AUTOMATIC FIRE FIGHTING ROBOT SYSTEM

A project report submitted in partial fulfillment of the requirements

for the award of credits to

Open-Source Hardware Tools for Electronics Engineers

a skill-oriented course of

Bachelor of Technology

In

ELECTRONICS & COMMUNICATION ENGINEERING

By

P. KRISHNA CHAITANYA
21BQ1A04A3

K. SRI SIVA KOTI REDDY
21BQ1A0459

K. SARADHI SAJEEV
21BQ1A0469

N.M.D. SUFIYAN
21BQ1A0491

K. MOHAN KRISHNA
22BQ5A0429

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Approved by AICTE and permanently affiliated to JNTUK)

Accredited by NBA and NAAC

NAMBUR (V), PEDAKAKANI (M), GUNTUR-522 508

JUNE 2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY: NAMBUR

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA

CERTIFICATE

This is to certify that the project titled "**AUTOMATIC FIRE FIGHTING ROBOT SYSTEM"** is a bonafide record of work done by **Mr. PINNIKA KRISHNA CHAITANYA , K.SRI SIVA KOTI REDDY, K. SARADHI SAJEEV , N.M.D.SUFIYAN , K.MOHAN KRISHNA** under the guidance of **Mrs.T.VINEELA, Assistant Professor** in partial fulfillment of the requirement for the award of credits to **Open-Source Hardware Tools for Electronics Engineers -** a skill-oriented course of Bachelor of Technology in Electronics and Communication Engineering, JNTUK during the academic year 2022–23.

Mrs. T. Vineela
Course Instructor

Dr. M. Y. Bhanu Murthy
Head of the Department

DECLARATION

We, **PINNIKA KRISHNA CHAITANYA, K.SRI SIVA KOTI REDDY, K. SARADHI SAJEEV, N.M.D.SUFIYAN , K.MOHAN KRISHNA** hereby declare that the project report entitled **"Open-Source Hardware Tools for Electronics Engineers"** done by me under the guidance of **Mrs.T.VINEELA, Assistant Professor, Department of Electronics and Communication Engineering** is submitted by partial fulfilment of requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING**

Date: -
Place: - VVIT, Nambur

Signature of the candidates

PINNIKA KRISHNA CHAITANYA (21BQ1A04A3)

KARASANI SRI SIVA KOTI REDDY (21BQ1A0459)

KURAGANTI SARADHI SAJEEV (21BQ1A0469)

NAVALI MOHAMMAD SUFIYAN (21BQ1A0491)

KONANGI MOHAN KRISHNA (22BQ5A0429)

ACKNOWLEDGEMENT

We express our sincere thanks wherever it is due

We express our sincere thanks to the Chairman, Vasireddy Venkatadri Institute of Technology, **Sri Vasireddy Vidya Sagar** for providing us well equipped infrastructure and environment.

We thank **Dr. Y. Mallikarjuna Reddy**, Principal, Vasireddy Venkatadri Institute of Technology, Nambur, for providing us the resources for carrying out the project.

We express our sincere thanks to **Dr. K. Giribabu**, Dean of Studies for providing support and stimulating environment for developing the project.

Our sincere thanks to **Dr. M. Y. Bhanumurthy**, Head of the Department, Department of ECE, for his co-operation and guidance which help us to make our project successful and complete in all aspects.

We also express our sincere thanks and are grateful to our guide **Mrs.T. Vineela**, Assistant Professor, Department of ECE, for motivating us to make our project successful and fully complete. We are grateful for her precious guidance and suggestions.

We also place our floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the project.

NAME OF THE CANDIDATES: PINNIKA KRISHNA CHAITANYA

KARASANI SRI SIVA KOTI REDDY

KURAGANI SARADHI SAJEEV

NAVALI MOHAMMAD SUFIYAN

KONANGI MOHAN KRISHNA

CONTENTS

LIST OF CONTENTS PAGE NO

LIST OF FIGURES

ABSTRACT

Fire Fighting is an important but dangerous occupation. Robots are designed to find a fire, before it rages out of control, could one day work with fire fighters greatly reducing the risk of injury to victims. I Fire Fighting Robot Competition is a contest purposely to simulate the real-world operation of an autonomous robot rescuing 10 victims (table tennis balls) and stop 5 fires (emergency candles) in a house within three minutes. The robot development is consisting of three elements which is the hardware, electronic, and programming.

The robot have three DC motor, two for driving system and another single DC motor for ball suction subsystem and the fire blowing subsystem. Various sensors are also interfaced with Arduino as a feedback to the robot such as photoelectric sensors, fiber optic sensor and RGB color sensors. LCD display also gives the graphical information of the robot status to the user. For the programming part, C++ language is used to determine the robot action gain from the sensors input.

CHAPTER 1

INTRODUCTON

1.1MOTIVATION

Cultural property management is entrusted with the responsibility of protecting and preserving an institution's buildings, collections, operations and occupants. Constant attention is required to minimize adverse impact due to climate, pollution, theft, vandalism, insects, mold and fire. Because of the speed and totality of the destructive forces of fire, it constitutes one of the more serious threats. Vandalized or environmentally damaged structures can be repaired and stolen objects recovered. Items destroyed by fire, however, are gone forever. An uncontrolled fire can obliterate an entire room's contents within a few minutes and completely burn out a building in a couple of hours. Hence it has become very necessary to control and cease the fire to protect the Life and costlier things. For that we purposed to design and fabricate the fire-fighting robot.

Autonomous robots can act on their own, independent of any controller. The basic idea is to program the robot to respond in a certain way to outside stimuli. The very simple bump-and-go robot is a good illustration of how this works. This sort of robot has a sensor to detect obstacles. When you turn the robot on, it zips along in a straight line. When it finally hits an obstacle, the impact is on sensors, i.e. sensors may get damaged. Using Ultrasonic sensor and programming logic, the robot is guided to turn right and move forward again, when the robot finds an obstacle in its way. In this way, the robot changes direction any time it encounters an obstacle. Advanced robots use more elaborate versions of this same idea. Roboticists create new programs and sensor systems to make robots smarter and more perceptive. Today, these automated robot systems can very effectively navigate in a variety of environments. The first step in a firefighting operation is reconnaissance to search for the origin of the fire and to identify the specific risks. Fires can be extinguished by water, fuel or oxidant removal, or chemical flame inhibition; though, because fires are classified depending on the elements involved, such as grease, paper, electrical, etcetera, a specific type of fire extinguisher may be required.

1.2 PROJECT OVERVIEW

The project is designed to develop a fire fighting robot using Arduino uno . The robotic vehicle is loaded with water pump which is controlled by servos. An ATMega 328 microcontroller is used for the desired operation. At the transmitting end using commands are sent to the receiver to control the movement of the robot either to move forward, and left or right etc. At the receiving end tow motors are interfaced to the microcontroller where two of them are used for the movement of the vehicle and the one to position the robot. The ultrasonic sensor adequate range with obstacle detection, while the receiver driver module used to drive DC motors via motor driver IC for necessary work. A water tank along with water pump is mounted on the robot body and its operation is carried out from the microcontroller output through appropriate command from the transmitting end. The whole operation is controlled by an ATmega 328 microcontroller. A motor driver IC is interfaced to the microcontroller through which the controller drives the motors, three ir flame sensors are fixed on robot chassis to sense the fire and to reach the destination to put off the fire.

1.3 COMPONENTS OVERVIEW

This system uses the following components.

1.3.1 Microcontroller

Microcontroller can be described as a computer embedded on a rather small circuit board. To describe the function of a microcontroller more precisely it is a single chip that can perform various calculations and task and send/receive signals from other devices via the available pins. Precisely what tasks and communication with the world it does, is what is governed by what instructions we give to the Microcontroller. It is this job of telling the chip what to do, is what we refer to as programming on it. However, the microcontroller by itself, cannot accomplish much, it needs several external inputs, power, for one, a steady clock signal, for another. Also, the job of programming it has to be accomplished by an external circuit. So a microcontroller is used along with a circuit which provides these things to it; this combination is called a microcontroller board. The Arduino Uno that you have received is one such microcontroller board. The actual microcontroller at its heart is the chip called Atmega328. The advantages that Arduino offers over other microcontroller boards are largely in terms of reliability of the circuit hardware as well as the ease of programming and using it.

