Smart Multi-Functional Fire Fighting & Gas/Smoke Detective Robot Car

Submitted By

Md Reduan Hossain	ID: EEE2201025064
Sinbad Siddiki Milon	ID: EEE2003021061
Apon Chandra Das	ID: EEE2201025102
Abu Rayhan	ID: EEE2201025103
Hasibul Rahman Khan	ID: EEE2201025080

Supervised By

Md. Ashikur Rahman

Lecturer

Department of EEE

Sonargaon University

Department of Electrical & Electronic Engineering
Sonargaon University (SU)
147/I, Panthapath, Dhaka-1215

Date of Submission: May, 2025

DECLARATION

We hereby, declare that the work presented in this project is the outcome of the investigation performed by us under the supervision of Md. Ashikur Rahman, Lecturer of the Electrical & Electronic Engineering Department, Sonargaon University. We also declare that no part of this project report and thereof has been or in being submitted elsewhere for the award of any degree.

We further undertake to identify the university against any loss or damage arising from breach of the forgoing obligation.

……..
Md Reduan Hossain
ID:EEE2201025064

……
Sinbad Siddiki Milon
ID: EEE2003021061

…….
Apon Chandra Das
ID: EEE2201025102

……..
Abu Rayhan
ID: EEE2201025103

…….
Hasibul Rahman Khan
ID: EEE2201025080

CERTIFICATEION

This is to certify that the project report **"Smart Multi-Funtional Fire Fighting & Gas/Smoke Detective Robot Car"** is the confide record of the work done by-for partial fulfillment of the requirement of their degree of B.Sc. in Electrical and Electronic Engineering at Sonargaon University.

The project report has been carried out under my guidance and is a confide record of the successful work. I wish them success in future.

Supervised By

…….

Md. Ashikur Rahman

Lecturer, Department of EEE

Sonargaon University

ACKNOWLEDGEMENT

Firstly, we express our sincere thanks and gratitude to the almighty for his divine blessings which made it possible for us to complete this project successfully.

We are grateful and express our deepest gratitude to Md. Ashikur Rahman, Lecturer, Department of Electrical and Electronic Engineering, Sonargoan University. The deep knowledge and keen interest of our supervisor influenced me to carry out this project. His infinite patience, scholarly guidance, constant encouragement, constant energetic supervision, constructive criticism, valuable suggestions at all levels made this project possible.

The successful completion of this thesis would not have been possible without his constant inspiration and constant guidance. Hence we are extremely pleased to acknowledge our profound gratitude and respect to the maintainer.

We would like to express our gratitude to our Head of Department, as this project was completed under the Head of Department of Electrical and Electronics Engineering, Sonargaon University, Bangladesh.

We would like to thank our entire course mates from Sonargaon University, who participated in this discussion while completing the course work.

The author is grateful to all the respected faculty members of the Department of Electrical and Electronics Engineering for their cooperation and vital assistance in the successful completion of the thesis work.

ABSTRACT

In recent years, the demand for intelligent and autonomous systems for fire detection and mitigation has grown exponentially due to increasing fire hazards in both residential and industrial environments. The "Multifunctional Fire-Fighting Robot Car" aims to offer a compact, efficient, and cost-effective solution for early fire detection, immediate alerts, and manual intervention via wireless control. This robot integrates multiple technologies including gas and smoke sensors, Bluetooth for manual control, Blynk for real-time LED notification, and GPS based location tracking with automatic SMS alerts in case of hazard detection. The robot is designed to detect harmful gases or smoke using onboard MQ-series sensors and notify the user through the Blynk IoT platform with an LED indicator. Additionally, it allows manual navigation through a Bluetooth-enabled mobile application, providing flexibility in firefighting operations in inaccessible or hazardous areas. One of the key features includes automatic SMS alerts with real-time location coordinates, enabling swift action by alerting authorities or users. This robotic platform is built on a mobile chassis with DC motors, controlled via a microcontroller (e.g., ESP32), and integrated with modules like GPS, GSM, and gas sensors. The hybrid control system ensures both automation and user-driven maneuverability, enhancing its utility in various emergency scenarios. The robot offers a practical blend of IoT and robotics to promote safety and proactive fire management. This project serves as a scalable and upgradable platform for future development in automated disaster management systems. Its multifunctional approach provides a comprehensive response system that could significantly reduce response time and damage during fire emergencies. The implementation of such a robot could be a valuable addition to safety infrastructure in homes, warehouses, factories, and public buildings.

Table of Contents

Page No

DECLARATION....ii

CERTIFICATEION....iii

ACKNOWLEDGEMENT....iv

ABSTRACT....v

TABLE OF CONTENTS....vi

LIST OF FIGURES....viii

CHAPTER 1

INTRODUCTION

1.1 Introduction....1

1.2 Objectives:....2

1.3 Problem Statement:....2

CHAPTER 2

LITERATURE REVIEW

2.1 Existing Systems....3

2.2 Literature Review....4

CHAPTER 3

METHODOLOGY

3.1 Description of Methodology:....6

3.2 Planning:....6

3.3 Requirement Analysis:....7

3.4 Hardware Requirements:....8

3.5 Software Requirements:....14

3.5.1 ESP32 software:....14

3.5.2 Proteus:....15

CHAPTER 4

DESIGN AND DEVELOPMENT

4.1 System Design: 17

4.2 Block Diagram 18

4.3 Flow Chart: 19

4.4 Circuit Diagram: 19

4.5 Working Procedure: 20

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of the project: 21

5.2 Results picture of the project: 22

CHAPTER 6

DISCUSSION & CONCLUSION

6.1 Discussion: 23

6.2 Advantages: 23

6.3 Disadvantages: 24

6.4 Future Scope: 24

6.5 Conclusion: 25

REFERENCES 26

LIST OF FIGURES

Figure No		Page No
Figure 3.1	System Planning Diagram	6
Figure 3.3	List of the Components with Price	7
Figure 3.4.1	ESP32	8
Figure 3.4.2	ESP32 Wi-Fi	10
Figure 3.4.4	Resistor	11
Figure 3.4.5	Relay	12
Figure 3.5.1	Arduino Software IDE working Interface	14
Figure 3.5.2	Proteus Software Interface	16
Figure 4.2	Block Diagram	18
Figure 4.3	Flow Chart Design	19
Figure 4.4	Circuit Diagram of the project	19
Figure 5.1	Normal mode of the project	22
Figure 5.1.1	Running mode of the project	22

CHAPTER 1

INTRODUCTION

1.1 Introduction

Fire-related incidents continue to pose significant threats to human life and property, emphasizing the necessity for rapid detection, communication, and response mechanisms. Traditional firefighting methods are often reactive and require human intervention, which can delay response times and put lives at risk. As a result, the development of autonomous and semi-autonomous robotic systems is gaining momentum in the field of disaster management. The proposed “Multifunctional Fire- Fighting Robot Car” is designed to address these challenges by integrating smart sensors, wireless communication, and control technologies to offer a mobile, multifunctional solution to fire emergencies. This robot is not only capable of detecting smoke and harmful gases in the environment, but it also provides immediate alerts to users via the Blynk IoT platform. Simultaneously, it activates a visual warning using onboard LEDs. For added safety and manual control, the robot can be navigated via a mobile app through Bluetooth, making it especially useful in tight or inaccessible spaces where human presence may be dangerous. The integration of a GPS module ensures accurate real-time location tracking, which is crucial for directing emergency response units. When a smoke or gas threat is detected, the system sends an automated SMS with the robot's exact coordinates to a pre-defined contact. This blend of automation, real-time communication, and remote control represents a significant advancement over conventional safety systems. Moreover, it provides a strong foundation for further enhancements like camera integration, AI-based path planning, and automated fire extinguishing mechanisms. This project aims to serve educational, research, and practical disaster- management purposes by demonstrating how embedded systems and IoT can be effectively combined to address real-world problems. The multifunctional robot presents a promising approach to achieving quicker, safer, and more efficient firefighting interventions in emergency scenarios.

