Iot smart traffic with emergency



BY

**SUI LIP CHUAN**

**LIM CHIEN VERN**

session 2018

**The project report is prepared in partial fulfilment for the**

**Diploma in Computer Engineering**

January

penang skills developmet centre

July 2018

# DECLARATION

I hereby declare that this work has been done by me. I also declare that I have not engaged in any unauthorised act of copying or reproducing or attempt to copy / reproduce or cause to copy / reproduce or permit the copying / reproducing or the sharing and / or downloading of any copyrighted material or an attempt to do so of any protected material.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Sui Lip Chuan

Student ID: 201600039

Date: 13/7/2018

# ABSTRACT

Traffic management system is a cornerstone of a Smart city. In the current problems of the world, urban mobility is one of the major problems, especially in metropolitan cities. Previous traffic management systems are not capable enough to tackle this growth of traffic on the road networks. The purpose of this paper is to propose a smart traffic management system using the Internet of Things and a decentralized approach to optimize traffic on the roads to manage all traffic situations more accurately.

This proposed system is overcoming the flaws of previous traffic management systems. The system takes traffic density as input from ultrasonic sensors, resultantly giving output as signals management into a database. An algorithm is used to predicts the traffic density for future to minimize the traffic congestion. Besides this, our device to control the traffic light are also used to prioritize the emergency vehicles like ambulance, fire brigade etc. by implementing our device in such vehicles. In the case of emergency situations, such as fire explosion or burning of something, the emergency vehicle can reach to the destination in the most minimum time.

Moreover, an email is sent to the emergency vehicle to determine which is the best route to take to reach to their destination in the most minimum time. An algorithm is used to calculate the route. In addition, the native user can see future traffic condition at a particular node. The proposed system is validated by constructing a prototype and deploying it in a city of Penang. A web application is also there to provide useful information in graphical formats to the higher authorities of the smart city which is fruitful in future road planning.

# ACKNOWLEDGEMENT

First and foremost, I would like to thank to our supervisor of this project, Mr.Teh Choon Leong for the valuable guidance and advice. He inspired us greatly to work in this project. His willingness to motivate us contributed tremendously to our project. I also would like to thank him for showing us some example that related to the topic of our project.

Besides, I would like to thank the authority of Penang Skill Development Centre (PSDC) for providing us with a good environment and facilities to complete this project. Also, for giving me opportunity and support to be independently implementing my Final Year Project (FYP) by applying those theories, concepts and knowledge that I’ve learned through my Diploma life in PSDC.

Furthermore, a special gratitude I give to Mr Christopher for guiding us and helping us whenever we have difficulties regarding the programming parts. He would gladly and willing to teach us whenever he had the time. Moreover, also a special thanks to my teammate, Lim Chien Vern, who did his best to complete his tasks. We are both satisfied with the work the we had done.

Finally, an honourable mention goes to our families and friends for their understandings and supports on us in completing this project. Without helps of the particular that mentioned above, we would face many difficulties while doing this project.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Declaration | i |
| Abstract | ii |
| Acknowledgement | iii |
| Table of Contents | iv,v,vi |
| List of Figures | Vii,viii,ix |
| List of Tables | x |
| **CHAPTER 1: INTRODUCTION** |  |
| * 1. Project Background   2. Objectives   3. Problem Statement | 1  2  3 |
| **CHAPTER 2: LITERATURE REVIEW** |  |
| 2.1 Literature Survey | 4-5 |
| **CHAPTER 3: METHODOLOGY** |  |
| 3.1 Project Planning  3.2 Delegation of works  3.3 Project Development & Flow Chart  3.4 Project Flow Chart  3.5 Project Block Diagram  3.6 Material & Hardware Used  3.6.1 Arduino Mega 2560  3.6.2 Ethernet Shield W5100  3.6.3 LED Bulbs (Green, Yellow, Red)  3.6.4 2-Digit 7 Segment Display  3.6.5 Ultrasonic Sensor (HC-sr04)  3.6.6 Soldering Board & Jumper Wires  3.7 Software Used  3.7.1 Xammp (Module Run: Apache HTTP & MySQL)  3.7.2 Sublime Text  3.7.3 Arduino  3.8 Software Development  3.8.1 ER Diagram  3.8.2 Database Connection  3.8.3 (typical.php) Page  3.8.4 (live.php) Page  3.8.5 (emergency.php) Page  3.9 Project Cost | 6  7  8-19  20-21  22  23  23  24-25  26  27  28-30  31  32  32-35  36  37  38  38-39  40  41-44  45-47  48-51  52 |
| **CHAPTER 4: TEST, DATA COLLECTION & RESULT** |  |
| 4.1 System Integration  4.2 Testing Phase  4.2.1 Data Collection  4.2.2 Result  4.3 Finalised model design  4.3.1 Top View  4.3.2 Front View  4.3.3 Side View  4.3.4 Circuit View | 53-56  57  57  58  59  59  59  60  60 |
| **CHAPTER 5: PROBLEMS – SOLUTIONS & PROJECT LIMITATION** |  |
| 5.1 Problems & Solutions  5.2 Project Limitation | 61-62  63 |
| **CHAPTER 6: CONCLUSION & RECOMMENDATION** |  |
| 6.1 Conclusion  6.2 Future Recommendations | 64  64 |
| **REFERENCES** | 65 |
| **APPENDIX** |  |
| Appendix A: Gantt Chart  Appendix B: Datasheet  Appendix C: Source Code | 66  67-68  69-114 |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| Figure 1: Intersection Road Layout | 8 |
| Figure 2: Congestion Mode Demonstration | 9 |
| Figure 3: Database for Congestion | 10 |
| Figure 4: Congestion Mode Flow Chart | 11 |
| Figure 5: Data Path Flow Chart | 12 |
| Figure 6: Smart Mode Demonstration | 13 |
| Figure 7: Smart Mode Flow Chart | 14 |
| Figure 8: Email notification | 15 |
| Figure 9: (emergency.php) Page | 15 |
| Figure 10: Traffic Light Control Page | 16 |
| Figure 11: Emergency Mode Demonstration | 17 |
| Figure 12: Emergency Mode Flow Chart | 18 |
| Figure 13: Email Sending Flow Chart | 19 |
| Figure 14: Finalized Model | 20 |
| Figure 15: Project Flow Chart | 21 |
| Figure 16: Project Block Diagram | 22 |
| Figure 17: Arduino Mega 2560 | 23 |
| Figure 18: Ethernet Shield 5100 | 25 |
| Figure 19: LED Bulbs | 26 |
| Figure 20: 2-Digit 7 Segment Display | 27 |
| Figure 21: Signal of Ultrasound Waves. | 28 |
| Figure 22: Ultrasonic Sensor (HC-sr04) | 28 |
| Figure 23: Ultrasonic Sensor (HC-sr04) Pin Description | 29 |
| Figure 24: Ultrasonic Sensor (HC-sr04) Working Principle | 30 |
| Figure 25: Soldering Board & Jumper Wires | 31 |
| Figure 26: HTTP Request & Response | 34 |
| Figure 27: ER Diagram | 38 |
| Figure 28: (case\_fyp) database table | 39 |
| Figure 29: (ambulance) database table | 39 |
| Figure 30: (typical.pHp) Page | 42,44 |
| Figure 31: (live.pHp) Page | 45,46 |
| Figure 32: Congestion Database Table | 46 |
| Figure 33: (emergency.php) Page | 48 |
| Figure 34: Best Route Calculation | 49 |
| Figure 35: Input interface for emergency | 49 |
| Figure 36: fa-fa icon for ambulance | 50 |
| Figure 37: Schematic Design Diagram | 53 |
| Figure 38: Schematic Diagram | 54 |
| Figure 39: 7 Segment Connection | 56 |
| Figure 40: Ultrasonic Sensor (HC-sr04) Graph (No.of getting sense against speed) | 58 |
| Figure 41: Traffic condition graph (Count against Time) | 58 |
| Figure 42: Top View of Finalized Model | 59 |
| Figure 43: Front View of Finalized Model | 59 |
| Figure 44: Side View of Finalized Model | 60 |
| Figure 45: Circuit View of Finalized Model | 60 |
| Figure 46: Messier Wire | 61 |
| Figure 47: Tidier Wire | 61 |
| Figure 48: Solder Board | 62 |
| Figure 49: Bread Board | 62 |
| Figure 50: Soldering error | 62 |
| Figure 51: Arduino Mega 2560 Datasheet | 67 |
| Figure 52: 2-Digit 7 Segment Display | 68 |

**LIST OF TABLES**

|  |  |
| --- | --- |
| Table 1: Timing for Traffic Light | 9 |
| Table 2: The Internet’s Request/Response Way of Working | 33 |
| Table 3: Project Cost | 52 |
| Table 4: Pins Listed Connected to Arduino | 56 |
| Table 5: Ultrasonic Sensor (HC-sr04) Testing | 57 |

**CHAPTER 1: INTRODUCTION**

* 1. **Project Background**

It is undoubtedly that the rapid growth in the vehicle ownership is one of the measures for economic growth of country. New York, London, Las Vegas, Hong Kong, Kuala Lumpur and others great cities in the world are often impressing us with lots of vehicles saturated on the road. This hustle and bustle lifestyle in cities which filled with honk sound are the real image in every cities and metropolis. Congestion is detrimental predicament that we need to tackle now for good.

Traffic congestion is a condition on road networks that occurs as use increases, and is characterized by slower speeds, longer trip times, and increased vehicular queuing. The most common example is the physical use of roads by vehicles. Congestion could cause many effects including delays, fuel consumption, pollution and road rage but most importantly, stuck emergency vehicles which can put a life in a danger state.

The exploitation of new trends and technologies requires fast transportation of goods, machinery and manpower for various reasons. Besides, we could not allow congestion to deteriorate our life anymore. Hence, our team is going to get rid of this problem with this project, IOT Smart traffic with Emergency.

* 1. **Objectives**

The current traffic light system is fixed, where their time interval is fixed to pass the either side of the road, which causes congestion problem. In order to solve the problem, we proposed this project to control the congestion problem on a real time basis. Moreover, it will be very unfortunate if any of the emergency vehicle is unable to reach their destination when there is a critical situation that a life is in danger. Therefore, this project will carry out the objectives to solve this problem.

The main objectives of this project are to create a safe and efficient road traffic, the following objectives are listed below:

* To upgrade the current traffic system.
* To help reduce waiting and traveling times.
* To monitor the traffic on a real time basis.
* To enable emergency vehicles reach their destination in the most minimum time.
  1. **Problem Statement**

We as a living human would definitely demand for a smooth traffic flow as in it will affect our life. In this current era, it is undoubtedly that a lot of cities are suffering a lot from “Traffic Congestion”. This issue occurs is because the existing systems of manual control of traffic or predefined time for change of traffic lights aren’t efficient enough. With this current issue not being solved it will cause many effects such as delays, fuel consumption, pollution, road rage, waste of drivers’ time and resources, but most importantly emergency vehicles that are stuck in between the traffic which can put a life in danger. Traffic congestion often happens during the beginning and the end of working hours or break time.

|  |  |
| --- | --- |
| Working Hours | Break Time |
| (In between) 7:00 a.m. - 8:00 a.m. (is the beginning of working hours)  (In between) 5:00 p.m. - 6.00 p.m. (is the end of working hours) | 12:00 p.m. - 1:00 p.m. |

This issue often occurs at the intersection road as in there is a lot of vehicles crossing from one road to another. We must solve this problem as soon as possible and do more research on this matter. Internet of Things a.k.a (IoT) is the solution to solve these problems.

**CHAPTER 2: LITERATURE REVIEW**

**2.1 Literature Survey**

In smart cities traffic related problems are controlled by using IoT in right way. Reservation-based system Connected and Automated Vehicles (CAV) and smart parking system are proposed as a part of traffic management by using IoT. Sensing and classifying roadway obstacles provides accident free environment and also a smooth drive to the vehicles.

The 2014 revision of world urbanization prospects, [1]which contains the latest estimates of the urban and rural populations of 233 countries. Population censuses are the most commonly used sources of data, although estimates obtained from population registers or administrative statistics. Classifying an area as urban may be based: a minimum population threshold; population density; proportion employed in non-agricultural sectors; the presence of infrastructure such as paved roads, electricity, piped water or sewers; and the presence of education or health services. In compiling information on city population size, the Population Division has endeavoured to use data or estimates based on the concept of urban agglomeration. The DOI: 10.18535/ijecs/v5i11.45 Anaswara R, IJECS Volume 05 Issue 11 Nov. 2016 Page No.18983-18986 Page 18984 method to project city populations as the last observed city growth rate converges towards an expected value, estimated on consistent and timely data on global trends in urbanization and city growth are critical for assessing current and future needs with respect to urban growth and for setting policy priorities to promote inclusive and equitable urban and rural development. Successful sustainable urbanization requires competentancy.

K.Dresner[2] propose a reservation based system for alleviating traffic congestion, specifically at intersections, and under the assumption that the cars are controlled by agents. A custom simulator is created to measure the different delays associated with conducting traffic through an intersection. A precise metric for evaluating the quality of traffic control at an intersection. This reservation-based system can perform two to three hundred times better than traffic lights and it can smoothly handle much heavier traffic conditions. This system very closely approximates an overpass, which is the optimal solution for the problem.

Current methods for controlling traffic, specifically at intersections, will not be able to take advantage of the increased sensitivity and precision of autonomous vehicles as compared to human drivers. K.Dresner[3] describe an autonomous intersection management system. Drivers and intersections in this mechanism are treated as autonomous agents in a multiagent system. In this multiagent system, intersections use a new reservation-based approach built around a detailed communication protocol. Demonstrate in simulation that new mechanism has the potential to significantly outperform current intersection control technology—traffic lights and stop signs. It subsumes the most popular current methods of intersection control. the basis of the city population and the growth rate of the overall urban population in the country. Globally, more people live in urban areas than in rural areas. Levels of urbanization vary greatly across regions. Most megacities and large cities are located in the global South. One in five urban dwellers worldwide lives in a medium-sized city with 1 million to 5 million inhabitants. Some cities have experienced population decline since 2000, most of which are located in low-fertility countries of Asia and Europe with stagnating or declining populations. Diversified policies to plan for and manage the spatial distribution of the population and internal migration are needed. Policies aimed at a more balanced distribution of urban growth. Accurate, this article also presents two extensions to the mechanism, the first extension allows the system to control human-driven vehicles in addition to autonomous vehicles described the construction of the simulator itself, as well as the communication protocol, the intersection manager, the driver agent, and several intersection control policies. The first policy, FCFS is only for fully autonomous vehicles. FCFS-Light extends FCFS to allow human interoperability using existing traffic light infrastructure. The last policy, FCFS-Emerg, extends FCFS to give priority to emergency vehicles without significant increasing delays for other vehicles. The second gives priority to emergency vehicles without significant cost to civilian vehicles. In this there is no switch among several different policies, learning from reservation histories which policy is best suited to particular traffic conditions, could significantly improve performance. There is no light model that could react not react to the presence of individual vehicles, might better be able to exploit the abilities of autonomous vehicles, without adversely affecting human drivers.

**CHAPTER 3: METHODOLOGY**

Methodology is the analysis of the principles of methods and rules. This chapter describes the methodology employed and considerations taken into account for this project. It begins with the discussion of the project methodology, followed by the system design procedure, techniques and tools utilized in this work.

This project follows three major footstep, which is planning, implementing or development and data analysis. All the method used for finding and analysing data regarding the project related.

**3.1 Project Planning**

This is the most important section because planning and scheduling acts as a guide that we had to follow to achieve the objective and to complete the project. The proposal stages including the sketching design, project cost consideration and projects research. Then we continued with the layout construction design, material selection and electrical design including the hardware and software on the projects. The Gantt Chart can be referred on the appendix.

**3.2 Delegation of Works**

To achieve the objectives and meeting project timeline, works related to this project were divided to each project members as per below: -

**Sui Lip Chuan (SID: 201600039)**

1. Research Data Evaluation (Software & Firmware)
2. Firmware block diagram (Planning)
3. Firmware design and development
4. Simulation and Analysis
5. Programming Software (Arduino/pHp/HTML)
6. Linking Arduino->pHp->pHpmyAdmin (Database)
7. Final Testing (Software)

**Lim Chien Vern (SID: 201600040)**

1. Research Data Evaluation (Hardware)
2. Hardware Selection
3. Hardware Design & Development
4. Simulation and Analysis (Hardware)
5. Construction of the circuit & Map
6. Wiring & Soldering
7. Fabrication (Hardware)
8. Final Testing (Hardware)

**3.3 Project Development & Flow Charts**

This project consists of input part which are the sensors, process part which is the database and output part which is from Arduino Mega 2560 5V. The combination the three parts is important to ensure the system operate automatically. Moreover, this project also consists of 3 modes which is:

1. Congestion Mode (operates at 5:00 a.m. - 12.00 a.m.)
2. Smart Mode. (operates at 12:01 a.m. - 4:59 a.m.)
3. Emergency Mode.

The Ultrasonic Sensor (HC-sr04) will act as an input part and 1 sensor only sense one lane but for 2 directions which is left and right side of the road. This project consist of 2 Ultrasonic Sensor (HC-sr04), the first Ultrasonic Sensor (HC-sr04)[S1] shown in Figure 1 is for Smart Mode, the second Ultrasonic Sensor (HC-sr04)[S2] shown in Figure 1 is for Congestion Mode. However, for the Emergency Mode it doesn’t need any sensors to operate. It uses a webpage to operate. Function of the modes are next page:

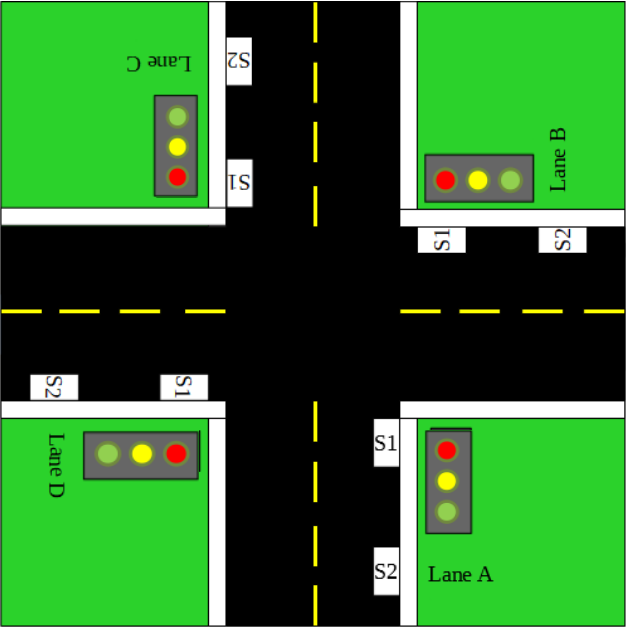


Figure 1: Intersection Road Layout

**Mode-1[Congestion Mode]:**

The Ultrasonic Sensor (HC-sr04)[S2] shown in to Figure 2. will sense whether there is vehicle present on the road. If it senses a vehicle then it will then the green light be will turned on for 6seconds shown in Figure 2 & Table 1 (Lane A). Furthermore, it will also send a data into the database refer Figure 3.

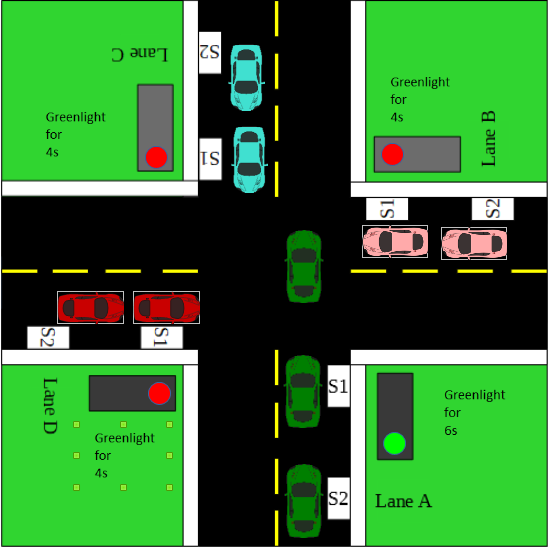


Figure 2: Congestion Mode Demonstration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lanes | Vehicle Presence | Time interval Red Light(seconds) | Time interval Yellow Light (seconds) | Time interval Green Light (seconds) |
| Lane A [Route 3] | Yes | 4 | 2 | 6 |
| Lane B [Route 2] | No | 12 | 2 | 4 |
| Lane C [Route 1] | No | 16 | 2 | 4 |
| Lane D [Route 4] | No | 20 | 2 | 4 |

Table 1: Timing for Traffic Light

As you can see the Figure 3 shown below is the database of this project and it is particular for the Ultrasonic Sensor (HC-sr04). It also comes with a time whenever it receives data from the Ultrasonic Sensor (HC-sr04).

