

Published online on the journal's website: <https://jes-tm.org/index.php/jestm/index>

## Journal of Engineering Science and Technology Management

| ISSN (Online) 2828 -7886 |



Article

# Smart Adaptive Traffic Light Prototype Using Ultrasonic Sensors and Arduino

Firman Hamdi<sup>1,a✉</sup>, Joko Musridho<sup>2, b</sup>, Hanantatur Adeswastoto<sup>3,c</sup>

<sup>1,2</sup> Informatics Engineering Study Program, Faculty of Engineering, Pahlawan Tuanku Tambusai University, Riau

<sup>3</sup> Civil Engineering Study Program, Faculty of Engineering, Pahlawan Tuanku Tambusai University, Riau

DOI: 10.31004/jestm.v6i1.351

E-mail: [firmanhamdi78@gmail.com](mailto:firmanhamdi78@gmail.com)

### ARTICLE INFORMATION

Volume 6 Issue 1  
Received: 28 December 2026  
Accepted: 11 February 2026  
Published Online: 10 March 2026  
*On line:* at <https://JESTM.org/>

### Keywords

Smart adaptive traffic light  
Ultrasonic sensor  
Arduino  
Vehicle density  
Prototype

### ABSTRACT

Traffic congestion at urban intersections is often caused by traffic light systems that operate using fixed timing and are unable to respond to real-time traffic conditions. This study aims to design and test a smart adaptive traffic light prototype capable of adjusting green light duration based on vehicle density. The research employs a Research and Development (RnD) method with a prototype approach, involving system analysis, hardware and software design, prototype assembly, and functional testing using a scaled simulation model. Vehicle density is detected using HC-SR04 ultrasonic sensors, while an Arduino Mega 2560 microcontroller processes sensor data to control traffic light signals adaptively. The test results indicate that the proposed system is able to differentiate traffic density levels and dynamically adjust green light duration accordingly, leading to more efficient traffic flow compared to static timing systems. Although the implementation is limited to a prototype-scale simulation, the results demonstrate the potential of ultrasonic sensor-based adaptive traffic light systems as an alternative solution for improving traffic management at intersections.

## 1. Introduction

Based on initial observations, Bangkinang City is the center of government and economic activity for Kampar Regency in Riau Province. This area is home to several intersections, one of which is located at the intersection of Jalan Ahmad Yani and Jalan Jenderal Sudirman, which serves as a gateway to the city center. Traffic control at this location utilizes traffic lights to control the flow of vehicles from various directions. While these traffic lights help maintain the smooth flow and safety of road users, congestion often occurs during rush hour due to the high number of vehicles passing through.

Traffic lights, officially referred to as Traffic Signal Devices (APILL), are electronic control systems that regulate vehicle and pedestrian movements at intersections using red, yellow, and green signals. These devices are intended to ensure traffic safety, order, and smooth flow, particularly at conflict-prone intersections in urban areas. In Indonesia, the installation and operation of traffic lights are regulated under the Regulation of the Minister of Transportation of the Republic of Indonesia Number PM 49 of 2014 concerning Traffic Signal Devices. Despite their essential role, conventional traffic light systems generally operate using fixed time settings, which limits their ability to respond to fluctuating traffic conditions.

Traffic congestion occurs when the volume of vehicles exceeds the capacity of the road, causing traffic flow to slow down or stop entirely. This condition is commonly observed in urban areas and is influenced by several factors, including rapid growth in vehicle numbers, high mobility of people, limited road infrastructure expansion, and inefficient traffic management systems. In many cases, congestion is exacerbated by traffic control systems that are unable to dynamically adapt to real-time vehicle density, resulting in inefficient allocation of green light durations at intersections.

One of the main problems identified at the intersection of Jalan Ahmad Yani and Jalan Jenderal Sudirman in Bangkinang City is the use of a static traffic light timing system. This system applies uniform green light durations for all directions regardless of actual traffic conditions. During peak hours, such as morning and evening rush periods, certain routes experience significantly higher vehicle volumes, while other routes remain relatively uncongested. The inability of the existing system to adjust signal timing based on real-time traffic density leads to long vehicle queues on busy lanes and inefficient traffic flow, highlighting the need for an adaptive traffic light control solution.

This problem is especially pronounced during peak hours, such as when children return from school or when people return from work in the morning and evening. During these times, traffic from Jalan Sudirman is typically heavier because

this route is heavily used by students and workers. However, the traffic light system still provides a uniform green light duration for all directions, including on less congested routes, such as Jalan Ahmad Yani. This results in long queues of vehicles forming on busy routes, while smoother lanes continue to receive the same green light duration, leading to inefficient traffic management.

