

 <p>ISSN NO. 2320-5407</p>	<p>Journal Homepage: - www.journalijar.com</p> <h2>INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)</h2> <p>Article DOI: 10.21474/IJAR01/22207 DOI URL: http://dx.doi.org/10.21474/IJAR01/22207</p>	
---	---	---

RESEARCH ARTICLE

SMART AMBULANCE DETECTION IN FOUR WAY JUNCTION WITH PRIORITY-BASED TRAFFIC CONTROL SYSTEM

Done Sandeep¹, Mrs. Nasreen², Darsi Vamsi¹, Shaik Inzamam¹ and Chilaka Bharath³

1.UG Student.

2.Assistant Professor (C), Department of ECE, RGUKT, Ongole, Andhra Pradesh, India.

3.UG Student, Department of CSE, RGUKT, Ongole, Andhra Pradesh, India.

Manuscript Info

Manuscript History

Received: 04 September 2025

Final Accepted: 06 October 2025

Published: November 2025

Key words:-

Emergency Vehicles, Emergency State, GPS Coordinates ,Immediate clearance, Priority checking, Micro controllers, Centralized Cloud server , Data Packet , Detection Zone

Abstract

In metropolitan areas, ambulances face critical delays due to heavy traffic and static signal systems, risking patient lives. To overcome this,a Smart Ambulance Priority Control System is proposed, featuring a centralized cloud-based model using GPS and live server data to track ambulance movement in real time. Each ambulance sends its latitudes and longitudes and emergency state no to the server, which prioritizes signals when multiple ambulances are detected. Traffic controllers fetch data, calculate distance, and automatically grant green signals to the prioritized ambulance. After passage, normal operation i.e smart traffic system resumes, ensuring faster response, reduced delays, and improved urban emergency traffic management.

"© 2025 by the Author(s). Published by IJAR under CC BY 4.0. Unrestricted use allowed with credit to the author."

Introduction

Traffic congestion has become a major concern in urban areas, particularly in developing cities with rapidly growing vehicle numbers. Among the most affected services are ambulances, which often face long delays at busy intersections. Such delays can be life-threatening during medical emergencies where every second counts. Conventional time-based traffic systems fail to respond dynamically to these situations, highlighting the urgent need for an intelligent and automated mechanism that ensures immediate clearance for emergency vehicles.[1] Over the years, several traffic management techniques have been introduced to overcome this issue, but most remain partially effective. The Green Corridor System enables the creation of a dedicated route for ambulances through manual coordination among multiple traffic stations. However, this approach is highly time-consuming and dependent on human communication, making it unsuitable for unexpected or high-frequency emergencies. The Smart Corridor with CCTV and IoT Sensors offers real-time monitoring and vehicle counting but lacks direct ambulance detection and integration with centralized emergency data. This makes automated responses difficult, especially when multiple emergency vehicles are operating simultaneously in different areas. On the other hand, Siren-Based Ambulance Identification depends on audio sensors to detect sirens.

However, this technique suffers from high noise interference limited range, and difficulty distinguishing between actual sirens and environmental sounds, making it unreliable in crowded city environments. Due to these limitations, there is a strong need for a more intelligent, reliable, and automated solution capable of real time ambulance

detection and dynamic signal control. To address these challenges, the proposed Smart Ambulance Detection and Priority-Based Traffic Control System introduces a centralized, GPS integrated approach that ensures seamless coordination between ambulances and traffic junctions. By transmitting live coordinates and emergency states to a central server, the system performs continuous tracking and intelligent priority assignment among multiple ambulances operating within the network. The micro controller units installed at each traffic pole communicate with the server to adjust signal timings automatically based on the detected priority. This eliminates manual intervention, reduces response time, and ensures that the ambulance facing the most critical emergency receives the right of way without delay. Furthermore, the system is scalable, allowing integration with IoT-based sensors and cloud data storage to enhance future performance, reliability, and adaptability in smart city environments.

LITERATURE SURVEY:

Nellore & Hancke (2016) proposed a vision-based, sensor-network approach to prioritise emergency vehicles (EVs) at signalised intersections by detecting EVs, estimating their distance to intersections, and coordinating time sensitive alerts to traffic controllers. The system combines video-based vehicle detection, distance estimation using multiple distance metrics (Euclidean, Manhattan, Canberra), local vehicle counting, and a priority MAC (PE-MAC) scheme in the sensor network to orchestrate signal preemption and reduce EV delays.[2]

In the paper “Ambulance Assistance for Emergency Services Using GPS Navigation” by Shantanu Sarkar, a system is described which enables hospitals to efficiently track their ambulances in real time. The primary objective of this work is to minimize the number of fatalities among critical patients by ensuring timely transportation to healthcare facilities. The integration of GPS technology allows hospitals to monitor ambulance locations continuously and take prompt action when emergencies occur. This approach significantly improves response efficiency, leading to reduced travel delays and quicker access to medical treatment. Overall, the system proves to be an effective solution for optimizing ambulance management and minimizing time consumption during emergency situations.[3]

