FIRE FIGHTING ROBO CAR

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Electrical and Electronics Engineering

by

KATTE GOPAL (219X5A02D7)

TUNGA SAI RATNA TEJA REDDY (209X1A02C4)

PANYAM VENKATA TEJA (209X1A0287)

BYREDYY UMESH REDDY (209X1A02B2)

Under the Esteemed Guidance of

Dr. N. Ravi Shankar Reddy, M. Tech., Ph. D

Associate Professor

Department of EEE.

Department of Electrical & Electronics Engineering

G. Pulla Reddy Engineering College (Autonomous): Kurnool

(Affiliated to Jawaharlal Nehru Technological University-Anantapur, Ananthapuramu)

2023-2024

Department of Electrical & Electronics Engineering
G. Pulla Reddy Engineering College (Autonomous): Kurnool
(Affiliated to Jawaharlal Nehru Technological University-Anantapur, Ananthapuramu)

Certificate

This is to certify that the project entitled **"Fire Fighting Robo Car"** is a bonafide work done by **Katte Gopal (219X5A02D7), Tunga Sai Ratna Teja Reddy(209X1A02C4), Panyam Venkata Teja(209X1A0287), Byreddy Umesh Reddy(209X1A02B2)** in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Electrical and Electronics Engineering** during the academic year 2023-2024.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree.

Project guide
Dr. N. Ravi Shankar Reddy, M. Tech., Ph. D
Associate Professor,
Department of EEE,
G. Pulla Reddy Engineering
College (Autonomous),
Kurnool, A.P.,

Head of the Department
Dr. K. Sri Gowri, M. Tech., Ph. D
Professor & Head of the Department,
Department of EEE,
G. Pulla Reddy Engineering
College (Autonomous),
Kurnool, A.P.,

External Examiner

DECLARATION

We hereby declare that the project titled "**Fire Fighting Robo Car**" is an authentic work carried out by us as a students **of G. PULLA REDDY ENGINEERING COLLEGE (Autonomous) Kurnool,** during 2023-24 and has not been submitted elsewhere for the award of any degree or diploma in part or in full to any institute.

KATTE GOPAL (219X5A02D7)

TUNGA SAI RATNA TEJA REDDY (209X1A02C4)

PANYAM VENKATA TEJA (209X1A0287)

BYREDDY UMESH REDDY (209X1A02B2)

Acknowledgement

We feel great pleasure to express our deepest sense of gratitude and sincere thanks toour respected guide **Dr. N. Ravi Shankar Reddy**, **Associate Professor**, Department of Electrical & Electronics Engineering, G. Pulla Reddy Engineering College for his valuable guidance, encouragement and help for completing this work. His useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

With a great sense of pleasure, we extend our gratitude to **Dr. K. Sri Gowri**, **Professor & Head** of the Electrical & Electronics Engineering Department, G. Pulla Reddy Engineering college and for her cooperation and providing necessary help for completing this project.

We wish to express our sense of gratitude **to Dr. B. Sreenivasa Reddy** for providing requisite facilitis and heliping us in providing such a good environment.

Finally, it is our pleasure to thank all faculty, staff, and friends for their constructive criticisms, which made us to work hard to produce a better report.

KATTE GOPAL (219X5A02D7)

TUNGA SAI RATNA TEJA REDDY (209X1A02C4)

PANYAM VENKATA TEJA (209X1A0287)

BYREDDY UMESH REDDY (209X1A02B2)

Abstract

The purpose of this project is to provide a design for an Arduino-based Fire Fighter Robot. Arduino is an open-source electronics platform that is based on easy-to-use hardware and software. Arduino board can be controlled by sending a set of instructions to the microcontroller. It has been designed to develop a fire-fighting robot using Arduino technology for remote operation. Firefighting is the act of extinguishing fires, i.e.; our robot sprinkles water onto the fire. The robotic vehicle is loaded with a water tanker and a pump to throw water. An Arduino microcontroller is sued for the desired operation.

A firefighter robot suppresses and extinguishes fires to prevent loss of life and destruction of property and the environment. This fire-fighting robot can be used as a supplementary to the firefighters in critical situations. To function this robot, a flame sensors has been used. The flame sensor is used to detect the fireplace, the robot can run in both a manual control system and an autonomic control system. This Robo car can also be manually operated via the Arduino application on mobile devices using a Bluetooth connection. This functionality enables users to direct the vehicle to areas where fires are located beyond the range of sensors.

Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
IDE	Integrated Development Environment
IC	Integrated Circuit
IoT	Internet of Things
IR	Infrared Rays
LED	Light Emitting Diode
HC	Host Controller
BO	Battery Operated

CHAPTER 1

Introduction

1.1 Foreword:

With the ever-increasing technology, the developments are increasing in the face of the situations that cause human life. Every day, the robot that is produced as an alternative to the human element in a new branch. Flying, robots, wheeled robots legged robots, humanoid robots, and underwater robots are just some of them. The growing world population is bringing involuntary problems together. Fires are among the most important of these problems. The robot industry has a lot of work in this area. Some of these are fixed mobile robots with different features, which are equipped with different sensors that detect before the fire is out, mobile rescue robots as fire search and rescue equipment, mobile locating robots used for fire detection, fire extinguishing robots in many different models designed to assist firefighters in the fire.

An embedded system is one kind of computer system mainly designed to perform several tasks like accessing, processing, and store, and controlling the data in various electronics-based systems. Embedded systems are a combination of hardware and software which is shown in Fig 1.1 where software is usually known as firmware that is embedded into the hardware. Embedded systems support to make the work more perfect and convenient. The applications of embedded systems mainly involve in our real-life for several devices like microwaves, calculators, neighborhood traffic control systems, etc.

