IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211

Smart Movable Road Divider System for Traffic Flow Management

Tulasi K L[1], Siddesh N R[2], Thejashreenivasa M[3], Vs Diwakran[4], Vinayaka M Bhari[5]

Department of Computer Science and Engineering, East West College of Engineering, Bengaluru, Karnataka, India[1-5]

Abstract: Increasing vehicle numbers and uneven traffic distribution create congestion, longer travel times, fuel wastage, and additional emissions. Conventional road dividers are fixed and cannot respond to changing traffic demand. This paper presents a prototype Smart Movable Road Divider System that monitors two lanes and adjusts divider movement according to detected traffic conditions. The system uses an Arduino Uno as the controller, infrared (IR) sensors for vehicle detection and counting, ultrasonic sensors for distance and obstacle monitoring, an L298N motor driver, a DC motor, and a buzzer for audible alerts. The Arduino compares lane counts and controls motor direction when a programmed traffic imbalance is detected, while ultrasonic readings provide a distance-related condition for movement. The project report describes prototype tests under simulated conditions, including vehicle detection and movement in both directions. The design demonstrates the potential of sensor-based automation for adaptive use of road space. However, the available project observations are qualitative and do not establish measured reductions in congestion, fuel consumption, or travel time. Full-scale road deployment would require a robust mechanical design, position feedback, redundant safety controls, extensive testing, and approval from relevant authorities.

Keywords: Smart traffic management; movable road divider; Arduino Uno; IR sensor; ultrasonic sensor; L298N motor driver; lane allocation; embedded control.

I. INTRODUCTION

Traffic congestion is a common transportation problem in growing urban areas. The number of vehicles on a road can change considerably by time of day, day of week, weather, nearby events, and road incidents. During morning and evening peak periods, one direction may carry substantially more traffic than the opposite direction. Nevertheless, conventional central dividers usually maintain a fixed road-space allocation. The lane width available to each direction therefore remains unchanged even when the demand is unequal.

Congestion can increase journey time, fuel use, driver frustration, and vehicle emissions. It may also delay buses, delivery vehicles, and emergency services. Road widening is one possible response, but it may be expensive, require additional land, and take a long time to complete. Traffic-signal coordination and public-transport improvements are also important, but they do not directly change the physical allocation of road width. These issues motivate the exploration of systems that can monitor traffic conditions and respond automatically.

The Smart Movable Road Divider System is a prototype that demonstrates the basic concept of adaptive lane allocation. Sensors collect information about vehicles and nearby objects. An Arduino Uno processes these inputs and sends control signals to a motor driver. The motor driver operates a DC motor that moves a small divider model to the selected side. A buzzer provides an audible indication, while the Arduino Serial Monitor can display sensor values and messages during development.

The proposed prototype is intended for laboratory or demonstration use. A successful model movement does not by itself prove that the design is safe for use on a public road. Full-scale operation would need reliable mechanical guidance, accurate position sensing, traffic-management coordination, and multiple independent safety measures. This paper therefore describes the design concept and the observations supported by the project report, while distinguishing demonstrated prototype behaviour from future possibilities.

A. Motivation

The central motivation is to explore whether a low-cost embedded system can identify a difference in vehicle demand between two lanes and use a motorized divider model to demonstrate a response. The project brings together sensing, microcontroller programming, motor control, and basic obstacle-distance monitoring in a single system.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211

B. Scope

The scope covers a two-lane prototype, vehicle detection using IR sensors, distance measurement using ultrasonic sensors, count comparison in Arduino code, bidirectional DC motor control through an L298N module, audible alerts, and serial debugging. It does not include a validated traffic-engineering algorithm, public-road certification, or quantitative proof of reduced congestion.

II. PROBLEM STATEMENT

Fixed dividers cannot change the allocation of road space according to short-term traffic demand. When one lane carries more vehicles than another, the available road width may not be used efficiently. Manual movement of barriers, where permitted, requires trained personnel, communication, and careful traffic control. An automated system must therefore detect conditions, make a controlled decision, operate the mechanism, and stop movement when the programmed safety condition is not satisfied.