Anu Philip et al.’s survey paper examines low-cost technological approaches for giving ambulances priority in urban traffic. The paper compiles and compares system architectures that combine mobile devices (typically Android apps), wireless communication (GSM/GPRS, sometimes MQTT or TCP/IP), and simple microcontroller-based traffic controllers (frequently Arduino UNO) to implement signal preemption or green-wave creation. The stated aim is pragmatic: identify affordable, easily deployable solutions for cities with constrained budgets so ambulance response time — and therefore mortality from delayed care — can be reduced.[4]

The paper titled “Traffic Congestion Detection and Alerting Ambulance Using IoT” presented an IoT-based approach that detects congestion and alerts ambulances by transmitting real-time data through cloud communication. The system employs GPS modules, ultrasonic sensors, and microcontrollers to monitor vehicle density and adjust signal timings based on traffic conditions. Ultrasonic sensors are used to detect the presence and movement of nearby vehicles, providing real-time data on traffic congestion levels. Although effective in identifying congestion and sharing updates with ambulances, this model primarily focuses on alert generation rather than automated ambulance prioritization or intersection-level traffic control.[5]

The paper "FIREBASE- OVERVIEW AND USAGE" by Pankaj Chougale, Vaibhav Yadav, and Dr. Anil Trimbakrao Gaikwad, discusses regarding the overview and utilization of the fire base as a central server for Data Handling the location updates [6].

Rezania Agramanisti Azdy and Febriyanti Darnis put forward a paper “Use of Haversine Formula in Finding Distance Between Temporary Shelters and Waste End Processing Sites” which discuss about the procedure followed to Calculate great-circle distance (shortest path on Earth's surface) . The Haversine Formula will Used in GPS and navigation systems. [7]

The paper "Advance Alert for Ambulance Pass by using IoT for Smart City" by Madhav Mishra, Seema Singh, Dr. Jayalekshmi K.R., and Dr. Taskeen Nadkar (2017) recommends an innovative approach to enhancing emergency medical services through the integration of Internet of Things (IoT) technologies. The authors propose a system wherein ambulance drivers utilize an Android application to request traffic signal controllers to turn green, facilitating unobstructed passage through intersections. This system employs a server-client architecture, with the ambulance acting as the client, transmitting real-time location data to a central server. The paper highlights the potential of IoT in transforming urban infrastructure to support critical services, contributing to the broader vision of smart cities.[8]

The paper titled “Smart Ambulance Traffic Control System” by Suren Krishnan, Rajan Thangaveloo, Shapi-ee Bin Abd Rahman, and Siva Raja Sindiramutty from Universiti Malaysia Sarawak planned an intelligent traffic management approach designed to minimize delays faced by ambulances during emergency situations. The authors emphasize the limitations of conventional traffic systems that operate on static signal cycles, which often hinder ambulance movement in congested areas. [9]

The paper “Design and Development of Green Corridor Using IoT” by Kumar R. and Mittal S. (2018) focuses on creating an IoT-based automated traffic control system to ensure uninterrupted movement for ambulances during emergencies. The system detects ambulances using sensors and GPS modules, then dynamically controls traffic lights to form a green corridor, significantly reducing response time and congestion.[10]

The book “Smart Sustainable Cities of the Future” by Simon Elias Bibri (2018) explores the integration of smart technologies and sustainability principles to design future urban environments. It emphasizes the role of data-driven systems, IoT, and urban informatics in enhancing environmental, social, and economic efficiency.[11]

UTILIZED COMPONENTS:-

ESP32 Microcontroller:

The ESP32 serves as the central communication and control hub in the proposed system. It gathers real-time GPS data from ambulances, uploads it to the cloud server via Wi-Fi, and retrieves traffic control commands sent back by the server. At each traffic junction, another ESP32 module receives these commands and adjusts the traffic lights accordingly. Its dual-core processing capability ensures fast and reliable operation, while its IoT connectivity supports scalability for multiple intersections and ambulances simultaneously.



Figure 1: Esp32 Micro Controller

GPS MODULE :

The project employs the Neo-6M GPS Module, a high-performance positioning device capable of providing accurate latitude and longitude coordinates in real time. This module communicates with the ESP32 microcontroller through serial communication (TX and RX pins) and continuously transmits the ambulance’s current location.. The Neo- 6M GPS operates on satellite-based navigation and provides high precision with minimal delay, even in dense urban environments. But in testing the prototype rather than generating the coordinates using gps we had generated random locations for testing and updates to the Central server.

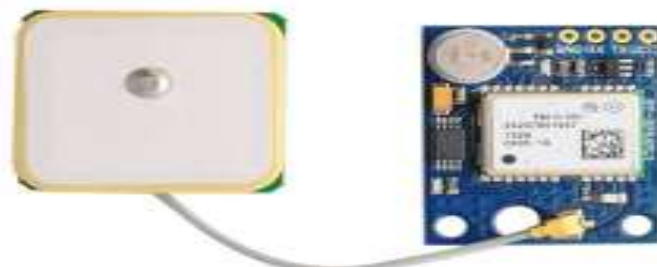


Figure 2: Gps Module (Neo-6)

RGB LED's and BREAD Board :

The breadboard setup provides a flexible prototype environment to test the coordination between cloud communication, GPS input, and traffic light control. Together, these components form an efficient and scalable prototype capable of demonstrating the effectiveness of real time ambulance prioritization in traffic management.

