A Major Project Phase-II Report

On

"SMART MOVABLE ROAD DIVIDER WITH AMBULANCE PRIORITY SYSTEM USING IOT"

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE AWARD OF DEGREE OF

BACHELOR OF ENGINEERING

IN

ELECTRONICS & COMMUNICATION ENGINEERING

BY

N. GOUTHAM KRISHNA	**(1608-21-735-019)**
SHERI MANASA	**(1608-21-735-033)**
B. HARIKA	**(1608-21-735-303)**

Under the guidance of

Mr. A.ABHISHEK REDDY

M.Tech, (Ph.D)

Assistant Professor

Department of Electronics and Communication Engineering

MATRUSRI ENGINEERING COLLEGE

(An Autonomous Institution)

(Sponsored by Matrusri Education Society, Estd1980)

(Approved by AICTE, Affiliated to Osmania University)

#16-1-486,Saidabad, Hyderabad, Telangana-500 059

www.matrusri.edu.in

2024-2025

Matrusri Engineering College

(An Autonomous Institution)

(Sponsored by: MATRUSRI EDUCATION SOCIETY, Estd: 1980)

(Approved by AICTE, Affiliated to Osmania University)

email: principal@matrusri.edu.inweb site: www.matrusri.edu.in Ph: 040-24072764

Department of Electronics and Communication Engineering

(NBA Accredited)

Date: 06/06/2025

Certificate

This is to certify that the Major Project Phase-2 report entitled **“Smart Movable Road Divider With Ambulance Priority System Using Iot”** being submitted by **Mr. N.Goutham Krishna (1608-21-735-019), Ms. Sheri Manasa (1608-21-735-033), Ms. B.Harika (1608-21-735-303)** in partial fulfillment for the award of the Degree of Bachelor of Engineering in Electronics and Communication Engineering of the Osmania University, Hyderabad, during 2024-25, is a record of bonafide work carried out under our guidance and supervision.

The result presented in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Major Project Phase-II Coordinator

Mr. A.Abhishek Reddy
M.Tech,(Ph.D)
Assistant Professor

Dr. N. Srinivasa Rao
HOD-ECE

DECLARATION

This is to declare that the work submitted in the present project work report titled "**Smart Movable Road Divider With Ambulance Priority System Using Iot**" is a record of bonafide work done by me in the Department of Electronics and Communication Engineering, Matrusri Engineering College. Saidabad, Hyderabad.

No part of the book is copied from books, journals, internet and wherever the subject content is taken, the same has been duly referred in the text. The report generated is based on the project work carried out entirely by us and not copied from any other source.

Mr. N.Goutham Krishna(1608-21-735-019)
Ms. Sheri Manasa (1608-21-735-033)
Ms. B.Harika (1608-21-735-303)

Place: Hyderabad,
Date: 06/06/2025.

ACKNOWLEDGMENT

We would like to take this opportunity to place it on the record, that this project would never have taken shape but for the cooperation extended to us by certain individuals. Though this is not possible to name all of them, it would be a pardonable on us part if we don’t mention some of the very important persons Sincerely we acknowledge our deep sense of gratitude to the project guide, **Mr.A.Abhishek Reddy**,Assistant Professor for his constant encouragement, help and valuable suggestions. We wish to thank him for his constant motivation and help throughout the project.

We would also like to express our deep gratitude to the Head of Department **Dr. N SRINIVASA RAO** and Project Coordinators **Dr. I. SHARATH CHANDRA** and **Dr. K. KOTESHWAR RAO** for their timely cooperation while carrying the project. It is their friendliness that made me feel free and learn more from them.

Last but not the least we would thank to all those people associated directly or indirectly with the project.

Mr. N.Goutham Krishna

Ms. Sheri Manasa

Ms. B.Harika

ABSTRACT

Traditional road dividers are static and unable to adapt to changing traffic conditions, often leading to congestion, delays, and blocked paths for emergency vehicles during peak hours. To address these challenges, this project proposes a Smart Movable Road Divider with Ambulance Priority System Using IoT. The system dynamically adjusts the position of the road divider based on real-time traffic density, optimizing lane allocation and improving traffic flow. Additionally, an ambulance priority mechanism is integrated using RFID technology, which detects approaching ambulances and automatically creates a clear path for them, minimizing response time. The system also includes a feature to detect signal violations, enhancing road safety and discipline. This intelligent and adaptive approach ensures better traffic management, reduces accident risks, and supports emergency response systems efficiently—ultimately contributing to safer and smarter urban transportation infrastructure.

Key words: Smart Movable Road Divider , Traffic Density , Ambulance Priority System, RFID Technology .Signal Violation Detection .

CONTENTS

	Page No.
Certificate	**ii**
Declaration	**iii**
Acknowledgement	**iv**
Abstract	**v**
List of Figures	**viii**
List of Tables	**x**
Chapter 1: Introduction	**1**
1.1 Introduction	1
1.2 Problem statement	2
1.3 Objectives	2
1.4 Methodologies	3
1.5 Motivation	3
1.6 Layout of the Project	4
Chapter 2: Literature Survey	**5**
Chapter 3: Problem Specification	**9**
3.1 Introduction to project	9
3.2 Block Diagram	10
3.3 Hardware Selection	11
Chapter 4: Hardware Implementation	**14**
4.1 Hardware Components	14
4.1.1 ESP32 Microcontroller	14
4.1.2 IR (Infrared) Sensors	17
4.1.3 RFID Technology	19
4.1.4 MG996R Digital Servo Motor	21
4.1.5 LM2596 Voltage Regulator	22
4.1.6 NTE3019 Light Emitting Diode (LED),5mm	24
4.1.7 16 X 2 LCD Display	24
4.1.8 Buzzer	26
Chapter 5: Software Implementation	**28**
5.1 Arduino IDE Installation	28
Chapter 6: Integration and testing	**33**

6.1 Introduction 33
6.2 Module-Level Testing 33
6.3 System Integration 34
6.4 Functional Testing 34
6.5 Load and Performance Testing 34
6.6 Final Validation 35
Chapter 7: Results **36**
7.1 Developed System 36
Chapter 8: Conclusion and Future Scope **38**
8.1 Conclusion 38
8.2 Future Scope 38
Appendices **39**
References **47**

List of Figures

Figure Number	Titles	Page Number
Figure 3.1	Block Diagram for Smart movable road divider with ambulance priority system using IOT	10
Figure 4.1	Pin Diagram Of ESP32 Microcontroller	14
Figure 4.2	IR Proximity Sensor	18
Figure 4.3	Interface to Arduino	19
Figure 4.4	Basic RFID System	20
Figure 4.5	5 EM-18 RFID Reader Module pin diagram	20
Figure 4.6	MG996R High Torque Metal Gear Dual Ball Bearing Servo Motor	22
Figure 4.7	LM2596 Voltage Regulator	23
Figure 4.8	NTE3019 Light Emitting Diode (LED) Red diffused	24
Figure 4.9	16 x 2 LCD Display	25
Figure 4.10	LCD 16×2 Pin Diagram	25
Figure 4.11	Buzzer	26
Figure 5.1	Download Arduino IDE Software	28
Figure 5.2	Launch Arduino IDE	29
Figure 5.3	Open a Project	29
Figure 5.4	Select Arduino board	30
Figure 5.5	Select Serial port	31
Figure 5.6	6Description of each symbol on Arduino IDE toolbar	31
Figure 5.7	Electric Vehicle BMS	32
Figure 7.1	Smart Movable Road Divider With Ambulance Priority Using IOT	36
Figure 7.2	IR Sensors detecting vehicles	36
Figure 7.3	Ambulance Detection Using RFID	37

List of Tables

Table Number	Title	Page Number
Table 4.1	Specifications Of ESP32 Microcontroller	15
Table 4.2	Specifications of IR Sensors	18
Table 4.3	Features of RFID	21
Table 4.4	Specifications of MG996R Digital Servo Motor	22
Table 4.5	Specifications of Buzzer	27

Chapter-1

INTRODUCTION

1.1 Introduction

Traffic congestion has emerged as one of the most pressing issues in urban areas, especially in developing and underdeveloped cities. With rapid population growth and a steady increase in vehicle usage, urban infrastructure often fails to keep pace with the demand. The lack of efficient traffic management leads to prolonged travel times, excessive fuel consumption, and environmental degradation due to the emission of harmful gases like carbon dioxide (CO_2). Individuals stuck in traffic not only waste valuable time and fuel but are also exposed to high levels of air pollution, which adversely impacts health. Furthermore, these congested conditions often provide a favourable environment for illegal activities such as robbery and chain snatching.

Despite multiple efforts by the government—including imposing restrictions on parking and halting vehicles on busy roads—the issue of traffic overcrowding continues to rise. A significant factor contributing to this is the inefficient utilization of existing resources rather than the absence of resources themselves. Thus, it is crucial to develop solutions that make smarter use of the current infrastructure, rather than relying solely on expansion. One critical challenge in congested traffic is the delay caused to emergency services. Ambulances and other emergency vehicles frequently get trapped in heavy traffic, leading to delayed medical assistance and, in severe cases, loss of life. To address these challenges, this project proposes an innovative solution: a Smart Movable Road Divider with Ambulance Priority System Using IoT.

The proposed system features a movable road divider capable of adjusting its position dynamically based on real-time traffic density on both sides of the road. By reallocating lane space according to the volume of traffic in each direction, the system ensures smoother traffic flow during peak hours. Additionally, an ambulance priority mechanism is integrated using RFID technology, where RFID tags placed in emergency vehicles are detected by RFID readers, prompting the system to create a clear lane for the ambulance.

Another major issue addressed in this project is signal violation. Vehicles that jump traffic signals are detected through the system, and an alert is automatically sent to the nearest traffic police station using a Wi-Fi module. This not only improves enforcement but also discourages reckless driving behaviour.

In summary, this project aims to present a smart, cost-effective, and adaptive solution to the day-to-day traffic problems faced in urban areas. By combining traffic density analysis, emergency vehicle prioritization, and violation detection, the system enhances urban mobility, supports emergency services, and promotes road safety—all while making the most of existing infrastructure.

1.2 Problem Statement

Urban and semi-urban areas face severe traffic congestion due to the inability of static road dividers to adapt to dynamic traffic conditions. These fixed dividers often lead to an imbalance in lane utilization, where one side experiences heavy traffic while the other remains underutilized, resulting in increased travel time, fuel wastage, and commuter frustration. Emergency vehicles such as ambulances frequently get trapped in traffic, causing critical delays in providing timely medical assistance. The absence of a dedicated or responsive system to prioritize emergency vehicles further worsens the situation.

Although RFID technology offers potential for vehicle detection and tracking, it is not widely implemented for emergency vehicle prioritization in traffic management systems. Additionally, traffic signal violations by regular vehicles are common and often go undetected due to a lack of automated monitoring and real-time alert mechanisms. This compromises road safety and challenges the enforcement of traffic rules.

1.3 Objectives

To develop a movable road divider system that can dynamically adjust its position based on real-time traffic density on both sides of the road to reduce congestion and optimize lane usage. To integrate an ambulance priority system using RFID technology, enabling the automatic detection of emergency vehicles and providing them with a clear path by temporarily adjusting traffic infrastructure. To implement a traffic signal violation detection mechanism that identifies unauthorized vehicle movement across red signals and immediately reports the incident to nearby traffic authorities via a Wi-Fi communication module. To utilize IoT-based components and sensors for real-time monitoring, control, and communication between the system modules for efficient and automated operation.

