PALESTINE POLYTECHNIC UNIVERSITY

College of IT and Computer Engineering

Department of Computer System Engineering

Introduction to Graduation Project

Project Name

Smart Traffic Lights Signals

Project Team

Haneen dwayat

Khalel Awwad

Sajeda Jawabreh

Supervisor

Dr.Mousa Farajallah

September 2020

Acknowledgement

In the name of "Allah", the most beneficent and merciful who gave us strength, knowledge and helped us to get through this project. Our heartiest thanks due to parents for their love and kind support, which helped us in the project.

We know the thanks are not enough and there are not enough words to describe how thankful to them.

We would like to express our thanks to our supervisor Dr.Mousa Farajallah. We are highly indebted to him for his support and help in the project. Thanks for Palestine Polytechnic University and all staff for their effort in teaching and guiding us, we are also grateful to all of our friends especially in the Computer Systems Engineering program and in the College of Information Technology and Computer Systems Engineering in Palestine Polytechnic University

Abstract

Traditional traffic lights are very important in regulating traffic. However, traffic jam problems have not been completely solved. Therefore, there is a real need for using modern technologies to upgrade traditional traffic lights to be particularly smart.

Our solution aims to upgrade normal traffic light to be smart traffic light; making traffic lights communicates with each other and letting an algorithm choose the state of each traffic light. By building an embedded system including a group of sensors, wireless communication component and microcontrollers, these devices will collect specific data and then send them to the system. System based on the implemented algorithm will decide which and when a traffic light will be green, and how many times will it do that. In addition, each traffic light communicates with the closest ones, by sending signals and data regarding the street traffic alerting the next traffic light of a possible crisis or emergency vehicles arriving. According to the number and the type of vehicles in each road.

Building a multiple smart traffic light system such as the ones described above would offer an optimal usage of the traffic lights on roads. Also, would be beneficial to all stakeholders of this system, by not only saving human lives and their most valuable possession (their time) but also helping the environment, by reducing carbon emissions, all this will be achieved by reducing the time the car takes at traffic lights.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abbreviations

RAM	Random Access Memory
USB	Universal Serial Bus
GPIO	General Purpose Input/Output.
IOT	Internet of Things
IDE	Integrated development environment.
PC	Personal computer
OS	Operating system
AC	alternating current
DC	direct current
Wi-Fi	wireless fidelity
API	Application program interface
IO	Input / Output
Hz	Hertz
PWM	Pulse Width Modulation
mA	Millie ampere
Cm	Centimeters
HTTP	Hypertext Transfer Protocol
IOS	IPhone operating system

Chapter 1: Introduction

1.1 Overview

Traffic jams have become an everyday problem affecting human being whether having a private vehicle or taking public transportation. This problem has been increasing due to the continuously rising number of vehicles around the world, and causing major inconveniences to everyone; farther more wasting valuable time and endangering the life of people during cases of emergencies; as in restricting the movement of ambulances or fire trucks.

1.2 Project Description

This project propose in order to reduce traffic jams; the methodology of the project is as follows:

The system contains a street with four crossroads as shown in figure 1.1, the system detects how many vehicles in each crossroad and calculates the distance between the sensor that placed on the end edge of the road- and first vehicle, calculating distance helps the microcontroller to detect the number of vehicles according to the capacity of the street, then it counts how many vehicles pass each delta seconds. The microcontroller takes these data and passes it into a specific algorithm that determines the priority for each traffic signal at each crossroad. The algorithm prioritizes the street that has the largest amount of vehicles for turning on the green light there and allowing the largest number of vehicles to pass through in an efficient and timely manner, other roads that have less number of vehicles have second priority, And so on.
Traffic lights can also communicate with one another preparing them for a crisis.

Figure 1.1 cross road

1.3 Project Importance

The high increase in the number of vehicles on the road and urbanization, leading to a problem of congestion that it is one of the most difficult things facing drivers in the streets, result, the delay, which occurs in the transportation system, endangers (private and public vehicles), the main goal for this project is to reach traffic lights system that will solve traffic congestion and efficiently use road:

1. Saving the waste of valuable time.
2. Reducing environmental pollution.
3. Decreasing the noise pollution.
4. Reducing fuel and transport costs.

1.4 Problem Statements

1.4.1 Problem Analysis

If you resident in a large city or any area where there are many vehicles on the road, then you probably know the stress caused by the delay that can make people late for work. Besides, at the end of the day, the afternoon rush hours are again a frustrating time because the workday is done. As a result, This gridlock can have a tremendous impact on our personal life, career, and even our safety. Moreover, in traffic jams fuel burns at a higher rate than in the open highway that means the environment is also one of traffic jams victims.

1.4.2 Purpose

The evolution of traffic lights is focused on safety, reduce power consumption, and performance. To make them more functional, environment-friendly and 'smart'. The achievement of all these performances relies on a well-designed real-time control system.

1.4.3 Objective

The smart traffic light system will achieve the following objectives:

1- Minimizing the problem of road congestion by prioritizing more congested streets.
2- Handling the empty streets that cause wasting time.
3- Contributing to make cars movement tidily and fast at traffic lights.

1.4.4 Motivations

Did you wonder why is the traffic light is green, in an empty street instead of another which has a crisis?

Vehicular congestion has become one of the most problematic issues that our society is facing in the modern era, and has been increasing at an exponential rate since it has begun.

The first motivation of this project is reducing the wasted time during traditional traffic lights. According to a study from the transportation-consulting firm INRIX, tens of billions of dollars are being lost because of the time stuck in traffic each year [2]. In addition, fuel consumption is being increased resulting emission of pollutant gas, so traffic congestion has a serious impact on the environment, which must be taken into consideration.

1.4.5 Requirements

The system requirements summarized as:

1. The system should have sensors modules, which provide reliable and accurate information like the count of vehicles on each road.
2. Microcontrollers that collect information from external modules then determine the status of a traffic light for each road.
3. Bluetooth module, to send and receive data from microcontrollers.

1.5 Expected results

1. This smart traffic light system smooth traffic flows and prioritizing traffic on demand for real-time.
2. Handle with emergency cases at any time.

1.6 Overview of report

The outline of the report is as follows: Next chapter “background”, contains the theoretical background and Literature review, options (design options for hardware components and design options for software components) and design constraints. The Third chapter “design”, includes a detailed conceptual description of the system (Hardware and Software), detailed design, schematic diagrams, block diagrams, structural diagrams, and any necessary information about the design will be presented in chapter three

Chapter 2: Background

2.1 Overview

This chapter introduces a theoretical background of the project, some description of hardware and software components are used in the system finally, discussion of specification and design constraints are presented.

2.2 Theoretical background

This project aims to reduce traffic crises that result from a lack of optimal usage of traffic lights. The delay that is caused by these crises could lead to dangerous situations that affect people lives.

Our project focuses on the idea of converting normal traffic lights into smart and communicate ones that help in solving three main cases, they should have the ability to make decisions about priority for each traffic light, and communicate with another traffic light in a different region.

The proposed system is supposed to solve these cases by first, counting the number of vehicles in the road, using proximity sensors, which set at the beginning and end edge of the road. Then the sensors send the data to a microcontroller in order to processed with specific algorithm, then prioritize each traffic light depending on features that have been calculated (time, distance and count of vehicles).

2.3 Literature Review

This section reviews some of previously proposed solutions for traffic congestion problems. Through some scientific papers and published sources about the problem of traffic congestion, we came across very limited solutions that are related to the current project.

1. Intelligent Traffic Light Guidance System

Place: Palestine Polytechnic University - December 2014.Team: Ansar Dabbas, Sahar Froukh. Supervisor: Dr.Radwan Tahboub.

The main idea of this project is to guide and direct drivers to the less crowded streets through devices linked to traffic lights, which determine the nature of this street and the degree of traffic congestion in it, all of this will be achieved by using cameras placed on the roads and linked to the microcontrollers, the microcontrollers have specific algorithm that defines the most streets busiest and gives drivers alerts avoid these roads.

Nevertheless, the question here, what if there was only one street leading to the place the driver wanted to go to and this street is crowded at this time?

This project helps drivers use the road, using a range of sensors related to a microcontroller, which arranges the priorities of the streets not just giving an alert about the crowded streets.

2. Waze

It is a mobile application, developed by Google, that collects information from drivers and people crossing the roads and verifies their information, users of this application can report incidents on the roads, the application contains street status, there are accidents or not, there is a crisis or not, and the speed limits allowed to drive [14].

All solutions related to solving the problem of road congestion only indicate the place of congestion and do not solve it, which means are not intelligent solution but what makes our project different is that is solves the problem of traffic congestion before it happens and try to decrease its impact on transportation and transport of emergency vehicles.

2.4 Design options of Hardware components

This section describes all of the hardware used in our project, it presents a figure for each one with short description about its work principle and why it is used in the system.

2.4.1 Microcontroller

- First Design Option: Raspberry Pi.
  A small single-board computer developed by the Raspberry Pi foundation, selling outside its target market for uses such as robotics. It has many specifications such as 1G RAM, 4 USB ports, 40 GPIO Pins, and Ethernet port. We can install many operating systems on it, such as Windows 10 IOT, Android things, and Ubuntu MATE.

- Second design option: Arduino.
  It is a microcontroller, open-source platform, used to make projects related to electronic or electrical objects. In addition, a part of it is used for programming Arduino hardware (IDE). Arduino makes several different boards Like (Arduino Uno R3, Arduino Mega2560, and Arduino Leonardo) and others.