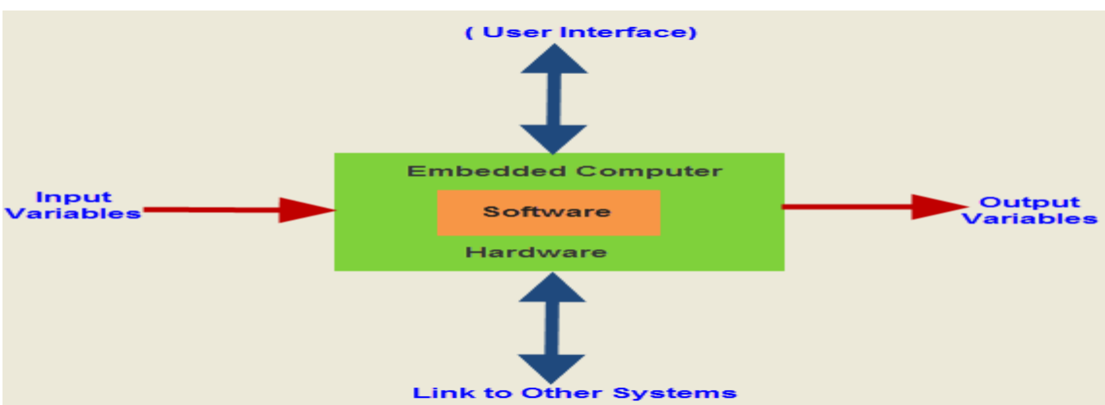

Fig 1.1: Overview of embedded system

1.2 Block Diagram:

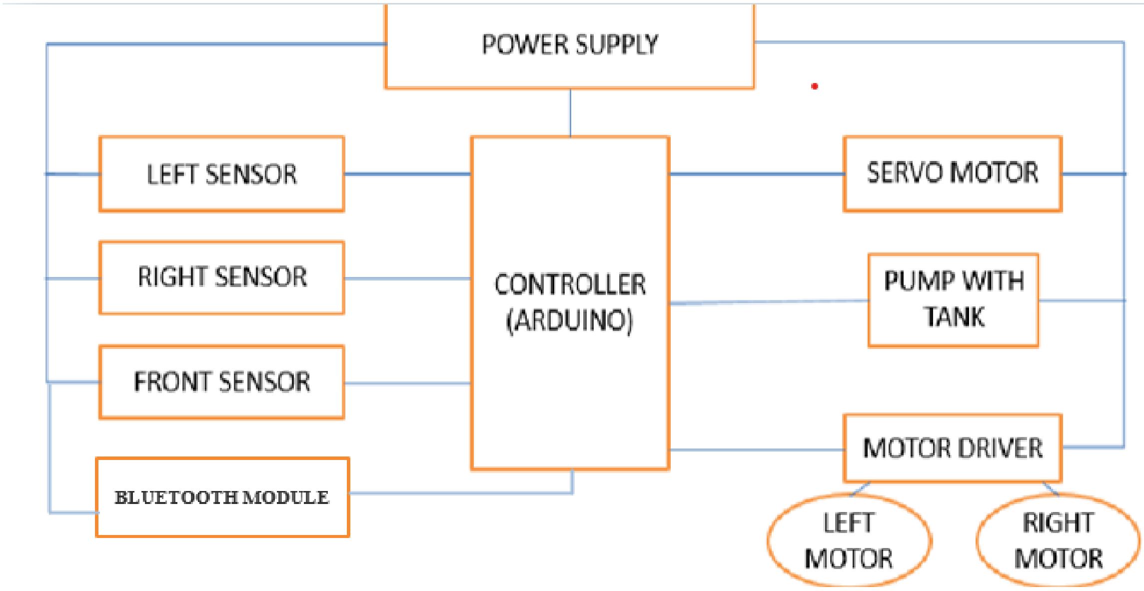

Fig1.2 Block diagram of project

This block diagram of Fig 1.2 consists of different parts that are used to make this model and this is the place of alignments of the parts according to the connections. Based on this block diagram we connected every part and made this prototype to a working model this isthe short and easy way to check all the parts that are present in the model.

1.3 Objective:

The death rate belongs to fire accidents and forest fires has increased. Initially, this fire starts from a smaller area. If we can find out the starting place before the fire gets widespread, we can inhibit it. In this project, the main objective is to stop the spread of this fire. For that, we have to detect the fire in its initial stage. The fire sensor which is placed in the project will determine the fire. The surroundings of the robot should be tracked for avoiding collisions. A wireless cathartic is fixed to it covers the surroundings. Robot motion plays a major role in the project. All the components are connected to microcontrollers and it acts as the heart of the project. All of the commands related to fire detection signals are given to that microcontroller it operates the remaining components of the robot in moving and pumping of the water.

CHAPTER 2

Problem Statement

2.1 Problem Formulation:

Fires are serious hazards that can cause severe damage to property and human life. In many instances, it can be challenging for firefighters to access and control the fire due to its location or intensity. Additionally, monitoring the affected area for potential re-ignition or other safety hazards can be a daunting task. Traditional firefighting methods are often inefficient and can put firefighters' lives at risk. To address these challenges, the implementation of an IoT-based firefighting and affected area monitoring robot is proposed. This robot will be equipped with various sensors and cameras to detect the location and intensity of the fire, as well as monitor the affected area for potential safety hazards. The robot will be able to navigate through difficult terrain and access hard-to-reach areas, reducing the risk to human life.

The design and implementation of this robot pose several challenges, including the integration of various sensors, the development of an efficient navigation system, and the establishment of reliable communication channels. Additionally, the robot's hardware and software must be designed to withstand high temperatures, smoke, and other hazards present in firefighting scenarios. Overall, the implementation of an IoT-based firefighting and affected area monitoring robot would significantly improve the efficiency and safety of firefighting operations, and this report will outline the challenges and solutions associated with its design and implementation.

2.2 Literature Survey:

[1] Tawfiqur Rakib, M. A. Rashid Sarkar proposed a fire fighting robot model which consists of a base platform made up of 'Kerosene wood', LM35 sensor for temperature detection, flame sensors to detect the fire and a water container of 1 litre capacity which is made up of a strong cardboard that makes it water resistant. The robot has two wheels for its movement.

[2] Saravanan P. ,Soni Ishawarya proposed a model which uses Atmega2560 micro-controller and in which the robot is divided into three basic units according to their functions which are as locomotive unit, fire detecting unit and extinguishing unit.Each unit performs their task in order to achieve the desired output of extinguishing fire.The locomotive unit is used for the movement of the robot and to avoid the obstacles with the help of four IR and four ultrasonic sensors.The fire detecting unit is used to detect fire using LDR and temperature sensor. The extinguishing unit is used to extinguish the fire using water container and BLDC motor. The robot also have a bluetooth module that is connected with the smartphones in order to navigate it in the proper direction.

[3] S. Jakthi Priyanka,R. Sangeetha proposed an android controlled fire fighting robot which uses Arduino UNO R3. The robot consists of gas sensor for fire detection, gear motor and motor drive for the movement of robot, a bluetooth module to connect the robot with the android device and to control the robot with the smartphone as well. Water pump and sprinkler is also used in this. To instruct the Arduino UNO an open source software which is Arduino IDE is required to code and to implement that code in Arduino UNO.

[4] Nagesh MS, Deepika T V , Stafford Michahial, Dr M Shivakumar proposed a fire extinguishing robot which employs DTMF (Dual Tone Multi Frequency Tones) technology for the navigation of the robot and uses a flame sensor for fire detection that is capable of sensing flame of the wavelength range 760 to 1100 nm and sensitivity varies from 10cm to 1.5feet.

[5] Sushrut Khajuria, Rakesh Johar, Varenyam Sharma, Abhideep Bhatti proposed an arduino based fire fighter robot which consists of RF based remote operation to operate the robot and water pump. The robot is controlled by the user within a range of 7 metres. It also consists of a wireless camera which helps user to move the robot in the required direction.

[6] Khaled Sailan, Prof. Dr.-Ing. Klaus-Dieter Kuhnert, Simon Hardt proposed an obstacle avoidance robot named as Amphibious Autonomous Vehicle. In this robot, a fuzzy controller is used to avoid static obstacle in real time. It aims to guide the robot or vehicle along its path avoiding all the obstacle that comes along the path.

CHAPTER- 3

Requirement Analysis

3.1 Basic requirements of the robo car:

To design and implement an IoT-based firefighting and affected area monitoring robot, the following requirements must be considered:

1. **Sensing capabilities:** The robot must be equipped with sensors to detect fire, smoke, temperature, and other safety hazards present in the affected area. The sensors must be accurate and reliable, with the ability to transmit data in real-time to the central control system.
2. **Navigation system:** The robot must have a navigation system that enables it to navigate through challenging terrain, avoid obstacles, and locate the source of the fire. The navigation system must be accurate, reliable, and efficient, with the ability to adapt to changing environments.
3. **Communication system:** The robot must have a reliable communication system to transmit data to the central control system. The communication system must be able to transmit data in real-time and must be able to operate in environments with poor connectivity.
4. **Power supply:** The robot must have a reliable power supply system that can support its operations for an extended period. The power supply system must be robust enough to withstand high temperatures and other hazards present in firefighting scenarios.
5. **Safety features:** The robot must have safety features that protect it from damage or destruction in case of accidents or hazards. The safety features must also protect firefightersand other personnel from harm during firefighting operations.
6. **Integration with the central control system:** The robot must be integrated with the central control system to enable firefighters to monitor and control its operations remotely. The central control system must be easy to use and must provide real-time data on the robot’s operations.
7. **Durability and maintenance:** The robot must be designed to withstand harsh environments and rough handling. Additionally, it must be easy to maintain and repair .

