**UNIVERSITY OF ZIMBABWE**

****

**INTELLIGENT TRAFFIC LIGHT**

**PERSEVERANCE MASHAYA R195863Q**

**BRUCE MWEDZI R195932F**

**DUNCAN J KACHASU R1810106**

**Project submitted for review to the Department of COMPUTER SCIENCE (CTHSC) in the Faculty of Computer Engineering Informatics and Communications at the University of Zimbabwe**

**Supervisor: MR MICHAEL MUNYARADZI**

**Table of Contents**

CHAPTER ONE

INTRODUCTION

CHAPTER TWO

LITERATURE REVIEW

CHAPTER THREE

METHODOLOGY

CHAPTER FOUR

SYSTEM ANALYSIS AND DESIGN

CHAPTER FIVE

IMPLEMENTATION AND TESTING

REFERENCES

APPENDIX

PROJECT MODULES

**Chapter One**

**INTRODUCTION**

Today, we face the challenge of inequality in the growth of road infrastructure compared to the growing number of vehicles in modern life. The current traffic signal system is a static system and cannot adapt to the changes in the traffic. The static system functions on a present timer system, which may lead to increased waiting time at signals, speed reduction, longer travel hours and an increase in vehicle queues. Around midday traffic from the University of Zimbabwe is congested as compared to the flow of traffic from the other ends of the T junction, this in turn with the traffic light that operates on a loop and fixed times makes the junction end from the University of Zimbabwe busy. This traffic congestion can cause more fuel consumptions followed by wastage of fuel and time, more cost of depreciation and fuel, environmental pollution and accident, and verbal and physical conflicts on the roads. The static timer system does not account for this increased congestion from the University of Zimbabwe and there is unnecessary waiting for vehicles on the congested road. In contrast, in the lane with less traffic congestion, all the vehicles disperse before the green timer expires which results in wastage of green time. These issues and problems in the current system lead towards a smart, intelligent traffic light. The system proposed in this document will implement artificial intelligence by means of the open CV library will allow the detection of objects using computer vision which in turn helps us to devise an algorithm that can count the number of objects. After counting it will compare all the ends of a T junction and allocate more time to the end which have large number of vehicles for example during the midday more time will be allocated to the end from University of Zimbabwe. Use of cameras will also come into play. For demonstration we are going to use saved videos because we do not have the cameras. In the article entitled “Dynamic Traffic System Based on Real Time

Detection of Traffic Congestion”, they used Raspberry Pi o W’s and in our project we are going to use Arduino Uno. We are going to use electricity as our power supply and maybe solar system as backup as we are experiencing electricity shortages. The reason why we choose electricity is that solar can be easily affected by bad weather conditions which may lead the traffic light not working. And also in the morning it will be not working this will result in accidents. In the coming chapters we are going to have literature research to see how others who have done the similar project how they do it. Also note down methods we used in the project. We are going to design how our project is going to look like in form of diagrams and we also analyzed the system. After the analysis we are going to test to see if it works well then we conclude.

**AIMS AND OBJECTIVES**

* Program a traffic light that allocates time based on an algorithm that implements external devices to detect which end of the T junction is more congested.
* Implementing artificial intelligence. Based on previous data, the traffic light will be able to predict which ends will be more congested at what time.
* Being able to allocate more time to the end with many cars on a T junction. The time allocated should have maximum so that we will give time to other ends so as to reduce waiting time to the ends with less cars.
* Using knowledge about road systems in Zimbabwe, the traffic light will be able to follow road rules for example which end of the junction has to give way. While one end of the T junction is green the other end should be red.

**Hardware**

Light emitting diodes (green, red, amber)

Resistors (of size 1 kilo ohm)

Arduino Uno

Bread board

Connecting wires

Cable

**Summary**

The following chapters will give more details about the project. The processes that take place to complete our objectives will be detailed in the following chapters. To ensure that our objectives are being completed we will conduct system tests, also explained in the following chapters.

**CHAPTER TWO**

**LITERATURE REVIEW**

**Overview of contents:**

* Image processing(the project aims to process vehicles and count their number so as to iterate where more vehicles are located)
* Prediction mechanism where Artificial intelligence comes into play
* Timer allocation(each end of the T junction is allocated time according to the situation at hand)

**Relevant work by other authors in line with this project:**

According to the article entitled Dynamic Traffic System Based on Real Time

Detection of Traffic Congestion, the abstract stated that with a sharp increase in the number of vehicles, there is a dire need for systems that can adapt towards the changes in traffic. The inability of current systems to deal with this increased traffic leads to inefficient traffic management.

This is the same problem we are aiming to solve in our ITL project. Instead of using raspberry pi which they used, we will be dealing with Arduino Uno.