Chosen design option: Arduino Mega.

We will work with an ArduinoMega2560, as it clear in Figure*2.1*; it includes lots of digital input and output pins, six analog inputs, a reset button, a power jack, a USB connection. It includes everything required to hold up the microcontroller as presented in *Table 2.1*; simply attach it to a PC with the help of a USB cable and give the supply to get started with an AC-to-DC adapter or battery, have no OS that means easily using of it. Therefore, it is great for hardware projects in which simply want things to respond to various sensor readings and manual input, and the Wi-Fi segment proposed in the project is needed, which is not supported by other controllers

Specification:

operating voltage	5v
digital input/output pins	54 where 15 of these pins will supply PWM
Memory	256 KB
Clock Speed	16 MHz
Current Rating per I/O pin	20 mA

Table 2.1: Arduino Mega Specification. [3]

Figure 2.1: Arduino Mega [3]

2.4.2 Detecting cars sensor

Many sensors can detect objects cross in front, whether it is a car or not:

- First Design Option: Ultrasonic sensor.

Ultrasonic sensor as shown in *Figure 2.2*, it is an ultrasound wave sensor when these waves reflect off an object, the sensor sends signals, indicating that there is an object in the range of this sensor, and it used to measure distance and more specification as presented in *Table 2.1* .

Specifications:

Supply voltage	5 v
Global Current	15 mA
Ultrasonic Frequency	40 Hz
Maximal Range	400 cm
Minimal Range	3 cm
Resolution	1 cm
Trigger Pulse Width	10 µs
Outline Dimension	43x20x15 mm

Table 2.2: ultrasonic sensor specification [4

Fig 2.2 ultrasonic sensor [4]

- Second Design Option: Infrared sensor.
  Infrared sensor (IR) as shown in *Figure 2.3* , it is an electronic device, it emits infrared radiation, if an object passes through it, it will detect it; here are two types of infrared sensors: active and passive. Active infrared sensors both emit and detect infrared radiation, and they are commonly used in obstacle detection systems (such as in robots) and more specification as presented in *Table 2.3*.

Specification:

Detection distance:	2-10cm
Power supply:	3.3 to 5VDC
output interface	3-wire output interface
Output	Digital output
Signal:	Active low
Sensitivity	Adjustable
Size:	30mm x 14mm x 5mm

Table2.3: IR sensor specification [5]

Fig 2.3 IR Sensor

- Third design option Inductive Proximity Sensor as shown in Figure 2.4, it is a sensor that detection of metallic objects, without any contact. Proximity sensors is featuring by high effectiveness in detecting objects because of the absence of mechanical parts and lack of physical contact between the sensor and the sensed object and more specification as presented in *Table 2.4*.

Specification:

Operating voltage	2.5 v– 3 v
Range	0-10 cm
Operating temperature	-20 – 70 c

Table 2.4: proximity sensor specification [6]

Figure 2.4: Proximity sensor

Chosen Design Option: Inductive proximity sensor

We have selected this sensor because it guarantees a range of features that other sensors does not provide such as:
1. Working in difficult weather status.

2. High precision provided.

3. It has a sleeping mode, which helps conserve energy.

4. fast readings, which helps speed up the decision-making process.

2.4.3 Communication between two microcontrollers

We need a communication method between two arduinos.

- First Design Option: NodeMCU ESP8266 Breakout Board
  NodeMCU as shown in Figure 2.5, it is an open source IOT platform, its Wi-Fi component using for wireless communication between the client (ambulance) and server (traffic light), it based on ESP2866 module and it use same technology, it have specification as presented in *Table2.5.*

Specification:

Operating Voltage	2.5V ~ 3.6V
Security	WPA/WPA2
Encryption	WEP/TKIP/AES
Network Protocols	IPv4, TCP/UDP/HTTP
Operating Temperature Range	–40°C ~ 125°C
Wi-Fi Mode	Station/SoftAP/SoftAP+Station

Table 2.5: NodeMCU Specification [8]

Figure 2.5 NodeMCU module [8]

- Second design option Bluetooth Module as shown in *Figure 2.6* , It is a module used to communicate between the components using AT commands and designed for transparent wireless serial connection setup, The HC-05 Bluetooth Module making a great solution for wireless communication , it have specification as presented in *Table2.6*.

Specification:

Voltage supply	3.3 v
Storage Temperature	40℃ to +150℃
Frequency Band	2.4GHz ISM Band
Application	Wireless Control

Table 2.6: Bluetooth Module HC-05 specification [7]

Figure 2.6: Bluetooth Module HC-05 [7]

Chosen Design Option: Bluetooth Module HC-05

We have selected this module because it guarantees a range of features that other modules do not provide:

Two ways are close enough to each other, security issues to avoid hacking, directly connection without any host network

2.5 System Software Component

This section provides some information about the main programs and software technologies are used in our project.

1. Hardware configuration

All hardware parts will be connected to the Arduino microcontroller to read data from them, so we will be using the following software:

Arduino IDE

The Arduino integrated development environment (IDE) or Arduino software: is a cross platform that is written in the programming language C. It is used to write and upload programs to the Arduino board. It is flexible and easy to use.

2.6 Programming Languages

- Arduino C

2.7 System Constrains

- The need of constant and uninterrupted voltage source connects with sensors module
- The project should be placed at a crossroad that contains at least two streets.
- Traffic rules must be abiding by drivers.

Chapter 3: Design

3.1 Overview

This chapter discusses the conceptual design of the system; it shows the system requirement analysis, a block diagram of the system, structural diagram, flow chart, detailed design, and schematic diagrams.

3.2 Detailed system description

The main objectives of project are as follows:

- Calculate the vehicle quantity.
- Regulate the traffic signal timings based on a car's quantity value of traffic.
- Implement a safe and reliable method to ensure vehicles can cross through gridlocked roads thus saving time.

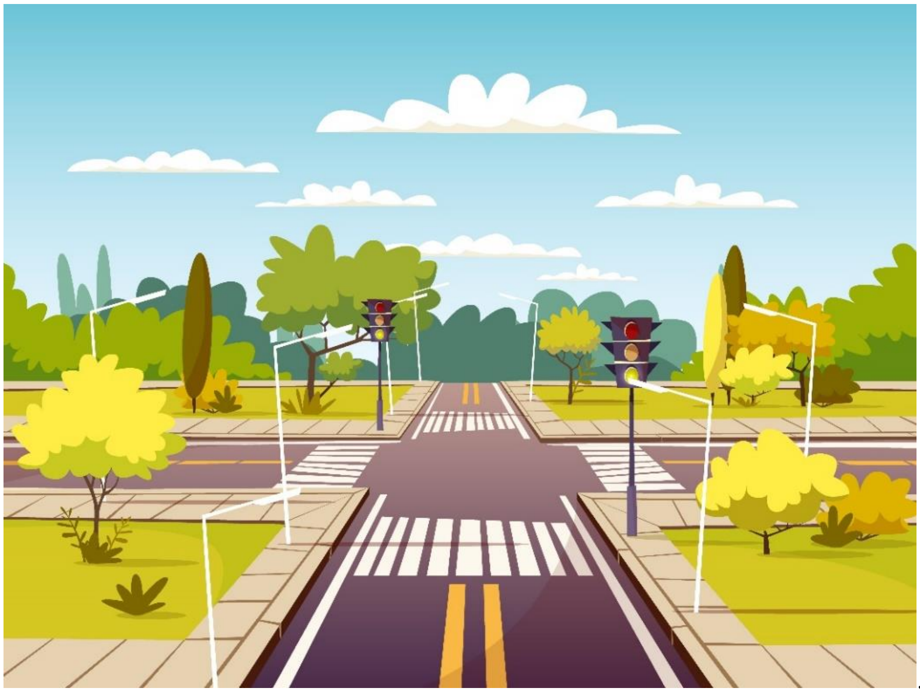


Figure 3.1 cross road with 4 ways [11]

A simple traffic light consists of a red, yellow, and green light as shown in Figure 3.2. With typical sequence as presented in Table 3.1.

1. Green (safe to cross).
2. Yellow (slow down, red light soon, get ready, green light soon).
3. Red (stop).

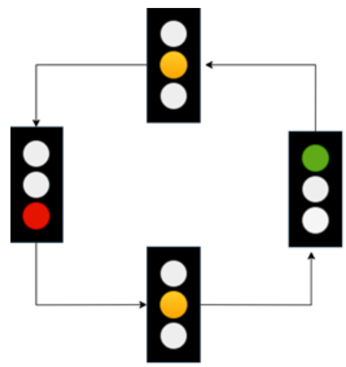


Figure 3.2 normal state of traffic lights

State .NO	East	West	North	South
1	G	R	R	R
2	Y	R	Y	R
3	R	R	G	R
4	R	Y	Y	R
5	R	G	R	R
6	R	Y	R	Y
7	R	R	R	G
8	Y	R	R	Y

In Table 3.1 presents steps or sequence of outputs to turn ON the traffic system lamps (RED, GREEN, YELLOW) in regular traffic lights- depend on timer.

3.2.1 Methodology

The project consists of three phases.

