Visvesvaraya Technological University

Belagavi

A Mini Project Report

on

FIRE FIGHTING ROBOT USING ARDUINO UNO

Submitted by

Lokesh Y 1NH18EC061

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS & COMMUNICATION ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING
New Horizon Knowledge Park, Ring Road, Marathalli
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

NEW HORIZON COLLEGE OF ENGINEERING
New Horizon Knowledge Park, Ring Road, Marathalli
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

BENGALURU-560103

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Mini project entitled **"Fire Fighting Robot Using Arduino Uno"** is carried out by **Lokesh Y(1NH18EC061)**, bonafide student of NHCE, Bengaluru in partial fulfilment for the award of Bachelor of Engineering in Electronics and Communication of the Visvesvaraya Technological University, Belagavi during the year 2020-21. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini project report has been approved as it satisfies the academic requirements in respect of the mini project work prescribed for the said degree.

Signature of the Guide	Signature of the HOD
Dr. Gurulakshmi A B	Dr. Sanjeev Sharma
Associate Professor	Professor & HOD
Department of ECE	Department of ECE
NHCE, Bengaluru	NHCE, Bengaluru

External Viva

Name of the Examiners	Signature with Date
1. ____________________	1. ____________________
2. ____________________	2. ____________________

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr.Manjunatha**, Principal, NHCE and **Dr. Sanjeev Sharma**, head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **Dr.Gurulakshmi A B** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of mini project.

Lokesh Y (1NH18EC061)

TABLE OF CONTENTS

ABSTRACT

CHAPTER 1

INTRODUCTION………………………………………………………………………… 5

CHAPTER 2

LITERATURE SURVEY………………………………………………………………. 6

CHAPTER 3

PROPOSED METHODOLOGY……………………………………………………….. 8

WORKING………………………………………………………………….............. 8

CHAPTER 4

PROJECT DESCRIPTION……………………………………………………………. 10

COMPONENTS REQUIRED…………………………………………………………. 10

CHAPTER 5

RESULTS AND DISCUSSION………………………………………………………. 33

CHAPTER 6

CONCLUSION AND FUTURE SCOPE……………………………………………... 35

REFERENCES…………………………………………………………………………36

APPENDIX..37

LIST OF FIGURES

SL No	FIGURE	FIGURE DESCRIPTION	Page No
1	Fig: 2.1	FIRE FIGHTING ROBOT OF LITERATURE SURVEY I	6
2	Fig: 2.2	FIRE FIGHTING ROBOT OF LITERATURE SURVEY II	7
3	Fig: 3.1	FIRE FIGHTING ROBOT CIRCUIT	9
4	Fig: 4.1	ARDUINO UNO	11
5	Fig: 4.2	ATMEGA328 MICROCONTROLLER	12
6	Fig: 4.3	ARDUINO UNO PINOUT	13
7	Fig: 4.4	L293D MOTOR DRIVER MODULE	15
8	Fig: 4.5	L293D MOTOR DRIVER PIN DIAGRAM	15
9	Fig: 4.6	CONNECTION OF MOTOR DRIVER MODULE WITH ARDUINO UNO	17
10	Fig: 4.7	H-BRIDGE CIRCUIT WHEN S1 AND S4 CLOSED	18
11	Fig: 4.8	H-BRIDGE CIRCUIT WHEN S2 AND S3 CLOSED	18
12	Fig: 4.9	TRANSISTOR DIAGRAM	19
13	Fig: 4.10	PULSE WIDTH MODULATION	20
14	Fig: 4.11	DC MOTOR	21
15	Fig: 4.12	DC MOTOR WORKING	21
16	Fig: 4.13	FLEMING’S LEFT HAND RULE	22
17	Fig: 4.14	FLAME SENSOR	22
18	Fig: 4.15	CONNECTION OF FLAME SENSOR WITH ARDUINO UNO	24
19	Fig: 4.16	POTENTIOMETER ANGULAR POSITION	25
20	Fig: 4.17	CONNECTION OF SERVO MOTORWITH WITH ARDUINO UNO	26
21	Fig: 4.18	CONNECTION OF WATER PUMP WITH ARDUINO UNO	27
22	Fig: 4.19	BLOCK DIAGRAM	28
23	Fig: 4.20	ARDUINO INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)	29
24	Fig: 4.21	ARDUINO INTEGRATED DEVELOPMENT ENVIRONMENT TOOLS	30
25	Fig: 4.22	ARDUINO INTEGRATED DEVELOPMENT ENVIRONMENT BUTTONS	30
26	Fig: 4.23	ARDUINO INTEGRATED DEVELOPMENT ENVIRONMENT ERROR WINDOW	31
27	Fig: 4.24	ARDUINO INTEGRATED DEVELOPMENT ENVIRONMENT OUTPUT WINDOW	31
28	Fig: 5.1	MODEL OF THE PROJECT 1	33
29	Fig: 5.2	MODEL OF THE PROJECT 2	34
30	Fig: 5.3	MODEL OF THE PROJECT 3	35

ABSTRACT

- Increasing population and technological growth and development all across the world has led to various problems such as accidents. In which fire accidents hold the major part.
- Till date fire accidents is a hazardous act that led to numerous major consequences. We rely on human resources till now. They are fire fighters. They risk their lives to extinguish it. Detecting fire and extinguishing it is a dangerous job that puts the life of a fire fighter at risk. There are many fire accidents which fire fighter had to lose their lives
- Fire accidents are one such major problems. To deal with this robotic industry has emerged with the project that can replace human resources with the machine called robot. Firefighting robots can be used.
- Detecting fire and extinguishing it has been a difficult task. In our project a fire extinguishing robot has been proposed which detects the fire location, navigates through it and extinguishes the fire by pumping water.
- In this project we are implementing a robot that detects the presence of fire in 3 directions i.e., straight, left and right. The robot is controlled automatically without using any remote controller for its control.

.

Keywords: Arduino UNO, Flame sensor, Motor driver, Water pump

CHAPTER 01

INTRODUCTION

Modern industrialization and economic growth have resulted in factories, complex buildings and dense apartments in metropolitan and cosmopolitan areas. With the ever-increasing technology, the developments are increasing in the face of the situations that cost human life. The growing population is bringing several problems together.

Every day, the robotic industry emerges as a model that is produced as an alternative to the human element in a new branch. Robots are the replacement for human resources. Fire accidents are one such major problem. To deal with these fire fighting robots can be used. These are developed in such a way that can automatically sense and detect fire n, then extinguish with water pumping mechanisms without human intervention. Flying robots, wheeled robots, underwater robots are some of the other robots that are developed. The industry has a lot of work in this area. The robots are operated using artificial intelligence which can detect the fire and act accordingly.

In this project we are implementing a robot that can not only detect the fire but can also detect the barriers that come in its way. There's no need for a person to monitor , the robot is controlled automatically without using any remote controller for its control. There are 3 flame sensors used to detect the presence of fire in 3 directions i.e. straight, left and right. This robot completely works on these sensors and Arduino Uno which controls it’s working.

CHAPTER 02

LITERATURE SURVEY

AUTONOMOUS FIRE DETECTING AND EXTINGUISHING ROBOT

By,

Mukul Diwanji,
Saurabh Hisvankar,
Chhaya Khandelwal,
MGM’s Jawaharlal Nehru Engineering College,
Aurangabad, Maharashtra
2020

RESULT AND CONCLUSION

- In this paper, represents the Fire Fighting Robot.
- Developed and divided into three element i.e hardware, electronic, and programming. It has DC motors for driving system and castor wheel for giving direction.
- For suction and spraying of water a 12 Volt DC pump is used. Interfacing of Servo Motor and various sensors with Arduino Uno Board is done .
- Arduino IDE language to determine the robot movement from the input is used for programming.

Figure 2.1: Fire Fighting Robot of Literature survey I

FIRE FIGHTING ROBOT

By,
Dr. Niranjan Bhattacharyya,
Akshat Sharma
Department of Electronics and Communication Engineering,
Bhagwan Parshuram Institute of Technology,
New Delhi
2018

RESULT AND CONCLUSION

- A system capable of navigating towards fire and then extinguishing it by pumping water was made.
- Potentially been useful in accompanying fire fighters and outbreak prevention is done.
- Fire fighting products used in various sectors such as defence, medicine,fire department etc

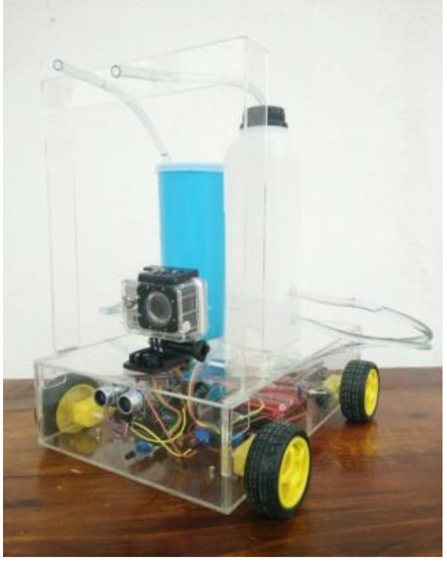


Figure 2.2: Firefighting robot of Literature survey II

CHAPTER 03

PROPOSED METHODOLOGY

Working

The working of the project is mainly based on the working of Arduino and fire sensors which have an IR receiver (photodiode) used for fire detection.

- In this project we use three flame sensors to detect the fire across 180 degrees.
- When fire is detected the output pin(D0) of the corresponding sensor gives a low voltage i.e. 0V and if there is no fire detected the D0 pin gives a high voltage i.e. 5V.
- And by means of these outputs of corresponding sensors, a motor driver module is controlled which is used to indicate the direction of fire and guides the movement of the robot.
- When the fire is detected by any of the sensors then LM1, LM2, RM1, RM2 values are changed accordingly to facilitate the movement of the robot.
- When NO fire is detected or to stop the robot,

  LM1 = HIGH RM1 = HIGH
  LM2 = HIGH RM2 = HIGH
- When fire is detected by the LEFT sensor then the robot is moved **left.** For that,

  LM1 = HIGH RM1 = HIGH
  LM2 = HIGH RM2 = LOW
- When fire is detected by the RIGHT sensor then the robot is moved **right.** For that,

  LM1 = HIGH RM1 = HIGH
  LM2 = LOW RM2 = HIGH
- When fire is detected by the STRAIGHT sensor then the robot is moved **forward.** For that,

  LM1 = HIGH RM1 = HIGH
  LM2 = LOW RM2 = LOW

- These conditions allow the robot to move towards the fire and the robot stops after reaching the fire.

Once it reaches the fire the extinguishing action takes place i.e. the motor pump turns ON and the water pumps through the pipe to extinguish the fire. This direction of water pumping is controlled by the servo motor.

