Smart RF-Based Traffic Signal Priority System for Ambulance Movement Using Arduino and Application-Based Demonstration

Kavish Jhaveri
The National High School Gujarat State Education Board

***Abstract*—Traffic congestion at signalized intersections is one of the major causes of delay for ambulances during emergency response. Even a short waiting period at a traffic signal can affect the ability of an ambulance to reach a hospital or emergency location quickly. This paper presents a prototype smart traffic control system designed to provide priority passage to ambulances at road intersections. The proposed system uses a dedicated control panel operated by a traffic police officer or control room operator. When an ambulance approaches a traffic signal, the operator presses a direction-specific push button on the transmitter unit. The transmitter sends a 433 MHz radio frequency signal to the receiver unit connected to the traffic signal controller. After receiving the command, the receiver activates the signal path for the ambulance and blocks traffic movement from the remaining three directions. The prototype uses Arduino Uno boards on both transmitter and receiver sides, push buttons, a 433 MHz RF transmitter and receiver pair, traffic signal LEDs, a NeoPixel LED, buzzer, Bluetooth module, and a basic application interface for ambulance movement and manual control demonstration. The system was evaluated through prototype-level testing rather than full-scale field deployment. The observed demonstration shows that the proposed design can represent emergency signal priority logic in a simple, low-cost, and understandable manner. The paper discusses the system architecture, working principle, hardware implementation, software demonstration, advantages, limitations, and future development scope.**

***Index Terms*—Ambulance priority, Arduino Uno, emergency vehicle preemption, intelligent traffic control, RF communication, smart traffic signal, 433 MHz module.**

I. INTRODUCTION

Urban traffic congestion is a serious challenge for emergency medical transportation. Ambulances often face delays at signalized intersections because regular traffic signal cycles do not automatically recognize emergency vehicle movement. In many cases, ambulance drivers rely on sirens, flashing lights, and the cooperation of surrounding drivers. However, these conventional methods are not always effective in crowded urban environments where vehicles may not have enough space to move aside. As a result, an ambulance may be forced to wait at a red signal even during a medical emergency.

A smart traffic signal priority system can reduce such delays by giving temporary right-of-way to an ambulance. The basic idea is to interrupt the normal signal cycle for a short emergency period, stop conflicting traffic directions, and clear the path required by the ambulance. This concept is commonly related to emergency vehicle preemption, where signal behavior is adjusted to support emergency vehicle movement.

The project presented in this paper focuses on a practical prototype for ambulance priority using radio frequency communication. Instead of using a fully automated camera or GPS-based detection system, the proposed model uses a control panel operated by a traffic police officer or control room operator. When the ambulance approaches a traffic signal, the operator selects the required direction by pressing a button. The transmitter sends a 433 MHz RF command to the receiver installed near the traffic signal control unit. The receiver processes the signal and changes the traffic signal LED states so that the ambulance direction is cleared while the other three directions are blocked. [1], [2], [3], [4]

The system also includes a basic application demonstration. The application is not presented as a fully deployed traffic management platform; rather, it demonstrates the concept of ambulance movement, location visibility, and manual control buttons for traffic congestion or special cases. This makes the project suitable for academic demonstration, embedded systems learning, and early-stage intelligent transportation system prototyping.

II. LITERATURE REVIEW

Emergency vehicle priority at intersections has been studied under intelligent transportation systems, traffic signal preemption, and smart city traffic management. Traditional emergency response depends mainly on sirens and warning lights. Although these methods alert nearby drivers, they do not directly control the traffic signal. In congested roads, the drivers ahead of an ambulance may be unable to create a clear path, especially near a red signal. This limitation has encouraged the development of emergency vehicle preemption systems [5].

Traffic signal preemption systems are designed to modify normal signal timing when an emergency vehicle approaches. Common technologies include optical emitters, acoustic detection, GPS-based tracking, radio communication, vehicle-to-infrastructure communication, and centralized control room systems. Optical and acoustic systems depend on line-of-sight or environmental conditions. GPS-based approaches provide location awareness but require network connectivity, mapping support, and software that is more complex. Vehicle-to-infrastructure systems are powerful but may require large-scale infrastructure upgrades [6], [7], [8].

Several research studies have also explored intelligent traffic control using fuzzy logic, machine learning, IoT, and connected vehicle systems. These methods can improve decision-making by considering real-time traffic density, vehicle location, and route priority. However, advanced approaches may be expensive or difficult to implement in small academic prototypes. For educational and low-cost demonstration purposes, RF-based communication remains useful because it can transmit simple control commands wirelessly with relatively simple hardware.

The 433 MHz RF transmitter-receiver pair is commonly used in embedded systems for short-range wireless control. It can send simple encoded commands from one microcontroller to another. In the context of this project, the RF module is used to transmit the selected ambulance direction from the control panel to the traffic signal receiver unit[9], [10].

III. METHODOLOGY

The methodology followed in this project includes problem identification, system requirement analysis, hardware selection, transmitter and receiver circuit design, control logic development, application interface demonstration, and prototype testing. First, the problem of ambulance delay at traffic signals was studied. The main functional requirement was defined as the ability to select an ambulance direction manually and change the traffic signal state accordingly. Second, the system was divided into two units: a transmitter-side control panel and a receiver-side traffic signal controller [11], [12], [13].

Arduino Uno was selected as the microcontroller platform because it is widely available, simple to program, and suitable for digital input-output control. A 433 MHz RF transmitter and receiver pair was used for wireless communication between the control panel and signal controller. The receiver logic was programmed to activate the ambulance route and block the other three directions. Finally, a basic front-end application was created to demonstrate ambulance movement, location display, and manual control options for special traffic conditions.

The prototype was tested in a controlled demonstration environment. Since field testing at a real traffic intersection was not conducted, the results are presented as qualitative prototype outcomes and clearly labeled sample demonstration graphs rather than measured traffic performance statistics.

IV. PROPOSED SYSTEM DESIGN

The proposed system consists of three main layers: the control input layer, wireless communication layer, and traffic signal output layer. The control input layer contains the operator control panel. It includes four push buttons representing four traffic directions, such as North, South, East, and West. When the operator identifies the direction from which an ambulance is approaching, the corresponding button is pressed.

