

Traffic Light Design Analysis to Overcome Congestion in Small Cities Using Simple Arduino

Sunaryo¹, Adris A Putra², Minson Simatupang³, Muh. Thahir Azikin⁴

ABSTRACT- The purpose of this study is to produce a traffic control plan at the intersection of the highway with a combination of the long queue method with the full speed method at the initial start or Full-start-speed (FSS). The planning research method using the desktop then in the Arduino simulation is very simple as an indicator before being launched into the field. The results of this study produce traffic controllers that are very accurate, easy to do, simple and inexpensive.

Keywords: Traffic control, Traffic light, Congestion, Arduino, Full start speed.

1 INTRODUCTION

Traffic control is largely determined by road equipment and facilities [1], based on a traffic density that is growing and developing in a city and region that is experiencing growth. The line at the intersection is a condition and infrastructure that is very widely used in the transportation system. But even so in a store that is equipped with traffic control, it is still possible for certain conditions and emergencies can be adjusted to the conditions of need.

The development of an integrated and algorithmic traffic control system is seen as more efficient with an adaptive fuzzy logic traffic control system [2]. This is very helpful in the development of traffic control in crowded areas of vehicles. Traffic management is very dependent or determined by the length of the queue of vehicles from a road, this method is very popular and very easy compared to other methods. In its development, the traffic queue at the intersection received the full attention of researchers and practitioners to get new methods to overcome the queue process.

Arduino hardware is very much helpful in human activities, including in the regulation of the vehicle queue at the intersection, using a variety of ways the hardware is modified as needed [3]. This is largely determined by the Arduino hardware which has been widely used for human needs in carrying out tasks, especially tasks performed with a repetitive system all the time

continuously, and other tasks that are more complex and complicated.

Arduino hardware is a very reliable microcontroller and the most important thing is that it is cheap and easy to modify according to human needs [4]. Besides having a low cost, Arduino is very flexible and easy to fill commands using programs and can be cloned using a PC (personal computer) or Laptop, the programming language is also easy and can be supported with Microsoft Excel and other software. Arduino treatment is quite easy and has a very sensitive and regular accuracy of punctuality.

Road traffic congestion can be overcome by applying the results of research with the traffic management system [5], the goal is to be able to reduce the queue of vehicles in highway traffic, especially at intersections that require a change in the use of intersections. Whatever method and effort to overcome congestion at the intersection is a result of human effort that is very useful to be used to overcome congestion and vehicle queues at the intersection.

In addition to Arduino hardware support, it is also necessary to support other electrical installations that can continue or display using LEDs or other devices, including the use of Li-Fi technology that is more powerful than other devices [6]. Thus between devices that work together to produce signals that can be useful for controlling the queue at the intersection of the highway.

The development of highway traffic control, the use of Arduino which can be combined with GSM which can send traffic signals to users or drivers [7], (Fault_Monitoring_) the use of this system combination is very useful and helps the driver because it involves the cellular communication system and can be detected directly by the driver, but this system also has shortcomings, especially in the blank spot area due to tall buildings that cannot be penetrated by cellular radio waves.

The development of highway traffic control based on sensing cameras and using Image Processing and Matlab software has been developed [8], this system can determine the length of the

- **Sunaryo**, Student at Post Graduate, Engineering Management Science, Halu Oleo University, Kendari, Indonesia, sunaryocim@gmail.com
- **Adris A Putra**, Associate Professor, Engineering Management Science, Halu Oleo University, Kendari, Indonesia,
- **Minson Simatupang**, Associate Professor, Engineering Management Science, Halu Oleo University, Kendari, Indonesia,
- **Muh. Thahir Azikin**, Associate Professor, Engineering Management Science, Halu Oleo University, Kendari, Indonesia,

vehicle queue at a highway traffic junction, from the identification results are sent to the control center to be able to be used as a gesture, so that the length of the queue can be overcome with this sensor system. However, this system requires very expensive costs and the costs of maintenance and supervision are very thorough.

The development of technology implementation of VANET simulation techniques has been used in India [9], this system is a traffic management system, this system uses an accurate and weighted algorithmic system, as a working system that can provide information to the controller then displayed on a system that can be used to overcome the highway traffic queue system at the intersection. This is related to TDT's ability to detect a significant increase in reaction time [10] when using traffic lights carefully.

Road traffic intersections controlled using traffic lights, in addition to regulating the queue of vehicles can also be used, to provide a sense of security and opportunities for pedestrians to use as a means of crossing [11]. This is very useful and dual function, but all must require the support of pedestrians to obey traffic rules as well as drivers.

Signaling intersection planning with methods and criteria based on the fundamental relationship between the capacity of road users [12], this is a plan that uses a system that has long been used, the system needs to get changes and new ways that can be used to contribute to, the previous regulatory system crossing at a crossroads which aims to unravel congestion.

2 LITERATURE REVIEW

Electronic; The electronic field is something that is related to controlling, controlling, disconnecting and turning on the flow of electric current through a circuit, it is especially for tools that use electronic switches, or a switch that is not visible in plain view [13]. Initially, before the discovery of an electronic switch, to turn on an electric current must use a mechanical switch, which requires a lever and requires mechanical movement, to be useful for connecting or disconnecting circuits. Although mechanical switches are acceptable, in certain devices it is no longer feasible to use the model or device, then the switches change a lot in accordance with the times, thus the switches are replaced by using switching. Thus the term electromechanical switch (called a relay) as such is not a requirement as an electronic device, because the switch uses electric current to produce it to run into mechanical motion that is used to break the current or switch. This electronic switch is becoming more widely used which can replace the action of a mechanical switch by using an electrical reaction inside the device so that there are no moving parts. Not using physical movements, this device can be activated in a very short and fast time and with much greater reliability. The material used to make this switch only produces electricity under certain conditions at a certain voltage or current level must be present at the input and output devices to open or close it. When the device is turned on, it conducts electricity with a certain amount of resistance. And when the device is turned off, it doesn't conduct

electricity and instead acts as an insulator. Models or types like this in electronic components are called "semiconductors" because they can be conductors or insulators depending on the electrical conditions they use.

Semi-conductor; The use of semi-conductors widely used as a substitute for mechanical switches is a circuit or circuit "electronic," because it will pass or flow an electrical signal to be used to activate at very high speeds, and this is not possible with mechanical or ordinary circuits. Thus many semi-conductors have different properties, and this is an important type used in most circuits.

Diode: Similar to a one-way valve that functions to regulate an electric current, this component only allows the electric current to pass through in one direction only and is very useful by itself, but is mostly used as a basis for solid-state electronic components.

Light Emitting Diode (LED): Components of this type of diode emit a little light when an electric current is inserted through it.

Light Dependent Resistor (LDR): This type of component is a semi-conductor which has a resistance that can vary depending on the amount of light available.

