AUTOMATIC FIRE FIGHTING ROBOT WITH OBSTACLE DETECTION

A Project Report

Submitted By

Titas Saha
Soujit Das
Sayantani Mondal
Maitri Mandal
Sweety Mandal

In partial fulfilment for the award of the degree
Of
BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Under the Guidance of
MR. ANIRBAN GHOSAL
ASSISTANT PROFESSOR

JIS College of Engineering
(An Autonomous Institute)
Block "A" Phase III, Kalyani, Nadia-741235
May, 2022

DECLARATION

We hereby declare that the project entitled **"Automatic Fire-Fighting Robot with Obstacle Detection"** submitted for the B. Tech. (ECE) degree is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

_____

Titas Saha
ECE, 4th Year

_____

Soujit Das
ECE, 4th Year

_____

Maitri Mandal
ECE, 4th Year

_____

Sayantani Mondal
ECE, 4th Year

_____

Sweety Mandal
ECE, 4th Year

Date: 19TH May, 2022
Place: Kalyani, Nadia

JIS College of Engineering
Block 'A', Phase-III, Kalyani, Nadia, 741235
Phone: +91 33 2582 2137, Telefax: +91 33 2582 2138
Website: www.jiscollege.ac.in, Email: info@jiscollege.ac.in

CERTIFICATE

This is to certify that the students mentioned below have completed their project entitled **"Automatic Fire-Fighting Robot with Obstacle Detection",** under the guidance of **MR. Anirban Ghosal** in partial fulfilment of the requirements for the award of the **Bachelor of Technology** in **Electronics and Communication Engineering** from **JIS college of Engineering** (An Autonomous Institute) is an authentic record of their own work carried out during the academic year 2021-22 and to the best of our knowledge, this work has not been submitted elsewhere as part of the process of obtaining a degree, diploma, fellowship or any other similar title.

Students Concerned:

1. **Titas Saha**
2. **Soujit Das**
3. **Sayantani Mondal**
4. **Maitri Mandal**
5. **Sweety Mandal**

Signature of External Expert	**Signature of the HOD**	**Signature of the Supervisor**

ACKNOWLEDGEMENT

We would like to express our gratitude to **Mr. Anirban Ghosal** for providing effective guidance in development of this project work. His conscription of the topic and all the helpful hints, he provided, contributed greatly to successful development of this work, without being pedagogic and overbearing influence.

We also express our sincere gratitude to **Mr. Biswarup Neogi**, HOD of the Department of Electronics and Communication Engineering of JIS College of Engineering and all the respected faculty members of Department of ECE for giving the scope of successfully carrying out the project work.

This report may contain errors and shortcomings. Thus, we remain open to all criticisms and suggestions which could present us with new sources of inspiration as we develop in our ability to research and learn.

INDEX

1. Abstract	**6**
2. Introduction	**7**
Survey on recent investigations	**8-10**
3. Basic Components	**11-23**
4. Working Principle	**23-24**
Flow chart	**27**
Arduino Code	**29-39**
5. Conclusion	**40**
Future Scope	**40-41**
6. Reference	**42**

ABSTRACT

Sometimes it has been seen, that it has been impossible for fire-fighting personnel to access the site of the fire for extreme danger in petrochemical, chemical dangerous product, toxicity or exploder fire accidents. In such environments, firefighting robots can be useful for extinguishing a fire. This paper aims to design an ‘Automatic Fire Fighting Robot’. This device can perform tasks in small fire to medium fire accidents. This device can spread water into fire and find routes to detect obstacles. This device 'Automatic Fire Fighting Robot' has been designed with one ultrasonic sensor, one Arduino UNO, three fire sensors, one servo motor, one water pump, four wheels and four dc motors. This robot has a hard base and structure that protects electronics from outside heat or water or pressure of any heavy thing. Also, we can attach any water tank behind the robot for extra water supply.

INTRODUCTION

Fire fighting is an important job but it is a very dangerous occupation. Due to that, Robots are designed to find a fire, before it rages out of control. Nowadays robots are very much efficient to do this kind of work and replacement of humans to reduce any kind of physical damage. It could be used to work with firefighters to reduce the risk of injury to victims. Firefighting robots are operated in places where firefighters are unable to work. Therefore, it also can reduce human injury from a serious fire burning. So it is necessary to keep a fire fighting robot to prevent some fire accidents and in JIS College of Engineering we have built this kind of robot naming it 'Blaze Bosom Buddy '. At first, the robot tries to find a clear path and also detect any signal of fire. Whenever the robot detects any signal of fire using the Fire sensor module the robot immediately moves towards the fire and then it also detects any obstacles blocking the robot's path or not using an ultrasonic sensor (HC-SR04). Detecting all the obstacles and using an algorithm the robot lastly reach the fire. After reaching the fire the Arduino calculates the distance from the fire and opens up the water outlet to douse the fire. The robot will spray all the water till the fire is not completely doused. We can change the range of detecting obstacles and fires just by adjusting the sensor's potentiometers. The structure is a very unique design of this robot. This structure protects robots and it is the most important part of the robot to save the robot from any damage. This robot is fully automatic so any kind of manual control or human help is not needed. The battery backup is enough to operate the robot for at least 2 hours. This robot can operate in daylight and also at the night adjusting its sensing power.

Study on recent investigations:

Current Fire Fighting Machine:

The rapid development in technology nowadays, directly improve the tools and equipment used in fire fighting. With these advance tools and equipment, fire fighting can be more effective and efficient. Besides, it also reduces the risk to minimum level. Moreover, the damages of fire incident can be reduced. Fire fighting mobile robot is one of the solution that able to reduce the fire hazards risk on fire fighter. Different type of current available fire fighting machine will be described.