The problem addressed by this project is to develop a small prototype that detects vehicles in two lanes, compares the resulting counts, and demonstrates controlled divider movement based on programmed conditions. The prototype also includes ultrasonic sensors to measure distance and provide a stop-related input. The aim is to demonstrate an embedded-control approach rather than to claim that the current arrangement is ready for real traffic.

III. OBJECTIVES

- To detect passing vehicles in two lanes using infrared sensors.
- To maintain separate vehicle counts and compare the relative demand of the lanes.
- To measure nearby-object distance using ultrasonic sensors.
- To use an Arduino Uno to process sensor signals and apply programmed movement conditions.
- To control the direction of a DC motor through an L298N motor driver.
- To provide audible feedback using a buzzer and status information through the Serial Monitor.
- To observe prototype responses under simulated conditions and identify improvements needed for reliable operation.

IV. RELATED WORK AND LITERATURE REVIEW

The project report describes several related approaches to automated lane adjustment and smart traffic management. These approaches broadly use vehicle-density sensing, microcontrollers, Internet of Things (IoT) communication, or connected monitoring to respond to traffic conditions. The general motivation is similar: traffic demand varies, and fixed infrastructure cannot easily respond to every change.

A. Traffic-Density-Based Lane Adjustment

Traffic-density-based approaches estimate the number or concentration of vehicles in a lane and use this information to guide a traffic-management decision. A simple implementation may count detections over a time window, while a more advanced implementation may use cameras, image processing, or several sensor locations. A count-based approach is easy to understand and can be implemented on a low-cost controller. Its accuracy, however, depends on sensor placement, detection reliability, and the way vehicles are counted.

B. IOT-Enabled Divider Concepts

IoT-based systems can connect traffic sensors to a remote monitoring platform. Such connectivity can support status reporting, event logging, and operator supervision. Remote communication is not required for the basic prototype described here, but it could be added in a future version. A connected system would also need communication-failure handling and protection against unauthorized commands.

C. Emergency-Vehicle Priority

Some smart traffic concepts consider giving priority to ambulances, fire engines, and other emergency vehicles. Such functionality requires reliable identification, authorization, and coordination with traffic-control infrastructure. The present prototype does not include a verified emergency-vehicle recognition function. Emergency priority should therefore be treated as a possible extension, not as a capability demonstrated by the current model.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511211

D. Sensor-based obstacle awareness

Ultrasonic sensors estimate distance by transmitting a sound pulse and measuring the time taken for the echo to return. They are commonly used in small robotic prototypes because they are inexpensive and easy to interface with microcontrollers. Their measurements can be affected by target shape, angle, surface properties, environmental conditions, and sensor placement. For a moving divider, a single distance threshold is not sufficient as the only safety measure.

E. Positioning of the Present Project

The present design focuses on a compact combination of two IR vehicle sensors, two ultrasonic distance sensors, an Arduino Uno, an L298N motor driver, a DC motor, and a buzzer. Compared with a camera-based system, it uses relatively simple sensing and control. This makes the prototype suitable for learning and demonstration, while limiting the level of traffic information it can collect.

TABLE I SUMMARY OF RELATED APPROACHES

Approach	Typical idea	Relevance to this prototype	Important consideration
Traffic-density-based control	Estimate demand and adapt lane allocation.	Lane counts are compared in the Arduino program.	Counting accuracy and decision thresholds must be tested.
IoT traffic monitoring	Transmit sensor readings for remote monitoring.	Could be added for status display and logging.	Communication failure and cybersecurity need attention.
Emergency priority	Identify an authorized emergency vehicle and adjust control.	Potential future extension only.	Requires dependable identification and traffic authority coordination.
Ultrasonic distance sensing	Measure distance using echo travel time.	Provides a distance-related stop condition.	Readings may fluctuate and should not be the only safety layer.
Motorized divider model	Move a physical barrier using an actuator.	Demonstrates left/right motor operation.	Position feedback and mechanical fail-safes are necessary for real use.

Note: The project report refers to specific literature authors and works. Their complete bibliographic information should be checked against the original publications before formal submission. The summary above describes the general approaches reported in the source material and does not claim independent verification of each cited publication.

