B\"><button>backward</button></a></p><center><p><a href=\"R\"><button>right</button></a></p><center><p><a href=\"L\"><button>left</button></a></p><p><center><a href=\"S\"><button>Stop</button></a></p><p><a href=\"b\"><button>PUMP_ON</button></a></p><p><a href=\"c\"><button>PUMP_OFF</button></a></p></body><p><a href=\"a\"><button>Servo_sweep</button></a></p>";

delay(1000);

pinMode(D2,OUTPUT);

pinMode(D3,OUTPUT);

pinMode(D4,OUTPUT);

pinMode(D5, OUTPUT);   // inputs for motor 1

pinMode(D6,OUTPUT);

pinMode(D7,OUTPUT);    // inputs for motor 2

pinMode(D8,OUTPUT);

pinMode(LED_BUILTIN,OUTPUT);   // For status of WiFi connection

digitalWrite(D5,LOW);

digitalWrite(D6,LOW);

digitalWrite(D7,LOW);

digitalWrite(D8,LOW);
```

```
Serial.begin(115200);
Serial.println();
Serial.print("Configuring access point…");
WiFi.softAP(ssid, password);
IPAddress myIP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(myIP);
servo1.attach(D2);
server.begin();
Serial.println("HTTP server started");
  server.on("/", webpage);
  server.on("/F",Forward);
  server.on("/B",Backward);
  server.on("/L",Left);
  server.on("/R",Right);
  server.on("/S",stop1);
  server.on("/b",PUMP_ON);// turns all the motor input pins low
  server.on("/c",PUMP_OFF);
  server.on("/a",servo_sweep);
  delay(200);
  server.begin();
  Serial.println("Web server started!");
}

void loop() {
//mstr=String(sensor);
server.handleClient();
//server.send(200,"text/plain",mstr);

}

void webpage()
{
server.send(200, "text/html", page+page2);
}


void Forward()
 {
    //analogWrite(D3,60);
    //analogWrite(D4,80);
    digitalWrite(D5,HIGH);
    digitalWrite(D6,LOW);
    digitalWrite(D7,HIGH);
    digitalWrite(D8,LOW);
    Serial.print('F');
    page2="<center><p> Robot Status : Forward </p></center>";
    server.send(200,"text/html", page+page2);
    delay(200);
  }
 void Left()
  {
```

```
  page2="<center><p> Robot Status : Left</p></center>";
  server.send(200,"text/html",page+page2);
//  analogWrite(D3,s1);
//  analogWrite(D4,s2);
  digitalWrite(D5,LOW);
  digitalWrite(D6,LOW);
  digitalWrite(D7,HIGH);
  digitalWrite(D8,LOW);
  delay(200);
  Serial.print('L');
}
void Right()
 {
  //analogWrite(D3,s1);
  //analogWrite(D4,s2);
  page2="<center><p> Robot Status : Right</p></center>";
  server.send(200,"text/html",page+page2);
  digitalWrite(D5,HIGH);
  digitalWrite(D6,LOW);
  digitalWrite(D7,LOW);
  digitalWrite(D8,LOW);
  delay(200);
  Serial.print('R');
}
 void Backward()
{
  page2="<center><p> Robot Status : Backward</p></center>";
  server.send(200, "text/html", page+page2);
  //analogWrite(D3,s1);
  //analogWrite(D4,LOW);
  digital Write(D5,LOW);
  digital Write(D6,HIGH);
  digital Write(D7, LOW);
  digital Write(D8,HIGH);
  delay(200);
  Serial.print('B');
 }
 void stop1()
 {
   page2="<center><p> Robot Status : Stop</p></center>";
//  page3="<center><p> motor 2 Status : off</p></center>";
 server. send(200,"text/html",page+page2);
 //analog Write(D3,s1);
  //analog Write(D4,LOW);
  digital Write(D5,LOW);
  digital Write(D6,LOW);
  digital Write(D7,LOW);
  digital Write(D8,LOW);
  Serial .print('S');
 }
 void PUMP_ON()
 {
```

```
String page5="<center><p> Bullet firing</p></center>";
server. send(200,"text/html",page+page5);
digital Write(D3,LOW);
digital Write(D4,HIGH);
delay(1500);
digital Write(D3,LOW);
digital Write(D4,LOW);