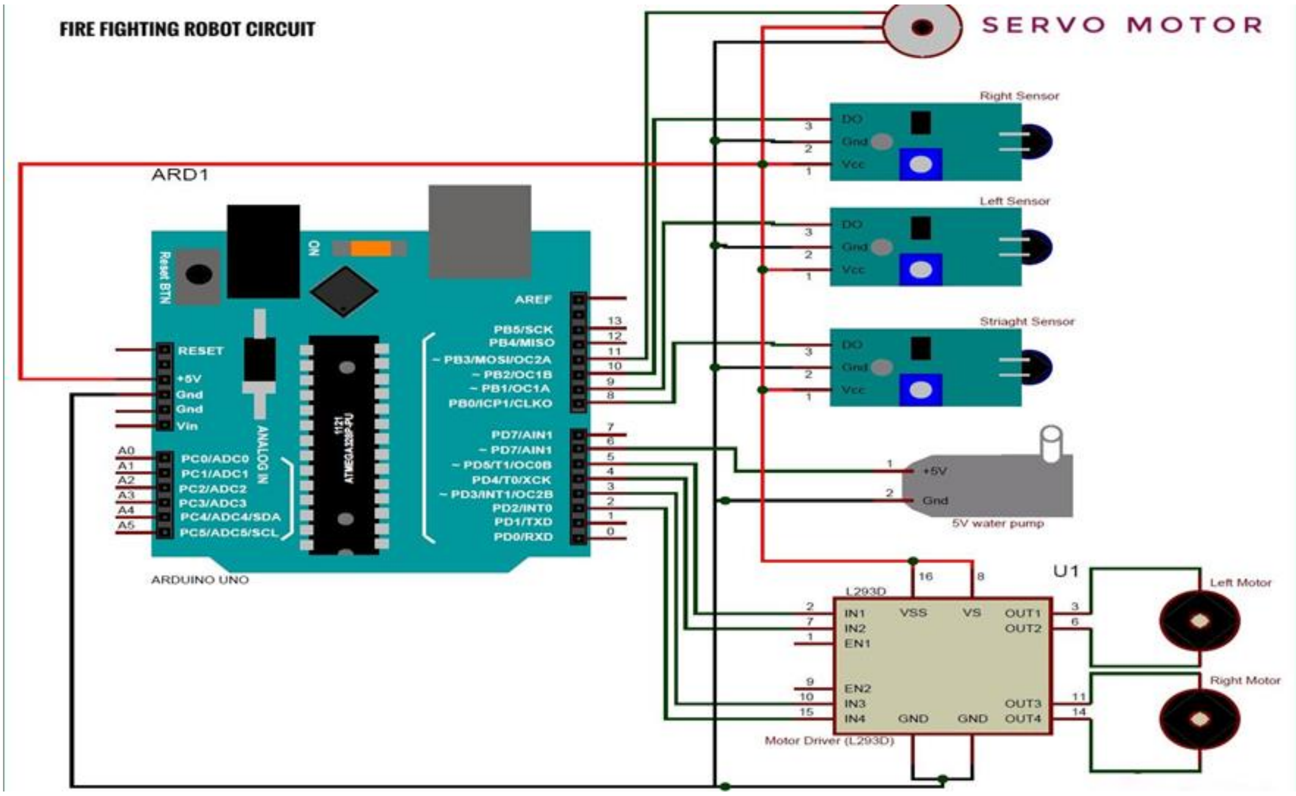


Figure 3.1: Fire Fighting Robot Circuit

CHAPTER 04

PROJECT DESCRIPTION

The circuitry of the robot is built on Arduino Uno which controls all the actions of the robot. There are 3 sensors used to detect the presence of the fire. A servo motor to indicate the direction of the fire and a motor driving module L293D to control the movement of the robot. This robot is completely automated using Arduino Uno and does require any relay to control its actions.

Components Required:

1. Arduino Uno
2. Flame sensors
3. Servo motor (SG90)
4. (L293D) Motor driver module
5. DC Motor
6. Mini breadboard
7. Water pump

Arduino Uno:

It is one of the most popular prototyping boards. It is widely used in robotic applications as it is small and packed with rich features. The board comes with a built-in boot ladder, has 14 digital pins out of which 6 pins can be used for PWM output, 6 analog pins and a microcontroller Atmega 320. This microcontroller has several features such as timers, counters, interrupts, PWM, CPU. It allows the designer to sense the external electronic devices in the real world.

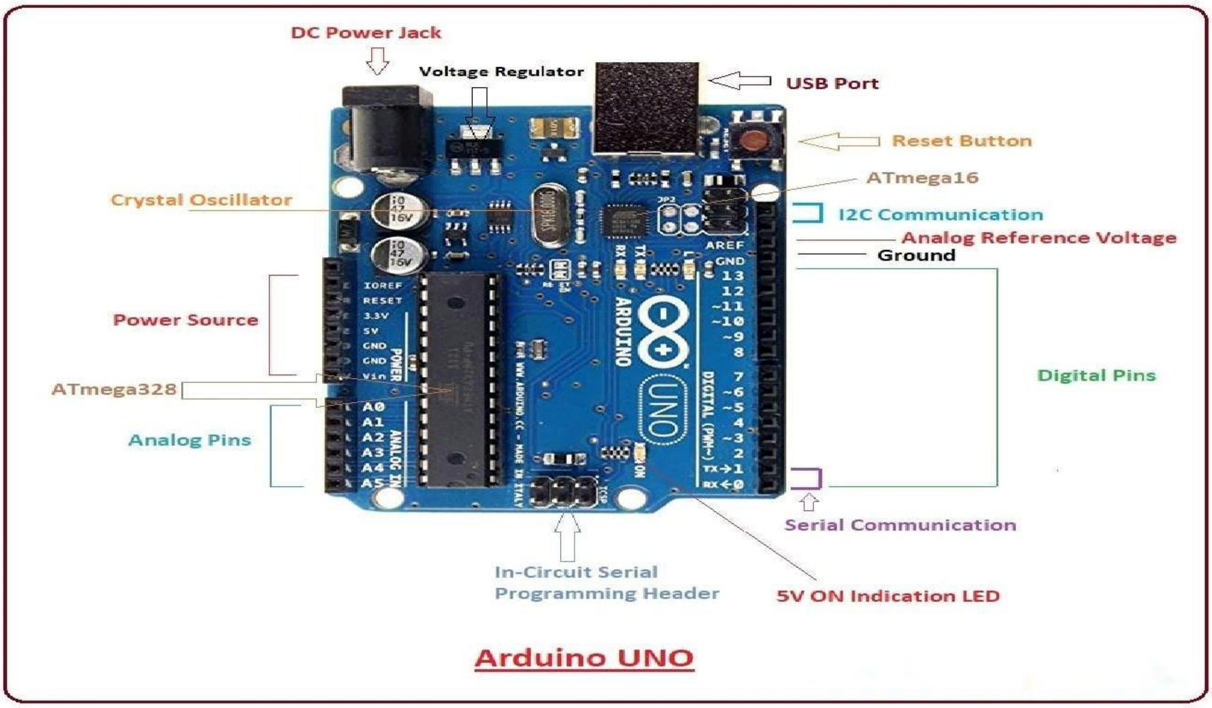


Figure 4.1: Arduino Uno

Features of Arduino Uno:

- It comes with USB interface i.e. A USB port is added on the board to develop serial communication with the computer. Atmega328 microcontroller is placed on the board that comes with several features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.
- It has an open source platform where anyone can modify and optimize the board based on the number of instructions and tasks that they want to achieve.
- This board comes with a built-in regulator which keeps the voltage under control when the device is connected to the external device.
- 13Kb of flash memory is used to store the number of instructions in the form of code.
- Only 5V is required to turn the board ON, which can be achieved directly using a USB port or external adapter.

Atmega 328 microcontroller:

It has 1Kb Electrically Erasable Programmable Read Only Memory (EEPROM). This property shows if the electric supply supplied to the microcontroller is removed, even then it can store the data and can provide results after providing it with the electric supply. Moreover, it has a 2Kb Static Random-Access Memory (SRAM). Atmega 328 has several features which makes 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, programmable lock for software security, through output up to 20MIPS etc.

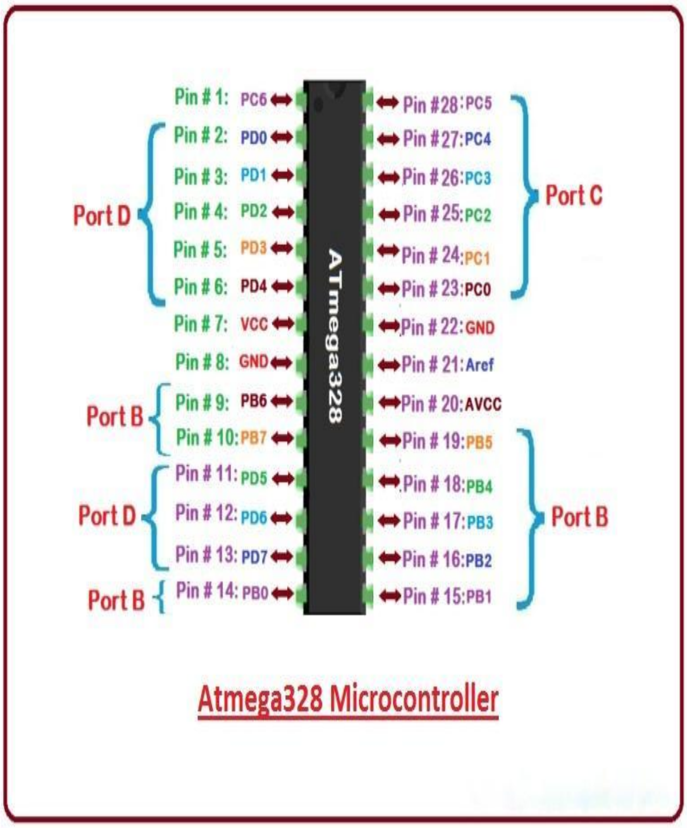


Figure 4.2: Atmega328 Microcontroller

Pin descriptions:

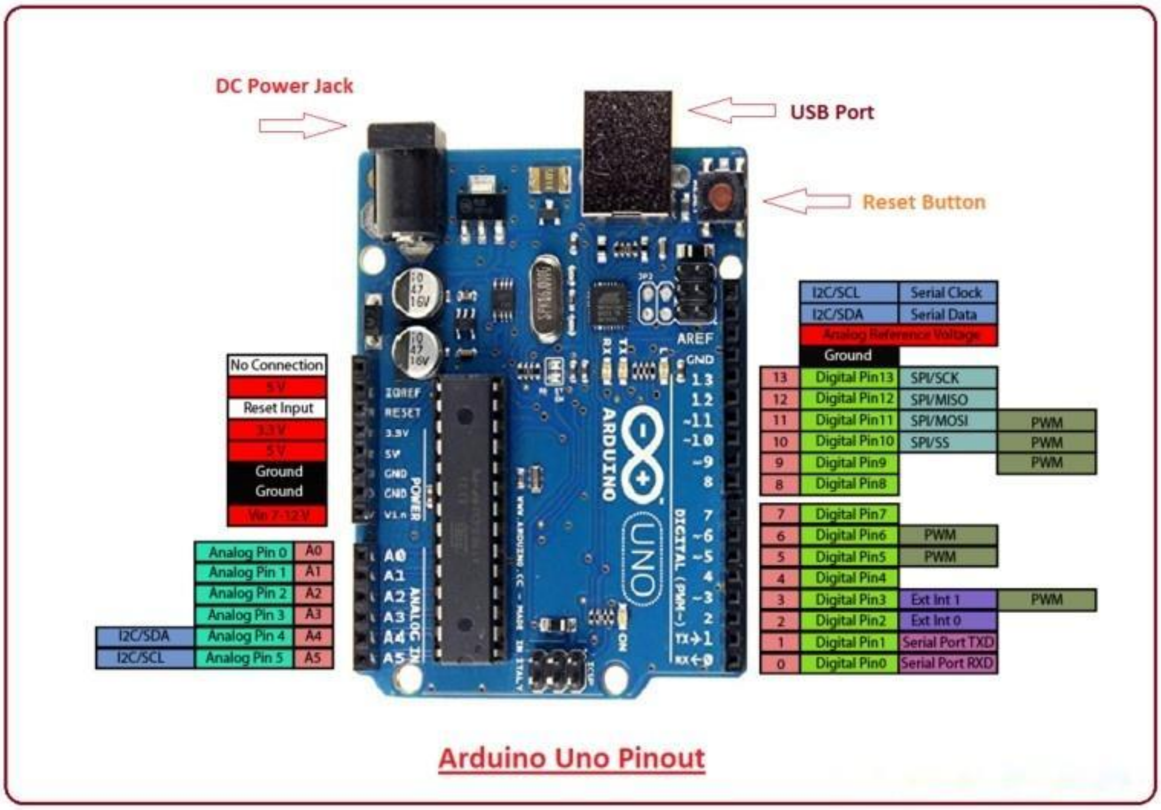


Figure 4.3: Arduino Uno Pinout

There are several I/O digital and analog pins placed on the board which operate at 5v. These pins come with a standard operating rating ranging between 20mA to 40mA. Internal pull-up resistors are used in the board that limits the current exceeding from the given operating conditions. However, too much increase in current makes these resistors useless and damages the device.

LED: Arduino UNO comes with built-in LED which is connected through pin 13. Providing high value to this pin will turn it ON and a low value will turn it OFF.

Vin: It is the input voltage provided to the Arduino Board. It is different from 5v supplied through a USB port. This pin is used to supply voltage. If a voltage is provided through the power jack, it can be accessed through this pin.

5v: this board comes with the ability to provide voltage regulation. A 5v pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack. USB supports voltage around 5v while Vin and power jack supports a voltage range between 7v to 12v. It is recommended to operate the board on 5v. It is important to note that, if the voltage is supplied through 5v or 3.3v pins, they result in bypassing the voltage regulation that can damage the board if voltage surpasses from its limit.

GND: These are ground pins. More than one ground pin is provided on the board which can be used as per requirements.

Reset: This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.

IOREF: This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then selects the proper power source.

PWM: PWM is provided by 3,5,6,9,10,11 pins. These pins are configured to provide 8-bit output PWM.

SPI: It is known as Serial Peripheral Interface. Four pins 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) provide SPI library.

AREF: It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

TWI: It is called Two-Wire Interface. TWI communication is accessed through Wire library A4 and A5 pins are used for this purpose.

Serial Communication: It is accessed through two pins called pin 0 (Rx) and pin 1 (Tx). Rx pin is used to receive data while Tx pin is used to transmit data.

External Interrupts: Pin and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

L293D Motor Driver Module:

This motor driver module allows us to control the speed and direction of two motors simultaneously. This is being designed and developed on the IC L293D.

L293D is a 16 Pin Motor Driver IC. This is designed to provide bidirectional drive currents at voltages from 5v to 36v.

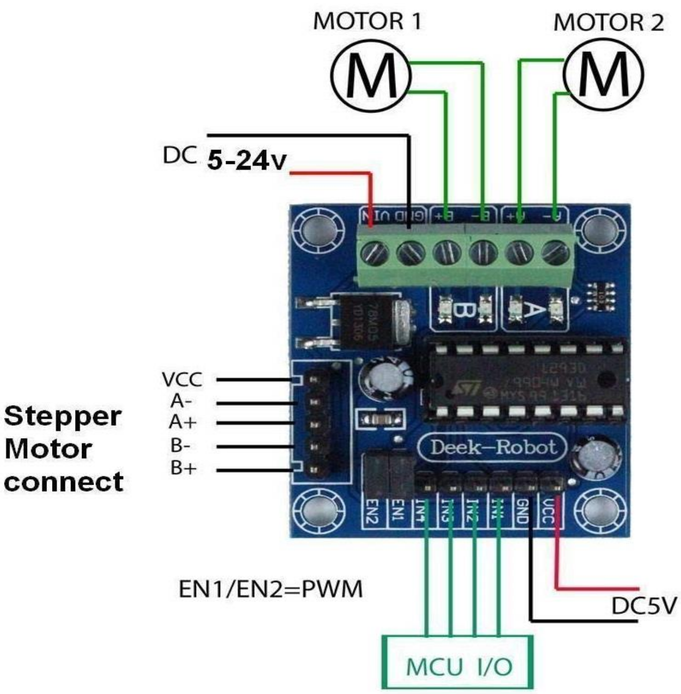


Figure 4.4: L293D motor Driver Module

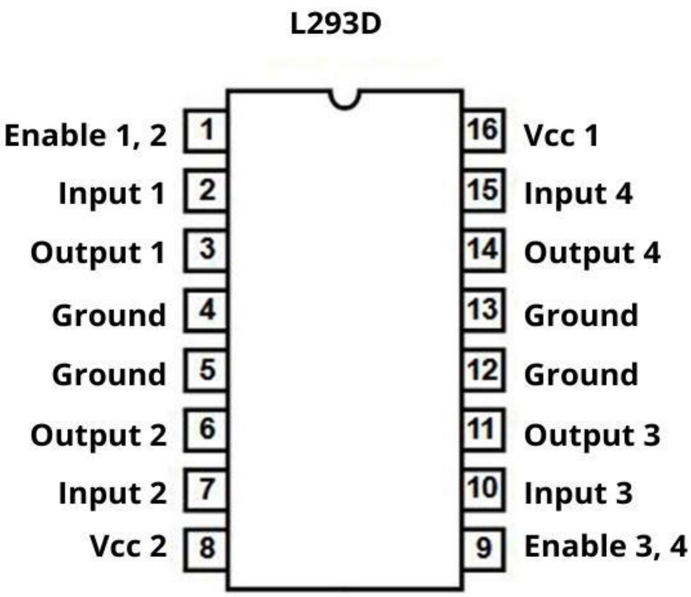


Figure 4.5: L293D motor Driver Pin Diagram

Pin 1: When Enable 1 or 2 is HIGH, the left part of the IC will work, i.e. the motor connected with pin 3 and 6 will rotate.

Pin 2: Input 1, when this pin is HIGH the current will flow through output 1.

Pin 3: Output 1, this pin relates to one terminal of the motor.

Pin 4 and 5: Ground pins.

Pin 6: Output 2, this pin relates to one terminal of the motor.

Pin 7: Input 2, when this pin is HIGH the current will flow through output 2.

Pin 8: VSS, this pin is used to give power supply to connected motors from 5v to 36v which depends on the motors connected.

Pin 9: When Enable 3 or 4 is HIGH, the right part of the IC will work, i.e. the motor connected with pin 11 and 14 will rotate.

Pin 10: Input 4, this pin relates to one terminal of the motor.

Pin 12 and 13: GND pins.

Pin 14: Output 3, this pin relates to one terminal of the motor.

Pin 15: Input 3, when this pin is HIGH the current will flow through output 3.

Pin 16: VCC, to supply power to IC i.e. 5v.

Working mechanism: Rotation of motor depends on enable pins and inputs given.

INPUT 1	INPUT 2	RESULT
0	0	Stop
0	1	Anti-Clockwise
1	0	Clockwise
1	1	Stop

Table no: 4.1

This IC is used to rotate the dc motors of the robot chassis accordingly to move the robot in the required direction. This module uses the concept of H-bridge. This is the main principle behind the movement of the robot. It consists of two h-bridge circuits.

Connection with Arduino Uno:

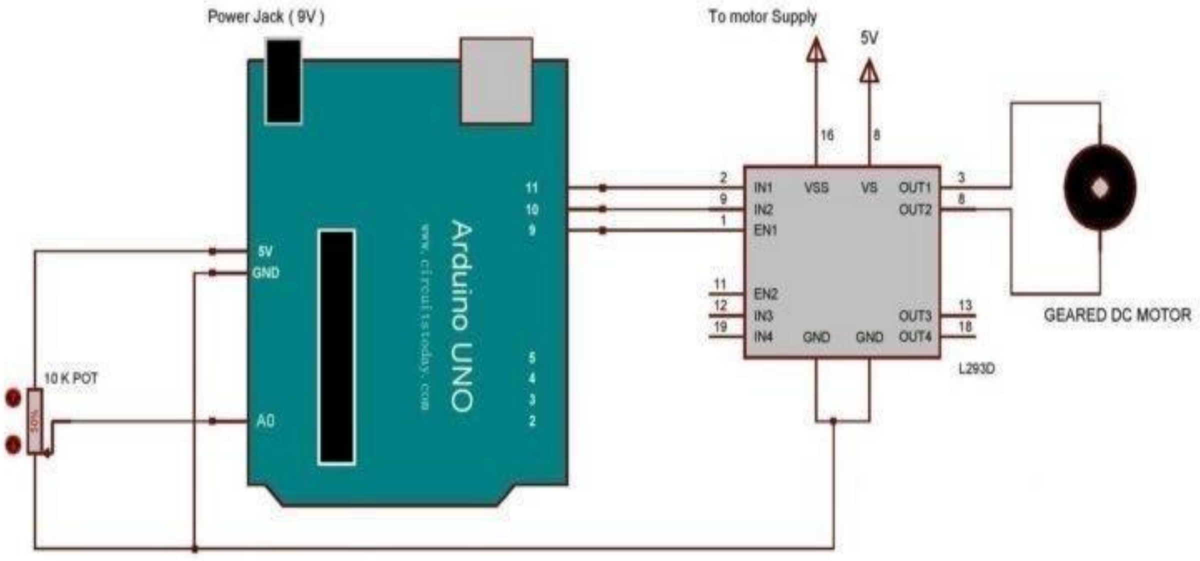


Figure 4.6: Connection of motor driver module with Arduino Uno

<u>H-Bridge circuit</u>:

It is an electronic circuit that allows the bidirectional supply of voltage to the load (dc motors). These are the circuits used in robotics to control the movement of dc motors i.e. to run backward and forward. These circuits are used in different converters like DC-AC, DC-DC, AC-AC and other power electronic converters. The basic principle of h-bridge is depended on the 4 switches of a H-bridge circuit.