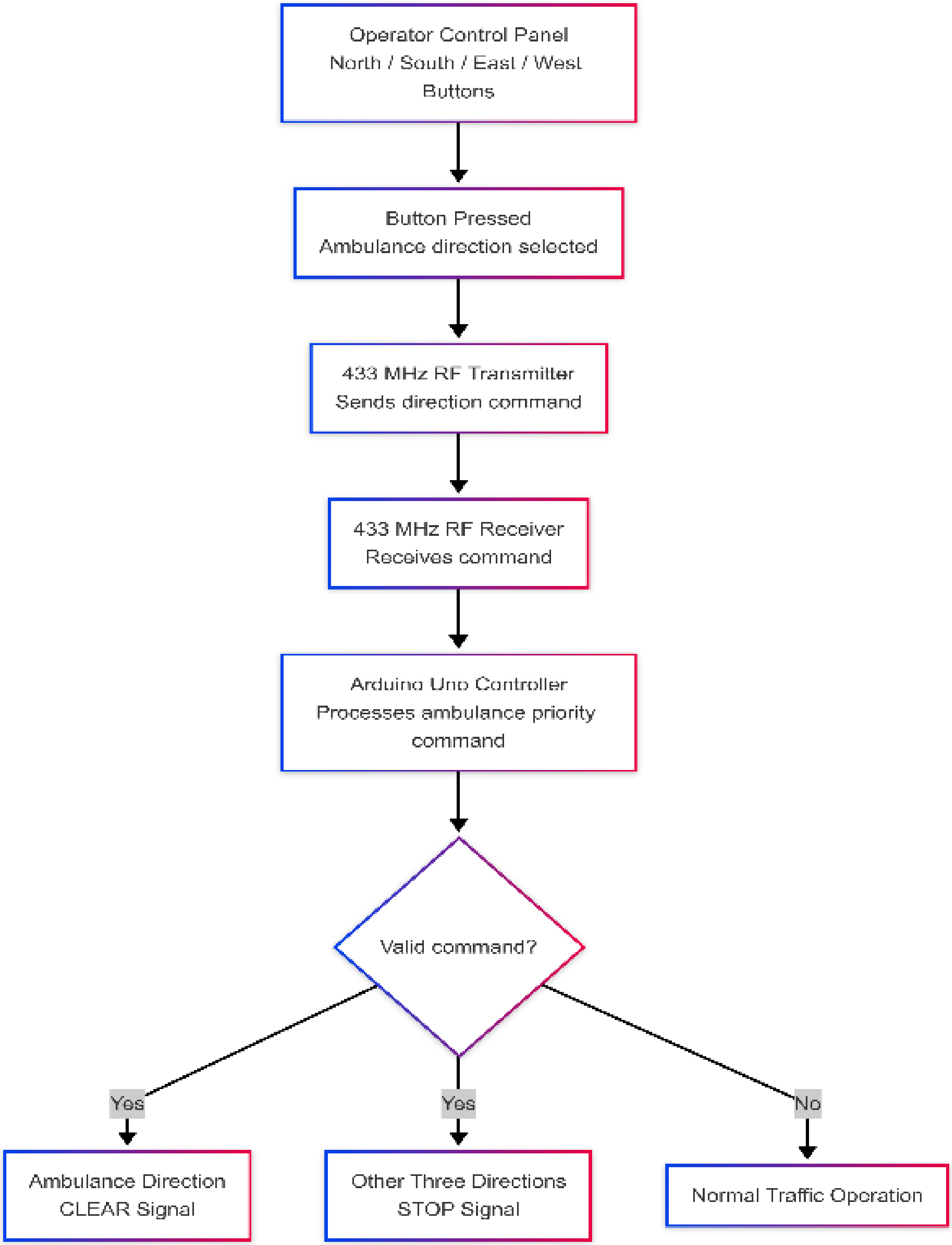

Figure 1 :- Flow diagram of the system architecture

The wireless communication layer uses a 433 MHz RF transmitter and receiver. The transmitter sends a direction-specific command. The receiver receives the command and passes it to the Arduino Uno at the traffic signal side. The traffic signal output layer consists of signal LEDs representing four directions. When a valid ambulance priority command is received, the receiver-side Arduino sets the ambulance direction to a clear state and sets the remaining three directions to a stop state.

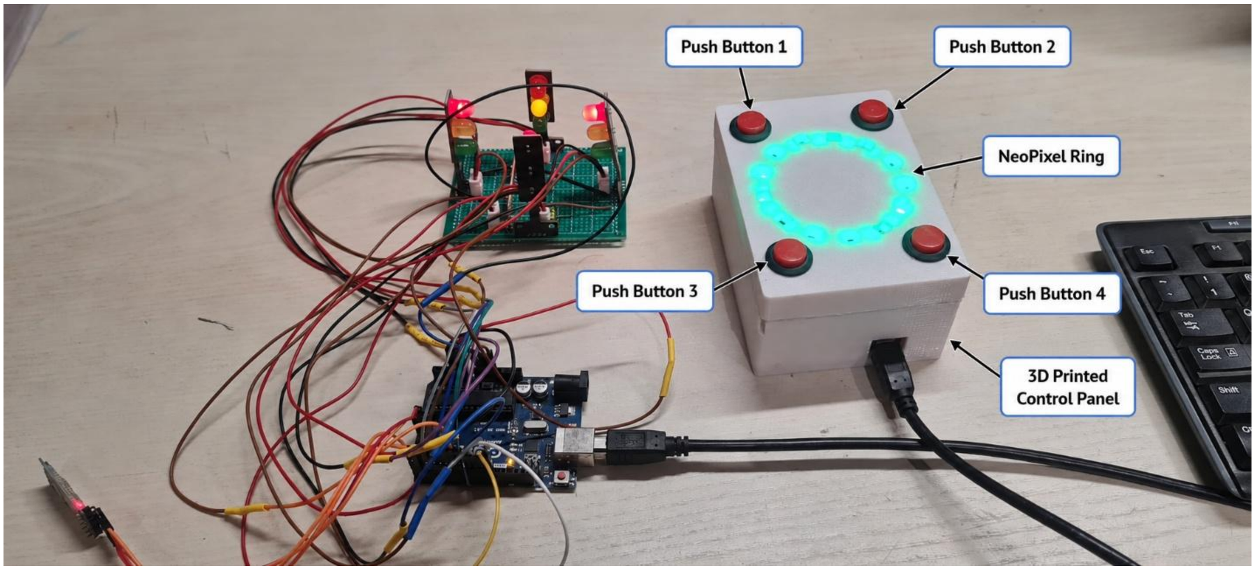

Figure 2 : - Circuit for the receiver module

The receiver unit represents the traffic signal control side. It includes another Arduino Uno, a 433 MHz RF receiver module, four traffic signal LEDs, and a Bluetooth module. The receiver module collects the transmitted signal. The Arduino then updates the LED outputs according to the selected ambulance route. The Bluetooth module may be used for application-based demonstration or local communication with a display/control interface.

