Smart Traffic Controlling System

By

Sk. Minhajul Abedin -18EEE051

Hasiba Hasbi Shaily – 18EEE059

Md. Shahriar Reza – 17EEE024

A project report submitted in partial fulfillment of the requirements for the completion of the degree of Bachelor of Science in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering

Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj-8100, Bangladesh

Declaration

This is to certify that the project work entitled "Smart Traffic Controlling System" has been carried out by Sk. Minhajul Abedin ,Hasiba Hasbi Shaily and Md. Shahriar Reza in the Department of Electrical and Electronic Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj, Bangladesh. It is hereby declared that the above project work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

Signature of the candidate

Date:

Signature of the Supervisor

Date:

Acknowledgement

Primarily, we would thank God for being able to complete this project with success. Then we would like to thank our supervisor **Dr. A.T.M. Saiful Islam**, Associate Professor, Dept. of EEE, whose valuable guidance has been the ones that helped us patch this project and make it full proof success as his suggestion and his instructions has served as the major contributor towards the completion of the project.

Sk. Minhajul Abedin

Hasiba Hasbi Shaily

Md. Shahriar Reza

8th February,2023

Abstract

With the tremendous technological progress and the widespread use of a variety of technologies, we note how smart cities are providing services efficiently by using technologies. The aim of this project is to build a Smart Traffic Control System (STCS) to facilitate and optimize traffic flow, minimize traffic congestion, and reduce the waiting time by detecting the density on each street.

Contents *PAGE*

CHAPTER 1 Introduction	6
1.1 Problem Definition	6
1.2 Objectives	6
CHAPTER 2 Literature Review	7-8
CHAPTER 3 Materials	9-14
3.1 Arduino Mega 2560	9
3.2 Arduino Uno R3	11
3.3 Ultrasonic Sensor	13
3.4 LDR Sensor Module	14
CHAPTER 4 Project Construction	15-18
CHAPTER 5 Methodology	19-20
5.1 Testing	19
CHAPTER 6 Arduino Programming	21-27
CHAPTER 7 Conclusion and Future Work	28
7.1 Limitation	28
Reference	29

Chapter 1

Introduction

Recently, technology represents an integral part of our life and it is impractical to imagine our life without embedded devices [1]. Technology changes the lifestyle to be more efficient and reliable [2]. The scientific development witnessed by the modern world in the technological industries, which took the form of innovation and ease of dealing, has a significant impact on different areas, especially in smart cities. An example of that is the use of a driverless minibus service, a pressure management system to leverage water network data and lighting controls. Smart cities distinguished by using high technologies to facilitate life for citizens [3]. The purpose of smart cities is to improve the quality of life, increase the level of development, increase economic growth and harnessing technology, especially technology that leads to efficient outcomes [4]. One of the problems we are trying to solve here is controlling the traffic congestion. Thus, we are working on building a prototype system for solving the traffic congestion issue. Throughout the literature review, the recommended solution was to design a density-based traffic signal system, for making traffic flow more efficient. Our aim is to facilitate and optimize traffic flow, minimize traffic congestion, and reduce the waiting time.

1.1 PROBLEM DEFINITION

Traffic congestion is a serious issue in several big cities worldwide. Traffic can be controlled in the main junctions by including either an automatic traffic light control system or by traffic police. However, the conventional traffic light system is based on a static/fixed time concept dedicated to each street regardless of the diversity of traffic density. Sometimes, the most congested road has to be prioritized, in terms of a number of vehicles. Thus, we propose a developed design that incorporates density as a factor that affects the traffic light system.

1.2 OBJECTIVE

The objective of our project is to regulate the traffic flow by means of introducing sensors at traffic lights on the main roads as a Smart Traffic Control System (STCS). Based on these sensors, the streets' density is detected, traffic monitoring, and automatic control of the traffic light to reduce traffic congestion. The sensors calculate the traffic density on a particular road and change the priority of traffic light based on the cars' numbers waiting on the same road thus reduce the waiting time.