The 4 switches controls the dc motor.

1. When switches S1 and S4 are closed, the motor runs in clockwise direction to provide the forward movement.

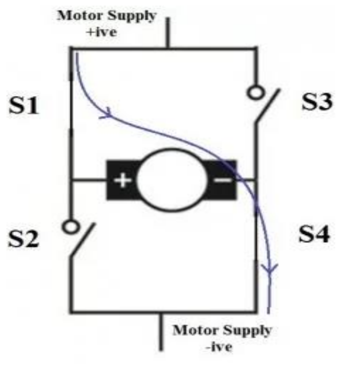


Figure 4.7: H-Bridge circuit when S1 and S4 closed

2. When switches S3 and S2 are closed, the motor runs in anti-clockwise direction.

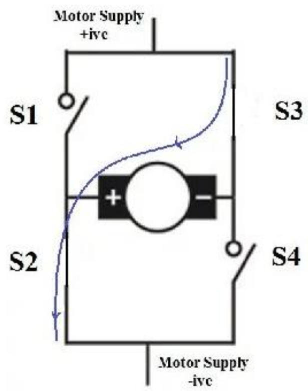


Figure 4.8: H-Bridge circuit when S2 and S3 closed

In actual practice these switches are replaced by transistors i.e. both npn and pnp transistors are used. The transistor diagram is given below:

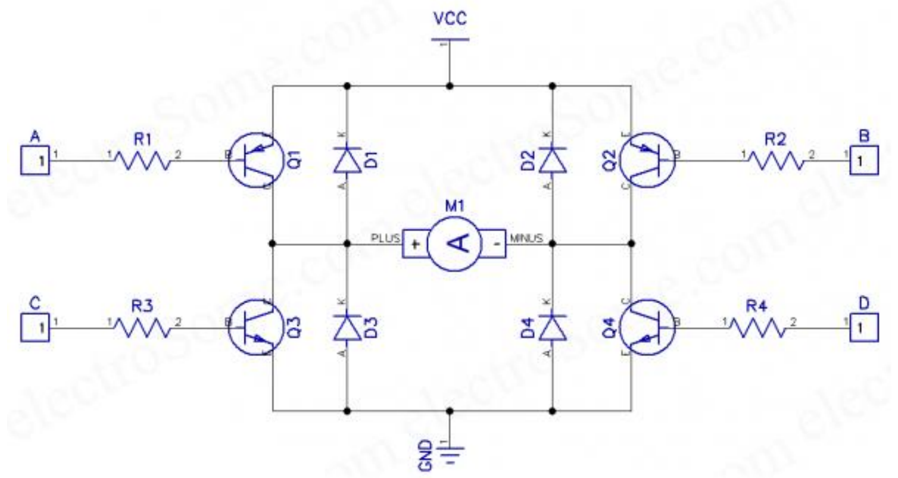


Figure 4.9: Transistor diagram

A npn transistor is turned ON when a high voltage is given to it and a pnp transistor is turned ON when a low voltage is given to it. When Q1 and Q4 are ON (A-low, B-high, C-low, D-high) the motor runs in a clockwise direction. When Q2 and Q3 are ON (A-high, B-low, C-high, D-low) the motor runs in an anti-clockwise direction. Diodes are used for fast switching between the transistors and reduces the problems caused due to negative voltage (back emf) by dc motors.

Pulse Width Modulation:

Pulse width modulation (PWM) or duty cycle variations are commonly used to control the speed of dc motors. Duty cycle is defined as the percentage of digital high to digital low plus digital high pulse width during 1 PWM period. The PWM waveform is shown below in fig.1.5. By changing the width of these pulses, the speed of the dc motor can be varied accordingly. If the pulse is 'ON' for a longer period, the motor rotates faster. Similarly, if the pulse is 'ON' only for a shorter period the motor rotates slowly. In other words, the more average voltage applied to the motor terminal, the stronger the magnetic flux produced inside the armature windings and the faster the motor will rotate. The advantage of using this technique is that the power loss in the switching transistor is very small because the transistor is either fully 'ON' or 'OFF' unlike analog signals. As a result, the transistor provides a linear type of control which results in better speed stability.

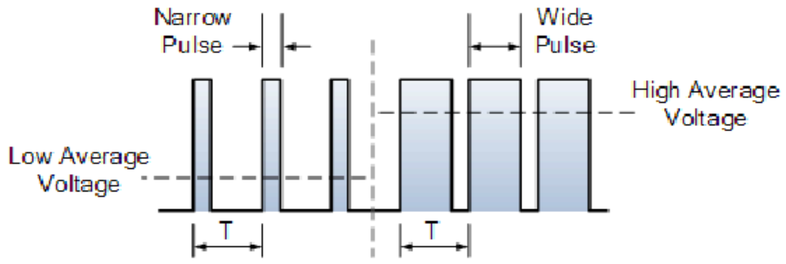


Figure 4.10: Pulse width modulation

Working of dc motors:

A dc motor is a rotary electrical machine that converts direct current electric energy into mechanical energy. The dc motor used to rely on the forces produced by the magnetic field. Nearly all types of dc motors have some internal mechanism, either electromechanical or electronic to change the direction of flow of current in the motor periodically.

Dc motors were the first form of widely used motors as they could be powered from existing direct-current lighting power distribution systems. A dc motor's speed can be controlled over a wide range by using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys and appliances.

The permanent magnet DC motor converts electrical energy into mechanical energy through the interaction of two magnetic fields. One field is produced by a permanent magnet assembly, the other field is produced by an electric current flowing in the motor windings. These two fields result in a torque which tends to rotate the rotor. As the rotor turns, the current in the windings is commutated to produce a continuous torque output. The stationary electromagnetic field of the motor can also be wire-wound like the armature (called a wound-field motor) or can be made up of permanent magnets.

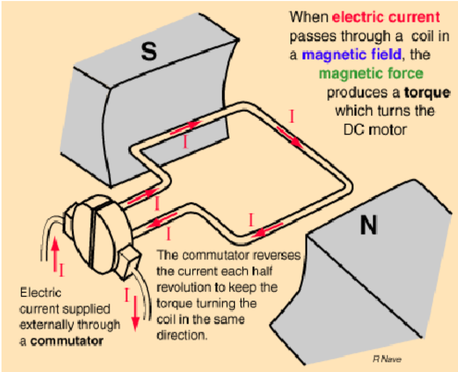


Figure 4.11: DC motor

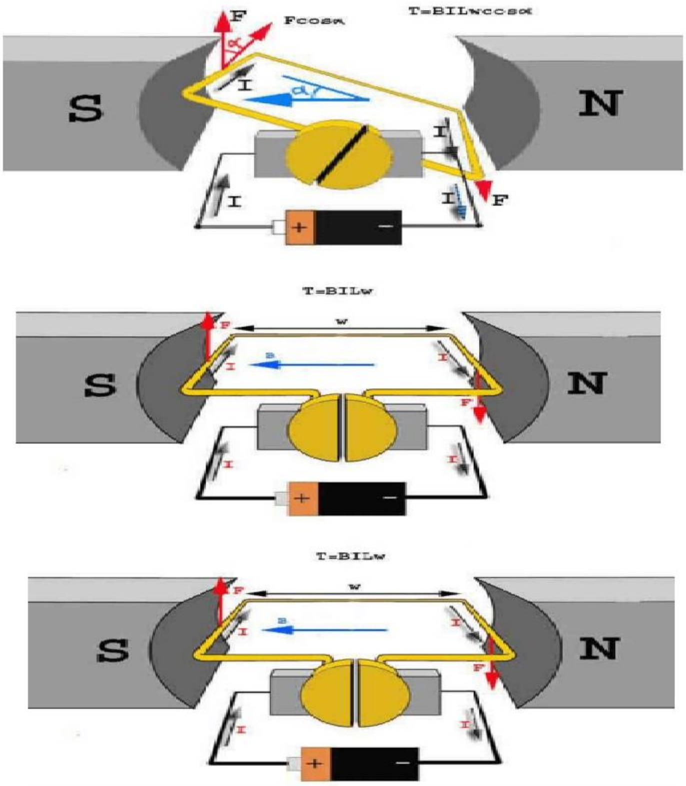


Figure 4.12: DC motor working

In either style (wound-field or permanent magnet) the commutator acts as half of a mechanical switch and rotates with the armature as it turns. The commutator is composed of conductive segments (called bars), usually made up of copper, which represent the termination of individual coils of wire distributed around the armature. The second half of the mechanical switch is completed by the brushes. These brushes typically remain stationary with the motor's housing but ride (or brush) on the rotating commutator. As electrical energy is passed through the brushes and consequently through the armature a torsional force is generated as a reaction between the motor's field and the armature causing the motor's armature to turn. As the armature turns, the brushes switch to adjacent bars on the commutator. This switching action transfers the electrical energy to an adjacent winding on the armature which in turn perpetuates the torsional motion of the armature.

The permanent magnet motors are probably the most commonly used DC motors, but there are also some other types of DC motors (types which use coils to make the permanent magnetic fields). DC motors operate from a direct current source. Movement of the magnetic field is achieved by switching current between the coils within the motor. This action is called "commutation". Many DC motors (brush-type) have built-in commutation, meaning that as the motor rotates, mechanical brushes automatically commutate coils on the rotor. A simple permanent dc motor is used in a variety of products such as toys, servo mechanisms, valve actuators, robots and automotive electronics.