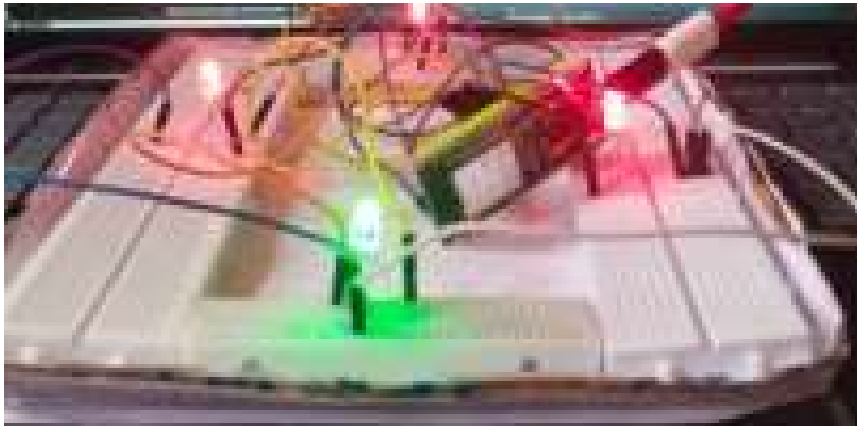


Figure 3:Utilization of Bread board and RGB LED'S for prototype

Fire Base:

Firebase was chosen due to its high reliability, real- time synchronization, and ease of integration with the ESP32 microcontroller through Wi-Fi connectivity. It provides an efficient and scalable medium for transmitting GPS coordinates, emergency states, and traffic control signals across multiple devices in real time.

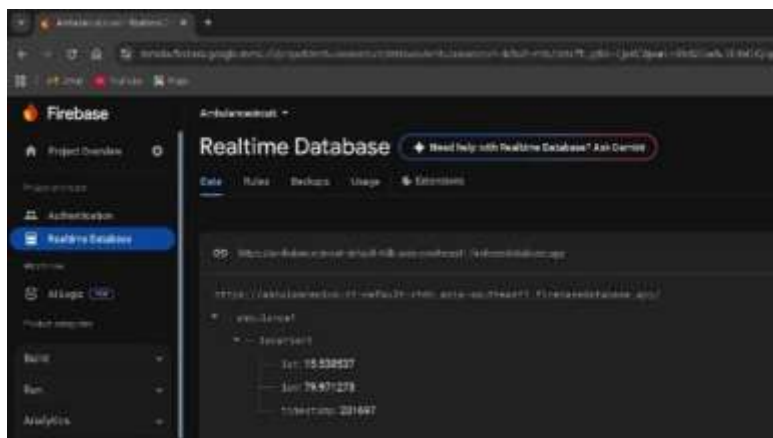


Figure 4:Picture represents the updating the of latitudes and longitudes to the Fire Base server

Arduino IDE:

It was used to write, compile, and upload programs to the ESP32 microcontroller, which acts as the central control unit for communication and traffic signal management. Through the Arduino IDE, all essential functions such as GPS data collection, Wi-Fi communication with the central server, and RGB signal control were effectively programmed and executed. The IDE provided a simple yet robust interface for integrating various libraries such as those for GPS (TinyGPS++), Wi-Fi, and cloud communication platforms like Firebase or ThingSpeak. Using the Serial Monitor, real-time data transmission between the ESP32, GPS module, and server was continuously observed and verified. Additionally, the Arduino IDE was utilized to generate output signals during the testing phase, allowing the RGB LEDs on the breadboard to simulate real-time traffic signal operations. This enabled performance validation and ensured the correct response of traffic lights based on ambulance priority before hardware-level deployment.

SYSTEM DESIGN AND ARCHITECTURAL OVERVIEW:-

The proposed Smart Ambulance Detection and Priority-Based Traffic Control System is architected as a three-tiered, interconnected framework. This design ensures automated, intelligent, and real-time traffic signal management for emergency vehicles.

The system's architecture is logically divided into three primary functional blocks:

1. Ambulance Identification and Data Update
2. Central Cloud Server and Data Handling
3. Detection and Signal Control

The overall view of total implementation from identification to signal conversion is represented in the form of pictorial Block diagram .And Let us go block by block for knowing regarding the working procedure.

Ambulance Identification and Data Update:-

This block serves as the system's data acquisition layer, responsible for collecting, packaging, and transmitting critical real-time information from the ambulance to the central cloud infrastructure. Its primary function is to transform the physical state of the emergency vehicle into a structured, machine-readable data stream that forms the foundational input for all subsequent decision- making processes.