3.2 Feasibility Studies and Risk Analysis of the Project

1. **Technical Feasibility:** This study assesses the technical feasibility of the project. It involves evaluating the technical requirements, such as sensors, communication systems, power supply, and navigation systems. This study will help determine the project'stechnical feasibility and identify potential technical challenges.
2. **Economic Feasibility:** This study assesses the economic feasibility of the project. It involves evaluating the project's cost, including the development, maintenance, andoperation costs. This study will help determine the project's viability in terms of financialresources.
3. **Environmental Feasibility:** This study assesses the environmental feasibility of the project. It involves evaluating the project's impact on the environment, such as the use of batteries and the disposal of electronic waste. This study will help identify potential environmental risks associated with the project.
4. **Social Feasibility:** This study assesses the social feasibility of the project. It involves evaluating the project's impact on society, such as job creation, improved safety, and reduced damage to property. This study will help determine the project's social viability and identify potential social risks associated with the project.
5. **Legal Feasibility:** This study assesses the legal feasibility of the project. It involves evaluating the project's compliance with legal regulations, such as safety standards and privacy laws. This study will help identify potential legal risks associated with the project.
6. **Environmental risks**, such as the robot's exposure to high temperatures and other hazardsin firefighting scenarios.
7. **Safety risks**, such as the potential for the robot to cause harm to firefighters or other personnel if it malfunctions or collides with them.
8. **Cybersecurity risks**, such as the potential for hackers to access the robot's communicationsystem or sensors, compromising its operations.
9. **Legal risks**, such as the potential for the robot to violate privacy laws if it captures imagesor data without consent.

3.3 Hardware Requirements:

Arduino:

Fig 3.1 Arduino Uno

The Arduino programming language is a simplified version of C/C++. If you know C, programming Arduino will be familiar. The power of the Arduino is not its ability to crunch code, but rather its ability to interact with the outside world through its input-output (I/O) pins. The Arduino has 14 digital I/O pins labeled 0 to 13 in Fig 3.1 that can be used to turn motors and lights on and off and read the state of switches.

Each digital pin can sink or source about 40 mA of current. This is more than adequate for interfacing to most devices, but does mean that interface circuits are needed to control devices other than simple LED's. In other words, you cannot run a motor directly using the current available from an Arduino pin, but rather must have the pin drive an interface circuit that in turn drives the motor. A later section of this document shows how to interface to a small motor. To interact with the outside world, the program sets digital pins to a high or low value using C code instructions, which corresponds to +5 V or 0 V at the pin. The pin is connected to external interface electronics and then to the device being switched on and off.

Fire sensor:

An infrared sensor is an electronic device, which emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects motion. These types of sensors measure only infrared radiation, rather than emitting it 9 which is called a sensor as shown in Fig 3.2. Usually, in the infrared spectrum, all objects radiate some form of thermal radiation. These types of radiation are invisible to our eyes, and can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode that is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances, and these output voltages, change in proportion to the magnitude of the IR.

Fig 3.2 Fire Sensor

Battery:

Fig 3.3 Battery

A rechargeable battery in Fig 3.3 is an energy storage device that can be charged again afterbeing discharged by applying DC current to its terminals.

Rechargeable batteries allow for multiple usages from a cell, reducing waste and generally providing a better long-term investment in terms of dollars spent for usable device time. This is true even factoring in the higher purchase price of rechargeable and the requirementfor a charger.

A rechargeable battery is generally a more sensible and sustainable replacement to one-time-use batteries, which generate current through a chemical reaction in which a reactive anode is consumed. The anode in a rechargeable battery gets consumed as well but at a slower rate, allowing for many charges and discharges.

In use, rechargeable batteries are the same as conventional ones. However, after discharge the batteries are placed in a charger or, in the case of built-in batteries, an AC/DC adapter is connected. Some types of batteries such as nickel-cadmium and nickel-metal hydride can develop a battery memory effect when only partially discharged, reducing theperformance of subsequent charges and thus battery life in each device. Rechargeable batteries are used in many applications such as cars, all manner of consumer electronics, and even off-grid and supplemental facility power storage.

Servo Motor:

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you 12 want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor that runs through a servo mechanism in Fig 3.4. Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at 3kg/cmor 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift ata particular distance. For example, A 6kg/cm Servo motor should be able to lift 6kg if theload is suspended 1cm away from the motor shaft, the greater the distance the lesser the weight-carrying capacity. The position of a servo motor is decided by an electrical pulse and its circuitry is placed beside the motor.

Fig 3.4 Servo Motor

DC Water Pump:

DC-powered pumps in Fig 3.5 use direct current from a motor, battery, or solar power to move fluid in a variety of ways. Motorized pumps typically operate on 6, 12, 24, or 32volts of DC power. Solar-powered DC pumps use photovoltaic panels with solar cells thatproduce direct current when exposed to sunlight.

Fig 3.5 Water Pump

HC05 Bluetooth module:

The HC-05 is a widely used Bluetooth module known for its versatility and ease of use in electronics projects. It operates on the Bluetooth 2.0 standard and offers a range of approximately 10 meters, making it suitable for short-range wireless communication.

The module can be configured to work as either a master or slave device, allowing for flexible integration into various projects.

One of the key features of the HC-05 is its simple serial communication interface, which makes it easy to connect to microcontrollers like Arduino. This enables Arduino projects to communicate wirelessly with other Bluetooth-enabled devices such as smartphones, tablets, or other microcontrollers. The HC-05 typically requires a 3.3V or 5V power supply and has pins for power, ground, and serial communication, making it compatible with a wide range of microcontroller boards.

The HC-05 in Fig 3.6 is commonly used in projects such as remote-controlled cars, home automation systems, and wireless sensor networks, where reliable wireless communication is essential. Its affordability, ease of use, and wide availability make it a popular choice among hobbyists and professionals alike for adding Bluetooth functionality to their projects.

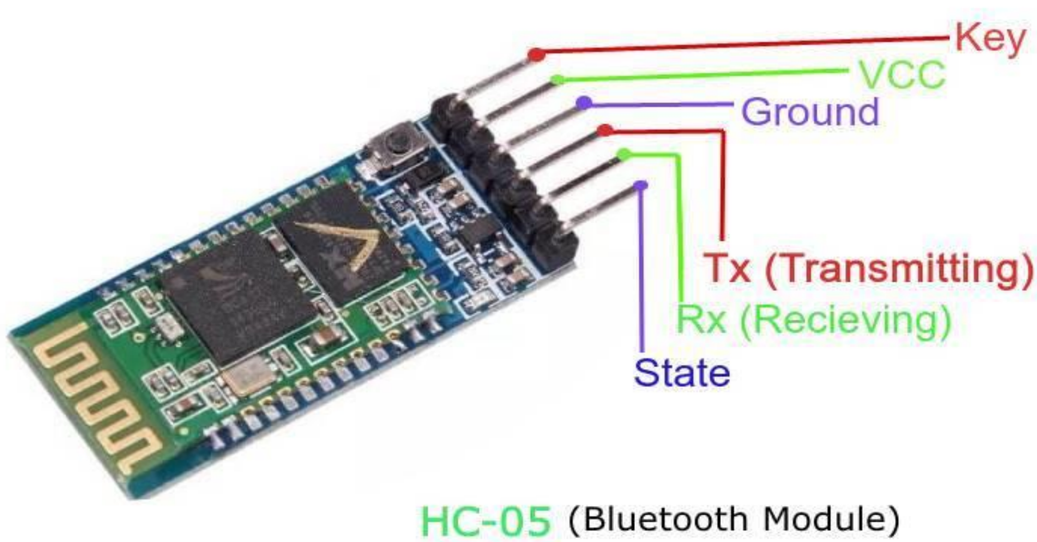

Fig 3.6 HC05 Bluetooth Module

Motor Driver:

A motor driver is an integrated circuit chip that is usually used to control motors in autonomous robots. Motor drivers act as an interface between Arduino and the motors. The most commonly used motor driver IC’s are from the L293 series such as L293D, L293N, etc. These IC’s are designed to control 2 DC motors simultaneously. L293D consists of two H-bridge. H-bridge is the simplest circuit for controlling a low current-rated motor. We will be referring to the motor driver IC as L293D only. Fig 3.7 has 16 pins.