V. SYSTEM ARCHITECTURE

The system can be divided into four main functional blocks: sensing, control, actuation, and indication. The sensing block consists of two IR sensors and two ultrasonic sensors. The IR sensors provide digital signals used to register vehicle passage. The ultrasonic sensors provide distance estimates for the sides monitored by the prototype. The control block is an Arduino Uno, which reads the sensors and executes the decision logic. The actuation block consists of an L298N motor driver and a DC motor connected to the movable divider model. The indication block consists of a buzzer and serial messages.

A. Sensing Block

The two IR sensors are assigned to the two lanes. The program interprets the sensor output according to the selected module's logic; in the current code notes, LOW represents a vehicle detected and HIGH represents no vehicle. The counter is updated on a programmed state transition. Correct sensor alignment and testing are important because a sensor may register a vehicle incorrectly if its field of detection is poorly positioned.

The two ultrasonic sensors use trigger and echo pins. The controller sends a short trigger pulse and measures the echo duration. The measured time is converted into an approximate distance. The code compares the relevant distance value with a configured stop threshold before continuing movement.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211

B. Control block

The Arduino Uno is responsible for reading the sensor inputs, maintaining lane counts, comparing the counts, and setting motor-control outputs. It also generates buzzer signals and prints diagnostic messages. The movement decision depends on the programmed thresholds and distance conditions. The logic must be reviewed carefully to ensure that contradictory sensor readings, stale counts, or a failed sensor do not lead to an unsafe command.

C. Actuation block

The Arduino output pins are not intended to supply the current required by a DC motor directly. The L298N module acts as a motor driver and allows the controller to select direction through its input pins. The enable input can be used to control motor activation or speed, depending on the wiring and code. A suitable motor power source and a common reference ground are important. The motor should be mechanically coupled to a track or mechanism that limits unwanted movement.

D. Indication and debugging

The buzzer can provide audible feedback when a vehicle is counted or when divider movement is commanded. Serial output helps the developer view distance readings, count values, and status messages. These features make the prototype easier to test, but neither a buzzer nor a serial message is a substitute for an independent emergency stop or a safety-rated control system.

E. Functional sequence

The controller initializes the sensor, motor-driver, and buzzer pins.

IR sensors monitor vehicle passage and the corresponding lane counts are updated.

Ultrasonic sensors measure distances to objects in their monitored areas.

The controller compares lane counts and checks the programmed movement conditions.

If a valid condition is met, the motor driver is commanded to move the divider model in the selected direction.

Movement is stopped when a programmed distance condition or maximum movement time is reached.

The system returns to monitoring and repeats the process.

VI. HARDWARE AND SOFTWARE REQUIREMENTS

A. Arduino Uno

The Arduino Uno is the central controller in the prototype. It reads digital signals from the IR sensors, communicates with the ultrasonic sensors, and generates control signals for the motor driver and buzzer. Its main benefit in this project is the straightforward development environment and availability of digital input/output pins. The program must avoid assigning the same pin to incompatible functions and must use valid syntax and appropriate timing logic.

B. Infrared sensors

IR sensor modules detect an object within their sensing range and commonly provide a digital output. In the prototype, one sensor is associated with each lane. The program uses the change in sensor state to count vehicle passage. A sensor should be tested individually before the full system is operated. The developer should verify whether the module output is active LOW or active HIGH, because different modules may use different output conventions.

C. Ultrasonic sensors

An ultrasonic sensor measures distance from the time taken for an emitted sound pulse to return after reflection. The approximate distance calculation used by the system is:

A. Distance (cm) = [Echo time (microseconds) × 0.0343] / 2

The division by two accounts for the sound pulse travelling to the object and then returning to the sensor. The speed of sound is approximated as 0.0343 centimetres per microsecond under typical room conditions. The value is an estimate and may vary with temperature and measurement geometry.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211

D. L298n motor driver and dc motor

The L298N is a dual H-bridge motor-driver module commonly used in educational robotics. Its input pins select the direction of the motor, while an enable input controls whether the channel is active. The DC motor provides the movement for the divider model. Motor voltage, current, mechanical load, and power-supply capacity must be matched to the actual motor and driver. The controller and motor power arrangement should follow the module documentation.