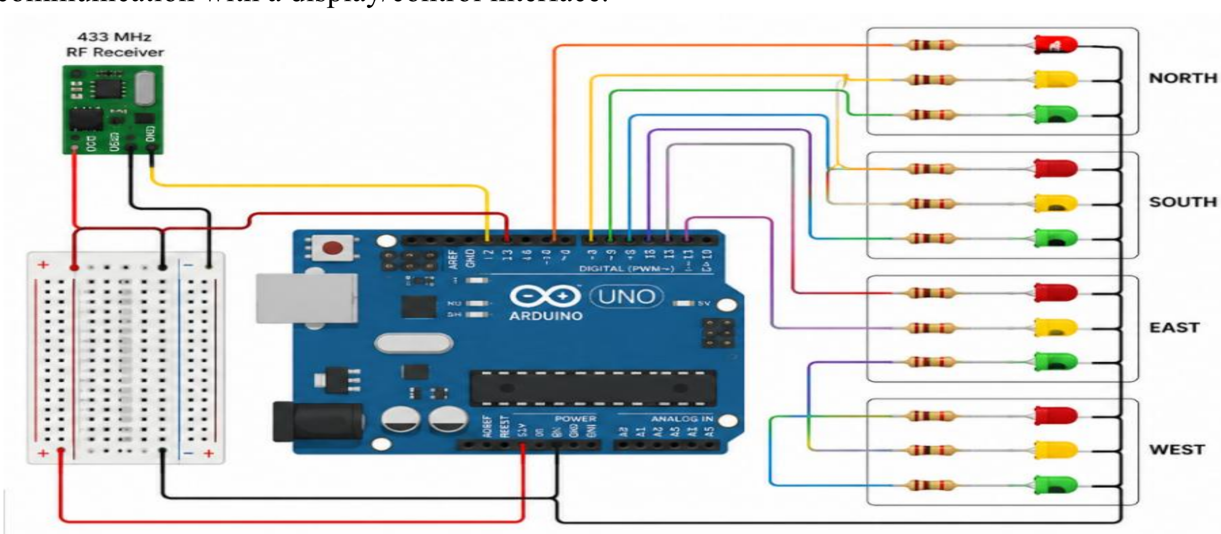

Figure 3 : - Circuit diagram of the receiver module

The receiver-side circuit of the proposed ambulance priority traffic signal system is built around an Arduino Uno and a 433 MHz RF receiver module. The RF receiver captures the wireless command sent from the transmitter side and forwards the received signal to the Arduino Uno for processing. Based on the received command, the Arduino controls the traffic signal LED units representing the North, South, East, and West directions.

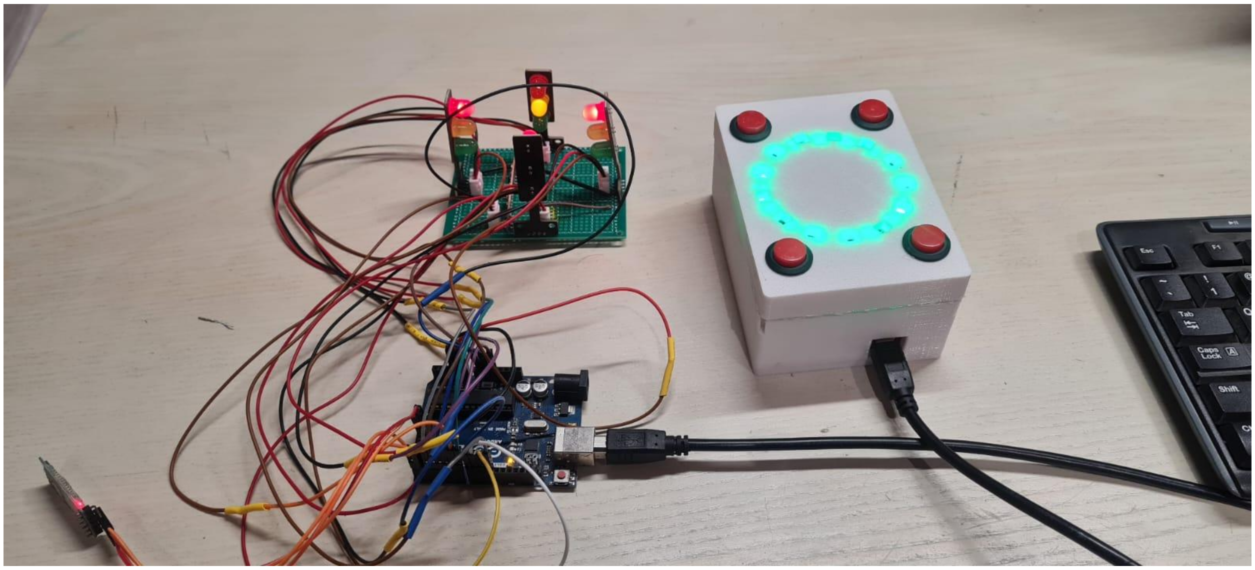

Figure 4 : - 3D Printed control panel having four push button with neo pixel for controlling the traffic light for transmitter module

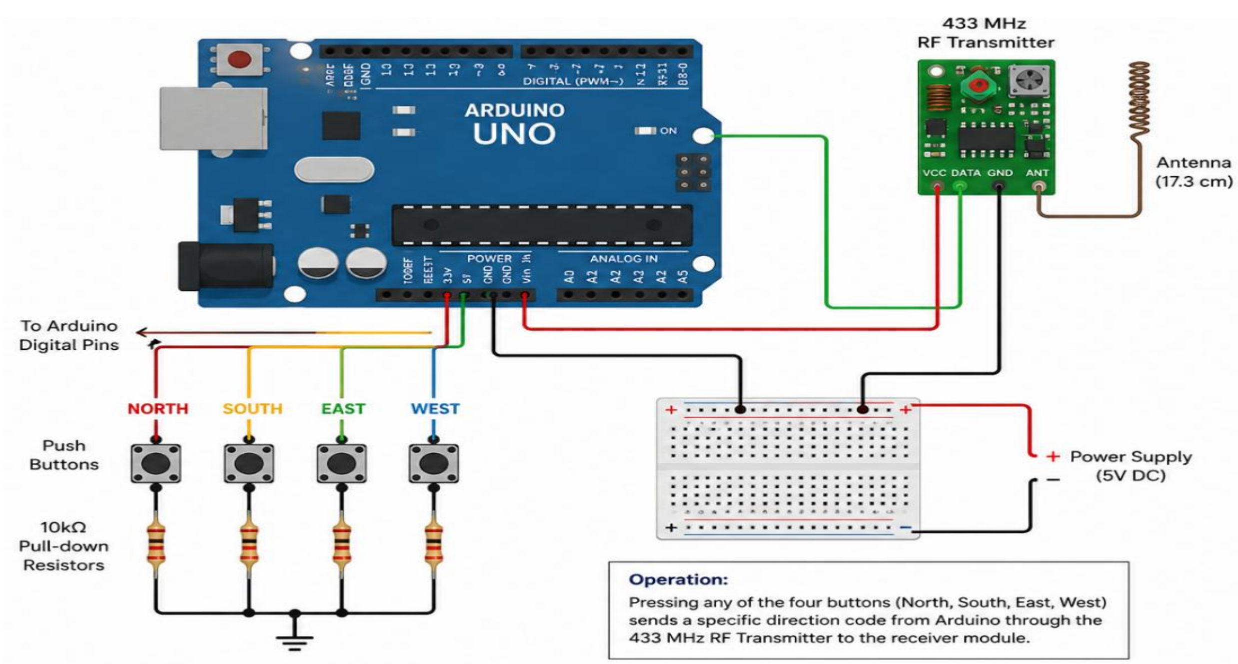

Figure 5 : - Circuit diagram of the transmitter module

The transmitter unit works as the control panel. It includes an Arduino Uno, four push buttons, a 433 MHz RF transmitter module, a NeoPixel LED, and a buzzer. Each push button corresponds to one traffic direction. When a button is pressed, the Arduino identifies the selected direction and sends the appropriate signal through the RF transmitter. The buzzer can provide an audio confirmation, while the NeoPixel LED can provide visual feedback.