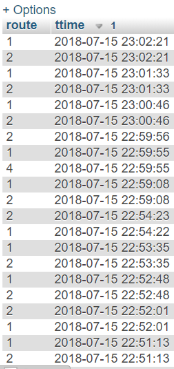


Figure 3: Database for Congestion

**Mode-1[Congestion Mode]: Flow Chart**

Start

Sensor Sense

Green light

for 4seconds

Car

Present??

No

Yes

Green light

for 6seconds

End

Figure 4: Congestion Mode Flow Chart

**Mode-1[Congestion Mode]: Data Path Flow Chart**

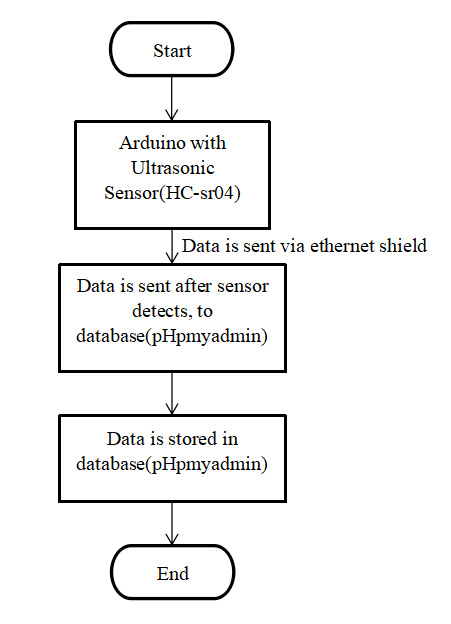


Figure 5: Data Path Flow Chart

**Mode-2[Smart Mode]:**

During the night hours, in the current traffic light system. It is fixed that the green light will be turned on with a sequence of Lane A to Lane D. It would be unnecessary to wait for the sequence of the traffic lights.

The Ultrasonic Sensor (HC-sr04)[S1] refer to Figure 6. will sense whether there is vehicle present on the road. If it senses a vehicle then it will straight turn the green light to the road that has vehicle on it refer Figure 6.

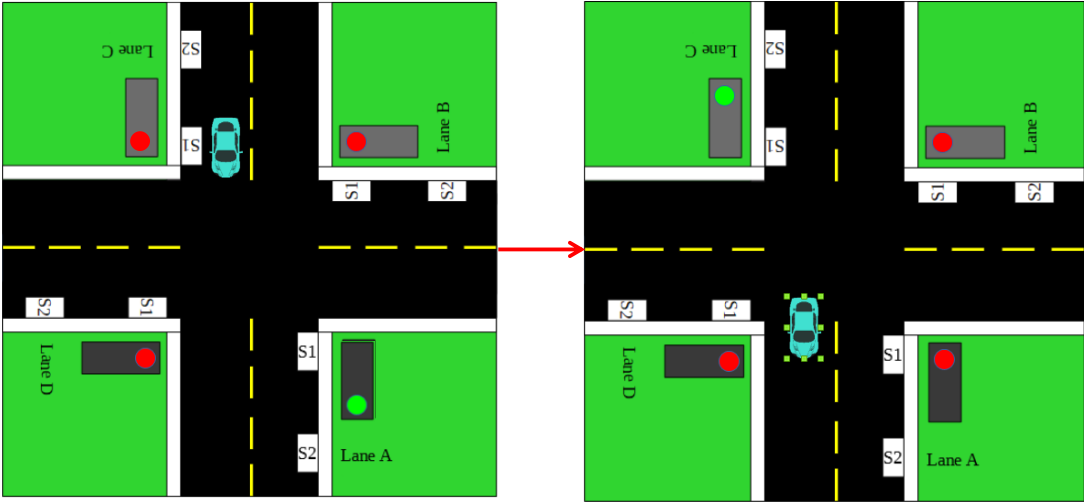


Figure 6: Smart Mode Demonstration

**Mode-2[Smart Mode]: Flow Chart**

Start

Sensor Sense

D

C

B

A

Car

Car

Car

Car

Yes

Turn on green light for that lane

End

Figure 7: Smart Mode Flow Chart

**Mode-3[Emergency Mode]:**

If an incident happened, authority will then receive a location where the incident had happened. The authority will then key-in the location in our webpage shown in Figure 9, requesting for Police/Ambulance/Fire Fighter. Then it will send an email shown in Figure 8. to the Police/Ambulance/Fire Fighter what incident had happened, and which is the best route to take to reach to the location in the most minimum time. If the emergency vehicles need to go through the intersection road, but if the traffic light is red, the emergency vehicle will have the ability to change the red light to green light shown in Figure 10 & Figure 11. However, before completely changing to green light, it will give a warning yellow light.

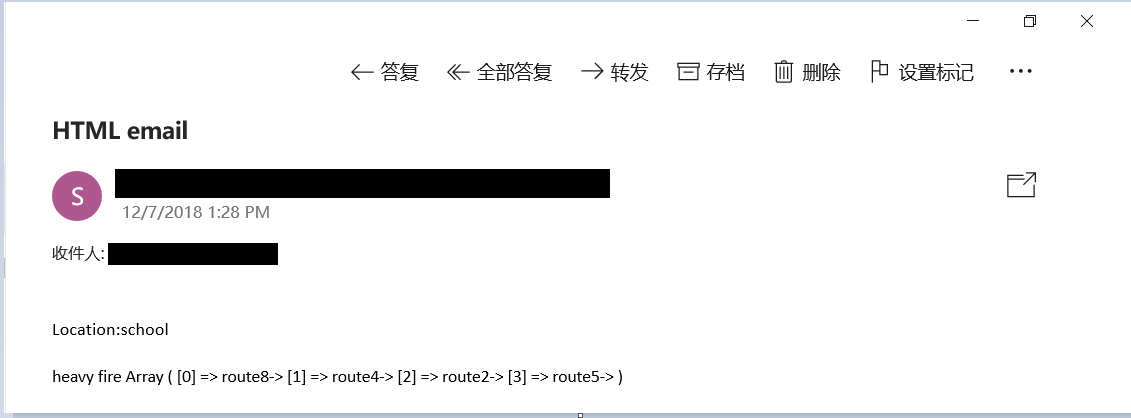


Figure 8: Email notification

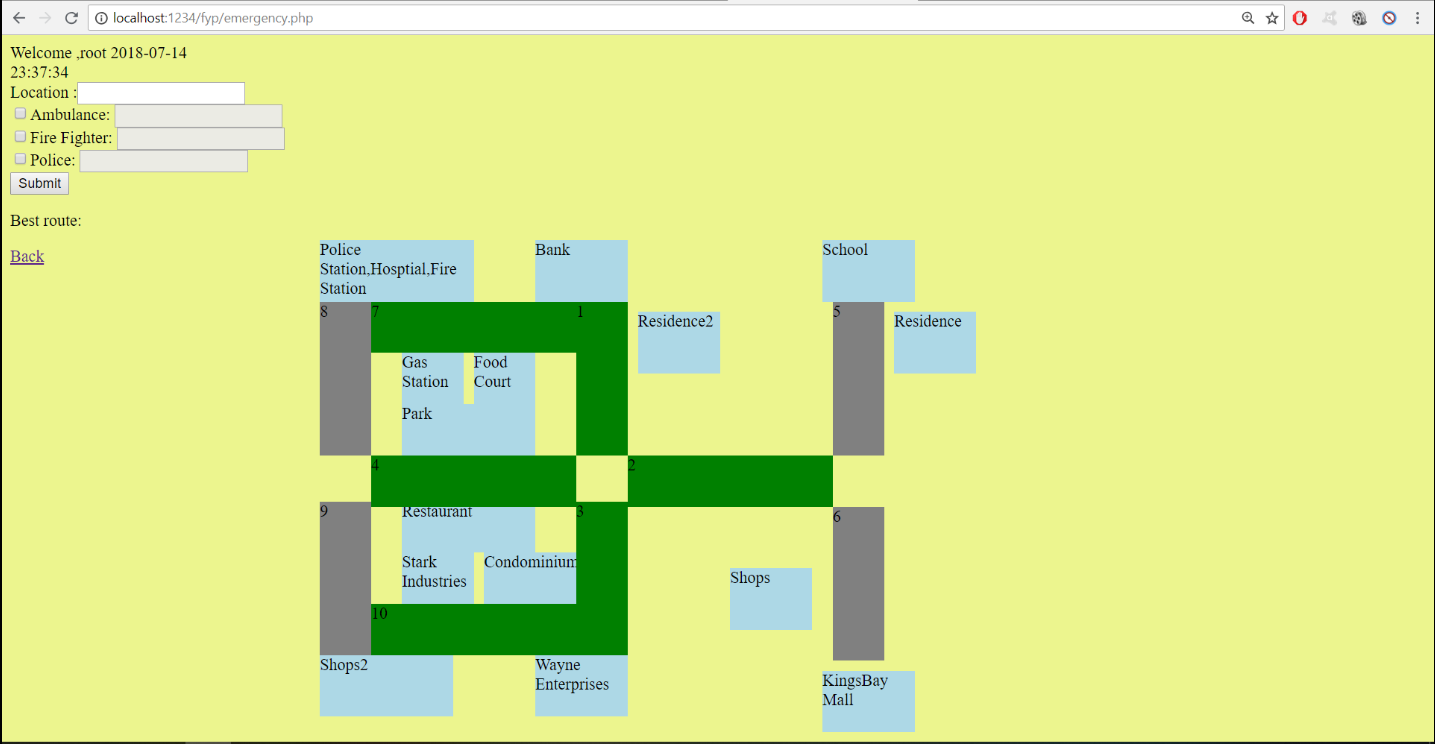


Figure 9: (emergency.php) Page

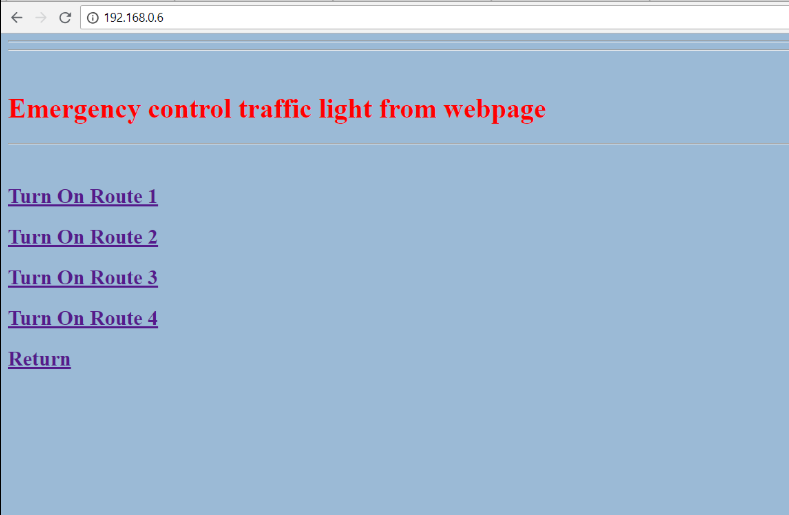


Figure 10: Traffic Light Control Page

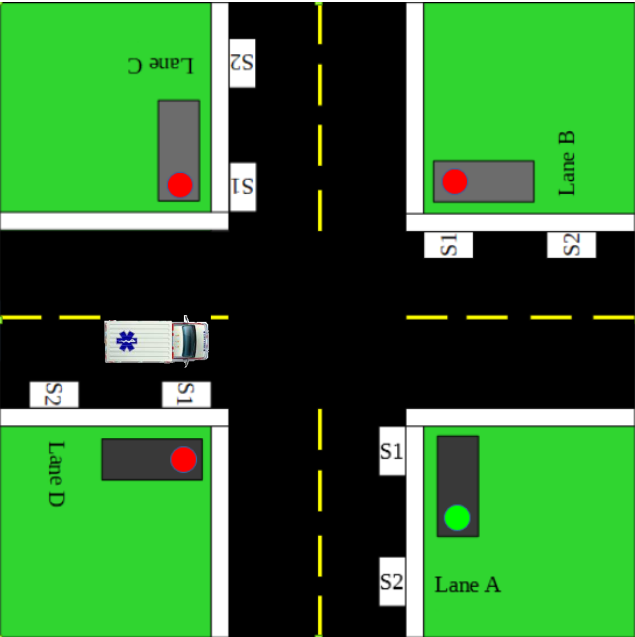


Figure 11: Emergency Mode Demonstration

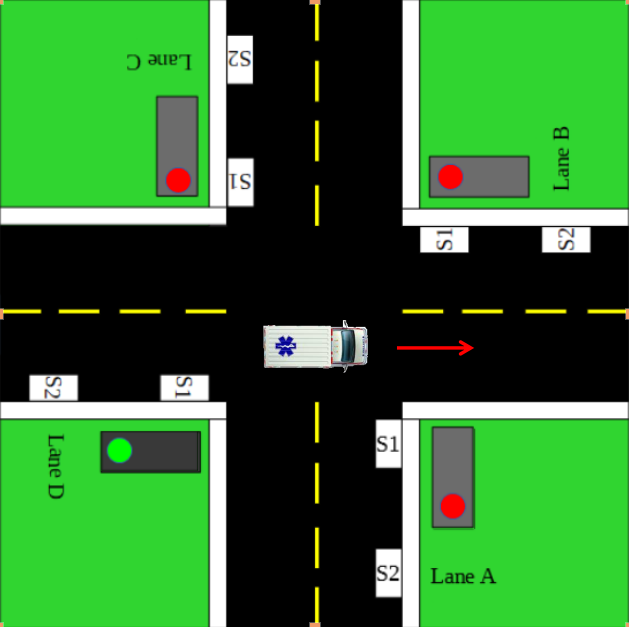


Figure 11: Emergency Mode Demonstration

**Mode-3[Emergency Mode]: Flow Chart**

Start

Submit

Traffic condition based on database

Choose the best path

Taking intersection route?

Take alternative route?

No

Yes

Require green light?

Turn on green light

Yes

End

Figure 12: Emergency Mode Flow Chart

**Mode-3[Emergency Mode]: Email sending**

Start

Input Data

Submit

Send email to authority

Email received by authority

End

Figure 13: Email Sending Flow Chart

**3.4 Project Flow Chart**

From the flow chart in Figure 15, it shows the plan for the project to follow all the way through the semester to complete the project. It started with a planning of how the project will be doing, what are the objectives of the project. After all the planning are discussed with the supervisor, after having the consent from our supervisor. We then started researching what and which hardware and software are required to make the project works. We then started to design the concept of the project into two parts which is software and hardware.

Hardware part, material and hardware are then decided and selected to use for the project. After collecting the material and hardware the we started to design the model, testing and debugging is conducted during the designing process. After making sure there is no bugs and errors, then we finalized the model.

Software part, software and component are then decided and selected to use for the project. After deciding software and component the we started to design the program, testing and debugging is conducted during the designing process. After making sure there is no bugs and errors, then we finalized the program.

At last a final test for the project is conducted to make sure there is no errors and bugs. After the final testing, we then finalized the project shown in Figure 14 and are prepared for the final presentation.

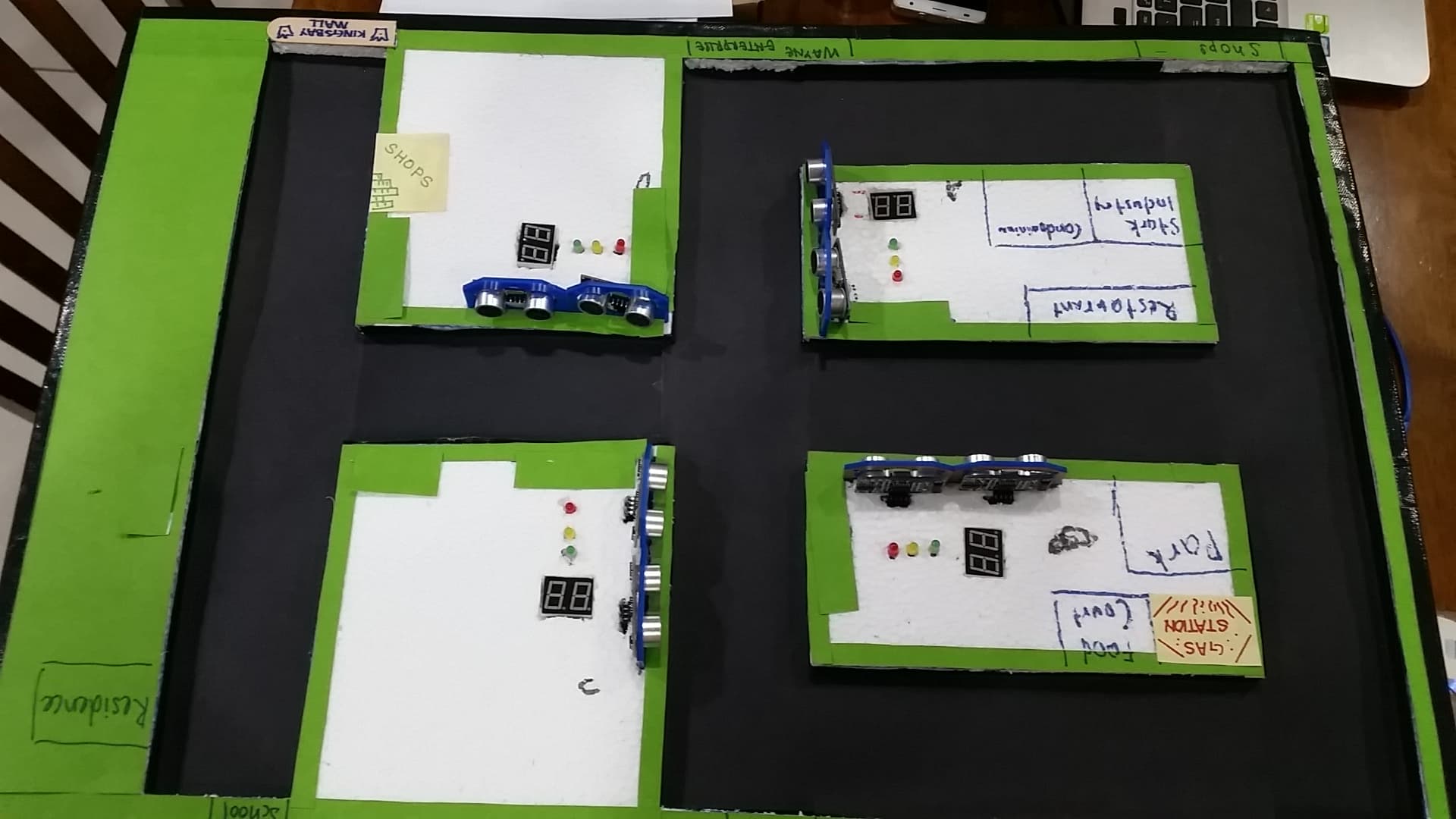


Figure 14: Finalized model

Start

Planning

Research

Software

Design concept

Hardware

Selection of software component

Selection of material & hardware

Design the program

Design the model

Test

Test

Finalize the program

Finalize the model

Final Test

Complete the making of hardware. E.G. Maps, Road ETC

Preparing report

Presentation

End

Figure 15: Project Flow Chart

**3.5 Project Block Diagram**

Ultrasonic Sensor (HC-sr04)

Arduino Mega 2560

2-digit 7 segment display

LED Bulbs (Green,Yellow, Red)

Ethernet Shield W5100

Website

Figure 16: Project Block Diagram

**3.6 Material & Hardware Used**

The Material and Hardware listed below are essential components to have for this proposed system to work. The components include:

**3.6.1 Arduino Mega 2560**

First and foremost, the project must have a microcontroller in order to run. The **Arduino Mega 2560** is a microcontroller board based on the [ATmega2560](http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf). It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

Arduino Mega 2560 is shown in Figure 17.

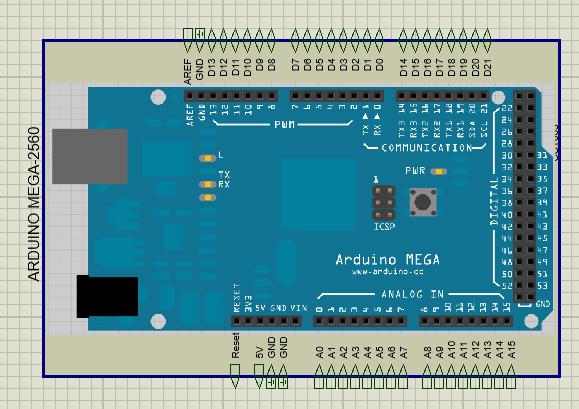


Figure 17: Arduino Mega 2560

**3.6.2 Ethernet Shield W5100**

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The latest revision of the shield adds a micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Duemilanove and Mega (using the Ethernet library coming in Arduino 0019). An SD card library is not yet included in the standard Arduino distribution, but the sdfatlib by Bill Greiman works well. See this tutorial from Adafruit Industries for instructions (thanks Limor!).

The latest revision of the shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up. The original revision of the shield contained a full-size SD card slot; this is not supported.

