AY 2022-23
PROJECT REPORT
ON

Density based traffic light controller using Arduino

Submitted for

ECL305: PRACTICUM-III

By
DIVYANSH SRIVASTAVA (21216)
III SEMESTER
BTECH ECE

SCHOOL OF ELECTRONICS

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY UNA
HIMACHAL PRADESH

DECEMBER 2022

BONAFIDE CERTIFICATE

This is to certify that the project titled Density based traffic light controller using arduino is a bonafide record of the work done by

DIVYANSH SRIVASTAVA (21216)

in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in ELECTRONICS AND COMMUNICATION ENGINEERING of the INDIAN INSTITUTE OF INFORMATION TECHNOLOGY UNA, HIMACHAL PRADESH, during the year 2021 - 2025.

under the guidance of

Mr. Satish Kumar

Project viva-voce held on: _____

Internal Examiner External Examiner

ORIGNALITY / NO PLAGARISM DECLARATION

I certify that this project report is our original report and no part of it is copied from any published reports, papers, books, articles, etc. We certify that all the contents in this report are based on our personal findings and research and we have cited all the relevant sources which have been required in the preparation of this project report, whether they be books, articles, reports, lecture notes, and any other kind of document. We also certify that this report has not previously been submitted partially or as whole for the award of degree in any other university in India and/or abroad.

We hereby declare that, we are fully aware of what constitutes plagiarism and understand that if it is found at a later stage to contain any instance of plagiarism, our degrees may be cancelled.

DIVYANSH SRIVASTAVA (21216)

ABSTRACT

Traffic lights allow everyone to cross the intersection point one by one, reducing conflicts between vehicles entering intersection points from different directions. It provides road safety, also helps to solve traffic in simple manners. This project is a simple three-way version of the traffic light controller using Arduino and very few components. It has a circuit diagram and the code for the traffic controller system. Traffic light controller helps to manage the traffic and to maintain proper traffic management. These systems are placed at the intersections of the road or at the crossings to avoid congestions and accidents. The systems indicate to the driver by using different colours of light. Therefore it is simple to avoid congestion at the intersections.The main purpose of this project is, if there will be no traffic on the other signal, one shouldn't wait for that signal. The system will skip that signal and will move on the next one.Arduino is the main part of this project and it will be used to read from ultrasonic sensor HC-SR04 and calculate the distance. This distance will tell us if any vehicle is near the signal or not and according to that the traffic signals will be controlled.

The working of the project is divided into three steps:

- If there is traffic at all the signals, then the system will work normally by controlling the signals one by one.
- If there is no traffic near a signal, then the system will skip this signal and will move on to the next one. For example, if there is no vehicle at signal 2, 3 and currently the system is allowing vehicles at signal 1 to pass. Then after signal 1, the system will move on to signal 4 skipping signal 2 and 3.
- If there is no traffic at all the 4 signals, system will stop at the current signal and will only move on the next signal if there will be traffic at any other signal.

ACKNOWLEDGEMENT

I would like to thank the following people Mr. Didrikhya Uzir, Mr. Kumar Deepak, Mr. Satpal for their support and guidance without whom the completion of this project in fruition would not be possible.

We would like to express our sincere gratitude and heartfelt thanks to Mr. Satish Kumar for their unflinching support and guidance, valuable suggestions and expert advice. Their words of wisdom and expertise in subject matter were of immense help throughout the duration of this project.

We also take the opportunity to thank our Director and all the faculty of School of Computing/Electronics, IIIT Una for helping us by providing necessary knowledge base and resources.

We would also like to thank our parents and friends for their constant support.

DIVYANSH SRIVASTAVA (21216)

TABLE OF CONTENTS

LIST OF ACRONYMS

ARDUINO a popular open-source electronics platform

DBCTL Density Based Traffic Light Controller

LED Light-Emitting Diode

LIST OF FIGURES

Chapter 1
Introduction

A density based traffic light controller system is a type of traffic management system that uses sensors to detect the density of vehicles on the road and adjust the traffic light timings accordingly. The goal of this system is to improve traffic flow and reduce congestion by allowing more time for vehicles to pass through intersections when there is a high density of vehicles on the road.

The project on a density based traffic light controller system typically involves designing and building the hardware and software components of the system, including the sensors, microcontroller, and traffic light controller. These components work together to detect the density of vehicles on the road and adjust the traffic light timings in real-time to optimize traffic flow.

The sensors used in the system can be infrared sensors, video cameras, or other types of sensors that can detect the presence and number of vehicles on the road. The microcontroller processes the sensor data and calculates the density of vehicles, and the traffic light controller uses this information to adjust the timing of the traffic lights.

In addition to designing and building the hardware and software components, the project may also involve testing and evaluating the performance of the system in different traffic scenarios to determine its effectiveness in improving traffic flow and reducing congestion. This can involve collecting data on traffic patterns, measuring the time it takes for vehicles to pass through intersections, and comparing the performance of the density based traffic light controller system to traditional traffic light controller systems.