1.4 Methodologies

The system operates through multiple integrated technologies to address real-time traffic congestion and emergency vehicle prioritization. Firstly, infrared (IR) sensors are installed on either side of the road to monitor the vehicle density. These sensors detect the number of vehicles on each side and send the data to an ESP32 microcontroller, which acts as the control unit. Based on this traffic density data, the system actuates a high-torque MG996R servo motor to shift the road divider, allocating more space to the congested side for efficient traffic flow. The project also integrates an RFID-based ambulance priority system, where ambulances are equipped with RFID cards that are detected by an EM-18 RFID reader positioned near the divider zone. Upon detection, the system immediately halts divider movement, activates an active buzzer and displays a priority message on a 16x2 LCD display, giving the ambulance a clear path. The system also monitors vehicle signal violations, and if a regular vehicle attempts to cross during restricted phases, an alert is triggered, and a message is sent wirelessly to the nearest traffic control center using the ESP32's Wi-Fi module. The entire setup is powered and stabilized using an LM2596 voltage regulator, ensuring smooth and safe operation of all components. This integrated approach ensures better traffic management, quick emergency response, and automation of routine traffic regulation tasks using IoT.

1.5 Motivation

This project aims to create an intelligent traffic management system using affordable technologies such as IoT, IR sensors, RFID, and the ESP32 microcontroller. The smart movable road divider dynamically changes lane configurations to ease traffic congestion. An ambulance priority system detects emergency vehicles and automatically clears their path by adjusting the divider, ensuring quicker response times. Additionally, a violation detection feature monitors and flags traffic rule breakers to improve overall road safety. This integrated approach seeks to make roads safer, reduce delays, and enhance emergency services.

1.6 Layout Of The Project

- **Chapter 1: Introduction**

  This chapter introduces the aim of reducing traffic congestion and prioritizing ambulances using a smart movable road divider system integrated with IoT.

- **Chapter 2: Literature Survey**

  It reviews existing traffic management techniques and IoT applications, identifying gaps that the proposed system addresses.

- **Chapter 3: Problem Specification**

  This chapter defines the core problem and explains the system architecture and working principles of the proposed solution.

- **Chapter 4: Hardware Implementation**

  It describes the components used and how they are physically connected to form the complete hardware setup.

- **Chapter 5: Software Implementation**

  This chapter outlines the logic, coding, and platform setup used to program the ESP32 and interface the sensors and actuators.

- **Chapter 6: Integration and Testing**

  It explains the integration of all modules and presents the testing procedures followed to validate the system.

- **Chapter 7: Results**

  This chapter highlights the successful functioning of each module and overall system performance based on practical testing.

- **Chapter 8: Conclusion and Future Scope**

  It summarizes the project outcomes and suggests future improvements like AI integration and wider deployment.

- **Appendices**

  This section includes the complete code, circuit diagrams, and snapshots of the working system and IoT dashboard.

Chapter 2

LITERATURE SURVEY

[1]Pranav Maheshwari, Deepanshu Suneja, Praneet Singh, Yogeshwar Mutneja, "Smart traffic optimization using image processing",2015 IEEE 3rd International conference on MOOC's, Innovation and Technology in Education (MITE), ISBN: 978-1-4673-9/15/$31.00, 2015 IEEE.

In this paper, Pranav Maheshwari and co-authors presented a smart traffic control system that leverages image processing techniques to reduce traffic congestion. The proposed system installs cameras on traffic signal poles to continuously monitor vehicle flow. These cameras capture real-time images of traffic at intersections, which are then processed to determine traffic density on each road. Based on the analysis, the system dynamically assigns optimal signal timings by adjusting the duration of green and red lights. This intelligent, vision-based approach eliminates the need for manual intervention or fixed signal cycles, enabling more responsive and efficient traffic management.

[2]Er. Faruk Bin Poyen, Amit Kumar Bhakta, B.Durga Manohar, Imran Ali, ArghyaSantra, AwanishPratap Rao, "Density based traffic control", 2016 International journal of advanced engineering, management and science (IJAEMS) , vol-2, issue-8, Aug-2016, ISSN:2454-1311.

In this paper, Er. Faruk Bin Poyen and co-authors proposed a traffic control system focused on minimizing traffic congestion through real-time traffic density monitoring. The study utilized Passive Infrared (PIR) sensors to detect the number of vehicles at intersections. Based on the density data collected, the system dynamically adjusted the duration of red and green traffic lights. This adaptive control mechanism ensures that roads with higher traffic density receive longer green light durations, while roads with lighter traffic experience shorter wait times. The approach aimed to optimize traffic flow, reduce unnecessary waiting time at signals, and improve overall road efficiency, especially in urban areas.

[3]B.Nandhu Rathi , M.Radha , T.U.Sugitha , V.Tharani and V.Karthikeyan, "Intelligent traffic control system for congestion control, emergency vehicle clearance and stolen vehicle detection",Asian journal of applied science and technology (AJAST), Volume 1, Issue 1, Pages 122 125, February 2017.

In this paper, B. Nandhu Rathi and co-authors proposed an intelligent traffic control system designed to address multiple challenges in urban traffic management. The system focuses on three primary objectives: reducing traffic congestion, providing a clear path for emergency vehicles, and detecting stolen vehicles. Emergency vehicles are equipped with RFID tags, enabling the traffic signals to automatically recognize and prioritize them by switching to green, thus ensuring swift and uninterrupted passage. Vehicle density at intersections is calculated using IR sensors, allowing for dynamic signal timing adjustments to optimize traffic flow. Additionally, the system can identify stolen vehicles using RFID tag verification, enhancing road safety and aiding law enforcement efforts. This integrated approach demonstrates the potential of combining sensor-based automation and RFID technology in smart city traffic systems.

[4]SabeenJavid, Ali Sufian, Saima Pervaiz, Mehak Tanveer, "Smart traffic management system using IOT", International conference on advanced communication technology (ICACT), ISBN: 979 11-88428-01-4, February 2018.

In this paper, Sabeen Javid and co-authors proposed a smart traffic management system that utilizes the Internet of Things (IoT) to improve urban traffic control and road safety. The system integrates sensors and microcontrollers to monitor real-time traffic conditions, vehicle flow, and road usage. Data collected from these IoT-enabled devices is analyzed to dynamically manage traffic lights, reduce congestion, and enhance the efficiency of transportation networks. The system also supports features like emergency vehicle detection, accident alerts, and data logging for future traffic planning. By connecting various elements of traffic infrastructure through IoT, the authors demonstrated how cities can move toward smarter, automated, and more responsive traffic ecosystems.

[5]HemlataDalmia, Kareddy Damini, Aravind Goud Nakka, "Implementation of movable road divider using IOT", 2018 International conference on computing, power and communication technologies (GUCON), Galgotias university, Greater Noida, UP, India. Sep 28-29, 2018.

In this paper, Hemlata Dalmia and co-authors presented a novel approach to dynamic traffic lane management through the implementation of a movable road divider controlled via IoT technologies. The system is designed to address the issue of traffic congestion by reallocating road lanes in real-time based on current traffic flow. Using IoT-based sensors and a microcontroller unit, the system detects traffic density on either side of the road and accordingly shifts the road divider to balance the flow. The goal is to maximize road utilization during peak hours or special circumstances such as public events or accidents. The authors demonstrated how automation and real-time responsiveness can improve traffic efficiency and reduce urban congestion through cost-effective and scalable technology.

[6] Satya Srikanth Palle, Sriraksha B M, Vibha H B, Yeshashwini A, "Implementation of Smart Movable Road Divider and Ambulance Clearance Using IoT," ***2019 4th International Conference on Recent Trends in Electronics, Information, Communication and Technology (RTEICT-2019)*****, May 17–18, 2019.**

In this paper, Satya Srikanth Palle and co-authors proposed a smart traffic management system that combines a movable road divider and ambulance clearance mechanism using IoT. The system aims to reduce traffic congestion and provide priority access for emergency vehicles. The movable road divider dynamically adjusts lane distribution based on real-time traffic density, which is detected through sensors. For ambulance clearance, RFID tags are used to identify approaching emergency vehicles, prompting the system to automatically switch traffic lights and move the divider to clear a path. The integration of IoT technologies ensures real-time data collection, control, and decision-making, thereby improving the overall efficiency and safety of urban traffic systems.

[7]Traffic Incident Detection Sensor and AlgorithmsR. Weil, J. Wootton, A. Garcia Ortiz 2015

In this paper, R. Weil, J. Wootton, and A. Garcia Ortiz [7] proposed a real-time traffic incident detection system using a combination of sensors and algorithms. The study utilized various types of sensors, such as loop detectors and acoustic and video-based devices, to continuously monitor traffic flow parameters like vehicle speed, count, and lane occupancy. These data were processed using intelligent algorithms to detect anomalies that indicate possible incidents, such as accidents or stalled vehicles. The system aimed to enable faster response times, reduce traffic congestion, and improve overall road safety. This concept supports our project's goal of integrating smart sensing and automated decision-making for effective traffic management and emergency vehicle prioritization.

[8]Rohini Temkar, Vishal Asrani, Pavitra Kannan, "IoT: Smart Vehicle Management System for Effective Traffic Control and Collision Avoidance," IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015.

This paper presents an IoT-based system that integrates GPS, real-time vehicle tracking, and collision avoidance to improve traffic control and road safety. The authors propose a smart vehicle management framework where vehicles communicate with a centralized system to share location and movement data. This enables better traffic regulation, accident prevention, and route optimization. The system uses real-time data to monitor traffic conditions and warn drivers about potential collisions, helping to reduce traffic congestion and enhance urban mobility.

[9]Z. Sun, G. Bebis, R. Miller, "On-road Vehicle Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.

This paper provides a comprehensive review of various sensor-based and image processing techniques used for on-road vehicle detection. It discusses methods involving camera vision, radar, and infrared sensors, analyzing their effectiveness in detecting vehicles under different driving conditions. The study highlights challenges like occlusion, varying lighting, and complex backgrounds, and evaluates the strengths and limitations of each detection approach, offering insights into the development of more accurate and reliable vehicle detection systems.

Chapter 3

PROBLEM SPECIFICATION

3.1 Introduction of the project

Urban traffic management today faces a critical challenge in adapting to the rapid growth of vehicles on city roads, especially in densely populated areas. Existing traffic systems predominantly rely on fixed-time signals and rigid, immovable road dividers that fail to respond dynamically to real-time traffic conditions. This inflexibility leads to chronic congestion, inefficient use of road space, and, most critically, delayed passage of emergency vehicles such as ambulances.

The inability to dynamically manage and allocate road lanes based on current traffic density results in severe delays, which can have dire consequences during emergencies. Ambulances stuck in traffic face crucial time losses that could mean the difference between life and death. Additionally, manual traffic enforcement struggles to monitor traffic violations effectively, resulting in unsafe driving behaviors and further aggravating congestion and accident risks.

The core problem addressed in this project is the lack of an intelligent, automated system capable of real-time traffic monitoring and control that prioritizes emergency vehicles while optimizing road usage. Current traffic infrastructures do not provide mechanisms for adaptive lane management or quick emergency clearance, especially in high-traffic urban settings.

This project proposes a smart traffic management solution that integrates Internet of Things (IoT) technologies, Infrared (IR) sensors, Radio Frequency Identification (RFID) tags, ESP32 microcontrollers, and motorized movable road dividers. The system is designed to detect traffic density using IR sensors and identify ambulances through RFID, enabling it to create a dedicated lane by automatically shifting movable road dividers. This real-time adaptability helps reduce ambulance transit times and alleviates congestion by better utilizing available road space.

Furthermore, the system can detect traffic violations through sensor inputs, supporting enhanced traffic discipline without the need for continuous human intervention. This automated enforcement capability improves road safety and ensures smoother traffic flow.

