Smart Traffic Light System At T-Junction Using Arduino and RFID

Muhammad Shamir Arshad Jehabar Sathique, Hashimah Ismail, Rajendran Sinnadurai, and
Rahmad Mohd Taib*

Faculty of Engineering and Life Sciences
Universiti Selangor (UNISEL), Bestari Jaya, Selangor, Malaysia

*Corresponding author: Rahmad Mohd Taib

ABSTRACT

As the city's population and the number of vehicles on the roads continue to increase, traffic congestion at intersections has become a significant issue for many major urban areas. One contributing factor to this problem may be the limitations of the current traffic light systems, which struggle to adapt to rapidly changing traffic conditions. This inadequacy not only exacerbates congestion but also leads to a rise in road accidents, making it especially difficult for emergency vehicles, such as ambulances and fire trucks, to navigate through traffic jams. In light of these challenges, there is a pressing need for a system that can alleviate traffic congestion, particularly during peak hours, and ensure a clear path for emergency vehicles. Existing literature often overlooks the critical issue of providing unobstructed routes for emergency vehicles during times of heavy traffic.To address these problems, an RFID-based system has been proposed to monitor and control traffic lights at intersections when an emergency vehicle is approaching. This technology enables emergency vehicles to bypass traffic jams more effectively. The proposed framework is demonstrated through an experimental setup using Arduino and LED displays, simulating a real-time traffic scenario. Simulation results indicate that the proposed framework significantly enhances the detection and management of

emergency vehicles, facilitating their passage through congested traffic during peak hours. By providing timely access for emergency vehicles, this system not only improves traffic flow but also contributes to saving lives during critical situations. Overall, this approach represents a promising advancement in traffic management technology, aiming to address the urgent need for more efficient systems in urban environments.

Keywords: Arduino, Emergency Vehicle, Smart Traffic Light System, RFID, Traffic Congestion

1. INTRODUCTION

Rapid transit and fast transportation systems are the backbone of any country's economic growth. If the classic traffic lights, known as Pre-set Cycle Time (PCT), do not meet the current condition, traffic jams will occur. To improve traffic flow at intersections and national development, it is ultimately vital to have a rapid, cost-effective, and efficient traffic signal smart system. In the meantime, the population of developing countries such as Malaysia is rapidly increasing. As the population grows, so does the number of automobiles on the road. As a result, there are a variety of issues such as major traffic congestion, violations of traffic rules, and even accidents. Furthermore, traffic congestion causes long waiting times, fuel waste, and financial waste. Traffic congestion, in particular, results in high pollution levels, which have an impact on the health of local citizens, commuters, and wildlife. In general, traffic congestion is linked to many additional traffic issues, such as the blocking of emergency vehicles. Specifically, traffic congestion frequently disrupts the path of emergency vehicles, which might turn fatal at any time. In addition, the

number of people who have died as a result of a delay in receiving medical care has increased.

Therefore, emergency vehicles such as ambulances and fire engines must arrive on time to avoid loss of human life. Assisting an emergency vehicle in moving out of traffic congestion is extremely important in today's traffic circumstances. Radio Frequency Identification technology (RFID) can be used to solve the problems listed above. RFID technology aids in the calculation of vehicle density to operate traffic signals. RFIDs of various ranges are used to detect emergency vehicles in general. When emergency vehicles are spotted, the particular lane is cleared to allow the emergency vehicles to pass more easily. In response to economic development, the number of vehicles on the road is increasing. This heavy traffic causes massive jams, which is a common occurrence in most cities throughout the world. Traffic congestion mostly happens in the mornings, before office hours, and in the evenings, after office hours. Massive jams always occur at any junction. Traffic congestion, in particular, frequently obstructs the flow of emergency vehicles' routes, which might be fatal. The main issue is, that to prevent the loss of human life, emergency vehicles such as ambulances and fire engines must arrive promptly. The traffic light system is a conventional way to solve the congestion problem. The current traffic signal system installed at any junction uses settings based on prior traffic counts that can be modified manually to create a pre-set traffic control plan. However, this system still cannot solve the congestion effectively. This project aims to propose a new traffic light system that can assist an emergency vehicle moving in or out of a T-junction more effectively. This study specifically to design and fabricate miniature traffic lights T-junction and propose a new traffic light control system using Arduino coding with RFID devices.

The major goal of this research is to provide a modern method for controlling traffic light systems at intersections using an RFID system. The system comprises an RFID reader that reads RFID signals from passing vehicles. The Arduino controller receives and processes these signals. These signals are processed by the Arduino, which subsequently makes the appropriate decisions for effective traffic control. The RFID module is made up of two parts: a transmitter and a receiver. The Arduino UNO receives the information from the RFID receiver and delivers it to the transmitter. Then the Arduino UNO estimates traffic density and adjusts the traffic signal accordingly. Various limitations in this project need to be considered as it moves further. To begin with, the Smart Traffic Light Control System is only applied on T-junction intersections. In addition, the system design is based on the image taken from available images by experiment. On the other hand, the timing and density of the traffic flow data are based only on the Persiaran Tengku Ampuan Rahimah traffic junction as illustrated in Figure 1.

(a) (b)

Figure 1. (a) Visual of T-junction, and (b) Satellite view of the location at Persiaran Tengku Ampuan Rahimah

The following processes are included in the image processing techniques used in this system. First, a background image with no traffic load was captured as a reference frame, and then live stream frames of the road were acquired from the web camera.

i. If numerous emergency vehicles are detected in a single lane, the lane with the nearest emergency vehicle to the traffic light is given priority.

ii. If an emergency vehicle is detected, the signal priority will be given to that lane.

iii. If no emergency vehicle is present, the signal priority is given to the lane with the highest traffic density.

iv. If traffic density is the same in all the lanes, no signal priority is given to any lane and the traffic signal will run according to the pre-set timer.