Chapter 2

Literature Review

Traffic Congestion is a persistent issue in several cities worldwide. Traffic lights play a vital role in traffic safety. Even though road safety is important [5] and even though we are in the 4th industrial revolution era, unfortunately, the current traffic system still has many problems since it's based on a fixed time concept dedicated to each street (side of the junction) regardless of the diversity of traffic density. Traffic delay has many side effects on daily life. To a commuter or a traveler, congestion means wasting time, wasting opportunities, and frustration. In addition, for business providers, congestion means losing trade opportunities, delivery delays, and increasing costs [6]. These issues need to be solved by developing automated transportation management systems. Many pieces of research and studies have been conducted that provided different hypotheses and solutions. Lo and Chow explained the Dynamic Intersection Signal Control Optimization (DISCO) system which is a platform for modeling dynamic traffic in Hong Kong [7]. Ref. [8] provided a traffic light control system based on Wireless Sensor Networks (WSN) that has the ability to revolutionize control technology and traffic surveillance due to its potential for large scale deployment and low cost. Ref. [9] designed an HMS Algorithm based on the Modified Round Robin algorithm. HMS reveals the enhancement of traffic light services in Iraq where it reduced waiting time and ensure the maximum use of roads. Ref. [7] explained another solution that used multiple IR sensors that had been installed on roads for density measurement. The increases in the number of IR sensors have improved the accuracy of density measurement. Moreover, it was explained that the traffic lights that are connected with the output port of the IR sensors and Arduino processor were linked with an input port of the Arduino processor through the feedback path. A system such as this acts as an interfacing device which supports the Lab VIEW software and better decision making on road base upon traffic density. Finally, the traffic light control system using Arduino is a system designed to make traffic lights more flexible, so that it can measure the proportion of traffic density, a central device where all the traffic lights are linked, it can be controlled remotely and it changes its scheduling without the need to change it manually. A Comparison between some traffic control systems is shown in Table 1.

Table 1. Comparison between systems**.**

Name	Objective	Tools/technique	Reference
Intelligent Traffic Light Flow Control System	Handle the case of controlling traffic over multiple intersections.	WSN Traffic system communication algorithm (TSCA) Traffic signals time manipulation algorithm (TSTMA)	[8]
IoT Based Traffic Signaling System	Analyze the traffic congestion as Heavy traffic and Normal Traffic with date and time based on the data that has been sent wirelessly to Raspberry Pi3 and updated on Cloud webpage which can be used for further planning and analysis by the Traffic department	Ultrasonic Sensor ESP8266 Wi-Fi Module Arduino Microcontroller (Arduino Uno can be programmed with Arduino Software IDE). Raspberry Pi3. Cloud webpage Cloud server.	[10]
Smart Autonomous Traffic Light Switching by Traffic Density Measurement through Sensors	Enable dynamic switching of traffic light and timing based on traffic density. Reduces the Average Trip Waiting Time (ATWT).	Infrared sensor Low power Energy Microcontroller Comparator Radio Communication module	[11]

Next, the methodology of conducting this study will be explained.

Chapter

3

MATERIALS

3.1 Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. 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

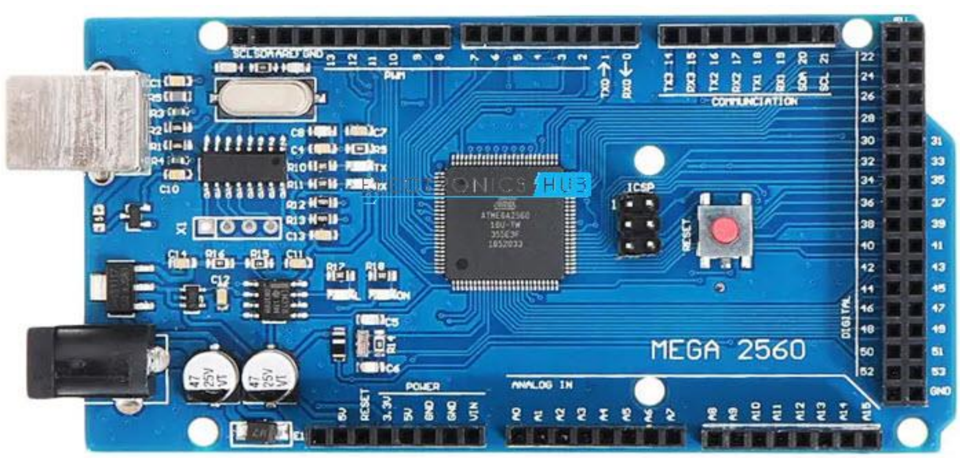

Figure 3.1: Arduino Mega 2560 Board

Features & Parameters:

Features:

1. **High Endurance Non-volatile Memory Segments**
    - Write/Erase Cycles: 10,000 Flash
2. **Atmel QTouch library support**
3. **JTAG (IEEE std. 1149.1 compliant) interface**

4. **Peripheral Features**
    - Real-time Counter with Separate Oscillator
    - Programmable Watchdog Timer with Separate On-chip Oscillator
    - On-chip Analog Comparator
    - Interrupt and Wake-up on Pin Charge
5. **Other special features**
    - Power-on Reset and Programmable Brown-out Detection
    - Internal Calibrated Oscillator
    - External and Internal Interrupt Sources
    - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

Parameters:

Name	Value
Program Memory Type	Flash
Program Memory Size (KB)	256
CPU Speed (MIPS/DMIPS)	16
SRAM (KB)	8,192
Data EEPROM/HEF (bytes)	4,096
Digital Communication Peripherals	4-UART, 5-SPI, 1-I2C
Capture/Compare/PWM Peripherals	4 Input Capture, 4 CCP, 16PWM
Timers	2 x 8-bit, 4 x 16-bit
Number of Comparators	1
Temperature Range (°C)	-40 to 85
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	100

3.2 Arduino Uno R3

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), 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 a AC-to-DC adapter or battery to get started. You can tinker with your Uno without worrying too much about doing something wrong, worstcase scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

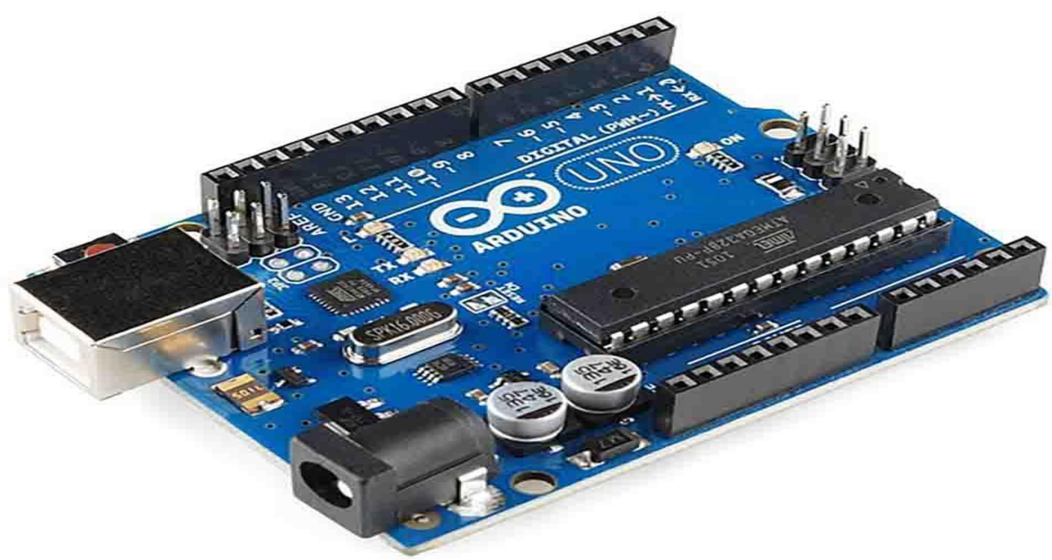

Figure 3.2:Arduino Uno R3.

All specification:

MICROCONTROLLER	ATmega328P
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	14 (of which 6 provide PWM output)
PWM DIGITAL I/O PINS	6
ANALOG INPUT PINS	6
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA
FLASH MEMORY	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	68.6 mm
WIDTH	53.4 mm
WEIGHT	25 g

3.3 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is D = ½ T x C (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be:

D = 0.5 x 0.025 x 343

Figure 3.3:Ultrasonic Sensor.

3.4 LDR Sensor Module

LDR sensor module is used to detect the intensity of light. It is associated with both analog output pin and digital output pin labelled as AO and DO respectively on the board. When there is light, the resistance of LDR will become low according to the intensity of light. The greater the intensity of light, the lower the resistance of LDR. The sensor has a potentiometer knob that can be adjusted to change the sensitivity of LDR towards light.

Specification:

1.Input Voltage: DC 3.3V to 5V

2.Output: Analog and Digital

3.Sensitivity adjustable

Figure 3.4 :LDR Sensor Module.

Chapter 4

Project Construction

We have 2 major parts for this project.

1.Traffic Controlling System(TCS)