This discrepancy indicates that the current system is ineffective in optimally managing traffic flow. Therefore, innovation is needed in the form of an adaptive, sensor-based traffic light system design that can automatically and in real time adjust the light duration according to the level of vehicle density in each lane. With the implementation of this system, it is hoped that traffic flow at the intersection will be smoother, vehicle waiting times will be shorter, and congestion that often occurs during rush hour can be significantly minimized.

A responsive traffic light system can be an alternative solution to address traffic congestion. Supported by sensor and microcontroller technology, this system can detect the volume of vehicles stopped at intersections and then dynamically adjust the duration of red and green lights. One type of sensor used is an ultrasonic sensor, which has the ability to accurately detect vehicle presence. This sensor operates by emitting ultrasonic waves and capturing the reflections of these waves from surrounding objects. (Deltania et al., 2021).

Arduino, as an easy-to-program microcontroller platform, enables flexible and efficient system design. More complex traffic light control logic can be implemented using Arduino, including adjusting light timing based on sensor data. Furthermore, support from the extensive Arduino user community and the availability of extensive documentation facilitate the development and troubleshooting process. (Rifqo & Aprianti, 2020).

The implementation of this adaptive traffic light system is expected to reduce vehicle queue times and improve traffic flow. This research aims to design and test a prototype of an intelligent traffic light system based on ultrasonic sensors and Arduino, and evaluate its performance in reducing congestion at intersections.

## 2. Literature Review

### 2.1 Prototype

A prototype is an initial form or basic model for software development. In English, this term is known as a prototype, which is also often called a prototype. Generally, a prototype is created as an initial working example that serves as a demonstration or as part of a stage in the software development process. Etymologically, the word "prototype" comes from the Latin, namely "proto"

meaning original, and "typus" meaning form or model. Outside of a technical context, a prototype refers to a specific example that represents a particular category. In the world of design, a prototype or prototype is also known as an archetype, namely an initial form that serves as an example or primary reference for an entity. Prototypes are usually created before a product is further developed, or even specifically designed as part of the development stage before full-scale or mass production.(Kurniawan & Budi, 2022).

## 2.2 Traffic Light

*Traffic light*A traffic light is a device that regulates the movement of vehicles and pedestrians at intersections. This system is designed to improve traffic flow, reduce congestion, and prevent accidents. Generally, traffic lights consist of three primary colors: red, yellow, and green, each of which has a specific role in signaling stop, caution, and proceed.(Arvinda et al., 2025).

Traffic lights are a crucial element in a city's transportation system, serving to regulate and control the flow of vehicles on the road. According to Law No. 22 of 2009 concerning Traffic and Road Transportation, traffic lights are categorized as Traffic Signaling Devices (APILL). These lights are generally installed at strategic points such as intersections, pedestrian crossings, and other designated locations. Traffic lights aim to regulate the movement of vehicles in turn to prevent congestion and reduce the number of traffic accidents. These lights use three internationally recognized primary colors: green, yellow, and red.(Arfyanti, 2025).

## 2.3 SSmart Adaptive Traffic Light

A smart adaptive traffic light system is an innovation in traffic light management that can change the duration and pattern of light changes in real time according to traffic conditions detected by sensors, such as induction loop sensors installed on the road. This technology quickly collects vehicle data and adjusts the light phases to reduce congestion, improve traffic flow, and shorten vehicle waiting times at intersections. Simulation results using software such as Simulation of Urban Mobility (SUMO) show that adaptive traffic lights are very effective, especially in moderate congestion situations, with a clear improvement in performance compared to conventional traffic lights. However, in severe congestion situations, the effectiveness of this system can vary and requires additional adjustments to maintain its function properly. This technology is an element of smart city solutions that utilize the Internet of Things (IoT) to improve traffic management in urban areas.(Judge, 2024).

## 2.4 Density

Congestion is a condition in which traffic slows or even stops completely due to the number of vehicles exceeding the road's capacity. This phenomenon commonly occurs in large cities, especially those lacking adequate public transportation systems, ineffective traffic management, or an imbalance between road capacity and population density.(Rahmawati & Hariyanto, 2020).

This situation is triggered by a number of factors, including an increase in the number of vehicles that is not balanced with the expansion of the road network, high activity of people from outside the city operating within the city, and a less than optimal traffic management system in urban areas.(Rahmawati & Hariyanto, 2020).