V. SOFTWARE AND APPLICATION DEMONSTRATION

Figure 5:- home screen of the ui for traffic police and Civilian patient

The software side consists of Arduino firmware and a basic application interface. The Arduino transmitter program continuously checks the push button states. When a button is pressed, the program assigns a direction code and transmits it through the RF transmitter. Debouncing logic may be added to prevent false multiple triggers from a single press.

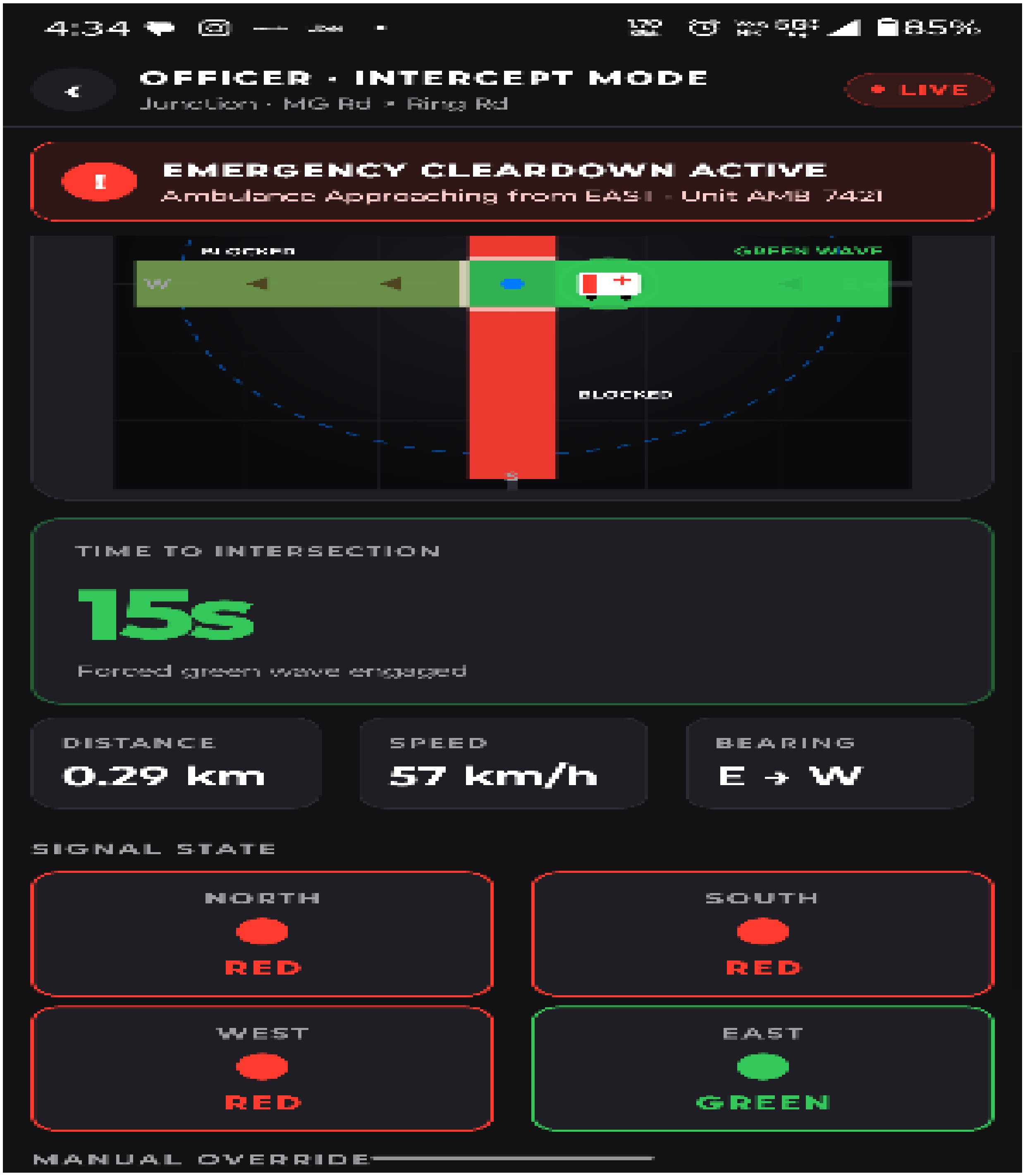

Figure 6 :- Traffic Police Screen which he will be able to see on app

The receiver-side program listens for incoming RF commands. When a valid command is received, it compares the received direction code with predefined values. Based on the decoded direction, the Arduino updates the signal LEDs. For example, if the ambulance is coming from the North direction, the North route is cleared while South, East, and West are stopped [14].