1.3.2 Power Supply

7805 is a voltage regulation IC which is used to supply 5V Direct current to the microcontroller

1.3.3 Motor Driver IC

L293D is a dual H-bridge motor driver integrated circuit (IC). They are use to control they the 4-motor used in project. There are 2 motor driver IC used in the project one to control front motor and other for rear motors. L293d IC is known as a motor driver. It is a low voltage operating device like other ICs. The other ICs could have the same functions like L293d but they cannot provide the high voltage to the motor.

L293d provides the continuous bidirectional Direct Current to the Motor. The Polarity of current can change at any time without affecting the whole IC or any other device in the circuit. L293d has an internal H-bridge installed for two motors.

1.3.4 Computer Interface

Finally, this project uses IDE compiler for interfacing the arduino with a PC. This interface is used to setup and compile the Arduino

Software Installation

1.4.1 Software Description

The software used by the Arduino is Arduino IDE. The Arduino IDE is a cross platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on a command line interface. Although building on command-line is possible if required with some third-party tools such as Inc.

1.4.2 Installation :

Visit the Arduino website: https://www.arduino.cc/en/ Main/Software

Then select your operating system and download the latest version of arduino IDE. Download the package, and run the executable file to start installation. It will download the driver needed to run Arduino IDE. After downloading, follow the prompts to install. After installing, you will see Arduino icon on your desktop and double click to open it. When the Arduino IDE first opens, it looks like this