2. MATERIALS AND METHOD

2.1 Traffic Light System

The evolution of traffic light system has grown rapidly to reduce traffic congestion and focus vitally on optimizing the traffic flow in traffic junctions to give emergency vehicles priority. A program is developed with several conditions that adapt to various situations that could occur at traffic junction that leads to effective traffic flow. In this methodology section, the parameters, hardware, and software are shown in Table 1.

Table 1. Hardware and Software Modules

Hardware Modules	Software Modules
Arduino Uno Rev3 microcontroller	Arduino IDE 2.0 RC (2.0.0-rc9)
Arduino Nano	
LED Traffic signal & LED Indicator	
RFID Module	

2.2 Proposed Project Architecture

This project architecture consists of four hardware which are monitor, mouse controller, Arduino Uno microcontroller, and keyboard. The keyboard, mouse controller, and Arduino Uno microcontroller were connected to PC as an input device while the monitor will be a device to show the output result from the software. All the devices connected using USB 2.0 connector except for the monitor, which uses HDMI cable as a connector. All these features are demonstrated in figure 2.

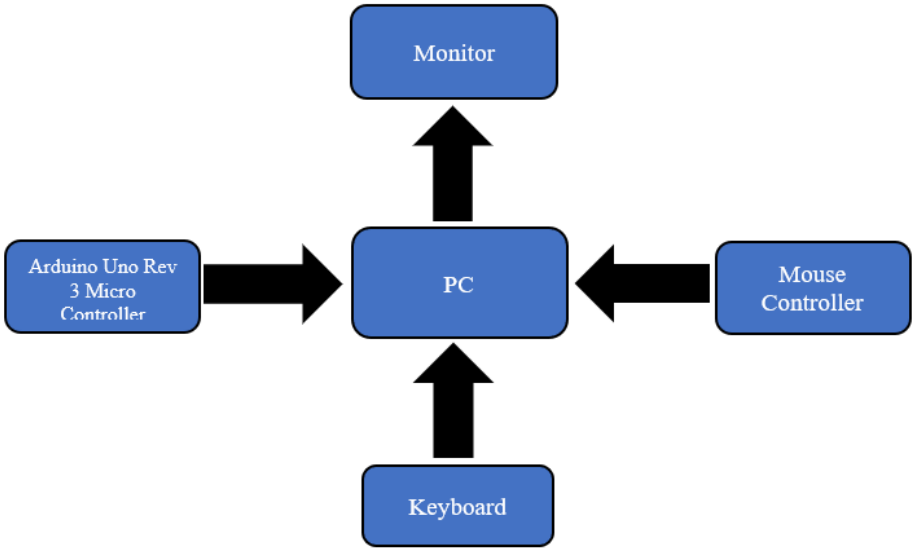


Figure 2. Block Diagram of the Proposed Framework

2.3 Software and hardware implementation

2.3.1 Arduino software

Arduino is an open-source platform for creating electronics projects. It consists of a physical programmable circuit board, often referred to as a microcontroller, and software known as an IDE (Integrated Development Environment) that runs on a computer to write and upload code to the board. The Arduino platform has become popular among beginners in electronics for several reasons. Unlike many earlier programmable circuit boards, Arduino eliminates the need for a separate programmer, allowing code to be loaded directly via a USB cable.

2.3.2 Install Arduino Software 2.0.0-rc9

The algorithm in Arduino Uno microcontroller is programmed using Arduino IDE 2.0 RC software. The Traffic light system is controlled by Arduino Uno board which programmed earlier using the most prominent application which is Arduino IDE 2.0 RC software. The program can be used both interactively and in batch mode using scripts

2.3.3 Arduino Uno Rev3 Micro controller Hardware

The Arduino UNO Rev3 is an ideal board for beginners in electronics and coding. It features a microcontroller based on the ATmega328P and includes 14 digital input/output pins, six of which can function as PWM outputs, along with six analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. This board contains everything needed to support the microcontroller, making it easy to start by connecting it to a computer via USB or powering it with an AC-to-DC adapter or battery.

In comparison to the Arduino Diecimila, the UNO can be reset from the computer, eliminating the need to manually press the reset button. The Diecimila uses a low dropout voltage regulator to reduce power consumption when using an external AC/DC adapter or battery. Additionally, it has a resettable poly-fuse to protect the computer's USB ports from shorts and surges. The board also includes pin headers for the reset line and for 3.3V, as well as a built-in LED on pin 13. Detailed specifications of the Arduino UNO Rev3 are provided in Table 2, and a model of the microcontroller board is illustrated in Figure 3.

Table 2. Arduino UNO Rev3 Specification

Specifications	***Details***
Microcontroller :	*ATmega328P*
Operating voltage :	*5V*
Input voltage :	*7-12V*
Input voltage (limit) :	*6-20V*
Digital i/o pins :	*14 (of which 6 provide PWM output)*
Pwm digital i/o pins :	*6*
Analog input pins :	*6*
Dc current per i/o pin :	*20 mA*
Dc current for 3.3v pin :	*50 mA*
Flash memory :	*32 KB (ATmega328P)*
Sram :	*2 KB (ATmega328P)*
Eeprom :	*1 KB (ATmega328P)*
Clock speed :	*16 MHz*
Led_built-in :	*13*
Length :	*68.6 mm*
Width :	*53.4 mm*
Weight :	*25 g*