1.2 Objectives:

- To detect the presence of smoke or harmful gases using onboard sensors.
- To send an instant SMS alert with GPS location upon detecting fire or gas leakage.
- To provide a visual alert using an LED through the Blynk IoT platform. To enable manual control of the robot via a Bluetooth-based mobile application.
- To design a compact, mobile robot capable of navigating through hazardous environments.
- To enhance fire safety response systems through automation and wireless technologies.

1.3 Problem Statement:

Natural gases such as Liquefied Petroleum Gas (LPG) are widely used in the whole world. LPG is used in industries for various reasons and it also for cooking in home or hotel. As for now, the use of natural gases instead of petroleum as the alternative fuel for mobile cars also has been increased. Although the procedure of installing

LPG-based system is very tight, we 2 could not give 100% guaranteed that the LPG-system will not having leakage Even though human is a perfect creation of God, they still have certain weakness. Human cannot detect the presence of natural gases as fast as the sensor do. Thus, the use of gas sensing system is hugely needed to give real-time monitoring of the gas system. In certain cases, gas leakage can cause fire that will destroy human property. The large scale of fire also could contribute to serious injury or death. This is due to the fire station got delay information about the fire occurred. Therefore, this project shall be able to resolve the problem stated. This is because this project is able to sense the presence of natural gases as well as fire. Besides that, it is also capable to send out an SMS alert automatically to the owner and also to the nearest fire station.

CHAPTER 2
LITERATURE REVIEW

2.1 Existing Systems

Several fire detection and firefighting systems currently exist, ranging from traditional manual tools to automated sensor-based alarm systems. Common conventional methods include fire extinguishers, sprinkler systems, and smoke detectors installed in residential and commercial buildings. While effective for static monitoring, these systems lack mobility and cannot actively explore fire-prone areas or confined spaces where early fire or gas leakage might occur undetected. They are also reactive in nature, requiring human presence or intervention for operation and often providing limited real-time information. In terms of robotics, some modern firefighting robots have been developed for industrial and military use. For instance, large-scale fire-fighting robots equipped with thermal cameras and fire suppression mechanisms are deployed in high-risk zones like oil refineries or nuclear plants. However, these systems are expensive, bulky, and not feasible for household or small-scale applications. Most are designed for very specific tasks, such as extinguishing large fires or assisting in military-grade rescue missions, making them inaccessible for common users or educational implementations. Smart fire detection systems that use IoT platforms like Blynk have gained popularity for remote monitoring of temperature, smoke, and gas levels. These systems can send real-time alerts to users through mobile apps but are typically fixed in position and cannot navigate autonomously or be redirected to inspect suspicious areas. Bluetooth-controlled robot cars exist as well, often used for educational or prototype development. These mobile platforms can be maneuvered via mobile apps but are not usually integrated with hazard detection or GPS/SMS functionality. Additionally, individual components such as gas sensors or GPS modules are often tested in isolation without combining them into a fully functional multifunctional safety system. Thus, there is a gap in the market for a compact, low-cost robot that combines wireless hazard detection, real-time alerts, manual control, and location tracking. The multifunctional robot car proposed in this project bridges this gap by integrating all these features into a single, scalable system. It offers a valuable solution for early detection and safe intervention in fire-prone or gas- leakage situations.

2.2 Literature Review

Fire detection and prevention systems have evolved significantly over the past decades, with research focusing on increasing accuracy, reducing response time, and improving user accessibility. Numerous scholarly articles and technical implementations have highlighted the need for mobile, automated systems capable of not only detecting hazards but also enabling early intervention. Traditional fixed alarm systems, such as those discussed in “Design and Implementation of an Automatic Fire Detection and Alert System” by A. Rahman et al., emphasize the importance of smoke sensors and temperature monitors in triggering alarms. However, their limitation lies in their static nature and dependency on human presence to respond. These systems do not support dynamic exploration or real-time user control, which are essential in unpredictable environments. Robotics-based firefighting solutions have gained attention in recent years. The paper “Fire Fighting Robot with Multisensor System” (IEEE, 2019) presents a robot with temperature and smoke sensors that navigates autonomously using line-following techniques. While this approach improves mobility, it lacks real-time user feedback and is not integrated with modern IoT platforms or communication tools like GPS and GSM modules. Studies such as “IoT Based Fire Detection System Using Blynk App” have explored the integration of mobile applications with embedded systems. These systems enable users to receive instant alerts on their smartphones via platforms like Blynk. Though this enhances user awareness, they do not provide active solutions to move toward danger zones or track the event location using GPS. A more advanced solution is presented in “Design of a GSM and GPS Based Fire Alert System” (IJEAT, 2020), which sends SMS alerts with location data upon detecting smoke or gas.

This method provides immediate situational awareness but lacks manual or autonomous mobility, limiting its scope to detection only. Bluetooth-based robot cars have long been used in academic projects for exploring remote control methods. In "Bluetooth Controlled Robot Using Android App" (IJRASET, 2020), students implemented a basic car chassis controlled via smartphone. While functional, this system does not incorporate any hazard detection or communication modules. Combining insights from all these studies, it is clear that there is a lack of systems that merge real-time wireless alerts, mobility, user control, and GPS-based location tracking into one compact robot. The present project builds upon these foundations to deliver a holistic safety system that can detect, notify, locate, and

navigate – all in one platform. By leveraging ESP32's capabilities, Blynk IoT, GSM/GPS modules, and Bluetooth control, this multifunctional fire-fighting robot car fills the technological and functional gaps identified in previous literature.

CHAPTER 3

METHODOLOGY

3.1 Description of Methodology:

Creating an idea for Design and construction of Microcontroller multifunctional Firefighting robot car system. And designing a block diagram & circuit diagram to know which components need to construct it.

Collecting the all components and programming for the microcontroller to controlled the system. Setting all components in a PCB board & soldering. Then assembling the whole block in a board and finally run the system & checking.

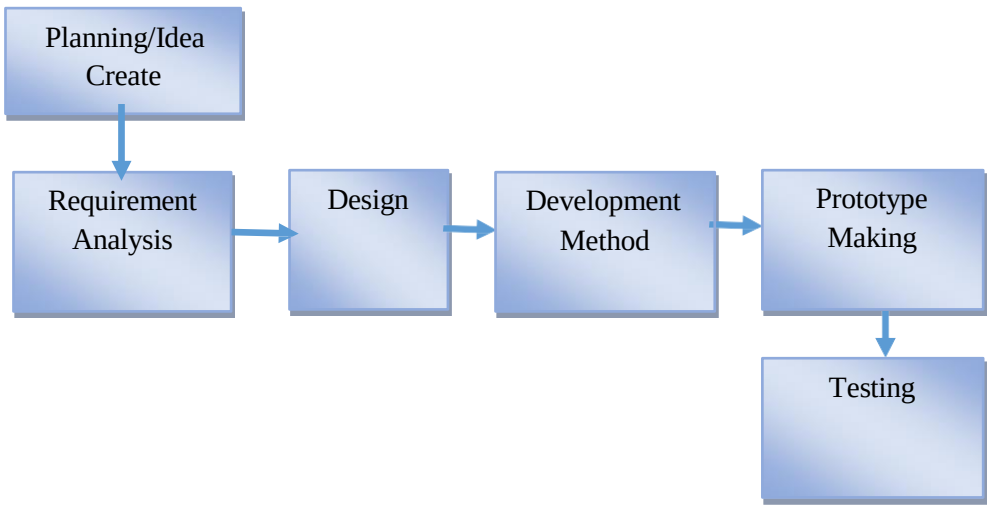


Figure 3.1: System Planning Diagram

3.2 Planning:

The planning phase for this project involved identifying essential features and selecting compatible hardware and software components to achieve a multifunctional design. The first step was to define the project's core objectives: real-time hazard detection, user alert system, manual control, and location tracking. A modular design approach was selected to simplify the integration and troubleshooting of each system—sensors, communication, mobility, and power management.