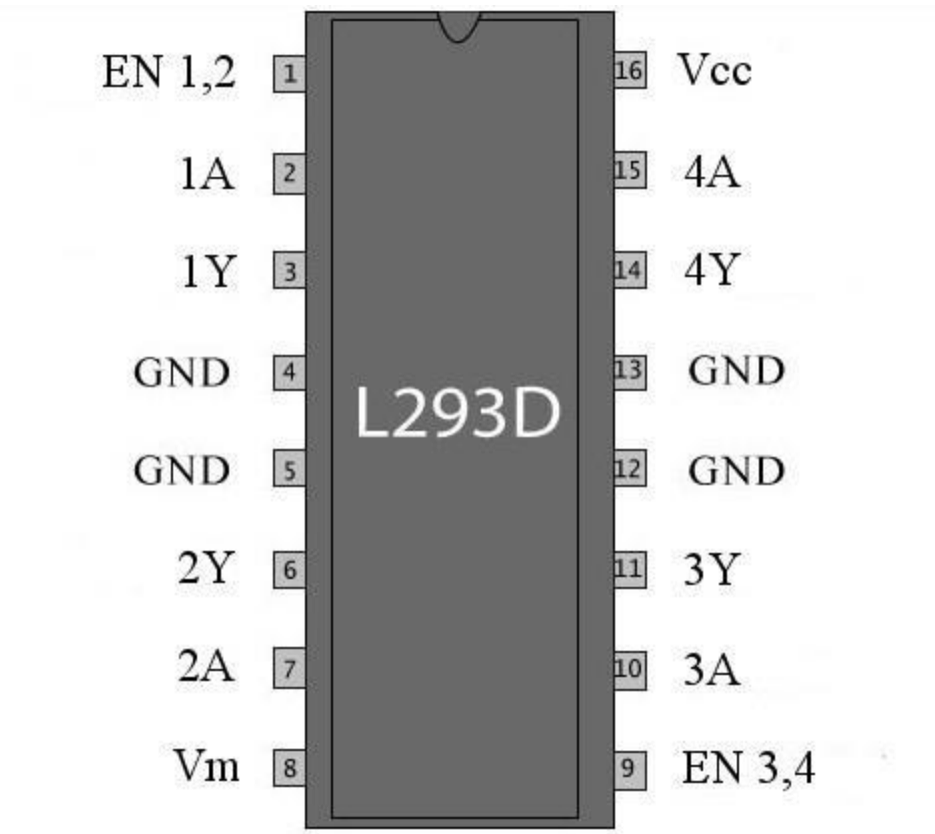

Fig 3.7 Pin Diagram of L293D Driver

Table 3.1: L293D Pin Configuration

Pin Number	Pin Name	Description
1	Enable 1,2	This pin enables the input pin Input 1(2) and Input 2(7)
2	Input 1	Directly controls the Output 1 pin. Controlled by digital circuits
3	Output 1	Connected to one end of Motor 1
4	Ground	Ground pins are connected to ground of circuit (0V)
5	Ground	Ground pins are connected to ground of circuit (0V)
6	Output 2	Connected to another end of Motor 1
7	Input 2	Directly controls the Output 2 pin. Controlled by digital circuits
8	Vcc2 (Vs)	Connected to Voltage pin for running motors (4.5V to 36V)
9	Enable 3,4	This pin enables the input pin Input 3(10) and Input 4(15)
10	Input 3	Directly controls the Output 3 pin. Controlled by digital circuits
11	Output 3	Connected to one end of Motor 2
12	Ground	Ground pins are connected to ground of circuit (0V)
13	Ground	Ground pins are connected to ground of circuit (0V)
14	Output 4	Connected to another end of Motor 2
15	Input 4	Directly controls the Output 4 pin. Controlled by digital circuits
16	Vcc2 (Vss)	Connected to +5V to enable IC function

DC MOTOR:

What is a DC Motor?

A dire revolutionized motor is a type of electric machine that converts electrical energy into mechanical energy. DC motors take electrical power through direct current and convert this energy into mechanical rotation. DC motors use magnetic fields that occur from the electrical currents generated, B0 motor in Fig 3.8 which power the movement of a rotor fixed within the output shaft. The output torque and speed depend upon both the electrical input and depend upon both the electrical input and the design of the motor.

Fig.3.8 BO Motor

How DC motors works:

The term 'DC motor' is used to refer to any rotary electrical machine that converts direct current electrical energy into mechanical energy. DC motors can vary in size and power from small motors in toys and appliances to large mechanisms that power vehicles, pull elevators and hoists, and drive steel rolling mills.

When the coils are turned on and off in sequence, a rotating magnetic field is created that interacts with the different fields of the stationary magnets in the stator to create torque, which causes it to rotate. These key operating principles of DC motors allow them to convert the electrical energy from direct current into mechanical energy through the rotating movement, which can then be used for the propulsion of objects.

3.4 Software Requirements

Arduino IDE:

Arduino IDE where IDE stands for Integrated Development Environment – An official software introduced by Arduino. Cc, which is mainly used for writing, compiling, and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software which is open source and is readily available to install and start compiling the code on the go.

Introduction to Arduino IDE:

Arduino IDE is an open-source software that is mainly used for writing and compiling code into the Arduino Module. It is an official Arduino software, making code compilation too 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, and Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing, and compiling the code in the environment.

A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro, and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded to the controller on the board. The IDE environment in Fig 3.9 mainly contains two basic parts: Editor and Compiler wherethe former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

Fig 3.9 Arduino IDE

One of the advantages of using the Arduino IDE is its versatility. It supports a wide range of Arduino boards and shields, and it can be used on different operating systems such as Windows, Mac OS, and Linux. It also has a large and active community of users who share their knowledge, code, and projects. Overall, the Arduino IDE is a powerful and easy-to-use tool for programming Arduino boards, making it an ideal choice.

CHAPTER 4

Description of the Proposed System

4.1 Methodology of the model:

This proposed project is designed to build control operations of the fire fighting robot. Fire detection, it is using fire sensors. Here we are using three fire sensors. Based on fire sensor detection the servo motor will move. The extinguishing system will be get activated when the fire detection system detects fire. The pump will start pumping water when it detects fire by using a servo motor. The proposed system is an IoT-based fire-fighting and affected area monitoring robot that is designed to detect and extinguish fires while also monitoring the affected area.The system is comprised of a robot equipped with sensors and an Bluetooth module to control manually.

The robot is designed to move around the area affected by the fire and gather data from flame sensors. This data is transmitted to the Arduino, which analyzes it to determine the location of the fire. If a fire is detected, the robot is programmed to activate its fire extinguishing mechanism, which may include a water pump. The robot will continue to monitor area until the fire is completely extinguished.