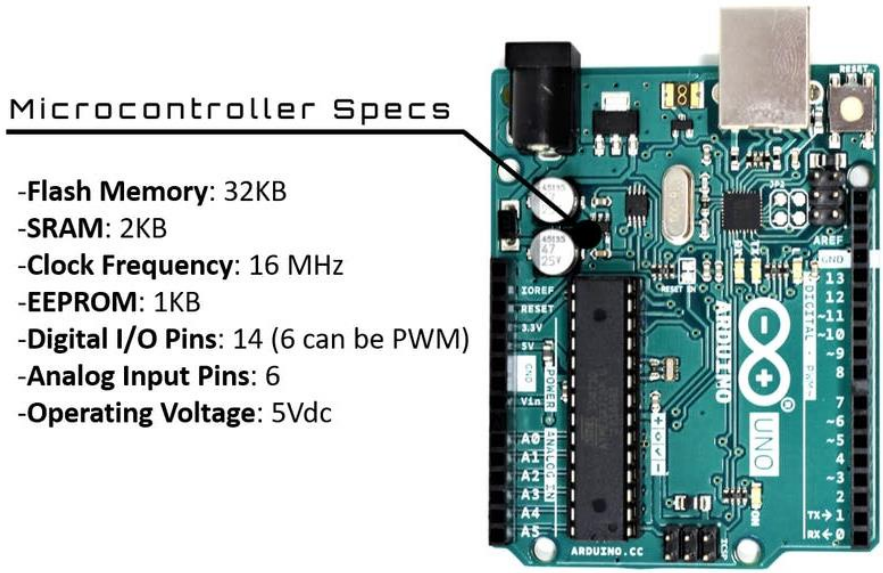


Figure 3. Arduino UNO Rev3

2.4 System Working Principle

Figure 4 illustrates the block diagram of the proposed framework. The system features an RFID reader designed to capture signals from moving traffic. The primary objective of this system is to efficiently manage traffic flow while ensuring safe passage for emergency vehicles through the use of RFID technology. The Arduino controller plays a key role in this setup, receiving and processing the gathered signals. After collecting the data, the Arduino analyzes it to make informed decisions for effective traffic management. The RFID reader, positioned at a specific distance from the traffic signals, collects RFID signals from vehicles in the vicinity.

Using the Arduino, the system calculates the number of vehicles in each lane at the intersection. The traffic signal at the junction is adjusted based on this vehicle density. When an RFID signal indicating an emergency vehicle is detected within a certain range, the system recognizes the vehicle’s status. If an emergency vehicle is on that route, the traffic server promptly notifies the Arduino at the intersection.

Once the Arduino receives this notification, it compares the information with the RFID signals it has gathered to confirm whether the vehicle is indeed an emergency vehicle. Upon verification, the Arduino activates the green light for the corresponding lane, effectively clearing the way for the emergency vehicle. This ensures that there are no obstructions for the emergency vehicle as it proceeds through the intersection.

With the traffic signal set to green for that lane, all other vehicles can move forward without delay, while the traffic signals for other lanes blink yellow. This system not only enhances the flow of traffic but also provides critical support for emergency vehicles, facilitating their swift passage during urgent situations. Overall, this approach aims to improve traffic efficiency while prioritizing the needs of emergency services.

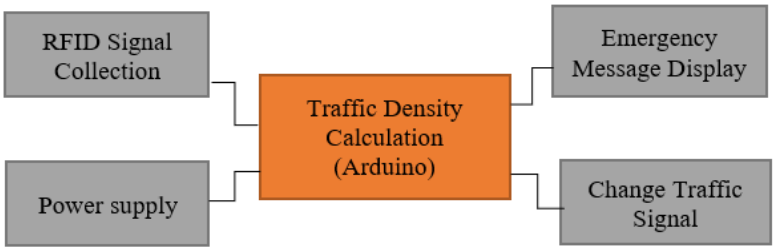


Figure 4. Block Diagram of the Proposed Framework

3. RESULTS AND DISCUSSION

3.1 Preliminary Pre-set Cycle Time (PCT) outcome

The developed program was tested for Pre-set Cycle Time (PCT), and the result was for inspection purposes. Table 3 shows the Pre-set Cycle Timing for 9 lights.