After initial research, ESP32 was chosen as the main microcontroller due to its dual support for Wi-Fi and Bluetooth. Next, compatible gas/smoke sensors (e.g., MQ-2), a GPS

module (NEO- 6M), and a GSM module (SIM800L) were sourced for environmental monitoring and alert communication. Motor drivers (L298N) and DC motors were selected for mobility. A basic robot chassis with wheels and suspension was used to house all components. The software environment was built using the Arduino IDE for programming and Blynk for IoT integration. Bluetooth app interfaces were developed using MIT App Inventor, allowing real-time user control. The project was divided into four development stages: component testing, system integration, software programming, and final deployment/testing. Time and budget were allocated accordingly, prioritizing critical modules like SMS alerts and Bluetooth control first, followed by GPS and Blynk integration. Contingency plans were prepared in case of component failure or signal reception issues, including manual reset options and modular reconfiguration.

3.3 Requirement Analysis:

Component	**Quantity**	**Unit Price (BDT)**	**Total (BDT)**
ESP32 Microcontroller	1	650	650
MQ-2 Gas/Smoke Sensor	1	180	180
SIM800L GSM Module	1	950	950
NEO-6M GPS Module	1	750	750
L298N Motor Driver	1	220	220
DC Gear Motors	4	200	800
Robot Car Chassis	1	600	600
Li-ion Batteries + Holder	2	300	600
Jumper Wires + Connectors	1 set	100	100
LEDs + Resistors	1 set	50	50
Bluetooth App Development	-	0	0
Misc. (PCB, casing, etc.)	-	300	300
Total Estimated Cost			**5,200 BDT**

Table 3.3: List of the Components with Price

3.4 Hardware Requirements:

Among the many raw materials, we have used to make this pack are ESP32, a series of circuit boards with a series of ESP32 microcontrollers. We have used some more electronics components which we are highlighting below step by

1. ESP32
2. Wi-Fi Module
3. MQ2
4. Relay Module
5. Motor
6. Wheel
7. GSM
8. LED, Buzzer
9. PVB, etc.

ESP32



Figure 3.4.1: ESP32

ESP32 is the world's leading open-source hardware and software ecosystem. The Company offers a range of software tools, hardware platforms and documentation enabling almost anybody to be creative with technology.

ESP32 is a popular tool for IoT product development as well as one of the most successful tools for STEM/STEAM education. Hundreds of thousands of designers, engineers, students, developers and makers around the world are using ESP32 to innovate in music, games, toys, smart homes, farming, autonomous vehicles, and more.

Originally started as a research project by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mallis at the Interaction Design Institute of Ivrea in the early 2000s, it builds upon the Processing project, a language for learning how to code within the context of the visual arts developed by Casey Reas and Ben Fry as well as a thesis project by Hernando Barragan about the Wiring board.

The first ESP32 board was introduced in 2005 to help design students — who had no previous experience in electronics or microcontroller programming — to create working prototypes connecting the physical world to the digital world. Since then it has become the most popular electronics prototyping tool used by engineers and even large corporations.

ESP32 is the first widespread Open Source Hardware project and was set up to build a community that could help spread the use of the tool and benefit from contributions from hundreds of people who helped debug the code, write examples, create tutorials, supports other users on the forums and build thousands of groups around the globe. We are eternally grateful for being supported by such an amazing community.

Since the ESP32 project's foundation, many new development boards and software libraries have been introduced, expanding the range of possibilities available to the community. Today, more than a decade later, ESP32 continues to provide open source hardware and software to bring new ideas to life.

The openness and ease-of-use of the project has led to mass adoption of micro-controller based electronics projects and was a catalyst in the creation of the Maker Movement. ESP32 has become the number one choice for electronics makers, especially for developing solutions for the IoT marketplace, which has been predicted to become a $6 trillion market by 2021.

ESP32 Wifi Module:



Figure 3.4.2: ESP32 Wi-Fi

- ESP32 embedded, Xtensa® single-core 32-bit LX6 microprocessor, up to 160 MHz +19.5 dBm output at the antenna ensures a good physical range
- Sleep current is less than 20 μA, making it suitable for battery-powered and wearable- electronics applications
- Peripherals include UART, GPIO, I2C, I2S, SDIO, PWM, ADC and SPI
- Fully certified with integrated antenna and software stacks
- **The ESP32** series, or family, of Wi-Fi chips is produced by Espressif Systems, a fabless semiconductor company operating out of Shanghai, China. The ESP32 series presently includes the ESP32EX and ESP8285 chips.
- **ESP32EX** (simply referred to as ESP32) is a system-on-chip (SoC) which integrates a 32-bit Tensilica microcontroller, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules into a small package. It provides capabilities for 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2), general-purpose input/output (16 GPIO), Inter-Integrated Circuit (I^2C), analog-to-digital conversion (10-bit ADC), Serial Peripheral Interface (SPI), I^2S interfaces with DMA (sharing pins with GPIO), UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and pulse-width modulation (PWM). The processor core, called "L106" by Espressif, is based on Tensilica's Diamond Standard 106Micro 32-bit processor controller core and runs at 80 MHz (or overclocked to 160 MHz). It has a 64 KiB boot ROM, 32 KiB instruction RAM

- And 80 KiB user data RAM. (Also, 32 KiB instruction cache RAM and 16 KiB ETS system data RAM.) External flash memory can be accessed through SPI. The silicon chip itself is housed within a 5 mm × 5 mm Quad Flat No-Leads package with 33 connection pads — 8 pads along each side and one large thermal/ground pad in the center.
- ESP32 based development boards/modules often incorporate a surface-mount PCB module, a on-board USB-to-serial bridge, and breakout to 0.1 inch pitch connections. For example, the NodeMCU Development Kits use Ai-Thinker modules, the Adafruit feather huzzah uses an Ai-Thinker ESP-12S module with a SiLabs CP2104 USB-to-serial bridge chip, and the weme1 version 2.3 uses an Ai-Thinker ESP-12S module with a WinChipHead CH340G USB-to-serial bridge chip. Other development boards don't use an intermediary module and instead directly incorporate the chip itself on-board — for example, WEMOS D1 Mini Pro uses ESP32EX and WEMOS D1 Mini Lite uses ESP8285
- Anywhere we will use it for tracking the geo location by helping of unwirelabs.com

Resistor:



Fig. 3.4.4: Resistor

Theory of operation

The behavior of an ideal resistor is dictated by the relationship specified by Ohm ‘slaw:

V = I.R

Ohm's law states that the voltage (V) across a resistor is proportional to the current(I), where the constant of proportionality is the resistance (R).

Equivalently, Ohm's law can be stated:

I = V/R

This formulation states that the current (I) is proportional to the voltage (V) and inversely

proportional to the resistance (R). This is directly used in practical computations. For example, if a 300-ohm resistor is attached across the terminals of a12 volt battery, then a current of12 / 300 =0.04 amperes flows through that resistor.