LUF60:

LUF60 is a diesel powered mobile fire fighting machine is equipped with air blower and a water beam fog. The machine is able to clear the hazardous obstacles by blowing the mixture of air and water. Besides, the machine is designed to withstand the rigor operating conditions and confined spaces. The LUF60 can be used in rail tunnels, aircraft hangers, parking garages, chemical plant, etc. There are some significant features of this machine. The monitor nozzle has a flow rate up to 800 GPM and it can blow the water beam as far as 80m. In order to enhance it mobility in high temperature condition, it is equipped with rubber track system which is rated to 400 degrees Fahrenheit. The rubber track system enables the robot tWo climb the stair. Besides, this machine able to operate on slopes of maximum 20 degrees with the ventilation tube at a maximum 45 degree angle (NRT, 2013).

FIREROB:

FIREROB is a remote controlled mobile robot that is used to fight fires, search and inspect fire scene. The FIREROB is protected from heat with a heat shield. It is equipped with high-pressure water mist extinguishers to control the fires. Besides, it also equipped with thermal imaging cameras and sensors for observation and monitoring purpose in fire scene. Two operation options for the FIREROB are available, which are with heat shield and without heat shield.

FR-1:

FFR-1 is a fire fighting robot that used to enable firefighters to carry out difficult missions. The FFR-1 is a remote controlled double-tracked robot that carry monitor and used to extinguish fire. The robot is designed to operate under hazardous environment conditions as high temperatures, poisonous materials and inside unsafe buildings.

Operating FFR-1 enables firefighters to execute missions in confined spaces, narrow streets, industrial buildings, stores, tunnels, airports, military installations, power plants, chemical plants and others. FFR-1 is easily transported on a modified vehicle or a trailer, imparting an operational advantage for firefighters at early stage of the fire as well as during the whole operation. The FFR-1 has a internal double-walled cooling system to avoid the overheating cause by high external temperature. FFR-1 is a highly maneuverable robot with crosscountry capabilities. It runs on treads and can climb up to 30 degree incline dragging a 3 inch fire hose (Mgsemi, 2005).

FIREMOTE 4800:

The Firemote 4800 illustrated in is an Unmanned Ground Vehicle. It is equipped with a firefighting nozzle, high pressure water equipment, spray nozzle, navigation cameras, thermal imaging camera, cooling system and roll-flat hose inlet. All the equipment is contained in a stainless steel insulated body which can reflect radiant heat. It can also protect tracks and sensitive parts by circulating water through chassis and using fixed nozzle to cool down the body. The Firemote 4800 is controlled from a panel by the help of color cameras consisted of 2 forward facing and 2 rearward facing, angled to give wide field of view. The panel uses a trans-reflective monitor that shows a software dashboard with the Firemote 4800's parameters and images from the 4 cameras simultaneously for ease of navigation. The command centre can receive video from the panel as it has an inbuilt computer that can record or send on video files.

MVF-5:

The MVF-5 as shown in Figure 5 is a multifunctional robotic fire fighting machine manufactured by Croatian manufacturer DOK-ING, to put out fires in unreachable areas and life threatening conditions. It is a fire fighting vehicle that operated by a single operator and controlled by using remote control. With the remote control technology GPS-INS (Global Position System- Inertial Navigation System), the system is able to be controlled in the range of 1500m.

Basic Components:

I. ARDUINO UNO:

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. 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 Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial

communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board.

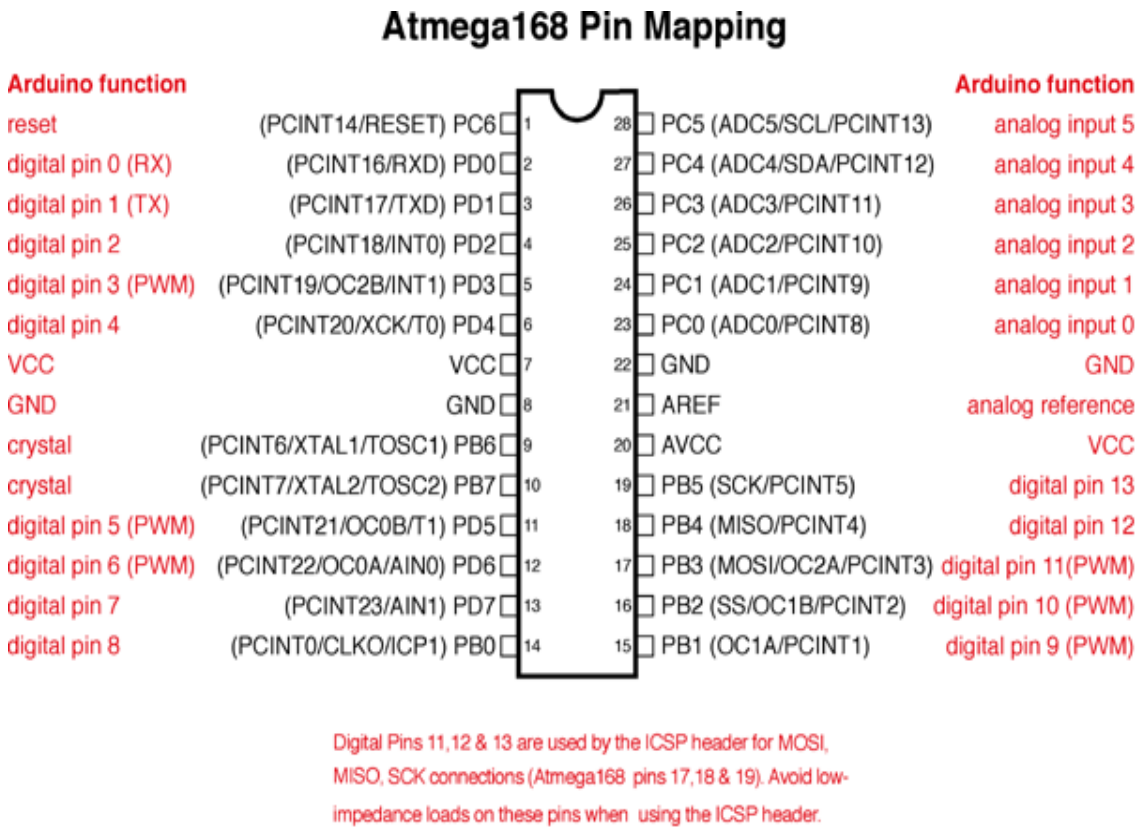


Fig1 : Pin Diagram of Arduino Uno

Technical specifications:

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA

- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g

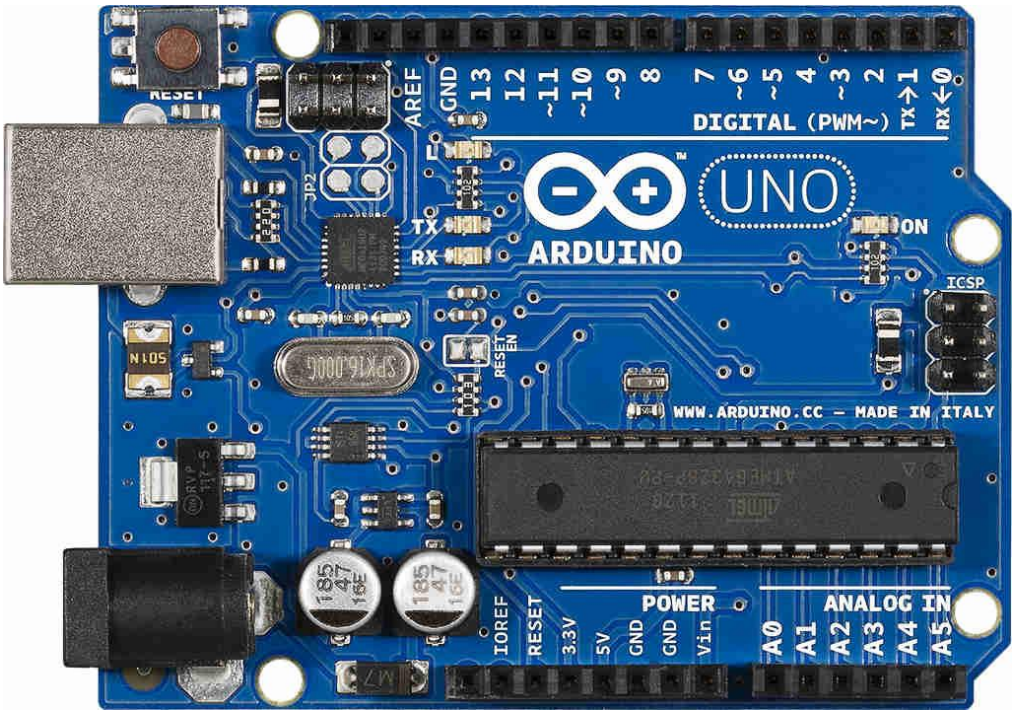


Fig 2: Arduino Uno

II. Flame Sensor:

A flame-sensor is one kind of detector which is mainly designed for detecting as well as responding to the occurrence of a fire or flame. The flame detection response can depend on its fitting. It includes an alarm system, a natural gas line, propane & a fire suppression system. This sensor is used in industrial boilers. The main function of this is to give authentication whether the boiler is properly working

or not. The response of these sensors is faster as well as more accurate compare with a heat/smoke detector because of its mechanism while detecting the flame.

This sensor/detector can be built with an electronic circuit using a receiver like electromagnetic radiation. This sensor uses the infrared flame flash method, which allows the sensor to work through a coating of oil, dust, water vapor, otherwise ice.

Fig 3: Flame Sensor

III. ULTRASONIC SENSOR– HC-SR04:

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity. High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

The acoustic wave signal is an ultrasonic wave travelling at a frequency above 18kHz. The famous HC SR04 ultrasonic sensor generates ultrasonic waves at 40kHz frequency. Typically, a microcontroller is used for communication with an ultrasonic sensor. To begin measuring the distance, the microcontroller sends a trigger signal to the ultrasonic sensor. The duty cycle of this trigger signal is 10µS for the HC-SR04 ultrasonic sensor. When triggered, the ultrasonic sensor generates eight acoustic (ultrasonic) wave bursts and initiates a time counter. As soon as the reflected (echo) signal is received, the timer stops. The output of the ultrasonic sensor is a high pulse with the same duration as the time difference between transmitted ultrasonic bursts and the received echo signal.

Fig 4: Ultrasonic Sensor HC-SR04

IV. CL 100 TRANSISTOR:

CL100 is a general purpose, medium power NPN transistor. It is mostly used as switch in common emitter configuration. The transistor terminals require a fixed DC voltage to operate in the desired region of its characteristic curves. This is known as the biasing. For switching applications, SL100/Cl100 is biased in

such a way that it remains fully on if there is a signal at its base. In the absence of base signal, it gets turned off completely.

The emitter leg of SL100/Cl100 is indicated by a protruding edge in the transistor case. The base is nearest to the emitter while collector lies at other extreme of the casing.

Fig 5: CL100

V. DC Geared Motor:

Gear motors are primarily used to reduce speed in a series of gears, which in turn creates more torque. This is accomplished by an integrated series of gears or a gearbox being attached to the main motor rotor and shaft via a second reduction shaft. The second shaft is then connected to the series of gears or gearbox to create what is known as a series of reduction gears. Generally speaking, the longer the train of reduction gears, the lower the output of the end, or final, gear will be.

Fig 6: DC Geared Motor

VI. L298N Motor Driver Module:

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output. This depends on the voltage used at the motors VCC. The module have an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board.

This module uses two techniques for the control speed and rotation direction of the DC motors. These are H-Bridge – For controlling rotation direction and PWM – For controlling the speed.

H-Bridge Techniques

L298n motor driver module uses the H-Bridge technique to control the direction of rotation of a DC motor. In this technique, H-Bridge controlled DC motor rotating direction by changing the polarity of its input voltage.

An H-Bridge circuit contains four switching elements, like transistors (BJT or MOSFET), with the motor at the center forming an H-like configuration. Input IN1, IN2, IN3, and IN4 pins actually control the switches of the H-Bridge circuit inside L298N IC.

We can change the direction of the current flow by activating two particular switches at the same time, this way we can change the rotation direction of the motor.

Case 1

When S1, S2, S3, and S4 all switches are open then no current goes to the Motor terminals. So, in this condition, the motor is stopped (not working).