Table 3. Pre-set Cycle Timing for each light

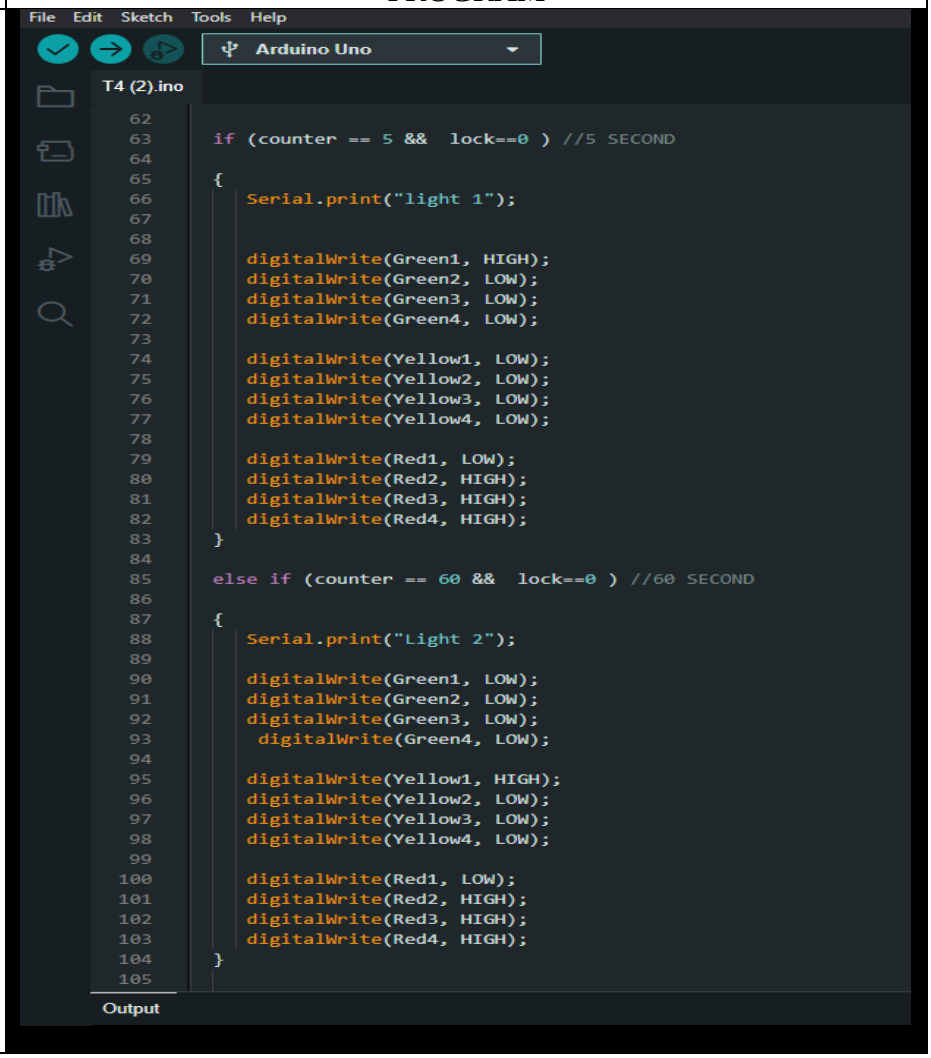
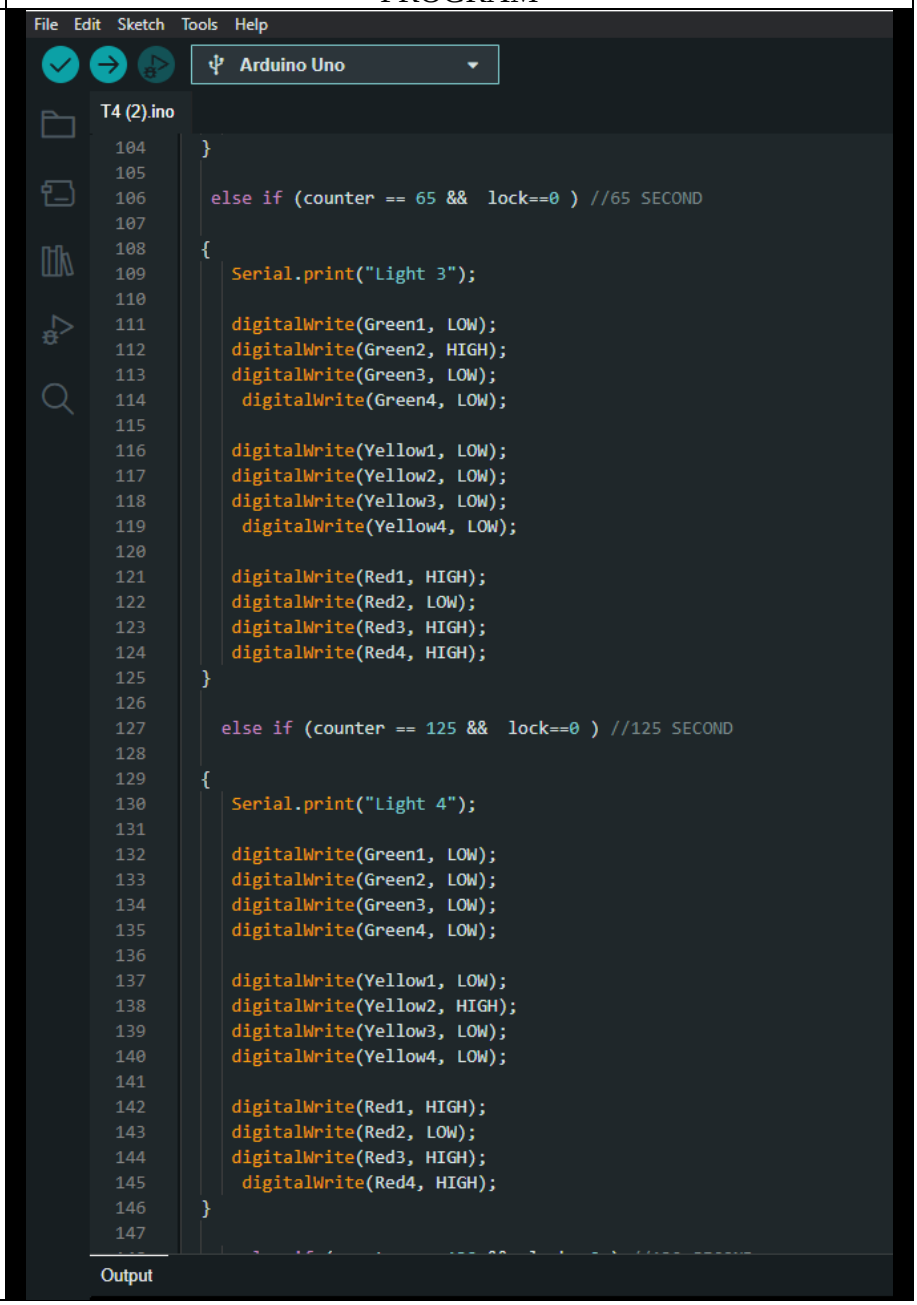
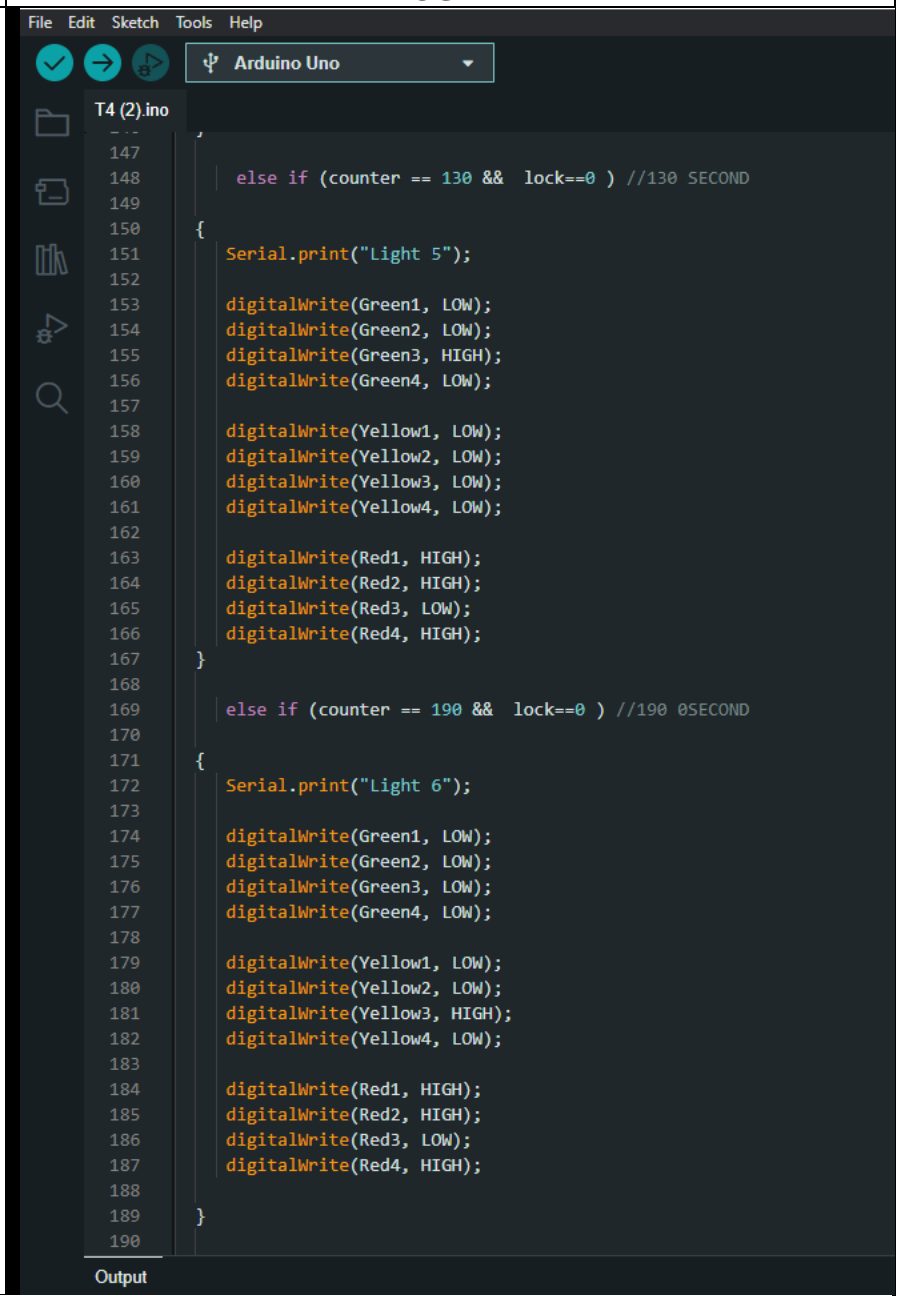
LIGHT NO.	PROGRAM
Light 1 & 2	