The working and flow of current and magnetic field is based on the Fleming's left hand rule:

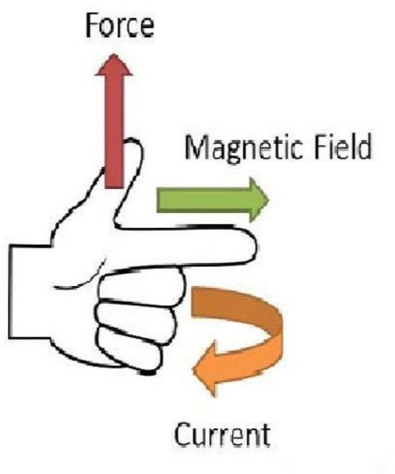


Figure 4.13: Fleming's left hand rule

Flame Sensor:

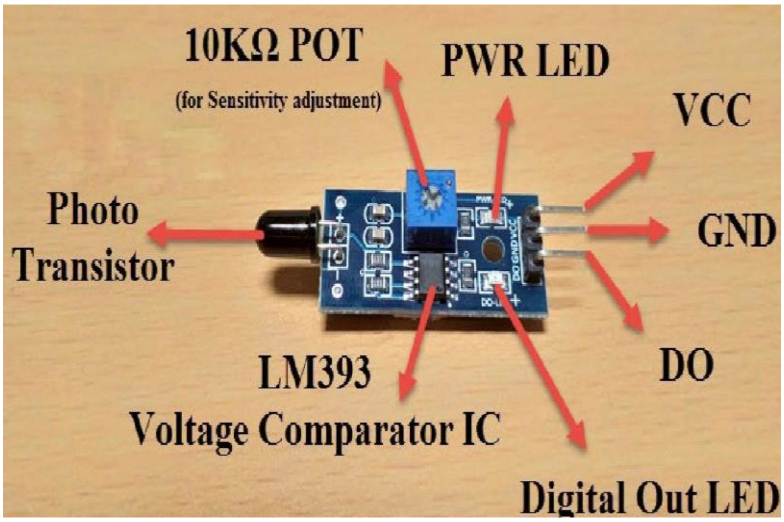


Figure 4.14: Flame sensor

These sensors operate by using an infrared wave band. These sensors can easily detect even small intensities of IR waves in the fire using photodiodes. It is based on the YG1006 sensor which is a high speed and highly sensitive NPN silicon photoresistor. Due to its black epoxy, the sensor is sensitive to infrared radiation. It can detect IR light with a wavelength ranging from 700 nm to 1000nm and its detection angle is about 60 degrees. When the sensor detects flame the signal LED will light up and the D0 pin goes LOW.

Features:

- High Photosensitivity
- Fast response time
- Sensitivity adjustable

Specifications:

- Working voltage is 3.3v-5v
- Detect range is 60 degree
- Digital/analog output
- On board LM303 chip

Connection with Arduino Uno:

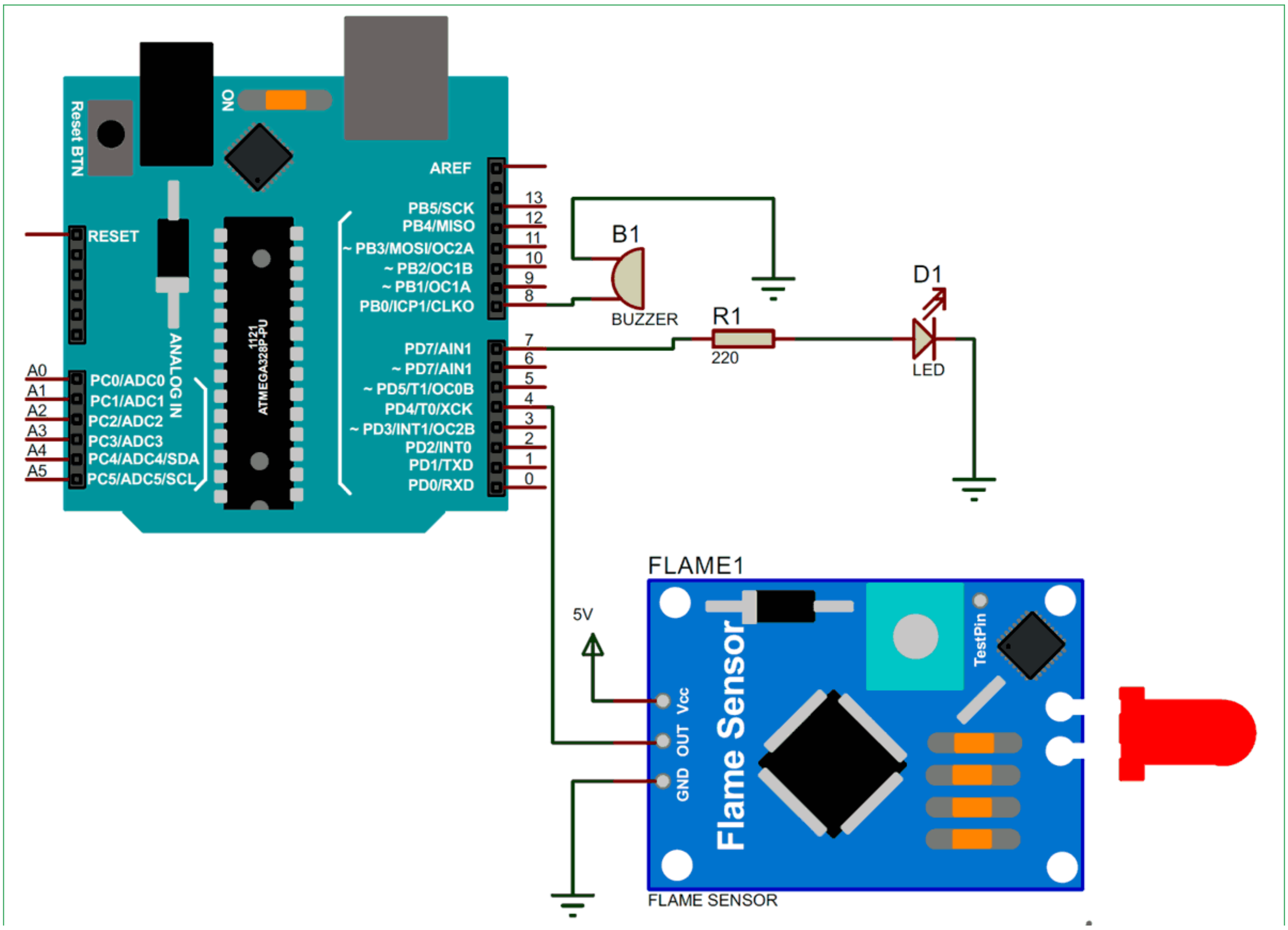


Figure 4.15: Connection of flame sensor with Arduino Uno

Servo Motor:

It is an electrical device which can push or rotate an object to a specific angle or distance. It is just made up of a motor which runs through servo mechanism. In this project we use a dc servo motor Sg90. These are used in many applications like RC helicopters and planes, robotics, machines, etc.

It is a closed loop system where it uses a positive feedback system to control motion and final position of the shaft. A servo motor consists of a potentiometer, gear assembly and a controlling circuit. Gear assembly is used to reduce RPM and increase torque of the motor. Initially there is no electricity generated at the output of the potentiometer port. Now an electric signal is given to another input of the error detector amplifier. The difference between the two signals is calculated and the output is fed back to the motor as an error signal and the motor starts rotating. This motor shaft is connected to the potentiometer and as the motor rotates so does the potentiometer which leads to the generation of a signal. So, as the potentiometer's angular position changes, its output feedback signal also changes. This continues until the position of the potentiometer reaches a value where the output of the potentiometer and the signal provided are the same. At this condition the motor stops rotating as there is no output signal from the amplifier to the motor input. The servo motor is controlled by PWM and has minimum, maximum pulse and repetition rate. The motor can turn up to 90degrees in either direction from a neutral position. This rotation of the motor is also dependent on the pulse width. It checks the pulse for every 20ms. Pulse of 1ms rotates the motor to 0degree, 1.5ms can rotate to 90degree and 2ms can rotate to 180degree. This can be clearly explained by the diagram given below.

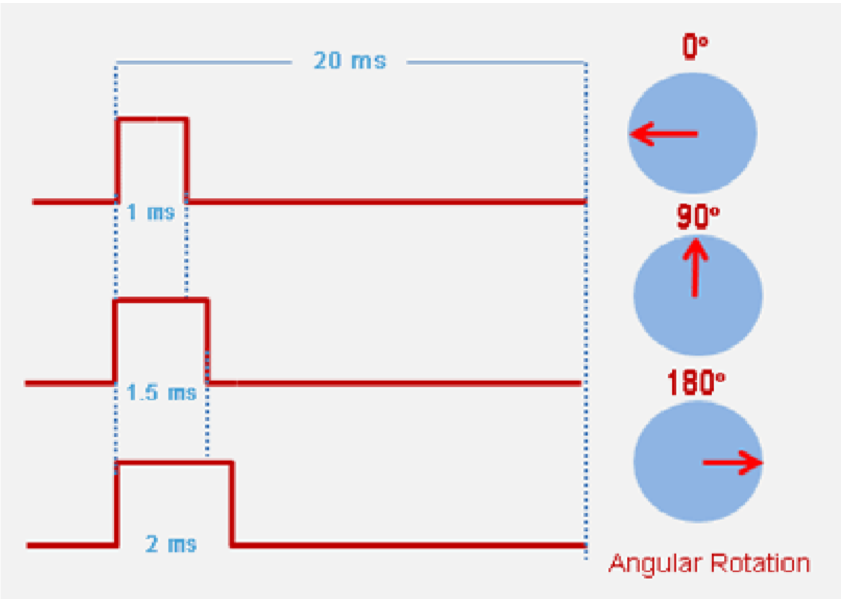
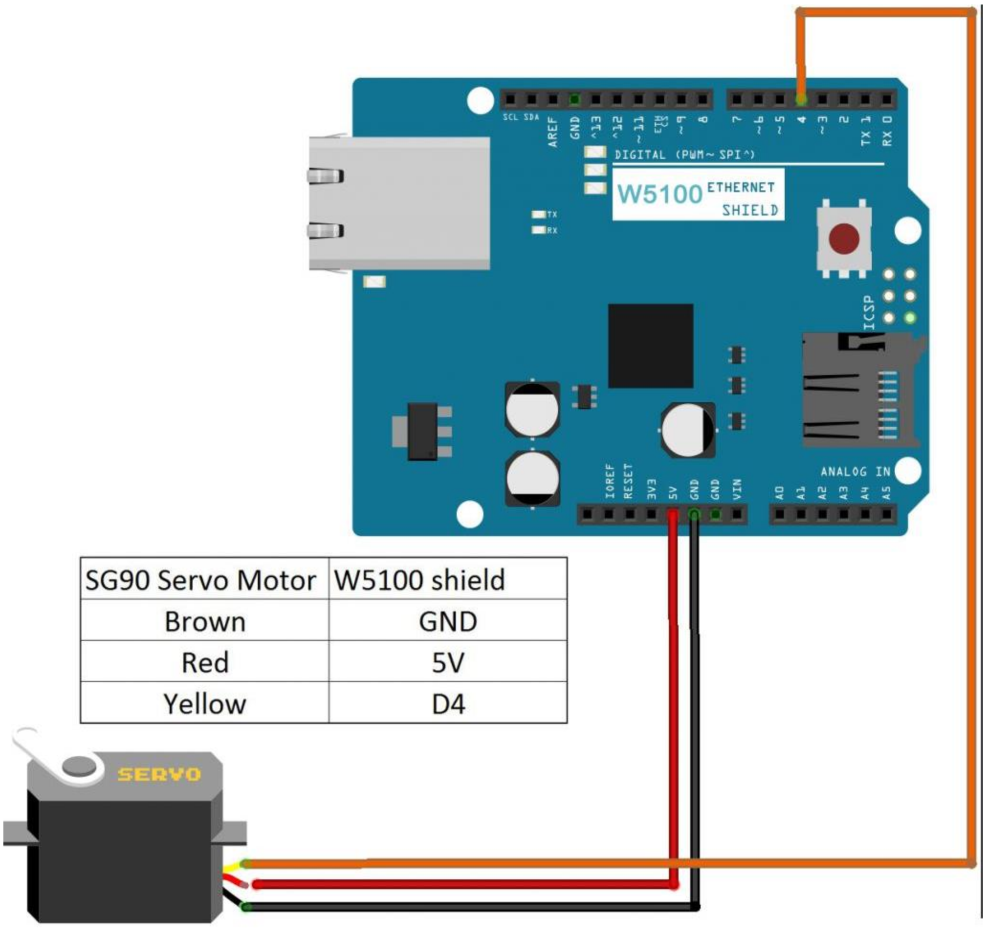