L298N Motor Driver Module Working of H-Bridge Case 1

L298N Motor Driver Module Working of H-Bridge Case 1

Case 2

When the switch S1 and S4 are closed, then the motor left terminal is getting a positive (+) voltage and the motor right terminal is getting a negative(-) voltage. So, in this condition motor start rotating in a particular direction (clockwise).

L298N Motor Driver Module Working of H-Bridge Case 2

L298N Motor Driver Module Working of H-Bridge Case 2

Case 3

When S2 and S3 switches are closed, then the right motor terminal is getting a positive (+) voltage and the left motor terminal is getting a negative (-) voltage. So, in this condition motor start rotating in a particular direction (anticlockwise).

L298N Motor Driver Module Working of H-Bridge Case 3

L298N Motor Driver Module Working of H-Bridge Case 3

PWM Techniques

L298n motor driver module uses the PWM technique to control the speed of rotation of a DC motor. In this technique, the speed of a DC motor can be controlled by changing its input voltage. Pulse Width Modulation is a technique where the average value of the input voltage is adjusted by sending a series of ON-OFF pulses. The average voltage is proportional to the width of the pulses, these pulses known as Duty Cycle. If the duty cycle higher, then the average voltage is applied to the DC motor (High Speed), and the lower the duty cycle, the less the average voltage being applied to the dc motor(Low Speed).

Fig 7: L298N

VII. **Li-ion Battery(3.7V)** :

A **lithium-ion battery** or **Li-ion battery** (abbreviated as **LIB**) is a type of rechargeable battery. Li-ion batteries use an intercalated lithium compound as one electrode material, compared to the metallic lithium used in a non-rechargeable lithium battery. The batteries have a high energy density, no memory effect (other than LFP cells)[10] and low self-discharge. They can however be a safety hazard since they contain a flammable electrolyte, and if damaged or incorrectly charged can lead to explosions and fires.

Fig 8: Li-ion Battery

VIII. **SERVO MOTOR SG-90**:

A servomotor (or servo motor) is a simple electric motor, controlled with the help of servomechanism. If the motor as a controlled device, associated with servomechanism is DC motor, then it is commonly known as a DC Servo Motor. If AC operates the controlled motor, it is known as a AC Servo Motor.

A servomotor is a linear actuator or rotary actuator that allows for precise control of linear or angular position, acceleration, and velocity. It consists of a motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in

terms of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

Fig 9: Servo Motor

IX. Jumper Wire:

A **jump wire** (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable – named for one manufacturer of them) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components.

Fig 10: Jumper Wire

x. Air Pump: This is a DC motor driver air pump or air inflator. This is a small air pump and can be used in any DIY projects like aquariums

to supply oxygen. It can be controlled by any controller like Arduino, Raspberry Pi, AVR, PIC or any other controller. The pump is very easy to assemble, interface and a low power device.

Fig 11: Air Pump

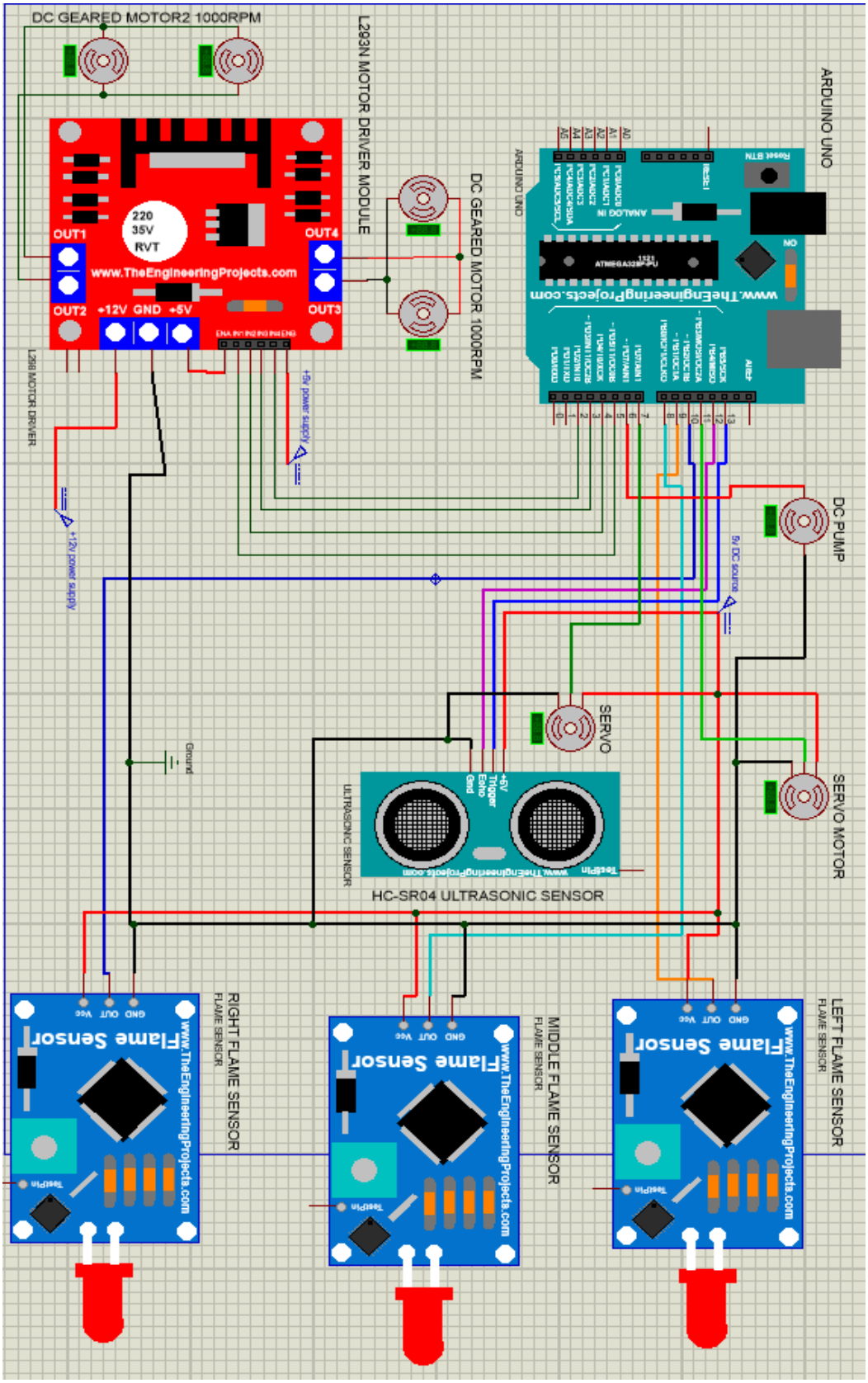
Working Principle:

This prototype project is based on detecting fire and also obstacles by IR sensor and Ultrasonic sensor and fighting with fire to douse it and help the firefighters. An IR sensor is an optoelectronic device which has spectral sensitivity in the infrared wavelength (780nm - 1100 µm). When a fire occurs, basically infrared radiation is generated as heat. So the sensor detects radiation coming from the fire and creates changes in the voltage as per the input radiation quantity in a defined angle range. Next coming to the ultrasonic sensor. An ultrasonic sensor is an electronic device that measures the distance from an object using ultrasonic waves. A transducer is used to send and receive ultrasonic waves and calculate the distance by measuring the change in voltage. The ultrasonic sensor sends ultrasonic waves through the transmitter and receives the reflective signal through the receiver end coming from the obstacle. Then the Arduino calculates the time taken by the ultrasonic wave from transmitter to receiver. Then using a mathematical formula it calculates the distance from the obstacle to the sensor. The formula

is D= (1/2) T * C (where D is the distance, T is the time taken, and C is the speed of sound (343 metres/second).

In our project, we used one ultrasonic sensor and three IR sensors (IR sensor 1, IR sensor 2 and IR sensor 3). We also used two servo motors (SG-90) to rotate the ultrasonic sensor and the water outlet. When we switch on the power supply the robot starts to find any sign of fire. In case a fire is detected, the robot first takes a turn towards the fire. Based on the left, right and middle IR sensor's value the Arduino will understand in which direction the robot needs to rotate. Then Arduino set a pin high as per the code and this will cause to supply voltage in the motor driver module's channel. Thus the DC motors will rotate and wheels. Then the code will set the fire value as HIGH. For that, a putt_off_fire function will execute caused douse the fire and spread water on the fire using the servo motor. While executing the putt_off_fire function the robot continuously checked whether the fire is still on or not. Until the fire will not completely douse, the robot stayed in its position and sprays water on it. After complete douse the fire the fire==True statement will be false. At that time the code will execute the distance measurement function to measure distance. First it will look middle and take distance value. Following this, the robot takes the right look value and the left look value. Then it will compare these three distance values and as per the algorithm, Arduino will take the decision that which distance is largest. Then Arduino sends the signal to the motor driver module and the robot will take a turn through the larger distance area. That way this robot avoids all the obstacles. While measuring the distance value and going forward avoiding obstacles if any of the flame sensors detect fire the robot immediately stops, makes a turn towards the fire and opens the pump.