Table 3. Pre-set Cycle Timing for each light (Continued)

LIGHT NO.	PROGRAM
Light 3 & 4	

```
}

 else if (counter == 65 &&  lock==0 ) //65 SECOND

{
   Serial.print("Light 3");

   digitalWrite(Green1, LOW);
   digitalWrite(Green2, HIGH);
   digitalWrite(Green3, LOW);
    digitalWrite(Green4, LOW);

   digitalWrite(Yellow1, LOW);
   digitalWrite(Yellow2, LOW);
   digitalWrite(Yellow3, LOW);
    digitalWrite(Yellow4, LOW);

   digitalWrite(Red1, HIGH);
   digitalWrite(Red2, LOW);
   digitalWrite(Red3, HIGH);
   digitalWrite(Red4, HIGH);
}

  else if (counter == 125 &&  lock==0 ) //125 SECOND

{
   Serial.print("Light 4");

   digitalWrite(Green1, LOW);
   digitalWrite(Green2, LOW);
   digitalWrite(Green3, LOW);
   digitalWrite(Green4, LOW);

   digitalWrite(Yellow1, LOW);
   digitalWrite(Yellow2, HIGH);
   digitalWrite(Yellow3, LOW);
   digitalWrite(Yellow4, LOW);

   digitalWrite(Red1, HIGH);
   digitalWrite(Red2, LOW);
   digitalWrite(Red3, HIGH);
    digitalWrite(Red4, HIGH);
}
```

Table 3. Pre-set Cycle Timing for each light (Continued)

LIGHT NO.	PROGRAM
Light 5 & 6	

```
File  Edit  Sketch  Tools  Help
Arduino Uno
T4 (2).ino
147
148      else if (counter == 130 &&  lock==0 ) //130 SECOND
149
150  {
151      Serial.print("Light 5");
152
153      digitalWrite(Green1, LOW);
154      digitalWrite(Green2, LOW);
155      digitalWrite(Green3, HIGH);
156      digitalWrite(Green4, LOW);
157
158      digitalWrite(Yellow1, LOW);
159      digitalWrite(Yellow2, LOW);
160      digitalWrite(Yellow3, LOW);
161      digitalWrite(Yellow4, LOW);
162
163      digitalWrite(Red1, HIGH);
164      digitalWrite(Red2, HIGH);
165      digitalWrite(Red3, LOW);
166      digitalWrite(Red4, HIGH);
167  }
168
169      else if (counter == 190 &&  lock==0 ) //190 0SECOND
170
171  {
172      Serial.print("Light 6");
173
174      digitalWrite(Green1, LOW);
175      digitalWrite(Green2, LOW);
176      digitalWrite(Green3, LOW);
177      digitalWrite(Green4, LOW);
178
179      digitalWrite(Yellow1, LOW);
180      digitalWrite(Yellow2, LOW);
181      digitalWrite(Yellow3, HIGH);
182      digitalWrite(Yellow4, LOW);
183
184      digitalWrite(Red1, HIGH);
185      digitalWrite(Red2, HIGH);
186      digitalWrite(Red3, LOW);
187      digitalWrite(Red4, HIGH);
188
189  }
190
Output
```