This solution is particularly relevant for urban areas in India, where traffic volumes are high, infrastructure is often stretched, and manual traffic control resources are limited.

By providing a low-cost, scalable, and effective approach to traffic management, this project addresses critical gaps in current systems and aims to improve safety, efficiency, and responsiveness on urban roads.

3.2 Block Diagram

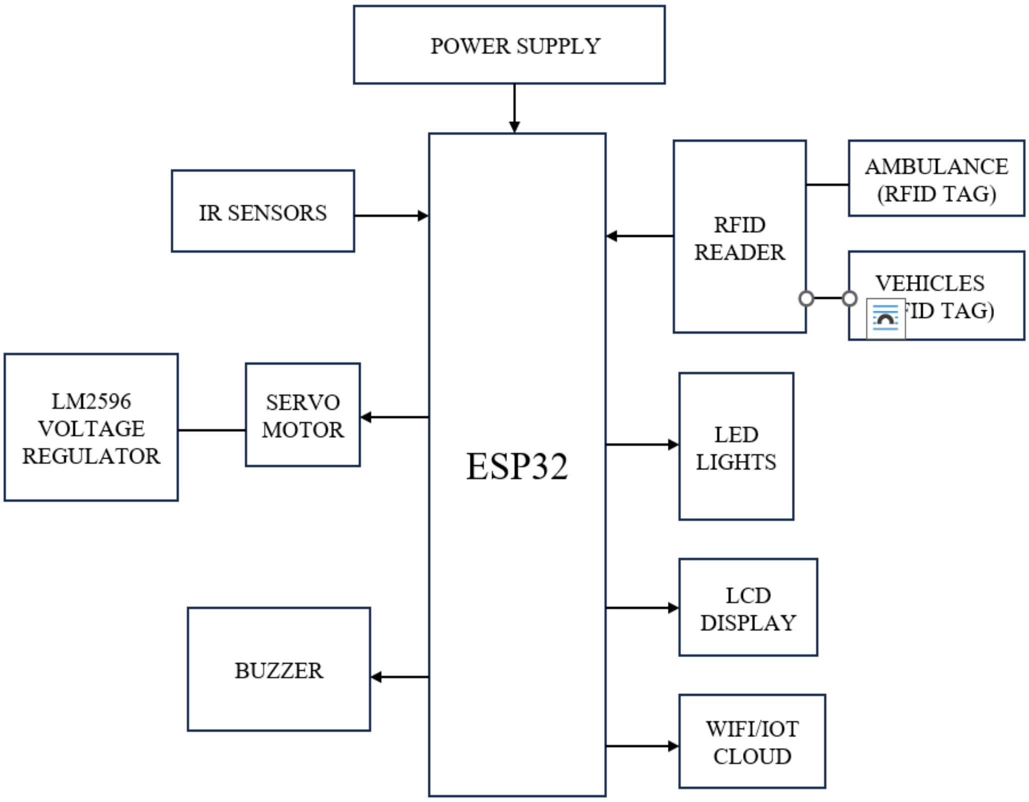

Fig 3.1 Block Diagram for Smart movable road divider with ambulance priority system using IOT

The block diagram of the "Smart Movable Road Divider with Ambulance Priority System Using IoT" represents an integrated and intelligent traffic management system designed to improve vehicular flow and prioritize emergency services. At the core of the system is the ESP32 microcontroller, which acts as the central processing unit responsible for receiving, analyzing, and executing real-time commands based on sensor and RFID inputs. The system uses Infrared (IR) sensors to detect the vehicle density on both sides of the road. Based on this input, the ESP32 evaluates which side has more traffic and activates a servo motor to move a physical road divider toward the side with lower density, thereby creating additional space for the congested side and easing traffic flow. The servo motor's operation is supported by an LM2596 voltage regulator, ensuring a stable power supply. An RFID reader is employed to identify

RFID tags fixed on vehicles, especially ambulances. When an ambulance is detected at a red signal, the ESP32 overrides the normal signal flow to give it priority by switching the signal to green, activating a green LED, sounding a buzzer for alert, and displaying an emergency message on an LCD screen. Additionally, if any regular vehicle attempts to violate a red signal, its RFID tag is read and its vehicle information is uploaded to a remote server via the built-in Wi-Fi/IoT cloud module, while a buzzer is simultaneously triggered to indicate the violation. The entire system is powered by a centralized power supply and is built to operate automatically in real-time, ensuring faster movement for emergency vehicles, efficient handling of traffic congestion, and effective monitoring of traffic rule violations.

3.3 Hardware Selection

The hardware components used in this project were selected based on functionality, compatibility, and cost-efficiency. Each component plays a specific role in implementing the smart traffic and emergency response system.

1. ESP32 Microcontroller

The ESP32 is a low-power, dual-core microcontroller with built-in Wi-Fi and Bluetooth capabilities.

Reason for selection: It acts as the brain of the system, handling sensor inputs, RFID data, motor control, and wireless communication. Its multiple GPIO pins and built-in connectivity features make it ideal for IoT-based applications like this one.

2. IR (Infrared) Sensors

IR sensors detect the presence of objects or vehicles using infrared light.
Reason for selection: These sensors are used to monitor vehicle density on both sides of the road. They are inexpensive, reliable for short-range detection, and easy to interface with the microcontroller.

3. RFID Reader

The RFID reader is a device used to read the data stored on RFID tags attached to vehicles.

Reason for selection: It allows the system to identify emergency vehicles like ambulances and detect vehicles violating red signals. It provides contactless identification, which is fast and secure.

4. RFID Tags

RFID tags are small electronic devices that contain a unique identification number. **Reason for selection:** Tags are attached to ambulances and regular vehicles to enable identification. This helps the system give priority to emergency vehicles and record data of vehicles that jump signals.

5. Servo Motor

A servo motor is an actuator used for precise control of angular position. **Reason for selection:** It is used to move the road divider left, right, or to the center based on traffic density. Its accuracy and ease of control make it ideal for mechanical movement tasks in real-time systems.

6. LM2596 Voltage Regulator

The LM2596 is a step-down (buck) voltage regulator that maintains a stable output voltage.

Reason for selection: It ensures that the servo motor and other components receive a constant voltage, preventing damage due to voltage fluctuations and enhancing system stability.

7. LED Lights (Red/Green)

LEDs are light-emitting diodes used for visual indication. **Reason for selection:** They are used to show traffic signal status. The green LED especially indicates ambulance priority, alerting other vehicles to make way.

8. LCD Display

The LCD (Liquid Crystal Display) shows text-based information to users in real-time. **Reason for selection:** It displays system messages such as “Ambulance Detected” or “Signal Violation”, making the operation of the system more transparent and informative.

9. Buzzer

A buzzer is an audio signaling device that emits a sound when activated. **Reason for selection:** It provides auditory alerts during important events like ambulance arrival or red signal violations, warning nearby drivers or authorities.

10. Wi-Fi / IoT Cloud (via ESP32)

The ESP32's built-in Wi-Fi module enables communication with cloud platforms. **Reason for selection:** It is used to upload data (e.g., vehicle numbers of signal violators) to the cloud for real-time monitoring, enabling remote access and traffic data analytics.

11. Power Supply

The power supply provides the necessary electrical energy to the entire circuit. **Reason for selection:** A reliable power source is crucial to ensure that all components operate without interruption and maintain consistent system performance.

Chapter 4

HARDWARE IMPLEMENTATION

4.1 Hardware Components

4.1.1 ESP32 Microcontroller

ESP32 is a series of powerful, power-efficient, cheap microcontrollers that comes with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series consist of a Tensilica Xtensa LX6 32-bit, dual-core microprocessor has two processors running at 240MHz and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. ESP32 is developed by a semiconductor-based company Espressif Systems in China, and it is manufactured by TSMC (Taiwan Semiconductor Manufacturing Company).

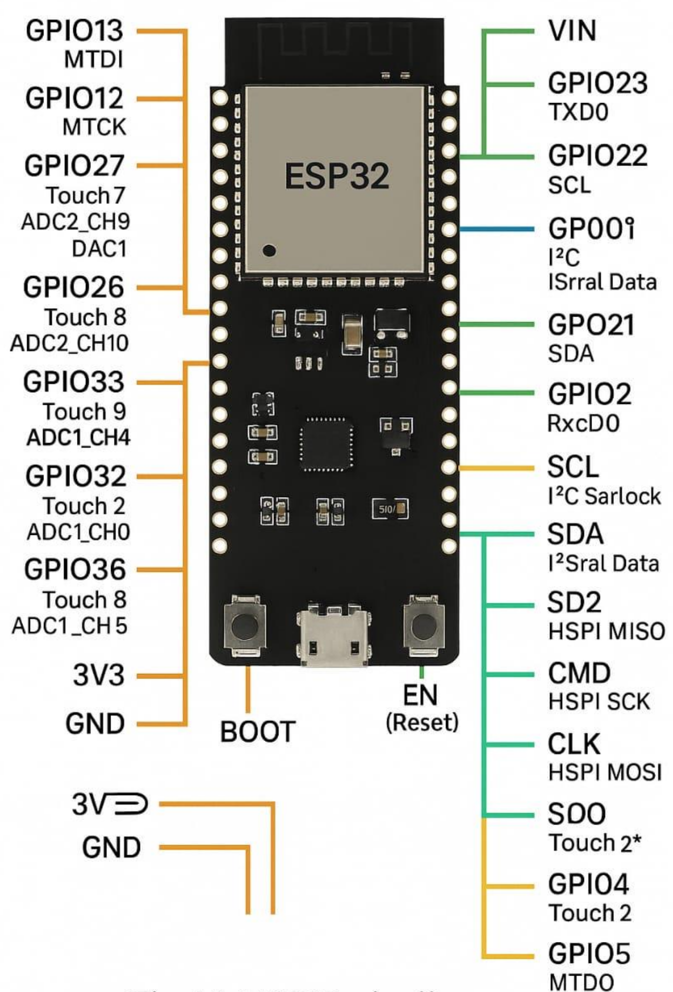

Fig 4.1 ESP32 pin diagram

Table 4.1 Specifications Of ESP32 Microcontroller

Parameter	**Details**
Microcontroller	ESP32
Processor	Dual-core Tensilica LX6 (up to 240MHz)
Memory	520KB SRAM, 448KB ROM, 4MB Flash (program storage)
Wi-Fi	802.11 b/g/n (2.4GHz)
Bluetooth	Bluetooth v4.2 and Bluetooth Low Energy (BLE)
Operating Voltage	2.2V to 3.6V
GPIO Pins	30 (General Purpose Input/Output)
ADC Pins	18 (Analog-to-Digital Converter)
Interfaces	UART, SPI, I2C, I2S, CAN
USB	USB 2.0 Type-C for programming and power
Power Supply	5V via USB or external power source (with appropriate voltage regulation)

Power

The ESP32 development board includes two primary power pins – a 5V pin and a 3.3V pin. The 5V pin is intended for directly supplying power to the ESP32 and its peripherals when using a regulated 5V source. The 3.3V pin, on the other hand, is the output of the onboard voltage regulator (CP2102) and is primarily used to power external components. Additionally, GND (Ground) pins are used to complete the electrical circuit and are essential for stable operation.