Figure 4.16: Potentiometer angular positions

Connection with Arduino Uno:



SG90 Servo Motor	W5100 shield
Brown	GND
Red	5V
Yellow	D4

Figure 4.17: Connection of servo motor with Arduino Uno

DC water pump:

The water pump can be defined as the pump which uses the principles like mechanical as well as hydraulic throughout a piping system and to make enough force for its future use.

The working principle of a water pump is mainly dependent on the positive displacement principle as well as kinetic energy to push the water. The pump used in this project uses a dc power supply.

Connection with Arduino Uno:

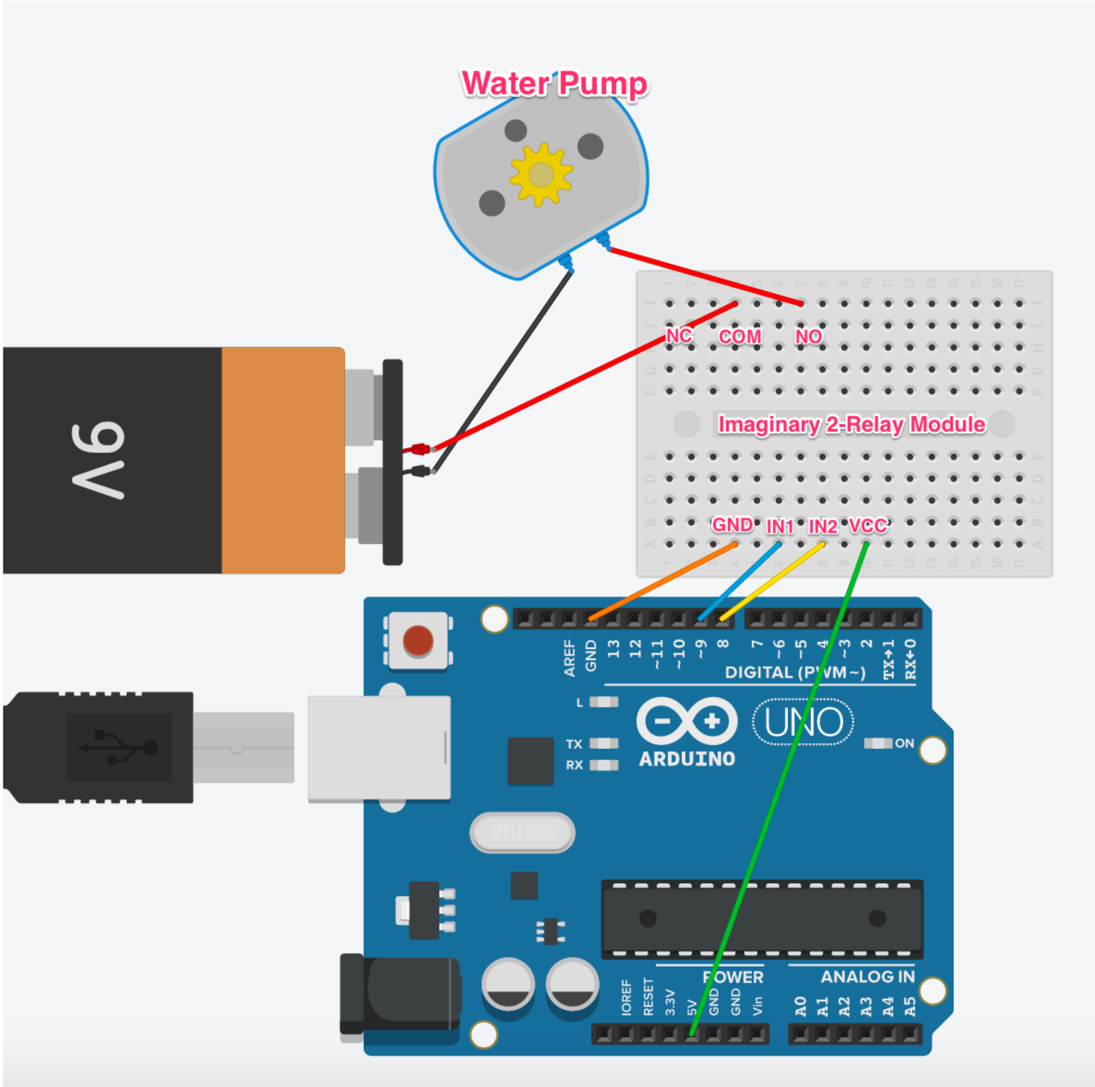


Figure 4.18: Connection of water pump with Arduino Uno

A relay can be used to control the water pumping action.

Block Diagram:

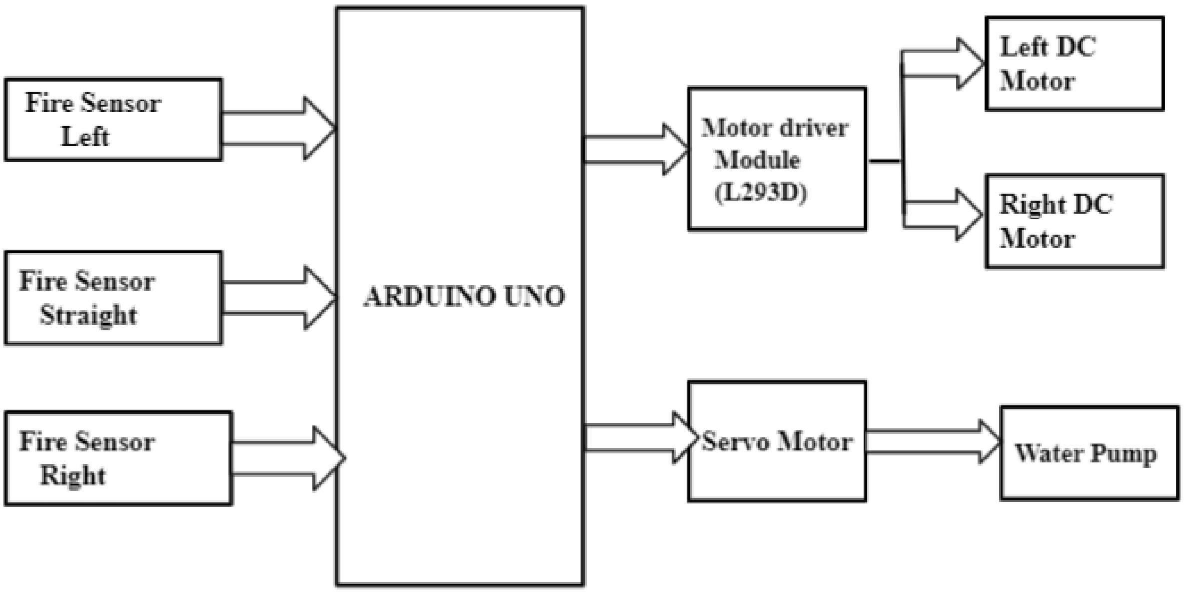


Figure 4.19: Block diagram

The flame sensors are given as inputs to the Arduino UNO which checks for any changes in the voltage and acts accordingly. The outputs of the Arduino UNO are the rotation of the left and right DC motors and water pump. The working of the dc motors is dependent on the voltage supplied by the motor driver module. The working of the water pump is dependent on the servo motor.

Software used:

ARDUINO IDE:

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions for starting the sketch, and the main program loop that is compiled and linked with a program stub main () into an executable cyclic executive program with the GNU toolchain, which is also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

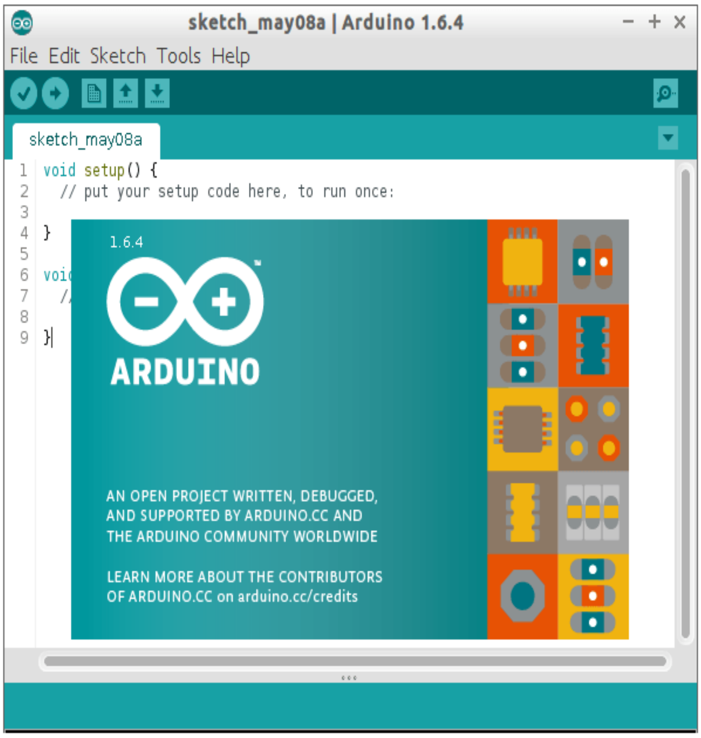


Figure 4.20: Arduino Integrated Development Environment (IDE)