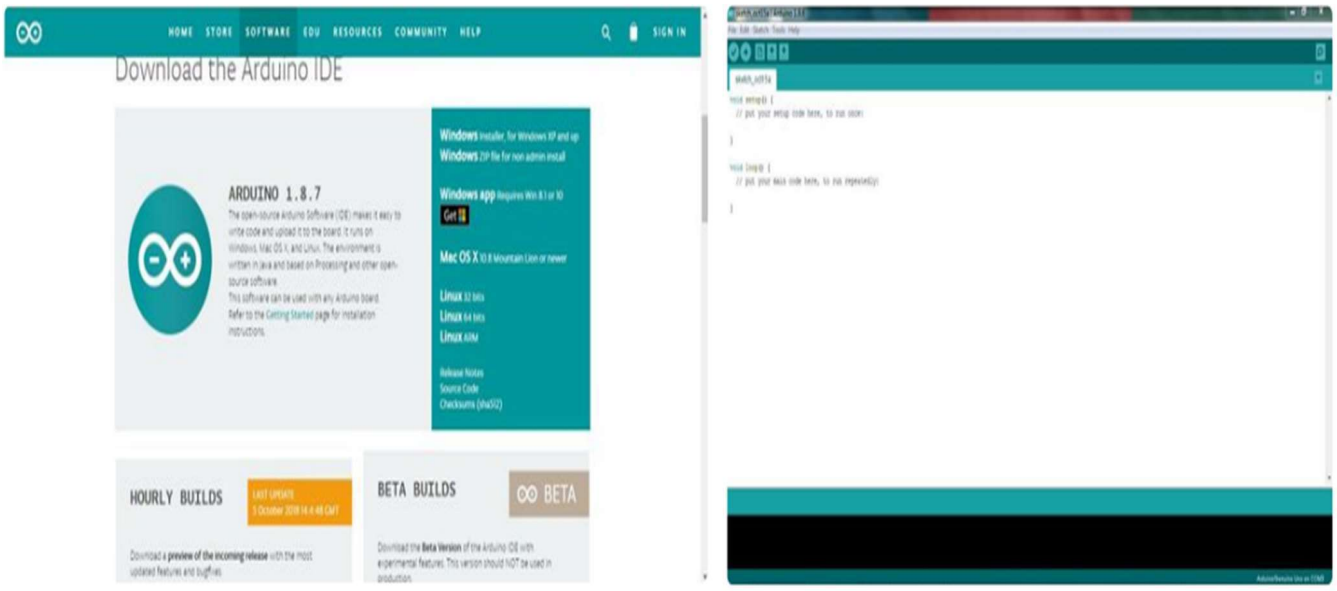


Fig 1.1 *: Installation*

1.4.3 Launch Arduino IDE:

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

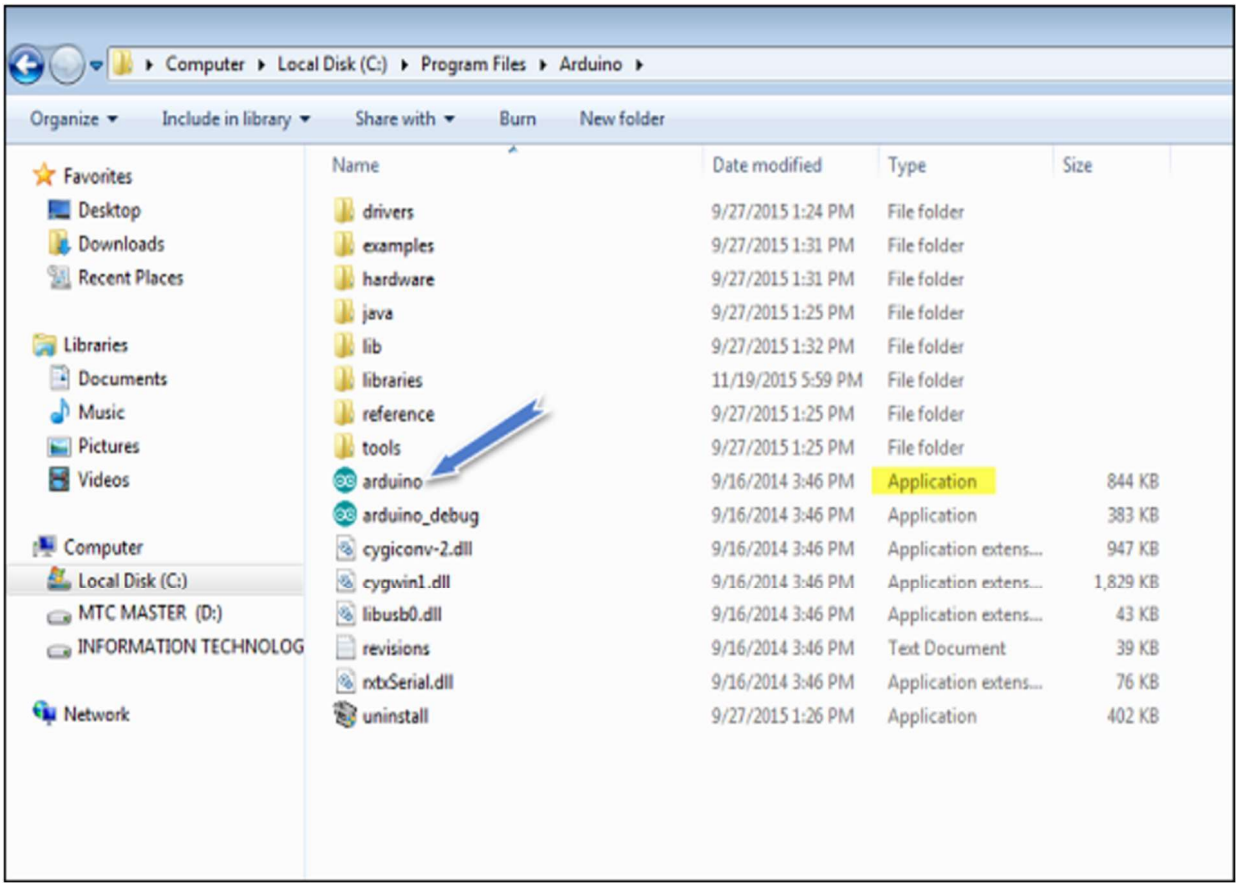


Fig 1.2 : *Launching Arduino IDE*

1.4.4 Opening project

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New.**

To open an existing project example, select File → Example → Basics → Blink

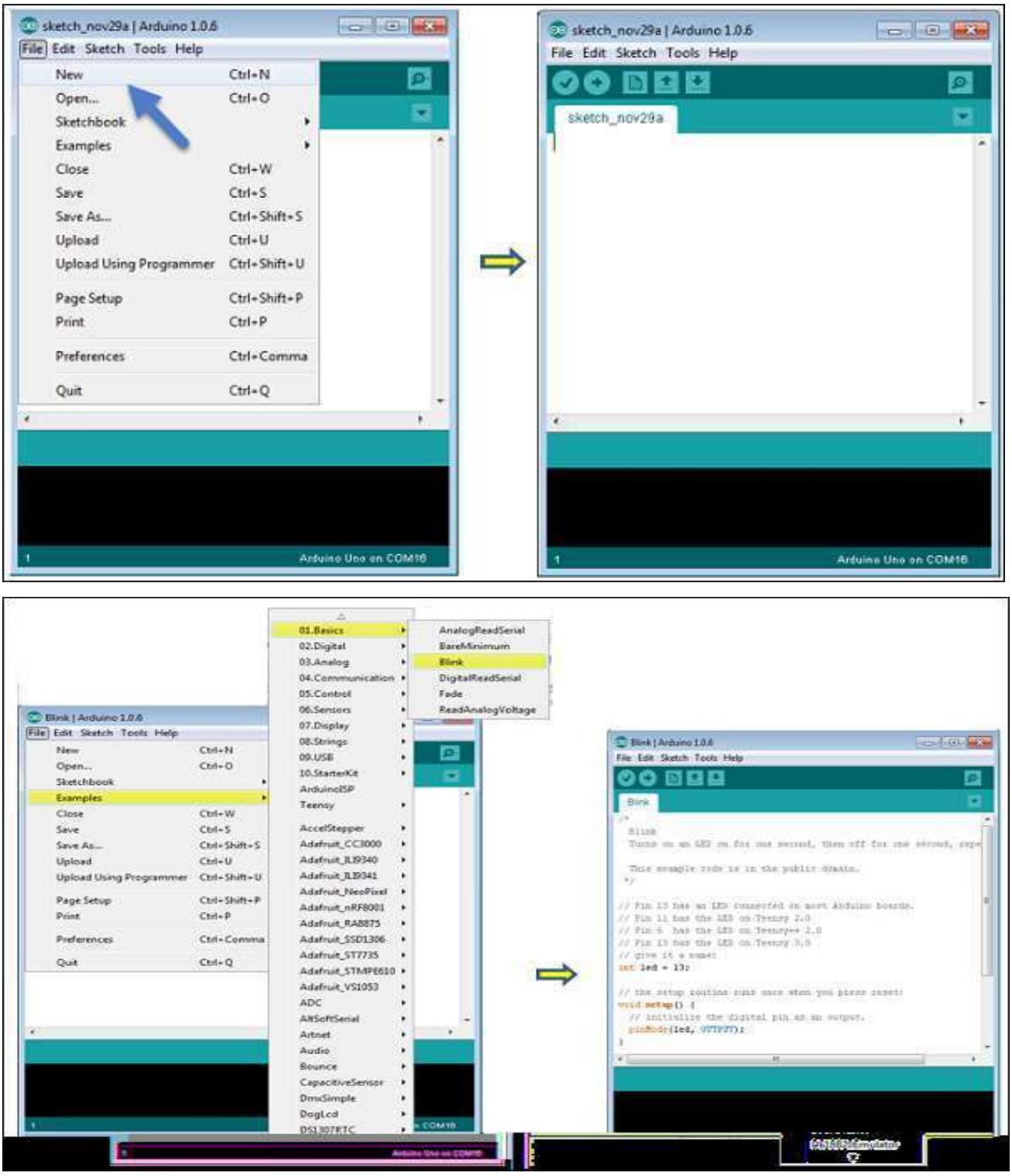


Fig 1.3: *Opening project*

1.4.5 Selecting Arduino board

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

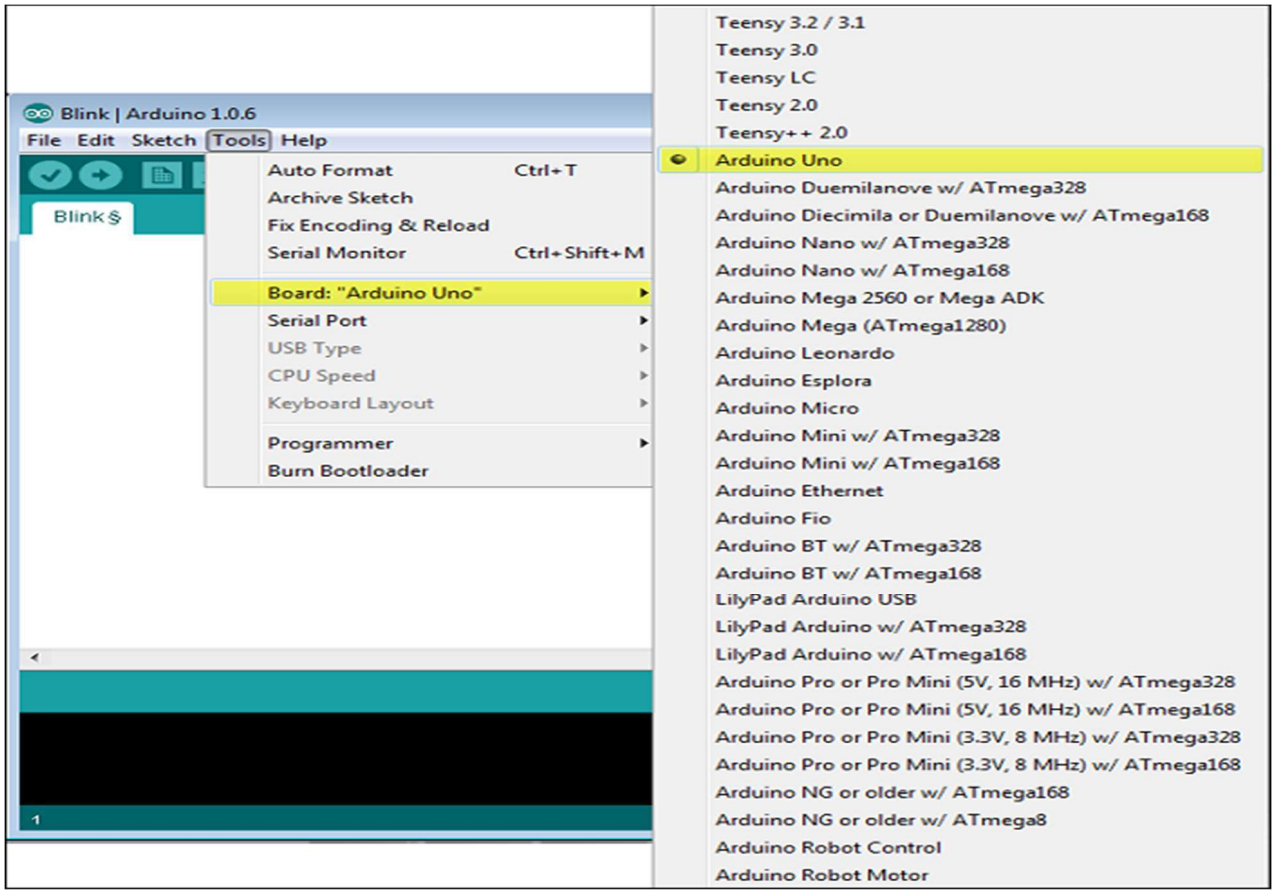


Fig 1.4: *Selecting Arduino Board*

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

1.4.6 Select your serial port

Select the serial device of the Arduino board. Go to **Tools** → **Serial Port** menu. This is likely to be COM3 or higher.