In addition to fire-fighting capabilities, the robot is also equipped with an Bluetooth module to control manually. This allows the robot to monitor the affected area and provide real-time updates to emergency responders. The system is also designed to be remotely controlled through a web interface, allowing authorized personnel to monitor the robot’s activity and adjust its settings as necessary.

Overall, the IoT-based fire-fighting robo car provides an efficient and effective solution for detecting and extinguishing fires while also controlled manually through Bluetooth.

4.2 Architecture Design of Proposed System:

The IoT-based fire-fighting robo car consist of two main components: hardware and software modules. These components work together to detect and extinguish fires.

Hardware:

The hardware component includes the physical components of the robot, such as sensors, motors, and the microcontroller. The robot is equipped with flame sensors. The robot’s fire extinguishing mechanism may include a water pump, depending on the requirements of the application. To enhance its versatility, the robot's design might allow for modular upgrades, enabling it to accommodate future advancements in sensor technology or firefighting techniques. Additionally, the chassis and structural components could be engineered for durability and heat resistance, ensuring the robot's ability to withstand the harsh conditions encountered during firefighting operations.

Software:

The software component includes the programs that run on the robot’s microcontroller and the web interface used to remotely control. The robot’s microcontroller is programmed to read data from the sensors, analyze it, and activate the fire extinguishing mechanism if necessary. The web interface provides a user-friendly dashboard that displays the robot’s sensor data and moves to the place with help of the sensor. The software architecture is designed with scalability and adaptability in mind, allowing for seamless integration of future software updates and enhancements. This ensures that the robot remains at the forefront of technological advancements in fire detection and extinguishing capabilities.

In this architecture, the robot’s flame sensors are connected to its microcontroller, which processes the data and activates the fire extinguishing mechanism if necessary. The robot is also equipped with a Bluetooth module, which is used to transmit data from the mobile application to control manually.

CHAPTER 5

Implementation Details

5.1 Development and Deployment Setup:

To design and implement an IoT-based firefighting and affected area monitoring robot, thefollowing Development and Deployment Setup can be used:

5.1.1 Hardware components:

- Arduino or similar microcontroller board
- Flame Sensors
- Motor driver and motors
- HC05 Bluetooth Module
- Power supply (battery or adapter)

5.1.2 Software components:

- Programming language (C++)
- IoT platform (such as Arduino IoT Platform)

5.1.3 Development and deployment process:

- Gather hardware components and assemble the robot.
- Write code for controlling the robot's movements, collecting data from the sensors.
- Set up the IoT platform and configure the Arduino to connect to it.
- Deploy the system to the target environment and test its functionality.

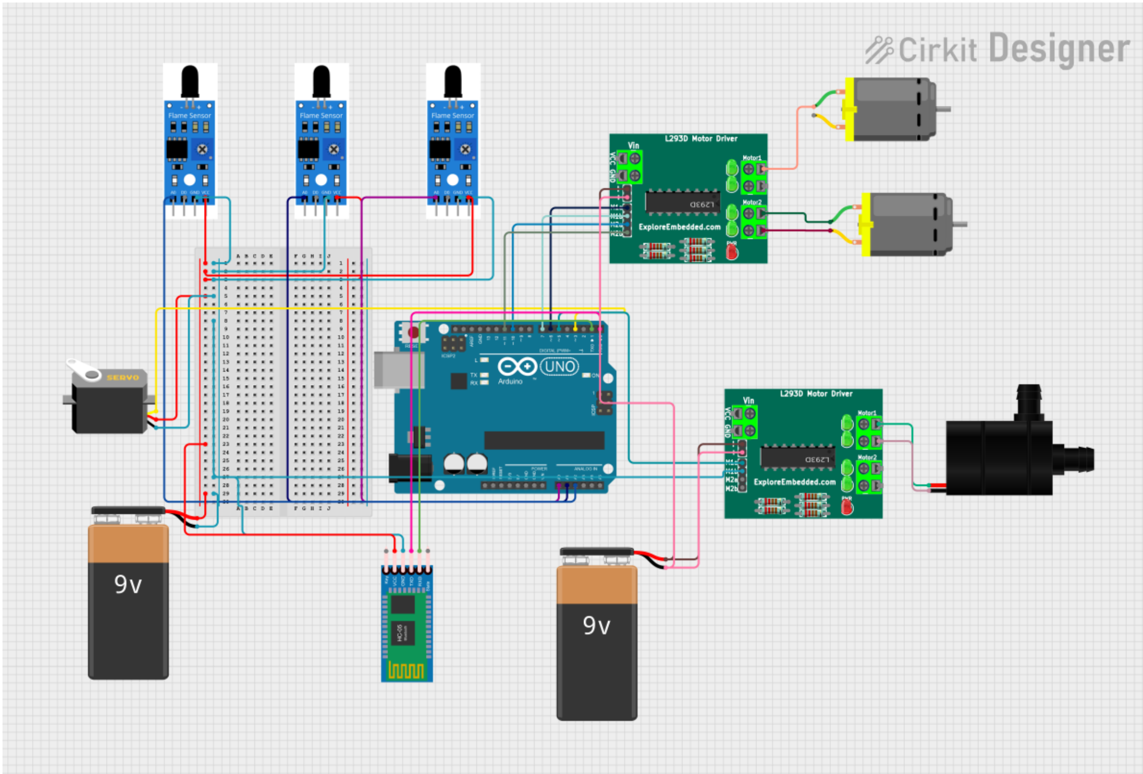

Fig.5.1 Circuit Diagram

5.2 Working:

The fire-fighting robot car operates on a basic yet effective principle, integrating various components to detect and extinguish fires autonomously. At its core are three Flame IR sensors strategically positioned on the robot to sense the presence of fire within their range. These sensors serve as the frontline detectors, promptly relaying signals to the microcontroller, typically an Arduino board, upon detecting flames. Upon receiving signals from the Flame IR sensors, the microcontroller springs into action, orchestrating the response mechanism. It commands a servo motor connected to a water pump, initiating the fire extinguishing process. The servo motor's precise control ensures the accurate direction of the water spray towards the detected fire, maximizing effectiveness in extinguishing flames.

Driving the robot's mobility are two DC motors powering its wheels. The microcontroller coordinates these motors, directing the robot towards the location of the detected fire. This autonomous navigation enables the robot to swiftly reach the source of the fire and deploy its extinguishing mechanism without human intervention. However, the robot's capabilities extend beyond autonomous operation. It also features a Bluetooth module, facilitating wireless communication with external devices like smartphones or computers. This enables manual control of the robot, offering users the flexibility to override autonomous functions and maneuver the robot manually when necessary.

In autonomous mode, the robot continuously scans its surroundings, ready to respond to fire emergencies at a moment's notice. Upon detecting a fire, it swiftly moves into action, deploying its extinguishing mechanism to suppress the flames effectively. Conversely, in manual control mode, users can take charge of the robot's movements via Bluetooth, providing additional flexibility and adaptability in firefighting scenarios where manual intervention or precise navigation is required. By seamlessly integrating these components and functionalities, the fire-fighting robot car stands as a versatile and efficient tool for combating fires, capable of operating autonomously or under manual control, depending on the needs of the situation.

5.3 Automatic Operation:

1. **Detection of Fire:**

The fire will be detected by flame sensors arranged on the chassis board in three directions: left, right, and center. The supply is provided to sensor by the batteries shown in Fig 5.1.