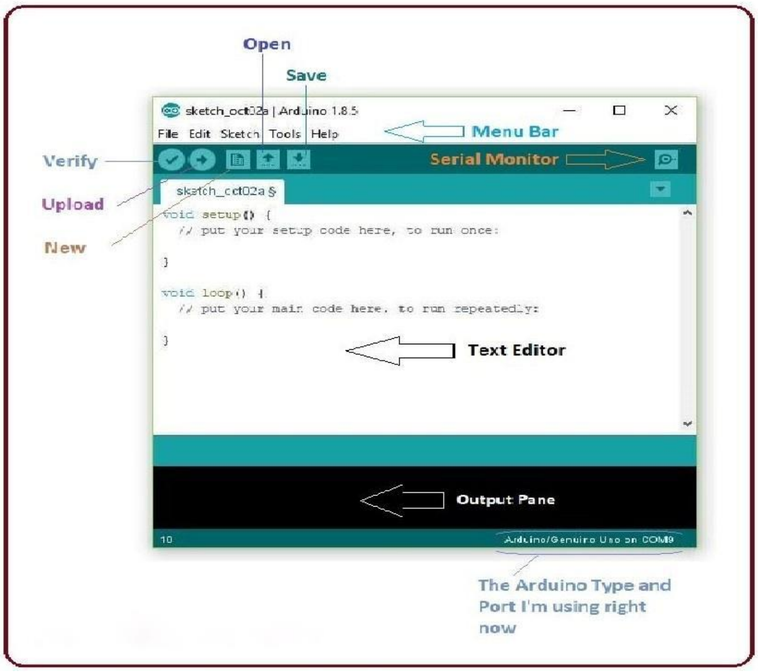


Figure 4.21: Arduino Integrated Development Environment (IDE) tools

There are 6 buttons in the menu tab which relate to the running program. The serial monitoring window helps us to debug the program where we can monitor the operation of the program. The 6 buttons are:

- Verify: to check the code for any errors.
- Upload: to transfer the code to the required Arduino board.
- New: to open a new tab.
- Open: to open any pre saved program.
- Save

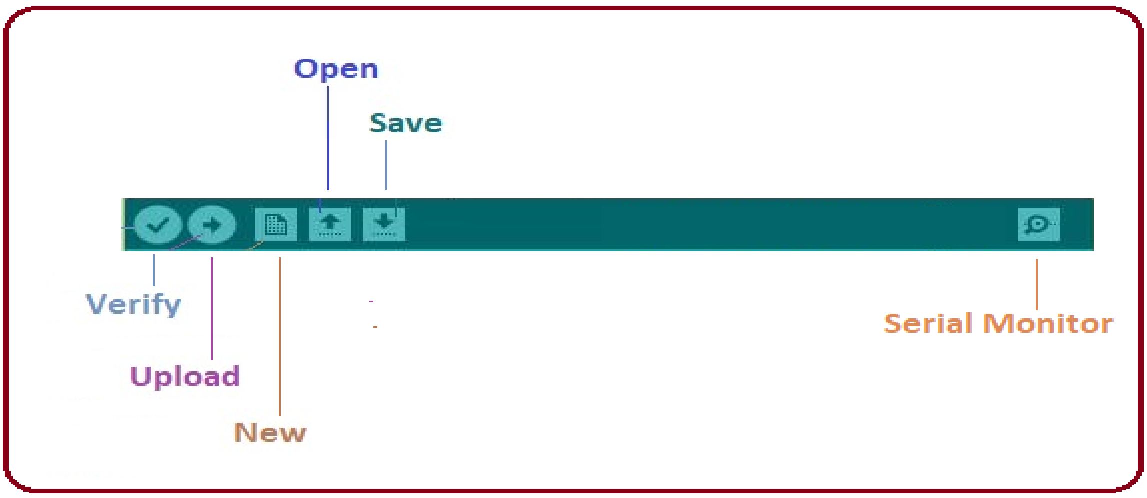


Figure 4.22: Arduino Integrated Development Environment (IDE) buttons

This window helps us to identify the errors easily.

Figure 4.23: Arduino Integrated Development Environment (IDE) error window

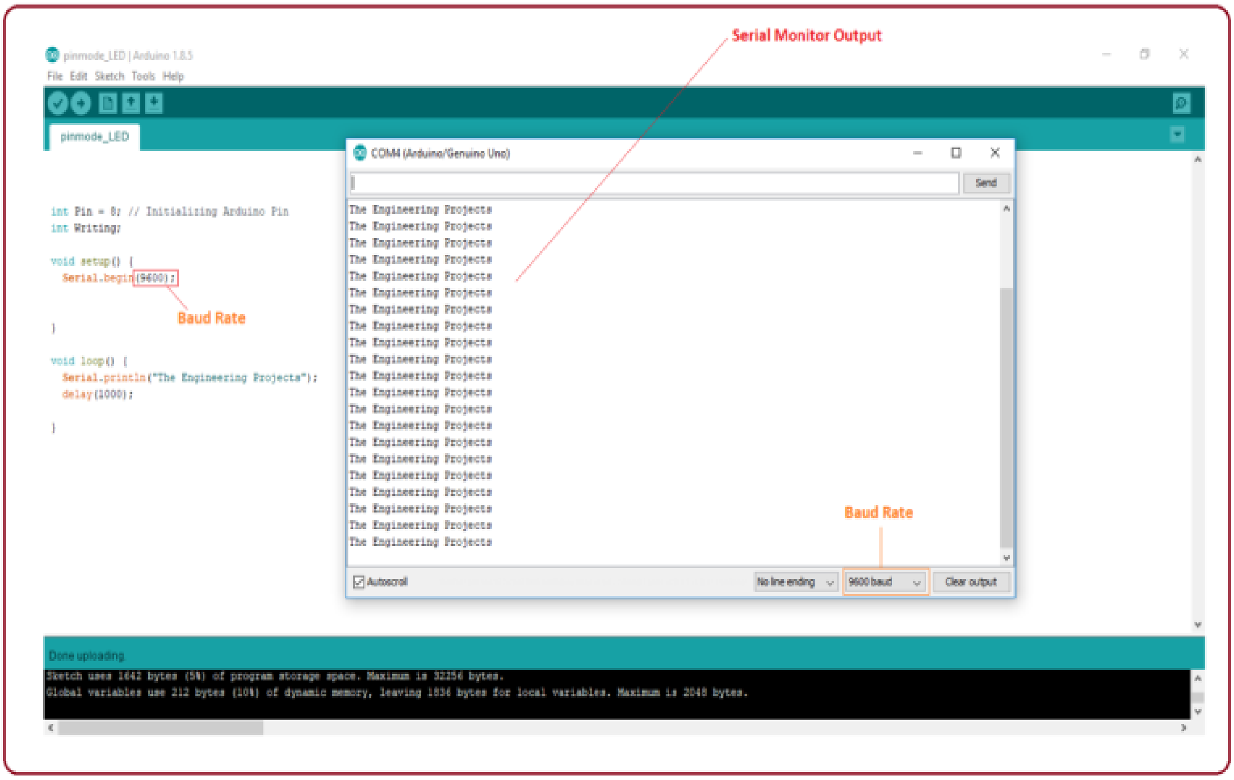


Figure 4.24: Arduino Integrated Development Environment (IDE) output window

Advantages:

- Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
- It is an official Arduino software, making code compilation so easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
- It is easily available for operating systems like MAC, Windows, Linux and runs on the Java platform that comes with inbuilt functions and commands that play a major role for debugging, editing and compiling the code in the environment.
- A range of Arduino modules are available in Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.
- Each of them contains a microcontroller on board that is programmed and accepts the information in the form of code.
- The main code also known as sketch which is created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
- The IDE environment mainly contains two basic parts: Editor and Compiler. The former one is used for writing the required code and the latter is used for compiling and uploading the code into the given Arduino Module.
- This environment supports both C and C++ languages.
- Contains a bootloader which helps in burning the code directly into the controller.
- Easy commands like digitalWrite or digitalRead are used for addressing the Arduino input and output pins.

CHAPTER 05

RESULT AND DISCUSSION

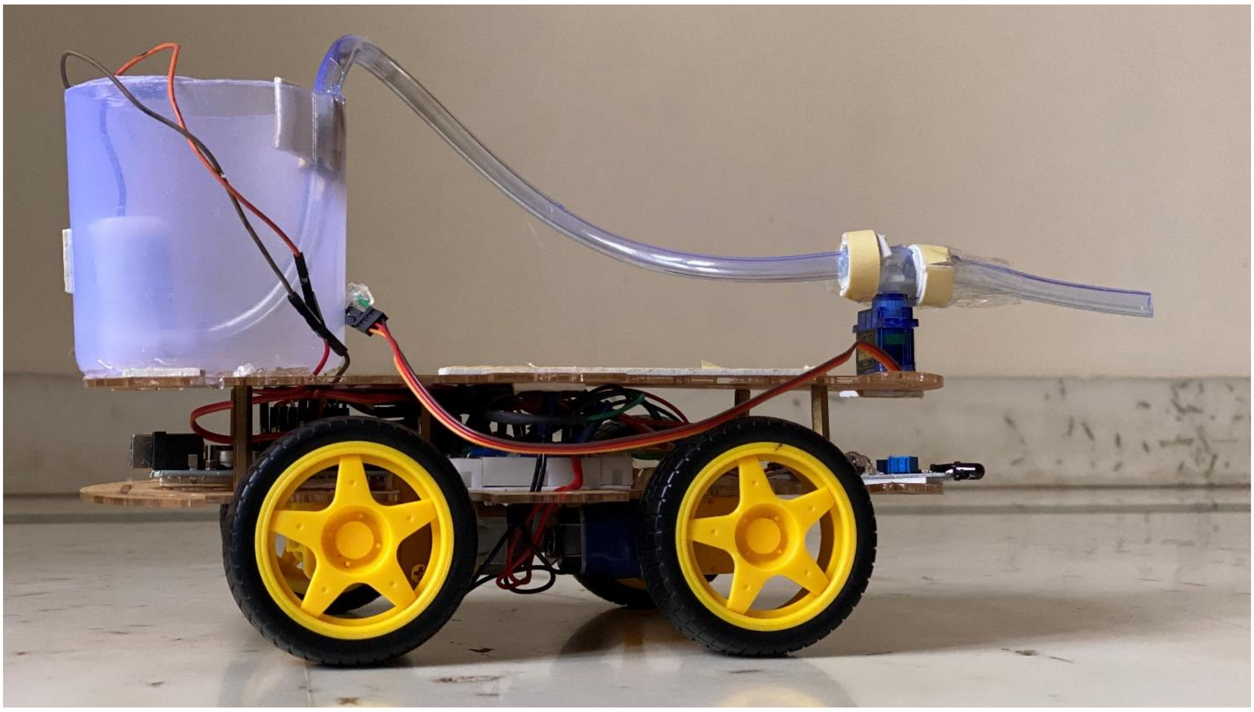


Figure 5.1: Model of the project 1

The above figure shows us that the project was successfully implemented, and the working of the robot can be clearly observed from the above picture. The robot is fully automated with the help of sensors and Arduino Uno. It was able to detect even a small amount of fire and act accordingly. The robot would move in the direction of fire detected. The movement of the robot would depend on the distance between the robot and fire. The threshold values given are 50 and 130 which indicates the starting and stopping position of the robot. It would move accordingly and stop in front of the fire in order to extinguish it. The servo motor turns ON when the robot stops and guides the water pumping action. It helps in detection of the direction of fire.