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 11, 12, and 13 on the Duemilanove and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general i/o. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

The shield provides a standard RJ45 ethernet jack. The reset button on the shield resets both the W5100 and the Arduino board. The shield contains a number of informational LEDs:

* PWR: indicates that the board and shield are powered.
* LINK: indicates the presence of a network link and flashes when the shield transmits or receives data.
* FULLD: indicates that the network connection is full duplex.
* 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s) .
* RX: flashes when the shield receives data.
* TX: flashes when the shield sends data.
* COLL: flashes when network collisions are detected.

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino. Arduino Ethernet Shield W5100 R3 is shown in Figure 18.



Figure 18: Ethernet Shield W5100

**3.6.3 LED Bulbs (Green, Yellow, Red)**

The main purpose of LED Bulbs (Green, Yellow, Red) is for traffic lights.

LED Bulbs (Green, Yellow, Red) is shown in Figure 19.



Figure 19: LED Bulbs

**3.6.4 2-Digit 7 Segment Display**

The 7-segment display, also written as “seven segment display”, consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimal and Hex) to be displayed. An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point, (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package. These individually LED pins are labelled from a through to g representing each individual LED. The other LED pins are connected together and wired to form a common pin.

So by forward biasing the appropriate pins of the LED segments in a particular order, some segments will be light and others will be dark allowing the desired character pattern of the number to be generated on the display. This then allows us to display each of the ten decimal digits 0 through to 9 on the same 7-segment display.

2 Digit 7 Segment Display is shown in Figure 20.

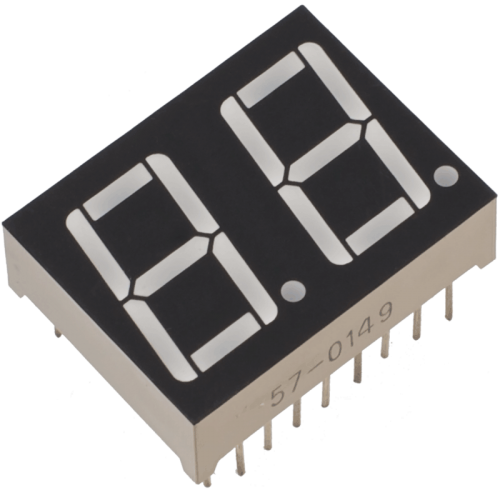


Figure 20: 2-Digit 7 Segment Display

**3.6.5 Ultrasonic Sensor (HC-sr04)**

The ultrasonic sensor works on the principle of SONAR and RADAR system which is used to determine the distance to an object. In this particular project, Ultrasonic Sensor (HC-sr04) is used to detect vehicles on road.

An ultrasonic sensor generates the high-frequency sound (ultrasound) waves. When this ultrasound hits the object, it reflects as echo which is sensed by the receiver as shown in below Figure 21.

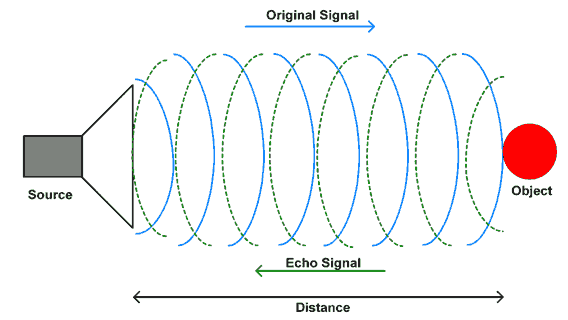


Figure 21: Signal of Ultrasound Waves.

**Ultrasonic Working Principle**

By measuring the time required for the echo to reach to the receiver, we can calculate the distance. This is the basic working principle of Ultrasonic module to measure distance. Ultrasonic Sensor (HC-sr04) is shown in Figure 22.



Figure 22: Ultrasonic Sensor (HC-sr04)

**Ultrasonic Module**

HC-SR-04 has an ultrasonic transmitter, receiver and control circuit.

In ultrasonic module HCSR04, we have to give trigger pulse, so that it will generate ultrasound of frequency 40 kHz. After generating ultrasound i.e. 8 pulses of 40 kHz, it makes echo pin high. Echo pin remains high until it does not get the echo sound back. So, the width of echo pin will be the time for sound to travel to the object and return back. Once we get the time we can calculate distance, as we know the speed of sound.

HC-SR04 can measure up to range from 2 cm - 400 cm.

# ****HC-SR04 Pin Description****



Figure 23: Ultrasonic Sensor (HC-sr04) Pin Description

**VCC** - +5 V supply

**TRIG** – Trigger input of sensor. Microcontroller applies 10 us trigger pulse to the HC-SR04 ultrasonic module.

**ECHO**–Echo output of sensor. Microcontroller reads/monitors this pin to detect the obstacle or to find the distance.

**GND** – Ground

# ****HC-SR04 Working Principle****

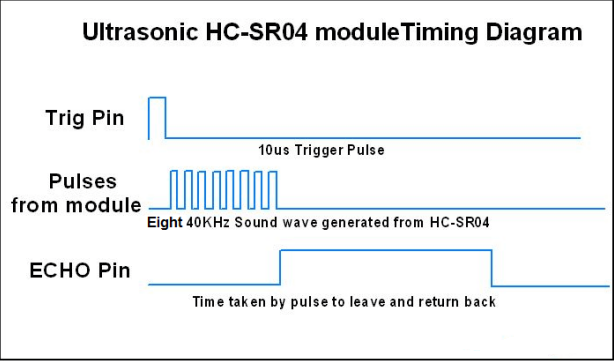


Figure 24: Ultrasonic Sensor (HC-sr04) Working Principle

**HC-SR04 Ultrasonic Module Timing Diagram**

1. We need to transmit trigger pulse of at least 10 us to the HC-SR04 Trig Pin.
2. Then the HC-SR04 automatically sends Eight 40 kHz sound wave and wait for rising edge output at Echo pin.
3. When the rising edge capture occurs at Echo pin, start the Timer and wait for falling edge on Echo pin.
4. As soon as the falling edge is captured at the Echo pin, read the count of the Timer. This time count is the time required by the sensor to detect an object and return back from an object.

**3.6.6 Soldering Board & Jumper Wires**

Soldering is the process of using a filler material (solder) to join pieces of metal together. Soldering occurs at relatively low temperatures (around 400 degrees Fahrenheit) as compared to brazing and welding, which actually melt and fuse the materials themselves at higher temperatures. In soldering the filler material becomes liquid, coats the pieces it is brought into contact with, and is then allowed to cool. As the solder cools it hardens, and the two materials are joined. Soldering is a quick way to join many types of materials, from copper pipe to stained glass. It creates an electrically conductive strong bond between components that can be re-heated (desoldered) if you should ever want to disconnect two items joined together. It’s great for joining electrical components and wires and is used in just about everything electronic.

Solder board & Jumper Wires are shown in Figure 25.



Figure 25: Soldering Board & Jumper Wires

**3.7 Software Used**

The list of Software below is essential to have for this proposed system to work. The software includes:

**3.7.1 Xampp**

Xampp plays a big role in this project as it helps Arduino to be connected to the database pHpmyAdmin. What is XAMPP?

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for:

* X (to be read as “cross” means cross-platform)
* Apache HTTP Server
* MySQL now known as MariaDB
* PHP
* Perl
* Tomcat

X means Cross-platform or multiplatform, is an attribute conferred to computer software or computing methods and concepts that are implemented and inter-operate on multiple computer platforms.



In XAMPP we run the following Module which is Apache and MySQL:

**Apache HTTP Server**

Apache is a public-domain (i.e. open source) **Web Server.** Apache is a web server notable for playing a key role in the initial growth of the World Wide Web and in 2009 became the first web server to surpass the 100 million web site milestone.

## The Internet’s Request / Response Way of Working

Here’s how the Internet’s, Browser -> Webserver, Request / Response paradigm works.

Whenever a Browser makes an http request such as**:** http**:**//www.opensourcevarsity.com/**index.html** the following happens**:**

|  |  |
| --- | --- |
| **http** | This is the protocol used for communication between the Browser and the Web server.Â  Since the Browser initiated the communication it has the privilege of setting up the communication protocol. |
| **://** | This is a separator that separates the protocol from the URL. |
| **www.opensourcevarsity.com** | This will be translated into a **name:value** pair i.e. an ip**:**URL by the **D**omain **N**ame **S**ervers (i.e. DNS servers). The DNS will translate the http call to an ip**:**websitename **74.86.170.172:www.opensourcevarsity.com** |

Table 2: The Internet’s Request/Response Way of Working

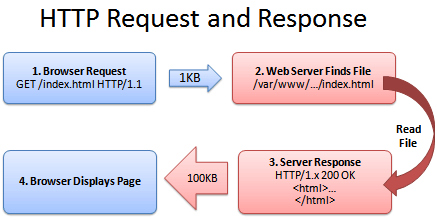
Hence only the **web server** physically located on the computer identified by the ip 74.86.170.172 will reply to the requesting Browser and a communication link will be established.

As soon as the web server (apache http server) receives a request for the file **index.html**, the web server will search within URL’s folder on its hard disk for the existence of index**.**html.

If the file is found it is streamed back to the requesting Browser, where it’s content is processed appropriately, and the web page is displayed in the Browser window.

If the file requested is not found an appropriate error message is streamed back to the requesting Browser which is displayed in the Browser window.

All of the above is described in a Figure .



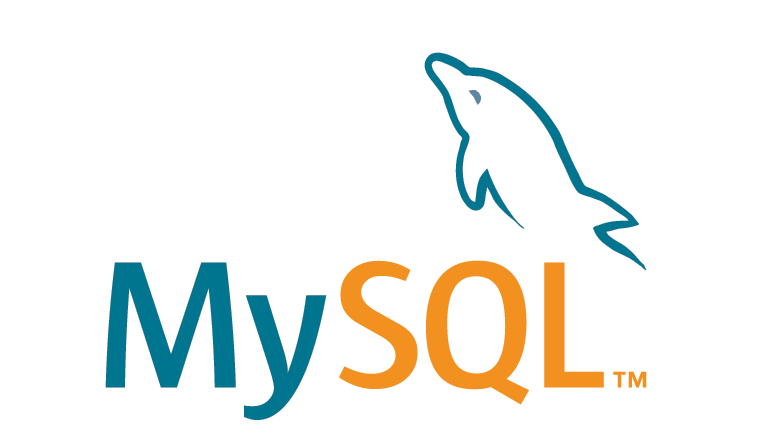
**Figure 26: HTTP Request & Response**

Hence, to sum up, a Browser **requests** for a file, the web server searches for this file and **responds** by delivering it to the Browser. Then the web server breaks all connections with the Browser. The web server does not remember the Browser in any way at all, hence this mode of communication is termed as Request / Response.

**MySQL**

MySQL is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). The features provide users the ability to browse and modify database structures, including schema objects and database storage, as well as maintain database security. An integrated query tool allows you to quickly create, edit and execute queries and scripts. Why use MySQL?

* MySQL is a database system used on the web server.
* MySQL is ideal for both small and large applications.
* MySQL is very fast, reliable and easy to use.
* MySQL complies on a number of platforms.
* MySQL is free to download and use.



**3.7.2 Sublime Text**

The software for coding for this project we use is Sublime Text. Sublime Text is a [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [source code editor](https://en.wikipedia.org/wiki/Source_code_editor) with a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API). It natively supports many [programming languages](https://en.wikipedia.org/wiki/Programming_languages) and [mark-up languages](https://en.wikipedia.org/wiki/Markup_languages), and functions can be added by users with [plugins](https://en.wikipedia.org/wiki/Plugins), typically community-built and maintained under [free-software licenses](https://en.wikipedia.org/wiki/Free_software_licenses). The list of features of Sublime Text:

* "Go to Anything," quick navigation to files, symbols, or lines
* "Command palette" uses adaptive matching for quick keyboard invocation of arbitrary commands
* [Simultaneous editing](https://en.wikipedia.org/wiki/Simultaneous_editing): simultaneously make the same interactive changes to multiple selected areas
* Python-based plugin API
* Project-specific preferences
* Extensive customizability via JSON settings files, including project-specific and platform-specific settings
* Cross-platform (Windows, [macOS](https://en.wikipedia.org/wiki/MacOS), and Linux) and Supportive Plugins for cross-platform.
* Compatible with many language grammars from [Text Mate](https://en.wikipedia.org/wiki/TextMate).



**3.7.3 Arduino**

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them. The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C++) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

****

* 1. **Software Development**

**3.8.1 ER Diagram**

Before we dive into the programming part, we would like to briefly explain the ER Diagram relationships for this project. As you can see in this ER Diagram, the ‘case\_fyp’ has a Primary Key (PK), meanwhile for the other 3 tables are connected with ‘case\_fyp’ has its own Primary Key (PK) and they inherit the Primary Key (PK) from ‘case\_fyp’ to become their Foreign Key (FK). You can refer the Figure 27 shown below.

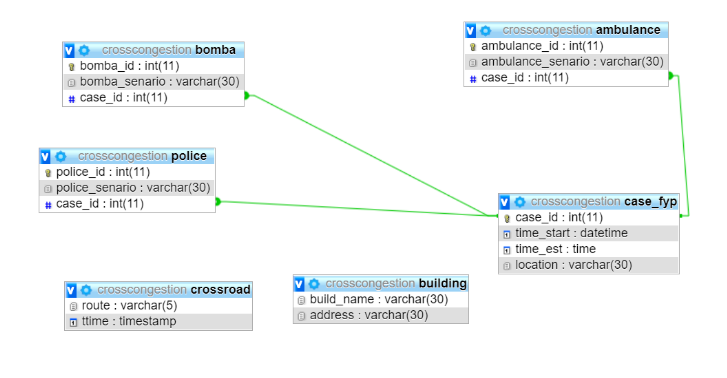
****

Figure 27: ER Diagram

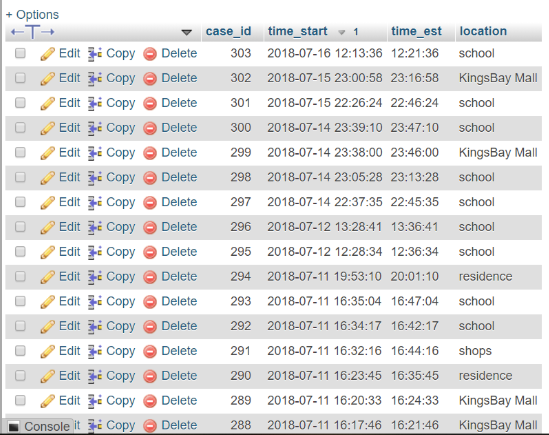
****

Figure 28: (case\_fyp) database table

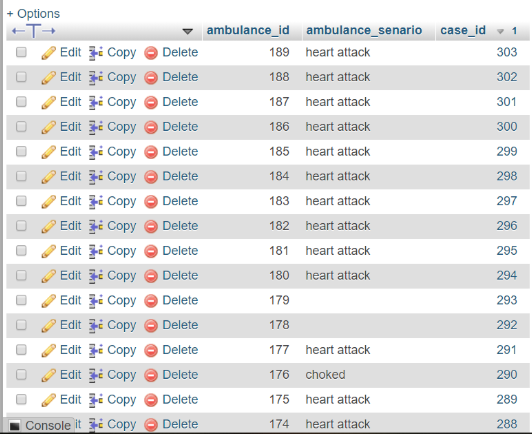
****

Figure 29: (ambulance) database table

**3.8.2 Database Connection**

The figure below shows the connection of database, as you can see:

DB\_USER: ‘your database user’

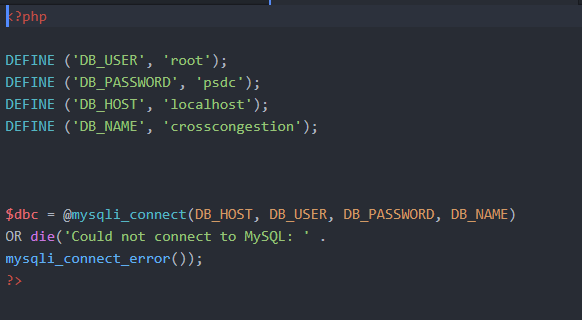
DB\_PASSWORD: ‘your database password’

DB\_HOST: ‘localhost’

DB\_NAME: ‘your database name’

Then save it, in this case our file is (‘C:xampp/mysqli\_connect\_cross.php’), F.Y.I [This file is not advisable to save within same folder for security purpose, must inside main xampp file].

Declare $dbc = @mysqli\_connect(DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME), $dbc will become the variable for connecting to database. In this case, if you want to connect a database, you can just call $dbc.



From there you can use this page at anywhere by calling it. Like so



**3.8.3 (typical.php) Page**

First, connect to your database.

php

The code below is for the <input=type> interface.



Dropdown Box for time

‘5:00 a.m. - 12:00 a.m’.

Dropdown Box to

Select ‘Sunday-Saturday’



This webpage is where you can check the historical data, you can choose which day you would like to see [Monday - Sunday] shown in Figure 30, you can also choose in between time. The result will show after u press the Submit button.

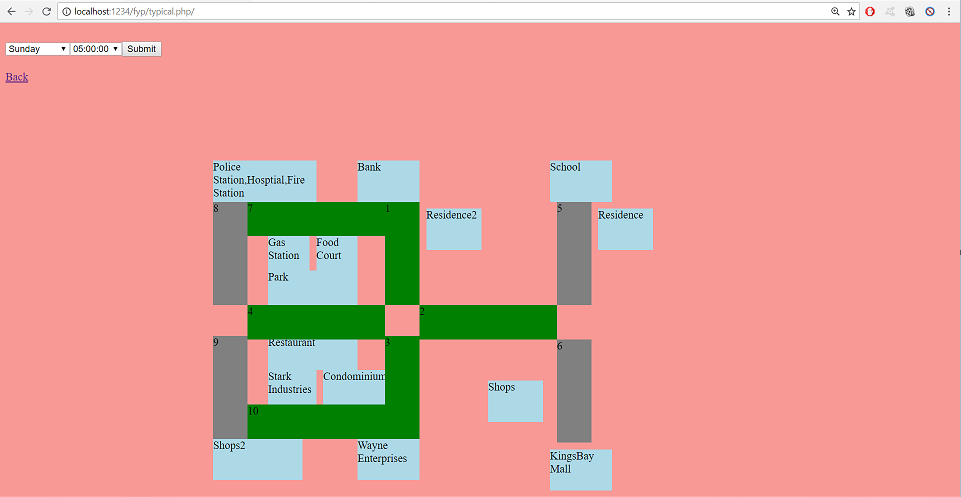


Figure 30: (typical.pHp) Page

As you can see the time shown in Figure 30 is Monday at 12:00 p.m. Route 1,4,7 is red because the traffic condition is in a worst state. Route 3,10 is yellow. Route 2 is green.