2. **Action of the Motor Driver:**

The operation of the robot in response to fire detection involves a series of coordinated actions driven by signals from the flame sensors. When the left flame sensor detects fire, the Arduino interprets this signal and sends corresponding pulses to the motor drivers. The motor driver connected to the left side motor receives a signal to stop the motor, while the motor driver for the right side motor receives a signal to run. Consequently, the motor on the left side of the chassis board stops rotating, while the motor on the right side propels the robot, causing it to turn and move in the left direction. This directional adjustment enables the robot to navigate towards the detected fire source, following the direction indicated by the left flame sensor. If the middle flame sensor detects fire, the Arduino responds by sending pulses to both motor drivers to activate the motors. Both motors receive signals to run in the forward direction, propelling the robot forward.

The robot continues to move forward until it reaches the location of the detected fire, guided by the signals from the middle flame sensor. When the robot reaches the fire, indicating the successful detection and navigation to the fire source, another motor driver dedicated to the pump is activated. The Arduino sends signals to the pump motor driver to activate the pump mechanism. Simultaneously, the servo motor, also controlled by the Arduino, is engaged and starts rotating according to the pre-programmed angle and direction. As the servo motor rotates, it directs the nozzle attached to the pump, allowing water or extinguishing agent to be sprayed onto the fire. This coordinated action of activating the pump and servo motor facilitates effective fire suppression, covering an arc of approximately 50 to 130 degrees around the robot. By integrating these functionalities, the robot is equipped to detect, navigate towards, and extinguish fires autonomously, ensuring prompt and effective response to fire incidents.

3. **Action of Servo motor:**

Upon reaching the fire, the robot's motor wheels come to a halt, signaling the start of a precise response. The servo motor engages, pivoting according to pre-set angles and directions. Simultaneously, the pump activates, propelling water towards the fire in a controlled arc spanning between 120 to 180 degrees. This synchronized action ensures targeted coverage, swiftly extinguishing the flames and mitigating potential damage.

5.3.1 Implementation of Automatic Operation:

The firefighting robot car is equipped with a versatile operational capability, enabling it to respond effectively to fires detected at different locations. With its array of sensors and responsive mechanisms, the robot can perform three distinct firefighting operations: center fire response, right-side fire response, and left-side fire response.

When a fire is detected at the center by the flame IR sensors onboard the firefighting robot car, it swiftly springs into action to extinguish the flames. The microcontroller, sensing the signal from the central sensor, orchestrates the response mechanism. It commands the servo motor connected to the water pump, which activates to spray water or fire retardant towards the detected fire. The servo motor's precise control ensures that the water spray effectively targets and suppresses the flames at the center as shown in Fig 5.2.

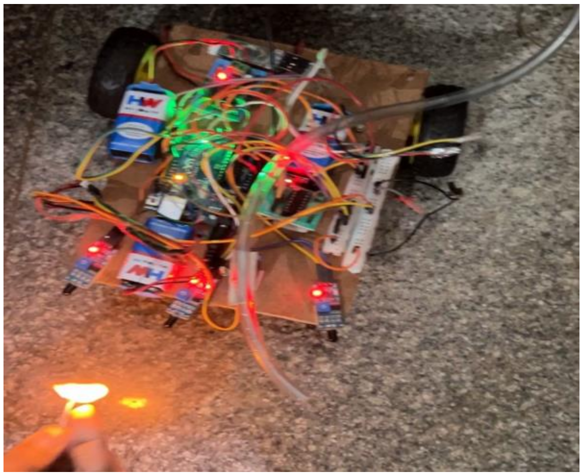

Fig 5.2 Fire detection at center

Similarly, if a fire is detected on the right side of the robot, the microcontroller reacts accordingly. It commands to stop right wheel and starts the left wheel until the fire detected at the center the car changes direction to right side as shown in Fig 5.3. This agile response mechanism enables the robot to effectively extinguish fires detected on its right side, mitigating the spread of the fire and minimizing damage.

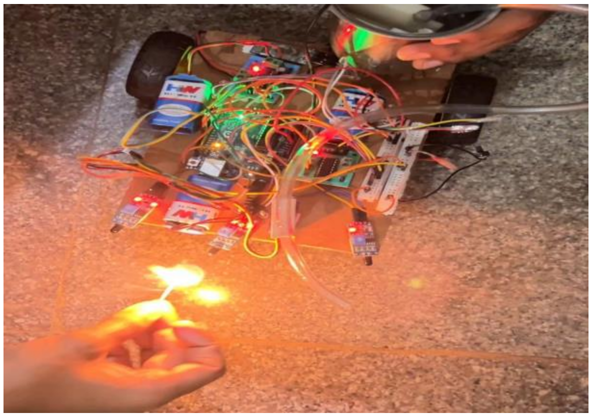

Fig 5.3 Fire detection at right side

When a fire is detected in Fig 5.4 on the left side of the robot, the microcontroller once again coordinates the response. It commands to stop left wheel and starts the right wheel until the fire detected at the center the car changes direction to left side. By swiftly adapting to the location of the detected fire, the firefighting robot car efficiently combats fires regardless of their position relative to the robot.

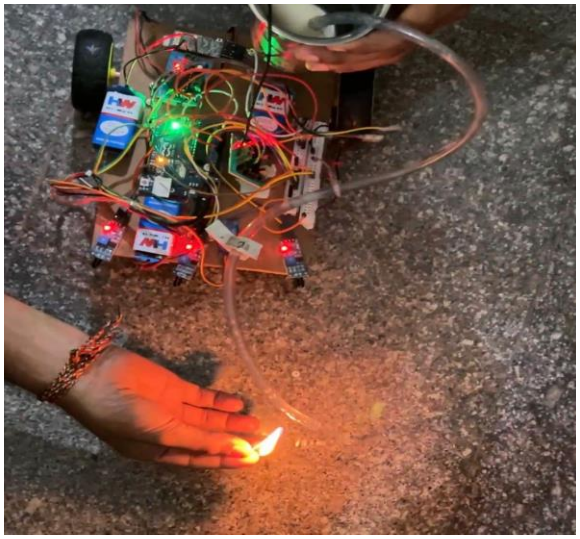

Fig 5.4 Fire detection at left side

5.4 Manual Control:

The Robo Car's manual control feature, accessible through a mobile application, enhances its versatility and adaptability in firefighting scenarios. Users can seamlessly direct the vehicle's movements via a programmed controller, which intuitively responds to button presses. Commands such as "f" for forward, "l" for left, and "r" for right enable users to navigate the car with precision, maneuvering it towards the desired direction. Additionally, the inclusion of commands like "p" for activating the pump and servo mechanisms empowers users to deploy firefighting capabilities remotely. This intuitive interface not only simplifies operation but also allows the car to be directed manually to fires, even when they are situated beyond the range of the onboard sensors.

The mobile application's controller serves as a user-friendly interface as shown in Fig 5.5, offering seamless interaction with the Robo Car's functionalities. By providing a simple yet comprehensive set of commands, users can effectively control the vehicle's movements and firefighting actions with ease. Whether it's navigating through challenging terrain or reaching fires located outside the sensor range, the manual control feature ensures the Robo Car remains agile and responsive in dynamic environments. This level of control not only enhances the effectiveness of firefighting efforts but also underscores the adaptability of the Robo Car in addressing diverse scenarios and challenges.

Moreover, the integration of manual control capabilities extends the Robo Car's utility beyond automated operations. In situations where real-time decision-making is crucial, such as navigating complex environments or responding to rapidly evolving fire conditions, manual control provides an invaluable means of ensuring precise and timely actions. This dual-mode functionality ensures that the Robo Car can seamlessly transition between autonomous and manual operation modes, maximizing its effectiveness in a wide range of firefighting scenarios.