Relay module:

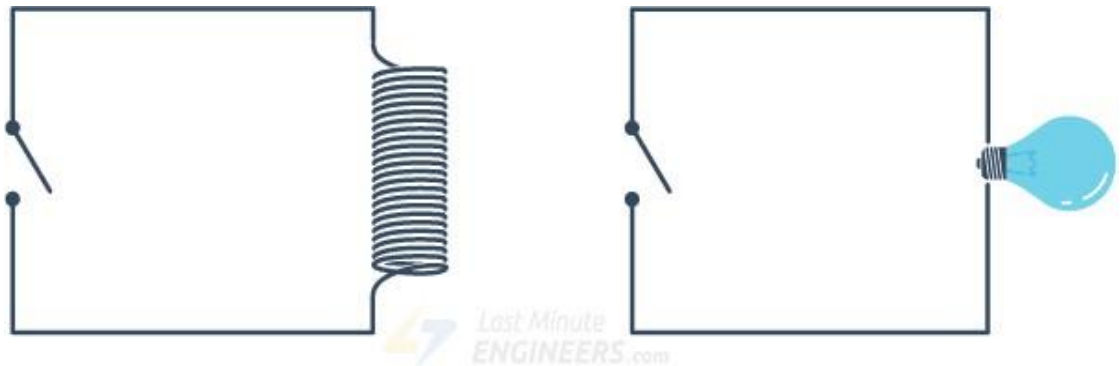


Fig. 3.4.5: Relay

At the core of a relay is an electromagnet (a wire coil that becomes a temporary magnet when electricity is passed through it). A relay can be thought of as an electric lever; you turn it on with a relatively small current, and it turns on another device with a much larger current.

Relay Basics

Here's a small animation showing how a relay links two circuits together.



To illustrate, think about two simple circuits: one with an electromagnet and a switch or sensor, and the other with a magnetic switch and a light bulb.

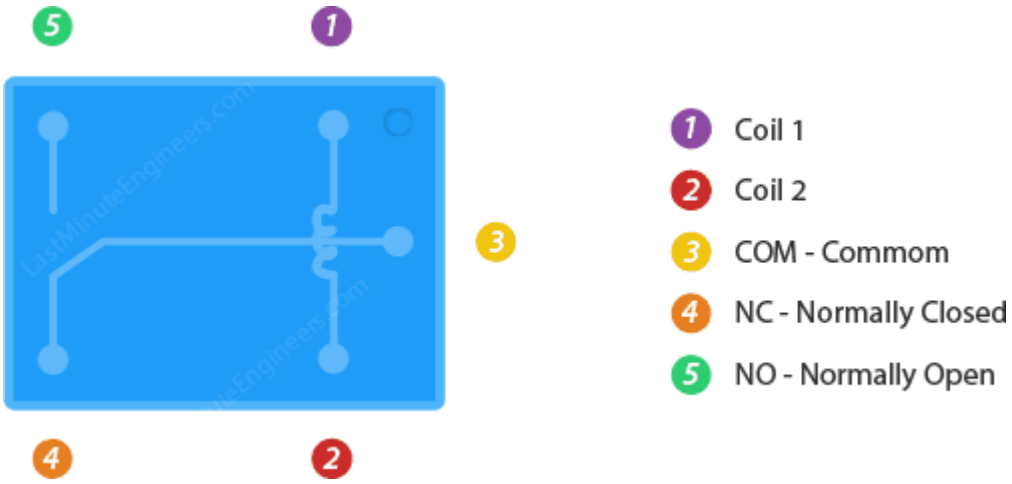
Initially, both circuits are open, with no current flowing through them.

When a small current flows through the first circuit, the electromagnet is energized, creating a magnetic field around it. The energized electromagnet attracts the second circuit's contact, closing the switch and allowing a large current to flow.

When the current in the first circuit stops flowing, the contact returns to its original position, reopening the second circuit.

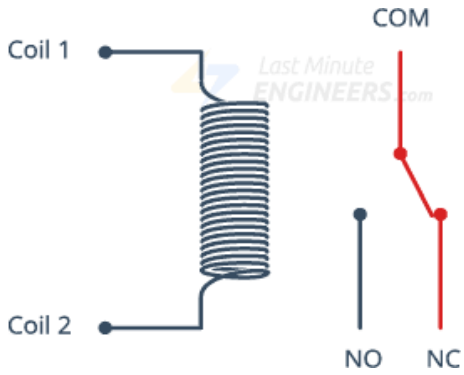
Relay Operation

A relay typically has five pins, three of which are high voltage terminals (NC, COM, and NO) that connect to the device being controlled.



The device is connected between the COM (common) terminal and either the NC (normally closed) or NO (normally open) terminal, depending on whether the device should remain normally on or off.

Between the remaining two pins (coil1 and coil2) is a coil that acts as an electromagnet.



Normally (initial position), the COM terminal is connected to the NC terminal and the NO terminal is open.

3.5 Software Requirements:

1. Arduino Software IDE.
2. Proteus

3.5.1 ESP32 software:

The smart microcontroller unit named as ESP32 Uno can be programmed with the ESP32 software. There is no any requirement for installing other software rather than ESP32. Firstly, Select "ESP32 Uno from the Tools , Board menu (according to the microcontroller on your board).The IC used named as ATmega328 on the ESP32 Uno comes pre burned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer.

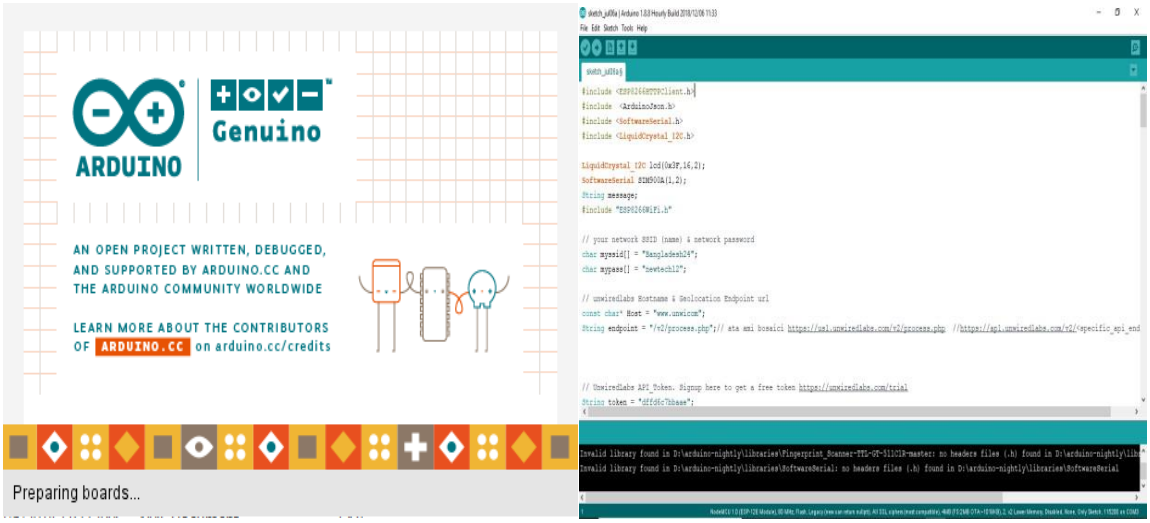


Figure 3.5.1: Arduino Software IDE working Interface

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

The ESP32 Uno is one of the latest smart microcontroller units and has a number of

facilities for communicating with a computer, another ESP32, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .inf file is required. The ESP32 software includes a serial monitor which allows simple textual data to be sent to and from the ESP32 board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial Communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The ESP32 software includes a Wire library to simplify use of the I2C bus. ESP32 programs are written in C or C++ and the program code written for ESP32 is called sketch. The ESP32 IDE uses the GNU tool chain and AVR Lab to compile programs, and for uploading the programs it uses argued. As the ESP32 platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the ESP32.