}

void PUMP_OFF()
{
String page6="<center><p> Bullet firing stopped</p></center>";
server. send(200,"text/html",page+page6);
digital Write(D3,LOW);
digital Write(D4,LOW);
delay(1500);
// digital Write(D7,LOW);
// digital Write(D8,LOW);
}
void servo sweep ()
{

String page6="<center><p> Bullet firing stopped</p></center>";
server.send(200,"text/html",page+page6);
for(int a=0;a<=180;a++)
{
servo1.write(a);
delay(50);
}
for(int b=180;b>=0;b--)
{
servo1.write(b);
delay(50);
}
}
```

WATER SPARYING MECHANISM

Creating a water spraying system for a real-time firefighting robot without sensors would require a programmable logic controller (PLC) or a microcontroller to control the spraying mechanism. Here's a detailed approach:

Mechanical Design: Design the robot with a water tank, pump, and spraying mechanism. Ensure the robot's structure can withstand the weight of the water tank and the force of the water pump.

Water Pump Control: Use a motor controller or relay connected to the microcontroller to turn the water pump on and off. The microcontroller can activate the pump when needed.

Spraying Mechanism: Design a mechanism to control the direction and intensity of the water spray. This could be achieved using servos or stepper motors to adjust the angle and height of the water nozzle.

Real-Time Response: Ensure the programming logic is efficient and responsive to quickly react to fire incidents. Test the system to verify its real-time performance.

Safety Measures: Implement fail-safes to prevent accidental activation of the water spraying system and ensure the robot's safety during operation.

Testing and Optimization: Test the robot in controlled environments to optimize its performance. Adjust the spraying mechanism's parameters as needed for better coverage and effectiveness.

Fig 11 : Water spraying in fire fighting robot

SERVO MOVEMENTS

In a firefighting robot project, servo movements play a crucial role in directing various components such as the water spraying mechanism and the robot's mobility. Let's delve into how servos are utilized in detail:

Water Spraying Mechanism: Servo motors can control the movement of nozzles or valves in the water spraying system. These servos enable precise aiming and adjustment of the water stream's direction and intensity. For instance, a servo could control the angle of a nozzle to target specific areas of the fire or adjust the flow rate to optimize extinguishing efficiency.

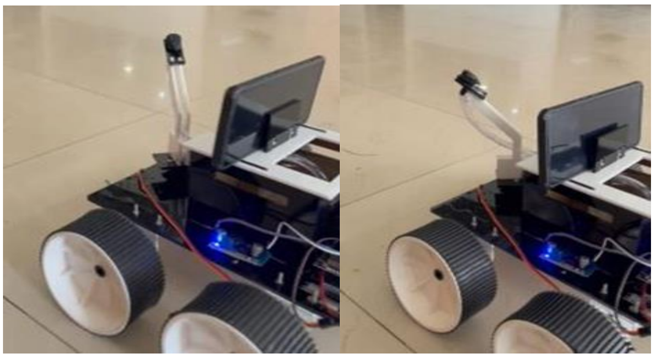


Fig 12 Movement of servo with a sweep of 180^0angle of rotation

SUMMARY

Hardware Assembly

Gather the following components:

- Arduino board (such as Arduino Uno), ESP8266 module, DC Motors, Motor driver (such as L298N), battery pack, wheels, blades, and chassis.
- Connect the ESP8266 to the Arduino board using the appropriate pins (for example, ESP8266 TX to Arduino RX and ESP8266 RX to Arduino TX).
- Connect the Motor driver to the Arduino board, ensuring proper connections for Motor control (inputs and outputs).
- Connect the DC Motors to the Motor driver, observing the correct polarity.
- Attach the wheels securely to the DC Motors.
- Mount the blades onto the chassis, ensuring they can rotate freely.

Component Purpose:

- **Arduino board**: Provides the microcontroller for controlling the robot's various components.
- **ESP8266 module**: Enables wireless communication, allowing you to control the robot remotely.
- **DC Motors:** Drive the wheels to move the robot.
- **Motor driver**: Acts as an interface between the Arduino and the Motors, controlling their speed and direction.
- **Blades:** Cut the grass when the robot moves.

Arduino and ESP8266 Code:

- Install the required libraries for ESP8266 support in the Arduino IDE.
- Write code to control the Motors, using the Motor driver and Arduino's PWM (Pulse Width Modulation) pins to vary Motor speed.
- Implement code to receive commands from the ESP8266 module via WIFI, interpreting them as movement instructions for the robot.
- Incorporate logic to control the blades, ensuring they activate only when the robot is in contact with the grass.

Safety Precautions:

- Always ensure the blades are properly covered or shielded to prevent injury.
- Use appropriate safety gear, such as gloves and goggles, when assembling or testing the robot.

- Keep by standers away during operation to avoid accidents.
- Be cautious of the robot's movement and ensure it doesn't come into contact with people, pets, or obstacles.