In conclusion, the Robo Car's manual control feature, facilitated by a mobile application, represents a significant advancement in firefighting technology. By empowering users with intuitive and responsive control over the vehicle's movements and firefighting capabilities, this feature enhances operational flexibility, adaptability, and effectiveness. Whether deployed in challenging terrains or remote locations, the Robo Car remains a reliable and versatile asset in combating fires and safeguarding lives and property.

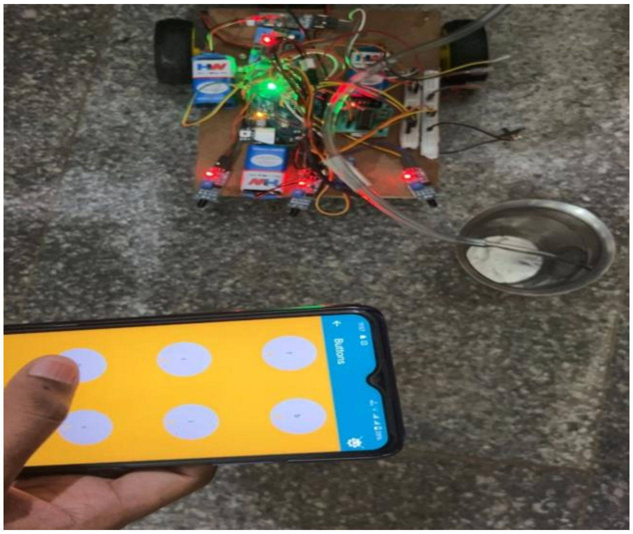

Fig 5.5 Manual control using bluetooth

5.5 Code:

```
#include <Servo.h>
#include <SoftwareSerial.h>
Servo myservo;
int pos = 0;
boolean fire = false;
#define Left A0
#define Right A2
#define Forward A1
#define RM1 6
#define RM2 7
#define LM1 10
#define LM2 11
#define pump 5
char val;
void setup() {
   pinMode(Left, INPUT);
   pinMode(Right, INPUT);
   pinMode(Forward, INPUT);
   pinMode(LM1, OUTPUT);
   pinMode(LM2, OUTPUT);
   pinMode(RM1, OUTPUT);
   pinMode(RM2, OUTPUT);
   myservo.attach(3);
   myservo.write(90);
   Serial.begin(9600);
}
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(30);
    }
    for (pos = 130; pos >= 50; pos -= 1) {
      myservo.write(pos);
      delay(30);
    }
    digitalWrite(pump, LOW);
    myservo.write(90);
    fire = false;
  }
  void loop() {
    if (Serial.available()) {
      while (Serial.available() > 0) {
        val = Serial.read();
        Serial.println(val);
      }
      if (val == 'f')
      {
        Serial.println("forward");
        digitalWrite(LM1, HIGH);
        digitalWrite(LM2, LOW);
        digitalWrite(RM1, HIGH);
        digitalWrite(RM2, LOW);
        delay(1500);
      } else if (val == 'l') {
        Serial.println("ledt");
        digitalWrite(RM1, HIGH);
        digitalWrite(RM2, LOW);
        digitalWrite(LM1, HIGH);
        digitalWrite(LM2, HIGH);
        Serial.println("l completed");
        delay(1000);
```

```
} else if (val == 'r') {
  {
    digitalWrite(RM1, HIGH);
    digitalWrite(RM2, HIGH);
    digitalWrite(LM1, HIGH);
    digitalWrite(LM2, LOW);
    delay(1000);
  }
} else if (val == 's') {
  digitalWrite(RM1, HIGH);
  digitalWrite(RM2, HIGH);
  digitalWrite(LM1, HIGH);
  digitalWrite(LM2, HIGH);
}
else if (val == 'b')
{
  digitalWrite(LM1, LOW);
  digitalWrite(LM2, HIGH);
  digitalWrite(RM1, LOW);
  digitalWrite(RM2, HIGH);
  delay(1000);
} else if (val == 'p') {
  digitalWrite(pump, HIGH);
  delay(500);
  for (pos = 50; pos <= 130; pos += 1) {
    myservo.write(pos);
    delay(30);
  }
  for (pos = 130; pos >= 50; pos -= 1) {
    myservo.write(pos);
    delay(30);
  }
  digitalWrite(pump, LOW);

  myservo.write(90);
```

```
      }
    } else if (!Serial.available()) {
      myservo.write(90);
      if (digitalRead(Left) == 1 && digitalRead(Right) == 1 &&
          digitalRead(Forward) == 1) {
        digitalWrite(LM1, HIGH);
        digitalWrite(LM2, HIGH);
        digitalWrite(RM1, HIGH);
        digitalWrite(RM2, HIGH);
      }
      else if (digitalRead(Forward) == 0) {
        digitalWrite(LM1, HIGH);
        digitalWrite(LM2, LOW);
        digitalWrite(RM1, HIGH);
        digitalWrite(RM2, LOW);
        fire = true;
      }
      else if (digitalRead(Left) == 0) {
        digitalWrite(RM1, HIGH);
        digitalWrite(RM2, LOW);
        digitalWrite(LM1, HIGH);
        digitalWrite(LM2, HIGH);
      }
      else if (digitalRead(Right) == 0) {
        digitalWrite(RM1, HIGH);
        digitalWrite(RM2, HIGH);
        digitalWrite(LM1, HIGH);
        digitalWrite(LM2, LOW);
      }
      delay(500);
      while (fire == true) {
        put_off_fire();
      }
    }
  }
```

5.6 Code Explanation

1. Libraries and Definitions:

The code includes two library imports: Servo.h for servo motor control and SoftwareSerial.h for software-based serial communication. It defines pins for various sensors, motors, and the pump.

2. Global Variables:

Global variables are declared, including a servo object (myservo), a boolean flag for fire detection (fire), and a character variable (val) for storing serial input.

3. Setup Function:

The setup() function configures pin modes, initializes the servo motor to a neutral position, and sets up serial communication at 9600 baud.

4. put_off_fire Function:

This function is responsible for extinguishing fires. It activates the motors and pump, moves the servo to direct the water spray, and then resets the system.

5. Loop Function:

The loop() function continuously checks for serial input. If a command is received, it executes the corresponding action (e.g., moving forward, left, right, stopping, activating the pump). If no serial input is received, it checks sensor inputs for fire detection and triggers the put_off_fire function if a fire is detected.

6. Fire Detection and Response:

The code utilizes sensor inputs to detect fires. When a fire is detected, it sets the fire flag to true and triggers the put_off_fire function.

7. Manual Control:

The code also includes provisions for manual control via Bluetooth or serial communication. Different commands ('f', 'l', 'r', 's', 'b', 'p') correspond to specific actions (forward, left, right, stop, backward, pump activation).

5.7 Algorithm Testing:

Testing the algorithm for the Fire Fighting Robot involves verifying its functionality in various scenarios, ensuring that it accurately detects fires, navigates towards them, and activates the extinguishing mechanism effectively. Here are the testing steps:

1. **Hardware Setup Check:**

Ensure all hardware components are properly connected to the Arduino board according to the wiring diagram. Verify the connections for flame sensors, motor wheels, pump, and servo motor.

2. **Power On:**

Power on the Arduino board and the robot.

3. **Serial Output Verification:**

Open the serial monitor in the Arduino IDE to observe the output messages. Verify that the serial communication is established and the “Fire detected!” message is displayed when a flame is detected.

4. **Flame Detection Testing:**

Introduce flames or heat sources near each flame sensor one at a time. Verify that the corresponding flame sensor readings increase significantly above the flame threshold value. Repeat this test for all flame sensors to ensure they detect fires accurately from different directions.