Manual Input Interface:

The ambulance operator inputs the Emergency State Number (a discrete value, e.g., 1 for Critical/Code-3, 2 for Medium, 3 for Low urgency/Routine) via a dedicated, intuitive mobile or web- based application. This parameter is crucial as it defines the trip's priority level within the traffic management ecosystem. Protocols: The emergency states are mapped to standardized hospital or emergency service protocols (e.g., Code-3 for life-threatening situations). This prevents misuse and ensures that high priority is reserved for genuine emergencies . The interface allows the operator to dynamically update the emergency state during transit. For instance, if a patient's condition stabilizes, the priority can be downgraded to reduce the disruption to general traffic. The application requires user authentication to prevent unauthorized activation, ensuring the integrity of the priority system. Each login can be linked to a specific ambulance unit for accountability.

GPS Coordinate Generation:

An onboard GPS Module (e.g., a high-sensitivity GNSS receiver) continuously acquires the ambulance's geolocation, generating real-time latitude and longitude coordinates with an associated timestamp. Beyond basic coordinates, the GPS module often provides additional metrics such as speed, heading (direction of travel), and altitude. The heading is particularly valuable for predicting the ambulance's approach vector towards an intersection, allowing for more anticipatory signal control. The system is designed to handle inherent GPS inaccuracies (e.g., urban canyon effects) by employing sensor fusion techniques. The GPS is configured for a high update rate (e.g., 1 Hz or higher) to provide a near-continuous location feed, which is essential for accurate real-time tracking.

Data Packet Formation Unit:-

This software unit aggregates the Emergency State, precise timestamp, and enriched location data (latitude, longitude, speed, heading) into a structured Data Packet. The packet is typically structured in a lightweight, human-readable, and machine- parsable format like JSON (JavaScript Object Notation) or Protocol Buffers for higher efficiency. A sample JSON packet is illustrated below. The packet includes a unique ambulance_id and a timestamp for tracking, logging, and resolving conflicts. Checksums or digital signatures can be added to ensure the data has not been corrupted or tampered with during transmission.

Data Transmission Unit:

The consolidated data packet is transmitted to the Central Cloud Server via a robust and secure IoT communication protocol. QTT (Message Queuing Telemetry Transport) is highly favored over HTTP in such IoT scenarios due to its publish-subscribe model, low power consumption, and efficiency in handling high- latency or unreliable networks. The ambulance publishes its data to a topic (e.g., ambulance/AMB- 7284/status), and the cloud server subscribes to it. The transmission unit includes logic for handling network dropouts. If a cellular network connection is lost, data packets can be temporarily cached in a local buffer on the ambulance's device and transmitted once the connection is re-established.

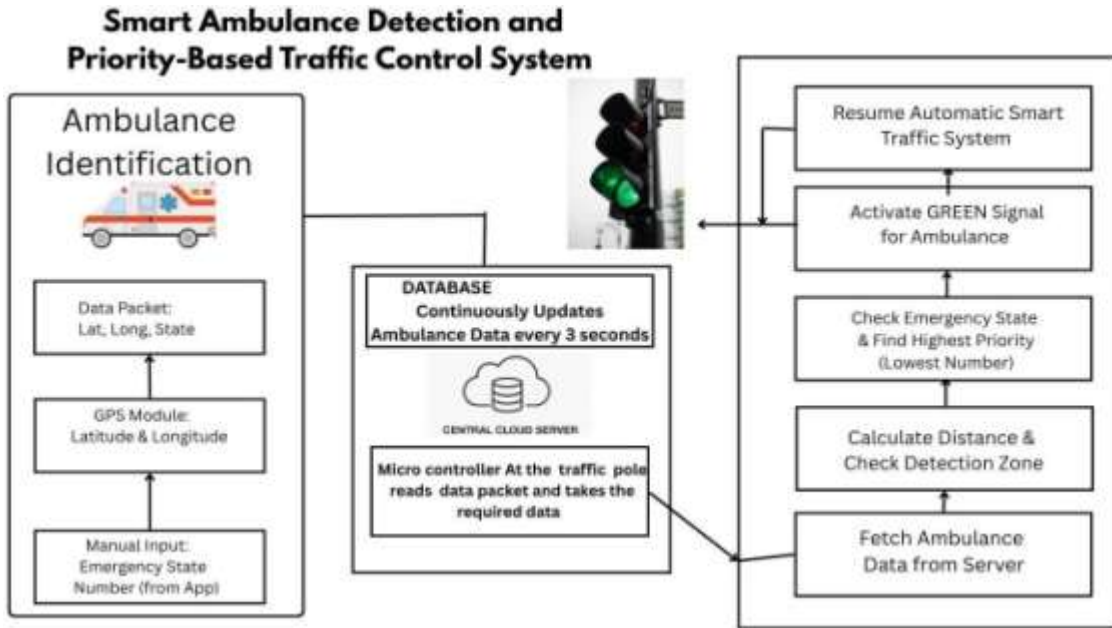


Figure 5: Block Diagram of Proposed Approach

```

json
{
  "ambulance_id": "AMB-7284",
  "timestamp": "2023-10-27T14:30:45Z",
  "location": {
    "latitude": 12.9716,
    "longitude": 77.5946,
    "speed": 65.5,
    "heading": 270
  },
  "emergency_state": 1,
  "metadata": {
    "version": "1.0"
  }
}
    
```