Bipolar Junction Transistors (BJT): Components of this type are electronic current-driven switches that are used for their fast switching properties.

Metal-Oxide-Semiconductor Field-Effect Transistor (mosfet): Component of this type is an electronic switch that is driven using voltage due to its fast switching properties, low resistance, and the ability to operate in parallel circuits. This component is basically the basis for most power amplifier circuits.

All electronic components have several layers of positively and negatively charged silicon attached to the chip by exposed conductive metal tin for soldering into the circuit. For transistor and MOSFET components it has a built-in diode to protect it from back-voltage and Back-EMF, so it is a good idea to review the datasheet of the part you are using.

Datasheet; All components must have their own datasheets issued by the manufacturer that made them. The data sheet stores all important electrical information about a hardware component. The upper limit, usually called "Absolute Maximum Ratings," shows you at what point the device will fail (see Figure 1). The lower limit (if applicable) tells you what level the device will no longer respond to input - this usually won't damage the device, it just won't work.



Figure 1. The sample datasheet from Fairchild Semiconductor

There is also a section called “Electrical Characteristics” that tells you at what level the device operates properly. This usually shows the exact voltage or current level that will turn the device on or off. These ratings are helpful in determining what other component values (i.e., resistors and capacitors) should be selected or whether the device will work for the intended purpose. The datasheet usually tells you far more than you know what to do with, ending with graphs and package dimensions. Some datasheets even have circuit layout recommendations and suggest ways to interface the component with a micro-controller. For popular or commonly used component parts, you can also check the manufacturer’s website for additional documents that further describe how to use the component—these are called “application notes,” and can be insightful.

The Arduino software is referred to as an Integrated Development Environment (IDE). This is the programming software that is used to upload code to the Arduino micro-controller. The IDE contains a text-editor and compiler that translates the simplified Arduino programming language (that we write) into a more complicated binary hex file that can be uploaded directly to the micro-controller. The Arduino language is a variant of the C++ programming language, but uses built-in libraries to simplify complicated coding tasks to make it easier for beginners to pick up. If you have no prior programming experience, you will benefit greatly from the Arduino reference pages. These pages show each Arduino command and how to use it with an example snippet of code.

Because the Arduino language is an open source project, it is constantly being improved and updated. New versions of the Arduino IDE are released often, so it is best to update your system with the newest release available. Most of the projects in this book use the IDE 0019–0021, which can be downloaded at the Arduino homepage.

Arduino Variants; The Arduino comes in many different shapes and sizes, but there are only two models that use completely different chips: the Standard and the Mega. The Standard is the basic Arduino and refers to the Atmega8/168/328 chip, whereas the Mega is a different Arduino board with more I/O pins and uses the beefier Atmega1280 chip. Because the Arduino design is open source, anyone can design a new version of the Arduino board and distribute it as he pleases. For this reason, several other manufacturers have created Arduino “clones” that operate as the standard Arduino, but are made by a third party or offered as a kit to build yourself. There are also Arduino boards that do not have an onboard USB converter, so you must use a special USB (FTDI) programming cable to program. The upside to using the FTDI chip on a separate programming cable instead of the Arduino board itself is that you can then easily make your own Arduino-type boards, using only an Atmega328 chip, 16mHz resonator, and a few other easy-to-find components. If you add a few header pins, you can even program your homemade Arduino boards incircuit (see Figure 2). After buying the FTDI programming cable from Sparkfun.com, I went on an unintended but inspired building spree and made about 15 different Arduino clones that had different pin configurations, screw-terminals, R/C

headers, powered Servo plugs, and even a few stackable Arduino extension boards. Although none of my homemade boards had onboard USB functionality, several had a 6-pin FTDI programming header to enable in-circuit programming.



Figure 2. The hardware keet Arduino

The Sketch; The sketch is nothing more than a set of instructions for the Arduino to carry out. Sketches created using the Arduino IDE are saved as .pde files. To create a sketch, you need to make the three main parts: Variable declaration, the Setup function, and the main Loop function.

Variable Declaration; Variable declaration is a fancy term that means you need to type the names of each input or output that you want to use in your sketch. You can rename an Arduino input/output pin number with any name (i.e., led_pin, led, my_led, led2, pot_pin, motor_pin, etc.) and you can refer to the pin by that name throughout the sketch rather than the pin number. You can also declare a variable for a simple value (not attached to an I/O pin) and use that name to refer to the value of that variable. Thus, when you want to use the value of the variable later in the sketch, it is easy to recall. These variables can be declared as several different types, but the most common that we use is an integer (int). In the Arduino language, an integer variable can contain a value ranging from -32,768 to 32,767. Other variable types are used in later examples (i.e., float, long, unsigned int, char, byte, etc.) and are explained when used. Following is an example variable declaration: `int my_led = 13;`

Instead of sending commands to the pin number of the Arduino (i.e., 2), we rename pin 13 to be “my_led.” Anytime we want to use pin 13, we call my_led instead. This is helpful when you have many references to my_led throughout the sketch. If you decide to change the pin number that my_led is attached to (i.e., to pin 4), you change this once in the variable declaration and then all references to my_led lead to pin 4—this is meant for easier coding.

The Sketch The sketch is nothing more than a set of instructions for the Arduino to carry out. Sketches created using the Arduino IDE are saved as .pde files. To create a sketch, you need to make the three main parts: Variable declaration, the Setup function, and the main Loop function.

3 RESEARCH METHODS

This study uses a traffic light simulation method using Arduino Hardware and Software, with the help of the goodness of

Microsoft Excel software as a data processor and then copying it to the Arduino software, without the help of Microsoft Excel the work will be longer and can not be controlled. Data that has been copied is in Arduino software then cloned or injected into Arduino

hardware, then the traffic lights will immediately turn on according to the commands that have been entered.

The next method is to combine the queue length method by using the full-start-speed (FSS) method, with this combination it will help reduce the length of the queue.



Fig.3. Research Map



Fig.4 Map Detail

4 RESULTS

This research needs to be considered is how to design a highway traffic regulator, using a microcontroller from Arduino, but in designing, we must use the help of Microsoft Excel, which

functions to process the input data entered. If you do not use the help of Microsoft Excel, Arduino cannot process data, Arduino only functions as a microcontroller.