3.5.2 Proteus:

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. Schematic Capture support followed in 1990, with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based auto routing was added in 2002 and 2006 saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Support for high speed design was added in 2017.Feature led product releases are typically biannual, while maintenance based service packs are released as required.

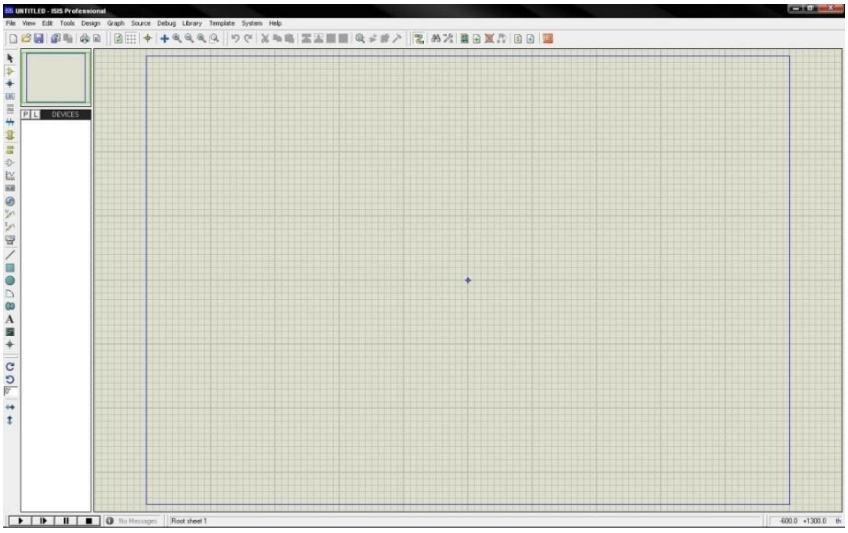


Figure 3.5.2: Proteus Software Interface

Microcontroller Simulation:

The micro-controller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables its use in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design. It also finds use in the general hobbyist community and, since no hardware is required, is convenient to use as a training or teaching tool. Support is available for co-simulation of:

- Microchip Technologies PIC10, PIC12, PIC16, PIC18, PIC24, dsPIC33 Microcontrollers.
- Atmel AVR (and ESP32), 8051 and ARM Cortex-M3 Microcontrollers
- NXP 8051, ARM7, ARM Cortex-M0 and ARM Cortex-M3 Microcontrollers.
- Texas Instruments MSP430, PICCOLO DSP and ARM Cortex-M3 Microcontrollers.
- Parallax Basic Stamp, Free scale HC11, 8086 Microcontrollers.

CHAPTER 4

DESIGN AND DEVELOPMENT

4.1 System Design:

The system design of the Multifunctional Fire-Fighting Robot Car focuses on integrating multiple modules to perform real-time detection, notification, location tracking, and remote control functionalities. The overall design is centered around a microcontroller, primarily the ESP32, chosen for its dual capabilities of Wi-Fi and Bluetooth communication, adequate GPIO support, and processing power to handle multiple simultaneous tasks. The system is composed of several interconnected sub-systems, each serving a specific purpose but collectively functioning as a single, cohesive robotic unit. The power subsystem consists of a rechargeable Li-ion battery pack with a voltage regulator circuit to supply a stable 5V/3.3V output to sensors and modules. Power distribution is carefully planned to prevent voltage drops during simultaneous sensor activation, motor operation, and communication tasks. The locomotion subsystem includes a DC motor-based 4-wheel robotic chassis, driven by an L298N motor driver module. This module interfaces with the ESP32 and allows bidirectional motor control using logic signals.

The robot can move forward, backward, left, or right, and the motion commands are relayed via Bluetooth from a mobile app. The motor driver receives these commands and applies appropriate voltages to the motors for directional control. For environmental monitoring, the system uses gas and smoke sensors such as the MQ-2 or MQ-135, which detect hazardous gas concentrations or smoke particles in the surrounding air. These sensors output analog values, which are read by the ESP32's ADC pins. The readings are continuously monitored in the firmware, and when the threshold limit is exceeded, an alert is triggered. Once a hazard is detected, the notification subsystem comes into action. The ESP32 sends a signal to the Blynk IoT platform, which updates the LED indicator status in the user's smartphone app. Simultaneously, the robot accesses its current GPS location using the NEO-6M GPS module. The latitude and longitude data are extracted via serial communication, parsed, and formatted into a message.

The communication subsystem includes a SIM800L GSM module, used to send an SMS containing the location details to a predefined mobile number. The ESP32 communicates with this module via AT commands over serial interface. This subsystem is critical for emergency alerts, especially when users are not actively monitoring the Blynk app. For manual control, the ESP32 receives directional commands through its built-in Bluetooth module. A custom mobile app developed using MIT App Inventor or similar platforms includes buttons for movement commands, which are transmitted as characters over Bluetooth. The ESP32 interprets these characters and controls the motors accordingly. All components are physically assembled onto a robot chassis with a focus on stability and balance. Sensors are placed at the front of the robot to ensure early detection of hazards. The GPS and GSM modules are mounted with sufficient clearance to avoid signal interference. A compact breadboard or custom PCB is used for tidy wiring and stable connections. In summary, the system design is modular, with each functional block handling specific tasks and interfacing seamlessly with others through the ESP32. The design ensures responsiveness, reliability, and expandability, making it suitable for real-world fire detection and emergency response scenarios.