Figure 6: Sample example of JSON Packet ambulances are relevant to each specific intersection. Upon request

Central Cloud Server and Data Handling:-

This block functions as the intelligent communication backbone, bridging the gap between the mobile ambulances and the fixed infrastructure at traffic intersections. Its operation can be distilled into two primary, continuous processes: ingesting data from the ambulances and serving filtered data to the microcontrollers. It continuously aggregates real-time data streams from all active ambulances into a centralized, authoritative database. It then performs intelligent geospatial filtering to determine which from a traffic pole's microcontroller, it serves a customized data packet containing only nearby emergency vehicles. It can be explained as follows .

- A. Connection and Data Ingestion from Ambulance Identification
- B. Connection and Data Servicing to Microcontrollers

Connection and Data Ingestion from Ambulance Identification:-

This step establishes the uplink from the edge (ambulances) to the cloud, ensuring all emergency vehicle data is captured reliably and in real-time. The cloud server exposes a secure, high-availability API endpoint (e.g., a RESTful POST endpoint like <https://api.trafficsystem.net/ambulance/update>) or utilizes a publish-subscribe messaging protocol like MQTT. Each ambulance, from Block 1, acts as a client and continuously transmits its structured data packet (containing ID, location, speed, and emergency state) to this endpoint at a high frequency (e.g., every 15 seconds). Each data packet is first validated. The ambulance's credentials (e.g., a unique API key) are

authenticated to prevent spoofing. The data structure and values are checked for integrity. Upon successful validation, the packet is written to a cloud database (such as Google Firebase Realtime Database or Amazon DynamoDB). The data is stored under a unique node or document for each ambulance_id. The "write" operation is designed to be atomic and fast, overwriting the previous state to always reflect the latest known status of the vehicle. This creates a single, authoritative source of truth for the live state of all active emergencies.

Connection and Data Servicing to Microcontrollers:-

This step establishes the downlink from the cloud to the edge (traffic poles), providing each intersection with a customized, relevant view of the data. Each traffic pole's micro controller (e.g., ESP32) is configured as an API client. It periodically (e.g., every 10 seconds) sends a secure HTTP GET request to a separate cloud endpoint (e.g., [https://api.trafficsystem.net/intersection/JN-45/priority vehicles](https://api.trafficsystem.net/intersection/JN-45/priority%20vehicles)). This request includes the unique intersection_id and authentication credentials. Upon receiving a request from a specific intersection, the cloud server does not return the entire data set of all ambulances. Instead, it executes a geospatial query on the database. This query retrieves only the records of ambulances whose latest coordinates fall within a large pre-defined geofence (e.g., a 2 km radius) around the requesting intersection's fixed coordinates. The cloud server compiles the results of this query into a filtered and lightweight JSON payload and sends it back as the response to the microcontroller's request. This payload contains only the list of ambulances that are potential candidates for triggering a signal change at that specific junction, drastically reducing the data processing burden on the micro controller.

Detection and Signal Control Block:-

This block functions as the autonomous decision-making and actuation unit of the system, physically deployed at each traffic intersection. It processes incoming data to execute precise traffic signal commands, ensuring the creation of a dynamic "green corridor." The internal operation can be mapped to a sequential block diagram flow, as detailed below. Architecturally, it operates on the edge, performing real-time computations locally to minimize latency between detection and action. The block is designed with a state machine logic, transitioning through distinct phases: idle, detection, arbitration, actuation, and reset. This ensures deterministic behavior and reliable handling of complex, real-world scenarios such as multiple simultaneous emergency requests. By processing geospatial data and priority rules directly on the intersection's microcontroller, it eliminates dependency on continuous cloud connectivity for critical signal switching functions. The block's output is a direct physical control signal, interfacing with the traffic light hardware to enforce the computed priority decision.

The following stages detail the internal data flow and functional components:

- A. Input Data Acquisition Unit
- B. Proximity Calculation & Detection Zone Module
- C. Priority Arbitration and Conflict Resolution Unit
- D. Signal Actuation Controller (Traffic Light Interface)
- E. System Reset and Monitoring Loop

Input Data Acquisition Unit:-

The Input is a Filtered data packet from the Central Cloud Server. This unit receives and parses the structured data (e.g., JSON) to extract critical parameters: ambulance_id, latitude, longitude, and emergency_state for all ambulances deemed relevant by the cloud. A clean, localized dataset ready for geometric processing is available from this Internal block.