Input data will be entered in the worksheet that has been created as in Figure 5. below:

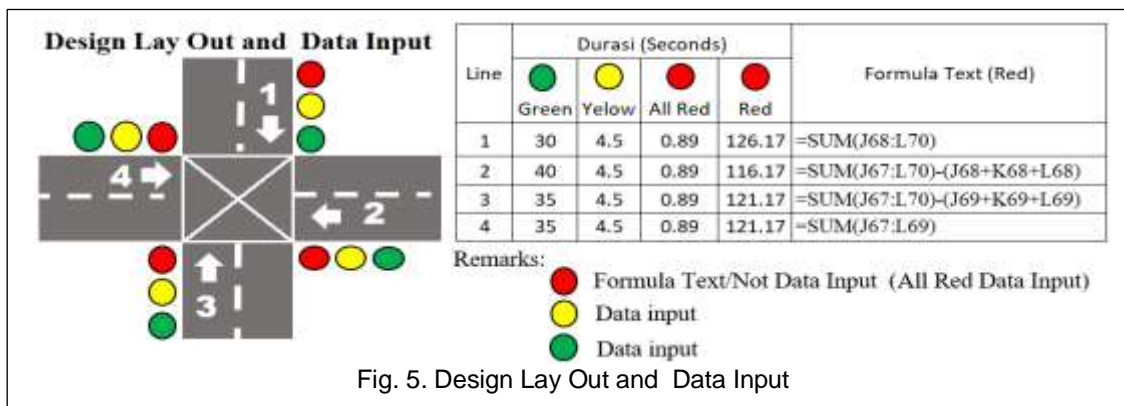


Fig. 5. Design Lay Out and Data Input

Based on picture 1 there is a picture of a four-way intersection consisting of road 1, road 2, road 3 and road 4, each space has different vehicle volumes, so in reality on the field when the green light is on each section different. In general, the way to determine how long it lasts for each lamp is determined by the stakeholders. But there is also in a study all the lights that will light must be determined or counted. On this occasion specifically for red lights except for All-Red not specified just waiting for all of the lights will light including All-Red then each red light will light based on the total number of lights that have been lit including All-Red. As in

the text formula (Red).

Thus the way the values in the input in table 1 green as an assumption are entered 30 seconds, yellow 4.5 seconds and All-Red 0.89 seconds. For line 2 the green assumption is entered 40 seconds, yellow 4.5 seconds and All-Red 0.89 seconds, line 3 green as the assumption is entered 35 seconds, yellow 4.5 seconds and All-Red 0.89 seconds and line 4 green as the assumption is entered 35 seconds, yellow 4.5 seconds and All-Red 0.89 seconds. For red lights will be determined by the formula text (Red).

TABLE 1
 Traffic Ligth Programming and Diagram Block Analysis Design

Fase	Programming	Ligth				Fase	Programming	Ligth				Fase	Programming	Ligth									
		1	2	3	4			1	2	3	4			1	2	3	4						
1	digitalWrite(2,0);					2	digitalWrite(2,1);					3	digitalWrite(2,1);					4	digitalWrite(2,1);				
	digitalWrite(3,0);						digitalWrite(3,0);						digitalWrite(3,0);						digitalWrite(3,0);				
	digitalWrite(4,1);						digitalWrite(4,0);						digitalWrite(4,0);						digitalWrite(4,0);				
	digitalWrite(5,1);						digitalWrite(5,1);						digitalWrite(5,1);						digitalWrite(5,1);				
	digitalWrite(6,0);						digitalWrite(6,0);						digitalWrite(6,0);						digitalWrite(6,0);				
	digitalWrite(7,0);						digitalWrite(7,1);						digitalWrite(7,0);						digitalWrite(7,0);				
	digitalWrite(8,1);						digitalWrite(8,1);						digitalWrite(8,1);						digitalWrite(8,1);				
	digitalWrite(9,0);						digitalWrite(9,0);						digitalWrite(9,0);						digitalWrite(9,0);				
	digitalWrite(10,0);						digitalWrite(10,0);						digitalWrite(10,1);						digitalWrite(10,0);				
	digitalWrite(11,1);						digitalWrite(11,1);						digitalWrite(11,1);						digitalWrite(11,0);				
	digitalWrite(12,0);						digitalWrite(12,0);						digitalWrite(12,0);						digitalWrite(12,0);				
	digitalWrite(13,0);						digitalWrite(13,0);						digitalWrite(13,0);						digitalWrite(13,1);				
	delay(30000)						delay(40000)						delay(35000)						delay(25000)				
	digitalWrite(2,0);						digitalWrite(2,1);						digitalWrite(2,1);						digitalWrite(2,1);				
	digitalWrite(3,1);						digitalWrite(3,0);						digitalWrite(3,0);						digitalWrite(3,0);				
	digitalWrite(4,0);						digitalWrite(4,0);						digitalWrite(4,0);						digitalWrite(4,0);				
	digitalWrite(5,1);						digitalWrite(5,0);						digitalWrite(5,1);						digitalWrite(5,1);				
	digitalWrite(6,0);						digitalWrite(6,1);						digitalWrite(6,0);						digitalWrite(6,0);				
	digitalWrite(7,0);						digitalWrite(7,0);						digitalWrite(7,0);						digitalWrite(7,0);				
	digitalWrite(8,1);						digitalWrite(8,1);						digitalWrite(8,0);						digitalWrite(8,1);				
digitalWrite(9,0);					digitalWrite(9,0);					digitalWrite(9,1);					digitalWrite(9,0);								
digitalWrite(10,0);					digitalWrite(10,0);					digitalWrite(10,0);					digitalWrite(10,0);								
digitalWrite(11,1);					digitalWrite(11,1);					digitalWrite(11,1);					digitalWrite(11,0);								
digitalWrite(12,0);					digitalWrite(12,0);					digitalWrite(12,0);					digitalWrite(12,1);								
digitalWrite(13,0);					digitalWrite(13,0);					digitalWrite(13,0);					digitalWrite(13,0);								
delay(4500)					delay(4500)					delay(4500)					delay(4500)								
digitalWrite(2,1);					digitalWrite(2,1);					digitalWrite(2,1);					digitalWrite(2,1);								
digitalWrite(3,0);					digitalWrite(3,0);					digitalWrite(3,0);					digitalWrite(3,0);								
digitalWrite(4,0);					digitalWrite(4,0);					digitalWrite(4,0);					digitalWrite(4,0);								
digitalWrite(5,1);					digitalWrite(5,1);					digitalWrite(5,1);					digitalWrite(5,1);								
digitalWrite(6,0);					digitalWrite(6,0);					digitalWrite(6,0);					digitalWrite(6,0);								
digitalWrite(7,0);					digitalWrite(7,0);					digitalWrite(7,0);					digitalWrite(7,0);								
digitalWrite(8,1);					digitalWrite(8,1);					digitalWrite(8,1);					digitalWrite(8,1);								
digitalWrite(9,0);					digitalWrite(9,0);					digitalWrite(9,0);					digitalWrite(9,0);								
digitalWrite(10,0);					digitalWrite(10,0);					digitalWrite(10,0);					digitalWrite(10,0);								
digitalWrite(11,1);					digitalWrite(11,1);					digitalWrite(11,1);					digitalWrite(11,1);								
digitalWrite(12,0);					digitalWrite(12,0);					digitalWrite(12,0);					digitalWrite(12,0);								
digitalWrite(13,0);					digitalWrite(13,0);					digitalWrite(13,0);					digitalWrite(13,0);								
delay(890)					delay(890)					delay(890)					delay(890)								