E. Buzzer and mechanical model

The buzzer provides an audible indication of a detection or movement event. The divider model requires a stable base, a guided path, and mechanical limits that prevent the moving part from leaving the track. A prototype should be operated at low speed and under supervision during initial tests. A full-scale barrier would have substantially different force, mass, weather, and safety requirements.

F. Software Environment

The program is developed in the Arduino IDE. The Serial Monitor is used to observe printed values and messages. During development, the source code should be compiled after each major change, and sensor readings should be verified before enabling motor movement. A missing value in an `analogWrite()` statement, for example, causes a compilation error and must be corrected before upload.

TABLE II MAIN COMPONENTS AND FUNCTIONS

Component	Function in the prototype	Testing focus
Arduino Uno	Reads sensors and runs decision logic.	Pin assignment, compile success, stable readings.
Two IR sensors	Detect vehicle passage in two lanes.	Detection range, output polarity, count accuracy.
Two ultrasonic sensors	Measure nearby-object distance.	Distance stability and stop-threshold behaviour.
L298N driver	Switches motor direction and activation.	Correct wiring, supply, direction response.
DC motor	Moves the divider model.	Direction, load, speed, mechanical limits.
Buzzer	Provides audible alerts.	Correct timing and alert distinction.
Arduino IDE	Compiles and uploads the program.	No compile errors and readable serial output.

VII. METHODOLOGY AND WORKING PRINCIPLE

A. Initialization

When power is applied, the Arduino configures the sensor pins as inputs and the motor-driver and buzzer pins as outputs. The vehicle counters are initialized, and the controller begins reading sensor states. A reliable startup routine should leave the motor disabled until the sensors have produced valid readings and the system has entered a known state.

B. Vehicle detection and counting

Each IR sensor monitors its assigned lane. When the sensor detects a passing object, the controller records a change in the digital signal. The project code notes use LOW as the detected state and increment the counter on a LOW-to-HIGH transition. This transition-based approach is intended to register a passage rather than increment repeatedly while the object remains in front of the sensor. However, sensor noise or a long interruption can cause missed or duplicate counts. Debouncing or a defined re-arming interval may be required.

C. Traffic comparison

The controller compares the two lane counters to determine whether one lane has reached a programmed threshold and has more counted vehicles than the other. The current project logic uses a threshold of approximately ten vehicles before it considers movement. A threshold can prevent the divider from reacting to every small difference, but its suitability depends on how counts are collected and reset. A more complete implementation should calculate demand over a fixed time window and avoid making decisions based only on counters that continue to increase indefinitely.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211

D. Distance monitoring

Before and during movement, the controller checks an ultrasonic distance reading. If the monitored distance is at or below the configured stop distance, the motor command is stopped. The project code notes use a stop distance of approximately 13 cm and a maximum movement time of approximately 1.5 seconds. These are prototype parameters, not universally safe values. They should be calibrated for the model, and real equipment would need a larger, validated safety envelope.

E. Motor control

When the lane-count condition and distance condition permit movement, the Arduino sets the L298N direction inputs. The enable input must be given a valid control value. A command such as `analogWrite(ENA_PIN, 255)` requests full PWM duty cycle on a compatible PWM pin; the actual value must match the intended speed and hardware configuration. The current pin allocation and the module wiring should be checked against the specific Arduino board and driver. If the motor is energized, movement should be monitored until a stop condition is reached.

F. Stop and return to monitoring

Movement ends when the distance condition indicates an unsafe proximity or when the maximum movement-time limit is reached. The controller then disables the motor and resumes monitoring. A time limit can reduce the chance of continuous movement if a sensor fails, but it does not measure the actual divider position. For more precise control, position sensors, limit switches, or a rotary encoder should be added. The software should also define a safe response if sensor data is missing or invalid.

G. Simplified control logic

Step	Condition or action	Expected result
1	Read both IR sensors.	Update lane counters when a passage is registered.
2	Read ultrasonic distances.	Obtain distance estimates for monitored sides.
3	Compare lane counts.	Determine whether a programmed imbalance exists.
4	Check distance and movement limit.	Prevent or stop movement when a stop condition is reached.
5	Set motor-driver direction and enable.	Move the model toward the selected side.
6	Disable motor and report status.	Return to a monitoring state.