## 2.4 HC-SR04 Ultrasonic Sensor

An ultrasonic sensor is a device that converts physical measurements (sound waves) into electrical signals and vice versa. This device operates on the principle of sound wave reflection, allowing it to determine the presence (distance) of an object using a specific frequency. It's called an ultrasonic sensor because it utilizes ultrasonic waves (sound waves inaudible to humans).

The HC-SR04 is a ready-to-use ultrasonic sensor that functions as a transmitter, receiver, and regulator of ultrasonic waves. This device can be used to measure the distance of objects from 2 cm to 4 m with an accuracy level of up to 3 mm. This tool is equipped with 4 pins: Vcc, Gnd, Trigger, and Echo. The Vcc pin serves to supply positive electricity, while Gnd is used for ground. The Trigger pin serves to output signals from the sensor, and the Echo pin is used to receive reflected signals from objects.(Azis et al., 2023).

## 3. Research Methodology

The applied method is Research and Development (RnD) with a prototype approach through the design, assembly, and simulation stages of a scaled prototype. Test results show that this system is able to regulate the duration of green lights based on vehicle density, thereby improving traffic flow at intersections..

### 3.1 Location and time of research

This research was conducted at the intersection of Jl. Ahmad Yani and Jl. Jendral Sudirman in Bangkinang City. Currently, the intersection still uses a manual timer system, resulting in inefficient traffic management and an inability to adapt to the number of vehicles. Therefore, during peak hours such as morning, afternoon, and evening, vehicle congestion often occurs, which can lead to traffic jams and accidents.

### 3.2 Research subjects

In this study, the subjects studied consisted of people and parties directly involved in the management and regulation of the traffic light system at the intersection of Jalan Ahmad Yani and Jalan Jendral Sudirman in Bangkinang City.

### 3.3 System development stages

The stages of system development include:

#### Analysis and Identification

The initial phase began with direct field observations to understand traffic conditions and vehicle movement patterns that frequently cause congestion. Interviews were conducted with the Transportation Agency, traffic light technicians, and road users to obtain information about the problems encountered and the needs of the new system. From this phase, the required components were identified, both in terms of hardware and control systems.

#### System Design

At this stage, the overall system design is carried out, including hardware and software integration. The hardware consists of an HC-SR04 ultrasonic sensor as a vehicle density detector, an Arduino Mega 2560 microcontroller that functions as the main control brain, LEDs as a traffic light simulation, a breadboard for assembling temporary circuits, and jumper cables as connectors between components. Meanwhile, the software is designed using the Arduino IDE to process data from the sensor, set the logic for changing the lights, and control the LEDs so that they can light up according to the detected vehicle density level. This system is designed to work automatically and can be reset according to conditions in the field.

#### Prototype Assembly

All designed components were then assembled into a single system. Ultrasonic sensors were positioned along the vehicle path as detectors, then connected to an Arduino Mega to process and control the lights. LEDs were installed as traffic light indicators (red, yellow, and green) that would alternately light up according to the settings. Assembly was performed on a breadboard for easy reconfiguration, with jumper cables connecting the components. This prototype was miniaturized to facilitate simulation and testing of the system's functionality.

#### Testing

The final stage is prototype testing, simulating the presence of a vehicle using a toy car. Ultrasonic sensors detect distance, and the Arduino Mega processes the data to automatically adjust the light duration. Testing focused on the system's rapid

response, sensor reading accuracy, and effectiveness in reducing vehicle queues. Analysis of the test results was used to evaluate the system's performance in supporting smooth traffic flow and the smart adaptive traffic light concept.

### 3.4 Data collection technique

Data collection in this study was carried out through literature studies, direct observations at the research location, and interviews with related parties as a basis for designing a smart adaptive traffic light prototype.

### 3.5 Validity techniques

In this study, data validity testing was conducted to ensure that the data obtained had a high level of validity and reliability. One method used was triangulation, an approach that combines various data collection and analysis methods. Through this technique, researchers were able to obtain more in-depth and objective information regarding traffic conditions at intersections and the effectiveness of implementing an adaptive traffic light system based on ultrasonic sensors and Arduino.

### 3.6 Data analysis

Data analysis was carried out descriptively by classifying data based on vehicle density problems, traffic light system requirements, and prototype testing results, and the results were used to assess the performance of the smart adaptive traffic light system based on ultrasonic sensors and Arduino.