ADC (Analog to Digital Converter) Pins

The ESP32 has 18 12-bit SAR ADCs, enabling it to measure analog signals. Out of these, 15 channels are typically available on analog-enabled GPIO pins. The ADC channels are divided across two ADCs (ADC1 and ADC2):

- ADC1 Channels:
  - GPIO36 (ADC1_CH0), GPIO37 (ADC1_CH1), GPIO38 (ADC1_CH2), GPIO39 (ADC1_CH3),
  - GPIO32 (ADC1_CH4), GPIO33 (ADC1_CH5), GPIO34 (ADC1_CH6), GPIO35 (ADC1_CH7)
- ADC2 Channels:
  - GPIO4 (ADC2_CH0), GPIO0 (ADC2_CH1), GPIO2 (ADC2_CH2), GPIO15 (ADC2_CH3),
  - GPIO13 (ADC2_CH4), GPIO12 (ADC2_CH5), GPIO14 (ADC2_CH6), GPIO27 (ADC2_CH7),
  - GPIO25 (ADC2_CH8), GPIO26 (ADC2_CH9)

DAC (Digital to Analog Converter) Pins

The ESP32 also includes two 8-bit DAC channels, which are used to convert digital signals into analog voltages. These are particularly useful in applications like audio signal generation and waveform output. The available DAC pins are:

- DAC_1: GPIO25
- DAC_2: GPIO26

UART (Universal Asynchronous Receiver-Transmitter) Pins

ESP32 boards offer three UART interfaces: UART0, UART1, and UART2, which are capable of asynchronous communication at speeds up to 5 Mbps. Each UART port consists of TX (transmit), RX (receive), and optionally CTS (Clear To Send) and RTS (Request To Send) flow control pins.

- UART0: TXD (GPIO1), RXD (GPIO3), CTS (GPIO19), RTS (GPIO22)
- UART1: TXD (GPIO10), RXD (GPIO9), CTS (GPIO6), RTS (GPIO11)
- UART2: TXD (GPIO17), RXD (GPIO16), CTS (GPIO8), RTS (GPIO7)

SPI (Serial Peripheral Interface) Pins

The ESP32 supports three SPI interfaces – SPI, HSPI, and VSPI – operating in both master and slave modes. These are commonly used to communicate with devices like displays, memory chips, and sensors.

- Default SPI Pins:
  - SPI_D (GPIO8), SPI_WP (GPIO10), SPI_HD (GPIO9), SPI_Q (GPIO7),
  - SPI_CLK (GPIO6), SPI_CS0 (GPIO11)

- HSPI Pins:
    - V_SPI_ID (GPIO23), V_SPI_WP (GPIO22), V_SPI_HD (GPIO21),
    - V_SPI_Q (GPIO19), V_SPI_CLK (GPIO18), V_SPI_CS0 (GPIO5)
- VSPI Pins:
    - HSPI_ID (GPIO13), HSPI_WP (GPIO2), HSPI_HD (GPIO4),
    - HSPI_Q (GPIO12), HSPI_CLK (GPIO14), HSPI_CS0 (GPIO15)

PWM (Pulse Width Modulation) Pins

The ESP32 features up to 25 PWM-enabled GPIO pins, allowing it to generate analog-like signals from digital output. These PWM outputs are often used to control LED brightness, motor speeds, or servo positions. Most of the GPIOs on the board support PWM functionality

EN (Enable) Pin

The EN pin, also known as the Enable pin, is connected to the 3.3V regulator. Pulling this pin LOW will reset the microcontroller, which is useful for rebooting or putting the chip into a low-power state during development.

4.1.2 IR (Infrared) Sensors

The Multipurpose Infrared Sensor is an add-on for your line follower robot and obstacle avoiding robot that gives your robot the ability to detect lines or nearby objects. The sensor works by detecting reflected light coming from its own infrared LED. By measuring the amount of reflected infrared light, it can detect light or dark (lines) or even objects directly in front of it. An onboard RED LED is used to indicate the presence of an object or detect line. Sensing range is adjustable with inbuilt variable resistor.

The sensor has a 3-pin header which connects to the microcontroller board or Arduino board via female to female or female to male jumper wires. A mounting hole for easily connect one or more sensor to the front or back of your robot chassis.

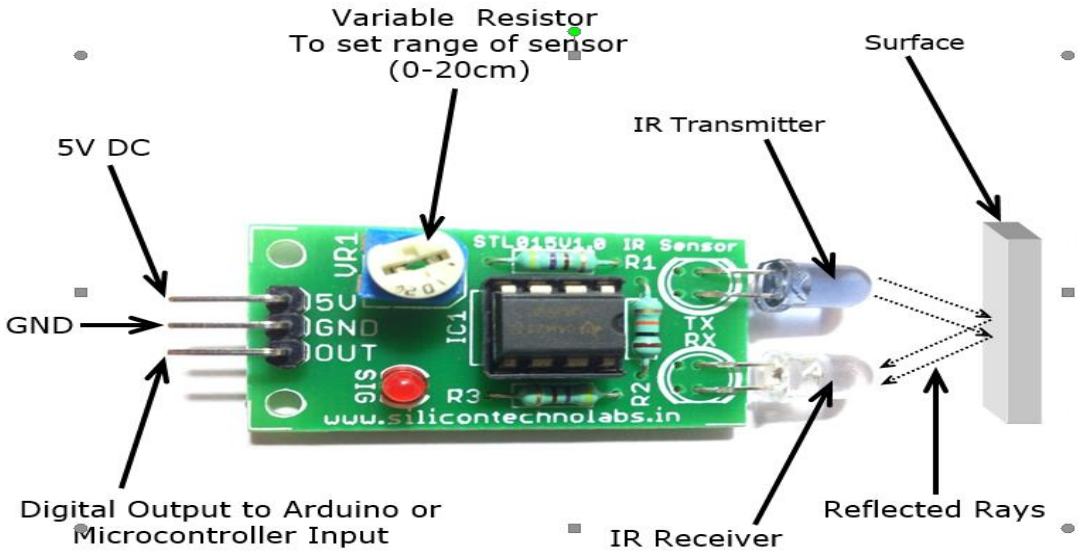

Fig 4.2 IR Proximity Sensor

Features

- 5VDC operating voltage
- I/O pins are 5V and 3.3V compliant
- Range: Up to 20cm
- Adjustable Sensing range
- Built-in Ambient Light Sensor
- 20mA supply current
- Mounting hole
    - Size: 50 x 20 x 10 mm (L x B x H)
    - Hole size: ϕ2.5mm

Table 4.2 Specifications of IR Sensors

Parameter	**Details**
Size	50 x 20 x 10 mm (L x B x H)
Hole size	Hole size

Interface to Arduino

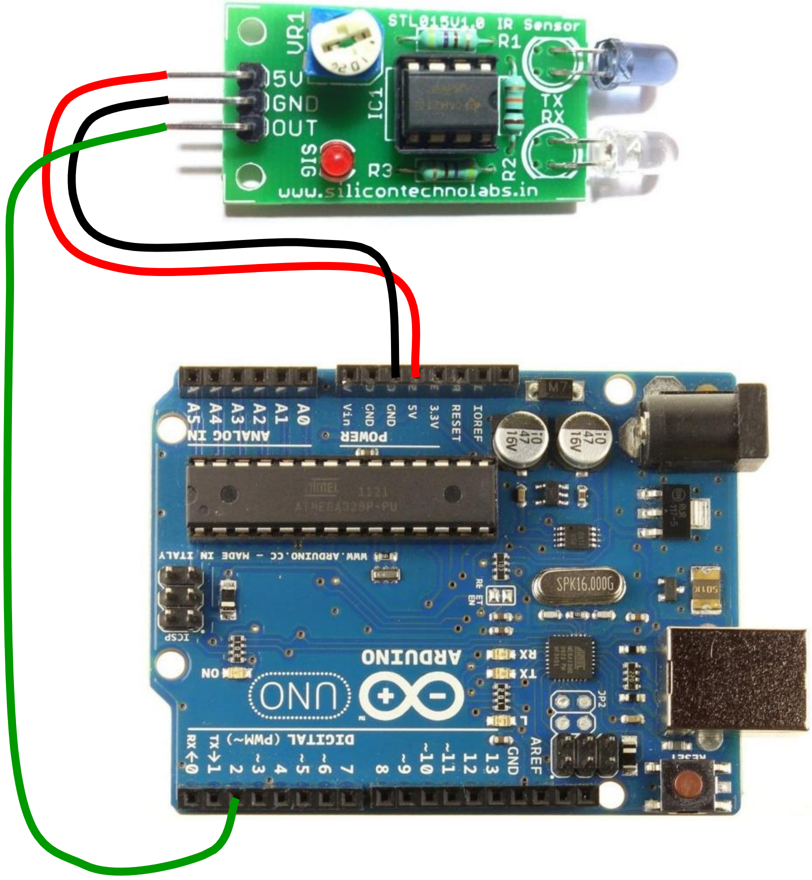

Fig 4.3 Interface to Arduino

4.1.3 RFID Technology

RFID (Radio Frequency Identification) is a technology that uses radio waves to automatically identify people or objects from a distance. It is a type of Automatic Identification (Auto-ID) system, similar to barcodes, magnetic strips, OCR, fingerprint recognition, and other identification technologies. An RFID system consists of a tag, which is attached to an object and contains a unique code, and a reader that detects this code when the tag comes within its range. The tag transmits the code using radio waves, which the reader then captures and processes to identify the object. Unlike barcode systems, RFID does not require line-of-sight, allowing for faster reading speeds, longer read ranges, and the ability to function through barriers. It also offers greater flexibility in tag placement and object movement. Due to these advantages, RFID technology is widely used in various fields such as manufacturing, agriculture, transportation, industrial automation, and supply chain management.

RFID Components

A combination of RFID technology and computing technology is called RFID system.

A RFID system consists of following components :

- Tag/Transponder (electronic label).
- Antenna (medium for tag reading).
- Reader /Interrogator (read tag information).
- Communication infrastructure (enable reader/RFID to work through IT infrastructure).
- Application software (user database/application/ interface).

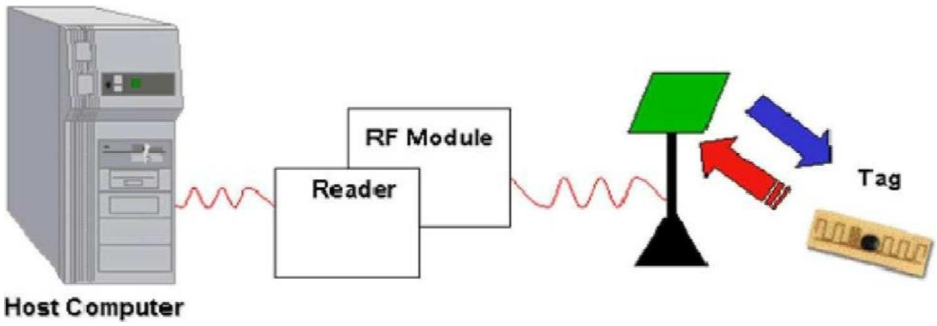

Fig 4.4 Basic RFID System

EM-18 RFID Reader

The EM-18 RFID Reader module operating at 125kHz is an inexpensive solution for your RFID based application. The Reader module comes with an on-chip antenna and can be powered up with a 5V power supply. Power-up the module and connect the transmit pin of the module to receive pin of your microcontroller. Show your card within the reading distance and the card number is thrown at the output. Optionally the module can be configured for also a Weigand output.