4.2 Block Diagram

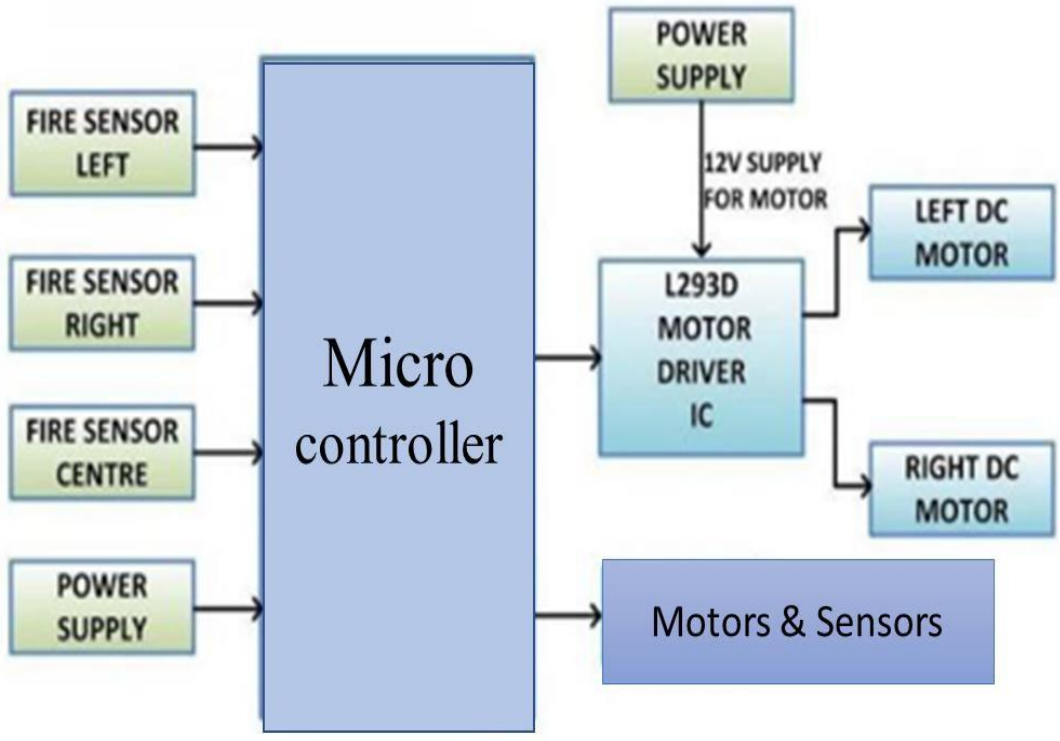


Figure 4.2: Block Diagram

4.3 Flow Chart:

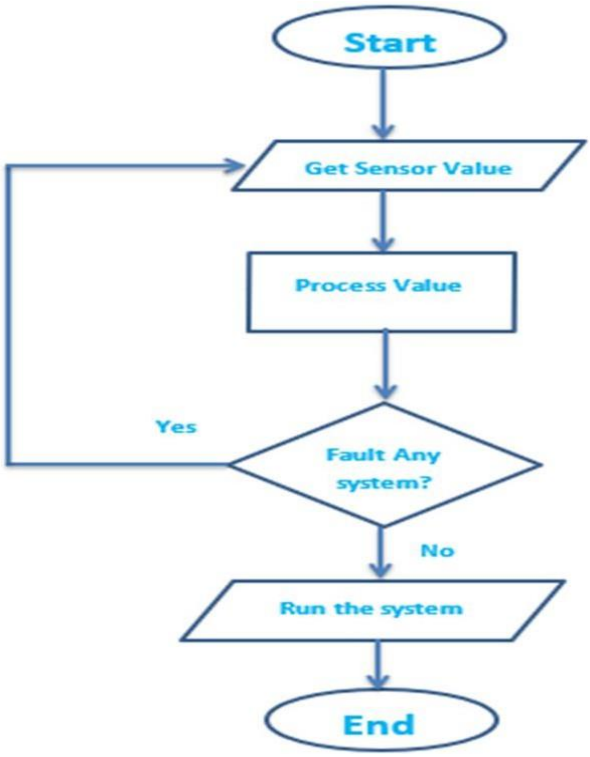


Figure 4.3: Flow Chart Design

4.4 Circuit Diagram:

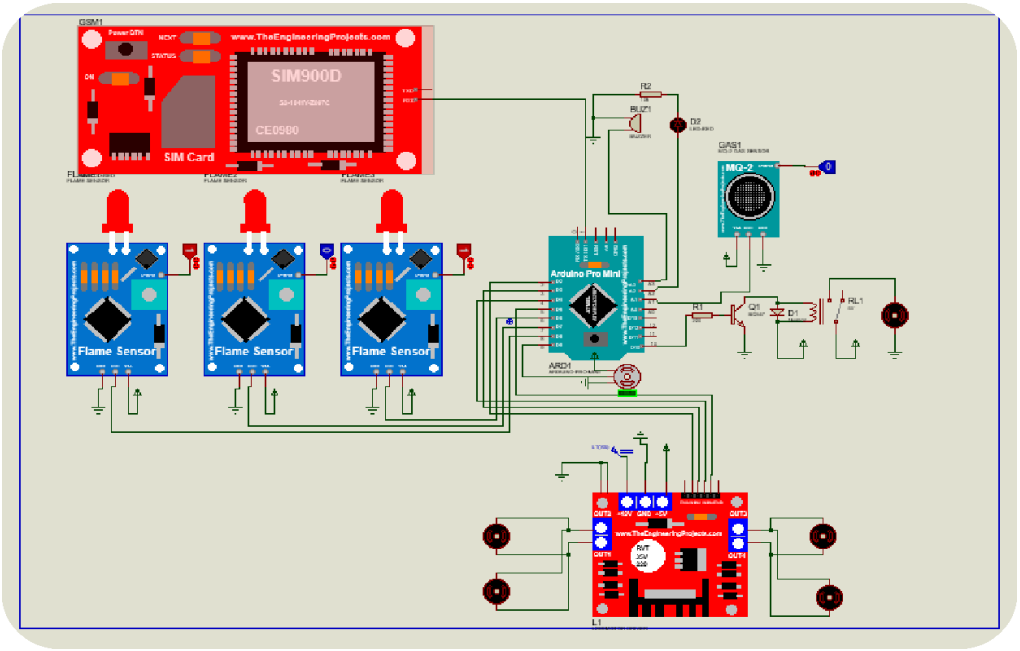


Figure 4.4: Circuit Diagram of the project

4.5 Working Procedure:

The Multifunctional Fire-Fighting Robot Car operates by continuously monitoring environmental conditions using onboard gas and smoke sensors. The sensors are connected to the ESP32 microcontroller, which acts as the brain of the system. The working cycle begins as the robot powers on and starts scanning for smoke or gas concentrations in its immediate surroundings. The MQ-series sensors output analog or digital signals proportional to the detected gas levels. These values are read and processed by the ESP32, which compares them against predefined threshold levels stored in its memory. If the sensor readings exceed the set threshold, the robot performs a sequence of actions simultaneously. First, it activates an alert on the Blynk app, triggering an LED notification on the user's smartphone. This allows the user to be instantly aware of the threat, regardless of location. Second, the ESP32 communicates with a GPS module (like NEO-6M) to fetch the robot's current coordinates. These coordinates are then passed to a GSM module (SIM800L), which constructs and sends an SMS to a pre-configured phone number. This SMS contains the message about the gas or smoke detection along with real-time latitude and longitude data. In addition to its autonomous functions, the robot includes a manual control system.

Using Bluetooth (either through the ESP32 or an external HC-05 module), the robot receives directional commands from a mobile app. The motor driver (L298N) interprets these commands and drives the DC motors accordingly, enabling the robot to move forward, backward, and turn left or right. The robot can be directed toward the source of fire or gas leakage by the user through Bluetooth-based control while continuously updating its status. The power system uses Li-ion batteries to drive all electronics and motors, with a voltage regulator ensuring each component receives stable power. By combining real-time detection, wireless communication, manual mobility, and location tracking, the robot efficiently serves its firefighting assistance role. Its core principle is simple yet powerful: detect danger, notify the user, and assist with safe remote intervention.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of the project:

The implementation of the Multifunctional Fire-Fighting Robot Car was carried out in distinct phases, beginning with the hardware setup, followed by software programming, module integration, and testing. The project was built around the ESP32 microcontroller, which served as the central controller for managing inputs from sensors, controlling the robot's movement, and handling communication tasks. Initially, the robot chassis was assembled with four DC motors connected to the L298N motor driver, which was wired to the ESP32 for receiving control signals. The motor driver was powered using a Li-ion battery pack, and a voltage regulator was added to supply 5V to the other modules. The MQ-2 smoke and gas sensor was connected to the analog input of the ESP32, with its digital pin also used for simple threshold-based detection. The NEO-6M GPS module was connected via serial communication to fetch real-time location data. Similarly, the SIM800L GSM module was wired to another serial port to send SMS alerts. Special attention was given to power supply stabilization for these modules, especially the GSM module, which requires a stable 4V and enough current during transmission bursts. The ESP32 was programmed using the Arduino IDE. The firmware included libraries for Blynk, Software Serial (if needed), GPS parsing, and AT command handling. The logic was structured to continuously monitor sensor input and trigger appropriate actions Blynk LED alert, GPS fetch, and SMS send—when gas or smoke was detected. A Bluetooth mobile app was created using MIT App Inventor, which included buttons for directional control. Each button sent a unique character over Bluetooth to the ESP32, which used Serial. read() functions to interpret commands and actuate the motors. Once hardware and software components were tested independently, the system was integrated and tested in various conditions. Threshold levels for gas/smoke were calibrated by observing sensor readings under controlled exposure. GPS and GSM functionalities were validated by checking the received SMS for correct location data. The robot was driven manually using the app to test movement and responsiveness.