5. **Movement Testing:**

Test the robot’s movement control by manually triggering each direction (forward, left, right) through the Arduino code. Verify that the robot moves in the expected direction when a fire is detected from a specificsensor.

6. **Navigation Testing:**

Simulate fire detection from different directions using flame sensors. Verify that the robot correctly determines the direction of the fire and navigates towards it.Ensure the robot's movement is smooth and obstacle-free.

7. **Pump Activation Testing**:

Simulate fire detection and verify that the pump activates when a fire is detected. Check that the servo motor adjusts the nozzle direction appropriately towards the detected fire.

8. **Extinguishing Effectiveness Testing:**

Direct the robot towards a controlled fire source. Verify that the pump dispenses water or fire-retardant liquid effectively to extinguish the flames. Ensure that the fire is adequately suppressed, and the robot maintains stability during the extinguishing process.

9. **Autonomous Operation Testing:**

Let the robot operate autonomously in a controlled environment with simulated fires. Verify that it detects and responds to fires promptly without manual intervention. Observe its overall performance, including fire detection accuracy, navigation efficiency, and extinguishing effectiveness.

10. **Safety Testing:**

Ensure that the robot's movements and operations do not pose safety risks to usersor surroundings. Verify that emergency stop mechanisms are effective in halting the robot's operation.

CHAPTER 6

Results

The fire-fighting robot prototype, designed using Arduino microcontroller along with essential components including three flame sensors, an L293D motor driver, a servo motor, a water pump, two motor wheels, and one castor wheel, as shown in Fig 6.1 that demonstrates a functional approach towards autonomous fire detection and suppression. Equipped with three flame sensors positioned strategically for comprehensive coverage, the robot can effectively detect fire hazards within its vicinity. Upon detecting a fire, the robot promptly initiates its firefighting protocol, halting its forward movement and activating the water pump to simulate extinguishing the flames. The servo motor sweeps across a predefined arc, mimicking the action of spraying water to suppress the fire. This automated response mechanism enables the robot to swiftly and autonomously address fire emergencies, reducing the potential for property damage and ensuring prompt intervention in critical situations.

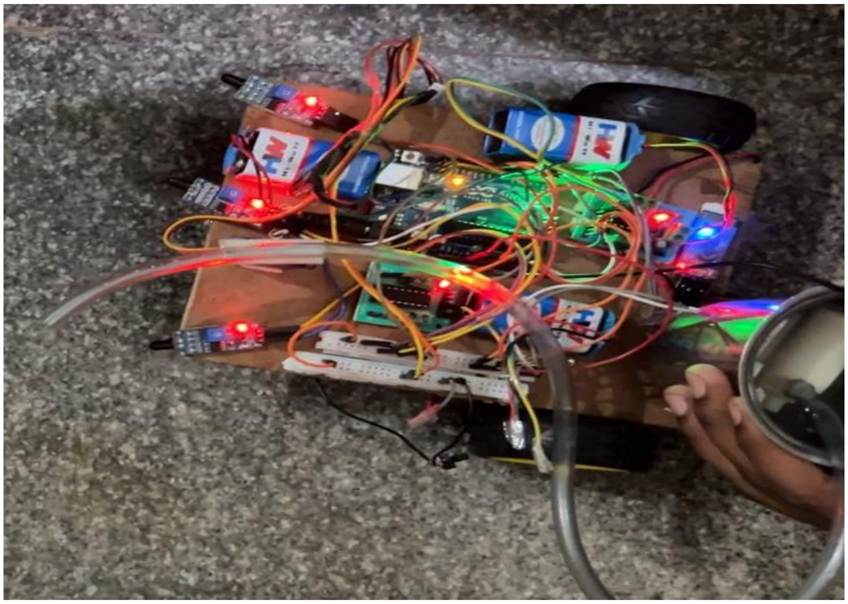

Fig 6.1 Overall working model

Furthermore, the integration of the L293D motor driver facilitates precise control over the movement of the robot, enabling it to navigate through obstacles encountered in its path. With two motor wheels for propulsion and one castor wheel for stability, the robot exhibits agile maneuverability, allowing it to navigate diverse terrains with ease. The modular design of the robot, coupled with the versatility of Arduino programming, provides ample scope for future enhancements and optimizations. Potential improvements may include refining the movement algorithms for smoother operation, implementing advanced firefighting techniques for enhanced efficiency, and incorporating safety features to mitigate risks in dynamic environments. Overall, the fire-fighting robot prototype underscores the potential of embedded systems and robotics in addressing real-world challenges, showcasing a promising avenue for the development of intelligent and responsive autonomous systems.

CHAPTER 7

Conclusion

7.1 Conclusion:

In conclusion, the development of the fire-fighting robot using Arduino, equipped with three flame sensors, L293D motor driver, servo motor, water pump, and a robust wheel setup, represents a significant step forward in autonomous fire detection and suppression technology. Through meticulous integration of hardware components and sophisticated programming, the robot demonstrates a proactive approach towards fire management, capable of swiftly identifying and mitigating fire hazards without human intervention. The successful execution of the robot's firefighting protocol, including the precise activation of the water pump and sweeping motion of the servo motor to simulate water spraying, underscores the effectiveness of the automated response mechanism in addressing emergency situations.

7.2 Future Scope:

Moreover, the versatility and scalability of the Arduino platform offer ample opportunities for further refinement and enhancement of the fire-fighting robot. Future iterations could explore advanced algorithms for more efficient movement control, incorporate machine learning techniques for adaptive response to varying fire scenarios, and integrate sensor fusion technology for enhanced situational awareness. Additionally, considerations for safety and reliability, such as implementing redundant sensor systems and fail-safe mechanisms, will be crucial for the widespread adoption of such autonomous systems in real-world environments. Overall, the fire-fighting robot serves as a testament to the potential of robotics and embedded systems in revolutionizing emergency response mechanisms, paving the way for safer and more resilient communities in the face of fire emergencies.

REFERENCES

[1] K. L. Su, "Automatic Fire Detection System Using Adaptive Fusion Algorithm for Fire Fighting Robot", *2006 IEEE International Conference on Systems*, pp. 966-971, 2006.

[2] H. Amano, "Present status and problems of firefighting robots", *Proceedings of the 41st SICE Annual Conference. SICE 2002*, vol. 2, pp. 880-885, 2002.

[3] T. L. Chien, H. Guo, K. L. Su and S. V. Shiau, "Develop a Multiple Interface Based Fire Fighting Robot", *2007 IEEE International Conference on Mechatronics*, pp. 1-6, 2007.

[4] J. H. Hwang, S. Jun, S. H. Kim, D. Cha, K. Jeon and J. Lee, "Novel fire detection device for robotic fire fighting", *ICCAS 2010 Gyeonggi-do, pp*. 96-100, 2010.

[5] L. Celentano, B. Siciliano and L. Villani, "A robotic system for firefighting in tunnels", *IEEE International Safety Security and Rescue Rototics Workshop 2005*. *Kobe,* pp. 253-258, 2005.

[6] A. Bradshaw, "The UK Security and Fire Fighting Advanced Robot project" in IEEE Colloquium on Advanced Robotic Initiatives in the, UK, London, pp. 1/1-1/4, 1991.

[7] D. J. Pack, R. Avanzato, D. J. Ahlgren and I. M. Verner, "Fire-fighting mobile robotics and interdisciplinary design-comparative perspectives", *IEEE Transactions on Education, vol. 47, no. 3, pp. 369-376, Aug. 2004.*