1. Vehicle Detection and Counting.
2. Decisions Based on Data.
3. Send information to the next street.

3.2.1.1 Vehicle Detection and Counting:

Steps are involved in this phase:

- Two inductive proximity sensors set, one at the beginning and another at the end for each road as shown in *Figure 3.3*.
- The sensor detects whether the vehicle is within the detection range.
- When a vehicle enters the sensing range of the device, the sensor sending this data to the microcontroller.
- Data processing and vehicle counting.

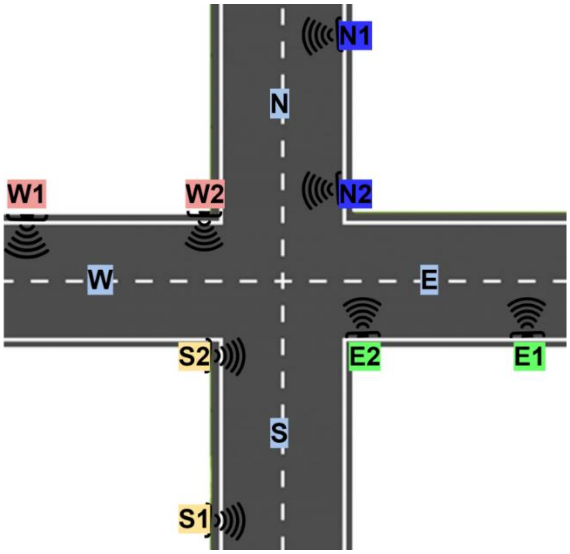


Figure 3.3 sensors positions in intersection ways.

Procedure:

Step 1: sensors place is a crucial step in detecting vehicles. The sensor should be placed such that it covers the beginning and the end of the entire lane see figure 3.3 as a example.
Step 2: the first sensor will detect the number of vehicles enter the path and the second one detecting the number of vehicles coming out. When the vehicle approaches the sensor, the induction current flow increases, which causes the load on the oscillation, the circuit to increase. Then, oscillation attenuates or stops. The sensor detects this change in the oscillation status and outputs a detection signal as shown in *Figure 3.4.* [12]

Figure 3.4 change of Sensor Behavior[12]

Step 3: the microcontroller receives, stores and processes these data.

3.2.1.2 Decisions Based on Data

The collected information is passed onto an Arduino and decides the state for each traffic light depending on the following cases:

- At normal case, the sensor at the beginning of the street returns count of vehicles that enter in the street and the second sensor returns count of vehicles that leaving out the street, the Arduino receives this data from each road, then it performing calculations, to change the timings of the LEDs, according to subtraction value between vehicles entering with vehicles leaving out this street, then return this calculation for every traffic light in each way in crossroad.

This study based on fact those streets have the same number of vehicles, otherwise, a new parameter should be considered.

$$E = \mathrm{E1} - \mathrm{E2}$$

$$W = \mathrm{W1} - \mathrm{W2}$$

$$N = \mathrm{N1} - \mathrm{N2}$$

$$S = \mathrm{S1} - \mathrm{S2}$$

E1, E2 value from the sensors at the beginning and end of the (East) street.

W1, W2 value from the sensors at the beginning and end of the (West) street.

N1, N2 value from the sensors at the beginning and end of the (North) street.

S1, S2 value from the sensors at the beginning and end of the (South) street.

Sorting count of vehicles in (E, W, N, and S) by ascending order using sort algorithm then prioritize for each traffic light.

Example E=20, N=5, S=10, W=30

W has first priority, E has second priority, S has third priority and N has fourth priority

3.2.1.3 Send information to the next street

in the previous example, the decision is the open traffic light for the 'W' path - as mentioned-, but this case depends on the number of cars in a crossroad, but if we assume that opening the traffic light at the 'W' path will lead to the closing of the other roads, this will result in greater traffic congestion in the rest of the roads - 'N', 'S', and 'E' - so the importance of this system. As the traffic light will communicate with each other.

Communication between traffic lights is very important, to preparation another traffic signal that is close to the main traffic light, for receiving a crisis, and determines its status based on the information received.

Starting communicate between two Arduinos using Bluetooth, main traffic signal as Master and closest one as a Slave as it shown in *Figure 3.5*.
The server will send data to the client, and according to this data, it changes their status as resented in *Table 3.2*.

Example : when south street fuel of cars , Master Arduino will send an order to bystreet to open automatically ; to avoid congestion at any main street

input	Output
Server sent that there is a crisis incoming and signal is red.	Signal turn yellow 3 second followed by green 10 second.
Server sent that there is a crisis incoming and signal is green.	Signal turn green extra 10 second.
No sent data	Signal works normally

Table 3.2 communication command reflection on system

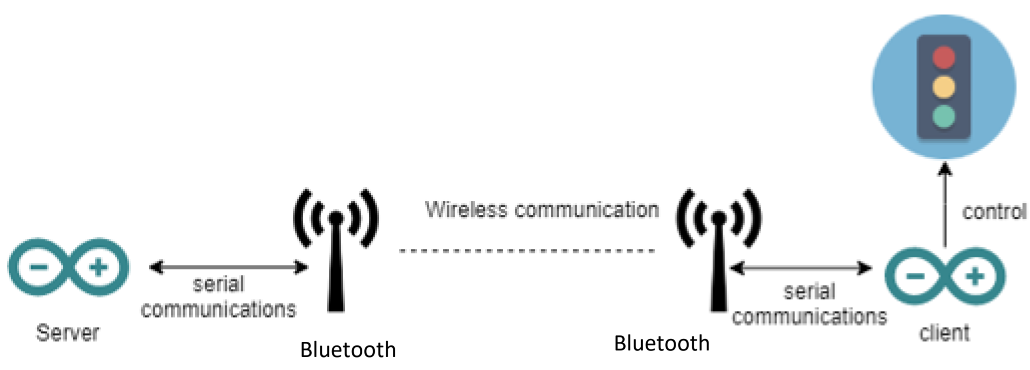


Figure 3.5 Single Peer Configuration

3.3 System Diagram

Figure 3.6 shows the general block diagram of the whole system, How the system's equipment's work with each other to achieve the needed Requirements of our project.