The Blynk platform was connected using Wi-Fi credentials hardcoded into the firmware, and an LED widget on the app was set to light up when the alert flag was triggered. Through systematic implementation and rigorous testing, the multifunctional robot car was

successfully developed and demonstrated as an efficient fire detection and notification system with user-guided mobility.

5.2 Results picture of the project:

Off Mode (Short Description):

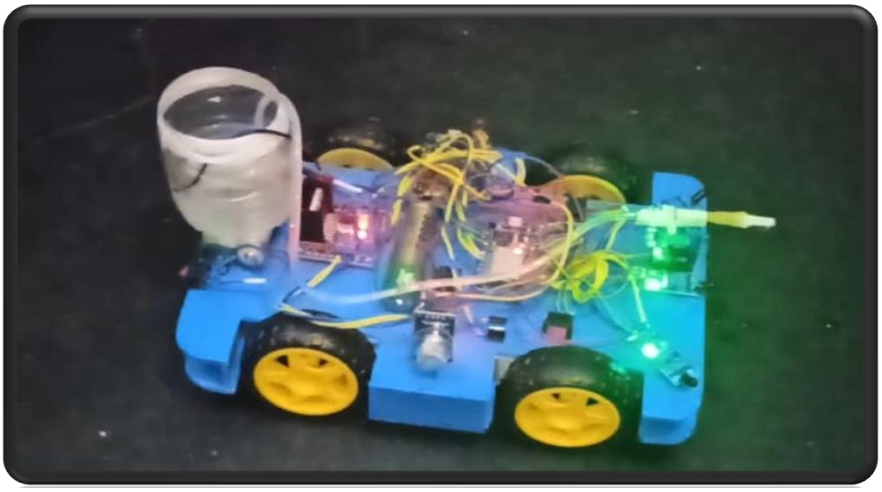


Figure 5.1: Normal mode of the project

In off mode, no fire or gas is detected. The robot remains idle, continuously monitoring the environment. Motors and the water pump stay off, conserving power and resources. Status updates confirm normal conditions.

Run Mode (Short Description):

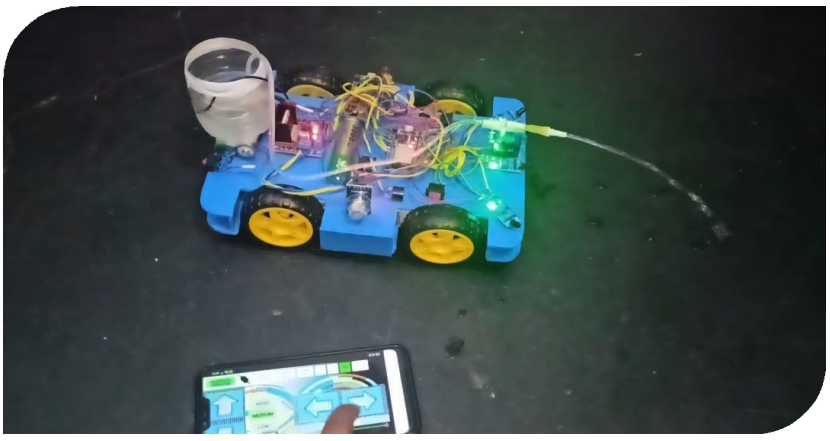


Figure 5.1.1: Running mode of the project

In run mode, the robot detects fire or gas and responds automatically. It activates the water pump, moves toward the fire, sends the GPS location via SMS, and displays the current status of sensors and actions taken.

CHAPTER 6

DISCUSSION & CONCLUSION

6.1 Discussion:

The multifunctional fire-fighting robot car integrates real-time hazard detection, alert communication, and manual control in a single mobile platform, offering a practical and cost- effective solution for early fire response. The project demonstrates the potential of combining low-cost microcontrollers, IoT platforms, and mobile apps to create a semi-autonomous robotic assistant capable of performing in emergency scenarios. The use of the ESP32 proved advantageous due to its built-in Wi-Fi and Bluetooth support, simplifying the architecture and reducing the need for external modules. One of the most notable outcomes was the successful implementation of simultaneous detection and alert systems. The robot could monitor for smoke/gas in real-time and respond immediately by sending an SMS with GPS location. This ensures that even when the user is not actively monitoring the Blynk app, they receive critical alerts that can prompt a quick response. However, some limitations became evident during the testing phase. For example, GSM communication was occasionally delayed due to poor signal reception, and the GPS module had difficulty acquiring a signal indoors. Moreover, Bluetooth control had limited range, requiring the user to be physically close to the robot. Despite these challenges, the robot fulfilled its core objectives. Its modular design allows easy upgrades, and its hybrid control system makes it suitable for use in places where human access is restricted or risky. This discussion confirms the project's relevance in today's safety-focused technological landscape, especially for environments like warehouses, homes, or factories that are at risk of fire hazards.

6.2 Advantages:

- Enables early detection of fire and gas leaks.
- Sends real-time location alerts via SMS.
- Remote manual control enhances safety in dangerous areas.
- Provides visual alerts via Blynk IoT.
- Reduces human risk in fire-prone environments.

6.3 Disadvantages:

- Limited battery life may restrict operation time.
- GSM and GPS modules may fail in signal-deprived environments.
- Manual control may lag due to Bluetooth connectivity issues.
- Not equipped with an automatic fire extinguishing system.
- Requires constant maintenance and calibration for sensors.

6.4 Future Scope:

The multifunctional fire-fighting robot car holds considerable potential for future enhancements. One of the most promising upgrades would be the integration of an automatic fire-extinguishing mechanism using a flame-detection module and servo-controlled nozzle. This would allow the robot not only to detect fires but also to actively suppress them, making it far more effective in real-time firefighting scenarios. In addition, the inclusion of obstacle detection sensors like ultrasonic or infrared modules, and AI-powered pathfinding algorithms would improve its navigation and adaptability in unpredictable or complex environments. Replacing Bluetooth with Wi-Fi or long-range RF communication could enhance remote control capability, especially in larger or more hazardous zones. Camera integration with real-time video streaming could offer users live visual feedback, enhancing manual control and situational awareness. Power systems could be improved by using solar charging or higher-capacity batteries to extend operational time. Cloud integration for data logging, alert history, and real-time monitoring across multiple devices could expand the usability for large buildings or industrial plants. These improvements would transform the current model into a fully autonomous firefighting assistant suitable for broader applications in smart cities, factories, and rescue missions.

6.5 Conclusion:

The development of a Multifunctional Fire-Fighting Robot Car represents a meaningful step toward enhancing fire safety and emergency response mechanisms through the integration of embedded systems and IoT technologies. This project successfully demonstrates how real-time monitoring, remote alert systems, and manual intervention can be harmonized into a single mobile platform to tackle fire hazards. By employing gas and smoke sensors, GPS and GSM modules, and Bluetooth control, the robot offers a comprehensive solution for early hazard detection, location reporting, and user-guided navigation in hazardous environments. The Blynk platform enables real-time visual alerts, while the SMS functionality ensures that emergency contacts are promptly informed with accurate location data. Its manual control option allows users to direct the robot in dangerous areas, thus minimizing human exposure to risk. Despite its limitations, such as the lack of autonomous fire suppression or short-range Bluetooth communication, the robot serves as a reliable prototype for early detection and response applications. The project underscores the potential of integrating multiple technologies in robotics to address real-world problems. With further enhancements, including obstacle avoidance, live camera feeds, and fire extinguishing capabilities, the robot could evolve into a fully autonomous system, valuable for use in homes, warehouses, offices, and industrial facilities. Overall, this multifunctional robot not only contributes to technological innovation but also supports public safety by facilitating faster and smarter responses to fire emergencies. It is a scalable and educational model that paves the way for future developments in smart disaster management systems.