H. Prototype Control Parameters

The documented code notes specify a movement time limit of around 1500 milliseconds and a distance stop threshold of around 13 centimetres. These values should be reported as implementation settings only. They do not represent a validated road-safety standard, and the system's movement direction must be confirmed physically because a software label such as 'left' or 'right' depends on motor wiring and the orientation of the model.

VIII. EXPERIMENTAL SETUP AND OBSERVATIONS

The project report describes testing the prototype under simulated conditions. The tests focus on whether the sensors produce usable readings, whether vehicle passage is registered, whether the controller compares the two lanes, and whether the divider model can move in the commanded directions. The available description is qualitative; it does not provide a complete set of numerical measurements for detection accuracy, response time, distance error, or congestion reduction.

A. Test setup

The prototype setup consists of an Arduino Uno connected to two IR sensors, two ultrasonic sensors, an L298N motor driver, a DC motor, and a buzzer. The motor is coupled to a small movable-divider mechanism. The Arduino IDE is used to upload the program, and the Serial Monitor is used to inspect readings and messages. During testing, simulated vehicle passages are presented to the IR sensors, while nearby objects can be used to check the ultrasonic distance response.

B. Experimental observations

The observations below summarize the qualitative behaviour described by the project report and the intended responses of the control logic. The table does not claim that every condition was tested over a statistically meaningful number of

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211

trials. Before a formal paper submission, the project team should record the actual number of trials, measured results, environmental conditions, and any failures.

No.	Experimental condition	Expected or reported response	Evidence to record
1	Power is applied to the system.	Controller starts and sensor readings can be observed.	Startup status and any error messages.
2	A vehicle passes the first IR sensor.	The first lane count is updated by the programmed transition logic.	Number of passages and displayed count.
3	A vehicle passes the second IR sensor.	The second lane count is updated.	Number of passages and displayed count.
4	Both lane counts are similar or below threshold.	The divider remains stationary according to the control logic.	Motor state and lane counts.
5	Lane 1 count meets the threshold and exceeds Lane 2.	The controller requests movement in the configured direction if distance conditions permit.	Direction command and observed movement.
6	Lane 2 count meets the threshold and exceeds Lane 1.	The controller requests movement in the opposite direction if conditions permit.	Direction command and observed movement.
7	An object is closer than the stop threshold.	Movement should be stopped or prevented by the distance condition.	Distance reading and motor state.
8	Movement-time limit is reached.	The controller stops the motor command.	Elapsed time and final motor state.

C. Interpretation

The qualitative observations indicate that the proposed system integrates the main functions needed for a small proof-of-concept: detection, comparison, actuation, and stopping logic. The design can therefore be used to demonstrate a simple automated response to an imbalance in lane counts. However, the observations do not provide enough evidence to quantify performance improvement. The paper should not state a percentage reduction in congestion or fuel consumption unless those values are measured using a documented experimental method.

D. Recommended Measurements

- Vehicle-count accuracy: compare sensor counts with a manually recorded ground-truth count over repeated trials.
- Distance error: compare ultrasonic readings with measured reference distances at several points.
- Response time: record the time from a valid traffic-imbalance condition to motor activation.
- Movement repeatability: measure the divider's final position across repeated left and right movements.
- Stop performance: test whether the motor reliably stops at the defined distance and time limits.
- Reliability: record false detections, missed detections, failed movements, and sensor timeouts.

IX. PROTOTYPE RESULT IMAGES AND EXPLANATION

The following photographs are taken from the supplied project report. They illustrate the physical prototype and the two divider movement positions. These images provide visual evidence of the model setup and movement demonstration; they do not represent numerical measurements of traffic reduction, fuel savings, or travel-time improvement.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211



Fig. 1 Initial setup of the prototype

Explanation: The initial setup shows the Arduino and connected electronic components mounted near the road model. The IR sensors are positioned to detect vehicle passage, while the ultrasonic sensors are used for distance monitoring. The DC motor and L298N driver provide the movement mechanism for the divider model.