## 4. Results and Discussion

### 4.1 Block diagram design

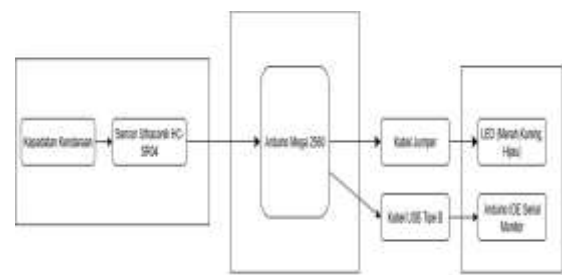
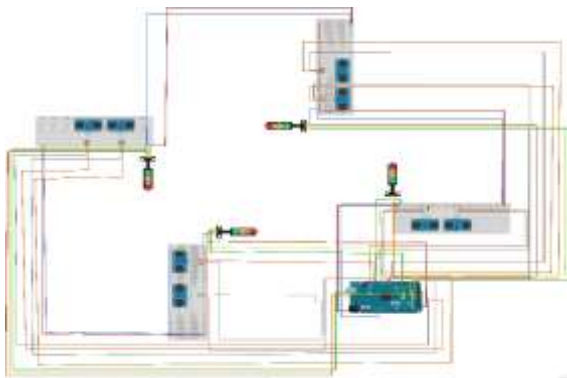


Figure 1. System block diagram

The system block diagram illustrates the integration of the HC-SR04 ultrasonic sensor in detecting vehicle distance with the Arduino Mega 2560 as the main controller that processes traffic density data to regulate the red, yellow, and green LED lights, and displays the results of vehicle distance readings directly via the Arduino IDE Serial Monitor.

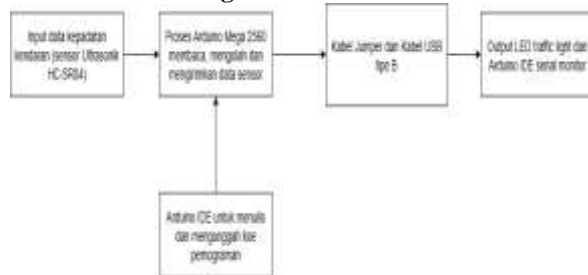
## 4.2 Hardware design



**Figure 2.**Hardware

The figure shows the system hardware design, where the HC-SR04 ultrasonic sensor functions to detect the distance of vehicles to determine the level of traffic density. traffic, then the data is processed by the Arduino Mega 2560 to control the red, yellow, and green LEDs as a representation of traffic lights, with a breadboard and jumper cables as assembly media and a USB cable as a means of programming and power supply.

## 4.3 Software design



**Figure 3.** Software architecture

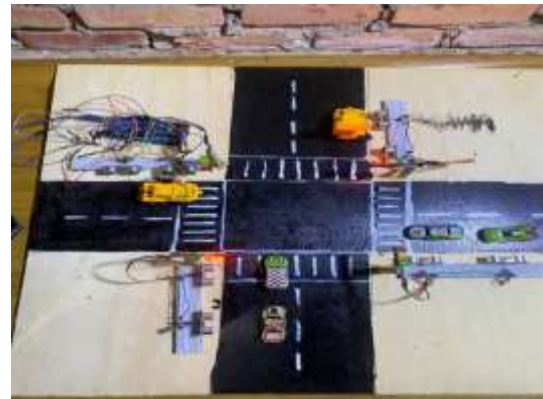
The software used in this system is a program installed on the Arduino Mega 2560 microcontroller and the Arduino IDE application as a place to compile and upload program code. Software development is carried out by utilizing the Arduino IDE because of its open nature, supports various types of microcontroller boards including the Arduino Mega 2560, and provides a complete and easy-to-use library. The programming language used is C/C++ with a structure and syntax that has been adapted for microcontrollers. This program is designed to retrieve data from the HC-SR04 ultrasonic sensor, process distance data into information about vehicle density, and regulate LED lights (red, yellow, green) as a simulated traffic light

output, which is also displayed on the Arduino IDE Serial Monitor.

## 4.4 Hardware assembly

This design is carried out by connecting each component, starting with connecting the Arduino Mega microcontroller to several breadboards and the sensors used. The reason for using a breadboard for this circuit is because the number of pins on the Arduino Mega microcontroller is not enough to connect all the necessary components. The breadboard serves as a connection between the components via jumper cables.

Before connecting the HC-SR08 ultrasonic sensor, the sensor was first connected to a breadboard. The sensor was then connected using jumper cables through the breadboard to an Arduino Mega. In this study, eight ultrasonic sensors were used to detect vehicles in the north, south, west, and east.