REFERENCES

1. MQ-2 Gas Sensor Datasheet. Hanwei Electronics.

   https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-2.pdf

2. ESP32 Technical Reference Manual. Espressif Systems.

   https://www.espressif.com/en/products/socs/esp32

3. GSM SIM800L Module Datasheet. SIMCom Wireless Solutions.

   https://cdn.sparkfun.com/datasheets/Cellular/General/SIM800 Series_AT Command Manual_V1.09.pdf

4. NEO-6M GPS Module Datasheet. u-blox AG.

   https://www.u-blox.com/en/product/neo-6- series

5. L298N Dual H-Bridge Motor Driver Module Datasheet. STMicroelectronics. https://www.st.com/en/motor-drivers/l298.html

6. Blynk IoT Platform Documentation. https://docs.blynk.io

7. HC-05 Bluetooth Module Datasheet. https://components101.com/wireless/hc-05-bluetooth-module

8. Rajalakshmi, P., & Mahalakshmi, D. (2016). IoT Based Smart Home System Technologies and Applications. International Journal of Computer Applications, 179(7), 38–43.

9. Kumar, K., & Verma, R. (2018). Development of a Smart Fire Detection and Notification System Using IoT. International Journal of Engineering and Technology, 7(3), 45–50.

10. Chen, C., et al. (2015). Mobile Robot for Fire Detection using Flame Sensors and Embedded System. IEEE Transactions on Instrumentation and Measurement, 64(10), 2714–2721.

11. Sharma, A., & Singh, R. (2021). Design and Implementation of a Smart Fire Fighting Robot Using Arduino. International Journal of Innovative Research in Computer and Communication Engineering, 9(4), 3200–3206.

12. Mittal, A., & Jain, S. (2019). GSM-Based Fire Alert System for Smart Buildings. International Research Journal of Engineering and Technology (IRJET), 6(2), 150–154.

13. Subitha, G., & Suganya, R. (2020). Design of IoT-based Fire Detection and Alerting System. International Journal of Scientific & Technology Research, 9(1), 110–114.

14. Khandoker, M. A. A., et al. (2018). Development of a Bluetooth Controlled Fire Fighting Robot. International Conference on Robotics and Automation Sciences, 72-76.

15. IEEE Firefighting Robots: https://ieeexplore.ieee.org/document/8894031

16. Jivani, A. G. (2014). GSM Based Home Automation System Using App-Inventor for Android Mobile Phone. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 3(9), 12121–12128.

17. Shah, S., & Parmar, K. (2017). Blynk Based Real-Time Home Automation Using IoT. International Journal of Computer Applications, 179(7), 33–37.

18. Bhaskar, K. (2021). Bluetooth Controlled Robotic Car using Android App. International Journal of Science and Research (IJSR), 10(4), 527–532.

19. Saxena, N., & Pandey, V. (2022). An IoT-Based Real-Time Fire Detection and Alert System. Procedia Computer Science, 185, 227–233.

20. Walia, A., & Sharma, D. (2019). IoT-Based Fire Detection and Safety System. International Journal of Computer Sciences and Engineering, 7(4), 210–215.

21. Arduino Documentation. https://www.arduino.cc

22. Thakur, R. (2020). Fire Alert and Safety System Using IoT. International Journal of Computer Applications, 176(10), 8–13.

23. Patil, M. P., & Mane, D. P. (2021). IoT Based Smart Fire Detector using ESP32. Journal of Emerging Technologies and Innovative Research (JETIR), 8(3), 106–112.

24. Sivanesh, P., et al. (2018). Smart Fire Detection and Notification System Using GSM and GPS. International Journal of Scientific Research in Engineering and Management, 2(6), 65–69.

25. Mittal, S., & Gupta, S. (2017). Bluetooth Based Wireless Controlling of Robot. International Journal of Engineering Sciences & Research Technology, 6(2), 24–27.

26. Singh, D., & Kaur, A. (2018). Fire Detection and Smart Notification System Using ESP32. International Journal of Engineering Development and Research (IJEDR), 6(2), 620–625.

27. Velasco, G., et al. (2017). Fire Detection System Using Smoke and Temperature Sensor. IEEE Sensors Conference, 1230–1235.

28. Subramanian, R., & Shobana, S. (2020). GPS and GSM Based Alert System for Fire Accidents. Journal of Engineering Research and Applications, 10(5), 22–26.

29. Solanki, P., & Vora, R. (2019). IoT Enabled Smart Safety Device for Fire and Gas Leakage. International Journal of Innovative Research in Science, Engineering and Technology, 8(9), 9000–9005.

30. Blynk + GSM + GPS Fire Alert Project on Hackster.io. https://www.hackster.io

Programing Details

```
#include <BluetoothSerial.h>

#include <TinyGPSPlus.h>

#include <HardwareSerial.h>


BluetoothSerial SerialBT;

TinyGPSPlus gps;

HardwareSerial sim800(1);

HardwareSerial gpsSerial(2);


#define gasSensorPin 34

#define flameSensor1 32

#define flameSensor2 33

#define relayPin 25


#define motor1A 26

#define motor1B 27

#define motor2A 14

#define motor2B 12


bool fireDetected = false;

bool gasDetected = false;


void setup() {
```

```
  Serial.begin(115200);
  SerialBT.begin("FireBot");
  sim800.begin(9600, SERIAL_8N1, 16, 17);
  gpsSerial.begin(9600, SERIAL_8N1, 18, 19);

  pinMode(gasSensorPin, INPUT);
  pinMode(flameSensor1, INPUT);
  pinMode(flameSensor2, INPUT);
  pinMode(relayPin, OUTPUT);

  pinMode(motor1A, OUTPUT);
  pinMode(motor1B, OUTPUT);
  pinMode(motor2A, OUTPUT);
  pinMode(motor2B, OUTPUT);
}

void loop() {
  // Check sensors
  fireDetected = digitalRead(flameSensor1) == LOW || digitalRead(flameSensor2) == LOW;
  gasDetected = analogRead(gasSensorPin) > 400;

  // Bluetooth manual control (optional)
  if (SerialBT.available()) {
    char cmd = SerialBT.read();
```

```
    manualControl(cmd);
  }

  // GPS update
  while (gpsSerial.available()) {
    gps.encode(gpsSerial.read());
  }

  // If fire or gas is detected
  if (fireDetected || gasDetected) {
    extinguishFire();

    if (gps.location.isValid()) {
      sendSMS(gps.location.lat(), gps.location.lng());
    }
  } else {
    stopMotors();
  }
}

void extinguishFire() {
  digitalWrite(relayPin, HIGH);
  moveForward();
  delay(5000);
```

```
  digitalWrite(relayPin, LOW);
}

void moveForward() {
  digitalWrite(motor1A, HIGH);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor2A, HIGH);
  digitalWrite(motor2B, LOW);
}

void stopMotors() {
  digitalWrite(motor1A, LOW);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor2A, LOW);
  digitalWrite(motor2B, LOW);
}

void manualControl(char cmd) {
  switch (cmd) {
    case 'F': moveForward(); break;
    case 'B': /* move backward */ break;
    case 'L': /* turn left */ break;
    case 'R': /* turn right */ break;
    case 'S': stopMotors(); break;
```

```
  }
}

void sendSMS(double lat, double lng) {
  sim800.println("AT+CMGF=1");
  delay(100);
  sim800.println("AT+CMGS=\"+1234567890\"");  // Your phone number
  delay(100);
  sim800.print("Fire/Gas Detected! Location: https://maps.google.com/?q=");
  sim800.print(lat, 6);
  sim800.print(",");
  sim800.print(lng, 6);
  delay(100);
  sim800.write(26);  // End of SMS
}
```