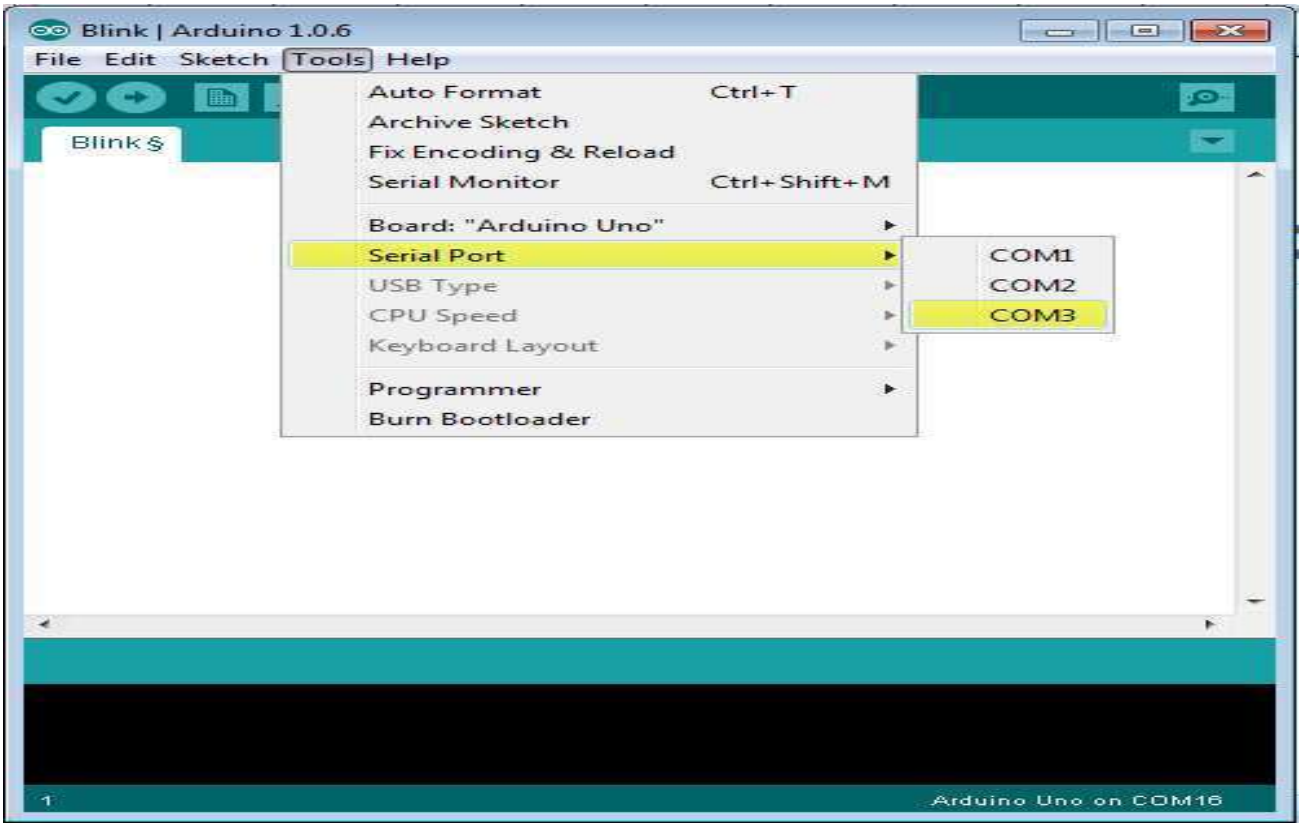
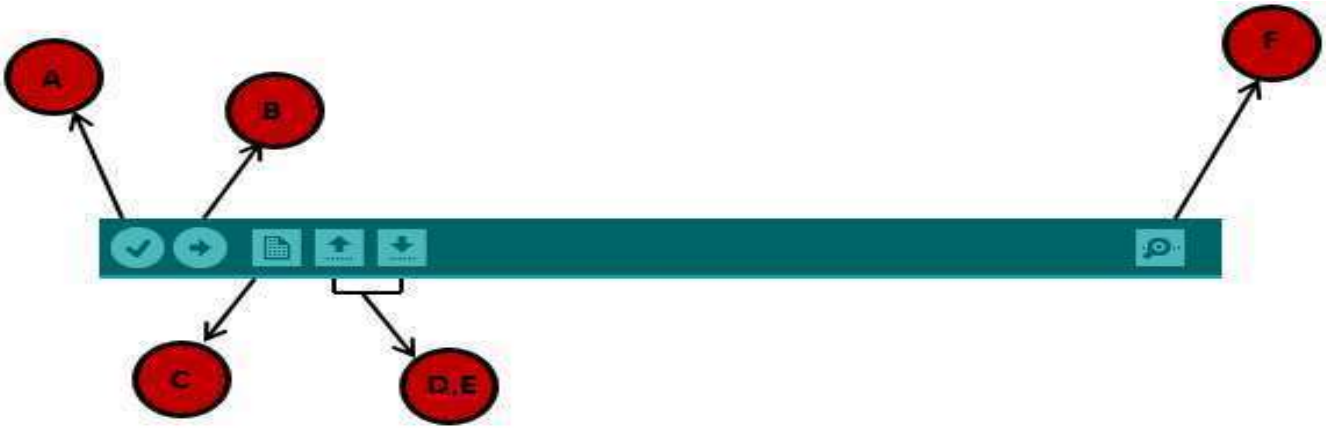


Fig 1.5 : *Selecting the port*

1.4.7 − Upload the program to your board:

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

CHAPTER 2
PROBLEM DEFINITION

As explained in the introduction chapter, the realization of complete potential of the sensors and the wired medium in information transfer is the major issue that the following thesis of the following project deals with.

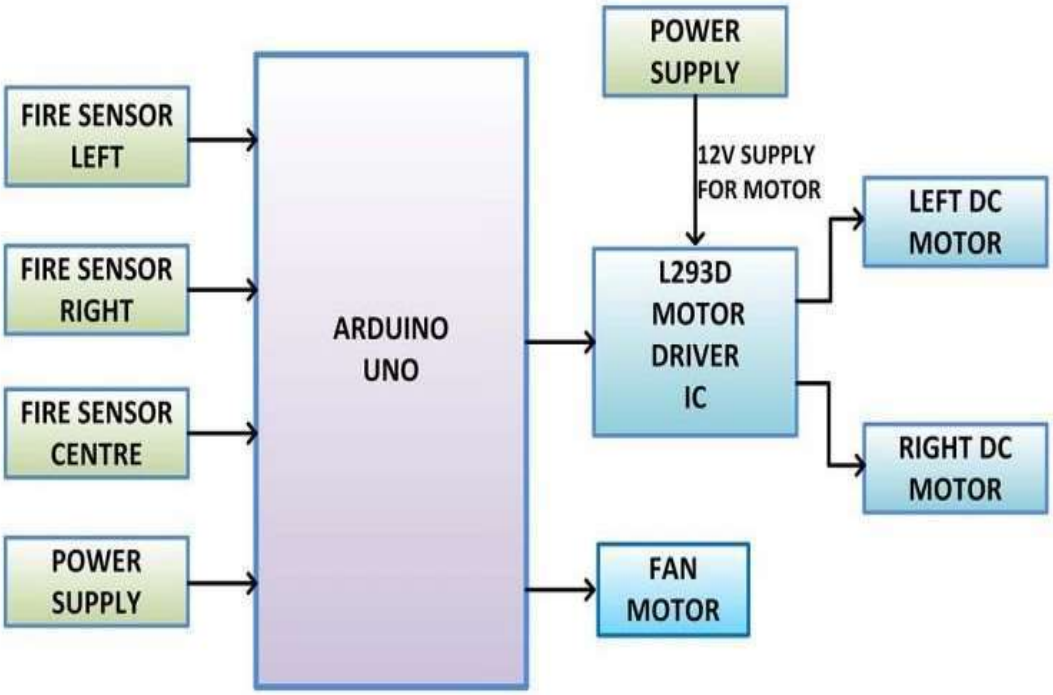


Fig 2.1: *Overview of system*

From Fig 2.1, there are at least five interfacing circuits, L293d driver module, Arduino-uno with Microcontroller, flame sensors, ultrasonic sensors, servo motor and 5v pump. here Arduino uno acts a heart of our project, in the above block diagram we can see that there are three flame sensors and ultrasonic sensor which acts as input interface to the microcontroller and servomotor, pump, driver module acts a output interface to the microcontroller, here the input and output interface can be indicated with the arrow lines with the respective the microcontroller performs with the respective commands and delay which is programmed on Arduino software.