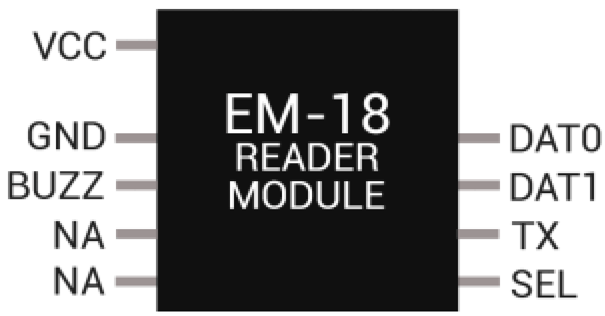

Fig 4.5 EM-18 RFID Reader Module pin diagram

Table 4.3 Features of RFID

RF Transmit Frequency	125kHz
Supported Standards	EM4001 64-bit RFID tag compatible
Communications Interface	TTL Serial Interface, Wiegand output
Communications Protocol	Specific ASCII
Communications Parameter	9600 bps, 8,N,1
Power Supply	4.6V – 5.5VDC +10% regulated
Current Consumption	50Ma <10 Ma at power down mode
Reading Distance	Up to 100mm, depending on tag
Antenna	Integrated
Size (L x W x H)	32 x 32 x 8mm

4.1.4 MG996R Digital Servo Motor

The MG996R is a high-torque digital servo motor featuring robust metal gearing that provides an impressive stall torque of up to 10kg·cm, making it suitable for applications requiring high strength in a compact form. It is an improved version of the widely used MG995, with enhanced shock-proofing, a redesigned PCB, and a more accurate IC control system. These improvements offer better precision, centering, and reduced dead bandwidth. The servo can rotate approximately 120 degrees (60° in each direction), making it ideal for robotic arms, RC cars, and hobby projects. It is compatible with any standard servo library or hardware, allowing easy control without the need for complex motor drivers or feedback mechanisms, which makes it beginner-friendly. The MG996R includes a 30cm cable with a standard 3-pin 'S' type female header that fits most popular receivers like Futaba, JR, GWS, Cirrus, and Hitec. Additionally, it comes with a variety of servo horns and mounting hardware for quick and versatile installation.

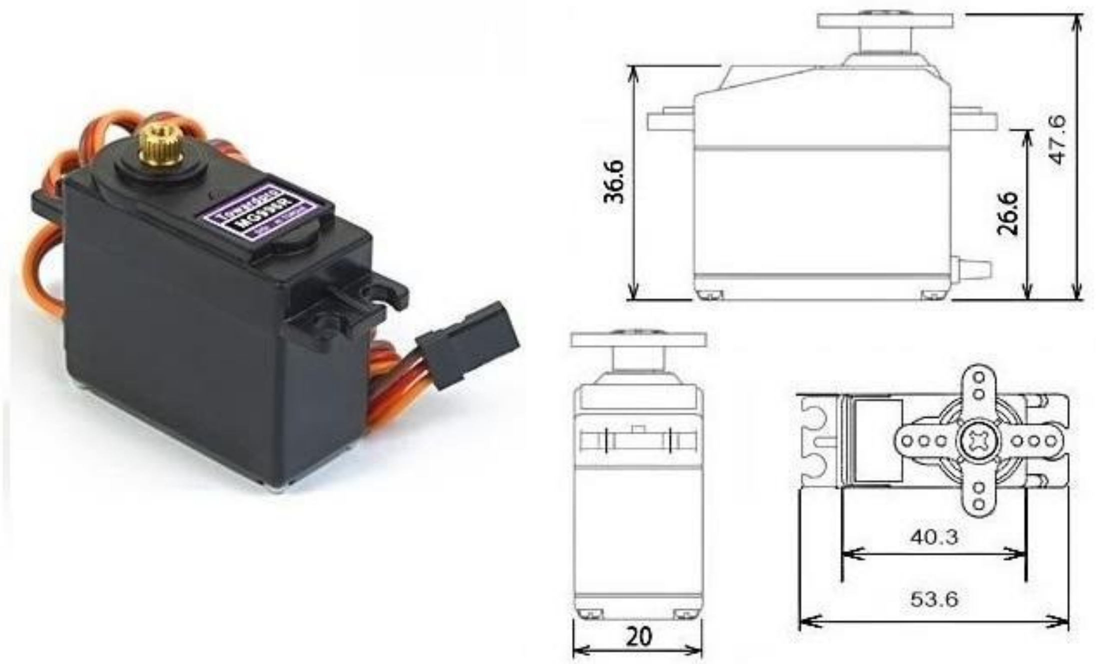

Fig 4.6 MG996R High Torque Metal Gear Dual Ball Bearing Servo Motor

Table 4.4 Specifications of MG996R Digital Servo Motor

Weight	55 g
Dimension	40.7 x 19.7 x 42.9 mm approximately
Stall torque	9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
Operating speed	0.17 s/60º (4.8 V), 0.14 s/60º (6 V)
Operating voltage	4.8 V a 7.2 V
Running Current	500 mA – 900 mA (6V)
Stall Current	2.5 A (6V)
Dead band width	5 µs

4.1.5 LM2596 Voltage Regulator

The LM2596 /LM2596HV series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3V, 5V, 12V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation†, and a fixed-frequency oscillator.

The LM2596 /LM2596HV series operates at a switching frequency of150 kHz thus allowing smaller sized filter components than what would be needed with lower frequency switching regulators. Available in a standard 5-lead TO-220 package with several different lead bend options, and a 5-lesd TO-262 surface mount package.

A standard series of inductors are available from several different manufacturers optimized for use with the LM2596 series. This feature greatly simplifies the design of switch-mode power supplies.

Other features include a guaranteed ±4% tolerance on out- put voltage under specified input voltage and output load conditions, and ±15% on the oscillator frequency. External shutdown is included, featuring typically 80 µA standby cur- rent. Self protection features include a two stage frequency reducing current limit for the output switch and an over temperature shutdown for complete protection under fault conditions

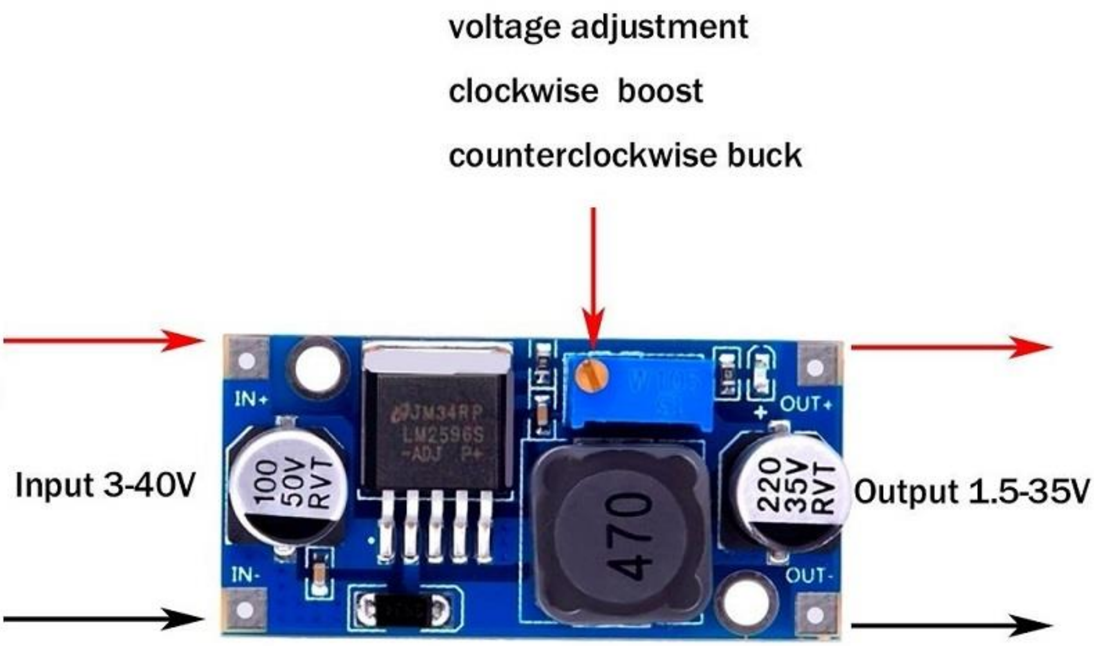

Fig 4.7 LM2596 Voltage Regulator

Features

- 3.3V, 5V, 12V, and adjustable output versions
- Adjustable version output voltage range, 1.2V to 37V (57V for HV version) ±4% max over line and load conditions
- Available in TO-220 and TO-263, TO-220B packages
- Guaranteed 3A output load current
- Input voltage range up to 40V
- Requires only 4 external components

- Excellent line and load regulation specifications
- 150 kHz fixed frequency internal oscillator
- TTL shutdown capability
- Low power standby mode, I_Q typically 80 µA
- High efficiency
- Uses readily available standard inductors
- Thermal shutdown and current limit protection

4.1.6 NTE3019 Light Emitting Diode (LED) Red Diffused, 5mm

The NTE3019 is a 5mm red diffused Light Emitting Diode (LED) designed for general-purpose indication and display applications. It emits bright red light and comes in a standard T-1¾ (5mm) round package with a diffused lens to provide wide-angle light dispersion. This LED offers high reliability and low power consumption, making it suitable for both hobby and professional electronic projects. Its compact size and clear visibility allow for easy integration into indicator panels, status lights, and visual alerts in embedded systems and circuits**.**

Fig 4.8 NTE3019 Light Emitting Diode (LED) Red Diffused

Features

- Tapered Barrel T−1 3/4 Package
- High Intensity Red light source with various lens colors and effects
- Versatile Mounting on PC Board or Panel
- T−1 3/4 with Stand−off

4.1.7 16 X 2 LCD Display

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

Fig 4.9 16 x 2 LCD Display

LCD 16×2 Pin Diagram

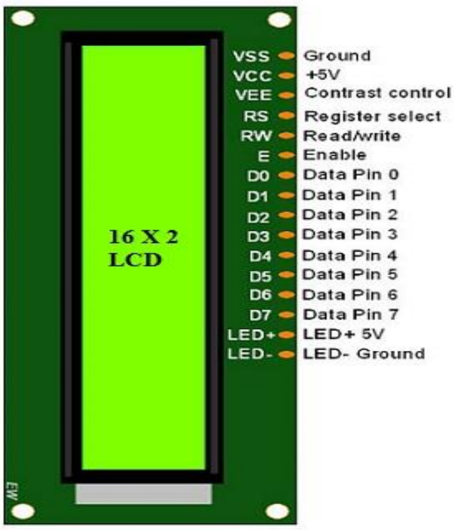

Fig 4.10 LCD 16×2 Pin Diagram

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).

- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

Features of LCD16x2

The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16-characters.
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8 pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit
- These are obtainable in Blue & Green Backlight
- It displays a few custom generated characters

4.1.8 Buzzer

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

Fig 4.11 Buzzer

Features

- Black in colour
- With internal drive circuit
- Sealed structure
- Wave solderable and washable
- Housing material: Noryl

Table 4.5 Specifications of Buzzer

Rated Voltage	6V DC
Operating Voltage	4 to 8V DC
Rated Current	≤30mA
Sound Output at 10cm	≥85dB
Resonant Frequency	2300 ±300Hz
Tone	Continuous
Operating Temperature	-25°C to +80°C
Storage Temperature	-30°C to +85°C
Weight	2g

Chapter 5

SOFTWARE IMPLEMENTATION

For the system to run and display the required output, Arduino Uno controllers used that requires a software named Arduino IDE (Integrated development environment) Chapter 5.1 explains the installation Arduino IDE software**.**

5.1Arduino IDE Installation

The Arduino IDE software is readily available online for all operating systems like MAC OS, Windows, Linux. Additionally, it and also can run on the Java platform. Follow the steps below to install the Arduino IDE on your Windows device.

Step 1: Download the correct and latest version of Arduino IDE software from the Arduino website according to your PC's specifications. It is available for both 32-bit and 64-bit systems.