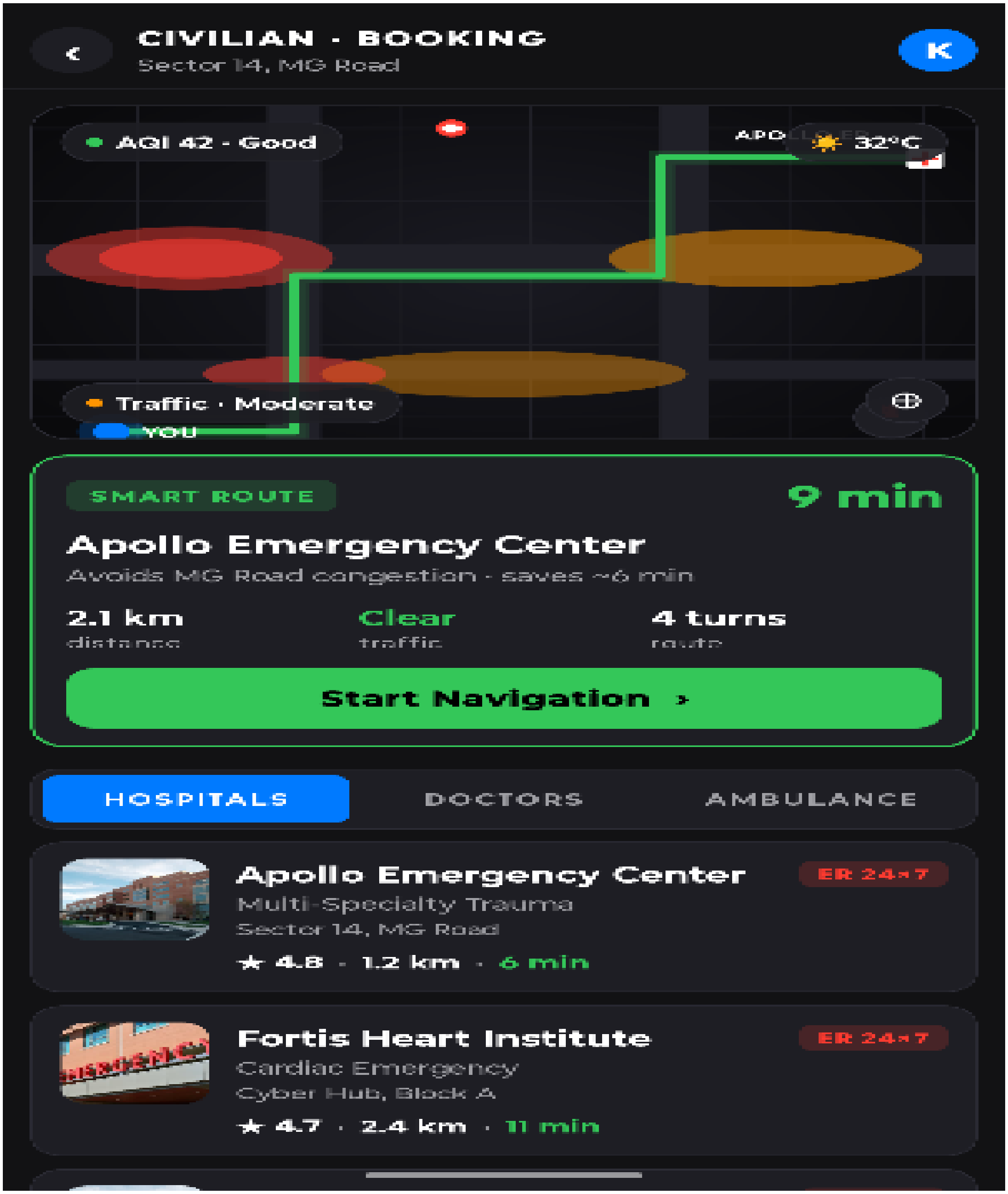

Figure 7 : - The Civilian screen app for making the ambulance booking and transit

The application interface demonstrates the movement of the ambulance and provides a visual representation of its location. It also includes manual control buttons that can be used conceptually during congestion or special traffic conditions. In the current prototype, the application is used mainly for concept demonstration.

VI. WORKING PRINCIPLE

The working principle of the system is based on manual emergency priority activation through RF communication. Under normal conditions, the traffic signal may operate according to regular timing. When an ambulance approaches the signal, the operator determines the required direction and presses the corresponding button on the transmitter panel.

After the button press, the transmitter-side Arduino generates a direction command. This command is sent wirelessly using the 433 MHz RF transmitter. The receiver module placed at the traffic signal side receives the RF signal and forwards it to the Arduino Uno. The Arduino checks the command and changes the traffic signal outputs.

If the received command corresponds to Direction 1, then Direction 1 is given priority. The LED for that direction indicates clearance, while the LEDs for the remaining directions indicate stop. The same logic is applied for Direction 2, Direction 3, and Direction 4. This creates a controlled emergency passage for the ambulance.

VII. RESULTS AND DISCUSSION

The system was tested as a prototype demonstration using Arduino-based transmitter and receiver units. The main objective of the test was to verify whether button-based direction selection could be transmitted wirelessly and reflected in the traffic signal LED outputs. The prototype successfully demonstrated the basic emergency priority logic. When a direction button was pressed on the transmitter panel, the receiver responded by activating the corresponding traffic signal state and stopping the other three directions.

The following python graphs are based on controlled prototype behavior and sample demonstration data. They are included to explain transmitter response, receiver transition logic, RF reception trends, and response latency under different operating conditions. These figures should not be interpreted as field-validated traffic performance statistics.

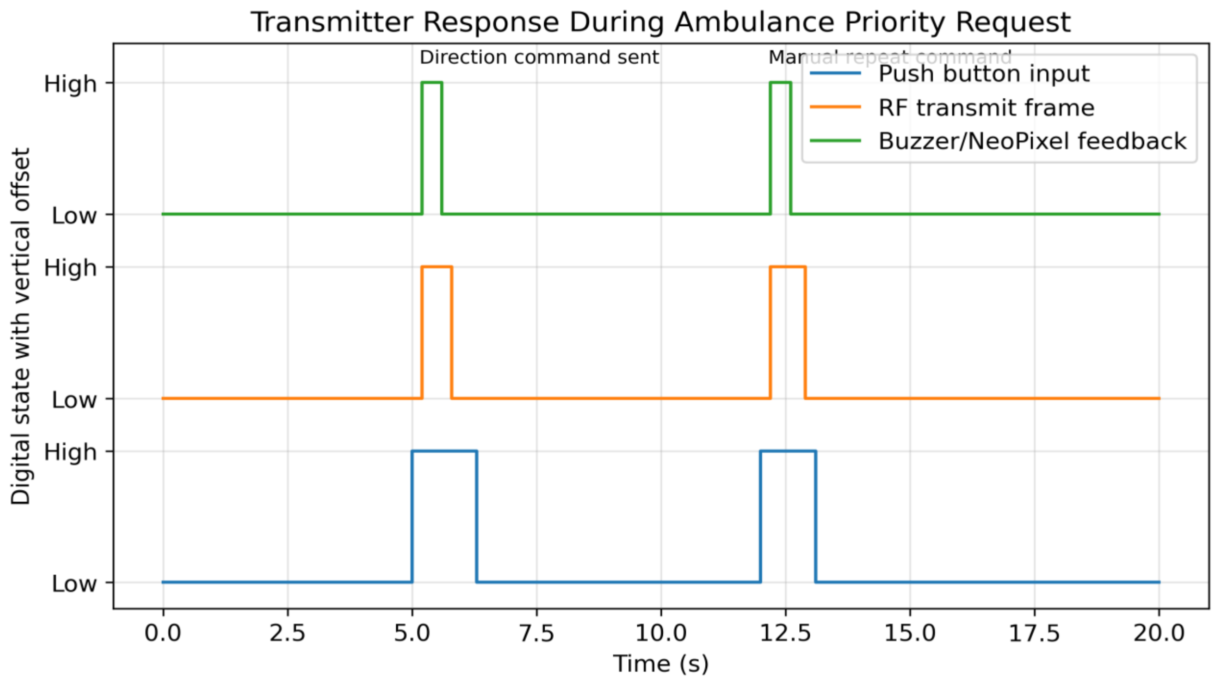

Figure 8 :- Transmitter response during ambulance priority request. The graph shows button activation, RF command transmission, and local feedback output using prototype demonstration data.