2.1 CIRCUIT DIAGRAM OF THE SYSTEM

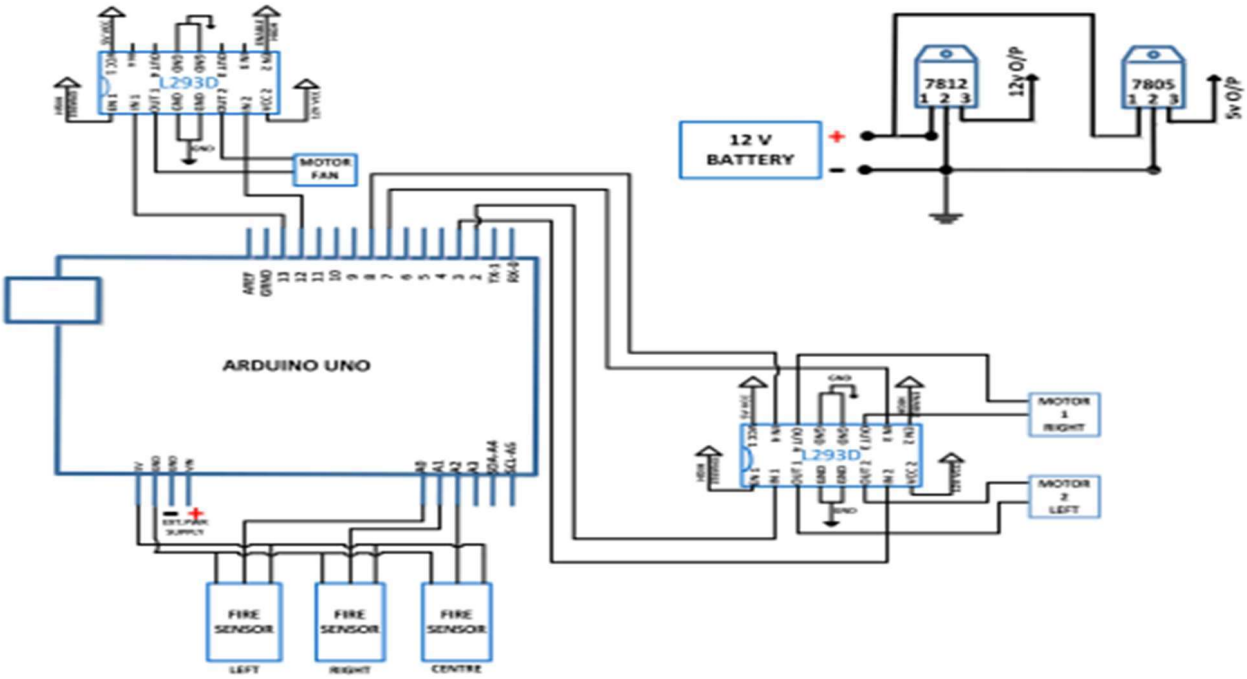


Fig 2.2: *Circuit diagram of the system.*

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 OpAmp to check for change in voltage across the IR Receiver, so that if a fire is detected the output pin 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.

CHAPTER 3
SYSTEM REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENTS

3.1.1 L293D Driver module

The Motor Driver is a module for motors that allows you to control the working speed and direction of two motors simultaneously. This Motor Driver is designed and developed based on L293D IC L293D is a 16 Pin Motor Driver IC. This is designed to provide bidirectional drive currents at voltages from 5 V to 36 V.

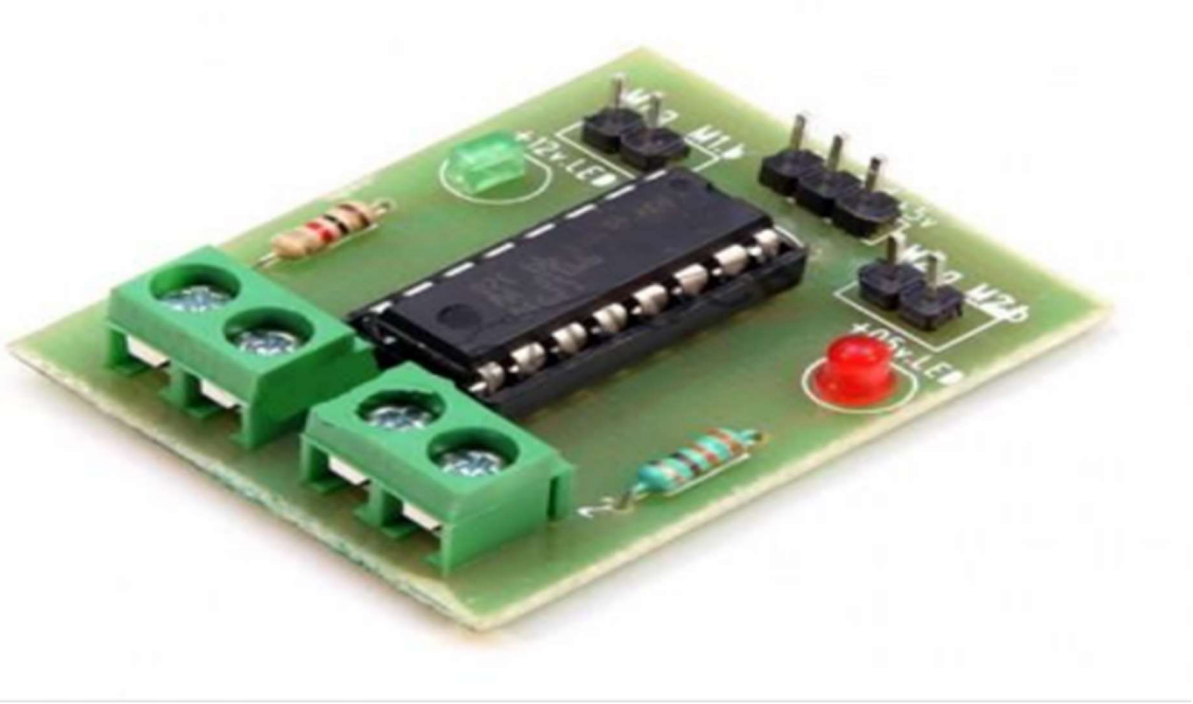


***Fig 3.1**: L293D motor driver module*

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors. L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

Hardware features

- can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible
- Motor voltage Vcc2 (Vs): 4.5V to 36V
- Maximum Peak motor current: 1.2A
- Maximum Continuous Motor Current: 600mA
- Supply Voltage to Vcc1(vss): 4.5V to 7V
- Transition time: 300ns (at 5Vand 24V)
- Automatic Thermal shutdown is available
- Available in 16-pin DIP, TSSOP, SOIC packages

Applications

- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED's can be driven
- Relay Driver module (Latching Relay is possible)

3.1.2 Flame Sensor Module

A flame sensor module that consists of a flame sensor (IR receiver), resistor, capacitor, potentiometer, and comparator LM393 in an integrated circuit. It can detect infrared light with a wavelength ranging from 700nm to 1000nm.The far-infrared flame probe converts the light detected in the form of infrared light into current changes. Working voltage is between 3.3v and 5.2v DC, with a digital output to indicate the presence of a signal.

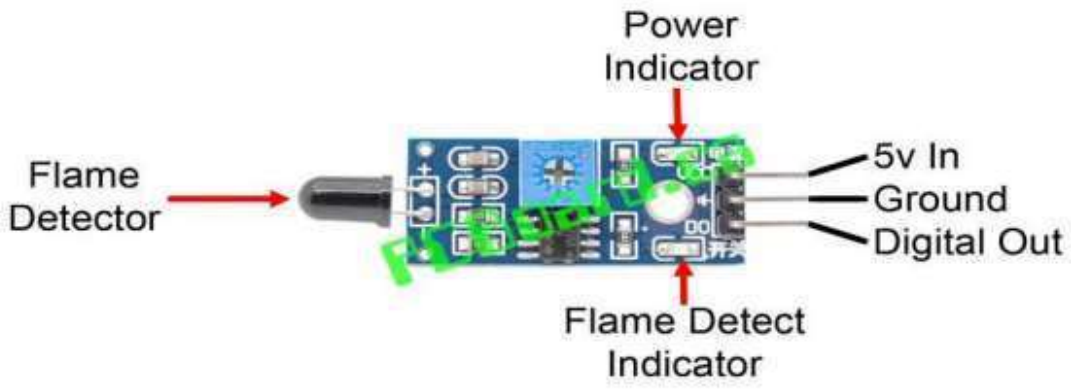


Fig3.2: *flame sensor module*

3.1.3 DC Motor:

Motors convert electrical energy into mechanical energy. A DC motor is an electric motor that runs on direct current (DC) electricity.

Fig 3.3 : dc motor

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate

rotational motion. Direct current (DC) motors are widely used to generate motion in a variety of products. Permanent magnet DC (direct current) motors are enjoying increasing popularity in applications requiring compact size, high torque, high efficiency, and low power consumption.

3.1.4 Water Pump

The water pump is operated at 5v which can be interfaced with Arduino.