2.Smart Lighting

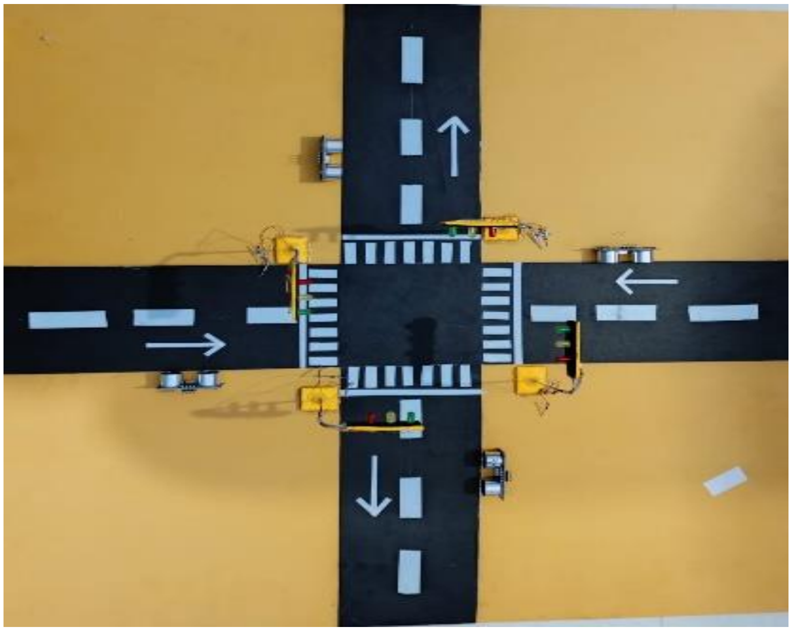

Figure 4.1 :Road Structure of the project.

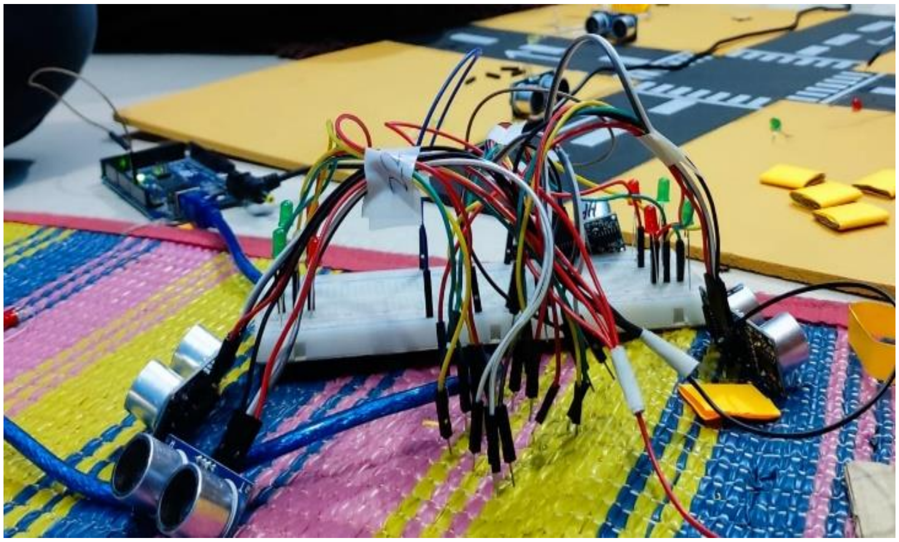

Figure 4.2 :Breadboard Connection for Traffic Light.

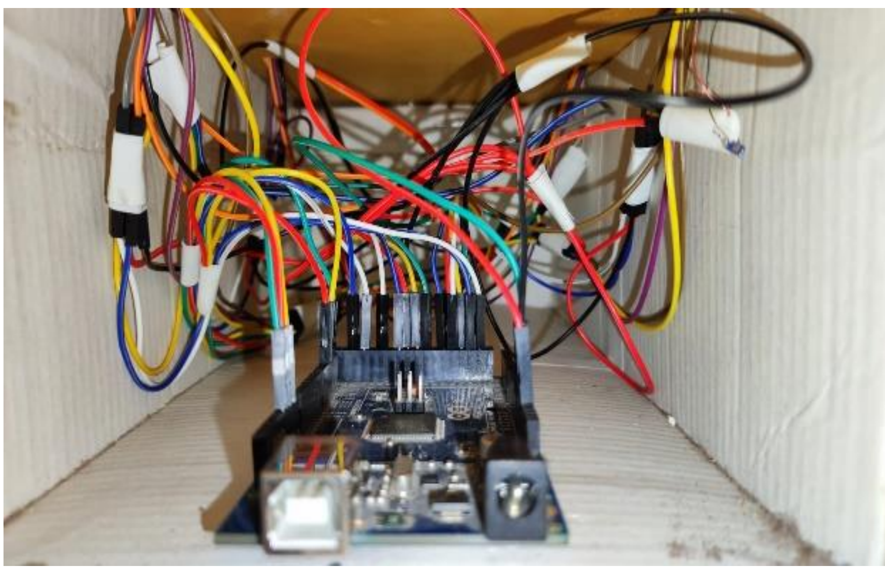

Figure 4.3 :Arduino Mega Connection.