Overall, the project on a density based traffic light controller system aims to provide a solution to traffic congestion and improve the efficiency of traffic flow on roads. By using sensors and advanced algorithms, the system can adapt to changing traffic conditions and optimize the timing of traffic lights to improve the flow of vehicles.

Chapter 2

Literature

2.1 FIXED TIME TRAFFIC LIGHT CONTROLLER SYSTEM

There is a significant amount of literature on density based traffic light controller systems. Many studies have been conducted to evaluate the effectiveness of these systems in improving traffic flow and reducing congestion on roads.

One study published in the Journal of Traffic and Transportation Engineering found that a density based traffic light controller system was able to reduce congestion and improve traffic flow compared to a traditional fixed-time traffic light controller system. The study used simulation models to evaluate the performance of the density based system in different traffic scenarios, and found that it was able to reduce the average waiting time for vehicles and improve the overall flow of traffic.

Another study published in the Journal of Intelligent Transportation Systems found that a density based traffic light controller system was able to improve the efficiency of traffic flow and reduce congestion on roads. The study used real-world data from a traffic intersection in China to evaluate the performance of the system, and found that it was able to reduce the average waiting time for vehicles and improve the overall flow of traffic.

2.2 DEVOLEPMENT OF A LOW-COST COMMUNITY BASED REAL TIME TRAFFIC LIGHT CONTROLLER SYSTEM

The proposed system employs the use of low-cost Arduino Uno micro controllers and other low-cost devices to detect traffic and alert the community in real time. Results obtained from the system prototype/field test demonstrated its capability in mitigating the devastating impacts of traffic especially for the poorest and most vulnerable communities in developing countries. Also, it is cost effective project that can be easily developed with the help of simple Arduino knowledge. Any prior knowledge of Arduino is not required, you can start with the basics as well.

Chapter 3

Goal and Methods

The methods for a density based traffic light controller system typically involves several steps, including designing and building the hardware and software components of the system, testing and evaluating the system in different traffic scenarios, and implementing the system in a real-world traffic environment.

1. Design and build the hardware and software components of the system. This typically involves selecting the appropriate sensors, microcontroller, and traffic light controller for the system, and designing the hardware and software to work together to detect the density of vehicles on the road and adjust the traffic light timings in real-time.

2. Test and evaluate the system in different traffic scenarios. This can involve using simulation models or real-world data to evaluate the performance of the system in various traffic conditions. The goal of this step is to determine the effectiveness of the system in improving traffic flow and reducing congestion.

3. Implement the system in a real-world traffic environment. This involves installing the hardware and software components of the system at a traffic intersection and collecting data on traffic patterns and the performance of the system. The collected data can be used to further refine the system and optimize its performance.

Overall, the methodology for a density based traffic light controller system involves designing and building the necessary hardware and software components, testing and evaluating the system in different traffic scenarios, and implementing the system in a real-world traffic environment to improve traffic flow and reduce congestion.