Troubleshooting and Improvements:

- If you encounter issues, double-check the wiring connections and ensure all components are correctly powered.
- Check for loose connections or damaged components.

4. FUTURE SCOPE

The future scope of firefighting robots is promising. They can be designed to autonomously navigate through hazardous environments, detect and extinguish fires, and even assist in search and rescue operations. With advancements in AI, robotics, and materials science, these robots can become more efficient, versatile, and capable of handling various firefighting scenarios, ultimately reducing risks to human firefighters and saving lives. Additionally, integrating technologies like machine learning can improve their ability to adapt to different situations and optimize firefighting strategies. The future scope of firefighting robots is promising. They can operate in hazardous environments, reducing risks to human firefighters

5. CONCLUSION

In conclusion, developing a fire-fighting robot project requires careful consideration of various components, including water spraying systems and servo movements. By integrating servo motors, the robot can achieve precise control over its water spraying mechanism, mobility, and surveillance capabilities. Enhance the accuracy and reliability of servo movements, ensuring the robot's effectiveness in extinguishing fires autonomously. Overall, servo technology plays a crucial role in the successful implementation of a fire-fighting robot project, enabling it to respond efficiently to fire emergencies and contribute to enhancing firefighting capabilities. In conclusion, firefighting robots equipped with cameras provide a range of advantages and applications for firefighting and emergency response situations. These robots can enter hazardous environments and provide real-time information to firefighters, enhancing their safety and ability to make informed decisions. Firefighting robots with cameras can be used in a variety of applications, including building and wildfire firefighting, search and rescue, hazardous material handling, and industrial firefighting.

6. ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to Guru Nanak Institute of Technology for their invaluable support and resources, which have been instrumental in the successful completion of our project. We wish to express our candid gratitude to **Dr. S. SREENATHA REDDY**, Principal and the management of the Guru Nanak Institute of Technology for providing us the best amenities which enabled us to complete our project in the stipulated time. We extend our deep sense of graduate to **Dr. B. VIJAYA KUMAR**, Professor & Head of the Mechanical Department for his masterly supervision and valuable suggestions for the successful International Journal of Engineering Technology Research & Management completion of our project. We owe our immense thanks to **Mr. JAYA SRI RAM** our project guide, Assistant Professor in Department of mechanical Engineering, Guru Nanak Institute of technology for the sustained interest, constructive criticism, and constant encouragement at every stage of this Endeavour. Finally, yet importantly, we are very thankful to our parents, friends, and other faculty of Mechanical Engineering Department for their constant support in completion of this project

7. REFERENCES

1. Prashant Badoni "Robotics Controller " Published in the International Journal of Engineering Research & Technology (IJERT), Issued 2278-018, Volume 4, Issue 10, Pages 502-508, October-2015.

2. Michael Shneier, Roger Bostelman "Literature Review of Mobile Robots for Manufacturing" Published in the Journal of National Institute of Standards and Technology (NIST), Volume 8022, Pages 1-7, May 2015.

3. Deepika Vashist, Parveen Kumar, Amit Dalal "A Review on Robotics Technology" Published in the Journal of

International Journal of Engineering Research & Technology (IJERT), Volume 5, Issue 3, Pages 1-4, 2017.

4. Lamber Royakkers, Rinie van Est "A Literature Review on New Robotics: Automation from Love to War" Published in the Journal of open access at SpringerLink, Volume 2017, Issued 10.1007/s12369-015-0295, Pages 1-22, 24 March 2015.

5. Sai Priya. T, Tejashwini. P.V "Literature Survey of Robot with Different Applications" Published in the Journal of International Advanced Research Journal in Science, Engineering and Technology, Volume 10, Issue 1, Pages 53-55, January 2023.