Fig 3.4 : *5v water pump*

3.1.5 Servo Motor

A servo is a small DC motor with the following components added: some gear reduction, a position sensor on the motor shaft, and an electronic circuit that controls the motor's operation. In other words, a servo is to a DC motor what the Arduino is the

ATmega microcontroller---components and housing that make the motor easy to use. This will become abundantly clear when we work with unadorned DC motors next week. The gear reduction provided in a servo is large; the basic hobby servo has a 180:1 gear ratio. This means that the DC motor shaft must make 180 revolutions to produce 1 revolution of the servo shaft. This large gear ratio reduces the speed of the servo and proportionately increases its torque. What does this imply about small DC motors? Servo motors are typically used for angular positioning, such as in radio control airplanes. They have a movement range of 0 up to 180 degrees, but some extend up to 210 degrees. Typically, a potentiometer measures the position of the output shaft at all times so the controller can accurately place and maintain its position.

Fig3.5 : *Servo Motor*

3.2 MICROCONTROLLER ATMEGA 328

The Atmel 8-bit AVR RISC-based microcontroller combines 32 KB ISP flash memory with read-while-write capabilities, 1 KB EEPROM, 2 KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byteoriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter

(8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughputs approaching 1 MIPS. ATmega 328 has several different features which make it the most popular device in today's market. These features consist of advanced RISC architecture, good performance, low power consumption, real timer counter having separate oscillator, 6 PWM pins, programmable Serial USART, programming lock for software security, throughput up to 20 MIPS etc. Further details about ATmega 328 will be given later in this section.

Fig3.6: *Arduino Uno*

3.2.1 Applications

Today the ATmega328 is commonly used in many projects and autonomous systems where a simple, low-powered, low-cost micro-controller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino Uno and Arduino Nano models.

3.2.2 Features

- 28-pin AVR Microcontroller
- Flash Program Memory: 32 Kbytes
- EEPROM Data Memory: 1 Kbytes
- SRAM Data Memory: 2 Kbytes
- I/O Pins: 23
- Timers: Two 8-bit / One 16-bit
- A/D Converter: 10-bit Six Channel
- PWM: Six Channels
- RTC: Yes with Separate Oscillator
- MSSP: SPI and I²C Master and Slave Support
- USART: Yes
- External Oscillator: up to 20MHz

The Atmega328 is a very popular microcontroller chip produced by Atmel. It is an 8-bit microcontroller that has 32K of flash memory, 1K of EEPROM, and 2K of internal SRAM. The Atmega328 is one of the microcontroller chips that are used with the popular Arduino Duemilanove boards. The Arduino Demilunove board comes with either 1 of 2 microcontroller chips, the Atmega168 or the Atmega328. Of these 2, the Atmega328 is the upgraded, more advanced chip. Unlike the Atmega168 which has 16K of flash program memory and 512 bytes of internal SRAM, the Atmega328 has 32K of flash program memory and 2K of Internal SRAM.

The Atmega328 has 28 pins, It has 14 digital I/O pins, of which 6 can be used as PWM outputs and 6 analog input pins. These I/O pins account for 20 of the pins.

CHAPTER 4
SYSTEM MODELLING AND DESIGN

4.1 FUNCTIONAL DESCRIPTION

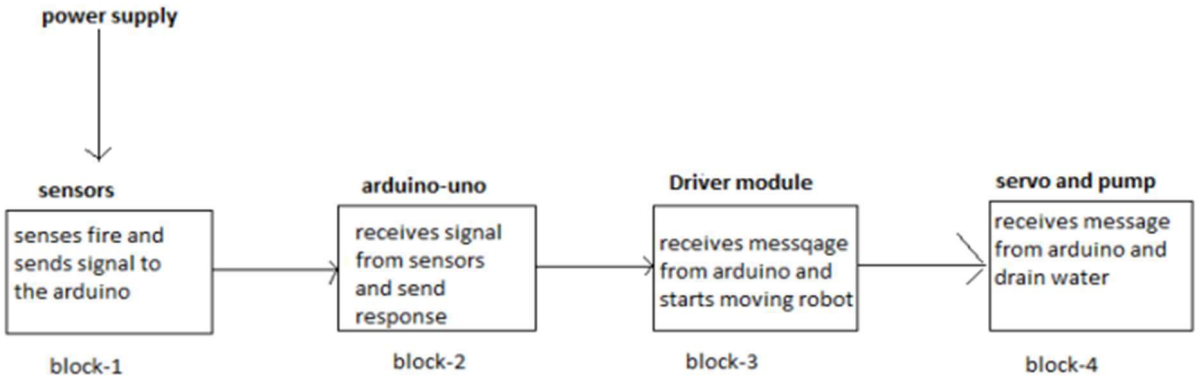


Fig 4.1: *Functional description diagram*

The constituent parts involved in the process are

- sensors
- Arduino with Atmel Atmega328 microcontroller
- L293 driver module
- Servo with pump

First block portrays to be sensors which receives, verifies and forwards the message to the Microcontroller. Micro is the second block. Micro processes the message and sends to the driver module. Driver module behaving as the third constituent part and servo pump acts as fourth part which extinguishes the fire.

4.2 DATA FLOW DIAGRAM

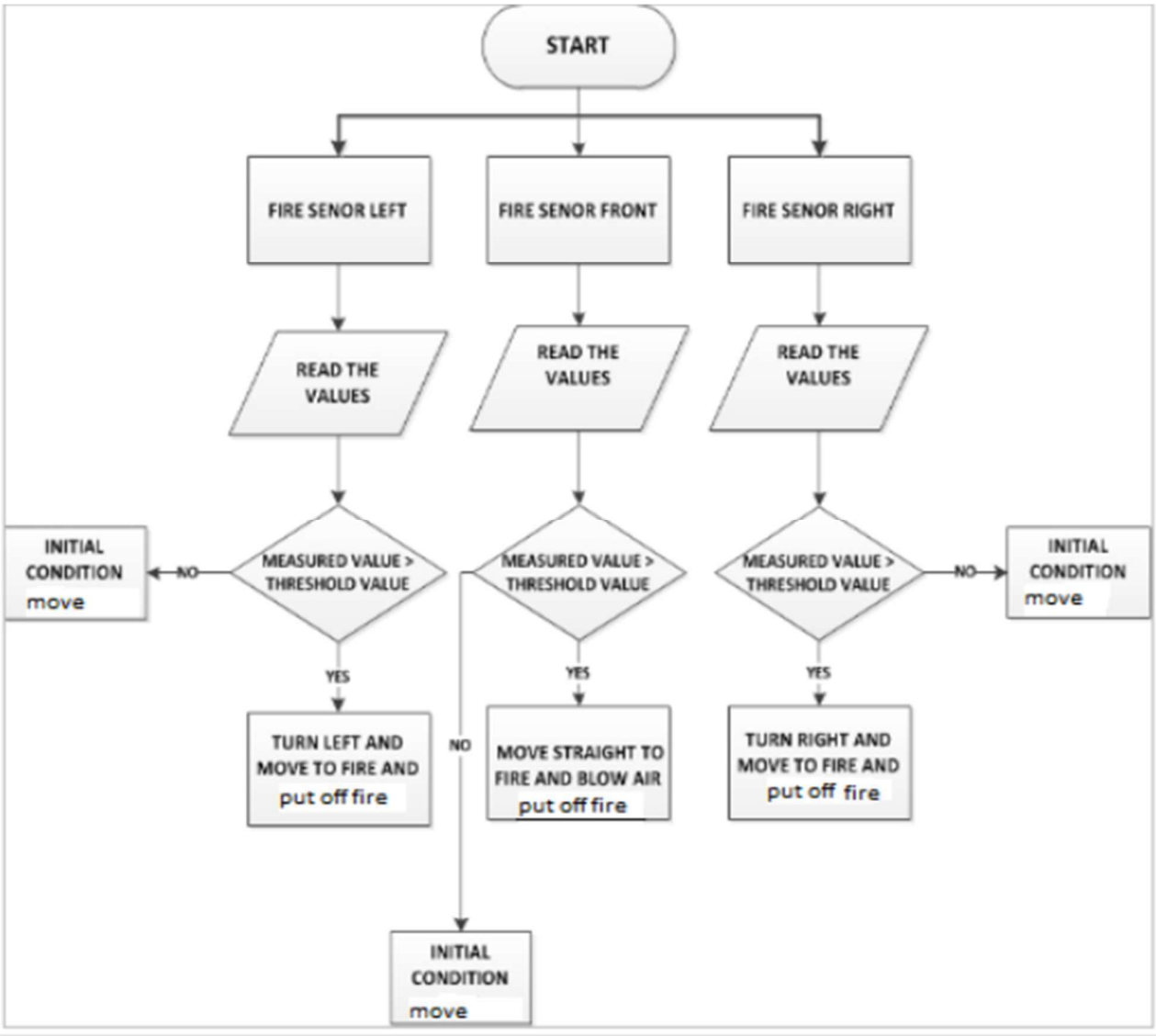


Fig 4.1 : *Flow chart*

The flow starts by initializing the ports of components. First the power supply should be on to the circuit and three sensors are there one on middle and remaining two on right and left side of chassis whenever the fire is occurred the respective value is read by the sensors when fire is occurred the voltage becomes zero and chassis is moved to the respective and put off fire whenever there is no fire then there is no input is occurred occurred voltage is more than 0 volts and the initial condition is move to other direction.