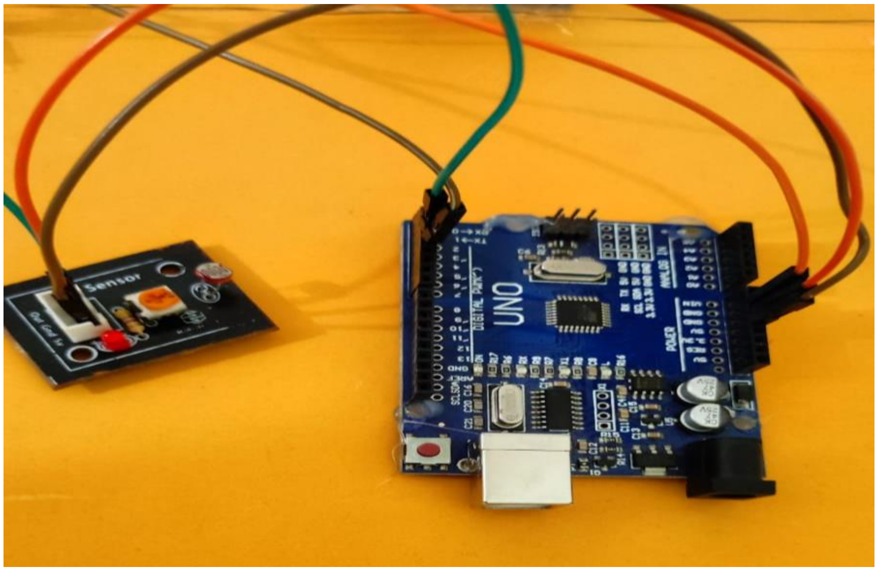

Figure 4.4 :LDR Sensor Module with Arduino Uno.

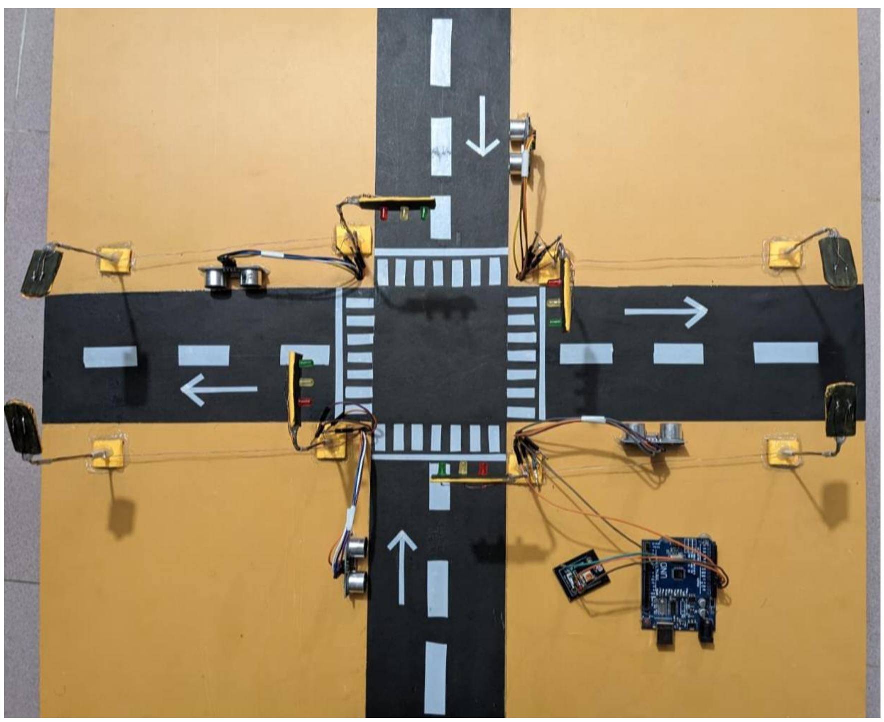

Figure 4.5 :Final Project View.

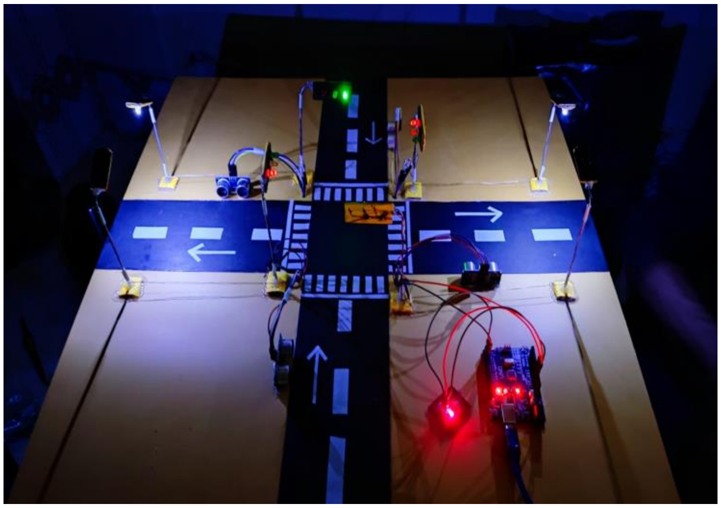

Figure 4.6 :Night View of Project.