Table 3. Pre-set Cycle Timing for each light (Continued)

LIGHT NO.	PROGRAM
Light 7 & 8	

File Edit Sketch Tools Help

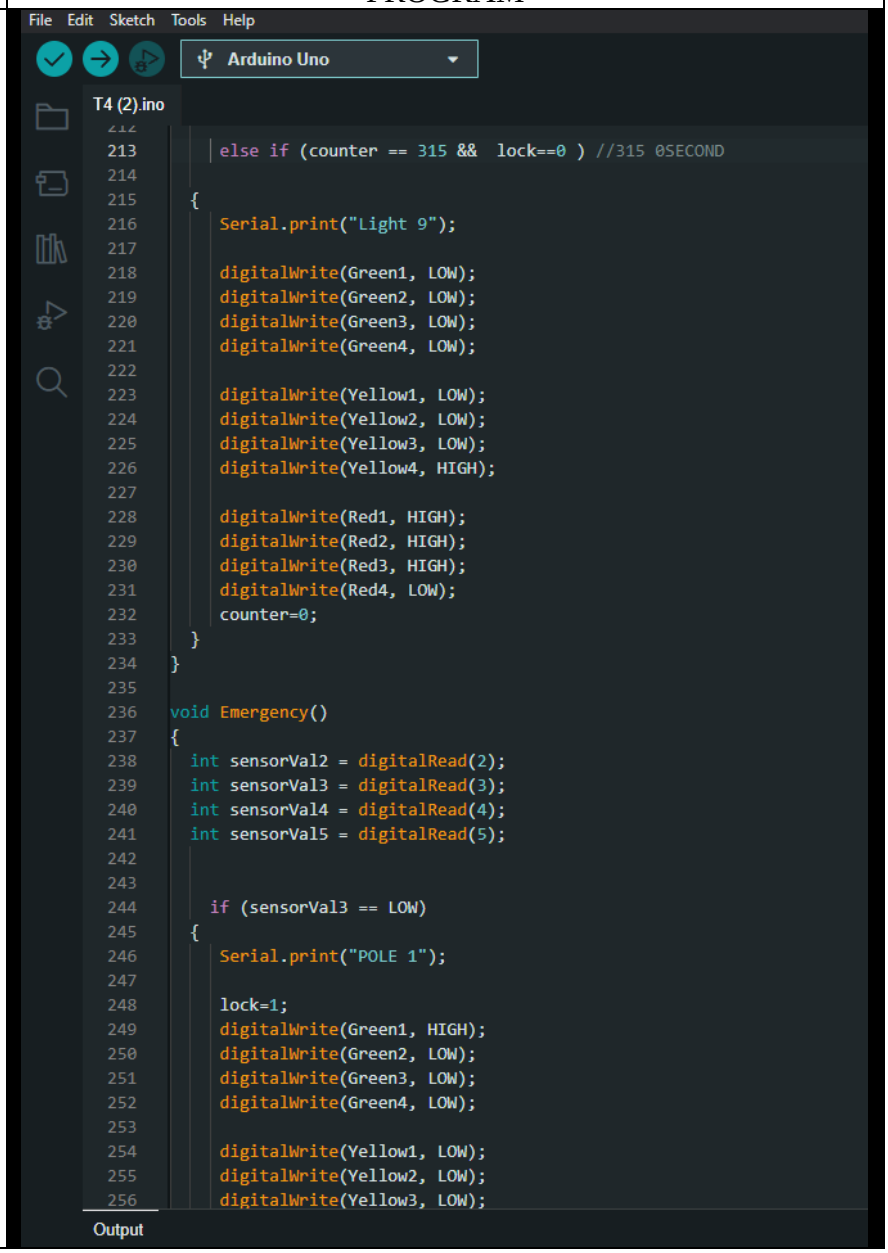
Arduino Uno

T4 (2).ino

```
191     else if (counter == 195 &&  lock==0 ) //195 0SECOND
192
193   {
194     Serial.print("Light 7");
195
196     digitalWrite(Green1, LOW);
197     digitalWrite(Green2, LOW);
198     digitalWrite(Green3, LOW);
199     digitalWrite(Green4, HIGH);
200
201     digitalWrite(Yellow1, LOW);
202     digitalWrite(Yellow2, LOW);
203     digitalWrite(Yellow3, LOW);
204     digitalWrite(Yellow4, LOW);
205
206     digitalWrite(Red1, HIGH);
207     digitalWrite(Red2, HIGH);
208     digitalWrite(Red3, HIGH);
209     digitalWrite(Red4, LOW);
210
211   }
212
213     else if (counter == 255 &&  lock==0 ) //255 0SECOND
214
215   {
216     Serial.print("Light 8");
217
218     digitalWrite(Green1, LOW);
219     digitalWrite(Green2, LOW);
220     digitalWrite(Green3, LOW);
221     digitalWrite(Green4, LOW);
222
223     digitalWrite(Yellow1, LOW);
224     digitalWrite(Yellow2, LOW);
225     digitalWrite(Yellow3, LOW);
226     digitalWrite(Yellow4, HIGH);
227
228     digitalWrite(Red1, HIGH);
229     digitalWrite(Red2, HIGH);
230     digitalWrite(Red3, HIGH);
231     digitalWrite(Red4, LOW);
232     counter=0;
233   }
234 }
```

Output

Table 3. Pre-set Cycle Timing for each light (Continued)

LIGHT NO.	PROGRAM
Light 9	

Moreover, the tested program for each light is according to the following sequence of the next light and the total number of lights on this current junction is 9. As a result, the total cycle for this 4-pole traffic light system is 5 minutes 25 seconds.