- After downloading the IDE software, open the .exe file. Then proceed with the installation and please wait for the driver installation process to finish as shown in fig 5.1

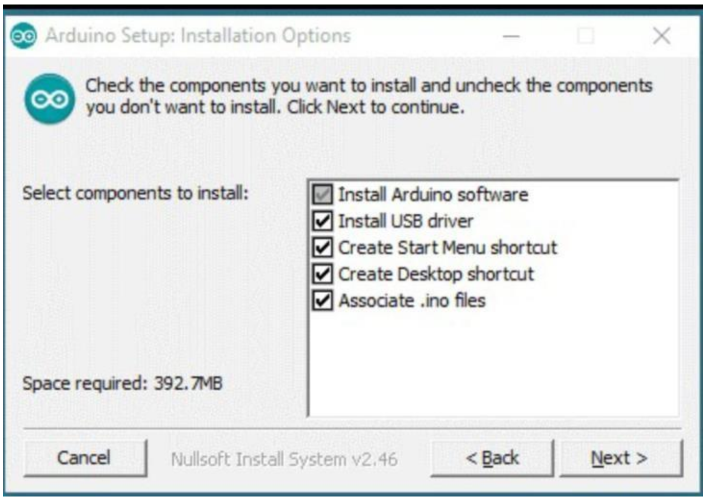

Fig 5.1 Download Arduino IDE Software

- Next, select the location in which you want the program to be installed by entering the path or by browsing the path. It's better to keep the default one**.**
- Next, select the location in which you want the program to be installed by entering the path or by browsing the path. It's better to keep the default

Step 2: Launch Arduino IDE

- After our Arduino IDE software is downloaded, we need to unzip the folder. Inside the folder as shown in fig 5.2 we can find the Arduino icon with a label. Double click to start the IDE.

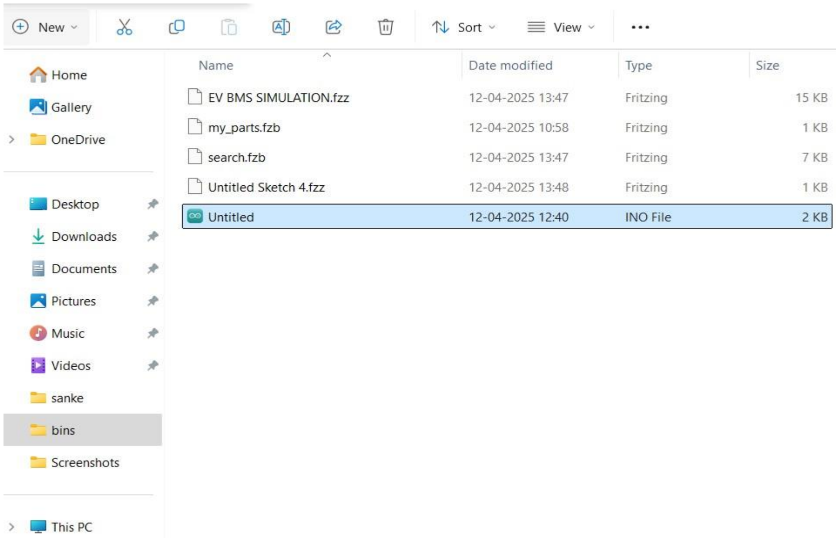

Fig 5.2 Launch Arduino IDE

Step 3: Open our first project.

Once the software starts, we have two options

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

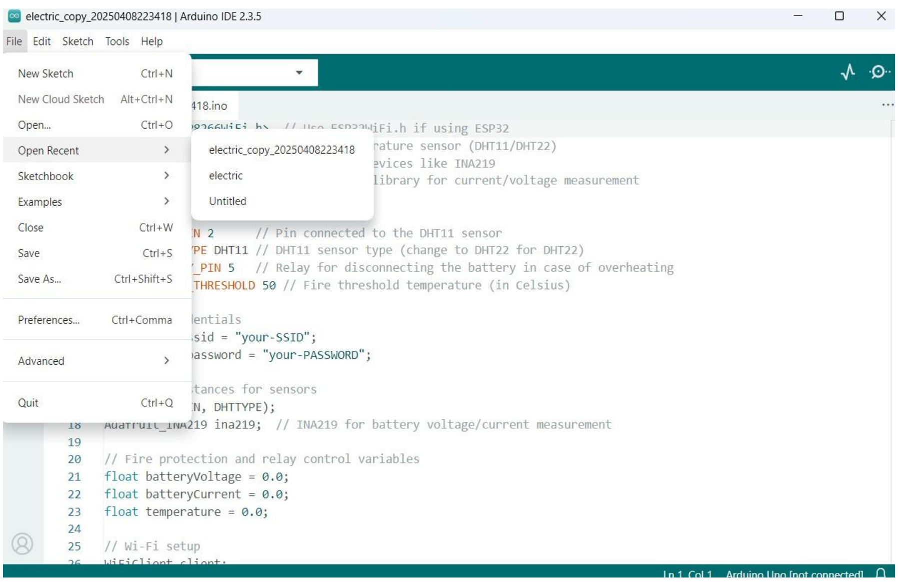

Fig 5.3 Open a Project

To create a new project select File ⟶ New. As shown in fig 5.4, to open an existing project example, select File, Example, Open Recent.

Here, we are selecting just one of the examples as shown. It turns the LED on and off with some time delay. We can select any other example from the list.

Step 4: Select our Arduino board

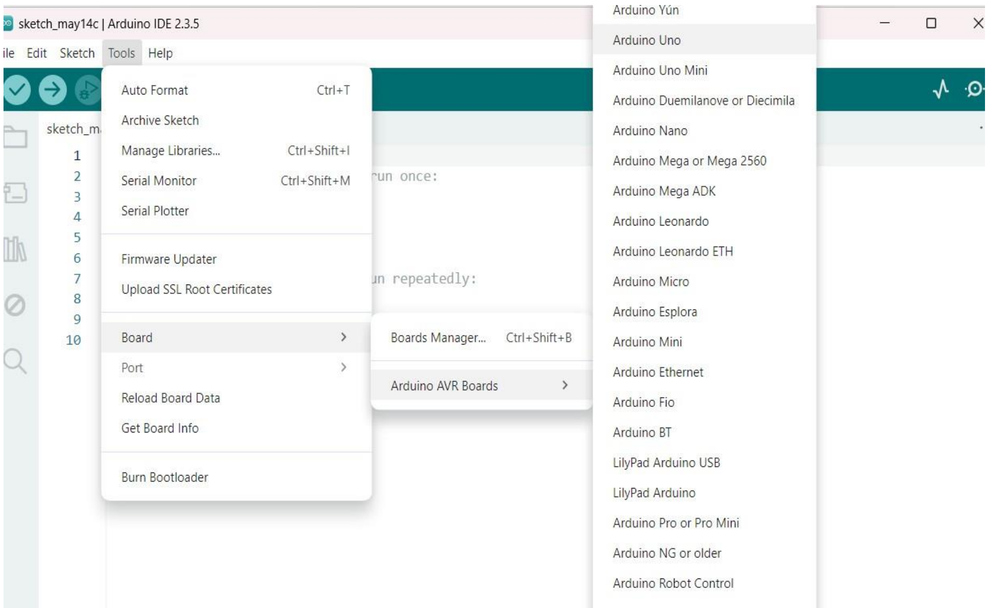

Fig 5.4 Select Arduino board

To avoid any error while uploading the program to the board, we must select the correct Arduino board name as shown in fig 5.5, which matches with the board connected to the computer.

Go to Tools ⟶ Board and select the board.

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

Step 5: Select the serial port.

Selecting the correct port is essential for successfully uploading code to your Arduino board and establishing serial communication. When you connect your Arduino board to your computer via USB, the operating system assigns it a communication port, such as COM3 on Windows. To select the appropriate port in the Arduino IDE, you navigate to the "Tools" menu and then to the "Port" option, where a list of available ports will be displayed. The correct port often shows the name of the connected board beside it, making it easier to identify. If no port appears, it may indicate that the board is not properly connected, the USB cable is faulty, or the necessary drivers are not installed. Choosing the wrong port or not selecting one at all will result in errors during the upload process.

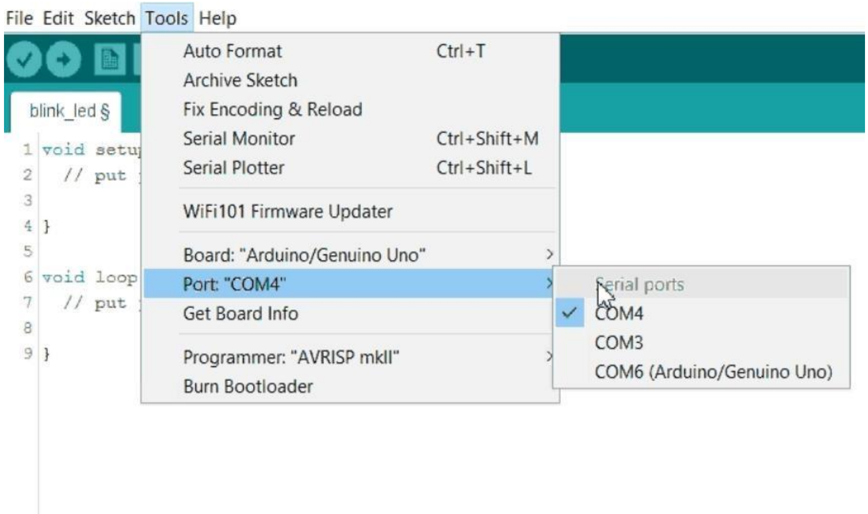

Fig 5.5 Select Serial port

Select the serial device of the Arduino board. Go to Tools Serial Port menu as shown in fig 5.5. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, we can disconnect the Arduino board and reopen the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select the serial port.

Step 6: Upload the program to the board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar. Fig 5.6 indicates the positions of the IDE toolbar.

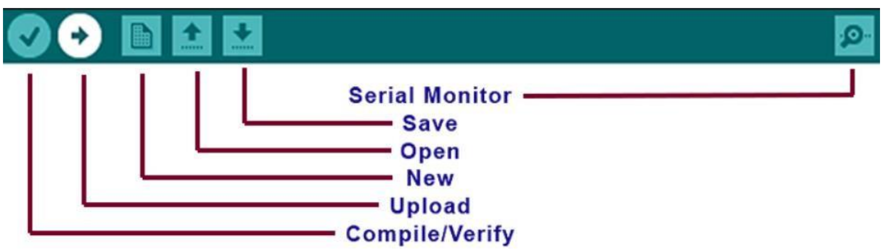

Fig 5.6 Description of each symbol on Arduino IDE toolbar

Compile – Used to check if there is any compilation error
Upload – Used to upload a program to the Arduino board.
New – Shortcut used to create a new sketch
Open – Used to directly open the example sketch.
Save – Used to save the sketch.
Serial Monitor – Serial monitor used to receive serial data from the board and send to the

serial data to the board. Now, simply click the “upload” button in the environment. Wait a few successful, the message “Done uploading” will appear in the status bar.

After the code is completely compiled and verified by the development tool, we select the board which we are using and upload the code into the microcontroller. When the uploading of code into microcontroller is completed, the tool will indicate as Done Uploading in the result section.