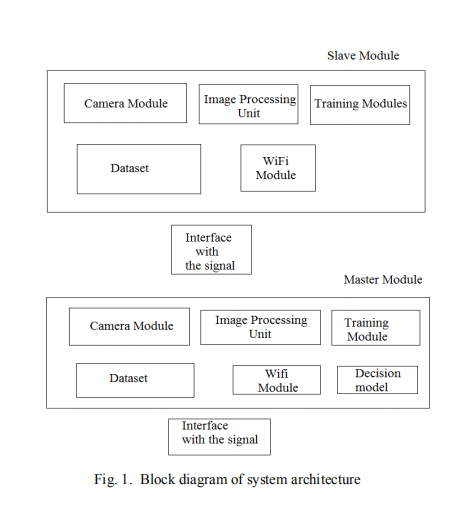
The article introduction sites, “Traffic is one of the major problems that we face in our day to day life. The current traffic signal system is a static system and cannot adapt to the changes in the traffic. The static system functions on a preset timer system, which may lead to increased waiting time at signals. During peak hours there is increased congestion on certain directions and in comparatively less congestion in other directions. The static timer system does not account for this increased congestion and there is unnecessary waiting for vehicles on the congested road. In contrast, in the lane with less traffic congestion, all the vehicles disperse before the green timer expires which results in wastage of green time. An extended effect of this traffic congestion is pollution and increase in fuel wastage. These issues and problems in the current system lead towards a smart, dynamic and real time traffic system.

The system proposed in the paper is dynamic in nature, which takes in present traffic video footage and calculates congestion from each respective lane. The congestion from each lane is compared, and the signal timer is allocated according to the congestion percentage. The objective is to design a dynamic system, that is able to modify the time allocation based on real time traffic congestion, make use of image processing techniques and a predictive mechanism to detect the percentage congestion on the road. This system will reduce the waiting time on signals and improve the efficiency of traffic signal systems.”

In our ITL project we also tried to reduce the congestion of traffic from each lane by using opencv library for image processing so that the number of vehicles is recorded. The number that was recorded will determine the switch from red to green on the lanes so that time allocation is reasonable and better for each end with more vehicles.

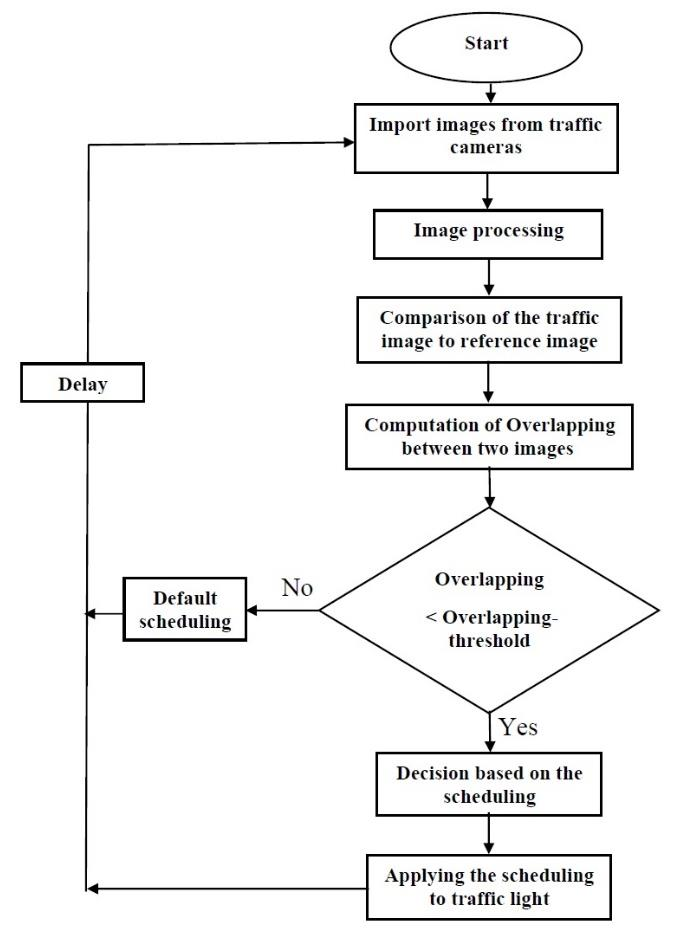
In that same article entitled “*Dynamic Traffic System Based on Real Time*

*Detection of Traffic Congestion”* they put up a diagram to illustrate what they were doing. The block diagram gives more insight of what they made use of in their project.

**

Another related project article entitled “Smart Traffic Light Scheduling in Smart City Using Image and Video Processing” sites the use of Raspberry-Pi board and OpenCV tool and our ITL project is using the OpenCV tool as well but Arduino Uno board instead of Raspberry-Pi board.

The flow chart which explains the “Smart Traffic Light Scheduling in Smart City Using Image and Video Processing”



Cameras were used for capturing images. In ITL we used a recorded video to test the functionality of the algorithm.

In the article with the same flow chart, number of vehicles passing through the main streets leading to the crossroads is determined. Therefore, the location of the

Camera to determine the number of vehicles is far from the crossroad and on the path leading to the crossroad. Videos from each direction are provided with Raspberry Pi. Then, the video processing is performed using the video background removal method. The moving vehicle is displayed with the use of filters as white stains and is separated from other parts of the video. The number of passing vehicles and their movement direction are determined by the two first and second stimulation lines created by coding on the video frame being processed.