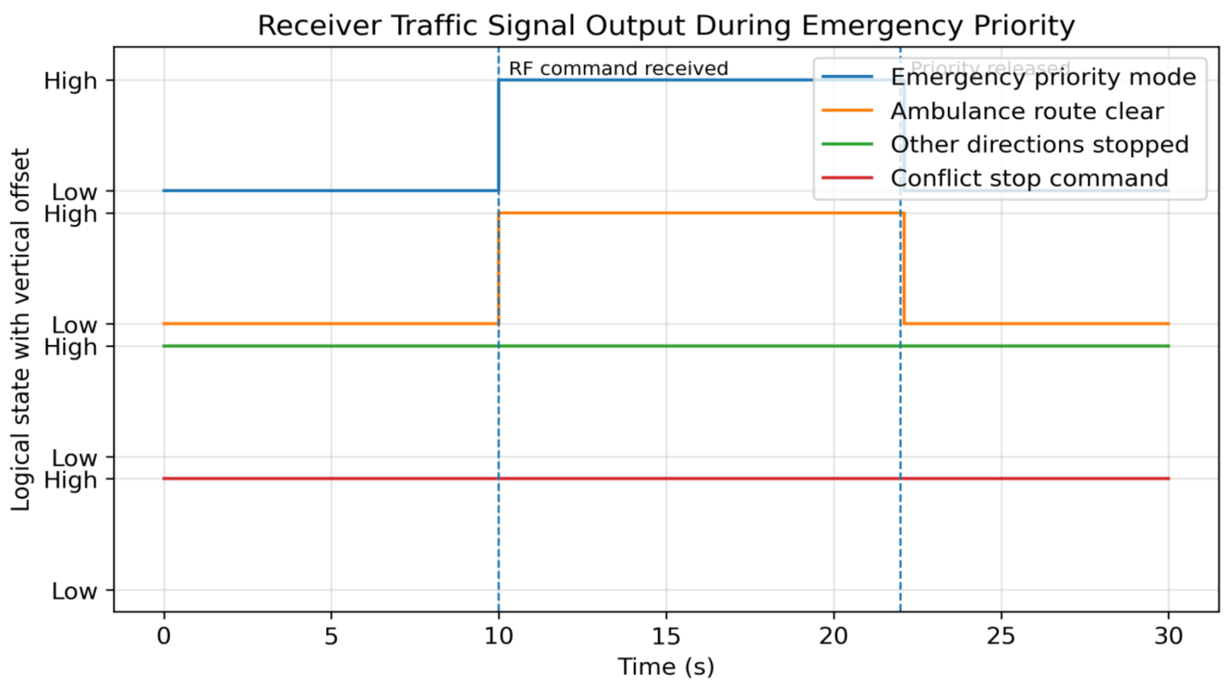

Figure 9:- Receiver traffic signal output during emergency priority mode. The ambulance route is cleared while the remaining three traffic directions are stopped.

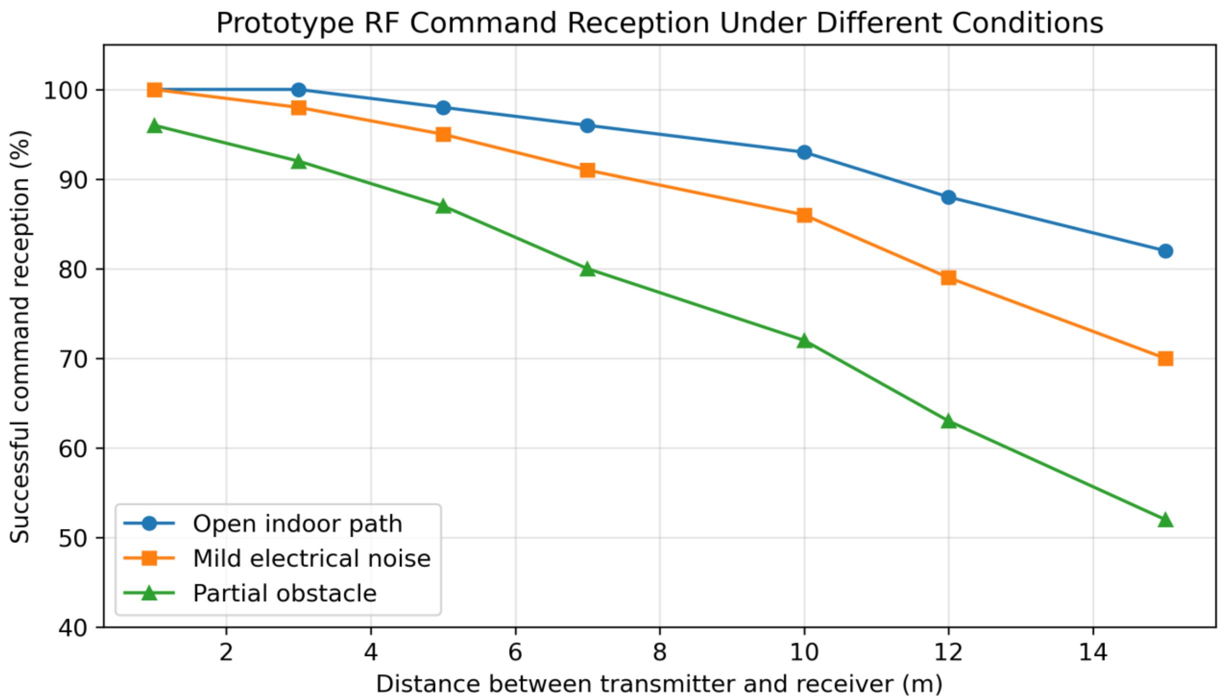

Figure 10 :- Prototype RF command reception under different operating conditions. Reception quality decreases with distance and obstruction. Values are based on sample demonstration data.

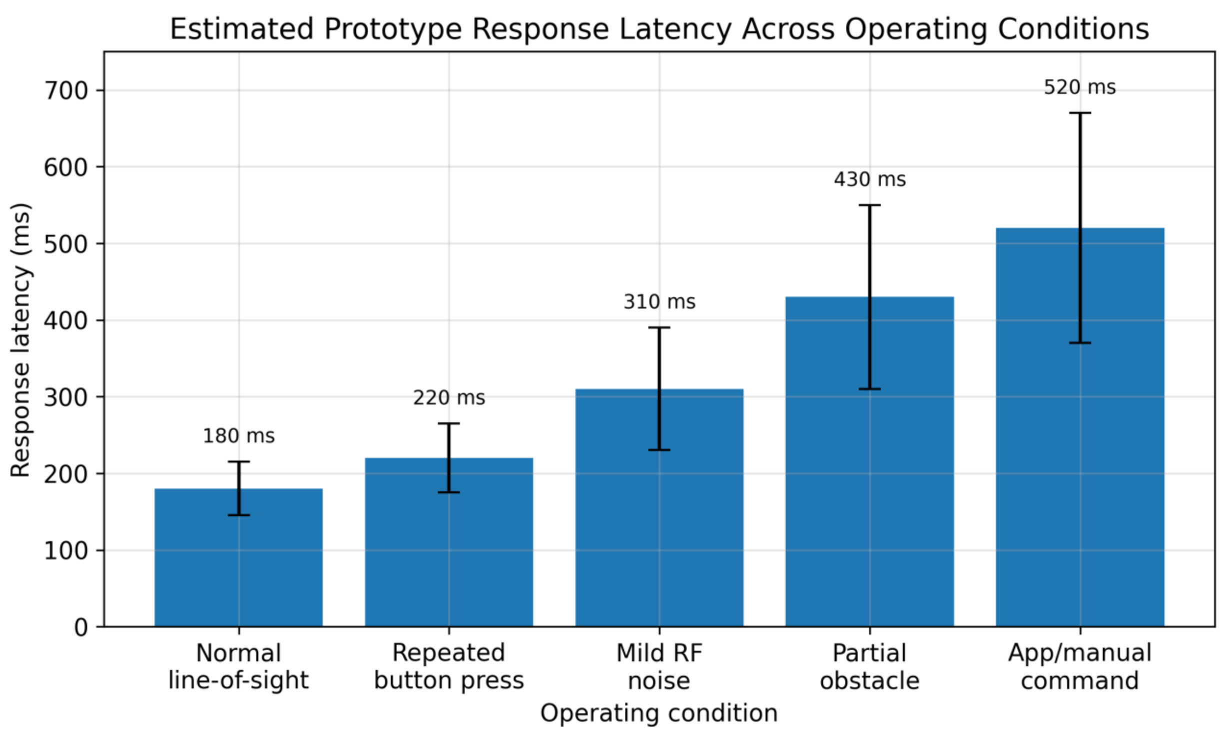

Figure 11 :- Estimated prototype response latency across operating conditions. The values are illustrative and represent controlled demonstration behavior rather than field deployment data.