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 MICROCONTROLLER – FLAME SENSOR INTERFACING

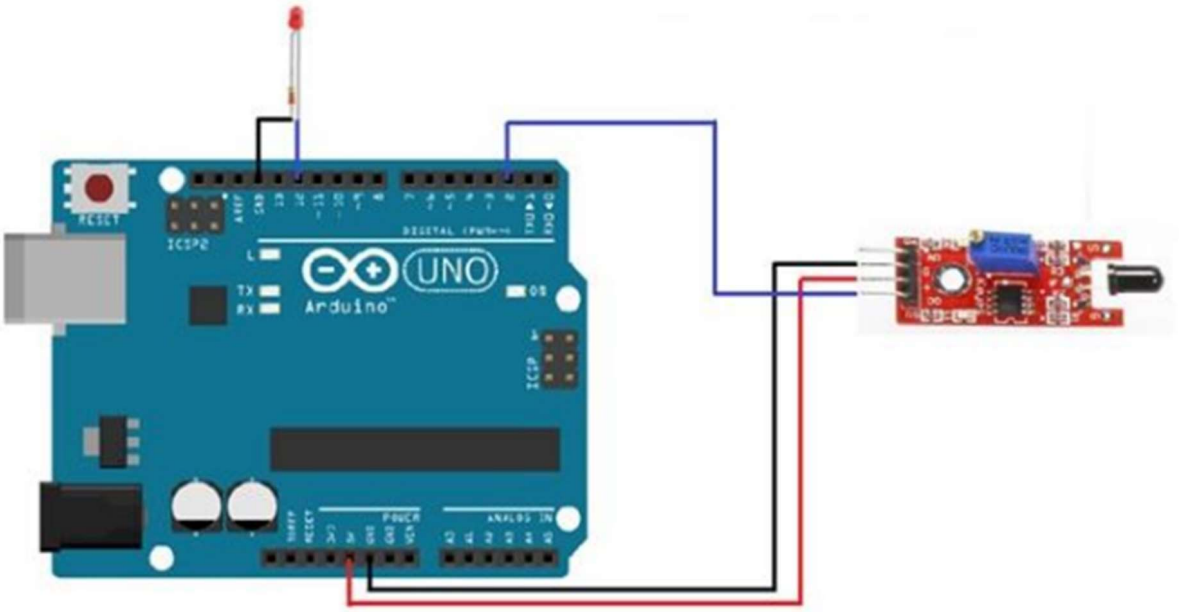


Fig 5.1 : *Arduino -flame sensor interfacing*

Fig 5.1 The flame sensor is used to detect the fire or other light sources which are in the range of wavelength from 760nm to 1100nm. The module consists of an IR sensor, OP-Amp circuitry and a led indicator. When a flame will be detected, the module will turn on its red led. This module is sensitive to flame but it can also detect ordinary light. The sensitivity of this sensor is adjustable and it also has a stable performance. It has both outputs, analog and digital.

The analog output gives us a real time voltage output signal on thermal resistance while the digital output allows us to set a threshold via a potentiometer. In our tutorial we are going to use both of these outputs one by one and see how the sensor works. We can use the flame sensor to make an alarm when detecting the fire, for safety purpose in many projects and in many more ways.

5.2 PROGRAMMING OVERVIEW

About Arduino Uno R3 Programming To programing Arduino Uno, you need the open source Arduino IDE software that the card manufacturer company wrote. The Arduino IDE Program is a software program written in Java language, used to program Arduino cards and to download Arduino cards to Arduino cards. Download the program that I downloaded from the firm's site (https://www.arduino.cc/) with this program. It has an editor that uses the Processing / Wiring language.

The commands that resemble the C language in some places, and the supporting utilities for the projects (Library - library). Along with this, another company's editor (IDE) has been developed since Arduino includes open source software. A bootloader (boot loader) has already been installed on ATmega328 on Arduino Uno. With this bootloader we can develop software without the need for an external programmer to program Arduino. The programming work can easily be performed by making the necessary settings and definitions in the IDE program.

5.3 INITIALIZATION AND WORKING OF MODULE

The initial stage of the project is the part of Finding fire, fire sensor LM393 The fire sensor detects fire at a certain distance. It does not receive data from areas outside of the determined area. It was decided to use two Reducing motors in order to realize the motion system. Both of these Reducing engines can move forward and backward. According to the obstacle state, if the motor is to be turned, one of the motors is given a reverse current by the processor and the axial rotation is provided and the obstacle less driving is provided. Thus, every obstacle was easily overcome in the environment where the system is located.

CHAPTER 6
RESULTS AND CONCLUSION

6. 1 RESULT

After the hardware implementation the system will be like as in following figure

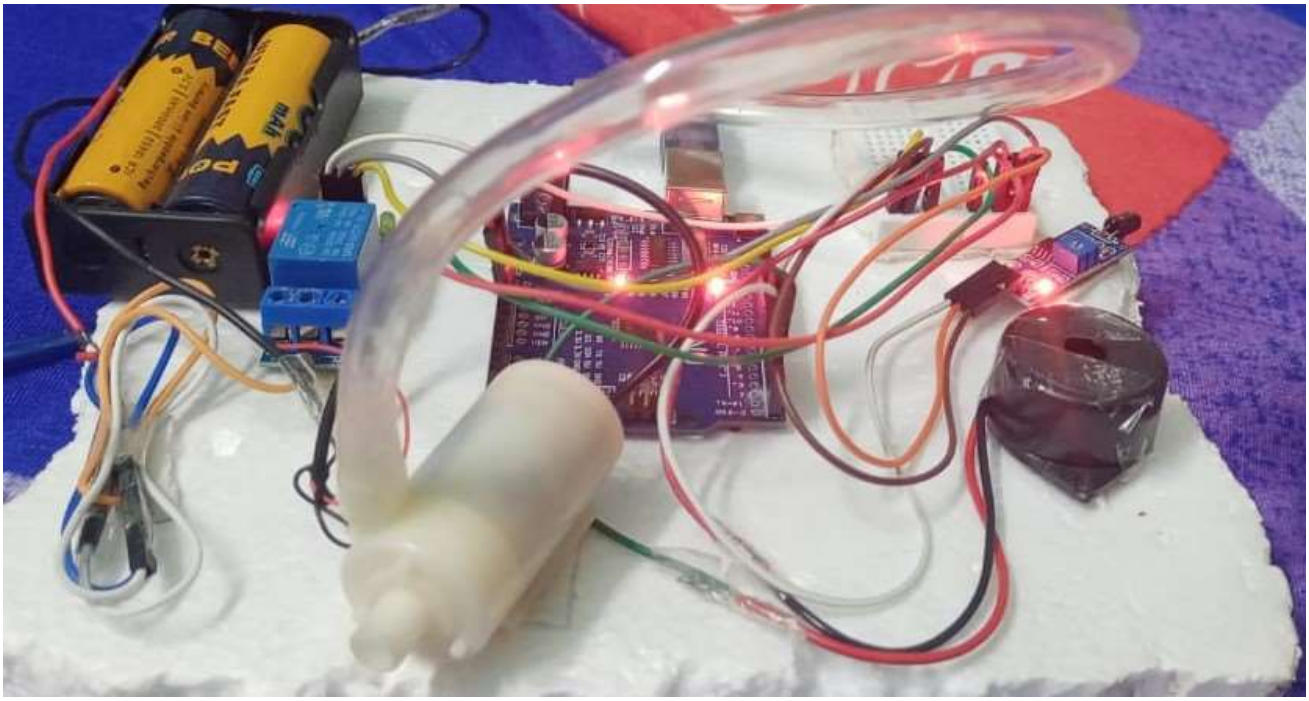


Fig 6.1 : *System Design*

This automated robot doesn't have any wired connection with the external sources .Because it already has DC batteries to run the system . It is free to move within the range of sensors when fire is detected. When we lit the fire the robot system automatically moves towards fire and pours the water. It has 3 flame sensors for three directions they are right ,center and left . if the fire is detected for right flame sensor the wheels of right side will move slowly and turns to right side with the help of left wheels. Similarly it happens to left sensor also , if center flame sensor is detected it goes straight to the flame detected area .