Remarks: ="delay("&K67*1000&"")

Based on Table 1, the writing delay will automatically adjust itself because Microsoft Excel uses the string function to overcome it. In the table above phase 1 group 1 there is a text formula written = "delay("&J67*1000&")" and in the table written 30000 ms, this means to delay or time for 30000 milliseconds and thus this command applies to sketches above it, in real terms the lights are red, red, red and yellow. In phase 1 group 2 in the table written 4500 ms, this means the delay or time for 4500 milliseconds and thus this command applies to the sketches above, in real terms the lights are green, red, red and red. In phase 1 group 3 in the table written 890 ms, this means the delay or time for 890 milliseconds and thus this command applies to the sketcher above, in real terms the lights are red, red, red and red (All-Red 1)

In phase 2 group 1 there is a text formula written = "delay("&J68*1000&")" and in the table it says 40000 ms, this means to delay or time for 40000 milliseconds and thus this command applies to the sketcher above, in real terms the lights are red, red, red and yellow. In phase 2 group 2 in the table written 4500 ms, this means delay or time for 4500 milliseconds and thus this command applies to the sketcher above, in real terms the lights are green, red, red and red. In phase 2 group 3 in the table written 890 ms, this means the delay or time for 890 milliseconds and thus this command applies to the sketcher above, in real terms the lights are red, red, red and red (All-Red 2).

In phase 3 group 1 there is a text formula written = "delay("&J69*1000&")" and in the table it says 35000 ms, this means to delay or time for 35000 milliseconds and thus this command applies to the sketcher above, in real terms the lights are red, red, red and yellow. In phase 3 group 2 in the table written 4500 ms, this means delay or time for 4500 milliseconds and thus this command applies to the sketcher above, in real terms the lights are green, red, red and red. In phase 3 group 3 in the table written 890 ms, this means the delay or time for 890 milliseconds and thus this command applies to the sketcher above, in real terms the lights are red, red, red and red (All-Red 3).

In phase 4 group 1 there is a text formula written = "delay("&J50*1000&")" and in the table it says 25000 ms, this means to delay or time for 25000 milliseconds and thus this command applies to the sketcher above, in real terms the lights are red, red, red and yellow. In phase 4 group 2 in the table written 4500 ms, this means delay or time for 4500 milliseconds and thus this command applies to the sketcher above, in real terms the lights are green, red, red and red. In phase 4 group 3 in the table written 890 ms, this means the delay or time for 890 milliseconds and thus this command applies to the sketcher above, in real terms the lights are red, red, red and red (All-Red 4).

The results of the analysis in table 1, then copied to the Arduino software, at this stage are called sketches, the initial appearance can be seen in Figure 6. The following:

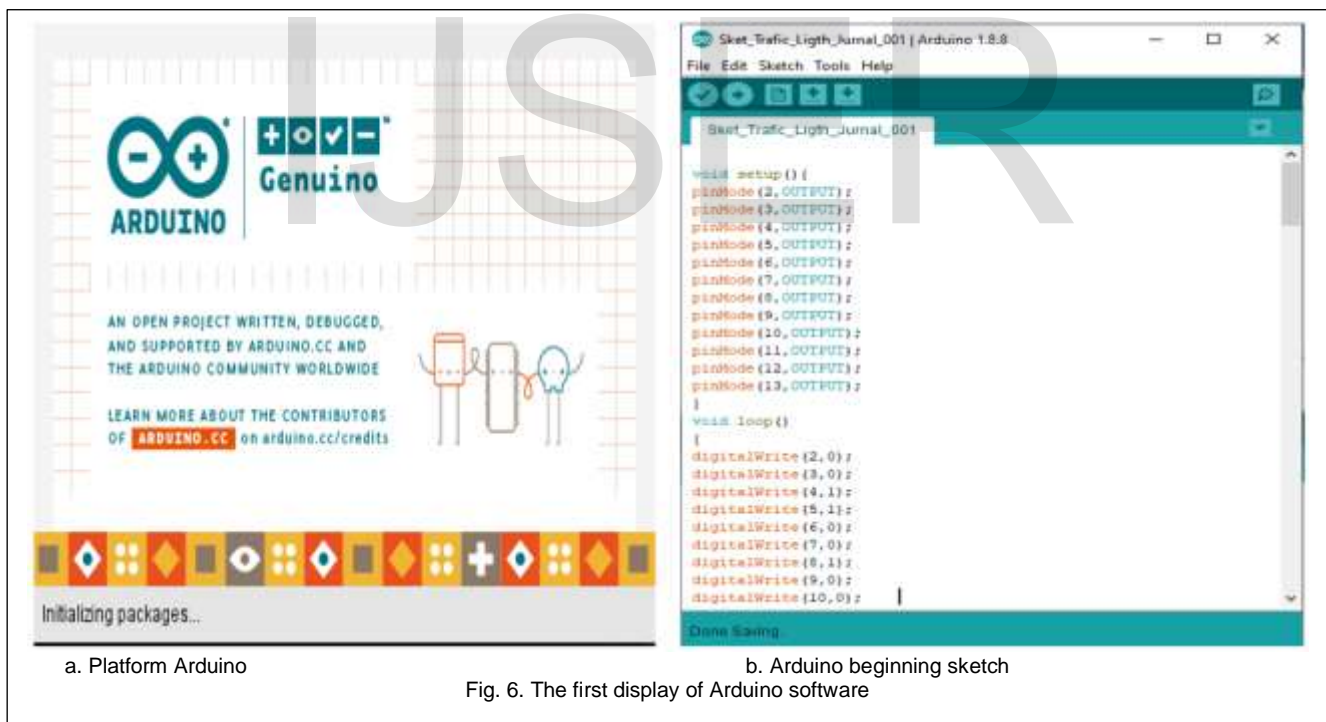


Fig. 6. The first display of Arduino software

Based on Figure 6.a. is a platform from Arduino as a provider of hardware and software services that are very concerned about consumer needs. He provides online service support to meet the needs of consumers. In Figure 6.b is the first display of the worksheet from Arduino software, at the top there is a tool bar that serves to give orders so that work can be forwarded to where we will take it.

In the workspace section is the most important part of this section, the commands that we make in Microsoft Excel can be input here by copying and pasting in this room, so it's usually called a sketch. In the sketch, the void setup command appears as the initial command for a sketch, then pinMode (2,OUTPUT) is seen; up to pinMode (13,OUTPUT); this means that the pin being used must be declared or introduced in camouflage and in real terms only humans know, the actual introduction of the contents

is that pin 2 to pin 4 phase 1, pin 5 to pin 7 phase 2, pin 8 to pin 10 phase 3 and pin 11 to pin 13 phase 4. The next command is a void loop which means the command to repeat all the sketched commands, digitalWrite (2,0) is written; this means that pin 2 is

given the number 0 means the lamp is off, digitalWrite (4,1); this means that pin 3 is given the number 1 means the lamp is on, the length of the command is 30000 milliseconds.