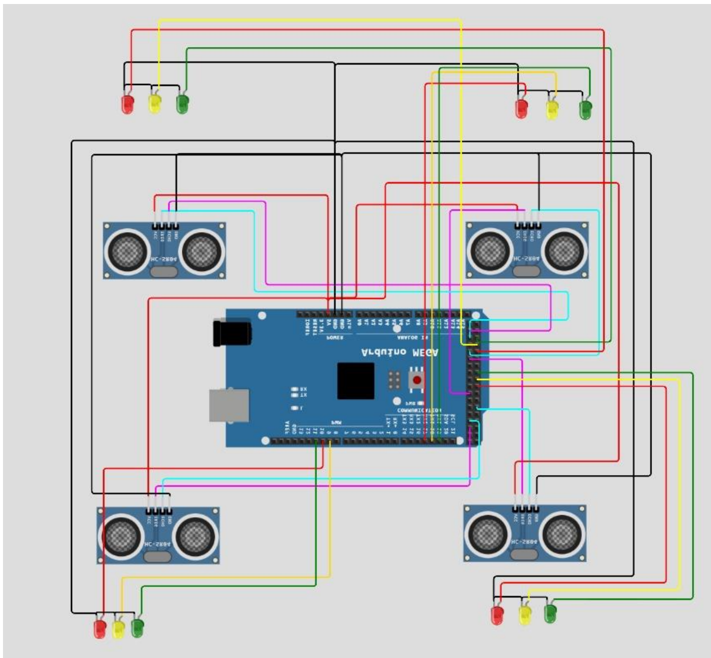

Figure 4.7:Circuit Diagram.

Chapter 5

Methodology

The smart traffic management systems help to detect the congestion area and according to it, it helps to reduce congestion. For this action, it uses sensors data to analyze and synchronize on a real-time basis. Traffic light blinks on the basis of congestion on given input data. Using this system it can reduce pollution and increase safety from road accidents.

In this system, a pairs of sensors are placed on each side of road and four traffic lights. Arduino Microcontroller controls the ultrasonic and counts the number of vehicles passing on the road where the signals are changed when the sensor assumes that there is traffic congestion on the road. There are 3 cases in this system as follows:

1. If there are no vehicles present on the road then light is Red (except one), until vehicles are arrived.

2. If there is varying number of the traffic density at signals, the system changes the priority of traffic light GREEN for the road that have the highest density.

3. In the case of all roads having an equal density, its actives the status of the sequential arrangement between the roads and the system will work normally by controlling the signals one by one.

5.1 Testing

In order to run the whole system, we needed:

• Led traffic light module: That showed the lights red, yellow and green

• Arduino Microcontroller: Physical programmable circuit board that contains the code from the Arduino IDE software [20]

• Wires male-female: A group of electrical wires in a cable, with a connector or pin at each end

• Breadboard: A construction tool base for the prototyping of electronics used to design and test the circuit

• Arduino Adaptor: Used to power the Arduino board via the plug from a wall outlet.

• Laptop

Then, we applied functional testing for evaluating the system and verifying each function of the system, to check if it conforms to the required specification, the tests include: unit testing, integration testing, compatibility testing and system testing. Unit Testing is done during the development (coding phase) of an application by the developers [21]. The unit testing was applied to all cases as described. Integration testing: a complete system is tested to guarantee that the requirements of the software have been met [22]. During the system testing we were focusing on validating the proper work complete system after the integration. four cases have been coved as follows:

• By default, all traffic lights are in red (except one), until vehicles are present

• In the case of a denser road, it has the priority to turn the traffic light into (green)

• In the case of all roads having an equal density, it activates the status of the sequential arrangement between the roads

• In the case of two roads with the same density, the priority is to turn the traffic light green to the road who started congesting first