Fig 3 Road divider moves towards the Right side

Figure 3 shows the divider movement toward the right side, which occurs when the IR sensors detect more vehicles on the left lane compared to the right lane. Once the Arduino

Figure 2. Divider moved toward the right

Explanation: This photograph shows the divider in the right-side position. It illustrates that the motorized mechanism can move the model in one direction when the controller issues the corresponding command.



Fig 4 Road divider move towards the Left side

Figure 3. Divider moved toward the left

Explanation: This photograph shows the divider in the left-side position. It illustrates movement in the opposite direction. The actual lane-selection direction must be confirmed against the sensor placement and motor wiring.

Result interpretation: The photographs support a qualitative demonstration that the prototype can be assembled and shown in different divider positions. To report system performance quantitatively, the team should additionally record actual vehicle counts, distance readings, response time, movement duration, stop success, and repeated-trial results. No performance percentages should be claimed unless they are measured.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211

X. DISCUSSION

The prototype illustrates how inexpensive sensors and an embedded controller can be combined to demonstrate adaptive divider movement. The IR sensors provide a simple method of detecting passage, while the ultrasonic sensors add a distance-related input. The Arduino Uno provides the logic for comparing lane counts, and the L298N module allows the motor direction to be controlled without driving the motor directly from the microcontroller pins. These elements are widely used in educational automation projects and provide a practical platform for experimentation.

The main strength of the design is its modular structure. The sensing elements, control program, motor driver, and alert mechanism can be tested separately. This simplifies troubleshooting and makes it possible to improve one part without replacing the entire system. The design also gives students experience with digital input handling, distance measurement, motor control, program logic, and serial debugging.

There are important limitations. First, an IR sensor does not inherently understand a vehicle; it only detects a change in its sensing field. Sensor placement and the shape of the object can affect whether one vehicle is counted once or more than once. Second, a cumulative count may not represent current traffic density unless the count is reset or evaluated over a defined time interval. Third, ultrasonic readings can fluctuate, and a single threshold cannot guarantee that the full movement path is clear. Finally, time-based motor operation is not equivalent to measuring the divider's actual position.

XI. ADVANTAGES

- Demonstrates the concept of changing lane-space allocation according to detected traffic imbalance.
- Uses accessible educational hardware for a low-cost proof-of-concept.
- Combines vehicle detection, distance sensing, motor control, and audible alerts in one system.
- Allows the program to be monitored and debugged through the Arduino Serial Monitor.
- Provides a platform for future extensions such as wireless reporting, improved sensing, and data logging.

XII. LIMITATIONS AND SAFETY CONSIDERATIONS

- The current prototype is not demonstrated as a road-ready or safety-certified system.
- IR sensor counts may be affected by placement, object spacing, ambient conditions, and sensor noise.
- Ultrasonic distance estimates can fluctuate and may not detect every object or surface reliably.
- Time-based movement does not confirm the divider's exact physical position.
- The project report does not provide quantitative proof of congestion reduction, fuel savings, or travel-time improvement.
- A full-scale system would require mechanical engineering, redundant sensing, emergency stop controls, secure control logic, operator override, weatherproofing, and approval by road-safety authorities.

XIII. FUTURE ENHANCEMENTS

Future development should begin with improved measurement and safety rather than adding complexity alone. Limit switches or encoders can provide feedback about the divider's position. A defined traffic-counting window can make lane-demand estimates more meaningful. Additional sensors placed along the movement path can improve obstacle awareness, while a physical emergency stop and a manual override can support safe testing.

A later version could include a wireless module for sending status information to a dashboard. A logging function could store counts, movement commands, distance readings, and errors for evaluation. Camera-based detection or other traffic sensors could be compared with IR counting to improve vehicle classification and density estimation. Predictive models may be explored only after a sufficiently large and reliable dataset has been collected. Any future system intended for public roads would need independent engineering validation and authorization.