Circuit Schematic



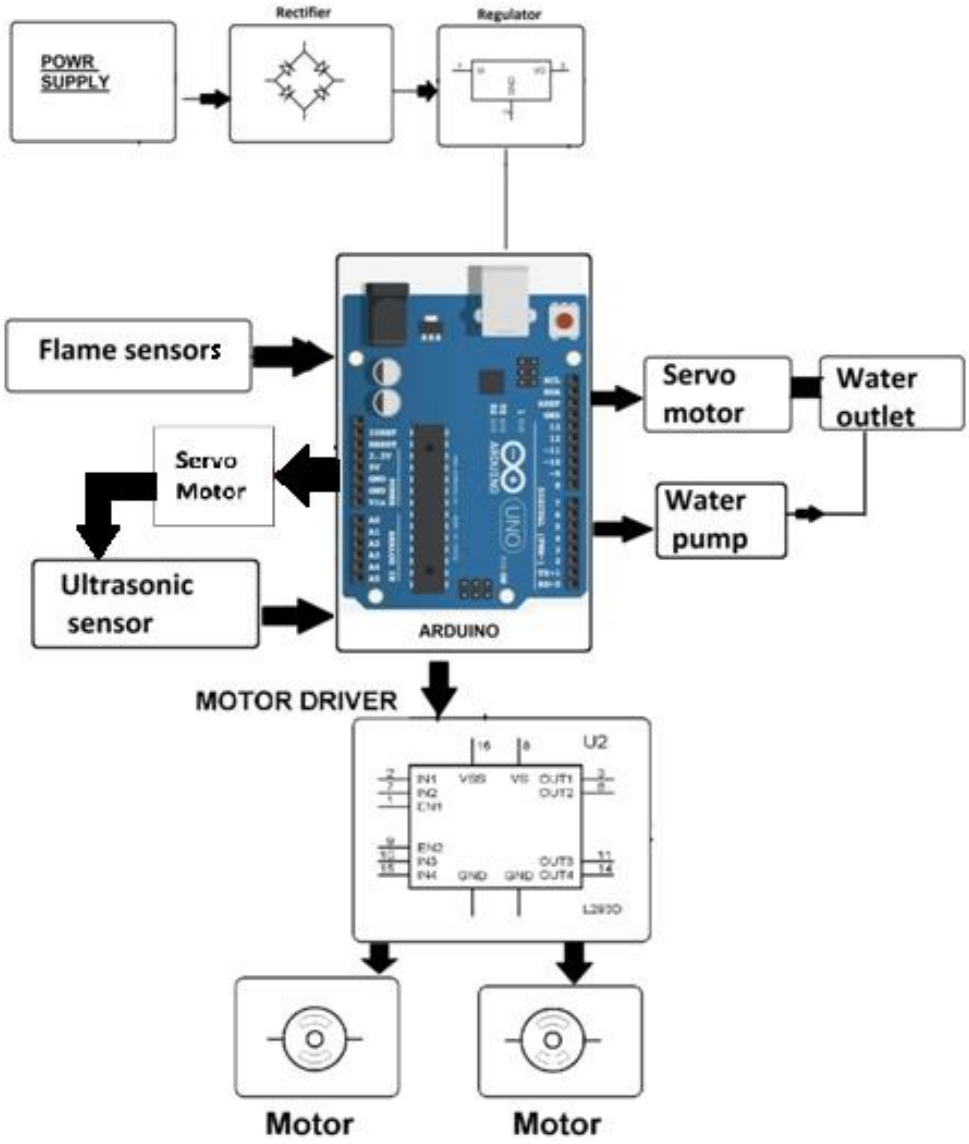
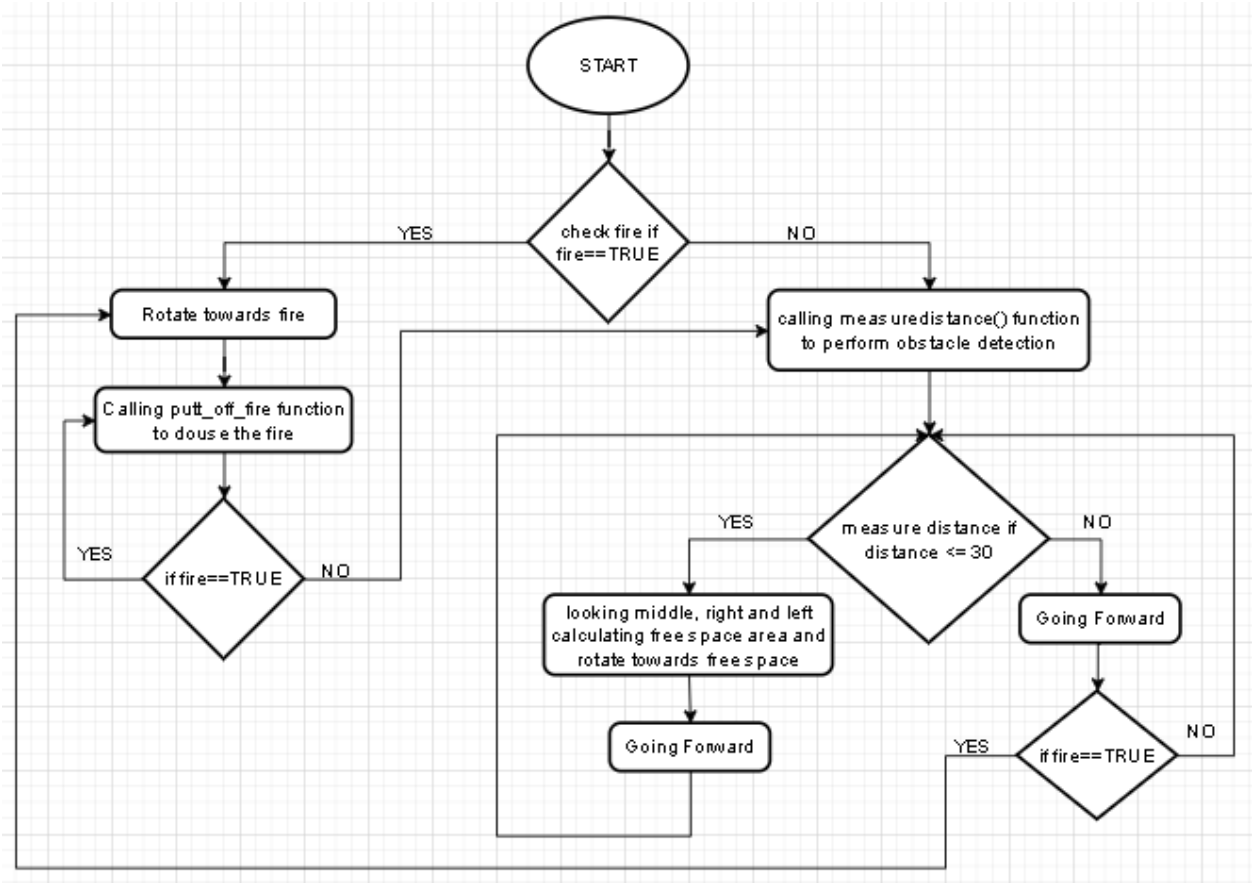
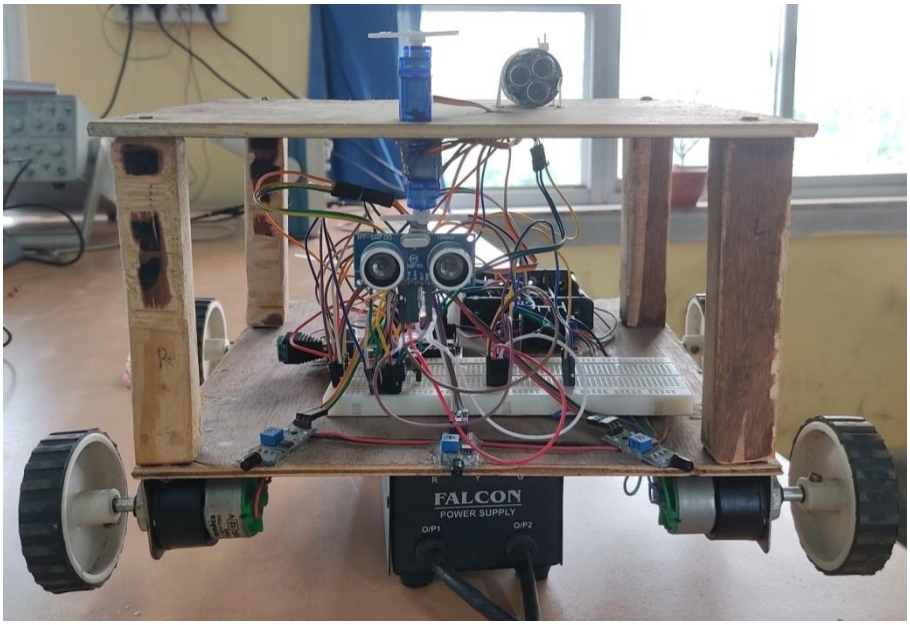


Fig 11: Block Diagram of this project


START
check fire if fire==TRUE
YES
NO
Rotate towards fire
Calling putt_off_fire function to douse the fire
if fire==TRUE
YES
NO
calling measuredistance() function to perform obstacle detection
measure distance if distance <= 30
YES
NO
looking middle, right and left calculating free space area and rotate towards free space
Going Forward
Going Forward
if fire==TRUE
YES
NO


FALCON
POWER SUPPLY

Now, whenever there is a need to stop the vehicle all one has to do is place any other finger except the one previously stored in the biometric device so that it doesn't matches with the stored data. As a result the output of the arduino will become low now and motor will be stopped.