Chapter 6

Arduino Programming

```
#include <TimerOne.h>
#define RST_PIN 5    // Configurable, see typical pin layout above
// Red,yellow,green
int signal1[] = { 10, 9, 11 };   // connect signal lights
int signal2[] = { 37, 39, 41 };  // connect signal lights
int signal3[] = { 47, 49, 51 };  // connect signal lights
int signal4[] = { 17, 18, 19 };  // connect signal lights
int i;
// delay
int redDelay = 8000;
int yellowDelay = 3500;
// trigger and echo pins
volatile int triggerpin1 = 24;  // ultrasonic signal out pin
volatile int triggerpin2 = 34;
volatile int triggerpin3 = 44;
volatile int triggerpin4 = 50;
volatile int echopin1 = 26;  // ultrasonic signal receiving pin
volatile int echopin2 = 31;
volatile int echopin3 = 46;
volatile int echopin4 = 52;
volatile long time;           // Variable for storing the time traveled
volatile int S1, S2, S3, S4;  // Variables for storing the distance covered

int t = 5;  // distance under which it will look for vehicles.

void setup() {
```

```
  Serial.begin(9600);
  Timer1.initialize(700000);          //Begin using the timer. This function must be called first. "microseconds" is the period of time the timer takes.
  Timer1.attachInterrupt(softInterr);  //Run a function each time the timer period finishes.

  // Declaring LED pins as output
  for (int i = 0; i < 4; i++) {
    pinMode(signal1[i], OUTPUT);
    pinMode(signal2[i], OUTPUT);
    pinMode(signal3[i], OUTPUT);
    pinMode(signal4[i], OUTPUT);
  }
  // Declaring ultrasonic sensor pins as output
  pinMode(triggerpin1, OUTPUT);
  pinMode(echopin1, INPUT);
  pinMode(triggerpin2, OUTPUT);
  pinMode(echopin2, INPUT);
  pinMode(triggerpin3, OUTPUT);
  pinMode(echopin3, INPUT);
  pinMode(triggerpin4, OUTPUT);
  pinMode(echopin4, INPUT);
}


//  driver code
void loop() {
  // If there are vehicles at signal 1
  if (S1 < t) {
    signal1Function();
  }
  // If there are vehicles at signal 2
  if (S2 < t) {
    signal2Function();
  }
  // If there are vehicles at signal 3
  if (S3 < t) {
    signal3Function();
  }
  // If there are vehicles at signal 4
  if (S4 < t) {
    signal4Function();
  }
}
// This is interrupt function and it will run each time the timer period finishes. The timer period is set at 100 milli seconds.
void softInterr() {
  // Reading from first ultrasonic sensor
  digitalWrite(triggerpin1, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin1, HIGH);
```

```
delayMicroseconds(10);
digitalWrite(triggerpin1, LOW);
time = pulseIn(echopin1, HIGH);
S1 = time * 0.034 / 2;
// Reading from second ultrasonic sensor
digitalWrite(triggerpin2, LOW);
delayMicroseconds(2);
digitalWrite(triggerpin2, HIGH);
delayMicroseconds(10);
digitalWrite(triggerpin2, LOW);
time = pulseIn(echopin2, HIGH);
S2 = time * 0.034 / 2;
// Reading from third ultrasonic sensor
digitalWrite(triggerpin3, LOW);
delayMicroseconds(2);
digitalWrite(triggerpin3, HIGH);
delayMicroseconds(10);
digitalWrite(triggerpin3, LOW);
time = pulseIn(echopin3, HIGH);
S3 = time * 0.034 / 2;
// Reading from fourth ultrasonic sensor
digitalWrite(triggerpin4, LOW);
delayMicroseconds(2);
digitalWrite(triggerpin4, HIGH);
delayMicroseconds(10);
```

```
  digitalWrite(triggerpin4, LOW);
  time = pulseIn(echopin4, HIGH);
  S4 = time * 0.034 / 2;
}

void signal1Function() {
  Serial.println("signal 1");
  low();
  // Make RED LED LOW and make Green HIGH for 5 seconds
  digitalWrite(signal1[0], LOW);
  digitalWrite(signal1[2], HIGH);
  digitalWrite(signal3[0], LOW);
  digitalWrite(signal3[2], HIGH);
  delay(redDelay);
  // if there are vehicels at other signals
  if (S2 < t || S3 < t || S4 < t) {
    // Make Green LED LOW and make yellow LED HIGH for 2 seconds
    digitalWrite(signal1[2], LOW);
    digitalWrite(signal1[1], HIGH);
    digitalWrite(signal3[2], LOW);
    digitalWrite(signal3[1], HIGH);
    delay(yellowDelay);
  }
}
```

```
void signal2Function() {
  Serial.println("signal 2");
  low();
  digitalWrite(signal2[0], LOW);
  digitalWrite(signal2[2], HIGH);
  digitalWrite(signal4[0], LOW);
  digitalWrite(signal4[2], HIGH);
  delay(redDelay);
  if (S1 < t || S3 < t || S4 < t) {
    digitalWrite(signal2[2], LOW);
    digitalWrite(signal2[1], HIGH);
    digitalWrite(signal4[2], LOW);
    digitalWrite(signal4[1], HIGH);
    delay(yellowDelay);
  }
}

void signal3Function() {
  Serial.println("signal 3");
  low();
  digitalWrite(signal3[0], LOW);
  digitalWrite(signal3[2], HIGH);
  digitalWrite(signal1[0], LOW);
  digitalWrite(signal1[2], HIGH);
  delay(redDelay);
```

```
  if (S1 < t || S2 < t || S4 < t) {
    digitalWrite(signal3[2], LOW);
    digitalWrite(signal3[1], HIGH);
    digitalWrite(signal1[2], LOW);
    digitalWrite(signal1[1], HIGH);
    delay(yellowDelay);
  }
}

void signal4Function() {
  Serial.println("signal 4");
  low();
  digitalWrite(signal4[0], LOW);
  digitalWrite(signal4[2], HIGH);
  digitalWrite(signal2[0], LOW);
  digitalWrite(signal2[2], HIGH);
  delay(redDelay);
  if (S1 < t || S2 < t || S3 < t) {
    digitalWrite(signal4[2], LOW);
    digitalWrite(signal4[1], HIGH);
    digitalWrite(signal2[2], LOW);
    digitalWrite(signal2[1], HIGH);
    delay(yellowDelay);
  }
}
```

```
// Function to make all LED's LOW except RED one's.
void low() {
  for (int i = 1; i < 3; i++) {
    digitalWrite(signal1[i], LOW);
    digitalWrite(signal2[i], LOW);
    digitalWrite(signal3[i], LOW);
    digitalWrite(signal4[i], LOW);
  }
  for (int i = 0; i < 1; i++) {
    digitalWrite(signal1[i], HIGH);
    digitalWrite(signal2[i], HIGH);
    digitalWrite(signal3[i], HIGH);
    digitalWrite(signal4[i], HIGH);
  }
}
```