Proximity Calculation & Detection Zone Module:-

A Localized dataset from the Data Acquisition Unit and the intersection's fixed GPS coordinates are placed into the block as input. This module contains the Distance Calculation Algorithm (e.g., Haversine formula). It computes the real-time distance to each ambulance. The result is passed to a Comparator that checks it against a pre-set Detection Zone Threshold (e.g., 500 meters). The comparator produces a binary output: 1 if the ambulance is within the zone (APPROACHING), and 0 if it is not. A list of ambulances that have triggered the detection zone alert is the obtained output to the further blocks.

Priority Arbitration and Conflict Resolution Unit:-

The list of "APPROACHING" ambulances with their emergency state are read as input for the block. This unit is the core decision logic. It functions as a Priority Encoder. It scans all active requests and selects the one with the lowest

emergency state number (highest urgency). In case of a tie, a secondary Tie-Breaker Algorithm (e.g., selecting the closest ambulance) is invoked. A single, definitive command containing the ambulance_id and the direction that has been granted priority is the produced output from this block \

The priority command from the Arbitration Unit is taken as input . This digital logic block maps the approved ambulance's direction to the corresponding physical traffic signals. It sends GPIO (General-Purpose Input/Output) commands to the output hardware. As producing output it will makes either GPIO_HIGH signal to the Green LED of the ambulance's lane and GPIO_HIGH signal to the Red LEDs of all other conflicting lanes.

System Reset and Monitoring Loop:-

Subsequent GPS data updates for the serviced ambulance from the cloud are taken as input for the block . A Monitoring Process continuously tracks the serviced ambulance's location. Once the calculated distance exceeds the detection zone threshold (indicating the junction has been cleared), a Timer or Logic Circuit is triggered. Then a System Reset Command that de- allocates the priority state and returns control to the standard traffic signal controller, resuming normal operation

RESULTS:-

The prototype was developed using the components described above. To test the proposed algorithm, random latitude and longitude values were generated in the Arduino IDE by adding a few lines of code. Simultaneously, random emergency state numbers were generated to assist in priority checking when more than two ambulances are detected within the same zone simultaneously. The prototype successfully controlled the traffic signals, granting green signals to the ambulances based on their emergency state numbers, allowing passage one by one. After the ambulances exited the detection zone, the microcontroller automatically reset the system to normal smart traffic operation without any manual intervention. The generated outputs of the prototype are provided below for reference.

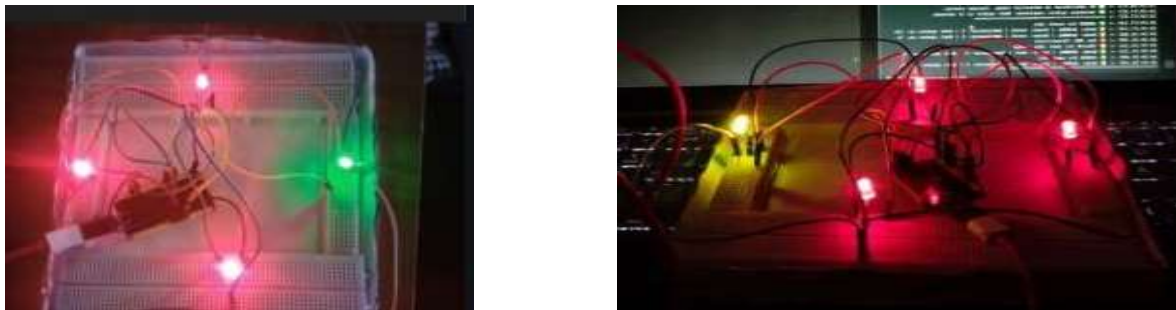


Figure 7: Go with normal mode when no ambulance is detected

```
READING DATA FROM FIREBASE (15s Update)...
Generated 1 ambulances:
AMB1 -> Road: east | Emergency: 2 (1=Most Critical)
Firebase updated successfully with current data
Scanning for ambulances in range:
  AMB1 added to queue (within 100m)
PRIORITY QUEUE CREATED (1 ambulances):
1. AMB1 | Road: east | Emergency: 2 | Distance: 86.4m | Score: 16
ACTIVATING PRIORITY 1 of 1
Blinking GREEN for east road
AMBULANCE 1 DETAILS:
ID: AMB1
Road: east
Distance: 86.4m
Emergency Scale: 2 (1=Most Critical)
Priority: 1 of 1
Duration: 5 seconds
-----
Database update completed. Next update in 15 seconds.
All priority ambulances served. Returning to normal cycle.
```

Figure 8:Detection of Single ambulance when eneterd into detection zone

```
READING DATA FROM FIREBASE (15s Update)...
Generated 2 ambulances:
AMB1 → Road: west | Emergency: 4 (1=Most Critical)
AMB2 → Road: south | Emergency: 2 (1=Most Critical)
Firebase updated successfully with current data
Scanning for ambulances in range:
  AMB1 added to queue (Within 50m)
  AMB2 out of range (105.7m > 100m)
PRIORITY QUEUE CREATED (1 ambulances):
1. AMB1 | Road: west | Emergency: 4 | Distance: 8.7m | Score: 28
ACTIVATING PRIORITY 1 of 1
Blinking GREEN for west road
AMBULANCE 1 DETAILS:
ID: AMB1
Road: west
Distance: 8.7m
Emergency Scale: 4 (1=Most Critical)
Priority: 1 of 1
Duration: 5 seconds

Database update completed. Next update in 15 seconds.

Mode: PRIORITY_QUEUE (1/1) | Green: west | Ambulances: 2 | Next update in:
Priority 1 of 1: 0s remaining
All priority ambulances served. Returning to normal cycle.
```