3.2 Emergency vehicle appearance outcome

Since the developed Smart Traffic Light System should be able to adapt to various traffic, emergency features have been added to the program. This program is now able to work in 2 different situations which are normal hours and emergency hours. In addition, pole-1,2,3 indicates 'IN' and pole-4 indicates 'OUT'. As a result, when an emergency vehicle is detected by the RFID it sends the signal to the microcontroller and the microcontroller analyses the signal, then sends it to the traffic light. When the microcontroller sends a signal to pole-1, 2, or 3, it becomes green until the emergency vehicle crosses pole-4 then it goes back to the normal running hours. Table 4 illustrates a 4-pole traffic light system with various traffic situations and time features.

Table 4. A 4-pole traffic light system with various traffic situations and time features.

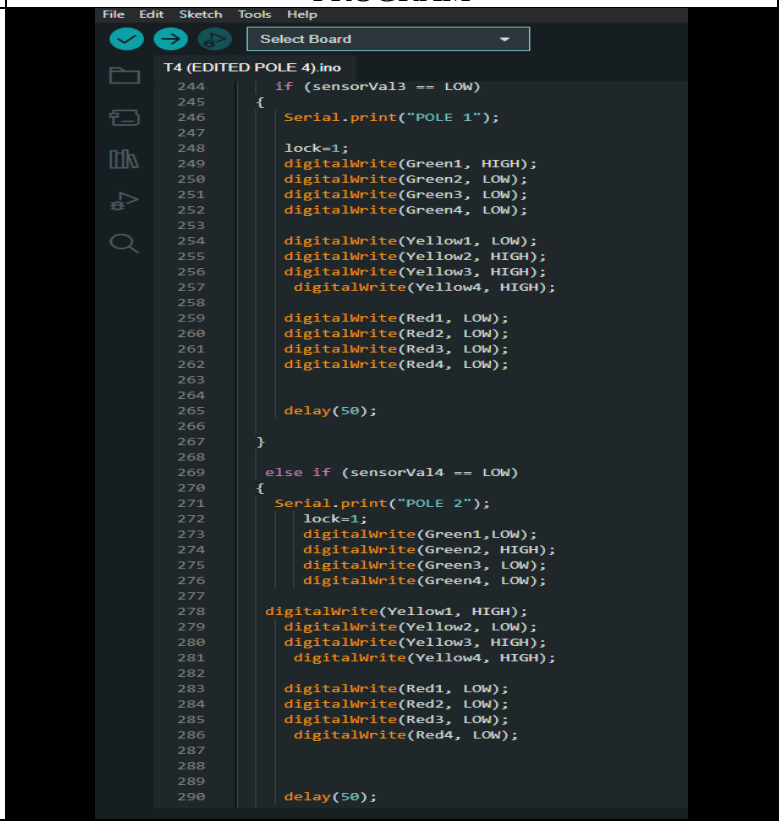
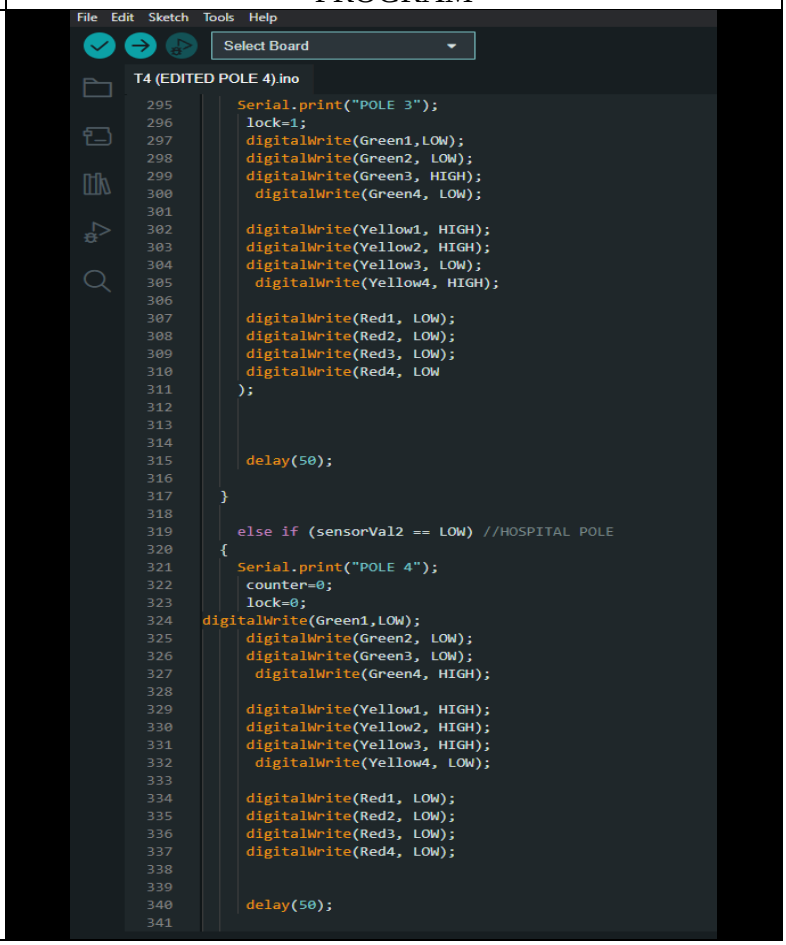
POLE NO.	PROGRAM
Pole 1 & 2 - IN	

Table 4. A 4-pole traffic light system with various traffic situation and time features. (Continued)

POLE NO.	PROGRAM
Pole 3- IN & 4- OUT	

3.3 Unification of Prototype Model

Figure 6 and Figure 7 illustrate STLCS and STLS, respectively on the components that present a solution to the traffic congestion problem as well as an efficient method to give a clear route for emergency vehicles. The emergency vehicles were immediately spotted, and there was a clear path provided. In the meantime, Figure 8 illustrates the landmark by the STLS.