**Figure 4.**Ultrasonic Sensor Configuration

## 4.5 Software assembly

The software design used to regulate the duration of the green traffic light controlled by a microcontroller based on sensors using the Arduino IDE (Integrated Development Environment) application. Version 1.8.19. from Arduino IDE. The programming language implemented in the Arduino IDE is based on the C++ programming language. The programming process is carried out on the Arduino Mega which functions as a processor for the duration of the green light based on the sensors and logic in the software.

Below is a program code table that explains the programming functions carried out using the Arduino IDE software and then entered into the Arduino Mega microcontroller.

**Table 1.**Arduino Mega Program Code in Arduino IDE

Program code	Explanation
<pre>// North Line const int trigNorth1 = 2, echoNorth1 = 3; const int trigNorth2 = 4, echoNorth2 = 5; const int redNorth = 22, yellowNorth = 24, greenNorth = 26; // East Line const int trigEast1 = 41, echoEast1 = 43; const int trigEast2 = 28, echoEast2 = 30; const int redEast = 36, yellowEast = 34, greenEast = 32; // South Line const int trigSouth1 = 44, echoSouth1 = 46; const int trigSouth2 = 12, echoSouth2 = 11; const int redSouth = 53, yellowSouth = 38, greenSouth = 52; // West Route const int trigWest1 = 6, echoWest1 = 7; const int trigWest2 = 8, echoWest2 = 13; const int redWest = 40, yellowWest = 42, greenWest = 10;</pre>	<p>using two ultrasonic sensors (trig/echo) and three LED lights (red, yellow, green). TrigNorth1 and echoNorth1 for the first sensor, TrigNorth2 and echoNorth2 for the second sensor. And so on for South, West, and East</p>
<pre>const int threshold = 10; const int minHijau = 3000; // Minimum 3 seconds const int maxHijau = 15000; // Maximum 15 seconds long duration; int CurrentPath = 0; // 0: North-South, 1: East-West</pre>	<p>threshold: distance threshold (in cm), if it is closer than this value it means there is a vehicle. minGreen &amp; maxGreen: duration limit for green light to stay on. CurrentLine: turn marker for which lane is active (alternating).</p>
<pre>void setup() { Serial.begin(9600); pinMode(trigNorth1, OUTPUT); pinMode(echoNorth1, INPUT); pinMode(trigNorth2, OUTPUT); pinMode(echoNorth2, INPUT); pinMode(trigEast1, OUTPUT); pinMode(echoEast1, INPUT); pinMode(trigEast2, OUTPUT); pinMode(echoEast2, INPUT); pinMode(trigSouth1, OUTPUT); pinMode(echoSouth1, INPUT); pinMode(trigSouth2, OUTPUT); pinMode(echoSouth2, INPUT); pinMode(trigWest1, OUTPUT); pinMode(echoWest1, INPUT); pinMode(trigWest2, OUTPUT); pinMode(echoWest2, INPUT); // Light pinMode(redNorth, OUTPUT); pinMode(yellowNorth, OUTPUT); pinMode(greenNorth, OUTPUT); pinMode(redEast, OUTPUT); pinMode(yellowEast, OUTPUT); pinMode(greenEast, OUTPUT); pinMode(SouthRed, OUTPUT); pinMode(SouthYellow, OUTPUT); pinMode(SouthGreen, OUTPUT); pinMode(redWest, OUTPUT); pinMode(yellowWest, OUTPUT); pinMode(greenWest, OUTPUT); }</pre>	<p>Set the sensor and light pins as input/output. Setting up serial communication for distance and status monitoring.</p>