Algorithm 1: Ambulance Priority Traffic Signal Control[15], [16], [17], [18], [19]
Input:
    Direction button pressed by operator:
    North, South, East, or West

Output:
    Green signal for ambulance direction
    Red signal for all conflicting directions

Begin
1. Initialize transmitter-side Arduino Uno.
2. Initialize receiver-side Arduino Uno.
3. Initialize RF transmitter and RF receiver modules.
4. Initialize four push buttons:
    B_North, B_South, B_East, B_West
5. Initialize traffic signal outputs:
    S_North, S_South, S_East, S_West
6. Initialize feedback devices:
    buzzer and NeoPixel LED
7. Set all traffic signals to normal operating mode.
8. Repeat continuously:
    8.1 Read the status of all direction buttons.
    8.2 If B_North is pressed, then
        command ← NORTH_PRIORITY
        transmit command using RF transmitter
        activate buzzer and NeoPixel feedback
    8.3 Else if B_South is pressed, then
        command ← SOUTH_PRIORITY
        transmit command using RF transmitter
        activate buzzer and NeoPixel feedback
    8.4 Else if B_East is pressed, then
        command ← EAST_PRIORITY
        transmit command using RF transmitter
        activate buzzer and NeoPixel feedback
    8.5 Else if B_West is pressed, then
        command ← WEST_PRIORITY
        transmit command using RF transmitter
        activate buzzer and NeoPixel feedback
    8.6 Else
        continue normal traffic signal operation

8.7 At the receiver side, check for incoming RF command.
8.8 If a valid RF command is received, then
  decode the received command
8.9 If command = NORTH_PRIORITY, then
  S_North ← GREEN
  S_South ← RED
  S_East ← RED
  S_West ← RED
8.10 Else if command = SOUTH_PRIORITY, then
  S_South ← GREEN
  S_North ← RED
  S_East ← RED
  S_West ← RED
8.11 Else if command = EAST_PRIORITY, then
  S_East ← GREEN
  S_North ← RED
  S_South ← RED
  S_West ← RED
8.12 Else if command = WEST_PRIORITY, then
  S_West ← GREEN
  S_North ← RED
  S_South ← RED
  S_East ← RED
8.13 Else
  ignore the command
8.14 Maintain the emergency priority signal state
  until the ambulance crosses the intersection.
8.15 Restore the normal traffic signal cycle.
9. End Repeat
End

Algorithm 2: RF Transmitter-Side Ambulance Direction Selection

Input:
  Direction selected by operator using push buttons:
  B_North, B_South, B_East, B_West

Output:
  Direction-specific RF command transmitted to receiver unit

Begin
1. Initialize Arduino Uno on transmitter side.
2. Initialize 433 MHz RF transmitter module.
3. Initialize four push buttons:
    B_North
    B_South
    B_East
    B_West
4. Initialize feedback devices:
    buzzer
    NeoPixel LED
5. Set all direction buttons as input.
6. Set buzzer and NeoPixel LED as output.
7. Repeat continuously:
    7.1 Read the status of B_North, B_South, B_East, and B_West.

    7.2 If B_North is pressed, then
        command ← NORTH_PRIORITY
        transmit command through RF transmitter
        activate buzzer
        set NeoPixel LED to confirmation state
        wait for debounce delay

    7.3 Else if B_South is pressed, then
        command ← SOUTH_PRIORITY
        transmit command through RF transmitter
        activate buzzer
        set NeoPixel LED to confirmation state
        wait for debounce delay

    7.4 Else if B_East is pressed, then
        command ← EAST_PRIORITY
        transmit command through RF transmitter
        activate buzzer
        set NeoPixel LED to confirmation state
        wait for debounce delay

    7.5 Else if B_West is pressed, then
        command ← WEST_PRIORITY
        transmit command through RF transmitter

activate buzzer
set NeoPixel LED to confirmation state
wait for debounce delay

7.6 Else
keep transmitter in standby mode
continue monitoring buttons

8. End Repeat

End

Mathematical Formation for Codding: Begin [1], [2], [3], [4], [11], [20]

1. Initialize transmitter unit:
   Arduino Uno, RF transmitter, push buttons, buzzer, and NeoPixel LED.

2. Define the button state function:

$$B_i(t) = \begin{cases} 1, & \text{if direction button } i \text{ is pressed at time } t \\ 0, & \text{otherwise} \end{cases}$$

where $i \in \{N, S, E, W\}$

3. Define the command generation function:

$$C(t) = \begin{cases} C_N, & \text{if } B_N(t) = 1 \\ C_S, & \text{if } B_S(t) = 1 \\ C_E, & \text{if } B_E(t) = 1 \\ C_W, & \text{if } B_W(t) = 1 \\ \emptyset, & \text{if } B_N(t) = B_S(t) = B_E(t) = B_W(t) = 0 \end{cases}$$

4. Continuously read all button states:

$$B(t) = [B_N(t), B_S(t), B_E(t), B_W(t)]$$

5. If any button state is equal to 1, generate the corresponding command C(t).

6. Transmit the generated command using the RF transmitter:

   RF_TX(C(t))

7. Activate feedback output:

$$F(t) = \begin{cases} 1, & \text{if } C(t) \neq \emptyset \\ 0, & \text{if } C(t) = \emptyset \end{cases}$$

   If F(t) = 1:
      Turn ON buzzer
      Turn ON NeoPixel confirmation indication

8. Apply debounce delay:

   $t = t + \Delta t$

   where Δt is the debounce time used to prevent repeated false triggering.

9. If no button is pressed:

   $C(t) = \emptyset$

   Keep transmitter in standby mode.

10. Repeat steps 4 to 9 continuously.

End

$B_i(t)$ = Button input for direction i at time t

C_i = Priority command assigned to direction i

$RF\_TX(C_i)$ = Transmission of direction command through 433 MHz RF module

F(t) = Feedback output for buzzer and NeoPixel LED

VIII. FUTURE SCOPE

Future work can improve the system in several ways. GPS-based ambulance tracking can be added so that the system can automatically detect the ambulance location and approaching direction. The RF system can be replaced or supported by more secure communication technologies such as GSM, LoRa, Wi-Fi, or vehicle-to-infrastructure communication. Authentication can be added so that only authorized police or control room devices can control the signal. The application can be expanded into a real-time dashboard showing ambulance route, signal status, hospital destination, estimated arrival time, and congestion level.