XIV. CONCLUSION

This paper presents the design concept and prototype operation of a Smart Movable Road Divider System for Traffic Flow Management. The system combines an Arduino Uno, two IR sensors, two ultrasonic sensors, an L298N motor driver, a DC motor, and a buzzer. The IR sensors are used to register vehicle passage in two lanes, the controller compares the resulting counts, and the motor driver operates the divider model when programmed conditions are met. Ultrasonic

DOI: 10.17148/IJARCCE.2026.1511211

measurements provide a distance-related input, while the movement-time limit and buzzer support the prototype's control and indication functions.

The project demonstrates the feasibility of integrating sensing and motor control in a small educational model. It also highlights the practical issues that arise when a control decision affects a moving physical mechanism. In particular, reliable counting, stable distance measurement, position feedback, and safe stopping are important requirements. The qualitative observations in the project report are useful for describing prototype behaviour, but they do not establish numerical improvements in traffic flow, fuel use, or emissions.

The next stage should focus on repeated, documented testing. Vehicle-count accuracy, distance error, response time, movement repeatability, and stop performance should be measured and reported transparently. The motor and mechanical structure should be improved with position feedback, physical limits, fail-safe behaviour, and an emergency stop. With those improvements and formal safety assessment, the concept could be explored further as a research prototype for adaptive traffic management. It should not be deployed on a public road without the required engineering validation and regulatory approval.

ACKNOWLEDGEMENT

The authors acknowledge the guidance and support provided by Dr. Anushmita Pathak, Assistant Professor, Department of Electronics and Communication Engineering, East West College of Engineering, Bengaluru. The authors also acknowledge the institution and laboratory resources used to develop and examine the prototype.

DECLARATIONS

A. Funding source

No external funding details were provided in the source project report. The authors should confirm whether the work received institutional or other financial support.

DATA AVAILABILITY

The paper summarizes qualitative prototype observations from the project report. A numerical experimental dataset was not supplied with the source material. If detailed test logs are collected, they may be included in an appendix or made available on request, subject to the authors' confirmation.

A. Use of AI-assisted tools

If AI-assisted tools were used to prepare or edit the manuscript, the authors should disclose their use according to the requirements of the intended institution, conference, or journal. The authors remain responsible for verifying the technical content, data, and references.

REFERENCES

[1]. The following entries reflect works identified in the project report's literature discussion. Full titles, publication venues, years, page numbers, and DOI or official URLs were not available in the extracted details used to prepare this paper. These entries are placeholders and must be verified against the original publications before formal submission.
[2]. T. Ranjitha et al., work on automated lane adjustment based on real-time traffic density, as cited in the project report. Complete publication details to be verified.
[3]. I. M. Kamatar et al., work on an IoT-enabled movable road divider with emergency-vehicle priority, as cited in the project report. Complete publication details to be verified.
[4]. S. Arvind et al., work on an IoT-controlled adjustable road divider, as cited in the project report. Complete publication details to be verified.
[5]. S. S. Sai Sri Vastava et al., work on an automatic movable divider using Arduino Uno and NodeMCU, as cited in the project report. Complete publication details to be verified.
[6]. K. N. Kishore Kumar et al., work on a microcontroller-based automatic road divider, as cited in the project report. Complete publication details to be verified.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511211

APPENDIX A. SUGGESTED TEST RECORD SHEET

For a stronger experimental section, the team can complete the following record sheet during repeated tests. All values should be measured rather than estimated.

Trial	Lane 1 actual / counted	Lane 2 actual / counted	Distance (cm)	Response time (s)	Movement / stop result
1	To be measured	To be measured	To be measured	To be measured	To be recorded
2	To be measured	To be measured	To be measured	To be measured	To be recorded
3	To be measured	To be measured	To be measured	To be measured	To be recorded
4	To be measured	To be measured	To be measured	To be measured	To be recorded
5	To be measured	To be measured	To be measured	To be measured	To be recorded

APPENDIX B. PRE-SUBMISSION CHECKLIST

- Verify every reference against the original publication and complete the citation details.
- Insert actual measured data and describe the test setup, number of trials, and limitations.
- Check the final Arduino code for compile errors and confirm the pin mapping matches the circuit.
- Confirm all authors' names, registration numbers, department, and guide details are correct.
- Follow the exact template and declaration requirements of the target conference, journal, or institution.