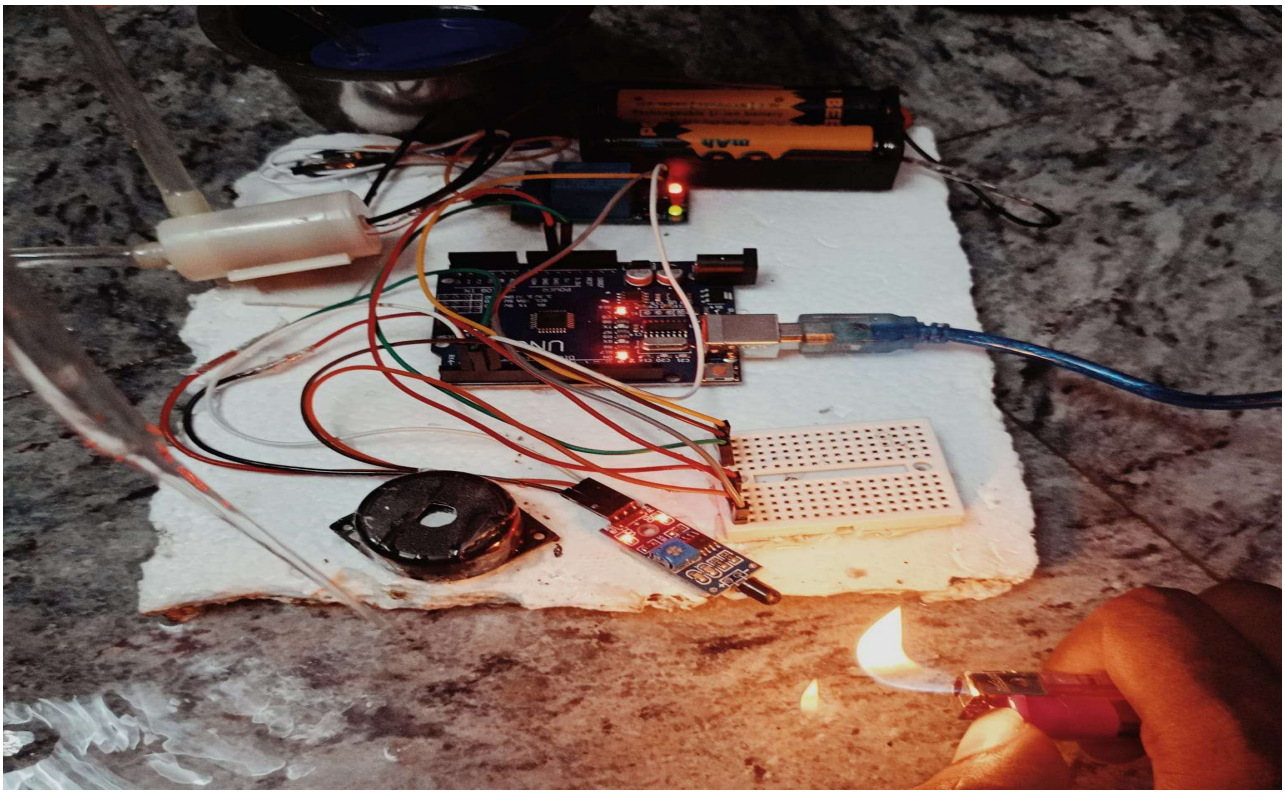


Fig 6.2 : *Final Output*

6.2 CONCLUSION

The prototype of the fire fighter robot was efficiently designed. This prototype has facilities to be integrated with many sensors making it move forward. The toolkit detects the infrared light emitted by the fire with photo diode and sends signal to controller. We intend to extend this work to provide a keypad programmed to allow manipulation of robot to move desired direction with help of motor driver module and extinguish the flames using water tank which is rotated at 180 degrees with help of servo in order for faster result. This future work will also explore to the use of a long distance sensor with suitable hardware to get more better and faster results addition to the characters.

ADVANTAGES AND APPLICATIONS

ADVANTAGES

- Prevention from dangerous incidents.
- Minimization of o ecological consequences
- financial loss can be prevented.
- a threat to a human life can be minimized.
- No supervision is required to control robot

DISADVANTAGES

- It is applicable only for shorter distances
- Doesn’t predict nor interfere with operator’s thoughts.
- Cannot force directly the operator to work

APPLICATIONS

- Can be used in extinguishing fire where probability of explosion is high. For eg. Hotel kitchens, LPG/CNG gas stores, etc.
- Can be used in Server rooms for immediate action in case of fire
- Can be used in extinguishing fire where probability of explosion is high. For eg. Hotel kitchens, LPG/CNG gas stores, etc.
- Every working environment requiring permanent operator's attention, At power plant control rooms
- Can be used in search and rescue operation
- Can Used in domestic cold storage places

FUTURE SCOPE

The project has been motivated by the desire to design a system that can detect fires and take appropriate action, without any human intervention. 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 number of mechanical actions. 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. Our experience suggests that designing a fire-fighting system with sensors and robots is within the reach of the current sensor network and mobile agent technologies. Furthermore, we believe that the techniques developed in this work will carry over to other areas involving sensing and reacting to stimulus, where we desire to replace the human with an automated mobile agent. 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]. P. D. Minns, C Programming For the PC the MAC and the Arduino Microcontroller System. Author House, 2013

[2]. Kirti Kadam, Ayush Bidkar, Vaishnavi Pimpale, Dhanashri Doke and Rachana Patil, "**Fire Fighting Robot**", *International Journal of Engineering and Computer Science*, January 2018.

[3]. Jayanth Suresh, "**Fire Fighting Robot**", *IEEE 2017 International Conference on Computational Intelligence in Data Science(ICCIDS)*, 2-3 June 2017.

[4]. Vynzarali Amenodin, Margie Rose, Hussein P Lawanza, Mojahid Maradia, D Apig, Dianne B. Falceso, et al., "**F2R:An Arduino Based firefighting robot**", *Researchgate*.

[5]. Circuit Digest : DIY Arduino Based Fire Fighting Robot Project with Code and Circuit Diagram (circuitdigest.com)-

[6]. Instructables : Fire Fighting Robot Using Arduino : 4 Steps - Instructables

[7]. Learn Robotics : Make a Fire Extinguishing Robot using Arduino - Learn Robotics

APPENDIX

```
#define SENSOR_PIN 2
#define BUZZER_PIN 3
#define RELAY_PIN 4
#define SPRINKLER_START_DELAY 5000  //5 seconds
#define SPRINKLER_ON_TIME 3000      //3 seconds Sprinkler on time

unsigned long previousTime = millis();

void setup()
{
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(SENSOR_PIN, INPUT);
}

void loop()
{
  //If there is fire then the sensor value will be LOW else the value will be HIGH
  int sensorValue = digitalRead(SENSOR_PIN);

  //There is fire
  if (sensorValue == LOW)
  {
    analogWrite(BUZZER_PIN, 50);                    //Turn on buzzer

    if (millis() - previousTime > SPRINKLER_START_DELAY)  //We will wait for few seconds
before sprinkler can be started once fire is detected.
    {
      digitalWrite(RELAY_PIN, LOW);                 //Relay is low level triggered relay so we
need to write LOW to switch on the light
      delay(SPRINKLER_ON_TIME);                     //Keep sprinkler on for sometime.
    }
  }
  else
  {
    analogWrite(BUZZER_PIN, 0);
    digitalWrite(RELAY_PIN, HIGH);
    previousTime = millis();
  }
}
```