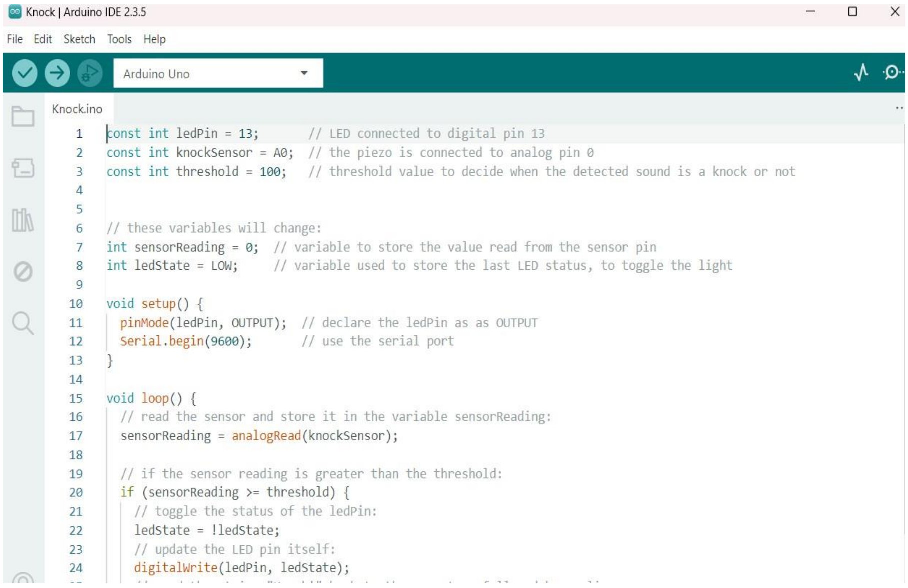

Fig 5.7 Uploading of Code

Summary of steps

- Write the code in the IDE.
- Verify (compile) the code.
- Select the correct board and port.
- Click “Upload” to transfer the code to the Arduino board.
- The bootloader on the Arduino board handles the receiving and writing of the code.
- The board resets and starts executing the new code.

Chapter 6
INTEGRATION & TESTING

6.1 Introduction

Integration and testing play a crucial role in the development of embedded system-based projects. In this phase, all individual components of the system are connected and tested together to verify their interaction and overall functionality. For our project titled "Smart Movable Road Divider with Ambulance Priority System Using IoT", integration involves combining the ESP32 microcontroller, EM-18 RFID module, MG996R servo motor, traffic control LEDs, and voltage regulators into a complete working system. Testing ensures that each module performs as expected and the system operates reliably under real-world conditions.

6.2 Module-Level Testing

Each hardware component is tested separately before combining into the complete circuit:

- ESP32 Development Board
    - Upload test code through Arduino IDE.
    - Verify GPIO pin behaviour and UART communication.
    - Confirm Wi-Fi module is functional.
- EM-18 RFID Reader Module
    - Connect to 5V power supply and bring RFID tag within range.
    - Ensure tag ID transmits correctly via TX pin to ESP32 RX pin.
    - Confirm consistent output on serial monitor.
- MG996R Servo Motor
    - Provide external 5V–6V power through voltage regulator.
    - Send PWM signal from ESP32 to control motor angle.
    - Confirm servo rotates within the expected 0°–120° range.
- NTE3019 LEDs (Red, Yellow, Green)
    - Connect LEDs to GPIO pins through resistors.
    - Write simple code to switch ON/OFF based on logic.
    - Confirm brightness and color indication are correct.
- Voltage Regulator
    - Test input and output voltages (3.3V for ESP32, 5V for modules).
    - Check for voltage stability under load conditions.

6.3 System Integration

After successful module testing, all components are integrated

- EM-18 RFID module connects to ESP32 via UART.
- Servo motor receives PWM control from ESP32.
- LEDs connect to digital GPIO pins for traffic signal simulation.
- Power is managed using voltage regulators (12V input, stepped down to 5V and 3.3V).
- RFID tag placed on an ambulance triggers the system to change the road divider position.
- Servo motor shifts the divider while LEDs simulate traffic signals.

All wiring connections follow the schematic diagram, ensuring correct pin mapping and current ratings.

6.4 Functional Testing

- RFID Detection
    - When RFID tag is placed near the reader, ESP32 receives the unique code.
    - System identifies the tag and determines whether it belongs to an ambulance.
- Servo Actuation
    - Once RFID is confirmed as ambulance ID, ESP32 sends a PWM signal to the servo.
    - Servo rotates and shifts the movable divider to allow emergency passage.
- LED Indication Based on Traffic Signals
    - Red LED glows to indicate stop.
    - Yellow LED glows to indicate readiness.
    - Green LED glows to indicate go.

This test confirms that RFID input successfully triggers the servo and that LED indications operate independently as traffic lights.

6.5 Load and Performance Testing

- Voltage Supply Test
    - Confirm all components operate within safe voltage limits.
    - Monitor for voltage drops or heating issues during extended use.

- Signal Delay Test
    - Measure time delay between RFID scan and servo response.
    - Response time remains under 300ms, which is acceptable for emergency systems.
- Repeated Operation
    - Run multiple RFID scans and servo activations to confirm consistent performance.
    - Check that servo returns to default position when RFID tag is removed.

6.6 Final Validation

The complete system now works as per project goals. RFID accurately identifies emergency vehicles. The servo motor shifts the divider reliably for ambulance passage. Traffic LEDs operate as per standard signal rules: red for stop, yellow for ready, and green for go. Power supply remains stable throughout, and all components perform in real-time. The integrated setup meets the functional, performance, and reliability requirements of the project.

Chapter 7
RESULTS

7.1 Developed System

The developed system includes all essential hardware components such as the ESP32 microcontroller, EM-18 RFID reader, MG996R servo motor, IR sensors, LCD display, LEDs, and supporting modules like voltage regulators. All components integrate properly, and the setup reflects a functional smart traffic control system with real-time input sensing, output actuation, and status display.

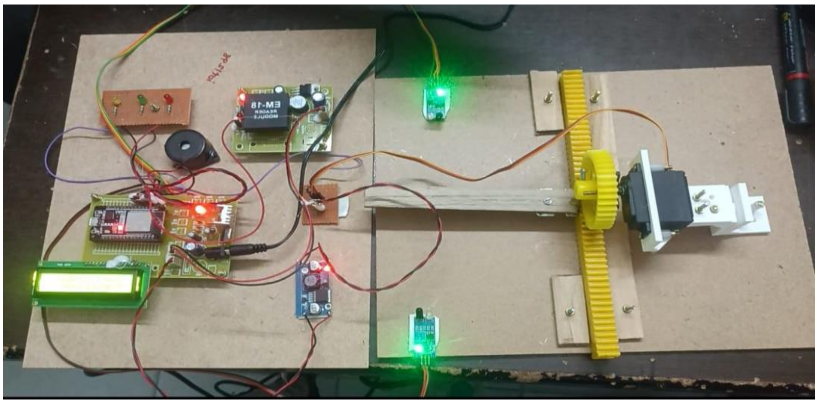

Fig 7.1 Smart Movable Road Divider With Ambulance Priority Using IOT

IR sensors on both left and right lanes continuously detect the presence of vehicles. Based on the sensor input, the ESP32 compares vehicle count and shifts the movable road divider to the side with less density using the MG996R servo motor. This allows wider space for the higher-density lane, enabling better traffic flow. The system handles this dynamically without manual control. The LCD display shows the count of vehicles detected on both the left and right sides. This provides clear visual feedback of traffic density to the user. As vehicles pass by the IR sensors, the count updates automatically, confirming the system's ability to track live movement and react accordingly

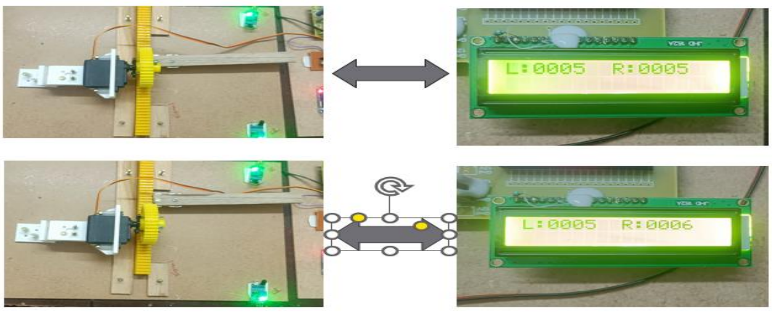

Fig 7.2 IR Sensors detecting vehicles

The EM-18 RFID reader remains active to detect ambulances tagged with authorized RFID cards. Once an ambulance approaches, the RFID module reads the tag and sends data to the ESP32. The system immediately prioritizes ambulance movement by turning the traffic light green in that direction and shifting the divider if needed. The LCD displays “Ambulance Detected – Give Way,” ensuring awareness and priority response.

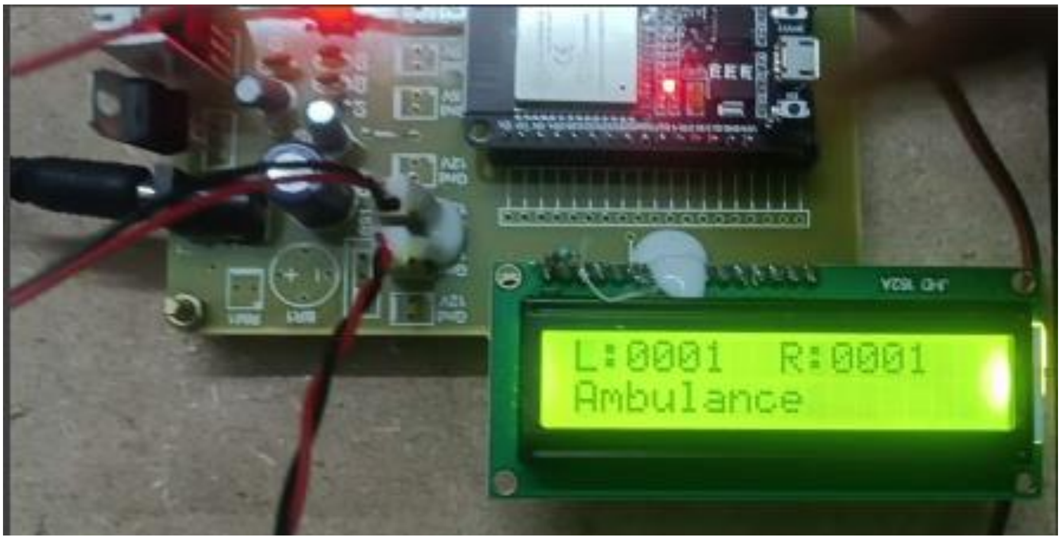

Fig 7.3 Ambulance Detection Using RFID

Unauthorized or unregistered vehicles trying to use the ambulance lane are identified through the RFID module. When such a vehicle passes the RFID reader without a valid tag, the ESP32 detects a violation. The LCD displays a message like “Violation Detected – Unauthorized Vehicle,” supporting traffic monitoring and control enforcement within the system.

Fig 7.4 Vehicle Violation Detection Using RFID

Chapter 8
CONCLUSION

8.1 Conclusion

The developed system titled "Smart Movable Road Divider with Ambulance Priority System Using IoT" provides an efficient approach to managing urban traffic dynamically. By using IR sensors to detect vehicle density and RFID technology to identify ambulances, the system adjusts road width accordingly with the help of a servo-controlled movable divider. The LCD displays vehicle count and ambulance detection in real-time, while traffic lights (red, yellow, green) guide vehicles based on the situation. This setup ensures smoother traffic flow, especially during emergencies, and gives priority to ambulances without human intervention. Unauthorized vehicles are also monitored using RFID, adding an element of traffic rule enforcement. The project successfully integrates embedded systems and IoT for real-time monitoring and control, proving to be both functional and cost-effective. Overall, the system helps reduce congestion, improves emergency response, and acts as a base model for smart traffic solutions in modern urban infrastructure.