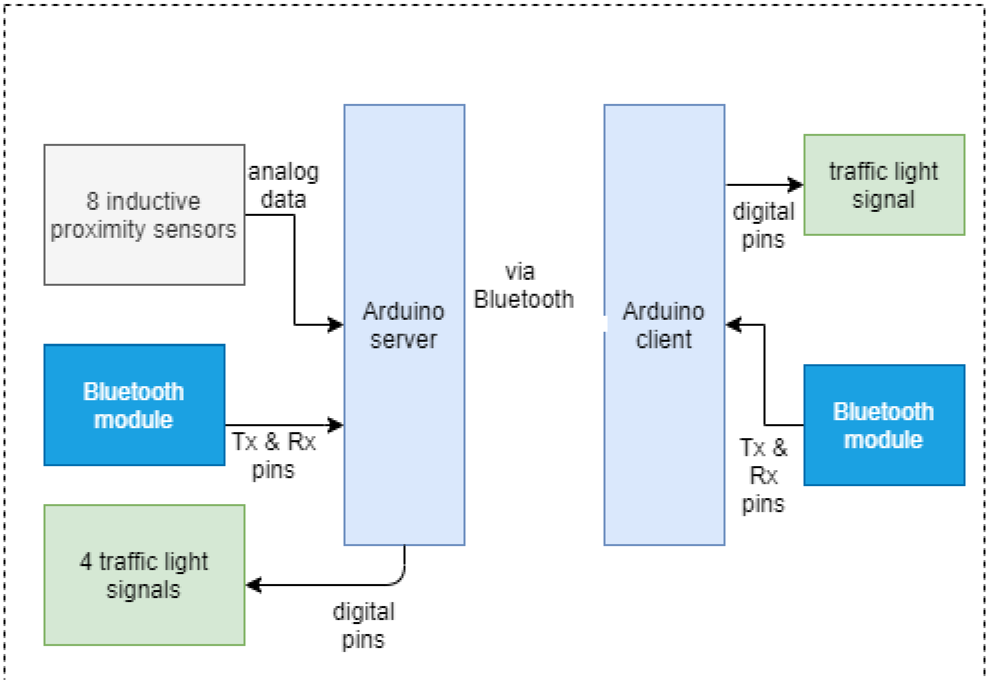


Figure 3.6 Block diagram of the system

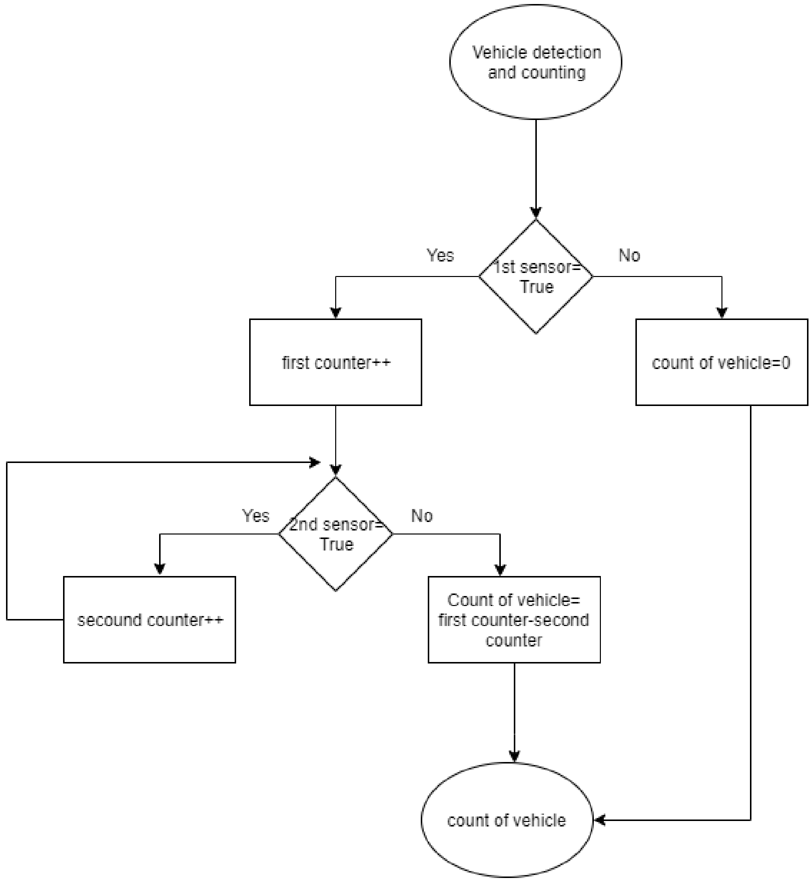
3.4 System architecture

1. Vehicle Detection and counting.

Algorithm 1: Vehicle Detection and Counting

```
Result: count of vehicles for each road
read sensor1,sensor2;
first counter,second counter=0;
while sensor1=True? do    //There is a car at the sensor field, where is at the begining of the road
    first counter++;
    if sensor2=True then  //There is a car at the sensor field, where is at the end of the road
        second counter;
        count of vehicle=first counter-second counter;
    else
        count of vehicle=first counter;;
    end
end
count of vehicle=0;
return count of vehicle;
```

Algorithm 1 vehicle detection



Flowchart 1 vehicle detection

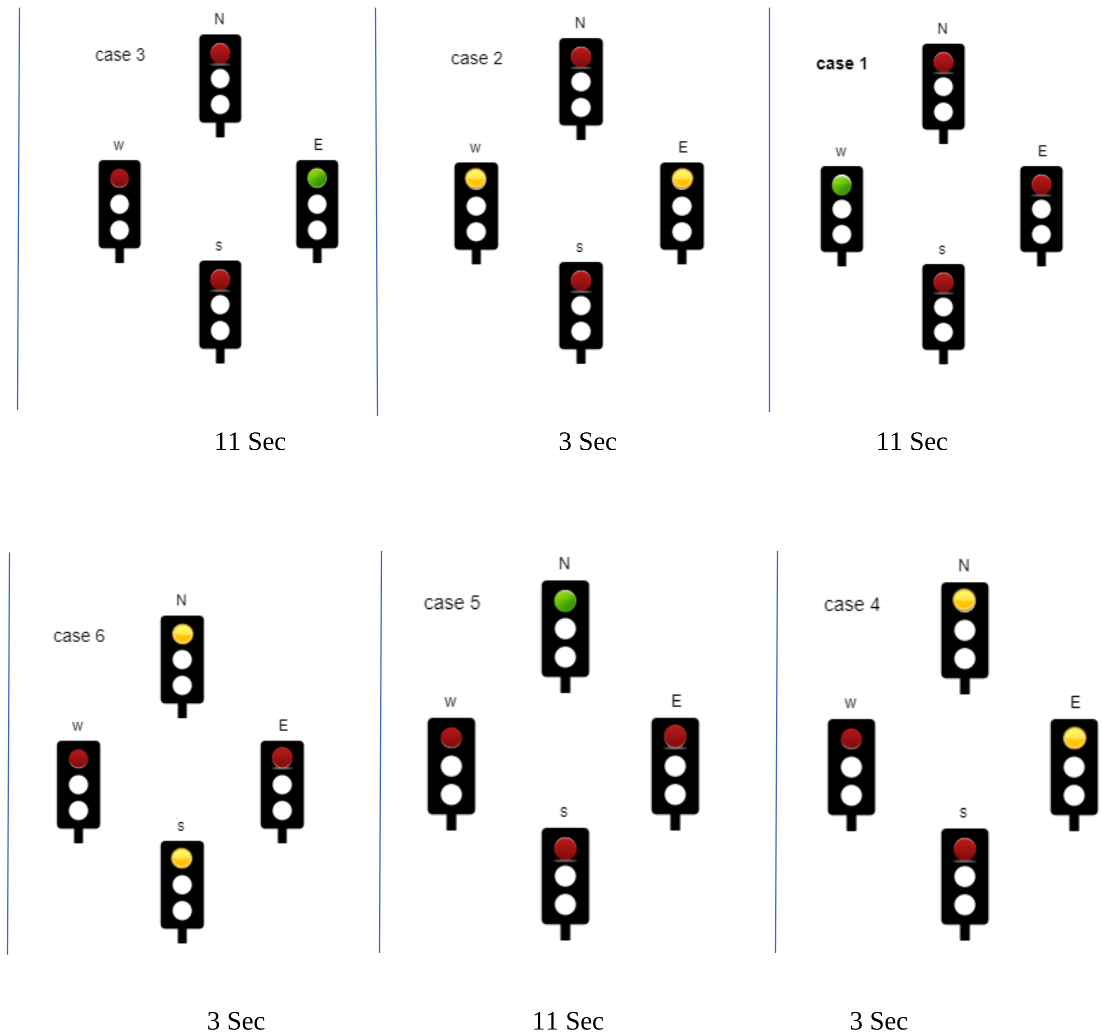
2- Decisions Based on Data

Algorithm 2: Prioritize for each traffic light based on decisions.

```
Result: Eight cases include all status of whole system for X seconds for each traffic light
INPUT : output of bubble sort algorithm // arranged array[0,1,2,3] ;
while True do
    CASE1 :
    arr[0] = output[red] for 11 second ; //lowest priority will lights red for 11 second
    arr[1] = output[red] for 11 second ; //second-lowest priority will lights red for 11-second
    arr[2] = output[red] for 11 second ; //third-lowest priority will lights red for 11-second
    arr[3] = output[green] for 11 second ; //highest priority will lights green for 11-second
    CASE2 :
    arr[0] = output[red] for 11 second ; // lowest priority will light red for 11 second
    arr[1] = output[red] for 11 second ; //second lowest priority will light red for 11 second
    arr[2] = output[yellow] for 3 second ; //third lowest priority will light yellow for 3 second
    arr[3] = output[yellow] for 3 second;// highest priority will light yellow for 3 second
    CASE3 :
    arr[0] = output[red] for 11 second ; // lowest priority will light red for 11 second
    arr[1] = output[red] for 11 second ; //second lowest priority will light red for 11 second
    arr[2] = output[green] for 11 second ; // highest priority will light green for 11 second
    arr[3] = output[red] for 11 second; // third lowest priority will light red for 11 second
    CASE4 :
    arr[0] = output[red] for 11 second ;
    arr[1] = output[yellow] for 3 second ;
    arr[2] = output[yellow] for 3 second ;
    arr[3] = output[red] for 11 second;
    CASE5 :
    arr[0] = output[red] for 11 second ;
    arr[1] = output[green] for 11 second ;
    arr[2] = output[red] for 11 second ;
    arr[3] = output[green] for 11 second
    CASE6 :
    arr[0] = output[yellow] for 3 second ;
    arr[1] = output[yellow] for 3 second ;
    arr[2] = output[red] for 11 second ;
    arr[3] = output[red] for 11 second;
    CASE7 :
    arr[0] = output[green] for 11 second ;
    arr[1] = output[red] for 11 second ;
    arr[2] = output[red] for 11 second ;
    arr[3] = output[red] for 11 second
    CASE8 :
    arr[0] = output[yellow] for 3 second ;
    arr[1] = output[red] for 11 second ;
    arr[2] = output[red] for 11 second ;
    arr[3] = output[yellow] for 3 second
end
```

Algorithm 2 Prioritize for each traffic light

All cases shown in below figure 3.7:



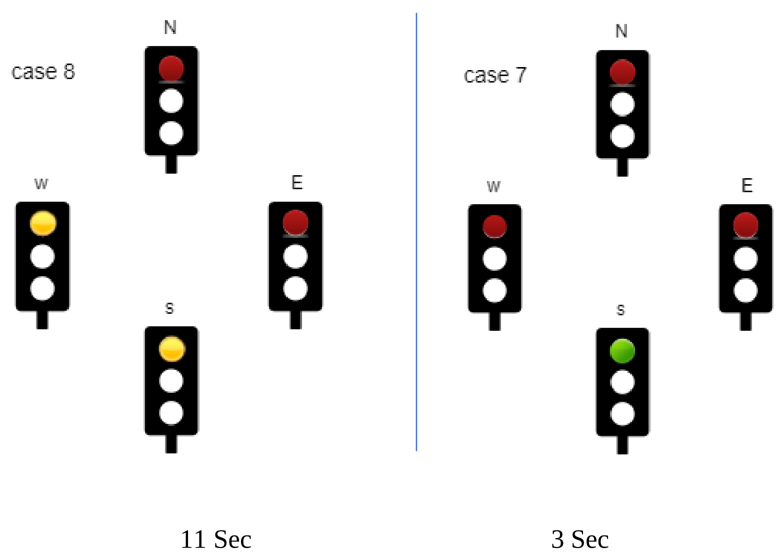


Figure 3.7 eight cases of traffic lights

4. Send information to the next street.

```
Algorithm 1: Send information to the next street
  Result: Bluetooth Communication Between Two Arduino Boards
  char msg = ' ';
  while True do
      msg=read();
      if msg=='H' then
          state=read(traffic);
          if state  =red or state=yellow then
              Write(green,HIGH);
              delay()
          else
              state=green;
              delay();
          end
      else
          work normal;
      end
  end
```

Algorithm 3 Send information to the next street

3.5 Detailed and Schematic diagrams

In this section we explained the detailed and schematic diagram for all components and explained the schematic diagram for all the system.