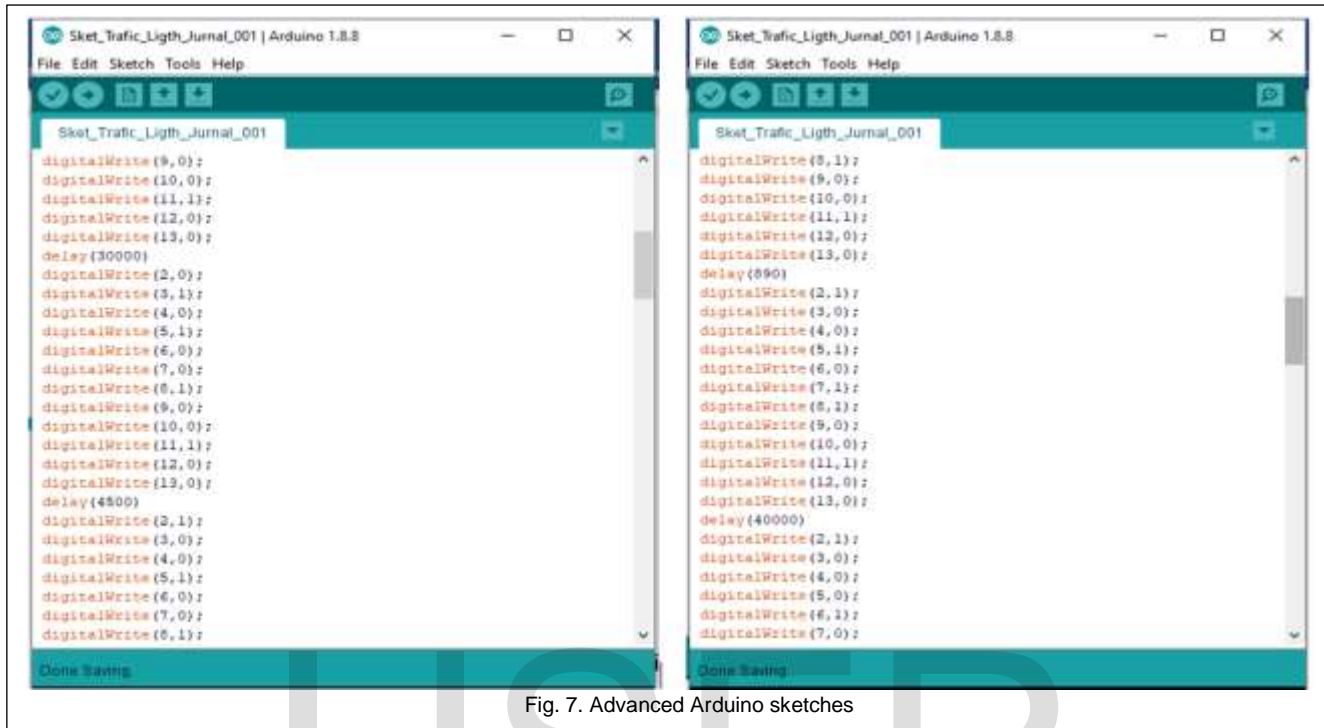


Fig. 7. Advanced Arduino sketches

At the end of the sketch is a cover, in Figure 7 it says digitalWrite (3,1); digitalWrite (5,1); digitalWrite (8,1); and digitalWrite (11,1); in a burning condition for 4500 milliseconds. And written digitalWrite (2,1); digitalWrite (5,1); digitalWrite

(8,1); and digitalWrite (11,1); in a light condition for 890 milliseconds. This sketch will start over after all of these commands are marked with a loop at the beginning of this sketch.

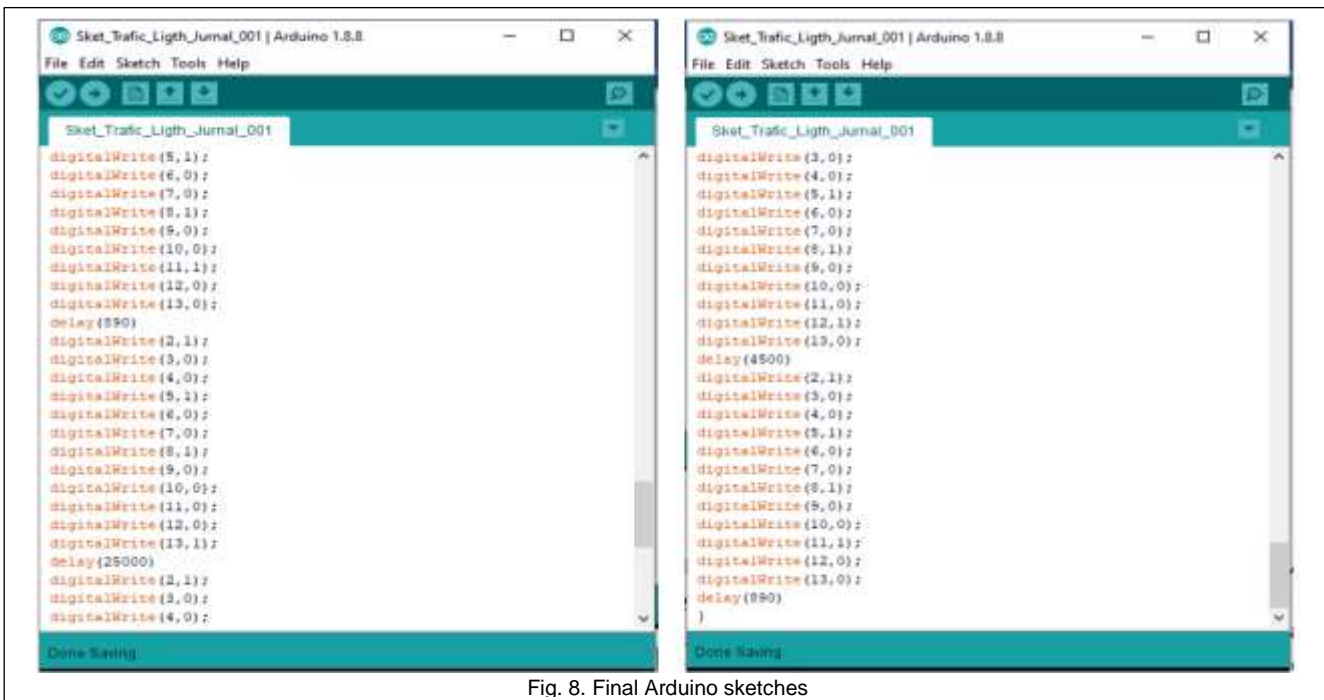


Fig. 8. Final Arduino sketches

At the end of the sketch is a cover, in Figure 8 it says digitalWrite (2,1); digitalWrite (5,1); digitalWrite (8,1); and digitalWrite (13,1); in a burning condition for 25000 milliseconds. And written digitalWrite (2,1); digitalWrite (5,1); digitalWrite (8,1); and

digitalWrite (11,1); in a light condition for 890 milliseconds. This sketch will start over after all of these commands are marked with a loop at the beginning of this sketch.