Program code	Explanation
<pre> void loop() {   Serial.println("=== Read sensor ===");   int  statusNorth  =  hitungStatus(getDistance(trigNorth1, echoNorth1), getDistance(trigNorth2, echoNorth2));   int  statusSouth  =  calculateStatus(getDistance(trigSelatan1, echoSelatan1), getDistance(trigSelatan2, echoSelatan2));   int  statusEast   =  calculateStatus(getDistance(trigEast1, echoEast1), getDistance(trigEast2, echoEast2));   int  statusWest   =  calculateStatus(getDistance(trigWest1, echoWest1), getDistance(trigWest2, echoWest2));   int  GreenDurationNS = getMaxGreenDuration(Northstatus, Southstatus);   int  GreenDurationTB =  getMaxGreenDuration(Eaststatus, Weststatus);   // Reset all red lights   setNorthLight("RED");   setSouthLamp("RED");   setLampuTimur("RED");   setWestLamp("RED");   if (currentpath == 0) {     Serial.println("== NORTH &amp; SOUTH GREEN ==");     setNorthLight("YELLOW");    setSouthLight("YELLOW");     delay(1000);     setNorthLight("GREEN"); setSouthLight("GREEN");     showCountdown(durationGreenNS);     setNorthLight("YELLOW");    setSouthLight("YELLOW");     delay(2000);     setNorthernLights("RED"); setSouthLight("RED");   } else {     Serial.println("== EAST &amp; WEST GREEN ==");     setLampuTimur("YELLOW");    setLampuBarat("YELLOW");     delay(1000);     setLampuTimur("GREEN"); setLampuBarat("GREEN");      displayCountdown(durationGreenTB);     setLampuTimur("YELLOW");    setLampuBarat("YELLOW");     delay(2000);     setLampuTimur("RED"); setLampuBarat("RED");   }   // Switch to the next pair   CurrentPath++;   if (CurrentPath &gt; 1) CurrentPath = 0;   delay(1000); } </pre>	<p>The main function that will continue to be carried out.</p> <p>Read all sensors for each path.</p> <p>Calculate traffic status per lane.</p> <p>Determining the duration of green depends on whether it is dense or not.</p> <p>Turns on the green light at the turn of the lane and displays the countdown.</p> <p>Once done, move on to the next path.</p>

Program code	Explanation
<pre>int calculateStatus(int distance1, int distance2) {   Serial.print(" Distance 1: "); Serial.print(distance1);   Serial.print(" cm   Distance 2: "); Serial.print(distance2);   bool s1 = (distance1 &lt; threshold);   bool s2 = (distance2 &lt; threshold);   if (s1 &amp;&amp; s2) {     Serial.println(" =&gt; Status: 2 (solid)");     return 2;   } else if (s1    s2) {     Serial.println(" =&gt; Status: 1 (medium)");     return 1;   } else {     Serial.println(" =&gt; Status: 0 (empty)");     return 0;   } }</pre>	
<pre>int getMaxGreenDuration(int status1, int status2) {   int statusMax = max(status1, status2);   int durationGreen;   if (Maxstatus == 2) durationGreen = 10000;   else if (Maxstatus == 1) Greenduration = 5000;   else durationGreen = 3000;   // Minimum and maximum limits   if (durationGreen &lt; minGreen) durationGreen = minGreen;   if (durationGreen &gt; maxGreen) durationGreen = maxGreen;   return durationGreen; }</pre>	<p>Assess track conditions based on two sensor readings:</p> <ul style="list-style-type: none"> <li>• If both detect vehicles → solid status (2)</li> <li>• If one detects → medium status (1)</li> <li>• If there is no vehicle → empty status (0)</li> </ul>
<pre>void displayCountdown(int durationMs) {   int timeRemaining = durationMs / 1000;   unsigned long StartTime = millis();   while (millis() - startTime &lt; durationMs) {     int secondsRunning = (millis() - StartTime) / 1000;     int remaining = remaining time - running seconds;     Serial.print(" &gt;&gt; Green time remaining: ");     Serial.print(remaining);     Serial.println(" seconds");     delay(1000);   } }</pre>	<p>Displays the remaining time for the green light in real-time on the serial monitor. Update every second until time runs out.</p>

Program code	Explanation
<pre>void setNorthernLights(String color) {   digitalWrite(redNorth, color == "RED");   digitalWrite(NorthYellow, color == "YELLOW");   digitalWrite(Northgreen, color == "GREEN"); } void setSouthLights(String color) {   digitalWrite(southred, color == "RED");   digitalWrite(yellowSouth, color == "YELLOW");   digitalWrite(hijauSouth, color == "GREEN"); } void setLampuTimur(String color) {   digitalWrite(redEast, color == "RED");   digitalWrite(yellowEast, color == "YELLOW");   digitalWrite(eastgreen, color == "GREEN"); } void setWesternLights(String color) {   digitalWrite(merahWest, warna == "RED");   digitalWrite(yellowWest, color == "YELLOW");   digitalWrite(hijauWest, color == "GREEN"); }</pre>	<p>Adjust the light according to color (red, yellow, green). Only one color is active at a time on each lane.</p>
<pre>int getDistance(int trigPin, int echoPin) {   digitalWrite(trigPin, LOW);   delayMicroseconds(2);   digitalWrite(trigPin, HIGH);   delayMicroseconds(10);   digitalWrite(trigPin, LOW);   duration = pulseIn(echoPin, HIGH, 30000);   int dist = duration * 0.034 / 2;   return dist; }</pre>	<p>Function to read distance from ultrasonic sensor. Using trig pulses and receiving reflections in echo. Returns results in centimeters.</p>

#### 4.6 Testing

Testing is carried out by providing vehicle density conditions at various object distances and observing the system's response to whether it is in accordance with the established logic. The testing method used includes two main stages, namely testing the ultrasonic sensor components simultaneously to ensure the sensor recognizes vehicles based on distance, as well as the overall system testing scheme to evaluate the system's response to, for example, vehicles. Sensor testing is carried out first as a basis for hardware validation whether the ultrasonic sensor can determine vehicles based on their minimum distance, while the testing scheme is designed to test the system's logic in determining how long the green light is on between the north and south lights, and also the west and east lights.