1. Inductive proximity sensors :

The schematic diagram of the Inductive proximity sensor with Arduino shown in *Figure 3.10.*

Red - VCC (+)	Yellow - input	**Black** Ground(-)

That is using for detection of metallic objects (for detecting only vehicles).

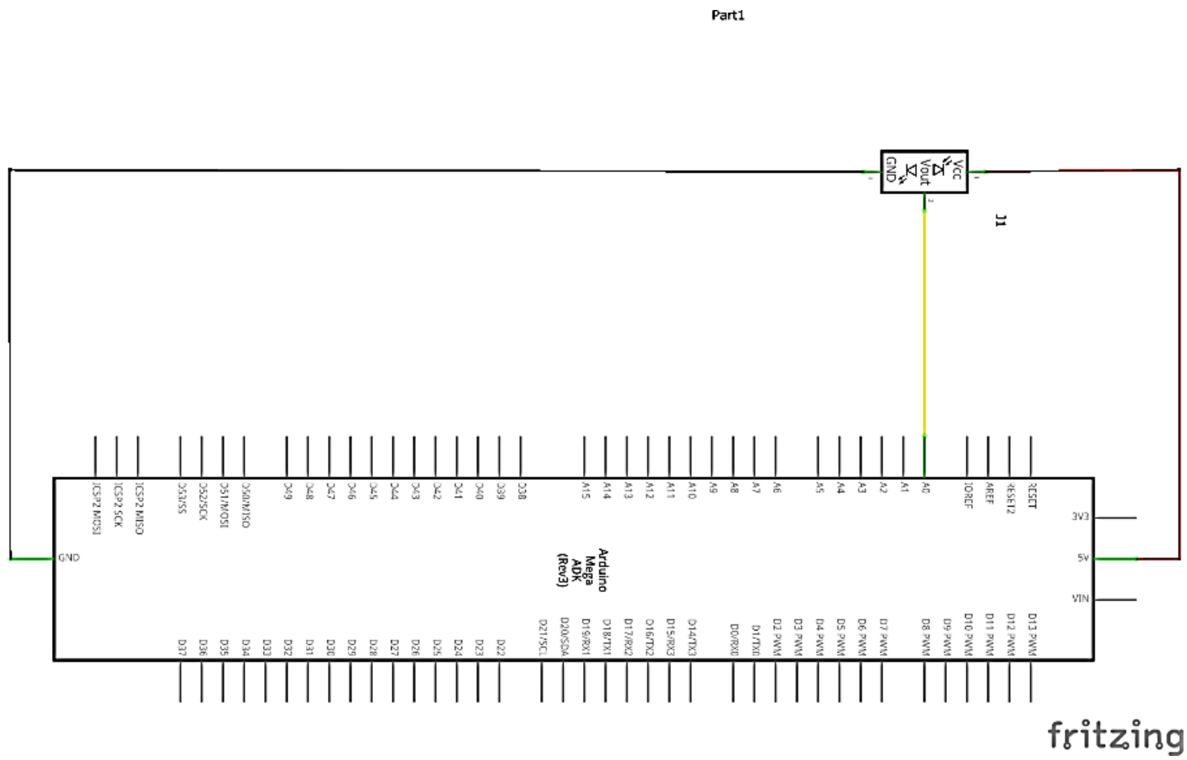


Figure 3.10 schematic diagram of the Inductive proximity sensor with Arduino

2. Bluetooth Module:

The *Figure 3.12* shows the connections of the Bluetooth module with Arduino.

.

Tx - Rx	**Rx** – Tx	Vin-5v	**ground** - Ground(-)

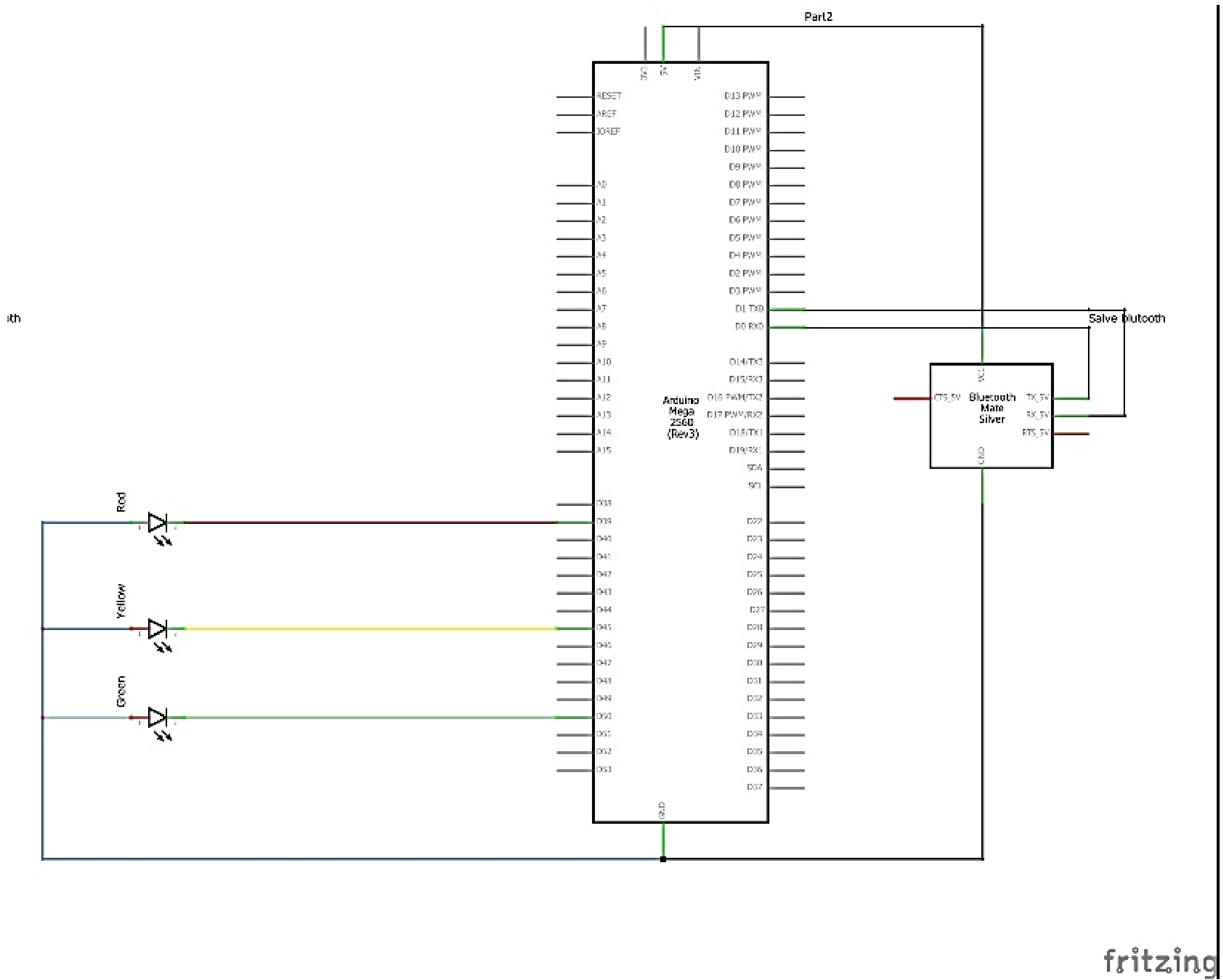


Figure 3.11 Bluetooth Module with Arduino

3. Traffic light LED’s

the schematic diagram of connection between LEDs and Arduino shown in *Figure 3.14*