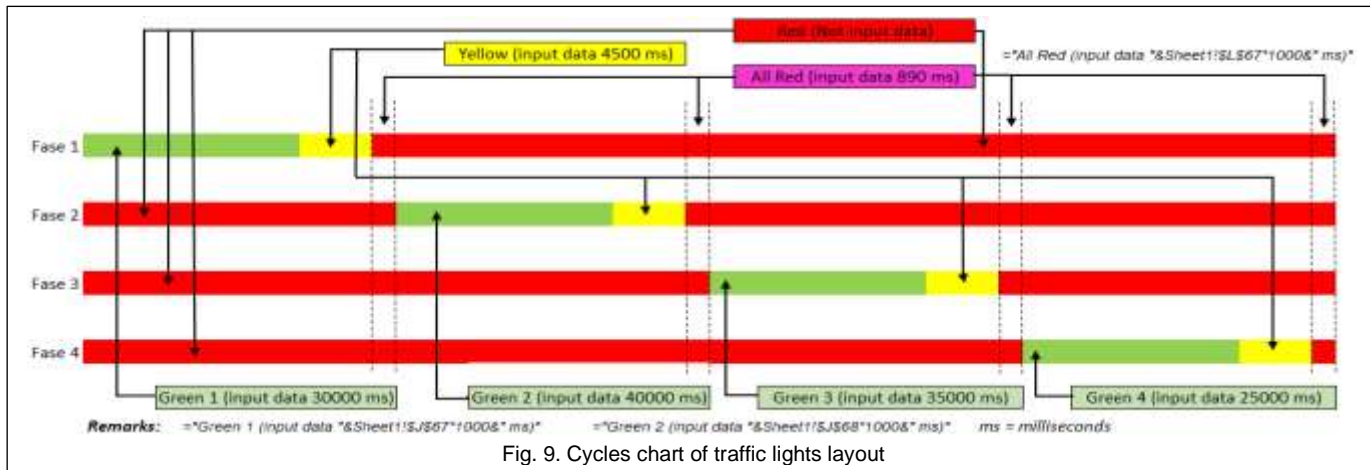


Fig. 9. Cycles chart of traffic lights layout

Based on Figure 9 it is very clearly printed on the bar, that in phase 1 the green light is lit based on a data input of 30000 ms, the yellow light is on with data input of 4500 ms, the All-Red light is on with the data input of 890 ms and the red light is not on using data input. In phase 2 the red light turns on based on not based on input, the green light turns on based on data input of 40000 ms, the yellow light is on with data input of 4500 ms, the All-Red light is on with data input of 890 ms and the red light is on does not use input data. In phase 3 the red light turns on based on not based on input, the green light turns on based on data input of 35000 ms, the yellow light lights up with data input of 4500 ms, the All-Red light turns on with data input of 890 ms and the red light turns on using no input data. In phase 4 the red light turns on based on not based on input, the green light turns on based on data input of 35000 ms, the yellow light turns on with data input of 4500 ms,

the All-Red light turns on with data input of 890 ms and the red light turns on using no input data.

Thus that the red light will turn on does not require data input commands and will turn on due to the accumulation of hilau, yellow and All-Red lights.

Based on the description above about the length of the light is based on each position, then the question arises? Why at each phase of the hilarious lamp lights up with different durations. The answer is that the length of the red light in each phase is due to the different queues of vehicles on each road, that is the answer. However, on this occasion I introduce a method, to help overcome the above problem, it needs to be applied to be combined using the Full-start-speed method with the description as shown in figure 10.

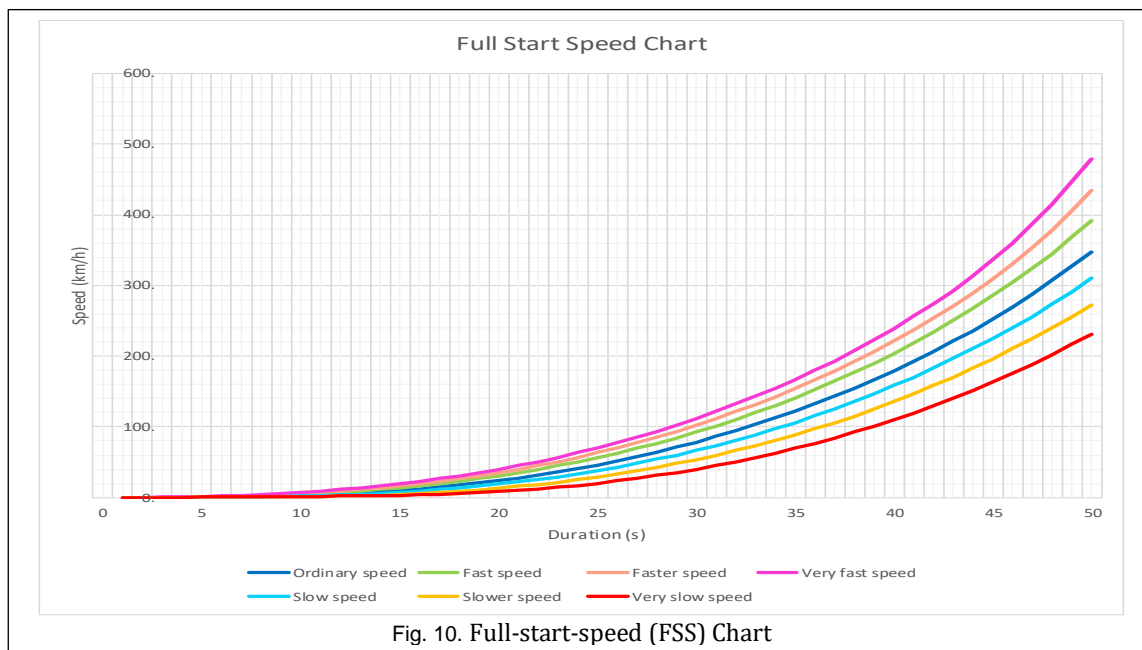


Fig. 10. Full-start-speed (FSS) Chart

Based on Figure 10, it can be seen that in normal conditions the vehicle's speed is according to survey data, the vehicle runs when the green light is on at normal speed, with an average speed value of 13.25 km/h with a long performance of approximately 132.50 meters in 36 seconds. This speed must be increased by the drivers to be faster, but the reality on the ground there is still a slower speed than the usual speed with an average value of 11.55 km/h with a long performance of about 115.55 meters with a time of 36 seconds, and worse again there are still drivers driving their vehicles at very slower speeds with an average value of 7.65 km/h with long achievements of about 76.50 meters in 36 seconds. On this occasion to help overcome the queue at the highway crossing traffic. It is expected that the driver can improve more skillfully

and increase speed at the initial start at the time of the green light at the crossroads.