#### 4.6.1 Ultrasonic sensor testing

Ultrasonic sensor testing is carried out to ensure that each sensor installed on the North, South, East, and West lines can read vehicle distances. In this system, each lane is equipped with two HC-SR04 sensors connected to an Arduino Mega microcontroller. These sensors detect vehicle presence based on distance, which is then used to determine traffic density in each lane.

The program code in this system uses the `getDistance(trigPin, echoPin)` function to read the distance from each sensor. This distance data is then compared to a threshold value of 10 cm. If the detected distance is less than the threshold, the vehicle is considered to be in that lane.

During testing, the vehicle or test object is placed at various positions in front of the sensor with gradual distances (e.g., 5 cm, 10 cm, 20 cm, up to 100 cm). The distance reading results are displayed via the Serial Monitor with the help of the

Serial print() function, so that the values read by the sensor can be directly observed. In addition, the code also includes a display Countdown() function that displays the duration of the active green light as

a form of feedback on the system's decision regarding traffic density.

The following is a table of tests carried out in simulation based on the objects placed in the sensor:

**Table 3.**HC-SR04 Ultrasonic Sensor Simulation Test Table Object vehicles in the north and west

No.	Sensor	Sensor distance 1	Sensor distance 2	Sensor distance 1	Distance 2	Density status	Length of green light
1	North and South	2cm	25 cm	55cm	53cm	currently	5 seconds
2	East and West	55 cm	25cm	2cm	49cm	currently	5 seconds

**Table 4.**HC-SR04 Ultrasonic Sensor Simulation Test Table Object of vehicles in the west

No.	Sensor	Sensor distance 1	Sensor distance 2	Sensor distance 1	Distance 2	Density status	Length of green light
1	North and South	2cm	25 cm	55cm	53cm	currently	3 seconds
2	East and West	2 cm	2cm	2cm	4cm	congested	10 seconds

**Table 5.**HC-SR04 Ultrasonic Sensor Simulation Test Table Without Vehicle Object

No.	Sensor	Sensor distance 1	Sensor distance 2	Sensor distance 1	Distance 2	Density status	Length of green light
1	North and South	20cm	25 cm	55cm	53cm	Empty	3 seconds
2	East and West	55 cm	25cm	2cm	49cm	Empty	3 seconds

## 5. Conclusion

This study successfully designed and implemented a smart adaptive traffic light prototype based on HC-SR04 ultrasonic sensors and an Arduino Mega 2560 microcontroller. The developed system is capable of detecting vehicle presence on each lane, classifying traffic density levels, and automatically adjusting the green light duration according to the detected conditions. Prototype-scale testing results indicate that the system operates in accordance with the designed control logic and responds adaptively to changes in vehicle density.

However, this research is subject to several limitations. The system evaluation was conducted using a scaled prototype simulation and has not yet been implemented under real traffic conditions. In addition, the system relies on a single type of sensor, and environmental factors that may affect sensor performance have not been fully examined. Future studies are recommended to conduct field testing, integrate additional sensing technologies or Internet of Things (IoT)-based systems, and develop more advanced control algorithms to improve system reliability and scalability.

## References

- Alfayed, M., & Purnomo. (2024). Prototype of an Automatic Cat Feeder Based on the Internet of Things. *JATI (Journal of Informatics Engineering Students)*, 8(1), 937–944. <https://doi.org/10.36040/jati.v8i1.8847>
- Arfyanti, I. (2025). *Prototype Of Gps-Based Adaptive Traffic Light System For Emergency*. 29(1), 1–6. <https://doi.org/10.46984/sebatik.v29i1.0000>
- Arvinda, SD, Purnamasari, E., Ramadhan, AF, & Sah, A. (2025). Simulation of 4-Way Intersection Traffic Lights for Vehicle Flow Optimization Using Omron CP1L Programmable Logic Controller (PLC).
- Azis, A., Emidiana, E., & Azukruf, F. (2023). Design of an Automatic Goods Carrying Robot Prototype Based on Arduino Uno DIP with Ultrasonic Sensor. *Elektrika*, 15(1), 29. <https://doi.org/10.26623/elektrika.v15i1.5924>
- Bhargawa, INB, Udayana, AAGB, & Artawan, CA (2022a). Visual Identity Redesign for the Tetuek Sangmong Brand in Denpasar. *Amarasi: Journal of Visual Communication Design*, 3(02), 192–204.
- Bhargawa, INB, Udayana, AAGB, & Artawan, CA (2022b). Visual Identity Redesign for the