Chapter

7

Conclusion and Future Work

With the rapid increase in motorization industry, urbanization, population growth, and changes in population density, the problem of traffic congestion has increased worldwide. Therefore, we built an STCS with the aim of regulating traffic flow and reducing traffic congestion. This will lead to providing a good service for the community. We wish to succeed in the future by applying this smart system nationally and internationally.

- The system can be replaced by image processing system which will give efficient results.

7.1 Limitation

- Ultrasonic Sensor work only for fewer distance.
- Detects any obstacle as a vehicle.
- Continuous power supply needed.

References

[1] S. Seiler, R. Sell, and D. Ptasik, "Embedded System and Robotic Education in a Blended Learning Environment Utilizing Remote and Virtual Labs in the Cloud, Accompanied by â?? Robotic HomeLab Kitâ??," International Journal of Emerging Technologies in Learning (iJET), vol. 7, no. 4, pp. 26-33, 2012. https://doi.org/10.3991/ijet.v7i4.2245

[2] S. A. Garba, Y. Byabazaire, and A. H. Busthami, "Toward the Use of 21 st Century Teaching-Learning Approaches: The Trend of Development in Malaysian Schools within the Context of Asia Pacific," International Journal of Emerging Technologies in Learning, vol. 10, no. 4, 2015. https://doi.org/10.3991/ijet.v10i4.4717

[3] R. P. Dameri, "Searching for smart city definition: a comprehensive proposal," International Journal of computers & technology, vol. 11, no. 5, pp. 2544-2551, 2013. https://doi.org/10.24297/ijct.v11i5.1142

[4] "What is Smart City." http://smartcities.gov.in/content/innerpage/what-is-smart-city.php (accessed 20/09/2019.

[5] T. A. Alhmiedat, F. Omar, A. A. Taleb, and A. Alsswey, "Road safety and energy saving proposed system: A ZigBee WSN approach," International Journal of Online and Biomedical Engineering (iJOE), vol. 11, no. 2, pp. 55-59, 2015. https://doi.org/10.3991/ij oe.v11i2.4430

[6] E. F. B. Poyen, A. K. Bhakta, B. D. Manohar, I. Ali, A. ArghyaSantra, and A. Rao, "Density based traffic control," International Journal of Advanced Engineering, Management and Science, vol. 2, no. 8, p. 239611, 2016.

[7] Arduino. https://www.arduino.cc (accessed.

[8] D. Jost. "FierceElectronics." https://www.fierceelectronics.com/sensors/what-ultrasonicsensor (accessed 26/03/2020.

[9] LabCenter. "PCB Design and Circuit Simulator Software " https://www.labcenter.com (accessed).

[10] Rajkumar. "What Is Software Testing Definition, Types, Methods, Approaches." https://www.softwaretestingmaterial.com/software-testing/ (accessed).

[11] I. Sommerville, Software Engineering GE. Pearson Australia Pty Limited, 2016.