It is expected that the minimum speed improvement can reach the faster category with an average fast speed value of 15.25 km/h with a long performance of approximately 152.50 meters with a time of 36 seconds, and a faster speed category of an average of 16.60 km/h with a length of approximately 160 meters with a time of 36 seconds. And even more extreme with the category is very faster speed with an average speed of 18 km/h with a long performance of about 180 meters with 36 seconds.

This method is very suitable to be combined with the previous method, only it must require socialization and regulation by related parties, and make the main conditions in the driver.

TABLE 2
Summary of Full start speed, Distance (m)

Fase	Green Duration (Seconds)	Full start speed, Distance (m)							
		Very slow speed	Slower speed	Slow speed	Ordinary speed	Fast speed	Faster speed	Very fast speed	Average speed
Fase 1	30	40.000	53.750	66.667	78.333	92.500	102.083	112.083	77.917
Fase 2	40	110.000	135.556	158.333	179.444	203.889	221.111	238.889	178.175
Fase 3	35	69.514	88.472	105.972	122.014	140.972	154.097	167.222	121.181
Fase 4	25	20.139	29.167	38.194	46.181	56.250	63.194	70.139	46.181

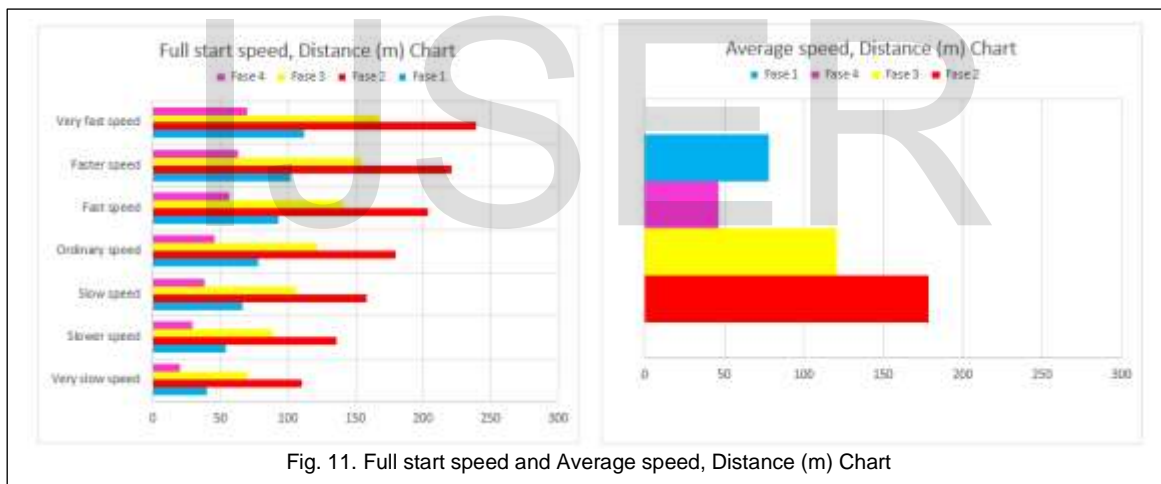


Fig. 11. Full start speed and Average speed, Distance (m) Chart

Based on table 2, then in phase 1, the green light is on with a duration of 30 seconds, with an average mileage of 77.917 meters, on phase 2 the green is on with a duration of 40 seconds, with an average mileage of 178.175 meters, on phase 3 the green light up with a duration of 35 seconds, with an average distance of 121.181 meters and in phase 4 green light up with a duration of 25 seconds, with an average distance of 46.181 meters, more fully explained using the graph in Figure 10.

Based on the results of calculations in table 2, the variation of 7 types of vehicle speed at initial start varies greatly, but from these variations can take the average value that can be used as a reference, that in that phase the length of the empty road or occupied by the next vehicle is appropriate the average value of the length of the road due to the accumulation of various variations in vehicle speed at initial start.

This method was first tried to be used, while still using the previous method, namely by determining the length of time the green light is on, according to the number of queues of vehicles in an intersection road space. But apparently, the method needs to get a combination with the Full-Start-speed (FSS) method. So that a combination of several methods can be applied, there needs to be continuity and outreach to road users in order to drive vehicles to be more nimble and more skilled in driving vehicles.

This research is a combination of several scientific disciplines, however, it does not become an obstacle for researchers, because all will be interrelated and cannot be run individually. It is also hoped that other researchers can adopt or develop this research so that it becomes more developed and a combination of other methods is highly expected for the continuity of this research.