Red: Worst State

Orange: Normal State

Green: Smooth State

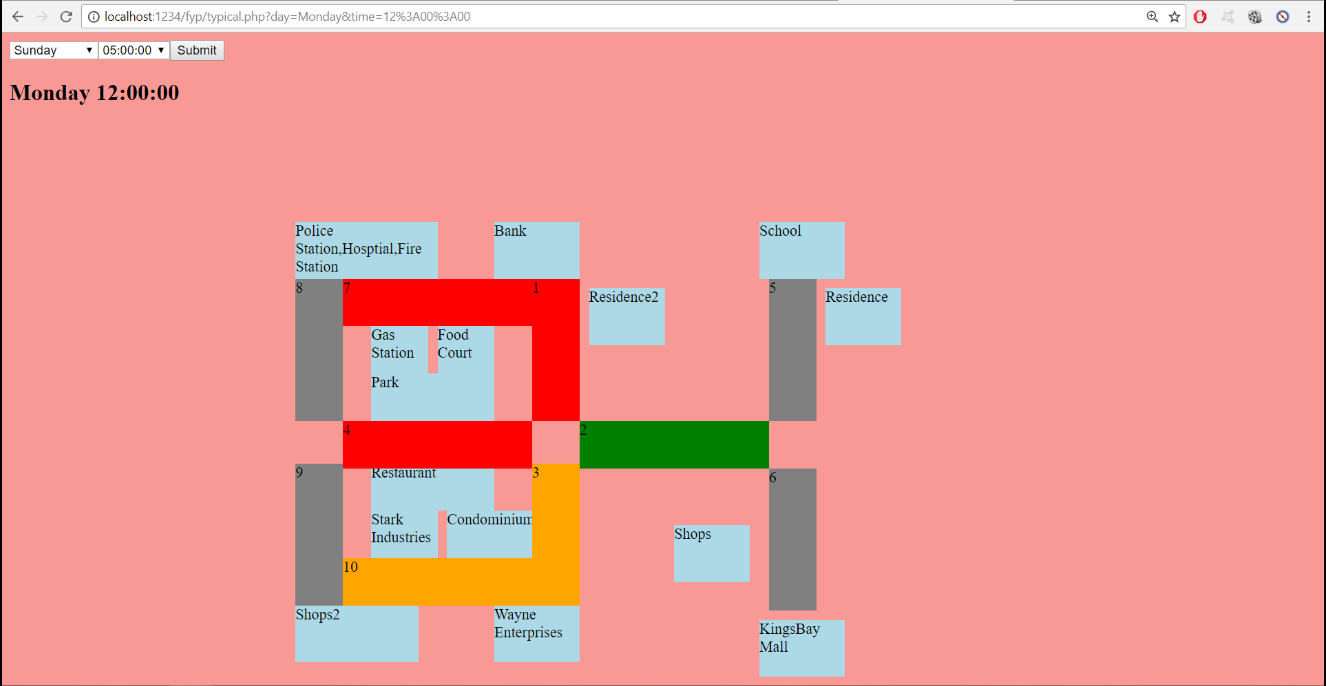
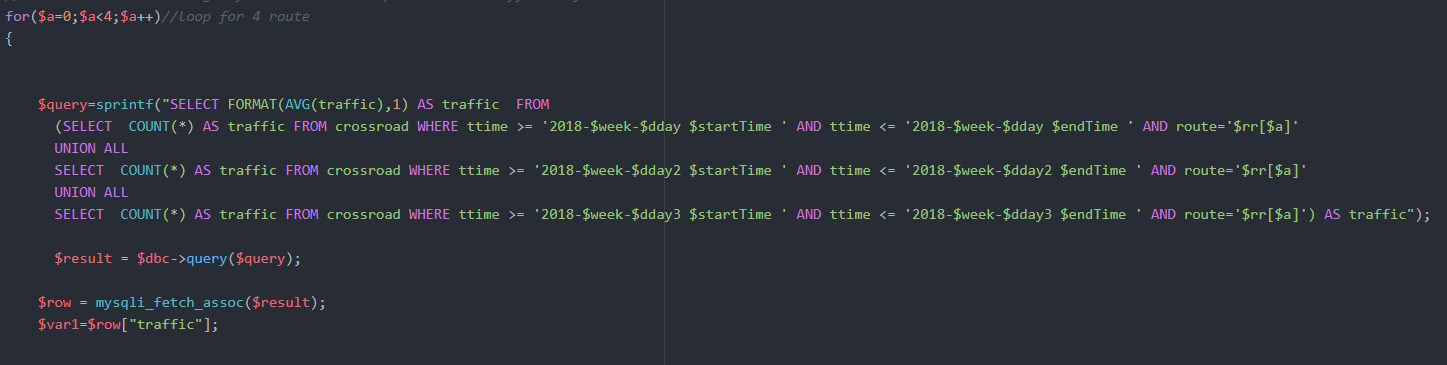
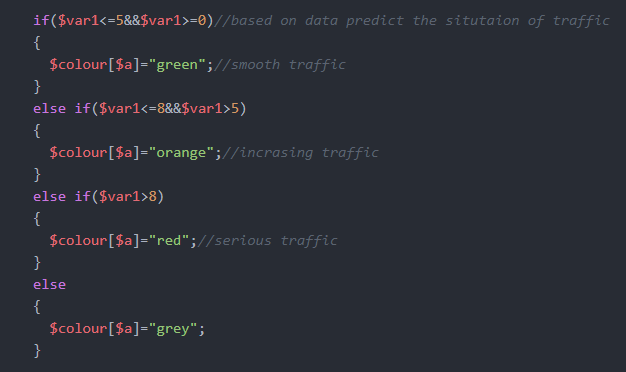


Figure 30

Below is the source code to calculate the average of 3 weeks data to predict the traffic in fourth week by selecting out the data from the database then compare.



After taking data out from the database, we then compare the number of times of the data getting received which is from the Ultrasonic Sensor (HC-sr04). Then by using if else statement, we can determine the condition of the traffic. As you can see the Figure 30 in next page, if the data <=5&&>=0 the color of the road will be green and so on.



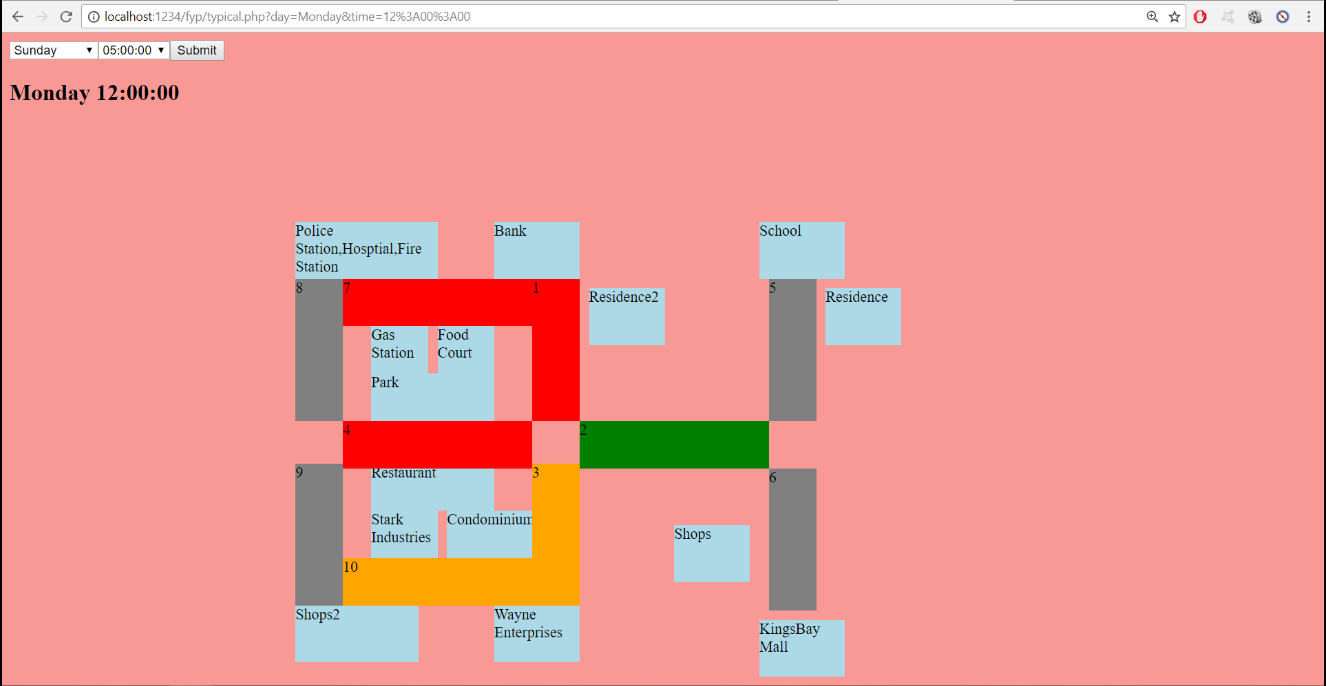


Figure 30

**\*ADDITIONAL CODES PLEASE REFER APPENDIX\***

**3.8.4 (live.php) Page**

First, connect to your database.

php

This page is where the road will be constantly change colors based on the density of the traffic. The Figure 31 shown below is the first sequence of traffic light (Lane A - Lane D).

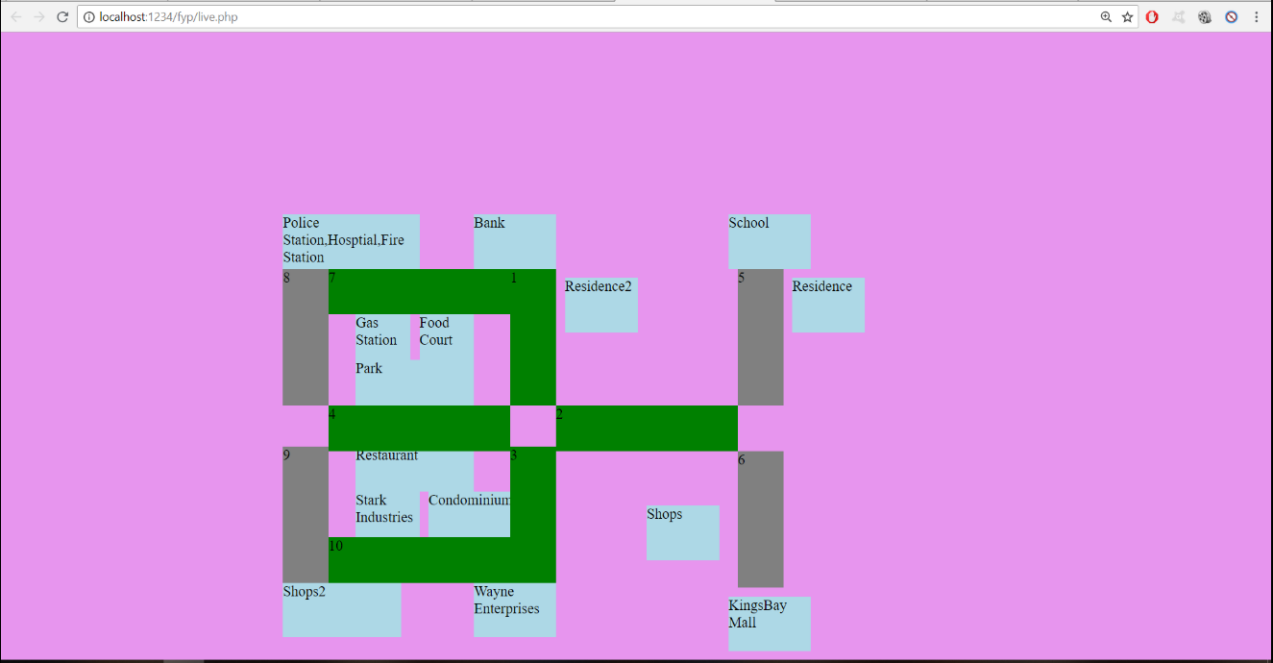
****

Figure 31: (live.php) Page

After a few cycle of the sequence of traffic light (Lane A - Lane D). As you can see the Figure 31 shown below route 1,7 (Lane C) is red, because it senses the presence of vehicles as it will then send which route has congestion example Route 2 will be “2” shown in Figure 32 to the database many times in between the cycle.

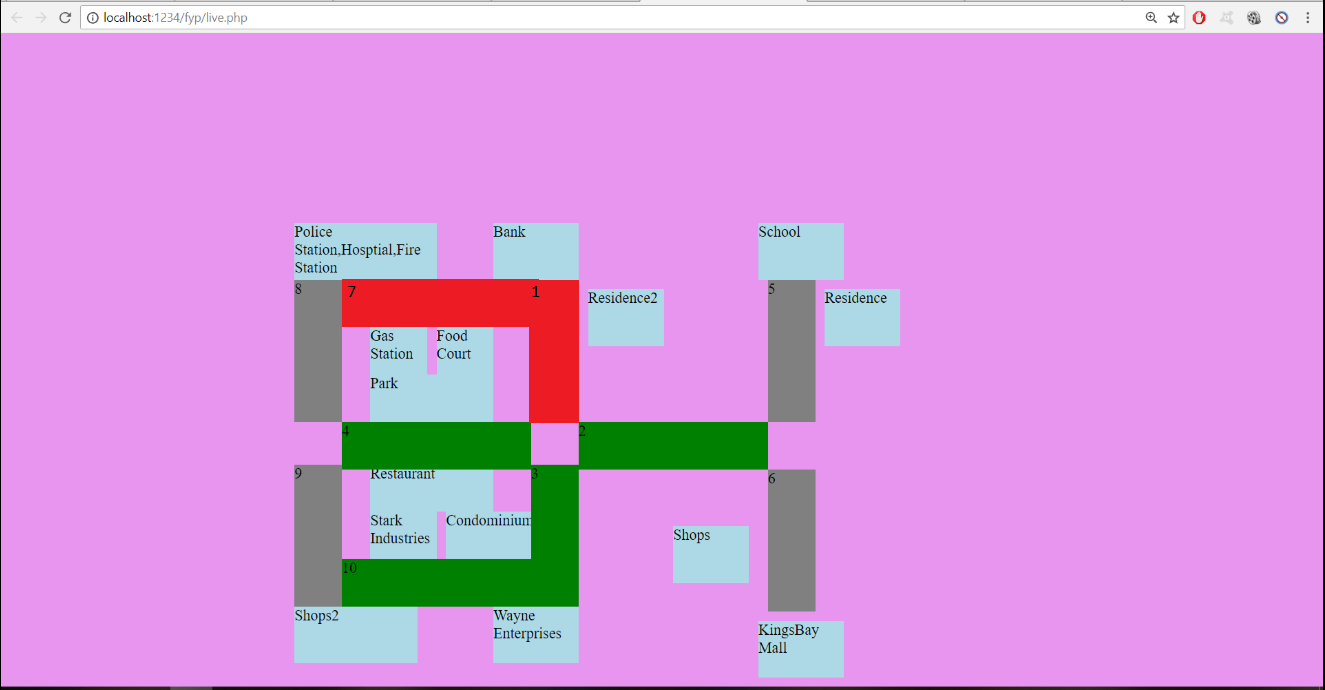


Figure 31

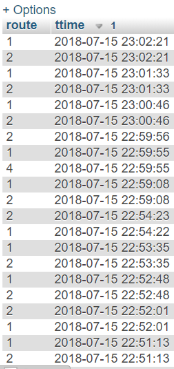
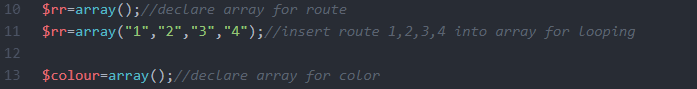
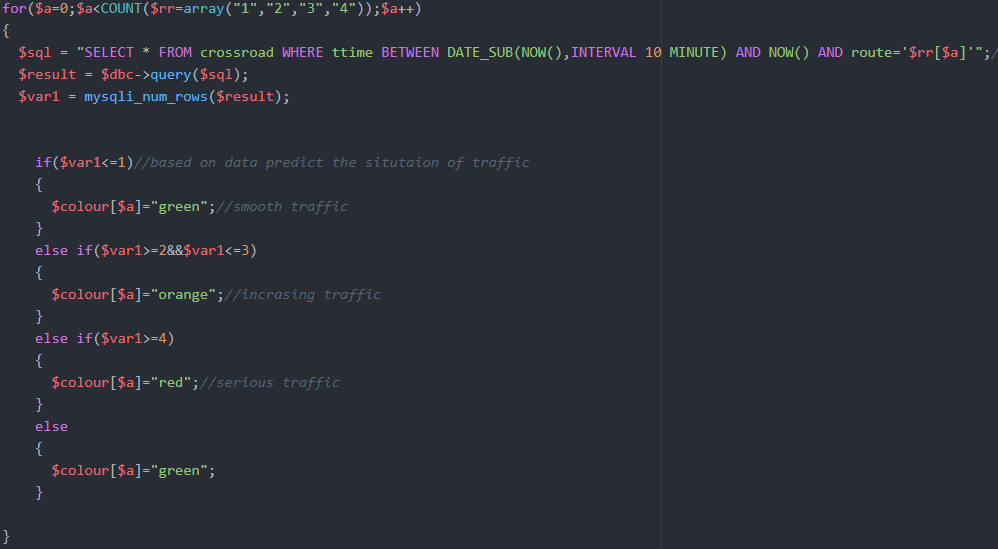


Figure 32: Congestion Database Table

The Declaration variable for array.



The code below is for determine the current state of the traffic condition for short we are able to see the traffic condition on real time. The time interval to determine is 10 minutes.

****

**\*ADDITIONAL CODES PLEASE REFER APPENDIX\***

**3.8.5 (emergency.php) Page**

First, connect to your database.

php

This page is where an incident happened, authority will then receive a location where the incident had happened. The authority will then key-in the location in our webpage shown in Figure 33, requesting for Police/Ambulance/Fire Fighter by ticking the checkbox and then enter scenario. Then by pressing submit, it will then show which is the best route to take to reach the destination in the most minimum time shown in Figure 34.

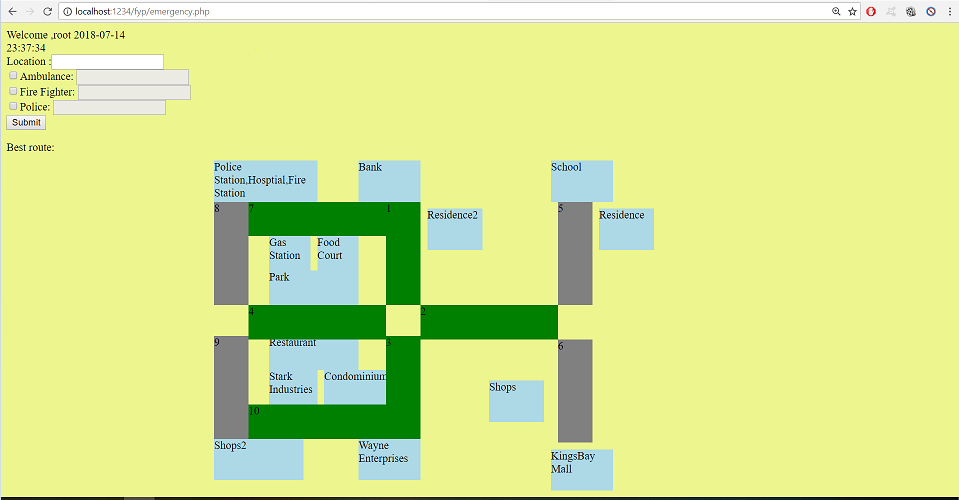
****

Figure 33: (emergency.php) Page

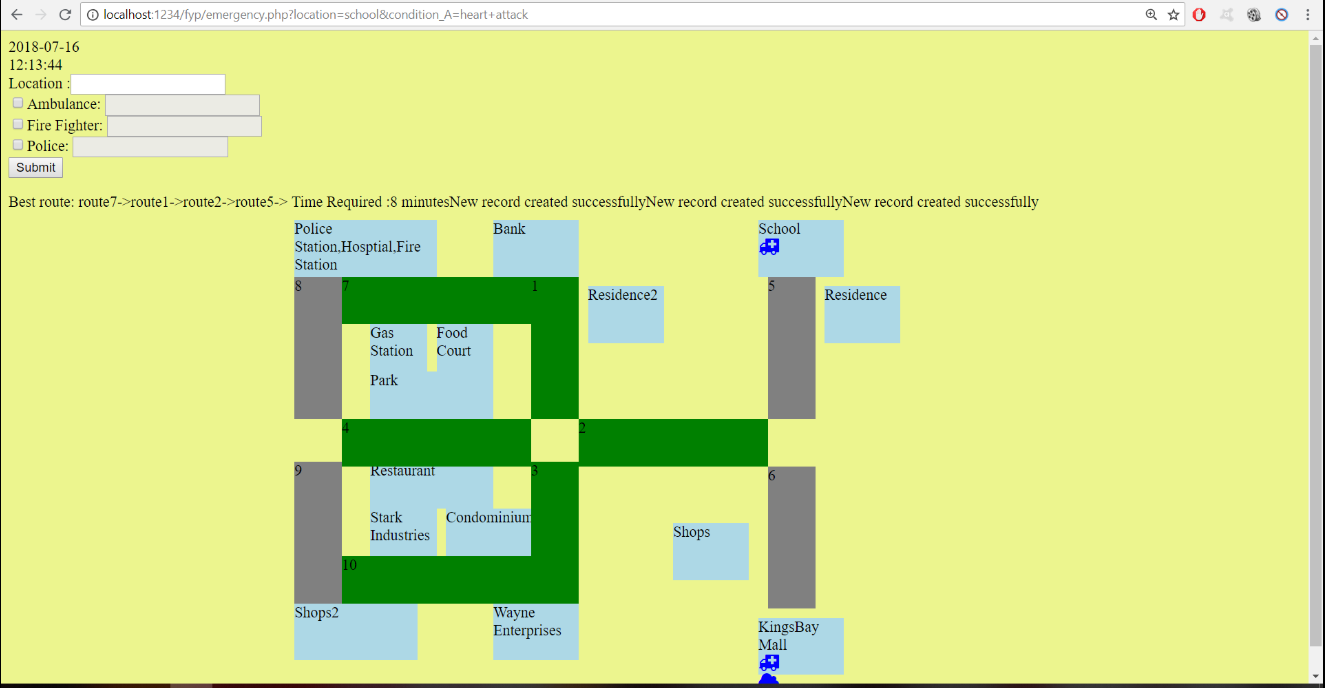
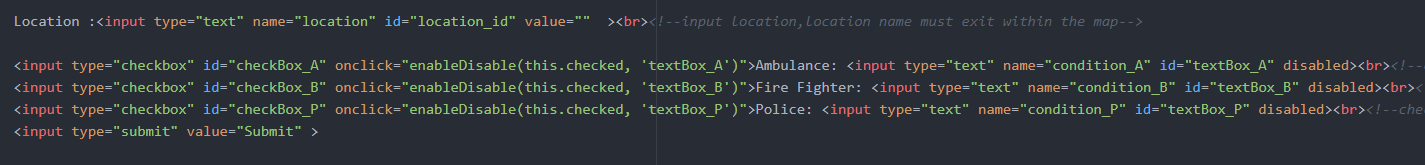


Figure 34: Best Route Calculation

The code shown below is the interface of Figure 35.