- Tetuek Sangmong Brand in Denpasar. *Amarasi: Journal of Visual Communication Design*, 3(02), 192–204. <https://doi.org/10.59997/amarasi.v3i02.1685>
- Cahyono, BD, & Talia, N. (2021). Development of Traffic Light Simulation Learning Media Using Proteus and Arduino. *Journal of Elementary Education and Social Humanities*, 1(1), 65–70.
- Deltania, DO, Djuniadi, D., & Apriaskar, E. (2021). Traffic Light Control with Ultrasonic Sensors. *Jetri: Scientific Journal of Electrical Engineering*, 19(1), 77–95. <https://doi.org/10.25105/jetri.v19i1.8660>
- Ghozali, MR, & Putra, A. (2025). PEOPLE WHO ENTER A ROOM.
- Judge, B. (2024). SIMULATION OF AN ADAPTIVE TRAFFIC LIGHT SYSTEM USING INDUCTION LOOP SENSOR BAIHAQI HAKIM, Dr. Raden Sumiharto, S.Sc., M.Kom.
- Kartiria Kartiria, Erhaneli Erhaneli, CYW (2021). Application of Arduino Mega 2560 Microcontroller as Monitoring for 3-Phase Current Readings at the 150 kV Lubuk Alung Substation. *Journal of Electrical Engineering*, 10(1), 37–45.
- Ministry of Transportation. (2014). Regulation of the Minister of Transportation of the Republic of Indonesia Number PM 49 of 2014 concerning Traffic Signal Devices. *Minister of Transportation of the Republic of Indonesia*, 1–27.
- Kurniawan, R., & Budi, M. (2022). UI/UX Design of Research Management and Community Service Applications Using Figma Application. *Information Systems Journal*, 5(1), 2–7.
- Prayitno, A., & Irham, M. (2023). Designing a web-based inventory system for Raphael's Divan. *Journal of Informatics Engineering Publication*, 2(1), 26–43.
- Rahmawati, I., & Hariyanto, T. (2020). Automatic Traffic Light System Using Radio Network at 433 MHz Frequency with Ultrasonic Sensor. *Proceedings of the 11th Industrial Research Workshop and National Seminar*, 607–612.
- Rai, K. M., & Vijayalakshmi, D. S. (2023). Density Based Traffic Light Control Using IR Sensors and Arduino. *International Journal for Research in Applied Science and Engineering Technology*, 11(1), 777–783. <https://doi.org/10.22214/ijraset.2023.48560>
- Rifqo, MH, & Aprianti, H. (2020). Traffic Light Response System for Driver Violations Using Ultrasonics. *Journal of Scientific and Applied Informatics*, 3(1), 57–64.
- Rosyady, PA, Ikhsan, ZA, & Feter, MR (2022). Adaptive Traffic Light Prototype Based on Vehicle Queue Length Based on Arduino Uno. *CIRCUIT: Scientific Journal of Electrical Engineering Education*, 6(2), 173–186.
- Sholahuddin, MR, Kusuma, BH, Informatics, T., Teknik, F., Nusantara, U., & Kediri, P. (2025). Smart Traffic Light System for Priority Vehicles. 4, 633–640.
- Suppa, R., Muhallim, M., Paembonan, S., & Palopo, K. (2025). WATER CONTENT TESTING EQUIPMENT IN CHOCOLATE FRUIT. 13(1).
- Suryana, T. (2021). Object Detection System for Home Security Using Infrared Sensors. *Object Detection System for Home Security Using Infrared Sensors*, 1(1), 1–17. <https://repository.unikom.ac.id/68733/1/Object-Detection-System-for-Home-Security-Using-Infrared-Sensors.pdf>
- Talib, Z., & Rahaim, D.L.A. (2024). Smart Traffic System Using Infrared Sensors-based IoT. *Journal of Internet Services and Information Security*, 14(3), 29–41. <https://doi.org/10.58346/jisis.2024.i3.003>