First when we turn on the power supply, the Arduino starts taking data from fire sensors. Three fire sensors were attached with Arduino's three digital pins 8, 9 and 10. VCC and ground are common for Arduino and sensors. The fire sensors detect fire and send data through digital pins. those data flow to Arduino and Arduino then takes the decision based on the code. Then Arduino set some digital pins (2, 3, 4, 5) HIGH and LOW as per coding. these pins are connected to the motor driver's module. For changing the signal of those Arduino digital pins, the motor driver module's state will also be changing. when the front flame sensor detects a fire, the Arduino set 5 & 3 HIGH. So for that through motor driver channels 1 and 2 current flows in the forward direction. The same is going on for left and right flame sensors only pin states are changing as per coding. When the fire is not detected ultrasonic sensor is on. The ultrasonic sensor has an echo and trigger pin and VCC-GND pin for the power supply. Echo pin connected with Arduino 12 no. digital pin. The trigger pin is connected with Arduino 11no. digital pin. Now for a certain time echo pin is set HIGH and he low. Now set the trigger pin HIGH and trigger pins calculates the deflect ultrasonic waves. Then the values goes to arduino through digital pins and jumper wires. Now as per code arduino calculate distance from obstacle and takes decision. If the obstacle is appear infront of the robot the arduino compare this value with right look distance and left look distance. The arduino compare which value is greater and as per that the Arduino set those digital pins high that are connected with motor drive module. If robot needs to move in right direction then pin 4 will

be HIGH and pin 3 will be HIGH. Rest of the movements as follows only the pins value will change. After that we will also connect servo with pin 7 which is rotate the ultrasonic sensor. We have also attached one pump with pin number 6 and one servo for pump by pin no. 11. When fire is detected, robot rotates towards fire and the stops and then set HIGH pin 11 & 7 for spraying water into the fire to douse.

The code of enrolment process and fingerprint authentication is given below.

Code for fingerprint enrolment:

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

Servo myservopump;  //servo attached with pump

Servo myservo;      //servo attached with ultrasonic

#define echoPin 12

#define trigPin 13

int pos = 0;

boolean fire = false;

long duration;

int distance;

#define Left 9      // left sensor

#define Right 10    // right sensor

#define Forward 8   //front sensor

#define RMF 2       // left motor line4

#define RMB 3       // left motor line3

#define LMF 4       // right motor line2
```

```
#define LMB 5      // right motor line1
#define pump 6
void setup()
{
  pinMode(Left, INPUT);
  pinMode(Right, INPUT);
  pinMode(Forward, INPUT);
  pinMode(RMF, OUTPUT);
  pinMode(RMB, OUTPUT);
  pinMode(LMF, OUTPUT);
  pinMode(LMB, OUTPUT);
  pinMode(pump, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  myservo.attach(7);
  myservopump.attach(11);
  myservopump.write(90);
  Serial.begin(9600);
}

void loop()
{
   checkfire();
```

```
}
void measuredistance(){
    myservo.write(90);
    calDis();
    int distanceRight = 0;
    int distanceLeft = 0;
    if (distance <= 30)
    {
    moveStop();
    delay(300);
    moveBackward();
    delay(300);
    moveStop();
    delay(300);
    distanceRight = lookRight();
    delay(300);
    distanceLeft = lookLeft();
    delay(300);
    myservo.write(90);

    if (distance >= distanceLeft)
    {
     turnRight();
```

```
      moveStop();
      myservo.write(90);
     }
     else
     {
      turnLeft();
      moveStop();
      myservo.write(90);
     }
  }
  else{
    moveForward();
    checkfire();
  }
  delay(10);
}

void put_off_fire()
{
  if(digitalRead(Left) ==0 || digitalRead(Right)==0 ||
digitalRead(Forward) ==0)
  {
    delay (500);
```

```
   digitalWrite(RMF, LOW);
   digitalWrite(RMB, LOW);
   digitalWrite(LMF, LOW);
   digitalWrite(LMB, LOW);
  digitalWrite(pump, HIGH);
  delay(500);
   for (pos = 50; pos <= 130; pos += 1) {
   myservopump.write(pos);
   delay(10);
  }
  for (pos = 130; pos >= 50; pos -= 1) {
   myservopump.write(pos);
   delay(10);
  }
  }
  else
  {
  fire=false;
  digitalWrite(pump, LOW);
  }
}

int calDis(){
```

```
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  return distance;
  Serial.print("Distance: ");
  Serial.println(distance);
}

int lookRight(){
  myservo.write(50);
  delay(500);
  int distance = calDis();
  delay(100);
  return distance;
}

int lookLeft(){
  myservo.write(150);
  delay(500);
```

```
  int distance = calDis();
  delay(100);
  return distance;
}
void moveStop(){
  digitalWrite(RMF, LOW);
  digitalWrite(LMF, LOW);
  digitalWrite(RMB, LOW);
  digitalWrite(LMB, LOW);
}

void moveBackward(){
  digitalWrite(LMB, HIGH);
  digitalWrite(RMB, HIGH);
  digitalWrite(LMF, LOW);
  digitalWrite(RMF, LOW);
}

void moveForward(){
    digitalWrite(LMF, HIGH);
    digitalWrite(RMF, HIGH);
    digitalWrite(LMB, LOW);
    digitalWrite(RMB, LOW);
```