****

After input the data, press submit

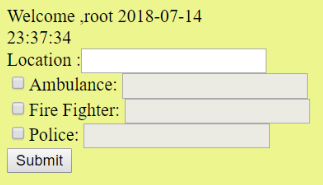
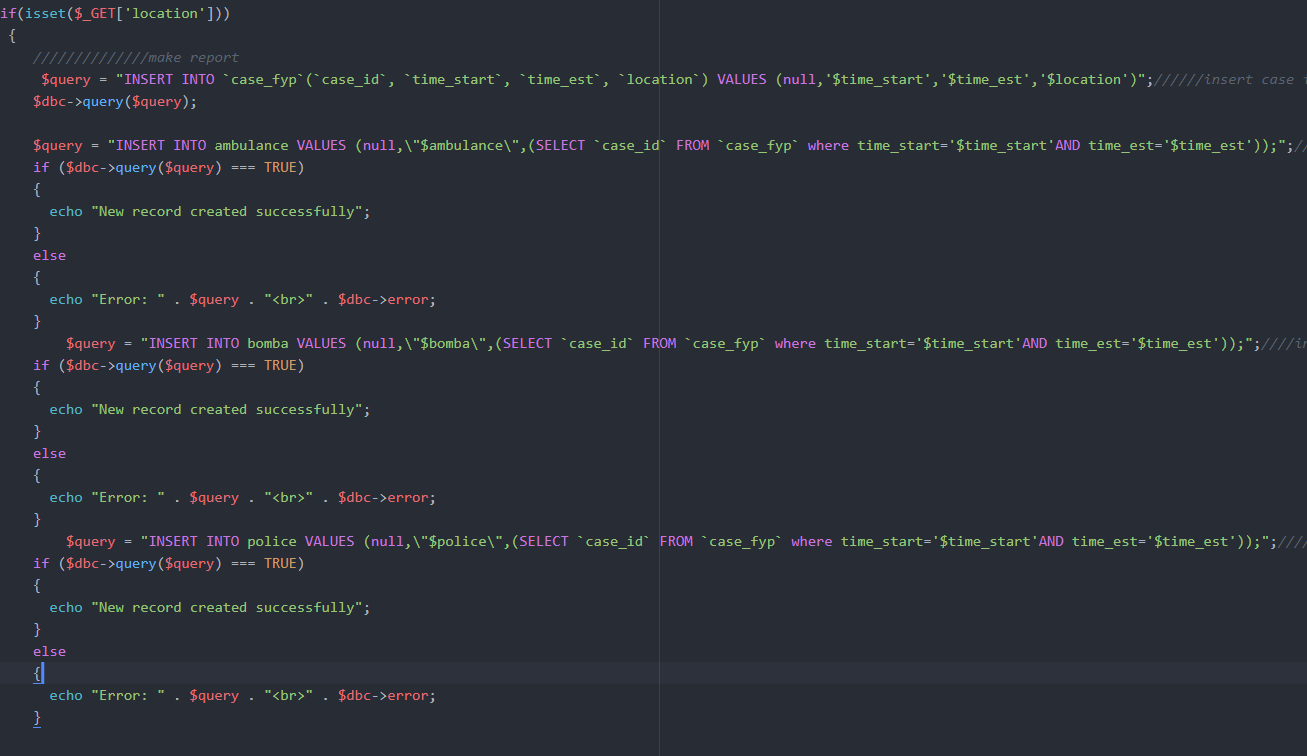


Figure 35: Input interface for emergency

After pressing submit, database is being inserted accordingly:

The source code below is to insert the data into the database that has been submitted from the (emergency.php) page.



The source code below is to show the icons shown in Figure whenever there is a scenario happening. It will only appear when a scenario is happening, will be gone when the scenario is settled.

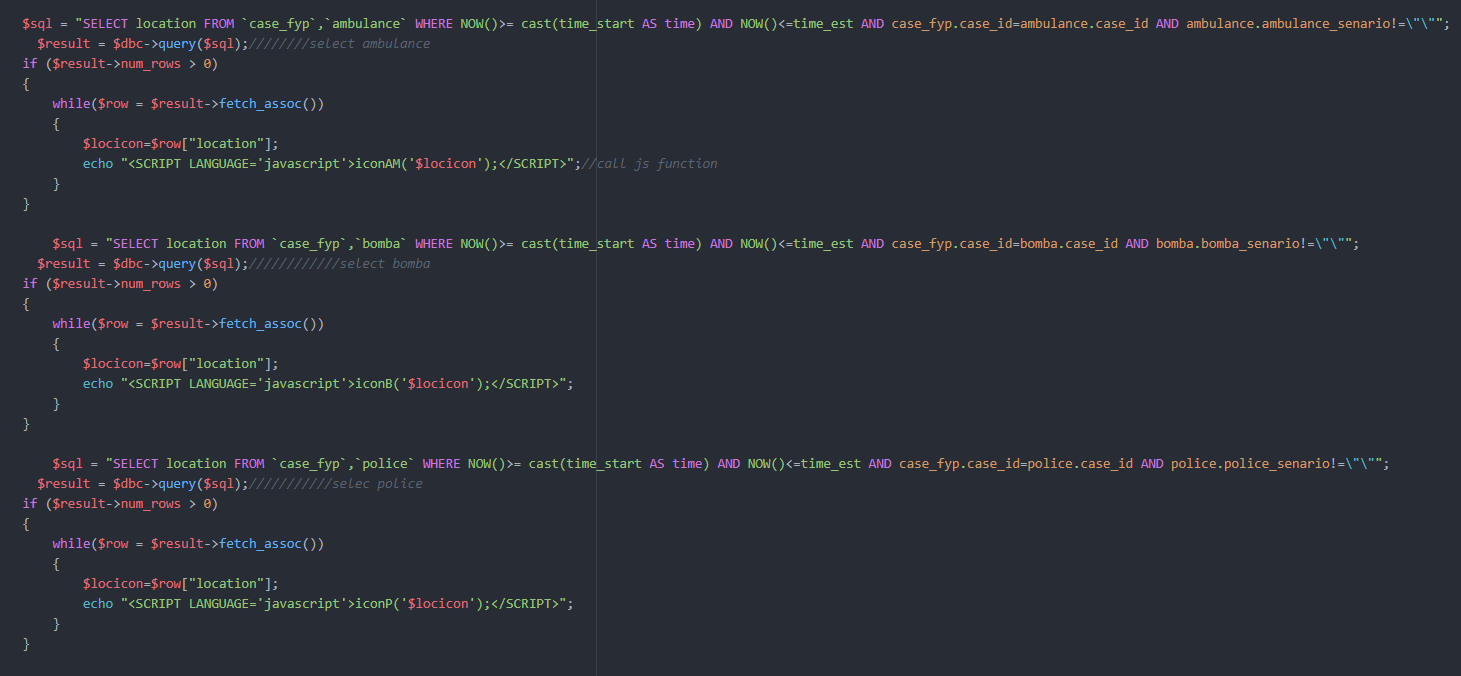
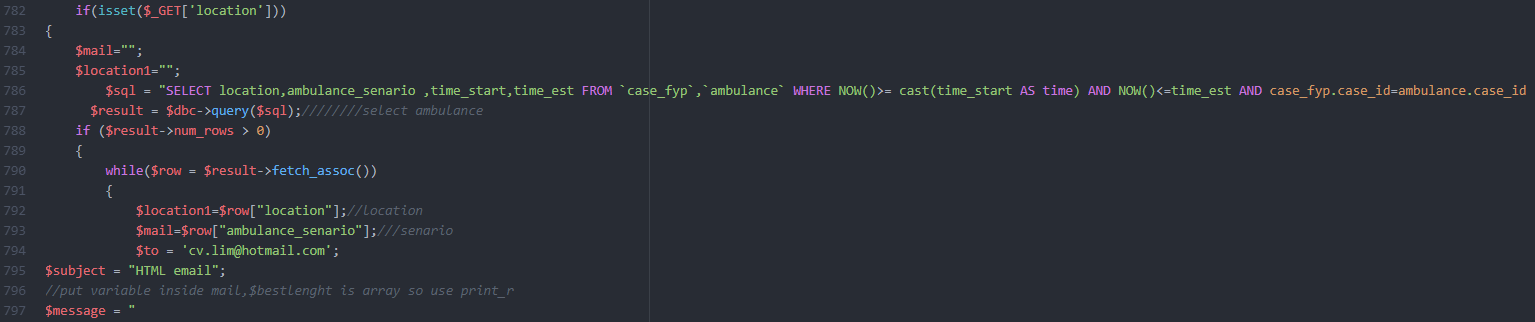


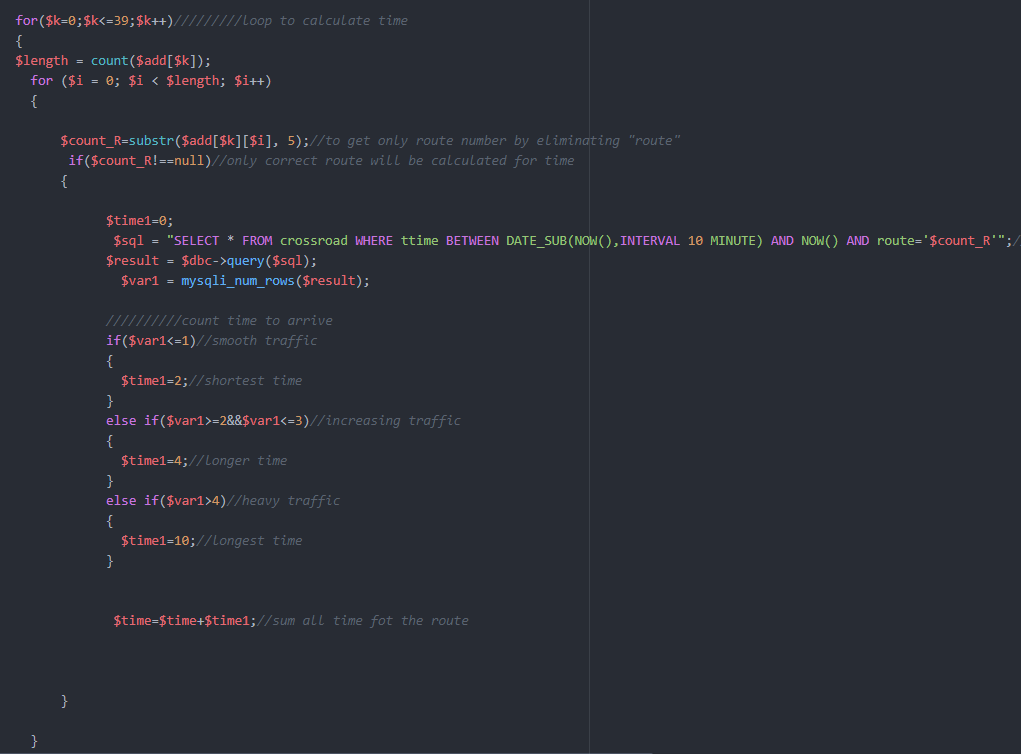


Figure 36: fa-fa icon for ambulance

The source code below is the code for email notification to the authority.



The source code below is the code for calculation the best route to take for the emergency vehicles.



**\*ADDITIONAL CODES PLEASE REFER APPENDIX\***

**3.9 Project Cost**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Components & Materials | Quantity | Price per Quantity(RM) | Total(RM) |
| 1 | Arduino Mega 2560 + USB Cable | 1 | 44.80 | 44.80 |
| 2 | Ultrasonic Sensors (HC-sr04) | 10 | 3.50 | 35.00 |
| 3 | Ultrasonic Sensors (HC-sr04) Holder | 8 | 1.20 | 9.60 |
| 4 | LED Bulbs (Green, Yellow, Red) x4 each | 1 | 1.00 | 1.00 |
| 5 | 2 Digit 7 Segment Display | 6 | 2.50 | 15.00 |
| 6 | Ethernet Shield | 1 | 30.00 | 30.00 |
| 7 | Crossover Cable | 1 | 13.00 | 13.00 |
| 8 | Lan Cable | 1 | 15.00 | 15.00 |
| 9 | Router | 1 | 80.00 | 80.00 |
| 10 | Jumper Wires | 6 | 5.00 | 30.00 |
| 11 | Soldering Board | 3 | 2.50 | 7.50 |
| 13 | Foam Board | 1 | 7.20 | 7.20 |
| 14 | Plastic Board | 1 | 8.50 | 8.50 |
| 15 | Card Board | 1 | F.O.C | F.O.C |
| 16 | Resistor (220 Ohm) x9 units | 18 | 0.04 | 0.08 |
| 17 | 40 Pin Female Header | 4 | 1.00 | 4.00 |
|  |  |  | Total (RM) | 300.68 |

Table 3: Project Cost

**CHAPTER 4: TEST, DATA COLLECTION & RESULT**

**4.1 System Integration**

This project IoT Smart Traffic with Emergency hardware is designed by the program “Fritzing” where the components and wire are place accordingly for us to prepare and build the projects. Each component is individually tested, hardware assembled, and software integrated on the hardware platform.

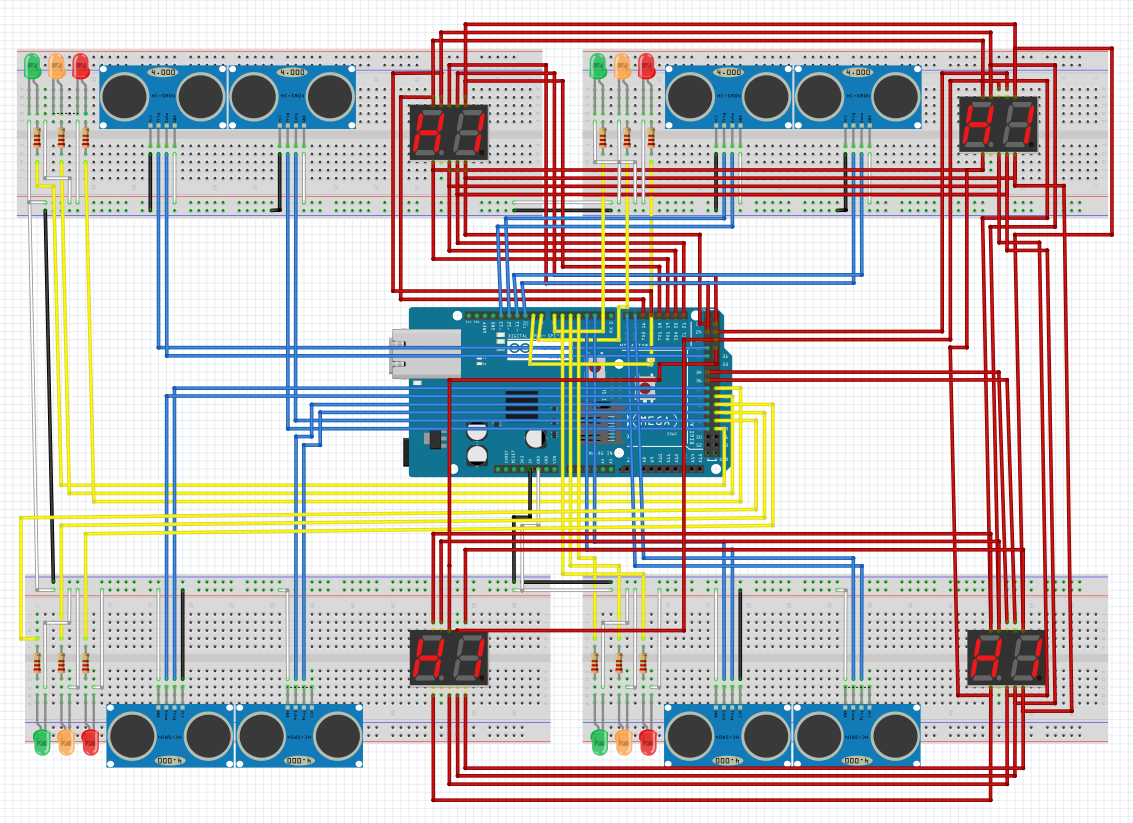


Figure 37: Schematic Design Diagram

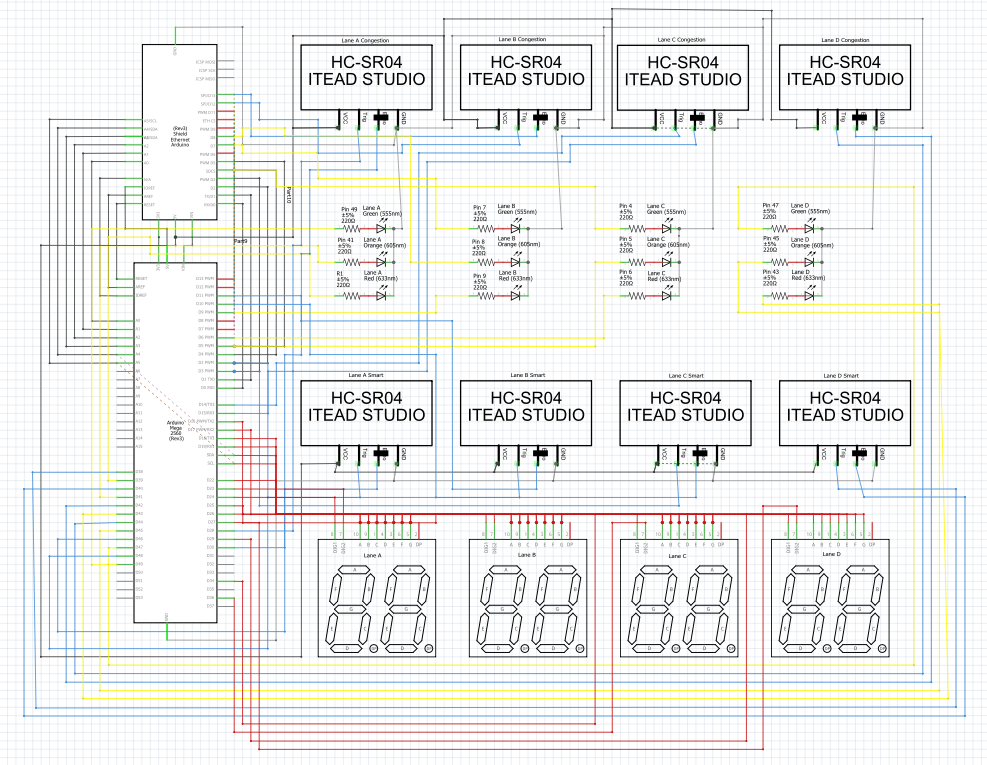


Figure 38: Schematic Diagram

* The main unit of this project IoT Smart Traffic with Emergency is Arduino Mega 2560 microcontroller.
* As you can see on the diagram refer Figure

1. Vcc 5V: Black wires
2. GND: White Wires (Figure 37) / Grey Wires (Figure 38)
3. Ultrasonic Sensor (HC-sr04) [trigPins & echoPins]: Blue Wires
4. 2-Digit 7 Segment Display: Red Wires
5. Led Bulbs (Green, Yellow, Red): Yellow Wires

* The echoPins of the Ultrasonic Sensor (HC-sr04) act as the INPUT data pin and the trigPins act as the OUTPUT data pin.
* Pins are connected accordingly shown in Table

|  |  |  |  |
| --- | --- | --- | --- |
| Lane | Pins | | |
|  | Traffic Light | Sensors[1-Smart]|[2-Congestion] | 2-Digit 7 Segment Display |
| A | 49, Green  41, Yellow  39, Red | [Sensor1] - 46, echoPin  [Sensor1] - 48, trigPin  [Sensor2] - 30, echoPin  [Sensor2] - 28, trigPin | Segment A - 16  Segment B - 17  Segment C - 19  Segment D - 21  Segment E - 20  Segment F - 18  Segment G - 22  Digit 1 - 24  Digit 2 - 23 |
| B | 7, Green  8, Yellow  9, Red | [Sensor1] - 11, echoPin  [Sensor1] - 10, trigPin  [Sensor2] - 13, echoPin  [Sensor2] - 12, trigPin | Segment A - 16  Segment B - 17  Segment C - 19  Segment D - 21  Segment E - 20  Segment F - 18  Segment G - 22  Digit 1 - 26  Digit 2 - 25 |
| C | 4, Green  5, Yellow  6, Red | [Sensor1] - 2, echoPin  [Sensor1] - 3, trigPin  [Sensor2] - 15, echoPin  [Sensor2] - 14, trigPin | Segment A - 16  Segment B - 17  Segment C - 19  Segment D - 21  Segment E - 20  Segment F - 18  Segment G - 22  Digit 1 - 34  Digit 2 - 36 |
| D | 47, Green  45, Yellow  43, Red | [Sensor1] - 40, echoPin  [Sensor1] - 38, trigPin  [Sensor2] - 42, echoPin  [Sensor2] - 44, trigPin | Segment A - 16  Segment B - 17  Segment C - 19  Segment D - 21  Segment E - 20  Segment F - 18  Segment G - 22  Digit 1 - 29  Digit 2 - 27 |

Table 4: List of Pins Connected to Arduino

* For the 2-Digit 7 Segment display, the concept to connect Segment A - Segment G is shown below Figure .