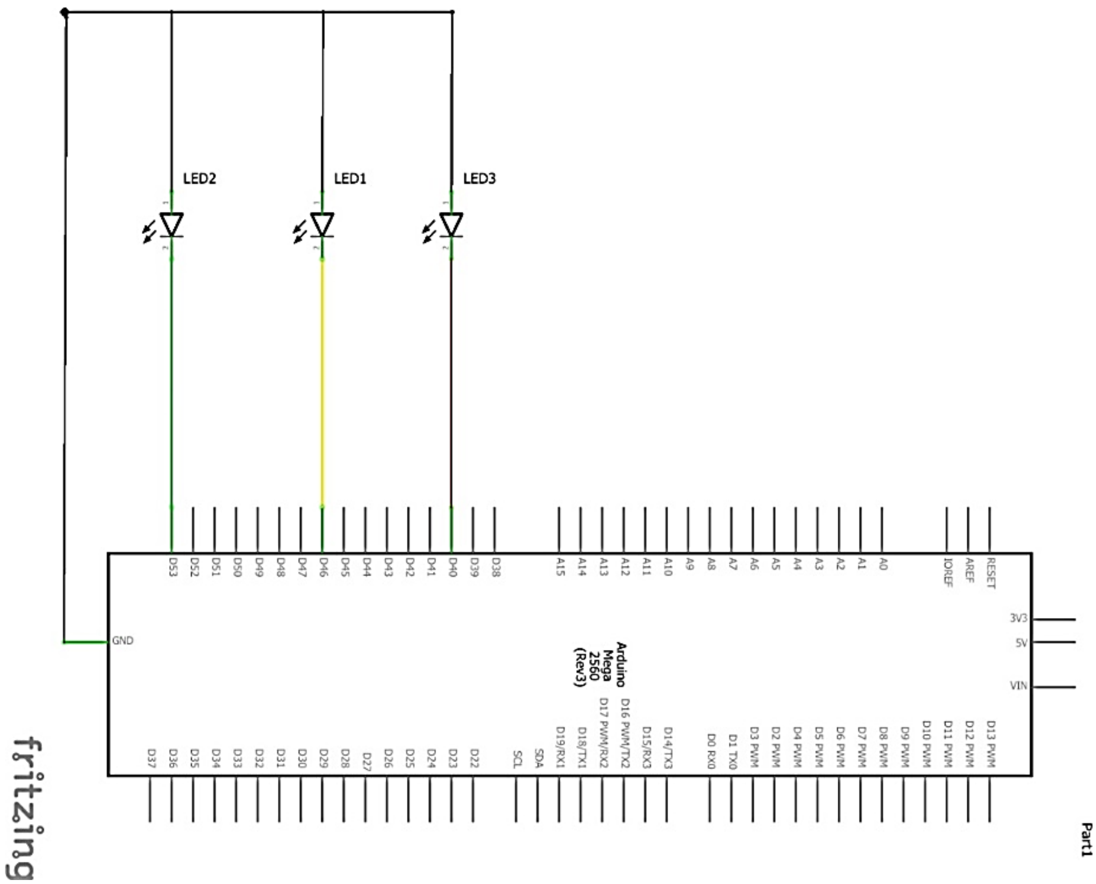


Figure 3.14 schematic diagram of LED’s with Arduino

All system component will be connected as shown in *Figure 3.16.*

.

Chapter 4 Implementation

4.1 Overview

This chapter discusses in detail the implementation part of the project.

4.2 System Hardware Assembly

Figure 3.17 shows a schematic diagram of the connection the whole system components. In this section attempts to describe the assembly process and arrangement of those components.

Hardware assembly of this system should be

- ✓ Check the direction of the wires, the port numbers, and the type of input data.
- ✓ Ensure that components are firmly installed and are not loose.
- ✓ Ensure good and accessible cable management.
- ✓ Ensure that components are working properly.

The component assembly process initiated by connecting a traffic light for each junction, which is consisting of three lights (Red, Green, and Yellow) as the Fig 4.1 below ; in this system, there are five traffic lights, four connecting with the master Arduino Mega, each junction in the four-way intersection has one traffic light, and the last one with the slave Arduino Uno.

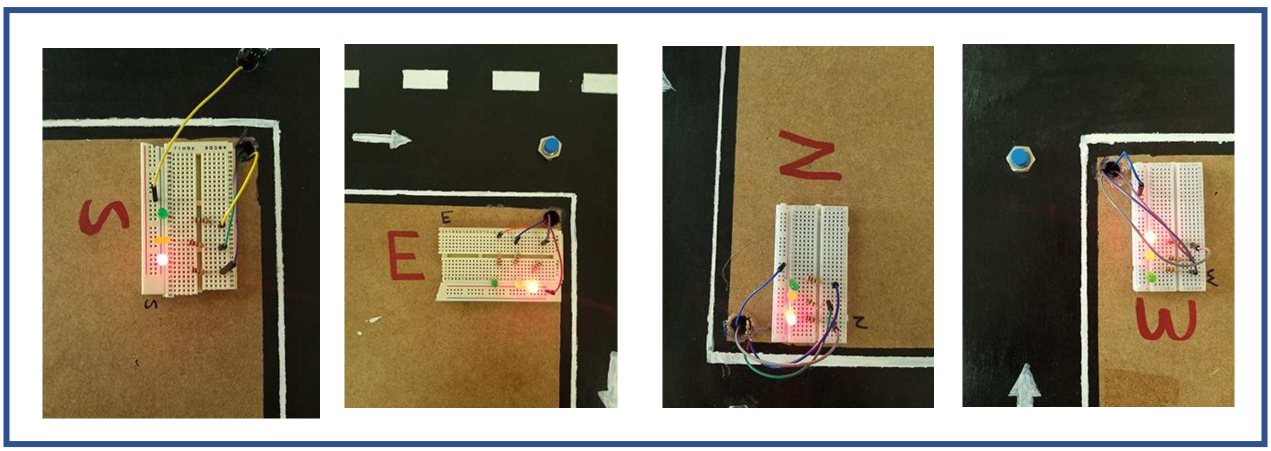


Fig 4.1: traffic lights for each junction

The second step is connecting eight sensors as an analog input to the master microcontroller Arduino (mega), which is each junction contains two sensors as shown in Figure 4.2, whereas one at the beginning of the way and one at the end of the way, then supplied these sensors with 12v voltage source, via a transformer shown in the Fig 4.3.

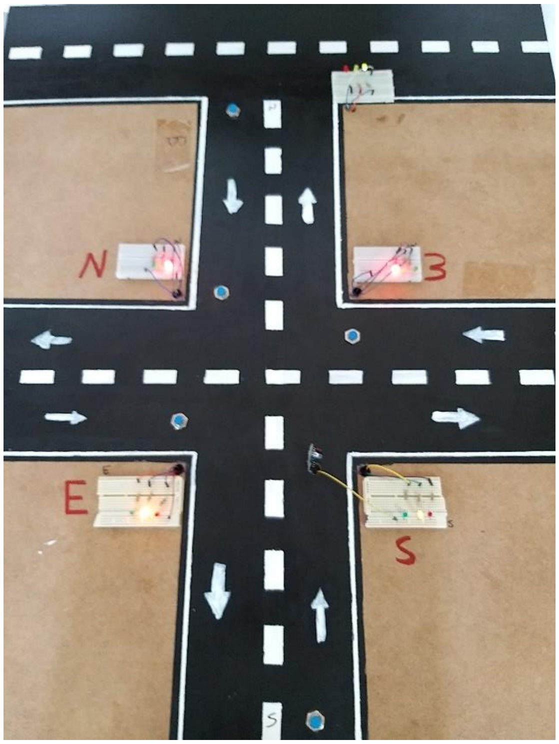


Fig 4.2 platform of the system

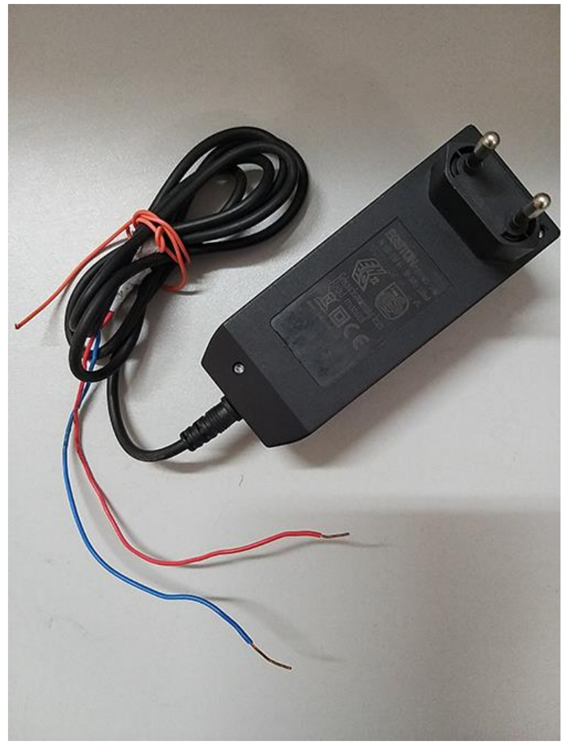


Fig 4.3 transformer

The last step is making Bluetooth Communication between master and slave Arduino.

First of all, we have to connect the Bluetooth module to an Arduino. Also, note that the Rx pin of the module goes to the Tx Pin of the Arduino and the Tx Pin of the module goes to the Rx pin of the Arduino mega as same as in Arduino Uno. Now Arduinos are ready for powering up. Before plugging in the USB cable, press the button on the Bluetooth module to enter the command mode. The next step is to open the Serial monitor to send some commands to the Bluetooth module. First, send the command AT to see if we can communicate with the module. The module returns OK. Then send the command AT+UART? To see the default baud rate as you can see it is set to 9600. Then send the command AT+ROLE? To find out if the module is in master or slave mode. The module returns 0 which means that the module is set at slave mode. Next, we need to find its address. We send the command AT+ADDR? And we get back the address of the module. We need this value later, now the slave Bluetooth module is ready as Fig 4.5 shown.

The configuration of the master shown in Fig 3.4 follows the same procedure but sets the module to master mode with the command AT+ROLE = 1. Then sends the command AT+CMODE=0, which means that the module will connect to a fixed address. The last step is to send the command AT+BIND = address of the slave module. If powering up the modules once more, after a few seconds, they are paired automatically, and the LEDs started flashing every two seconds.