Output across the flame sensors:

When no fire detected the signal, the LED is turned OFF.

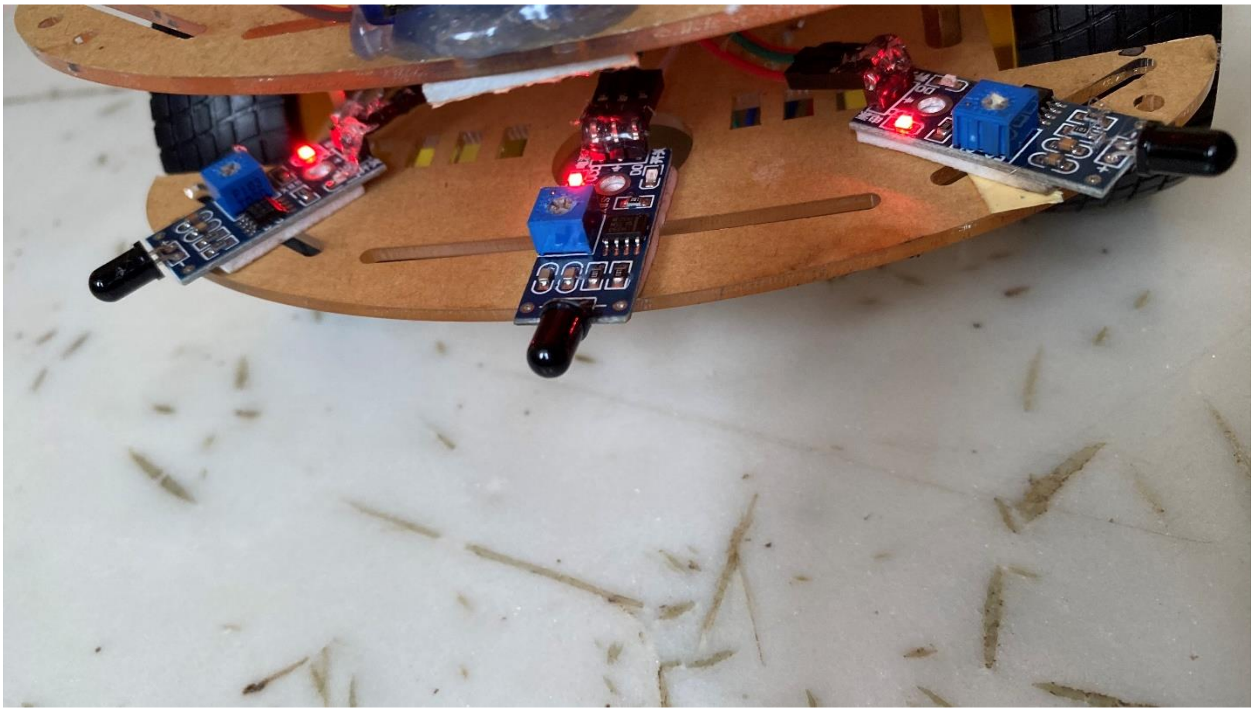


Figure 5.2: Model of the project 2

When the fire is detected the D0 pin of the sensor goes LOW. As a result, the signal LED turns ON.

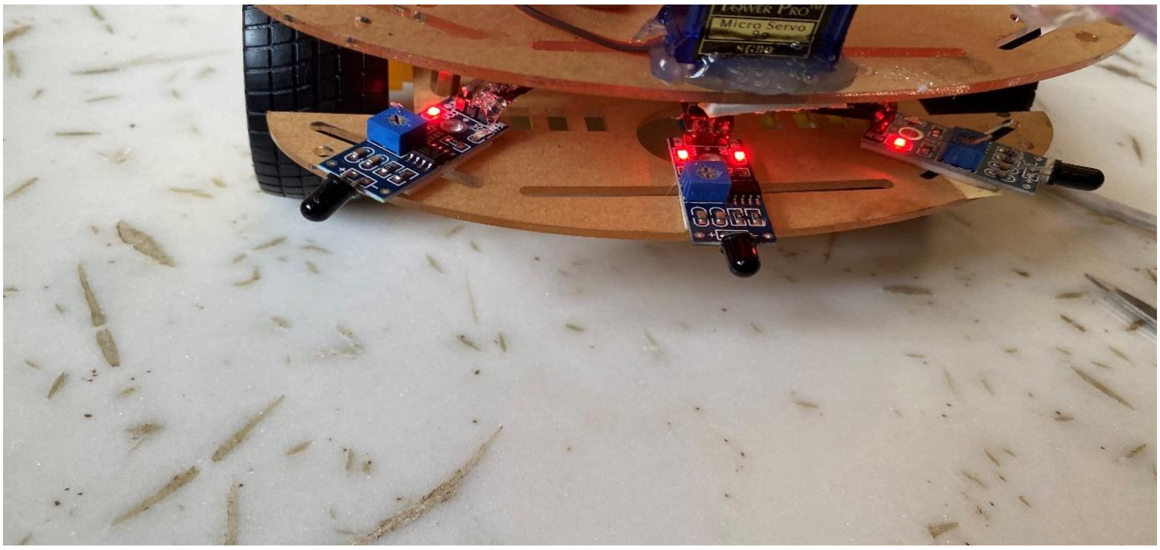


Figure 5.3: Model of the project 3

CHAPTER 06

CONCLUSION AND FUTURE SCOPE

The project has been motivated by the desire to design a system that can detect fire and take appropriate actions without any human interference. 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. This provides us the opportunity to pass on to robot's tasks that traditionally humans had to do which were life threatening.

Firefighting is an obvious step for such automation. Given the number of lives lost regularly in firefighting, 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 enhancement of technology in this field will surely lead to its progress and these robots can be made much more automated which can completely work without any human interference. With further improvements this can be used in areas which are prone to such fire accidents causing the loss of many lives. With the addition of a location tracking system, the robot can easily track the location of fire accidents so that it can reach that place within no time to extinguish it.

Advantages:

- To detect the exact direction of the fire source.
- Capability of sensing accurately with increased flexibility.
- Reduces human effort.
- Reliable and economical and not sensitive to weather conditions.
- Arduino Uno is an open source and extensible hardware.

Challenges:

- Our system is used for less than 3.5kg applications.
- It is not used to put out large fires.
- No monitoring system for the vehicle as it is not remote controlled.
- To use Arduino Uno the project must be made as small as possible.

REFERENCES

1. Aswinth Raj, "Arduino Based Fire Fighting Robot", *Circuit Digest*.

2. Rutuja Gadkari," **Survey on Fire Fighting Robot Controlled Using Android Applications**", IJIRSET, IJIRSET, 2319-8753, 2007.

3. Abdulkadir CAKIR and Nyan Farooq Ezzulddin," **Fire-Extinguisher Robot Design Using Arduino**", Publication Name: IOSR journal of Computer Engineering (IOSR-JCE), 2278-0661, 2016.

4. Dr. Niranjan Bhattacharyya and Akshat Sharma," **Fire Fighting Robot**", Publication Name: IJIET, 2319-1058, 2018

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

6. Mukul Diwanji, Saurabh Hisvankar, Chhaya Khandelwal, "**Autonomous Fire Detecting and Extinguishing Robot"**, Publisher: IEEE, 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)

7. Krishanu Bhattacharjee, Malabika Chatterjee, Rima Nayek, Debapriya Ghosh, "**Fire Fighting Robotic Vehicle Using Arduino"**, Published on: ijareeie.com, August 2020.

APPENDIX

code for fire extinguisher robot is as follows:

```
/*------ Arduino Fire Fighting Robot Code----- */


#include <Servo.h>

Servo myservo;          //creating an object named myservo to which we connect our servo


int pos = 0;

boolean fire = false;


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

#define Left_S 9      // left sensor

#define Right_S 10      // right sensor

#define Forward_S 8 //forward sensor


/*-------defining Outputs------                          forward       right    left    stop    */

#define LM1 5      // left motor        2(input 1)          high           high    high    high

#define LM2 4      // left motor        7(input 2)           low           low     high     high

#define RM1 3      // right motor   10(input 3)            high           high    high    high

#define RM2 2      // right motor   15(input 4)             low           high    low    high

#define pump 6
```

```
void setup()
{
  pinMode(Left_S, INPUT);
  pinMode(Right_S, INPUT);
  pinMode(Forward_S, INPUT);
  pinMode(LM1, OUTPUT);
  pinMode(LM2, OUTPUT);
  pinMode(RM1, OUTPUT);
  pinMode(RM2, OUTPUT);
  pinMode(pump, OUTPUT);

  myservo.attach(11);  //servo is connected to 11th pin of arduino
  myservo.write(90);   //servo rotation is initiased to 90 degree
}

void put_off_fire()
{
    delay (500);
    digitalWrite(LM1, HIGH);
    digitalWrite(LM2, HIGH);
    digitalWrite(RM1, HIGH);
    digitalWrite(RM2, HIGH);

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

```
   for (pos = 50; pos <= 130; pos += 1) {
   myservo.write(pos);
   delay(10);
  }
  for (pos = 130; pos >= 50; pos -= 1) {
   myservo.write(pos);
   delay(10);
  }

  digitalWrite(pump,LOW);
  myservo.write(90);

  fire=false;
}

void loop()
{
   myservo.write(90); //Sweep_Servo();

   if (digitalRead(Left_S) ==1 && digitalRead(Right_S)==1 && digitalRead(Forward_S)
==1) //If Fire not detected all sensors are zero
   {
   //Do not move the robot
   digitalWrite(LM1, HIGH);
```

```
digitalWrite(LM2, HIGH);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, HIGH);

}


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

{

//Move the robot forward

digitalWrite(LM1, HIGH);

digitalWrite(LM2, LOW);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, LOW);

fire = true;

}


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

{

//Move the robot left

digitalWrite(LM1, HIGH);

digitalWrite(LM2, HIGH);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, LOW);

      fire = true;

}
```

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

   {

   //Move the robot right

   digitalWrite(LM1, HIGH);

   digitalWrite(LM2, LOW);

   digitalWrite(RM1, HIGH);

   digitalWrite(RM2, HIGH);

          fire = true;

   }


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


    while (fire == true)

    {

     put_off_fire();

    }

}
```