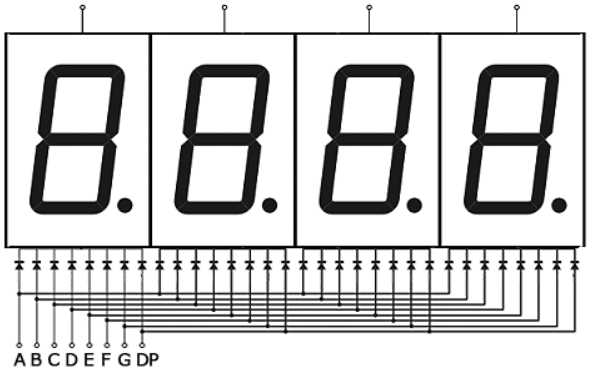


Figure 39: 7 Segment Connection

**4.2 Testing Phase**

**4.2.1 Data Collection**

**Ultrasonic Sensor (HC-sr04) Testing**

In this project, Ultrasonic Sensor (HC-sr04) plays a big role as it is use for detecting the presence of a vehicle on road and also to send data to the database. Therefore, testing the Ultrasonic Sensor (HC-sr04) is conducted to ensure that it is able to detect object in time that pass through the detect range immediately to prevent that miss anything that pass through the area. The result for this data that been collected is based on the speed of object pass through the area in how many times that make sure the sensor can sense the object and the distance it uses to sense the object. The distance and speed are tested simultaneously.

|  |  |  |
| --- | --- | --- |
| Speed(m/s) (20sec) | Distance(m) | Number of getting sense |
| 0.015 | 0.3 | 3 |
| 0.0125 | 0.25 | 7 |
| 0.01 | 0.2 | 14 |
| 0.0075 | 0.15 | 19 |
| 0.005 | 0.1 | 26 |
| 0.01 | 0.05 | 31 |

Table 5: Ultrasonic Sensor (HC-sr04) Testing

After successfully testing the Ultrasonic Sensor (HC-sr04), we then proceed to the Data Collecting testing phase. If the Ultrasonic Sensor (HC-sr04) sense which Route has a vehicle on the road, it will then send which Route that has congestion and the current time into the database. Then according to the database, you can see the prediction shown in Figure of the traffic state (Blue 1st Week, Green 2nd Week, Red 3rd Week, Yellow 4th Week [Prediction]). The graph shown in Figure is Count (the number of times send of that specific route example if its Route 1/2 it will send ‘1’,‘2’ respectively at that time based on to the database) against time.

**4.2.2 Result**

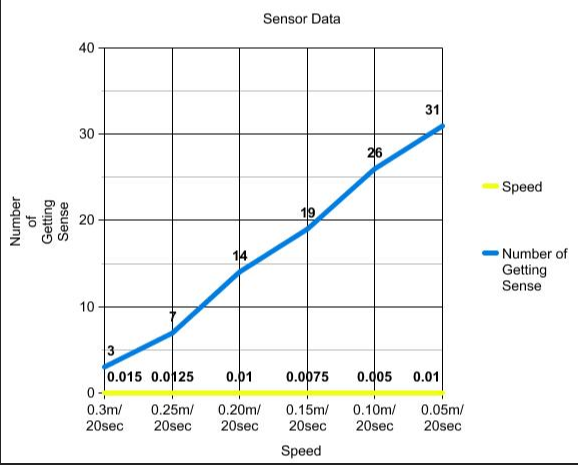


Figure 40: Ultrasonic Sensor (HC-sr04) Graph (No.of getting sense against speed)

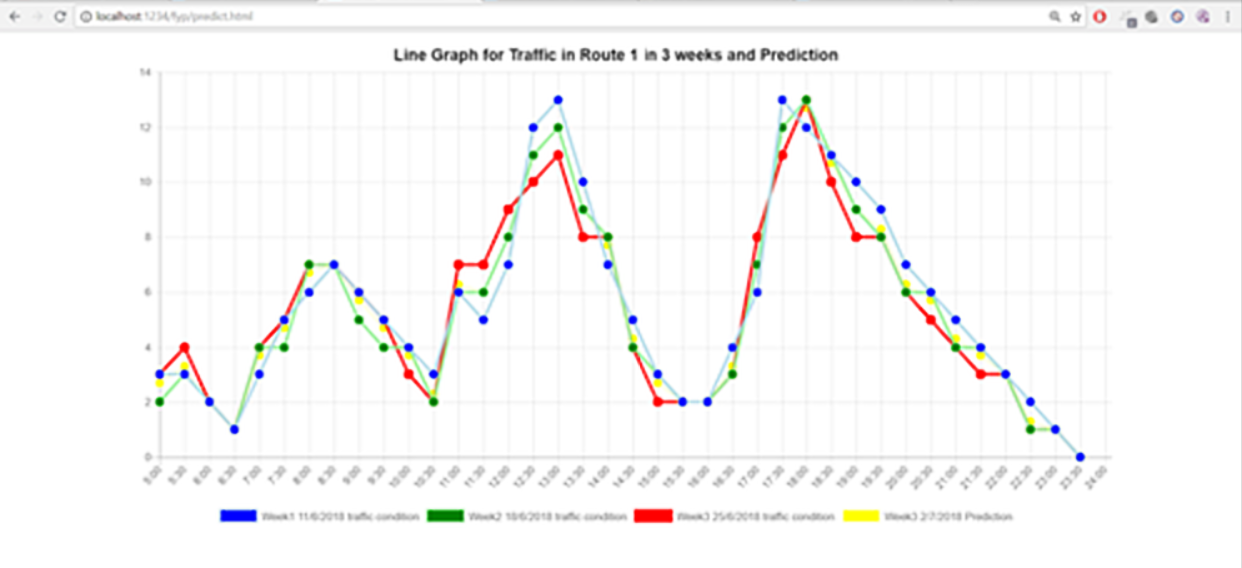
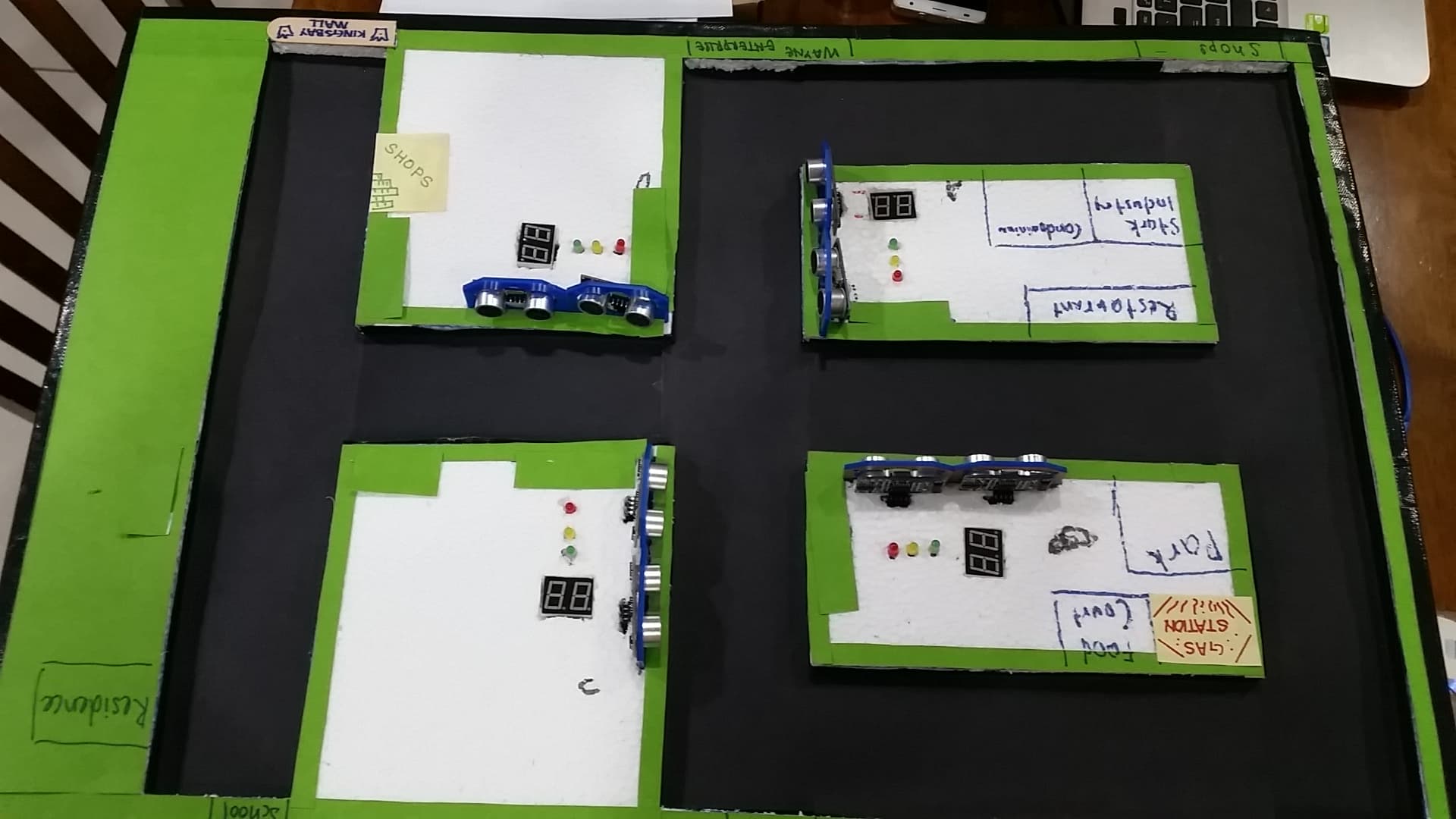


Figure 41: Traffic condition graph (Count vs Time)

**4.3 The finalized model design**

**4.3.1 Top View**



Traffic Light

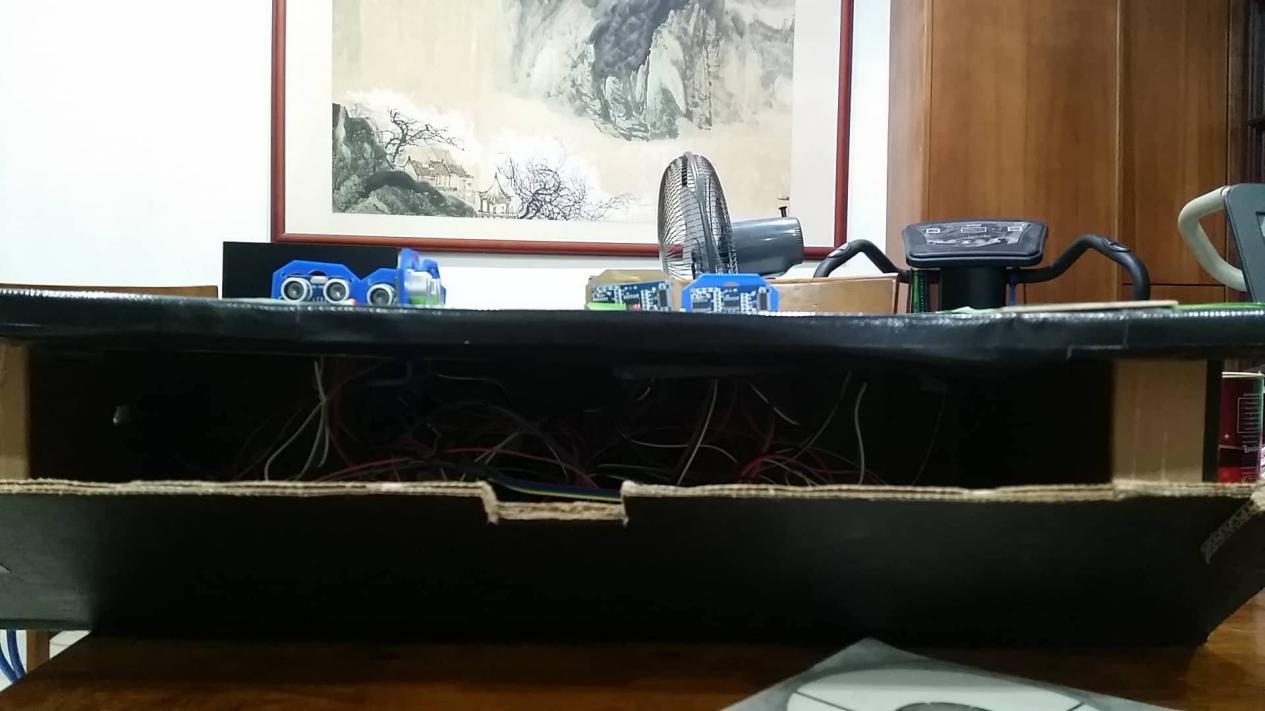
2-Digit 7 Segment Display

Sensor 2

Sensor 1

Figure 42: Top View of Finalized Model

**4.3.2 Front View**



Door

Circuit

Figure 43: Front View of Finalized Model

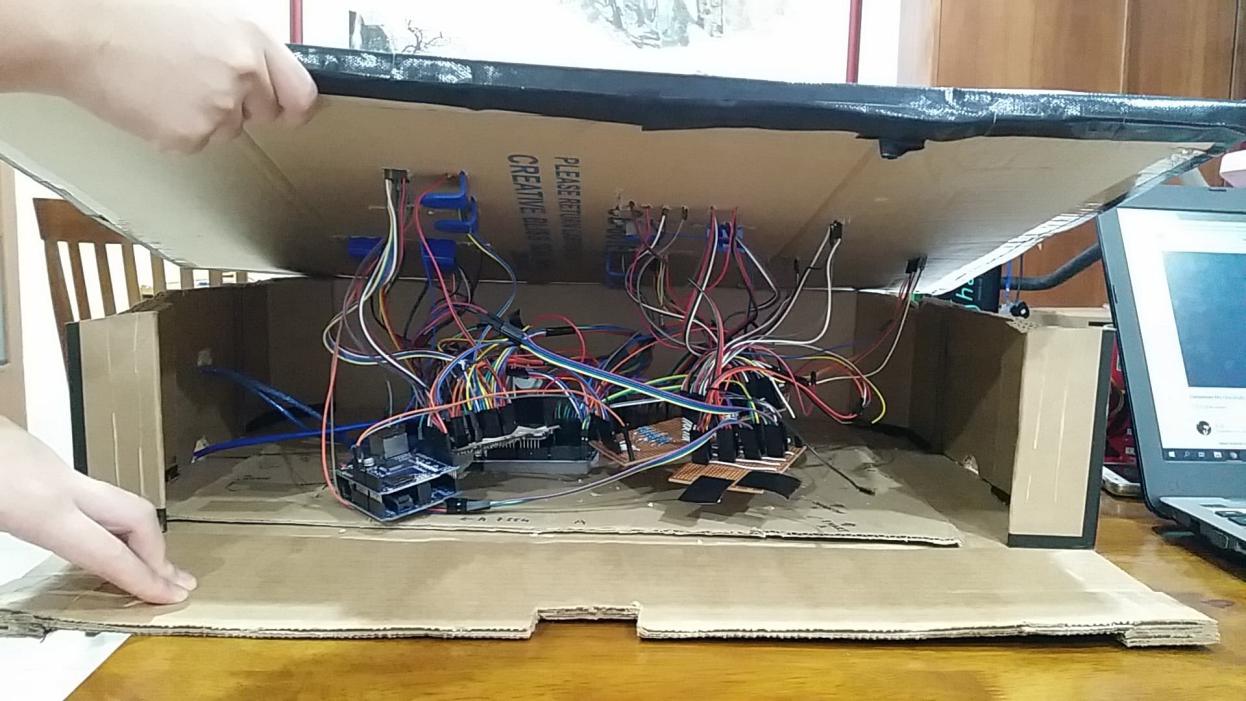
**4.3.3 Side View**



Wire coming through hole (Arduino USB Cable, Crossover Cable)

Figure 44: Side View of Finalized Model

**4.3.4 Circuit View**



Circuit

Figure 45: Circuit View of Finalized Model

**CHAPTER 5: PROBLEMS – SOLUTIONS & PROJECT LIMITATION**

**5.1 Problems & Solutions**

The first problem encountered from this project is the complications of wires shown in Figure 46. The wires are always loose, whenever there are loose wires or any pins connected wrong, it is very difficult for us to detect the errors because if we were to move the wires to find errors it would then accidentally pluck out other wires. The solution to this problem is to tidy up the wires shown in Figure 47.



Figure 46: Messier Wire

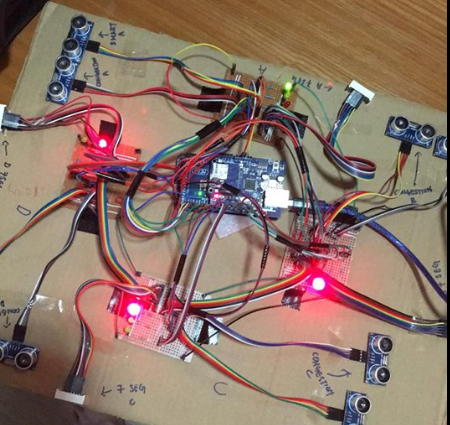


Figure 47: Tidier Wire

The second problem this project encountered will be the soldering part as in we totally had no experience in soldering. The reason why it is hard is because the connections of the Soldering Board is different from the normal Breadboard that we used. The connections are horizontal shown in Figure 48 that is different from Breadboard which the connections are vertical shown in Figure 49. The most crucial thing is that you can’t solder the top connections and the bottom connections together as a short circuit will occur if that happens. You can refer Figure 50 what is the correct method to do soldering.

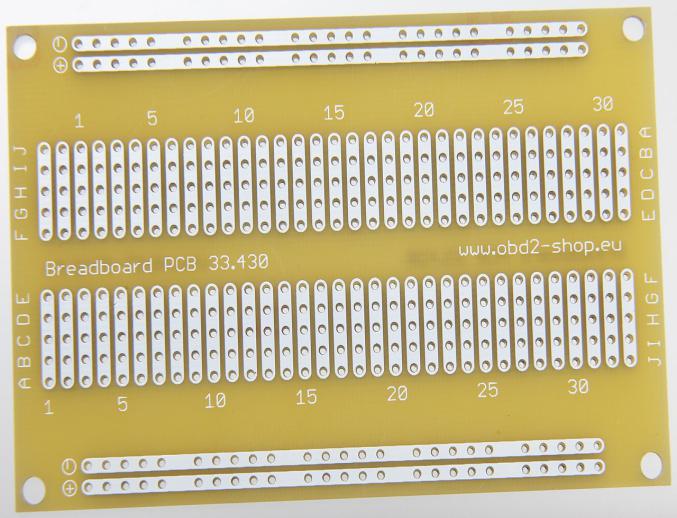
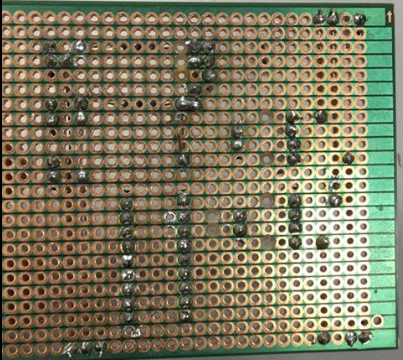


Figure 48: Solder Board Figure 49: Bread Board

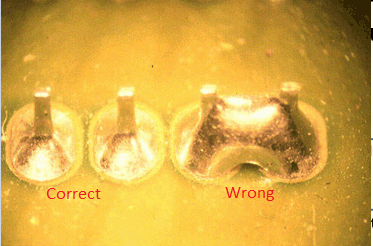


Figure 50: Soldering error

**5.2 Project Limitation**

The limitation of this project is we can only sense one side of the road and only at the intersection road which is very inefficient and inaccurate to read and analyze data as when it calculates which is the best route to take for the emergency vehicle, but it doesn’t matter where the traffic is going forward or backward. Moreover, the data will be only send when the sequence of the traffic light is finished which is from (Lane A - Lane D) as in the data received for calculations for which route to take will be inaccurate and slower, thus there will be a delay for the emergency vehicle to depart.

**CHAPTER 6: CONCLUSION & RECOMMENDATION**

**6.1 Conclusion**

This year is a fulfilling year for me because I learned so much about advanced programming which do not been taught in college.The main programming I have learned throughout the year are mysql and php(these 2 works perfectly together).If isnt this final year project ,I would not learn these programming language.