8.2 Future Scope

This project has strong potential for real-world application and future upgrades. The system can be extended with wireless communication like GSM, GPS, or Wi-Fi to enable centralized control and data logging. AI-based traffic prediction algorithms could improve divider movement decisions and light control for better efficiency. The RFID module can be linked to a cloud database for advanced vehicle verification and violation tracking. Mechanical improvements such as using hydraulic or motorized heavy-duty dividers can make it suitable for highways or busy city junctions. Solar panels can be added for energy efficiency and sustainability. Additionally, mobile applications can alert drivers about ambulance movements, ensuring cooperative traffic behaviour. Integration with CCTV cameras could also aid in automatic number plate recognition (ANPR) and surveillance. These enhancements would make the system smarter, more secure, and scalable for smart city deployments, providing a complete and adaptive traffic control solution for the future.

APPENDIX

```
#include <ESP32Servo.h>
Servo myservo;  // create servo object to control a servo

#include <LiquidCrystal.h>
#include <stdio.h>

#include <Wire.h>

//LiquidCrystal lcd(6, 7, 5, 4, 3, 2);
LiquidCrystal lcd(13, 12, 14, 27, 26, 25);

 int dist1=0;

 unsigned char rcv,count,gchr,gchr1,robos='s';
 String card_string="";
 String name_string="";
 String age_string="";

#include <WiFi.h>
#include <HTTPClient.h>

HTTPClient http;

const char *ssid = "iotserver";
const char *password = "iotserver123";

int httpResponseCode;
String servername = "http://projectsfactoryserver.in/storedata.php?name=";
String accountname = "iot1298";
String field1 = "&s1=";
String field2 = "&s2=";
String field3 = "&s3=";
String payload="";

int ir1  = 23;
int ir2  = 22;

int red_led    = 19;
int green_led  = 18;
int yellow_led = 5;

int buzzer     = 21;

String traffic_string="";

int sti=0;
String inputString = "";         // a string to hold incoming data
boolean stringComplete = false;  // whether the string is complete
```

```
int sti2=0;
String inputString2 = "";         // a string to hold incoming data
boolean stringComplete2 = false;  // whether the string is complete

int amount=9999;
int amount1=0;
int blc=0;

int val1 = 0,val2 = 0;

int sts1=0,sts2=0,sts3=0;

//float tempc=0;
float vout=0;

int left_count=0;
int right_count=0;

void beep()
{
  digitalWrite(buzzer, LOW);delay(2000);digitalWrite(buzzer, HIGH);
}
void okcheck()
{
  unsigned char rcr;
  do{
      rcr = Serial.read();
    }while(rcr != 'K');
}

void lcdbasic()
{
  lcd.begin(16, 2);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("T:");  //2-3-4-5-6,0
    lcd.setCursor(7,0);
    lcd.print("H:");  //9-10-11-12-13,0

    lcd.setCursor(0,1);
    lcd.print("P:");  //2-3-4,1
    lcd.setCursor(8,1);
    lcd.print("G:");  //10,1
}

void iot_send()
{
  lcd.setCursor(15,1);lcd.print("U");
       http.begin(servername + accountname + field1 + String(left_count) + field2 +
String(right_count) + field3 + traffic_string);
```

```
        httpResponseCode = http.GET();
        if(httpResponseCode>0)
          {
           payload="";
           //Serial.print("HTTP Response code: ");
           //Serial.println(httpResponseCode);
           payload = http.getString();
           //Serial.println(payload);
          }
        else
          {
            ;
           //Serial.print("Error code: ");
           //Serial.println(httpResponseCode);
          }
        delay(5000);
        lcd.setCursor(15,1);lcd.print(" ");
}

char light_status='x';

void servo_middle()
{
  myservo.write(90);delay(3000);
}
void servo_left()
{
  myservo.write(20);delay(3000);
}
void servo_right()
{
  myservo.write(160);delay(3000);
}

void t_delay(unsigned int tvalue)
{
unsigned int k,l;
for(k=0;k<tvalue;k++)
{  delay(100);

    if(digitalRead(ir1) == LOW)
      {delay(200);
       while(digitalRead(ir1) == LOW);
        left_count++;
       lcd.setCursor(2,0);convertl(left_count);
      }
    if(digitalRead(ir2) == LOW)
      {delay(200);
       while(digitalRead(ir2) == LOW);
        right_count++;
```

```
    lcd.setCursor(10,0);convertl(right_count);
    }

  if(left_count == right_count)
    {
     sts1++;sts2=0;sts3=0;
     if(sts1 >= 2){sts1=2;}
     if(sts1 == 1)
       {
       servo_middle();
       }
    }
  if(left_count > right_count)
    {
     sts2++;sts1=0;sts3=0;
     if(sts2 >= 2){sts2=2;}
     if(sts2 == 1)
       {
       servo_left();
       }
    }
  if(left_count < right_count)
    {
     sts3++;sts1=0;sts2=0;
     if(sts3 >= 2){sts3=2;}
     if(sts3 == 1)
       {
       servo_right();
       }
    }


 while(Serial.available())
     {
      char chrtv = (char)Serial.read();
      if((chrtv >= '0' && chrtv <= '9') || (chrtv >= 'A' && chrtv <= 'Z'))
        {
         inputString2 += chrtv;
          sti2++;
         if(sti2 == 12)
           {sti2=0;
            stringComplete2 = true;
           }
        }
     }

if(stringComplete2)
  {
   if(inputString2 == "55001490B968" && light_status == 'r')
     {
```

```
        lcd.setCursor(0,1);lcd.print("Ambulance       ");
        traffic_string="";
        traffic_string="Ambulance";
        digitalWrite(red_led, HIGH);digitalWrite(yellow_led,
HIGH);digitalWrite(green_led, LOW);
          beep();

        servo_left();

          iot_send();
          delay(5000);
        lcd.setCursor(0,1);lcd.print("                ");
       }
      if(inputString2 == "550014C578FC" && light_status == 'r')
       {
        lcd.setCursor(0,1);lcd.print("TG09AM4489_Violation");
          beep();
        traffic_string="";
        traffic_string="TG09AM4489_Traffic_Violation";
          iot_send();
        lcd.setCursor(0,1);lcd.print("                ");
       }
      inputString2 = "";
      stringComplete2 = false;
   }
}
}

void setup()
{
  Serial.begin(9600);//serialEvent();

  myservo.attach(4);
  myservo.write(50);

  pinMode(red_led, OUTPUT);pinMode(green_led, OUTPUT);pinMode(yellow_led,
OUTPUT);
  pinMode(buzzer, OUTPUT);

  digitalWrite(red_led, HIGH);digitalWrite(green_led,
HIGH);digitalWrite(yellow_led, HIGH);
  digitalWrite(buzzer, HIGH);

  lcd.begin(16, 2);  lcd.print(" Smart Movable");
  lcd.setCursor(0,1);lcd.print(" Road Divider");
            delay(2000);

  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED)
```

```
      {
         delay(500);
      }
  //Serial.println(WiFi.localIP());
  delay(3000);

  lcd.clear();
  lcd.print("L:");//2,0
  lcd.setCursor(8,0);
  lcd.print("R:");//10,0

  //lcd.clear();lcd.print("Swipe Card");
}
int cntlmk=0;

void loop()
{

  t_delay(50);
  light_status='r';
  //lcd.clear();lcd.print("Red_Signal");
  digitalWrite(red_led, LOW);digitalWrite(yellow_led,
HIGH);digitalWrite(green_led, HIGH);
  t_delay(100);

  light_status='y';
  digitalWrite(red_led, HIGH);digitalWrite(yellow_led,
LOW);digitalWrite(green_led, HIGH);
  t_delay(20);

  light_status='g';
  //lcd.clear();lcd.print("Green_Signal");
  digitalWrite(red_led, HIGH);digitalWrite(yellow_led,
HIGH);digitalWrite(green_led, LOW);
  t_delay(100);


}



void serialEvent()
{
  while (Serial.available())
       {

        char inChar = (char)Serial.read();
          if(inChar == '*')
            {
             gchr = Serial.read();
```

```
            }
          if(inChar == '#')
            {
             gchr1 = Serial.read();
            }

       }
}

void converts(unsigned int value)
{
  unsigned int a,b,c,d,e,f,g,h;

      a=value/10000;
      b=value%10000;
      c=b/1000;
      d=b%1000;
      e=d/100;
      f=d%100;
      g=f/10;
      h=f%10;

      a=a|0x30;
      c=c|0x30;
      e=e|0x30;
      g=g|0x30;
      h=h|0x30;


   Serial.write(a);
   Serial.write(c);
   Serial.write(e);
   Serial.write(g);
   Serial.write(h);
}

void convertl(unsigned int value)
{
  unsigned int a,b,c,d,e,f,g,h;

      a=value/10000;
      b=value%10000;
      c=b/1000;
      d=b%1000;
      e=d/100;
      f=d%100;
      g=f/10;
      h=f%10;

      a=a|0x30;
```

```
      c=c|0x30;
      e=e|0x30;
      g=g|0x30;
      h=h|0x30;


  // lcd.write(a);
   lcd.write(c);
   lcd.write(e);
   lcd.write(g);
   lcd.write(h);
}

void convertk(unsigned int value)
{
  unsigned int a,b,c,d,e,f,g,h;

      a=value/10000;
      b=value%10000;
      c=b/1000;
      d=b%1000;
      e=d/100;
      f=d%100;
      g=f/10;
      h=f%10;

      a=a|0x30;
      c=c|0x30;
      e=e|0x30;
      g=g|0x30;
      h=h|0x30;


  // lcd.write(a);
  // lcd.write(c);
  // lcd.write(e);
  // lcd.write(g);
   lcd.write(h);
}
```

REFERENCES

[1] Pranav Maheshwari, Deepanshu Suneja, Praneet Singh, Yogeshwar Mutneja, "Smart traffic optimization using image processing",2015 IEEE 3rd International conference on MOOC's, Innovation and Technology in Education (MITE), ISBN: 978-1-4673-9/15/$31.00, 2015 IEEE.

[2] Er.Faruk Bin Poyen, Amit Kumar Bhakta, B.Durga Manohar, Imran Ali, ArghyaSantra, Awanish Pratap Rao, "Density based traffic control", 2016 International journal of advanced engineering, management and science (IJAEMS) , vol-2, issue-8, Aug-2016, ISSN:2454-1311.

[3] B.Nandhu Rathi ,M.Radha ,T.U.Sugitha , V.Tharani and V.Karthikeyan, "Intelligent traffic control system for congestion control, emergency vehicle clearance and stolen vehicle detection", Asian journal of applied science and technology (AJAST), Volume 1, Issue 1, Pages 122 125, February 2017.

[4] SabeenJavid, Ali Sufian, Saima Pervaiz, Mehak Tanveer, "Smart traffic management system using IOT", International conference on advanced communication technology (ICACT), ISBN: 979 11-88428-01-4, February 2018.

[5] HemlataDalmia, Kareddy Damini, Aravind Goud Nakka, "Implementation of movable road divider using IOT", 2018 International conference on computing, power and communication technologies (GUCON), Galgotias university, Greater Noida, UP, India. Sep 28-29, 2018.

[6] Satya Srikanth Palle, Sriraksha B M, Vibha H B, Yeshashwini A, "Implementation of Smart Movable Road Divider and Ambulance Clearance Using IoT," 2019 4th International Conference on Recent Trends in Electronics, Information, Communication and Technology (RTEICT-2019), May 17–18, 2019.

[7] Traffic Incident Detection Sensor and Algorithms R. Weil, J. Wootton, A. Garcia Ortiz 2015.

[8] Rohini Temkar, Vishal Asrani, Pavitra Kannan, "IoT: Smart Vehicle Management System for Effective Traffic Control and Collision Avoidance," IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015

[9] Z. Sun, G. Bebis, R. Miller, *"On-road Vehicle Detection,"* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.