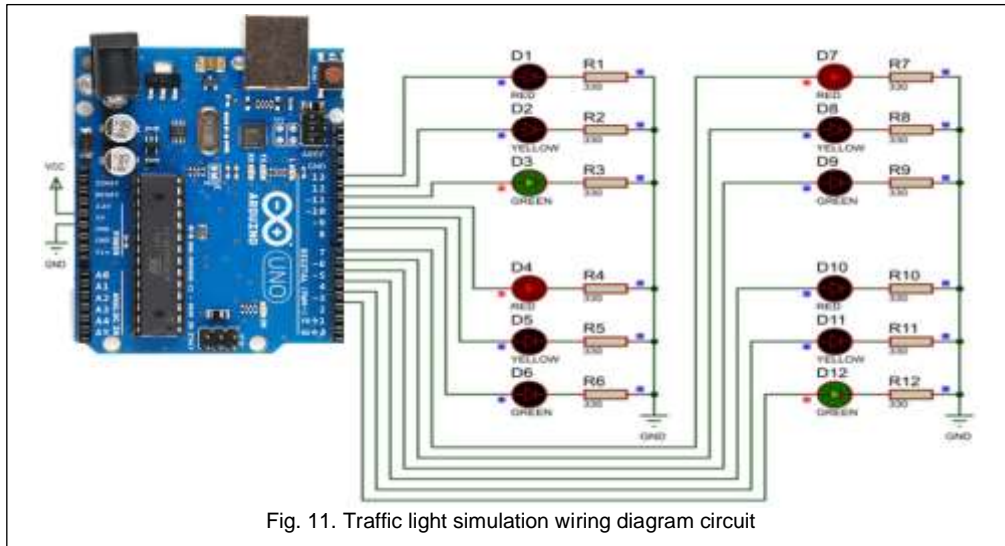


Fig. 11. Traffic light simulation wiring diagram circuit

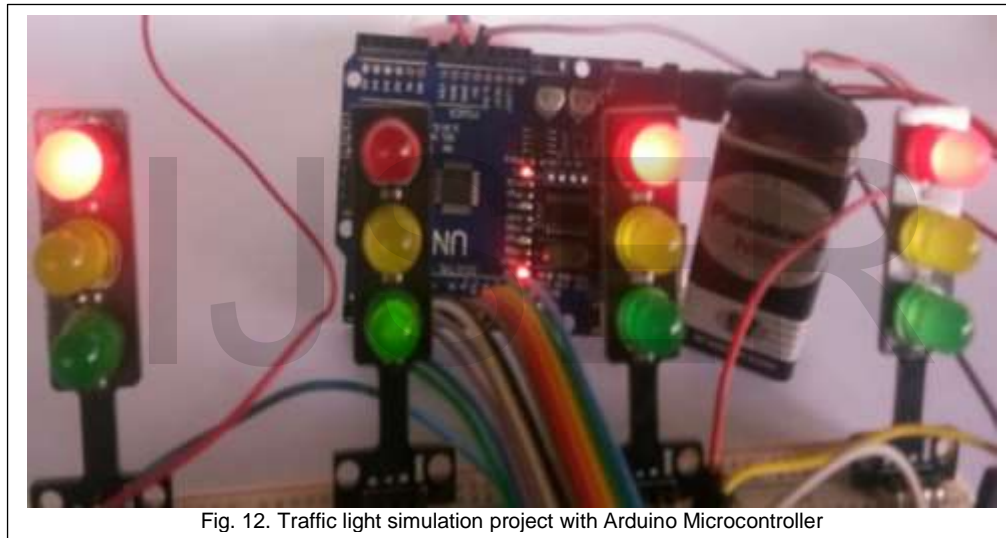


Fig. 12. Traffic light simulation project with Arduino Microcontroller

Based on Figure 11, is a circuit diagram of a simple simulator project with a simple traffic light using an Arduino microcontroller, each LED is given an R or resistor with a value of 330 ohms, can be mounted on the ground or on the positive part. For deadlines that enter the VCC using a 9 volt DC deadline, you can use a battery or power supply.

Based on Figure 12, a traffic light simulation project that is tried by using PCB Breadboard is more easy and versatile. This simulator is very useful to test the software that we made on Arduino can work properly, without a simulator, we don't know what happened to the software.

5 CONCLUSIONS

In this study, red lights (except All-red) were not determined for the duration of the lights, to determine the duration of the lights, using the accumulation of all lights that have been determined using input data.

Based on the method used to determine the duration of the green light, it is determined by the number of vehicle queues at each of the four intersections. So in this study, the method must be combined with a new method, namely the full-speed method at the beginning of the start Full-start-speed (FSS). This method is to change the driver's behavior from slow motion to fast motion, with this method will produce a very long distance using a relatively short duration. Of course, you have to go through preparation, practice, and testing, so that the driver will be reliable and professional and not become a driver who is just learning to drive a vehicle all his life.

ACKNOWLEDGEMENTS

Special thanks to Mr. Prof. DR. Ir. R. Marsuki Iswandi, MS as the Postgraduate Director of the University of Halu Oleo, who has provided much guidance, motivation and by promoting local wisdom and originating locally to build an era of 4 point zero that is more creative and innovative.

REFERENCES

- [1] Alam, J. dan Pandey (2016) "Design and Analysis of a Two Stage Traffic Light System Using Fuzzy Logic," *Information Technology & Software Engineering*, (November 2015). doi: 10.4172/2165-7866.1000162.
- [2] Yusupbekov, N. R. *et al.* (2016) "Application of soft-computing technologies to the traffic control system design problems," *Procedia - Procedia Computer Science*. Elsevier Masson SAS, 102(August), hal. 540–546. doi: 10.1016/j.procs.2016.09.440.
- [3] Winardi, S. *et al.* (2019) "Determination Of Traffic Lights Duration By Identification Of Vehicle Numbers Using IoT," *International Journal of Computer, Network Security and Information System (IJCONSIST)*, (September), hal. 6–12.
- [4] Otasowie, O. dan Sunday, O. O. (2018) "Simulation Of An Intelligent Traffic Light System Embedded Technique," *Anale. Seria Informatică*, XVI.
- [5] Faruk, E. *et al.* (2019) "Density Based Traffic Control," *International Journal of Advanced Engineering, Management and Science (IJAEMS)*, (August 2016). Tersedia pada: www.ijaems.com.
- [6] Yeasmin, N., Zaman, R. dan Mouri, I. J. (2016) "Traffic Control Management And Road Safety Using Vehicle To Vehicle Data Transmission Based On Li-Fi Technology," *International Journal of Computer Science, Engineering and Information Technology (IJCEIT)*, 6(3), hal. 1–7. Tersedia pada: www.ijaems.com.
- [7] Marzukia, M. Z. A. *et al.* (2015) "Fault Monitoring System for Traffic Light," *Jurnal Teknologi (Sciences & Engineering)*, (April). doi: 10.11113/jt.v73.4407.
- [8] Darshan, J. *et al.* (2017) "Real Time Traffic Light Control System Using Image Processing," *International Journal of Engineering Technology Science and Research IJETSR*, 4(4), hal. 142–146. Tersedia pada: www.ijetsr.com.
- [9] Gafoor, S., Ki, K. dan Indore, B. (2016) "Optimizing City Traffic Light Management for Improving Traffic System in Smart Cities," *International Journal of Computer Applications*, 133(14), hal. 23–28. Tersedia pada: www.ijcaonline.org.
- [10] Krause, M., Knott, V. dan Benger, K. (2015) "Implementing the Tactile Detection Task in a Real Road Experiment to Assess a Traffic Light Assistant," *The Eighth International Conference on Advances in Computer-Human Interactions Implementing*, 5(Figure 1), hal. 43–48.
- [11] Oskarbowski, J. *et al.* (2016) "Analysis of signalized intersections in the context of pedestrian traffic," *Transportation Research Procedia*. Elsevier B.V., 14, hal. 2138–2147. doi: 10.1016/j.trpro.2016.05.229.
- [12] Singh, S. K. dan Khubani, P. (2018) "Traffic Analysis of Grade Intersections and Measures of Congestion Mitigation at Indore," *Journal of Engineering and Manufacturing Technology*, 3(2), hal. 10–19. Tersedia pada: <http://www.pyrexjournals.org/pjemt>.
- [13] Warren, J. (2011) *Arduino Robotics - Arduino Robotics By Jhon-David Warren, Josh Adams and Harald Molle.pdf*. New York: Apress. doi: 10.1007/978-1-4302-3184-4.

IJSER