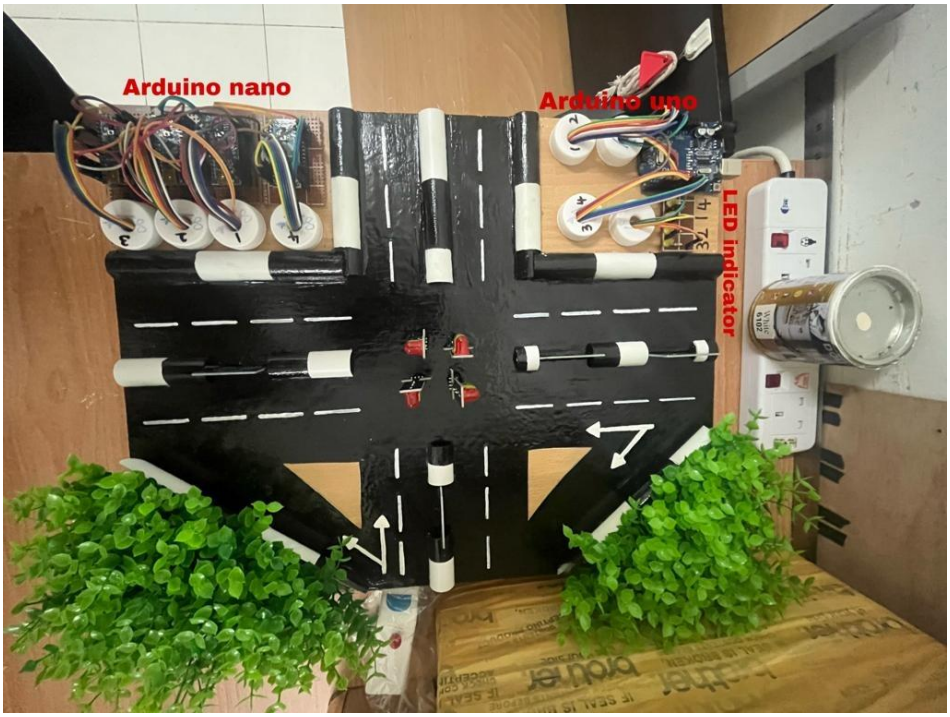


Figure 6. Illustrates the components of STLCS

Figure 7: Illustrates the components of STLS

Figure 8: Illustrates the landmark by the STLS

4. CONCLUSIONS

The current system is ineffective to ensure a clear route for emergency vehicles amid traffic congestion. During peak hours, the Smart Traffic Light Control System (STLCS) offers a solution not only to traffic congestion but also provides an efficient way to clear path for emergency vehicles.

By implementing a smart traffic light system at a T-junction that utilizes RFID technology, traffic flow can be improved, and congestion reduced. The outcomes of these projects can be leveraged to effectively disperse traffic congestion, facilitating the swift passage of emergency vehicles at intersections equipped with traffic lights. This approach not only enhances overall traffic management but also prioritizes the safety and efficiency of emergency services.

REFERENCES

Akoum, A. H. (2017). Automatic Traffic Using Image Processing. *Journal of Software Engineering and Applications*, *10*(09), 765–776. https://doi.org/10.4236/jsea.2017.109042

Anubhav Mehrotra & Abhinav. (2019). *IoT Based Smart Traffic Signal Monitoring System*. *7*(10), 1–3.

Chandrasekhar, M., Saikrishna, C., Chakradhar, B., Phaneendra, K. ., & Sasanka, C. (2013). Traffic Control Using Digital Image Processing. *International Journal of Advanced Electrical and Electronics Engineering*, *2*(5), 2278–8948.

De Oliveira, L. F. P., Manera, L. T., & Luz, P. D. G. Da. (2021). Development of a Smart Traffic Light Control System with Real-Time Monitoring. *IEEE Internet ofThings Journal*, *8*(5), 3384–3393. https://doi.org/10.1109/JIOT.2020.3022392

Eddy, T. B. A. L. A. O. B. R., & Owmya, V. S. (2017). *An Advanced Traffic Light Controller using Verilog HDL*. *05*(07), 657–661.

Javaid, S., Sufian, A., Pervaiz, S., & Tanveer, M. (2018). Smart traffic management system using Internet of Things. *International Conference on Advanced CommunicationsTechnology(ICACT)*,393–398. https://doi.org/10.23919/icact.2018.8323769

Lee, W. H., & Chiu, C. Y. (2020). Design and implementation of a smart traffic signal control system for smart city applications. *Sensors (Switzerland)*, *20*(2). https://doi.org/10.3390/s20020508

Shenbagavalli, S., Priyadharshini, T., Sowntharya, S., Manikandan, P., & Saravanan, S. (2020). Design and Implementation of Smart Traffic Controlling System. *International Journal of Engineering Technology Research & Management IJETRM*, 28–36. http://ijetrm.com/

Swathi, K., Sivanagaraju, V., Manikanta, A. K. S., & Dileep Kumar, S. (2016). Traffic Density Control and Accident Indicator Using WSN. *International Journal for Modern Trends in Science and Technology*, *04*, 2455–3778. https://www.ijmtst.com/documents/15.IJMTST020409.pdf

Vani, R., Thendral, N., Kavitha, J. C., & Bhavani, N. P. G. (2018). Intelligent Traffic Control System with Priority to Emergency Vehicles. *IOP Conference Series: Materials Science and Engineering*, *455*(1). https://doi.org/10.1088/1757-899X/455/1/012023