IOT includes of 3 steps :collecting data from real time situation,process the data to make good use of it and utilizes the information to contribute to the people.Mysql and php play main role in these process.I collect the data from arduino and store it inside Mysql through php web page.Then i need to select desirable data by writing correct and suitable query which is not an easy task.Writing the algorithm to instant calculate best route with the shortest time to arrive is my biggest feat until now. I glad that I could have this final year project .I also improved my basic C language while programming arduino to create web page that could control traffic light.

Everyday I have to travel from Prai to Bayan Lepas.I encounters traffic congestion everyday and wastes much precious time.Sometimes, the emergency unit such as ambulance has to jump red light for emergency cause .This is why our project is about solving traffic congestion and allow emergency unit to arrive as fast as possible.

I hope our 1 year of efforts could contribute to the people by solving the traffic problem that have been a thorn in our daily life .I am also starting to experience how to be a programmer especially a back-end programmer.Programming has unlimited potential and I cannot wait to explore more about it.For programming,nothing is impossible.

**6.2 Future Recommendations**

1. Improving the web page to be a more friendly interface web page. Enhance the web page to be more responsive for smart phone because nowadays people usually using smart phone more than pc or laptop.Develop login page so that user could customize their own traffic web page based on their needs .
2. Change the ultrasonic sensor to another type of sensor such as weight sensor that is installed inside the road.This could increase the accuracy of data collected.Install camera could be a great alternative to collect data of congestion.
3. Change the network from ethernet to wifi.This could enlarge the coverage area and collect much more data.This could be cost effective.
4. Increase the security of web page which could change traffic light.Only authorities could use that web page.

**REFERENCES**

[1] What is Internet of Things (IoT)?:

<https://tools.ietf.org/html/draft-lee-iot-problem-statement-04>

[2] Deepti Goel, Santanu Chaudhury, and Hiranmay Ghosh. 2017. An IoT Approach for Context-aware Smart Traffic Management Using Ontology. In Proceedings of WI ’17, Leipzig, Germany, August 23-26, 2017, 8 pages. DOI: 10.1145/3106426.3106499

[3] Sarjo Das, Priyankar Roychowdhury. 2016. Smart Urban Traffic ManagementSystem. DOI:10.13140/RG.2.1.3414.6324.

[4] Abida Sharif, Jianping Li, Mudassir Khalil, Rajesh Kumar, Muhammad Irfan Sharif, Atiqa Sharif. 2017. Smart Traffic System. In Proceedings of Cheng Du, China, December 15-17, 2017. DOI: 10.1109/ICCWAMTIP.2017.8301496

[5] Learning pHp/HTML/JavaScripts/CSS Codings:

<https://www.w3schools.com/pHP/default.asp>

<https://www.w3schools.com/htmL/>

<https://www.w3schools.com/jS/default.asp>

<https://www.w3schools.com/css/>

[6] Ultrasonic Sensor(HC-sr04) & LED Bulbs to make traffic light:

<http://www.instructables.com/id/Arduino-Distance-Detector-with-a-Buzzer-and-LEDs/>

1. Learning Arduino together with Ethernet Shield

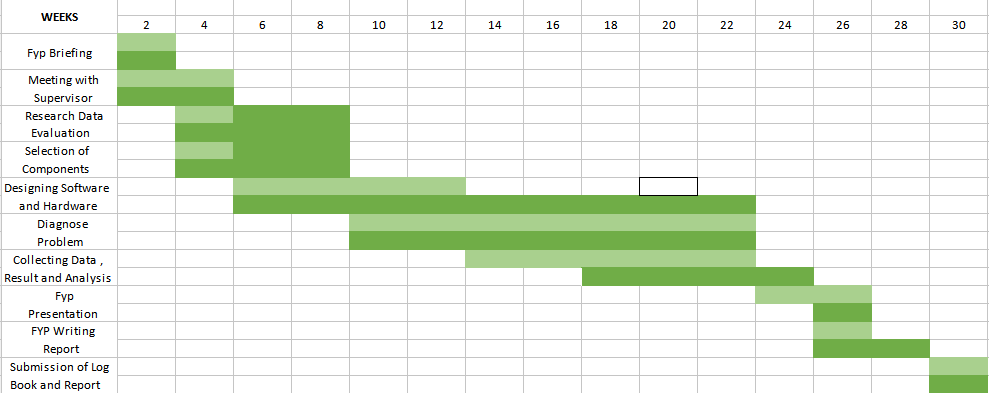
http://www.instructables.com/id/Automatic-Car-Parking-System/

1. Apply HTML Coding in Arduino to make a WebPage

<https://alselectro.wordpress.com/2016/10/30/arduino-ethernet-shield-led-onoff-from->webpage/

**APPENDIX**

**Appendix A: Gantt chart**



**Appendix B: Datasheet**

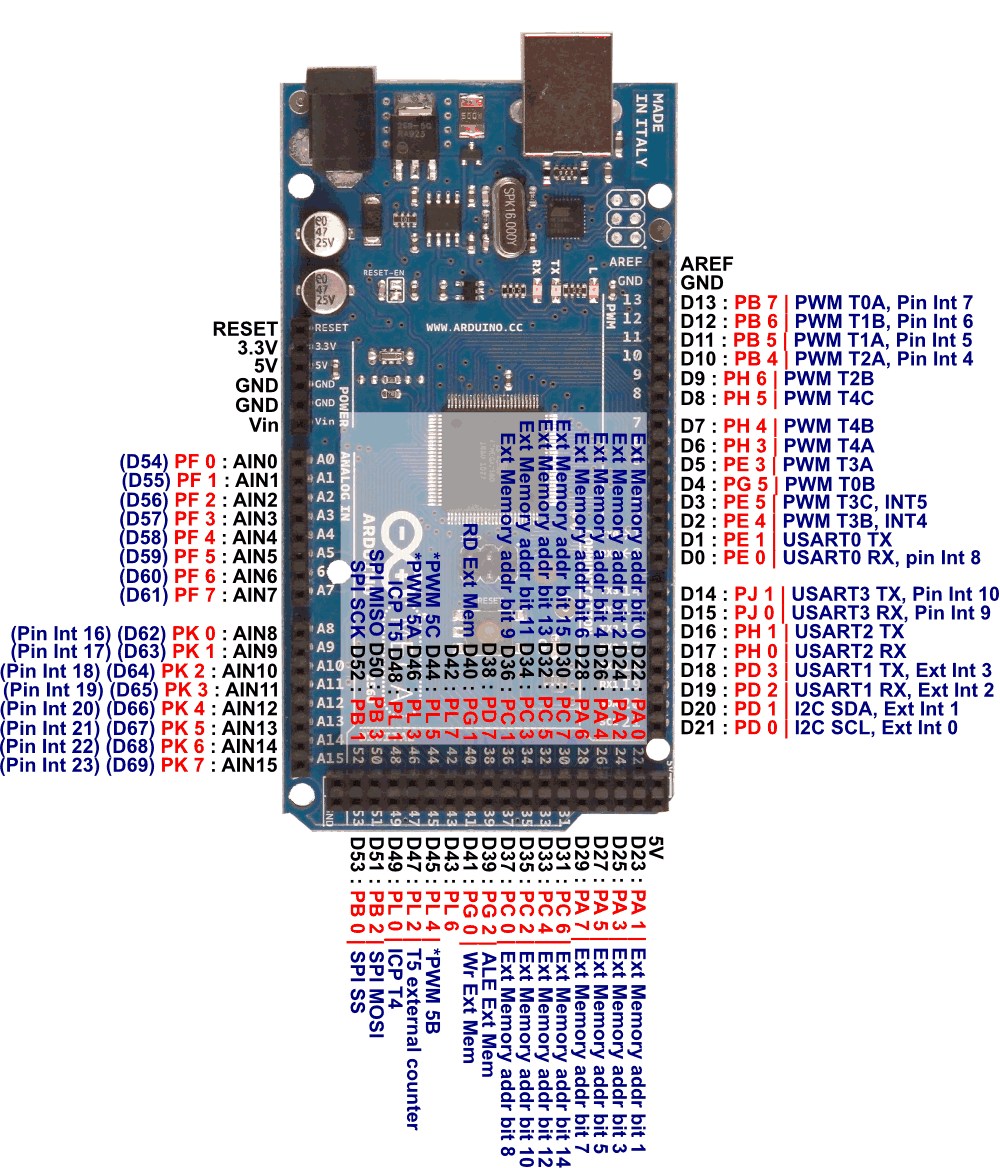


Figure 51: Arduino Mega 2560 Datasheet

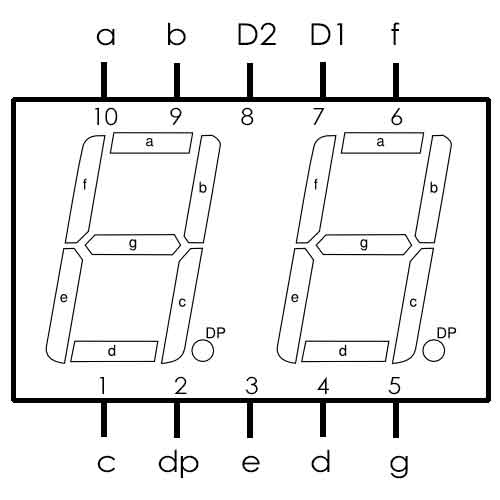


Figure 52: 2-Digit 7 Segment Display Datasheet

**Appendix C: Source Code**

**Arduino Coding [ Ultrasonic Sensor (HC-sr04), 2-Digit 7 Segment Display, Connecting with Database, Sending Data]**

#include <SPI.h>

#include <Ethernet.h>

EthernetClient client;

byte mac[]=

{

0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED

};

IPAddress arduinoIP(192, 168, 0, 6);

IPAddress dnsIP(192, 168, 0, 1);

IPAddress gatewayIP(192, 168, 0, 1);

IPAddress subnetIP(255, 255, 255, 0);

EthernetServer server(80); //server port

String readString;

////////////////////////////////////////////////////////////////////

int c=3;//for 7 seg run

int y=2;//for yellow light

int j = 0;

int PIN\_0 = 16; // A

int PIN\_1 = 17; // B

int PIN\_2 = 19; // C

int PIN\_3 = 21; // D

int PIN\_4 = 20; // E

int PIN\_5 = 18; // F

int PIN\_6 = 22; // G

int S1\_1 = 24; // ds1//one//////////////////A

int S1\_2 = 23; // ds2//ten

int S2\_1 = 26; // ds1//one/////////////////B

int S2\_2 = 25; // ds2//ten

int S3\_1 = 34; // ds1//one/////////////////C//27

int S3\_2 = 36; // ds2//ten//29

int S4\_1 = 29; // ds1//one/////////////////D//51

int S4\_2 = 27; // ds2//ten//53

int Arduino\_Pins[7] = {PIN\_0, PIN\_1, PIN\_2, PIN\_3, PIN\_4, PIN\_5, PIN\_6}; // an array of pin numbers to which LEDs

int Segment\_Pins[10][7] = {{1,1,1,1,1,1,0}, // 0

{0,1,1,0,0,0,0}, // 1

{1,1,0,1,1,0,1}, // 2

{1,1,1,1,0,0,1}, // 3

{0,1,1,0,0,1,1}, // 4

{1,0,1,1,0,1,1}, // 5

{1,0,1,1,1,1,1}, // 6

{1,1,1,0,0,0,0}, // 7

{1,1,1,1,1,1,1}, // 8

{1,1,1,1,0,1,1}, // 9

};

int digit1a, digit1b;

int digit2a, digit2b;

int digit3a, digit3b;

int digit4a, digit4b;

/////////////////////////////////////////////////////////////////

#define trigPin\_A2 28//A

#define echoPin\_A2 30

#define trigPin\_B2 12//B

#define echoPin\_B2 13

#define trigPin\_C2 14//C/////////congestion 2

#define echoPin\_C2 15

#define trigPin\_D2 44///////sensor cacat abit (D)

#define echoPin\_D2 42

#define trigPin\_A1 48//A

#define echoPin\_A1 46

#define trigPin\_B1 10//B

#define echoPin\_B1 11

#define trigPin\_C1 3//C/////////smart 1

#define echoPin\_C1 2

#define trigPin\_D1 38//D

#define echoPin\_D1 40

int greenPin\_A=49;//Route1

int yellowPin\_A=41;//change 51-41

int redPin\_A=39;//change 53-39

int greenPin\_B=7;//Route2

int yellowPin\_B=8;

int redPin\_B=9;

int greenPin\_C=4;//Route3

int yellowPin\_C=5;

int redPin\_C=6;

int greenPin\_D=47;//Route4

int yellowPin\_D=45;

int redPin\_D=43;

int time=250;

String z="";

int aa,bb,cc,dd;

void setup()

{

Serial.begin (9600);

Ethernet.begin(mac, arduinoIP, dnsIP, gatewayIP, subnetIP);

server.begin();

pinMode(trigPin\_A2 , OUTPUT);//A2 congestion

pinMode(echoPin\_A2 , INPUT);

pinMode(trigPin\_A1 , OUTPUT);//A1 smart

pinMode(echoPin\_A1 , INPUT);

pinMode(greenPin\_A, OUTPUT);

pinMode(yellowPin\_A, OUTPUT);

pinMode(redPin\_A, OUTPUT);

///////////////////////////////////////////////////

pinMode(trigPin\_B2, OUTPUT);//B2 congestion

pinMode(echoPin\_B2, INPUT);

pinMode(trigPin\_B1, OUTPUT);//B1 smart

pinMode(echoPin\_B1, INPUT);

pinMode(greenPin\_B, OUTPUT);

pinMode(yellowPin\_B, OUTPUT);

pinMode(redPin\_B, OUTPUT);

///////////////////////////////////////////////////

pinMode(trigPin\_C2, OUTPUT);//C2 congestion

pinMode(echoPin\_C2, INPUT);

pinMode(trigPin\_C1, OUTPUT);//C1 smart

pinMode(echoPin\_C1, INPUT);

pinMode(greenPin\_C, OUTPUT);

pinMode(yellowPin\_C, OUTPUT);

pinMode(redPin\_C, OUTPUT);

//////////////////////////////////////////////////

pinMode(trigPin\_D2, OUTPUT);//D2 congestion

pinMode(echoPin\_D2, INPUT);

pinMode(trigPin\_D1, OUTPUT);//D1 smart

pinMode(echoPin\_D1, INPUT);

pinMode(greenPin\_D, OUTPUT);

pinMode(yellowPin\_D, OUTPUT);

pinMode(redPin\_D, OUTPUT);

/////////////////////////////////////////////////////////////7seg

for (int i = 16; i <= 29; i++)

{

pinMode(i, OUTPUT);

}

pinMode(34, OUTPUT);

pinMode(36, OUTPUT);

}

void loop()

{

long duration\_A, distance\_A,duration\_B, distance\_B,duration\_C, distance\_C,duration\_D, distance\_D;

digitalWrite(trigPin\_A2, LOW); ////////////////////////////////A2

delayMicroseconds(2);

digitalWrite(trigPin\_A2, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin\_A2, LOW);

duration\_A = pulseIn(echoPin\_A2, HIGH);

distance\_A = (duration\_A/2) / 29.1;

digitalWrite(trigPin\_B2, LOW); ///////////////////////////////B2

delayMicroseconds(2);

digitalWrite(trigPin\_B2, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin\_B2, LOW);

duration\_B = pulseIn(echoPin\_B2, HIGH);

distance\_B = (duration\_B/2) / 29.1;

digitalWrite(trigPin\_C2, LOW); ///////////////////////////////C2

delayMicroseconds(2);

digitalWrite(trigPin\_C2, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin\_C2, LOW);

duration\_C = pulseIn(echoPin\_C2, HIGH);

distance\_C = (duration\_C/2) / 29.1;

digitalWrite(trigPin\_D2, LOW); /////////////////////////////////////D2

delayMicroseconds(2);

digitalWrite(trigPin\_D2, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin\_D2, LOW);

duration\_D = pulseIn(echoPin\_D2, HIGH);

distance\_D = (duration\_D/2) / 29.1;

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

long duration\_A1, distance\_A1,duration\_B1, distance\_B1,duration\_C1, distance\_C1,duration\_D1, distance\_D1;////////////////smart=1

digitalWrite(trigPin\_A1, LOW); ////////////////////////////////A2

delayMicroseconds(2);

digitalWrite(trigPin\_A1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin\_A1, LOW);

duration\_A1 = pulseIn(echoPin\_A1, HIGH);

distance\_A1 = (duration\_A1/2) / 29.1;

digitalWrite(trigPin\_B1, LOW); ///////////////////////////////B2

delayMicroseconds(2);

digitalWrite(trigPin\_B1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin\_B1, LOW);

duration\_B1 = pulseIn(echoPin\_B1, HIGH);

distance\_B1 = (duration\_B1/2) / 29.1;

digitalWrite(trigPin\_C1, LOW); ///////////////////////////////C2

delayMicroseconds(2);

digitalWrite(trigPin\_C1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin\_C1, LOW);

duration\_C1 = pulseIn(echoPin\_C1, HIGH);

distance\_C1 = (duration\_C1/2) / 29.1;

digitalWrite(trigPin\_D1, LOW); /////////////////////////////////////D2

delayMicroseconds(2);

digitalWrite(trigPin\_D1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin\_D1, LOW);

duration\_D1 = pulseIn(echoPin\_D1, HIGH);

distance\_D1 = (duration\_D1/2) / 29.1;

z="";

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

if (distance\_A <= 3) //////////////////////////////

{

z="3L";

sendaway(z);

z="";

aa=6;

}

else

{

aa=4;

}

if (distance\_B <= 3) //////////////////////////////

{

z="2L";

sendaway(z);

z="";

bb=6;

}

else

{

bb=4;

}

if (distance\_C <= 3) //////////////////////////////

{

z="4L";

sendaway(z);

z="";

cc=6;

}

else

{

cc=4;

}

if (distance\_D <= 3) //////////////////////////////

{

z="3L";

sendaway(z);

z="";

dd=6;

}

else

{

dd=4;

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

setLights\_A(0,0,1);

setLights\_B(1,0,0);

setLights\_C(1,0,0);//1

setLights\_D(1,0,0);

Serial.print("A\n");

//setDelay(aa);

seven\_seg(aa,aa,aa+y,aa+bb+2\*y,aa+bb+cc+3\*y);

{

setLights\_A(0,1,0);

setLights\_B(1,0,0);

setLights\_C(1,0,0);//2yellow

setLights\_D(1,0,0);

//setDelay\_2();

seven\_seg(2,y,y,bb+2\*y,bb+cc+3\*y);

}

{

setLights\_A(1,0,0);

setLights\_B(0,0,1);//3

setLights\_C(1,0,0);

setLights\_D(1,0,0);

Serial.print("B\n");

//setDelay(bb);

seven\_seg(bb,bb+cc+dd+3\*y,bb,bb+y,bb+cc+2\*y);

}

{

setLights\_A(1,0,0);

setLights\_B(0,1,0);

setLights\_C(1,0,0);//4

setLights\_D(1,0,0);

//setDelay\_2();

seven\_seg(2,cc+dd+3\*y,y,y,cc+2\*y);

}

{

setLights\_A(1,0,0);

setLights\_B(1,0,0);//5

setLights\_C(0,0,1);

setLights\_D(1,0,0);

Serial.print("C\n");

//setDelay(cc);

seven\_seg(cc,cc+dd+2\*y,aa+cc+dd+3\*y,cc,cc+y);

}

{

setLights\_A(1,0,0);

setLights\_B(1,0,0);

setLights\_C(0,1,0);//6

setLights\_D(1,0,0);

//setDelay\_2();

seven\_seg(2,dd+2\*y,aa+dd+3\*y,y,y);

}

{

setLights\_A(1,0,0);

setLights\_B(1,0,0);//7

setLights\_C(1,0,0);

setLights\_D(0,0,1);

Serial.print("D\n");

//setDelay(dd);

seven\_seg(dd,dd+y,aa+dd+2\*y,aa+bb+dd+3\*y,dd);

}

{

setLights\_A(1,0,0);

setLights\_B(1,0,0);//8

setLights\_C(1,0,0);

setLights\_D(0,1,0);

//setDelay\_2();

seven\_seg(2,y,aa+2\*y,aa+bb+3\*y,y);

}

aa=0;bb=0;cc=0;dd=0;

}//last

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void setLights\_A(int red, int yellow, int green)

{

digitalWrite(redPin\_A, red);

digitalWrite(yellowPin\_A, yellow);

digitalWrite(greenPin\_A, green);

}

void setLights\_B(int red, int yellow, int green)

{

digitalWrite(redPin\_B, red);

digitalWrite(yellowPin\_B, yellow);

digitalWrite(greenPin\_B, green);