```
}

void turnRight(){
  digitalWrite(LMF, HIGH);
  digitalWrite(RMB, HIGH);
  digitalWrite(LMB, LOW);
  digitalWrite(RMF, LOW);
  delay(500);
  digitalWrite(LMF, HIGH);
  digitalWrite(RMF, HIGH);
  digitalWrite(LMB, LOW);
  digitalWrite(RMB, LOW);
  delay(500);
}

void turnLeft()
{
  digitalWrite(LMF, LOW);
  digitalWrite(RMB, LOW);
  digitalWrite(LMB, HIGH);
  digitalWrite(RMF, HIGH);
  delay(500);
  digitalWrite(LMF, HIGH);
```

```
  digitalWrite(RMF, HIGH);
  digitalWrite(LMB, LOW);
  digitalWrite(RMB, LOW);
  delay(500);
}

void checkfire()
{
 myservopump.write(90);
   if (digitalRead(Left) ==1 && digitalRead(Right)==1 &&
digitalRead(Forward) ==1)
   {
   digitalWrite(RMF, HIGH);
   digitalWrite(RMB, LOW);
   digitalWrite(LMF, HIGH);
   digitalWrite(LMB, LOW);
   fire = false;
   }

   else if (digitalRead(Forward) ==0)
   {
   digitalWrite(RMF, HIGH);
   digitalWrite(RMB, LOW);
```

```
digitalWrite(LMF, HIGH);
digitalWrite(LMB, LOW);
fire = true;
}

else if (digitalRead(Left) ==0)
{
digitalWrite(RMF, HIGH);
digitalWrite(RMB, LOW);
digitalWrite(LMF, LOW);
digitalWrite(LMB, LOW);
fire = true;
}

else if (digitalRead(Right) ==0)
{
digitalWrite(RMF, LOW);
digitalWrite(RMB, LOW);
digitalWrite(LMF, HIGH);
digitalWrite(LMB, LOW);
fire = true;
}
```

```
delay(300);      //change this value to increase the distance


  while (fire == true)
   {
       put_off_fire();
   }
  while (fire == false)
   {
    measuredistance();
   }
}
```

CONCLUSION

Fire incident is a disaster that can potentially cause the loss of life, property damage and permanent disability to the affected victim. So, viewing from the business side this robot is more profitable. Although the overall cost is low and all the components are easily available in the market. All these valuable features makes this project very efficient and value for money product. As per the sketch of design and enforcement of a fire-fighting device that moves towards the fire and pumps out gas to extinguish the fire is presented in this project. The project explained how to interface ARM7 with different components. The system may be useful for accompaning fire fighters and preventing an outbreak. This is an one of the ample opportunity to automation. It will be used in the location or sites where it is impossible to reach or dangerous for humans.

FUTURE SCOPE:

The development of sensor networks and the maturity of robotics suggests that we can use mobile agents for tasks that involve perception of an external stimulus and reacting to the stimulus, even when the reaction involves a significant amount of mechanical actions. However, there has been research on many of these pieces in different contexts, e.g., coordination among mobile agents, techniques for detecting and avoiding obstacles. It will be both interesting and challenging to put all this together into a practical, autonomous fire fighting service.

This project has been motivated by the desire to design a system that can detect fires and intervention. In the present condition it can extinguish fire only in the way and not in all the rooms. It can be extended to a real fire extinguisher by replacing the fan by a carbon-

di-oxide carrier and by making it to extinguish fires of all the room using microprogramming. This provides us the opportunity to pass on to robots tasks that traditionally humans had to do but were inherently life threatening. Fire-fighting is an obvious candidate for such automation. Given the number of lives lost regularly in fire-fighting, the system we envision is crying for adoption. Of course, this project has only scratched the surface. As in the design simplifications and the implementation constraints in suggest, our project is very much a proof-of-concept. In particular, a practical autonomous fire-fighting system must include a collection of robots, communicating and cooperating in the mission; furthermore, such a system requirs facilities for going through obstacles in the presence of fire, and ability to receive instructions on-the fly during an operation. All such concerns were outside the scope of this project. However, there has been research on many of these pieces in different contexts, e.g., coordination among mobile agents, techniques for detecting and avoiding obstacles, on-the-fly communication between humans and mobile agents, etc. It will be both interesting and challenging to put all this together into a practical, autonomous fire-fighting service.

REFERENCES:

[1] William Dubel, Hector Gongora, Kevin Bechtold and Daisy Diaz, "An Autonomous Firefighting Robot".

[**2]** University Malaysia Perlis, UNIMAP, "Fire Fighting Robot Competition, Theme & Rules", 2009.

[3] J. Suresh, "Fire-fighting robot," 2017 International Conference on Computational Intelligence in Data Science(ICCIDS), Chennai, 2017.

[4] Cease Fire: The Fire Fighting Robot: 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)

[5] Dr.Wael R. Abdulmajeed; Dr.Ali I.Mahdi; Karzan M Taqi, "Human Wireless Controlling Fire Fighting Robot (Ffr) With 3-Axis Hose." International Journal of Advanced Computer Technology (IJACT) ISSN: 2319- 7900

[6] A Fully Automated Fire Fighting Robot, Aman Sharma,1239,sector 9a,Gurgaon, Haryana – 122001

[7] J. Raju, S. S. Mohammed, J. V. Paul, G. A. John and D. S. Nair, Development and implementation of arduino microcontroller based dual mode fire extinguishing robot, IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), 2017

[8] D.J. Pack, A.M. Mankowski, and G.J. Freeman, A FireFighting Robot and Its Impact on Educational Outcomes

[9] T. AlHaza,A. Alsadoon,Z. Alhusinan,M. Jarwali,K. Alsaif. New Concept for Indoor Fire Fighting Robot[J]. ElsevierLtd,2015,195

[10] K. Altaf, A. Akbar and B. Ijaz, "Design and Construction of an Autonomous Fire Fighting Robot," 2007 International Conference on Information and Emerging Technologies, Karachi, 2007.