Figure 9: Detection of two ambulances at a time

```
https://ambulance-75d97-default-rtdb.firebaseio.com/.firebase/database/app/
- ambulances
  - amb1
    active: true
    distance: 0
    emergencyScale: 5
    id: "AMB1"
    lat: 15.53866
    lon: 79.97124
    road: "west"
    timestamp: 417040
  - amb2
    active: true
    distance: 0
    emergencyScale: 5
    id: "AMB2"
    lat: 15.53845
    lon: 79.97178
    road: "west"
    timestamp: 415306
- trafficSignal
```

Figure 10: Updation of Latitudes and Longitudes along

```
scanning for ambulances in range:
  AMB1 added to queue (Within 50m)
  AMB2 added to queue (Within 100m)
Sorting priority queue...
PRIORITY QUEUE CREATED (2 ambulances):
1. AMB2 | Road: north | Emergency: 2 | Distance: 23.4m | Score: 14
2. AMB1 | Road: east | Emergency: 5 | Distance: 46.2m | Score: 37
ACTIVATING PRIORITY 1 of 2
Blinking GREEN for north road
AMBULANCE 1 DETAILS:
ID: AMB2
Road: north
Distance: 23.4m
Emergency Scale: 2 (1=Most Critical)
Priority: 1 of 2
Duration: 5 seconds

Database update completed. Next update in 15 seconds.

ACTIVATING PRIORITY 2 of 2
Blinking GREEN for east road
AMBULANCE 2 DETAILS:
ID: AMB1
Road: east
Distance: 46.2m
Emergency Scale: 5 (1=Most Critical)
Priority: 2 of 2
Duration: 5 seconds

Mode: PRIORITY_QUEUE (2/2) | Green: east | Ambulances: 2 | Next update in: 1s
Priority 2 of 2: 5s remaining
```

Figure 11: Enables priority checking after entering the detection zone

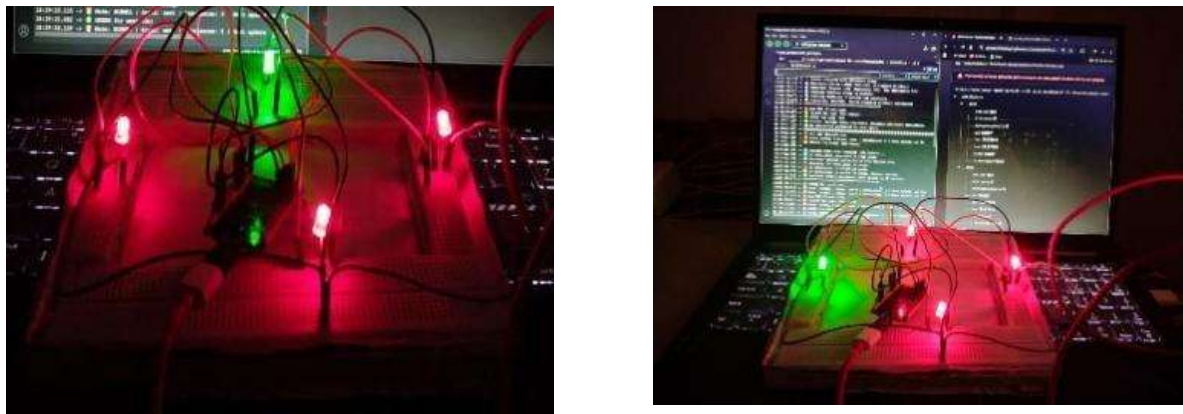


Figure 12: Changing of signal based on the prioritization of two ambulances one by one

Conclusion

In this work, a smart traffic management system for prioritizing ambulance movement was successfully designed and implemented. The proposed prototype demonstrated the capability to detect multiple ambulances within a defined zone, calculate priority based on emergency state numbers, and control traffic signals to allow safe and timely passage. The integration of location data and emergency priority ensured that multiple ambulances could be managed efficiently without manual intervention, highlighting the potential of automated systems in improving emergency response times in urban traffic environments. Future enhancements could include integrating real-time GPS tracking from ambulances, expanding the detection zones across multiple intersections, and incorporating predictive algorithms to anticipate ambulance routes for even faster response.

Future Scope:-

Potential for the Future of Smart Ambulance Detection and Priority Checking. This research can be further enhanced by integrating advanced IoT, GPS, real-time data analytics, AI, and cloud technologies. Using sensors, machine learning, and adaptive traffic management algorithms, the system can become more efficient, scalable, and intelligent, ensuring faster emergency response while minimizing misuse. Several areas for future improvements and applications are identified below:

Login and Authentication for Web or App:-

To prevent unauthorized access and misuse, future systems can implement secure login and authentication mechanisms for ambulance drivers, traffic authorities, and administrators. This ensures that only verified personnel can access the control and monitoring features of the system.