}

void setLights\_C(int red, int yellow, int green)

{

digitalWrite(redPin\_C, red);

digitalWrite(yellowPin\_C, yellow);

digitalWrite(greenPin\_C, green);

}

void setLights\_D(int red, int yellow, int green)

{

digitalWrite(redPin\_D, red);

digitalWrite(yellowPin\_D, yellow);

digitalWrite(greenPin\_D, green);

}

void setDelay(int p)

{

for(int s=p;s>=1;s--)

{

Serial.println(s);

control();

delay(1000);

}

}

void setDelay\_2()

{

for(int s=1;s<=2;s++)

{

Serial.println(s);

control();

delay(1000);

}

}

void blinkyellow()

{

for(int x=0;x<4;x++)

{

setLights\_A(0,1,0);

setLights\_B(0,1,0);

setLights\_C(0,1,0);

setLights\_D(0,1,0);

delay(700);

setLights\_A(0,0,0);

setLights\_B(0,0,0);

setLights\_C(0,0,0);

setLights\_D(0,0,0);

delay(700);

}

}

void control()

{

a:

{

// Create a client connection

EthernetClient client = server.available();

if (client){

while (client.connected()) {

if (client.available()) {

char c = client.read();

//read char by char HTTP request

if (readString.length() < 100) {

//store characters to string

readString += c;

}

//if HTTP request has ended– 0x0D is Carriage Return \n ASCII

if (c == 0x0D) {

client.println("HTTP/1.1 200 OK"); //send new page

client.println("Content-Type: text/html");

client.println();

client.print ("<body style=background-color:#9bbad6>");

client.println("<HTML>");

client.println("<HEAD>");

client.println("<TITLE> ARDUINO ETHERNET SHIELD</TITLE>");

client.println("</HEAD>");

client.println("<script language=\"javascript\">");

client.println("if(document.referrer=="")>");

client.println("{");

client.println("window.location.href = \"http://192.168.0.5:1234/fyp/loginfyp.php\";");

client.println("}");

client.println("</script>");

client.println("<BODY>");

client.println("<hr>");

client.println("<hr>");

client.println("<br>");

client.println("<H1 style=\"color:red;\">Emergency control traffic light from webpage</H1>");

client.println("<hr>");

client.println("<br>");

client.println("<H2><a href=\"/?ROUTE1ON\"\">Turn On Route 1</a><br></H2>"); ///////////using route1

client.println("<H2><a href=\"/?ROUTE2ON\"\">Turn On Route 2</a><br></H2>"); ////////////using route 2

client.println("<H2><a href=\"/?ROUTE3ON\"\">Turn On Route 3</a><br></H2>"); ///////////////using route 3

client.println("<H2><a href=\"/?ROUTE4ON\"\">Turn On Route 4</a><br></H2>"); /////////////using route 4

client.println("<H2><a href=\"/?ROUTEOFF\"\">Return</a><br></H2>");

client.println("</BODY>");

client.println("</HTML>");

delay(10);

//stopping client

client.stop();

if(readString.indexOf("?ROUTE1ON") > -1)

{

blinkyellow();

digitalWrite(redPin\_A, HIGH);

digitalWrite(redPin\_B, HIGH);

digitalWrite(greenPin\_C, HIGH);

digitalWrite(redPin\_D, HIGH);

readString="";

}

else if(readString.indexOf("?ROUTE2ON") > -1)

{

blinkyellow();

digitalWrite(redPin\_A, HIGH);

digitalWrite(redPin\_B, HIGH);

digitalWrite(redPin\_C, HIGH);

digitalWrite(greenPin\_D, HIGH);

readString="";

}

else if(readString.indexOf("?ROUTE3ON") > -1)

{

blinkyellow();

digitalWrite(greenPin\_A, HIGH);

digitalWrite(redPin\_B, HIGH);

digitalWrite(redPin\_C, HIGH);

digitalWrite(redPin\_D, HIGH);

readString="";

}

else if(readString.indexOf("?ROUTE4ON") > -1)

{

blinkyellow();

digitalWrite(redPin\_A, HIGH);

digitalWrite(greenPin\_B, HIGH);

digitalWrite(redPin\_C, HIGH);

digitalWrite(redPin\_D, HIGH);

readString="";

}

}

}

}

}

}

while(readString.indexOf("?ROUTEOFF") >-1)

{

return;

}

readString="";

goto a;

}

////////////////////////////////////////////////////////send to php

void sendaway(String o)

{

if (client.connect("192.168.0.5", 1234))

{

Serial.println("connected");

client.print("GET /fyp/receivecross.php?routeA=");

client.print(o );

client.println("HTTP/1.1");

client.println("Host: 192.168.0.5");

client.println("Connection: close");

client.println();

client.stop();

}

else

{

Serial.println("connection failed");

}

}

/////////////////////////////////////////////////////////////////////7seg

int seven\_seg(int countdown,int countA,int countB,int countC,int countD)

{

for (int j = countdown; j >= 1; j--)

{

control();

digit1b = countA / 10;

digit1a = countA % 10;

digit2b = countB / 10;

digit2a = countB % 10;

digit3b = countC / 10;

digit3a = countC % 10;

digit4b = countD / 10;

digit4a = countD % 10;

for (int TIME=0;TIME<=100;TIME++)

{

s1d1();//one

s1d2();//ten

s2d1();

s2d2();

s3d1();

s3d2();

s4d1();

s4d2();

}

countA--;

countB--;

countC--;

countD--;

}

}

void s1d1()

{

digitalWrite(S1\_1,0);///////0=high

digitalWrite(S1\_2,1);

digitalWrite(S2\_1,1);

digitalWrite(S2\_2,1);

digitalWrite(S3\_1,1);

digitalWrite(S3\_2,1);

digitalWrite(S4\_1,1);

digitalWrite(S4\_2,1);

dis(digit1a);

delay(c);

}

void s1d2()

{

digitalWrite(S1\_1,1);///////0=high

digitalWrite(S1\_2,0);

digitalWrite(S2\_1,1);

digitalWrite(S2\_2,1);

digitalWrite(S3\_1,1);

digitalWrite(S3\_2,1);

digitalWrite(S4\_1,1);

digitalWrite(S4\_2,1);

dis(digit1b);

delay(c);

}

void s2d1()

{

digitalWrite(S1\_1,1);///////0=high

digitalWrite(S1\_2,1);

digitalWrite(S2\_1,0);

digitalWrite(S2\_2,1);

digitalWrite(S3\_1,1);

digitalWrite(S3\_2,1);

digitalWrite(S4\_1,1);

digitalWrite(S4\_2,1);

dis(digit2a);

delay(c);

}

void s2d2()

{

digitalWrite(S1\_1,1);///////0=high

digitalWrite(S1\_2,1);

digitalWrite(S2\_1,1);

digitalWrite(S2\_2,0);

digitalWrite(S3\_1,1);

digitalWrite(S3\_2,1);

digitalWrite(S4\_1,1);

digitalWrite(S4\_2,1);

dis(digit2b);

delay(c);

}

void s3d1()

{

digitalWrite(S1\_1,1);///////0=high

digitalWrite(S1\_2,1);

digitalWrite(S2\_1,1);

digitalWrite(S2\_2,1);

digitalWrite(S3\_1,0);

digitalWrite(S3\_2,1);

digitalWrite(S4\_1,1);

digitalWrite(S4\_2,1);

dis(digit3a);

delay(c);

}

void s3d2()

{

digitalWrite(S1\_1,1);///////0=high

digitalWrite(S1\_2,1);

digitalWrite(S2\_1,1);

digitalWrite(S2\_2,1);

digitalWrite(S3\_1,1);

digitalWrite(S3\_2,0);

digitalWrite(S4\_1,1);

digitalWrite(S4\_2,1);

dis(digit3b);

delay(c);

}

void s4d1()

{

digitalWrite(S1\_1,1);///////0=high

digitalWrite(S1\_2,1);

digitalWrite(S2\_1,1);

digitalWrite(S2\_2,1);

digitalWrite(S3\_1,1);

digitalWrite(S3\_2,1);

digitalWrite(S4\_1,0);

digitalWrite(S4\_2,1);

dis(digit4a);

delay(c);

}

void s4d2()

{

digitalWrite(S1\_1,1);///////0=high

digitalWrite(S1\_2,1);

digitalWrite(S2\_1,1);

digitalWrite(S2\_2,1);

digitalWrite(S3\_1,1);

digitalWrite(S3\_2,1);

digitalWrite(S4\_1,1);

digitalWrite(S4\_2,0);

dis(digit4b);

delay(c);

}

void dis(int num)

{

for(j=0;j<7;j++)

{

digitalWrite(Arduino\_Pins[j],Segment\_Pins[num][j]);

}

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body{

background-color: #F89995;

}

/\*responsive webpage\*/

@media screen and (max-width:663px)

{

.boxwrapper2

{

margin-left:-83%;

margin-top:30%;

}

}

</style>

</head>

<body>



Figure

<!--building block-->

<div class="boxwrapper2" id="emer" style='width:150px; height:60px; background-color:#ADD8E6; position: absolute; left: 310px; top: 200px;'>Police Station,Hosptial,Fire Station</div><!--police,Hosptial,Fire Station-->

<div class="boxwrapper2" id="school" style='width:90px; height:60px; background-color:#ADD8E6; position: absolute; left: 800px; top: 200px;'>School</div><!--school-->

<div class="boxwrapper2" id="residence" style='width:80px; height:60px; background-color:#ADD8E6; position: absolute; left: 870px; top: 270px;'>Residence</div><!--Residence route 5-->

<div class="boxwrapper2" id="residence2" style='width:80px; height:60px; background-color:#ADD8E6; position: absolute; left: 620px; top: 270px;'>Residence2</div><!--Residence route 1-->

<div class="boxwrapper2" id="shops" style='width:80px; height:60px; background-color:#ADD8E6; position: absolute; left: 710px; top: 520px;'>Shops</div><!--Shops route 6-->

<div class="boxwrapper2" id="KingsBay Mall" style='width:90px; height:60px; background-color:#ADD8E6; position: absolute; left: 800px; top: 620px;'>KingsBay Mall</div><!--shopping mall-->

<div class="boxwrapper2" id="bank" style='width:90px; height:60px; background-color:#ADD8E6; position: absolute; left: 520px; top: 200px;'>Bank</div><!--Bank-->

<div class="boxwrapper2" id="gas station" style='width:60px; height:50px; background-color:#ADD8E6; position: absolute; left: 390px; top: 310px;'>Gas Station</div><!--GasStation-->

<div class="boxwrapper2" id="food court" style='width:60px; height:50px; background-color:#ADD8E6; position: absolute; left: 460px; top: 310px;'>Food Court</div><!--Food-->

<div class="boxwrapper2" id="park" style='width:130px; height:50px; background-color:#ADD8E6; position: absolute; left: 390px; top: 360px;'>Park</div><!--Park-->

<div class="boxwrapper2" id="Stark Industrie" style='width:70px; height:50px; background-color:#ADD8E6; position: absolute; left: 390px; top: 505px;'>Stark Industries</div><!--StarkIndustries-->

<div class="boxwrapper2" id="condominium" style='width:90px; height:50px; background-color:#ADD8E6; position: absolute; left: 470px; top: 505px;'>Condominium</div><!--Condo-->

<div class="boxwrapper2" id="restaurant" style='width:130px; height:50px; background-color:#ADD8E6; position: absolute; left: 390px; top: 455px;'>Restaurant</div><!--Restaurant-->

<div class="boxwrapper2" id="shops2"style='width:130px; height:60px; background-color:#ADD8E6; position: absolute; left: 310px; top: 605px;'>Shops2</div><!--Shopsroute 10-->

<div class="boxwrapper2" id="Wayne Enterprises" style='width:90px; height:60px; background-color:#ADD8E6; position: absolute; left: 520px; top: 605px;'>Wayne Enterprises</div><!--wayne enterprises-->

<!--all road-->

<div class="boxwrapper2" id="1" style='width:50px; height:150px; background-color:<?php echo $colour[0]; ?>; position: absolute; left: 560px; top: 260px;'>1</div><!--1-->

<div class="boxwrapper2" id="2" style='width:200px; height:50px; background-color:<?php echo $colour[1]; ?>; position: absolute; left: 610px; top: 410px;'>2</div><!--2-->

<div class="boxwrapper2" id="3" style='width:50px; height:150px; background-color:<?php echo $colour[2]; ?>; position: absolute; left: 560px; top: 455px;'>3</div><!--3-->

<div class="boxwrapper2" id="4" style='width:200px; height:50px; background-color:<?php echo $colour[3]; ?>; position: absolute; left: 360px; top: 410px;'>4</div><!--4-->

<div class="boxwrapper2" id="5" style='width:50px; height:150px; background-color:grey; position: absolute; left: 810px; top: 260px;'>5</div><!--5-->

<div class="boxwrapper2" id="6" style='width:50px; height:150px; background-color:grey; position: absolute; left: 810px; top: 460px;'>6</div><!--6-->

<div class="boxwrapper2" id="7" style='width:200px; height:50px; background-color:<?php echo $colour[0]; ?>; position: absolute; left: 360px; top: 260px;'>7</div><!--7-->

<div class="boxwrapper2" id="8" style='width:50px; height:150px; background-color:grey; position: absolute; left: 310px; top: 260px;'>8</div><!--8-->

<div class="boxwrapper2" id="9" style='width:50px; height:150px; background-color:grey; position: absolute; left: 310px; top: 455px;'>9</div><!--9-->

<div class="boxwrapper2" id="10" style='width:200px; height:50px; background-color:<?php echo $colour[2]; ?>; position: absolute; left: 360px; top: 555px;'>10</div><!--10right-->

</body>

</html>

**\*HTML(have more pHp coding inside) & JAVA SCRIPTS CODING REFER DISC ATTACHED ON COVER PAGE\***

**pHp Coding (for typical.php)**

<?php

require\_once('C:xampp/mysqli\_connect\_cross.php');//require to connect to database,this file is not advisable to save within same folder for security purpose,must inside main xampp file

$day="";//variable to accept day input from dropbox

$time2="";//variable to accept time input from dropbox,cannot use time because got time function,may be confusing

///////////////////////////////////////////////////////////////////////////////dropbox for day

echo '<form action="/fyp/typical.php" action="GET">';

echo '<select name="day">';

echo'<option value="Sunday">Sunday</option>';

echo '<option value="Monday">Monday</option>';

echo '<option value="Tuesday">Tuesday</option>';

echo ' <option value="Wednesday">Wednesday</option>';

echo ' <option value="Thursday">Thursday</option>';

echo ' <option value="Friday">Friday</option>';

echo ' <option value="Saturday">Saturday</option>';

echo '</select>';

//////////////////////////////////////////////////////////////////////////////////dropbox fir time,only from 5am to 12am

echo '<select name="time">';

for($hours=5; $hours<24; $hours++)

{

for($mins=0; $mins<60; $mins+=30)

{

$time = str\_pad($hours,2,'0',STR\_PAD\_LEFT).':'.str\_pad($mins,2,'0',STR\_PAD\_LEFT).':00';

echo '<option value= "'.$time.'">'.$time.'</option>';

}

}

echo '</select>';

echo '<input type="submit" value="Submit">';

echo '</form>';

if(isset($\_GET['day'])) //accept value day

{

$day = $\_GET["day"] ;

}

if(isset($\_GET['time'])) //accept value time

{

$time2 = $\_GET["time"] ;

}

echo "<h2>$day $time2</h2>";//print out time and value so user could interface better

///////////////////////////to increase range of time2 by 30 minutes,to calculate data within this tme range , based on the number to determine how situation of traffic

$time3 = strtotime($time2);

$startTime = date("H:i:s", strtotime('-0 minutes', $time3));

$endTime = date("H:i:s", strtotime('+29 minutes', $time3));

$rr=array();//declare array for route

$rr=array("1","2","3","4");//insert route 1,2,3,4 into array for looping

$colour=array();//declare array for color

$dday=0;//for increment inside for loop for 3 week

$week=0;

$sql = "SELECT DATE\_FORMAT(ttime, \"%d\") AS dayy,DATE\_FORMAT(ttime, \"%m\") AS weekk FROM `crossroad` WHERE ttime >= DATE(NOW()) - INTERVAL 5 WEEK AND ttime <= DATE(NOW()) - INTERVAL 2 WEEK AND DAYNAME(ttime)='$day' LIMIT 1";//to select first date and time of the 3 weeks

$result = $dbc->query($sql);

if ($result->num\_rows > 0)

{

// output data of each row

while($row = $result->fetch\_assoc())

{

$dday=$row["dayy"];

$week=$row["weekk"];

}

}

$dday2=$dday+7;//for week 2 date

$dday3=$dday2+7;//for week 3 date

//to calculate the average of 3 weeks data to predict the traffic in fourth week

for($a=0;$a<4;$a++)//loop for 4 route

{

$query=sprintf("SELECT FORMAT(AVG(traffic),1) AS traffic FROM

(SELECT COUNT(\*) AS traffic FROM crossroad WHERE ttime >= '2018-$week-$dday $startTime ' AND ttime <= '2018-$week-$dday $endTime ' AND route='$rr[$a]'

UNION ALL

SELECT COUNT(\*) AS traffic FROM crossroad WHERE ttime >= '2018-$week-$dday2 $startTime ' AND ttime <= '2018-$week-$dday2 $endTime ' AND route='$rr[$a]'

UNION ALL

SELECT COUNT(\*) AS traffic FROM crossroad WHERE ttime >= '2018-$week-$dday3 $startTime ' AND ttime <= '2018-$week-$dday3 $endTime ' AND route='$rr[$a]') AS traffic");

$result = $dbc->query($query);

$row = mysqli\_fetch\_assoc($result);

$var1=$row["traffic"];

if($var1<=5&&$var1>=0)//based on data predict the situtaion of traffic

{

$colour[$a]="green";//smooth traffic

}

else if($var1<=8&&$var1>5)

{

$colour[$a]="orange";//incrasing traffic

}

else if($var1>8)

{

$colour[$a]="red";//serious traffic

}

else

{

$colour[$a]="grey";

}

}

?>

**\*HTML(have more pHp coding inside) & JAVA SCRIPTS CODING REFER DISC ATTACHED ON COVER PAGE\***

**pHp Coding(for emergency.php)**

<?php

require\_once('C:xampp/mysqli\_connect\_cross.php');//require to connect to database,this file is not advisable to save within same folder for security prupose,must inside main xampp file

date\_default\_timezone\_set("Asia/Kuala\_Lumpur");//display date

echo $ddate=date("Y-m-d") . "<br>";

//////////////////////////////////////////////////////code same with live traffic,to show live traffic which is happening now

$rr=array();//declare array for route

$rr=array("1","2","3","4");//insert route 1,2,3,4 into array for looping

$colour=array();//declare array for color

$var1=0;//for calculating time needed to arrive at location

for($a=0;$a<COUNT($rr=array("1","2","3","4"));$a++)//live traffic

{

$sql = "SELECT \* FROM crossroad WHERE ttime BETWEEN DATE\_SUB(NOW(),INTERVAL 10 MINUTE) AND NOW() AND route='$rr[$a]'";//select data between now and 10mins ago

$result = $dbc->query($sql);

$var2 = mysqli\_num\_rows($result);

if($var2<=1)//based on data predict the situtaion of traffic

{

$colour[$a]="green";//smooth traffic

}

else if($var2>=2&&$var2<=3)

{

$colour[$a]="orange";//incrasing traffic

}

else if($var2>4)

{

$colour[$a]="red";//serious traffic

}

else

{

$colour[$a]="grey";

}

}

?>

**\*HTML(have more pHp coding inside) & JAVA SCRIPTS CODING REFER DISC ATTACHED ON COVER PAGE\***

**pHp Coding(for live.php)**

<?php

require\_once('C:xampp/mysqli\_connect\_cross.php');//require to connect to database,this file is not advisable to save within same folder for security prupose,must inside main xampp file

$rr=array();//declare array for route

$rr=array("1","2","3","4");//insert route 1,2,3,4 into array for looping

$colour=array();//declare array for color

for($a=0;$a<COUNT($rr=array("1","2","3","4"));$a++)

{

$sql = "SELECT \* FROM crossroad WHERE ttime BETWEEN DATE\_SUB(NOW(),INTERVAL 10 MINUTE) AND NOW() AND route='$rr[$a]'";//select data between now and 10mins

$result = $dbc->query($sql);

$var1 = mysqli\_num\_rows($result);

if($var1<=1)//based on data predict the situtaion of traffic

{

$colour[$a]="green";//smooth traffic

}

else if($var1>=2&&$var1<=3)

{

$colour[$a]="orange";//incrasing traffic

}

else if($var1>=4)

{

$colour[$a]="red";//serious traffic

}

else

{

$colour[$a]="green";

}

}

?>

**\*HTML(have more pHp coding inside) & JAVA SCRIPTS CODING REFER DISC ATTACHED ON COVER PAGE\***