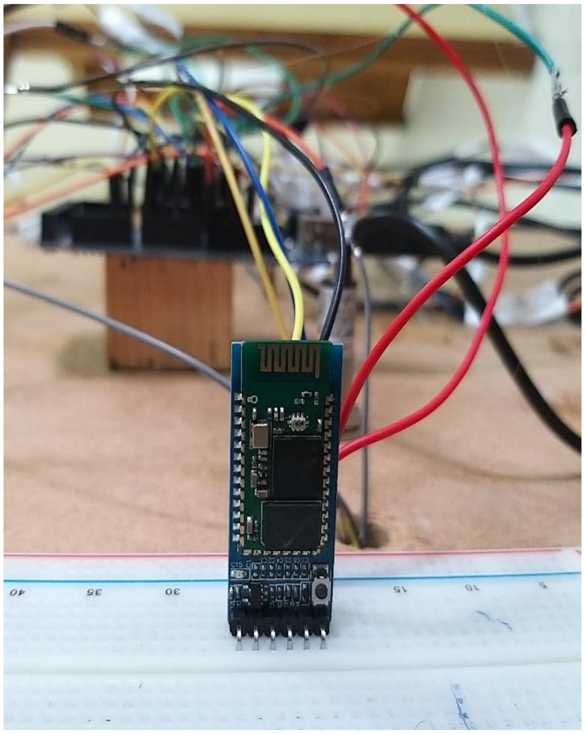


Fig 4.4 master Bluetooth

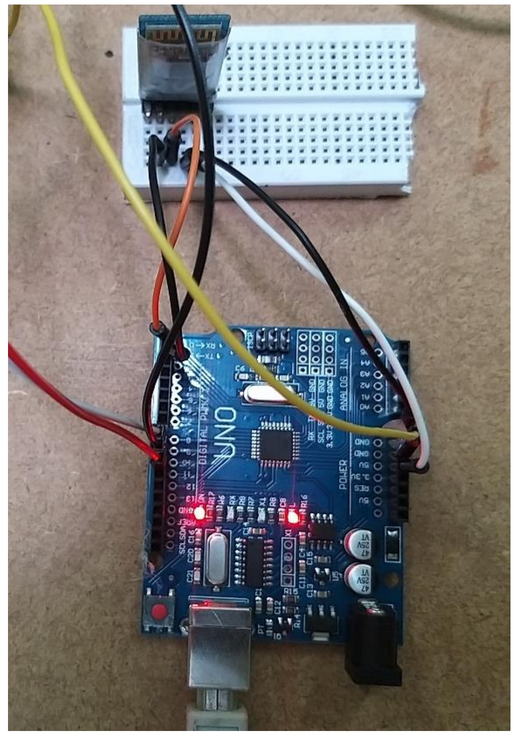


Fig 4.5 slave Bluetooth

The communication between the controllers (Master Arduino, Slave Arduino) has successfully been achieved, so when there is a traffic crisis on a street S, a signal will be sent to the slave, instructing it to turn on light green, and thus the traffic crisis problem has been solved.

Chapter 5 Validation & Testing

5.1 Overview

This chapter discusses the testing of all components of the system and the obtained results. Test all parts in order to insure that all functions are working efficiently and without errors.

5.2 Hardware Testing

This section discusses the testing of all components.

1. Testing proximity sensors:

First testing, we connected the proximity sensors to Digital Arduino Pins, which failed. Then we transmitted the Data pin of each sensor with an analog pin in Arduino, Vcc pin to (12v) on the power supply, and ground pins with Arduino ground, as shown in Fig 5.1 and Fig 5.2.

```
East1 counter = 0
East2 counter = 0
East = 0
 No  cars funded in the East
West1 counter = 0
West2 counter = 0
West = 0
 No  cars funded in the west
1023
North1 counter = 0
North2 counter = 0
North = 0
 No  cars funded in the North
1023
south1 counter = 0
south2 counter = 0
South = 0
 No  cars funded in the South
```

Fig.5.1 serial monitor reading values

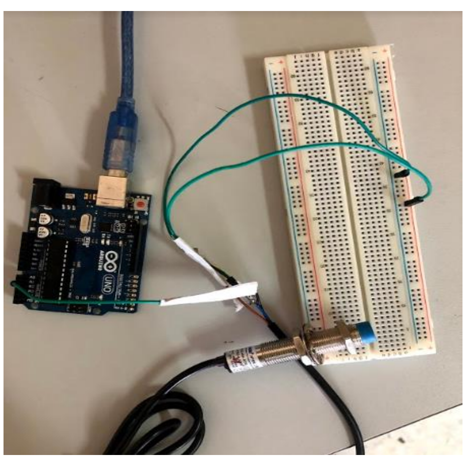


Fig 5.2 Sensor Testing

When some cars passing alongside sensors, it gives new readings as shown in Fig 5.3 and Fig 5.4

```
East1 counter = 4
East2 counter = 0
East = 4
West1 counter = 0
West2 counter = 0
West = 0
 No cERs funded in the west
North1 counter = 1
North2 counter = 0
North = 1
south1 counter = 8
south2 counter = 0
South = 8
```

Fig 5.3 counter value

Fig 5.4 sensor collect data

Problems that we occurred: each sensor has a different value of senses that cause test each separately to know how much reading value for each one.

2. Testing Lights

To test the Lights, the positive side of led connected with a resistor (100 ohms) then to a digital Arduino pin and the negative side with ground pins in Arduino to check if led takes orders from Arduino as shown in Fig 5.5

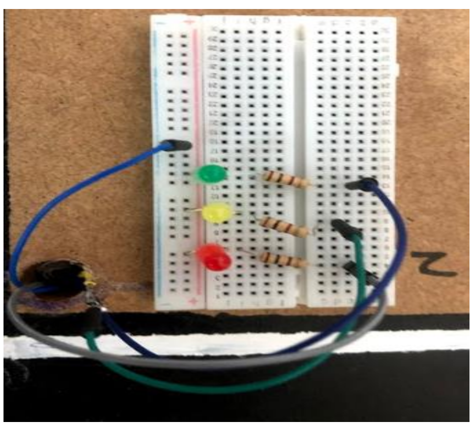


Fig 5.5 test LEDs

After sending orders to LEDs by Arduino to examine if they take that into correctly and orderly or not as shown in Fig 5.6

Problems that occurred: each light needs a test alone with its wires and Arduino pins to insure if it gets a correct order and light on for its specific time.

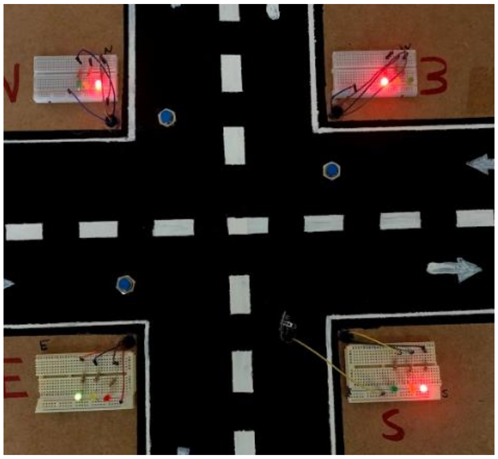


Fig 5.6 testing all lEDs

3. Testing Bluetooth Module.

To test the two Bluetooth modules, the first one connected to Master Arduino, as shown in Table 5.1and Fig 5.7 , the second one is connected to the slave Arduino which Takes orders from the master Arduino as shown in Table 5.2 and Fig 5.8.

1st Bluetooth	Master Arduino
Rx	Tx
Tx	Rx
VCC	5v
GND	GND

Table 5.1

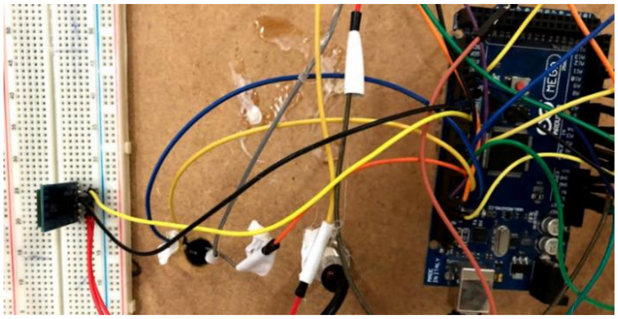


Fig 5.7

2nd Bluetooth	Slave Arduino
Rx	Tx
Tx	Rx
VCC	5v
GND	Gnd

Table 5.2

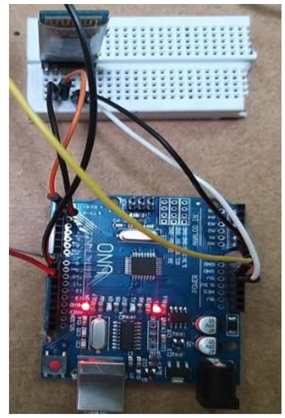


Fig5.8

By using AT commands, to examine if they connected and communicate with each other. At the end, the process succeeded and Master Arduino can send orders to Slave one as shown in Fig 5.9

```
W
value1=0
East1 counter = 2
East2 counter = 0
East = 2
West1 counter = 1
West2 counter = 0
West = 1
1023
North1 counter = 0
North2 counter = 0
North = 0
 No cERs funded in the North
1023
south1 counter = 6
south2 counter = 0
South = 6
Bluetooth connected
```

Fig 5.9 serial monitor Bluetooth connect

Problems that occurred: how to configure each one of the Bluetooth modules as master or slave, and how to make safe communication between each other.

5.3 System Test

After ensuring that all the parts are working properly, started assembling and integrating the parts for making the system ready to operate.