AI-Driven Routing and Nearby Hospital Guidance:-

An AI-based module can provide dynamic routing suggestions to ambulances, including alternative paths during traffic congestion. The system can also identify and display the nearest hospitals in real time, helping drivers reach medical facilities faster.

Backup Operation for Internet or Network Failures:-

To ensure reliability, the system can incorporate fallback mechanisms that function during internet or network outages. Alternative offline routing methods, local signal control, or mesh network communication can maintain emergency vehicle priority until connectivity is restored.

References:-

1. Article regarding Ambulance makes delay in metropolitan cities
<https://timesofindia.indiatimes.com/city/mumbai/ambulance-delay-due-to-traffic-stalls-emergency-care/articleshow/122888805.cms?>
2. Nellore, K., & Hancke, G. P. (2016). Traffic management for emergency vehicle priority based on visual sensing. *Sensors*, 16(11), 1892. <https://doi.org/10.3390/s16111892>

3. "Ambulance Assistance for Emergency Services Using GPS Navigation" Shantanu Sarkar ,School of Computer Science, VIT University, Vellore ,International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 03 Issue: 09th Sep-2016
4. "A Survey on Smart Ambulance Traffic Control System "— Anu Philip et al., International Journal of Computer Sciences and Engineering, Vol 5, Issue 4 (April 2017). It references Android mobile device + GSM module + MQTT + Arduino UNOtraffic signals.
5. Roopashree V., Malavika D. N., Nikitha Bai E., Suman A., & Dr. Shashikala. "Traffic Congestion Detection and Alerting Ambulance using IoT." International Journal of Engineering Research & Technology (IJERT), Vol. 9, Issue 07, July 2020. DOI: 10.17577/IJERTV9IS070590
6. The paper "FIREBASE- OVERVIEW AND USAGE" by Pankaj Chougale, Vaibhav Yadav, and Dr. Anil Trimbakrao Gaikwad, published in the Journal of Engineering and Technology Management in August 2022.
7. Rezania Agramanisti Azdy and Febriyanti Darnis,"Use of Haversine Formula in Finding Distance Between Temporary Shelters and Waste End Processing Sites " Published: May 29, 2020 Journal: Journal of Physics.
8. Mishra, M., Singh, S., Jayalekshmi, K. R., & Nadkar, T. (2017).Advance alert for ambulance pass by using IoT for smart city.In Proceedings of the International Conference on Computing, Communication and Automation (ICCCA 2017) (pp. 241–245). IEEE. <https://doi.org/10.1109/CCAA.2017.8229784>
9. SUREN KRISHNAN, RAJAN THANGAVELOO*, SHAPI-EE BIN ABD RAHMAN & SIVA RAJA SINDIRAMUTTY Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, "Smart Ambulance Traffic Control System" published on : 2021-06-29 .
10. In 2018 4th International Conference on Recent Advances in Information Technology (RAIT)" Design and Development of Green Corridor Using IoT" by Kumar, R., & Mittal, S..
11. A book named "Smart Sustainable Cities of the Future" by Simon Elias Bibri-2018 .
12. Portable wireless traffic light system (PWTLs). By Isa, I., Shaari, N., Fayeez, A., & Azlin, N. (2014). Journal of research in Engineering and doi:10.15623/ijret.2014.0302041 Technology,, 242-247
13. " Internet of Things for Smart Traffic Monitoring Systems: A Survey" by Mahdavejad M. S. et al. (2018)"
14. International Journal of Advanced Research in Computer Science and Electronics Engineering, by Ramani, B. A., & Jeyakumar, A. (2018). "Smart ambulance system"
15. "Implementation of Haversine Formula for Counting Event Visitor in The Radius Based on Android Application," in 2016 4th International Conference on Cyber and IT Service Management, 2016, by C. N. Alam, K. Manaf, A. R. Atmadja, and D. K. Aurum.
16. News Article : <https://www.straitstimes.com/asia/se-asia/patients-die-as-manila-traffic-jams-block-ambulances>
17. Website of UNECE https://unece.org/transport/road-traffic-safety-0?utm_ https://www.who.int/health-topics/road-safety?#tab=tab_1 Road safety precautions regarding to traffic.
18. Reliability ofEmergency Rescue System on Highway Dian liang Xiao, Yu-jia Tian., IEEE, 2009.
19. Traffic management systems: A classification, review, challenges, and future perspectives: https://smarnet.niua.org/sites/default/files/resources/155014771_6683612.pdf
20. Ferreira, M. C., Fernandes, R., Conceição, H., Viriyasivat, W., & Tonguz, O. K. (2017) Self-organized traffic control. IEEE Transactions on Intelligent Transportation Systems, 18(11), 3092–3106. <https://doi.org/10.1109/TITS.2017.2703142>