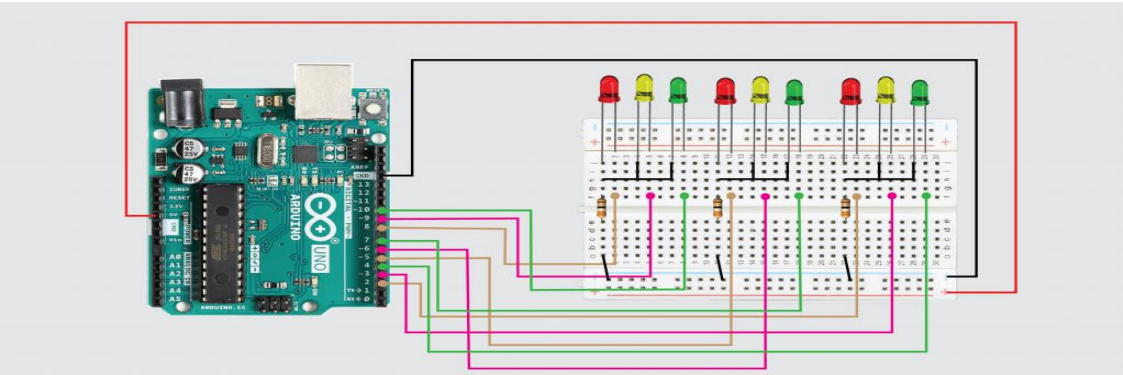

Fig -1.1: Basic Model of the System

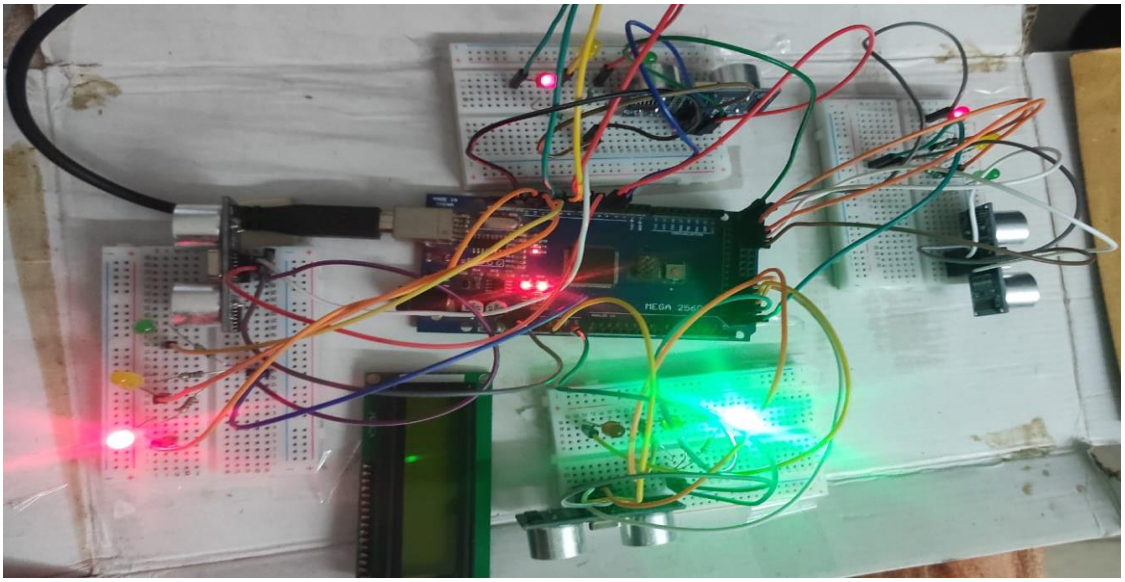

Fig -1.2: Completed Model of the System

References

[1]. M. A.A. Parkhi, Mr. A.A. Peshattiwar, Mr. K.G. Pande "Intelligent Traffic System Using Vehicle Density". Yeshwantrao Chavan College of Eng., Nagpur. International Journal of Electrical and Electronic Engineers, 2016.

[2]. Bilal Ghazal, Khaled ElKhatib "Smart Traffic Light Control System". Conference Paper- April 2016.

[3]. Dinesh Rotake, Prof. Swapnil Karmore "Intelligent Traffic Signal Control System Using Embedded System". G.H Raisoni College of Engineering, Nagpur. Innovative Systems Design and Engineering, 2012.

[4]. Malik Tubaishatr, Ti Shang and Hongchi Shi "Adaptive Traffic Light Control with Wireless Sensor Networks". Article- January 2007.

Appendices

Appendix A

Code Attachments

A.1 Arduino Code

```
#include<TimerOne.h>

int signal1[] = {23, 25, 27};
int signal2[] = {46, 48, 50};
int signal3[] = {13, 12, 11};
int signal4[] = {10, 9, 8};

int redDelay = 500;
int yellowDelay = 200;

volatile int triggerpin1 = 31;
volatile int echopin1 = 29;
volatile int triggerpin2 = 44;
volatile int echopin2 = 42;
volatile int triggerpin3 = 7;
volatile int echopin3 = 6;
volatile int triggerpin4 = 5;
volatile int echopin4 = 4;

volatile long time;                 // Variable for storing the time travelled
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(115200);
  Timer1.initialize(100000);  //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<3; 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);
}

void loop()
{
  // If there are vehicles at signal 1
  if(S1<t)
  {
    signal1Function();
  }
```

```
  // If there are vehicles at signal 2
  else if(S2<t)
  {
    signal2Function();
  }

  // If there are vehicles at signal 3
  else if(S3<t)
  {
    signal3Function();
  }

  // If there are vehicles at signal 4
  else 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;

// Print distance values on serial monitor for debugging
Serial.print("S1: ");
```

```
  Serial.print(S1);
  Serial.print("  S2: ");
  Serial.print(S2);
  Serial.print("  S3: ");
  Serial.print(S3);
  Serial.print("  S4: ");
  Serial.println(S4);
}

void signal1Function()
{
  Serial.println("1");
  low();
  // Make RED LED LOW and make Green HIGH for 5 seconds
  digitalWrite(signal1[0], LOW);
  digitalWrite(signal1[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);
    delay(yellowDelay);
  }
}

void signal2Function()
{
  Serial.println("2");
  low();
```

```
  digitalWrite(signal2[0], LOW);
  digitalWrite(signal2[2], HIGH);
  delay(redDelay);

  if(S1<t || S3<t || S4<t)
  {
    digitalWrite(signal2[2], LOW);
    digitalWrite(signal2[1], HIGH);
    delay(yellowDelay);
  }
}

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

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

void signal4Function()
{
  Serial.println("4");
  low();
```

```
  digitalWrite(signal4[0], LOW);
  digitalWrite(signal4[2], HIGH);
  delay(redDelay);

  if(S1<t || S2<t || S3<t)
  {
    digitalWrite(signal4[2], LOW);
    digitalWrite(signal4[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);
  }
}
```