The system can also be improved using traffic density sensors, cameras, or ultrasonic sensors to manage queue clearance more intelligently. Instead of simply blocking three directions, the future version can calculate the safest transition phase and gradually clear traffic before the ambulance reaches the intersection. A field-ready system should also include fail-safe operation, logging, manual override, power backup, and compliance with local traffic authority standards.

IX. CONCLUSION

This paper presented a prototype smart traffic control system for giving priority passage to ambulances at traffic signals. The system uses a dedicated operator control panel, Arduino Uno microcontrollers, four push buttons, 433 MHz RF transmitter and receiver modules, traffic signal LEDs, a NeoPixel LED, buzzer, Bluetooth module, and a basic application demonstration. When an ambulance approaches a signal, the operator presses the appropriate button. The transmitter sends a wireless command to the receiver, and the receiver changes the signal state to clear the ambulance route while blocking the other three directions.

The prototype successfully demonstrates the core logic of ambulance priority control in a low-cost and easy-to-understand manner. However, the results are limited to controlled prototype demonstration and do not represent real world field performance. For practical deployment, the system would require secure communication, automatic detection, and integration with real traffic controllers, safety validation, and testing under actual traffic conditions. Overall, the proposed system provides a useful foundation for developing smart ambulance priority solutions for intelligent transportation and emergency response applications.

REFERENCES

[1] H. Wu, M. Dyson, and K. Nazarpour, “Arduino-based myoelectric control: Towards longitudinal study of prosthesis use,” Sensors (Switzerland), vol. 21, no. 3, pp. 1–13, 2021, doi: 10.3390/s21030763.

[2] E. O. Fernandez et al., “Arduino-based Sound Acquisition System Using Fast Fourier Transform Algorithm,” 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2019, 2019, doi: 10.1109/HNICEM48295.2019.9072845.
[3] M. Hasibuzzaman, A. Shufian, R. K. Shefa, R. Raihan, J. Ghosh, and A. Sarker, “Vibration Measurement Analysis Using Arduino Based Accelerometer,” 2020 IEEE Region 10 Symposium, TENSYMP 2020, no. June, pp. 508–512, 2020, doi: 10.1109/TENSYMP50017.2020.9230668.
[4] P. A. Parikh, R. S. Tolaney, C. M. Shah, A. P. Head, and U. G. Students, “Wireless Multichannel Data Acquisition System using ARDUINO and MATLAB GUI,” International Journal of Engineering Development and Research, vol. 4, pp. 1711–1716, 2016, [Online]. Available: www.ijedr.org
[5] C. Kwan et al., “Real-time and deep learning based vehicle detection and classification using pixel-wise code exposure measurements,” Electronics (Switzerland), vol. 9, no. 6, pp. 1–21, 2020, doi: 10.3390/electronics9061014.
[6] J. H. Cho, D. S. Pae, M. T. Lim, and T. K. Kang, “A Real-Time Obstacle Avoidance Method for Autonomous Vehicles Using an Obstacle-Dependent Gaussian Potential Field,” J. Adv. Transp., vol. 2018, 2018, doi: 10.1155/2018/5041401.
[7] P. Parikh, R. Vasani, and S. Sheth, “Velocity analysis of a DC brushed encoder motor using Ziegler-Nichols algorithm: A case of an automated guided vehicle,” Indian J. Sci. Technol., vol. 9, no. 38, 2016, doi: 10.17485/ijst/2016/v9i38/100884.
[8] P. Parikh, R. Vasani, and S. Sheth, “Velocity analysis of a DC brushed encoder motor using Ziegler-Nichols algorithm: A case of an automated guided vehicle,” Indian J. Sci. Technol., vol. 9, no. 38, 2016, doi: 10.17485/ijst/2016/v9i38/100884.
[9] P. Parikh, R. Vasani, S. Sheth, and J. Gohil, “Actuation of Electro-Pneumatic System using MATLAB Simulink and Arduino Controller-A case of a Mechatronics systems Lab,” in Proceedings of the International Conference on Communication and Signal Processing 2016 (ICCASP 2016), Dr. Brijesh Iyer, Ed., ATLANTIS Press, Springer Nature, 2017, pp. 59–64. doi: 10.2991/iccasp-16.2017.10.
[10] P. Parikh, R. Vasani, S. Sheth, J. Gohil, and A. Professor, “Actuation of Electro-Pneumatic System using MATLAB Simulink and Arduino Controller-A case of a Mechatronics systems Lab,” vol. 137, pp. 59–64, 2017.
[11] K. C. Okafor, G. C. Ononiwu, U. Precious, and A. C. Godis, “Development of Arduino Based IoT Metering System for On - Demand Energy Monitoring,” vol. 7, no. 23, pp. 3208–3224, 2017.
[12] D. Sharma and S. Sarkar, “Enabling Inference and Training of Deep Learning Models for AI Applications on IoT Edge Devices,” Internet of Things, pp. 267–283, 2022, doi: 10.1007/978-3-030-87059-1_10.
[13] C. J. Munoz Martinez, R. Castro Salguero, R. Palomares, and J. Cornejo, “Mechatronics Development of Terrestrial Mobile Robot for Exploring and Monitoring Environmental Parameters at Mine Analogue Sites using IoT Platform,” Proceedings of the 2020 IEEE 27th

International Conference on Electronics, Electrical Engineering and Computing, INTERCON 2020, pp. 9–12, 2020, doi: 10.1109/INTERCON50315.2020.9220227.

[14] J. Wu, M. O'Hern, and J. Ye, "The impact of user innovator mindset on feedback volume, feedback diversity and new product development performance," European Journal of Innovation Management, 2021, doi: 10.1108/EJIM-04-2021-0178.

[15] S. Kucuk, "Optimal trajectory generation algorithm for serial and parallel manipulators," Robot. Comput. Integr. Manuf., vol. 48, no. August 2016, pp. 219–232, 2017, doi: 10.1016/j.rcim.2017.04.006.

[16] I. H. Usoro, U. T. Itaketo, and M. A. Umoren, "Control of a dc motor using fuzzy logic control algorithm," Nigerian Journal of Technology, vol. 36, no. 2, p. 594, 2017, doi: 10.4314/njt.v36i2.35.

[17] P. Kalra, P. B. Mahapatra, and D. K. Aggarwal, "On the solution of multimodal robot inverse kinematic functions using real-coded genetic algorithms," Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1840–1845, 2003, doi: 10.1109/icsmc.2003.1244679.

[18] S. Mathiyazhagan and M. S. G. Devasena, "Motor imagery EEG signal classification using novel deep learning algorithm," Sci. Rep., vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-00824-7.

[19] S. M. H. Rostami, A. K. Sangaiah, J. Wang, and X. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," EURASIP J. Wirel. Commun. Netw., vol. 2019, no. 1, Dec. 2019, doi: 10.1186/s13638-019-1396-2.

[20] F. Ortega-Zamorano, J. L. Subirats, J. M. Jerez, I. Molina, and L. Franco, "Implementation of the C-mantec neural network constructive algorithm in an arduino uno microcontroller," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7902 LNCS, no. PART 1, pp. 80–87, 2013, doi: 10.1007/978-3-642-38679-4_6.