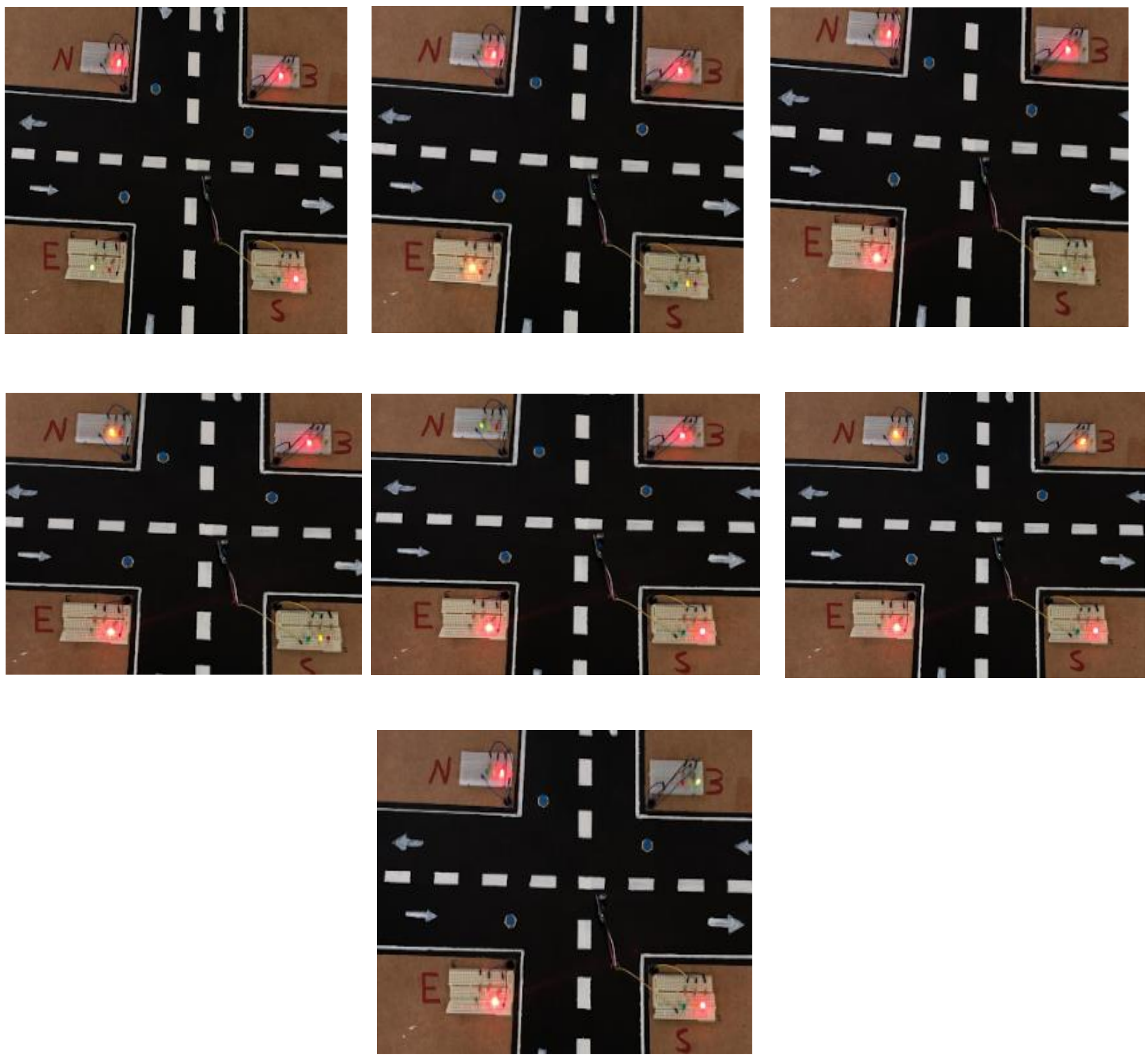
5.3.1 Test cases

This system contains 24 cases. During the system testing process, we will test any case for making sure that the system works properly.

Notice: If whole streets have the same number of cars, lights will work normally.

After successful testing of 24 cases, we found that the system succeeded in dealing with the problem of traffic crisis.

Example: case (E, S, N, W)



In a previous example, streets order by count of cars in each street. at first east street light green for 10 second, then east and south street light Yellow, then south street light green …

5.3.2 Test Bluetooth communication

When the south street has the greatest number of vehicles, the distant street will change its red light to yellow followed by green for crossing cars and avoid congestion as shown in Fig 5.11 and Fig 5.12

```
value1=0
East1 counter = 3
East2 counter = 0
East = 3
West1 counter = 1
West2 counter = 0
West = 1
1023
North1 counter = 0
North2 counter = 0
North = 0
 No cERs funded in the North
1023
south1 counter = 14
south2 counter = 0
South = 14
Bluetooth connected
```

Fig 5.11 serial monitor values

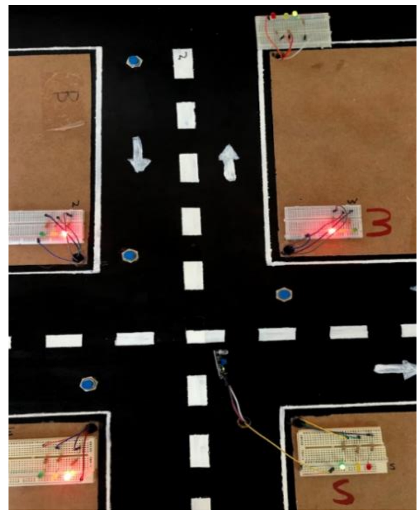


Fig 5.12 Bluetooth connection is successfully

Chapter 6 : Conclusion & Future work

6.1: Overview

This chapter, will conclude final result and future work of project.

6.2: Final Result:

The system will be able to know the number of cars in each street, either after entering or leaving and then making the appropriate decision to give priority to the occupied street. This system knows the number of cars and gives priority for each crossroads based on the highest value, while giving the priority, it calculates the number of vehicles then, after completing the first case it performs a new update on the values so thus moving to another case or staying in the same case based on the available values and so in, also this system communicate with the bystreet to be opened to avoid traffic congestion while crossing cars

6.3: Future Work

A sensor will be added at the beginning of each street from the other side in order to communicate effectively and more with the rest of the streets after the passage of the vehicles

An option will be added to give priority to ambulances or the police to give orders to open roads so as not to be late or endanger people's lives

6.4: Conclusion

The main purpose of this system is achieved, so the traffic crisis is reduced, building a multiple smart traffic light system is a complicated issue because there are many cases to be taken, such as pedestrians on the road.

This project solved a crisis problem and achieved communication between two traffic light systems. This system hopes to solve additional problems, such as the sudden crossing of emergency vehicles.

References

[1],freepik, @vectorpocket,2019. “freepik.com”. accessed [8Dec2019]. https://www.freepik.com/free-vector/modern-metropolis-street-life-cartoon-vector-concept-with-people-hurrying-business-city-street-townsfolk-walking-sidewalk-pedestrians-passing-crossroads-transport-moving-road-illustration_4997276.htm.

[2] Traffic’s Mind-Boggling Economic Toll, SCHNEIDER, FEBRUARY 7, 2018.”citylab.com”.accessed[2Sep2019].https://www.citylab.com/transportation/2018/02/traffics-mind-boggling-economic-toll/552488/

[3] Arduino Mega.”Arduino.com”.accessed [15 Oct 2019]. https://www.arduino.cc/en/Main/arduinoBoardMega/.

[4] Ultrasonic sensor. “components101.com”.accessed [15 Oct 2019]. https://components101.com/ultrasonic-sensor-working-pinout-datasheet.

[5] IR Sensor Module for Remote Control Systems. 01-Jan-2019.” vishay.com". Accessed [19 Oct 2019].http://www.vishay.com/docs/82486/tsmp58138.pdf

[6] Chip Immune Inductive Prox.” http://www.allcomponents.ru/”. accessed [1 Dec 2019].http://www.allcomponents.ru/omron/e2ez.htm

[7] Bluetooth Sensor Module. July 24, 2018.” techzeero.com”.Accessed [19 Oct 2019]. https://techzeero.com/sensors-modules/bluetooth-module-hc-05/

[8]ESP8266 NodeMCU WiFi Devkit.” handsontec.com”. Accessed [19 Oct 2019].https://www.handsontec.com/pdf_learn/esp8266-V10.pdf

[9] Arduino to Arduino Serial Communication. Case, JennWed, 2013-02-06 13:10.” robotic-controls.com”. Accessed [19 Oct 2019] “https://robotic-controls.com/learn/arduino/arduino-arduino-serial-communication

[10] XBee S2C – RF Module.Accessed [19 Oct 2019].” components101.com” https://components101.com/wireless/xbee-s2c-module-pinout-datasheet

[11] freepik, @vectorpocket,2017. “freepik.com”. Accessed [8Dec2019]. https://www.freepik.com/free-vector/street-crossroad-traffic-lane-pedestrian-crossing-crosswalk_2238511.htm#page=1&query=%20cross%20road&position=34

[12] What is an Inductive Proximity Sensor?. Accessed [10 Oct 2019].https://www.keyence.com/ss/products/sensor/sensorbasics/proximity/info/

[13] Bubble Sort. chitranayal, SumitBM, rathbhupendra.” geeksforgeeks.org”. Accessed [15Dec2019].www.geeksforgeeks.org/bubble-sort

[14]Waze.”waze.com”.Accessed[31Dec2019]. https://www.waze.com/ar