There have been different studies conducted on traffic management for a smart city. Most of them have been designed and implemented Raspberry Pi controllers. These controllers take the information from the sensors and issue relative commands by analyzing data that have been programmed on them.

G. S. Khekare proposed a vehicular ad-hoc networks (VANETs) that is a typical example of a variety of networks in wireless technologies. Their methodology is based on the smart city framework that will transfer information about traffic conditions and help drivers choose a suitable direction to prevent traffic congestion.

S. Badura et al. introduced a new model for intelligent traffic systems that consolidate monitoring features through existing cameras at intersections and allow users to access data using data delivery systems. Image analysis and modeling of previous designs are important elements of monitoring, and data transfer is performed on a mobile ad-hoc network.

A. S. Salama presented a design of an integrated intelligent system for managing and controlling traffic lights with the help of photoelectric sensors. The FPGA processor was used in this model. One of the most important criteria in this system is the

installation and implementation of sensors, because the traffic lights are scheduled and controlled according to the sensor values based on an algorithm with a proper relative weight for each of the directions. In addition, the system can be programmed for emergency scenarios, such as passing the president, ministers, ambulances and fire engines through a technology based on active radio frequency detection.

H. Zhao introduced a design of an intelligent traffic control system based on DSP and NIOS. Their model used intelligent traffic control system with FPGA processor, which includes functions such as mutual phase adjustment, exchange and production of relevant information and human-computer interaction [9].

A. Kanungo et al. used a MATLAB simulator and calculated traffic congestion by estimating spaces occupied by vehicles and empty spaces in the path to reduce the image matrix, calculate the congestion of all red directions, and signal the green light to the path with maximum congestion. The traffic congestion of a previously green path is converted to zero, and congestion of the two red lights remains the same on the other directions. The green signal schedule is also calculated using the number of vehicles that can be passed in each congestion per second. This design showed a 35% improvement compared to the usual situation, in which the same time is considered to the green light signal in each direction.

The proposed system in suffers from the limitation of the VANET hardware installation in each vehicle. The models designed in [8, 9] also demand constant and costly maintenance as well as system analysis. Moreover, most of them are more vulnerable due to the hard external conditions. The method introduced in the present paper largely eliminates these limitations. Furthermore, it is implemented on crossroads and does not require any connection with a car.

From another article entitled, **“Iktishaf+: A Big Data Tool with Automatic Labeling for Road Traffic Social Sensing and Event Detection Using Distributed**

**Machine Learning”, it was cited in the abstract part of the article that**

“The most specific contribution of this research is an automatic labelling method for machine learning-based traffic-related event detection from Twitter data in the

Arabic language. The proposed method has been implemented in a software tool called Iktishaf+ (an Arabic word meaning discovery) that is able to detect traffic events automatically from tweets in the Arabic language using distributed machine learning over Apache Spark. The tool is built using nine components and a range of technologies including Apache Spark, Parquet, and Mongo DB.

In section 2.2 of the article, they monitored events on the road through the tweets people share on twitter and that is how they detected traffic jams even at T junctions:

“A very limited number of studies have proposed to analyze Arabic social information for traffic event detection, so first we review the studies about detecting any event not necessarily related to traffic. Then, we review the works that focus on transport and traffic events. Finally, we discuss the works that use big data. Alkouz and Alghbari analyzed English and Arabic data, including standard Arabic and UAE dialectical posts from Twitter and Instagram to detect and predict traffic jams.”

There was the use of machine learning in the **Iktishaf+: A Big Data Tool with Automatic Labeling for Road Traffic Social Sensing and Event Detection Using Distributed Machine Learning article** which is also another objective in our Intelligent Traffic Light project where we need to train a build and train a model so that it meets the standards of the aims of ITL.

The need to solve congestion in our ITL project was also stated in the same article, “Increased urbanization is giving rise to the evolution of cities into mega cities where traffic congestion is a leading problem causing devastating economic, social, and ecological losses. The annual cost of congestion in the US is USD305 billion, not to mention the damages to health and the number of deaths. Congestion is caused due to the steadily growing traffic in the cities over the years, road damages, road works, traffic accidents, bad weather, and other contingencies. There is a need to detect these causes or events to enable timely planning and operations. Many times, congestion is caused due to events that are beyond the direct scope of physical road sensors, and therefore physical sensors cannot detect these events until the effects of these events are visible on the roads and can be sensed by the on-road sensors”

There is need to monitor the current movement of vehicles on the road and in this project we used image processing which we were able to accomplish using the OpenCV library to count the number of vehicles from each end.

**Other T junctions**

The T junction at Zimre Park turn off has traffic lights that do not work well when the weather is not favorable with what the solar panels require early in the morning before sunrise. The traffic lights do not work on Sundays since most people stay at home and visit their respective churches on this day and a few have the need to travel through that route.

There is also the T junction at Sunning dale turn off which works just like any other T junction. Seldom are times when there are accidents occurring at that junction but there are few cases from time to time, but with Artificial intelligence added to that robot’s functionality it will make it a lot more favorable for the conditions when there is too much traffic

**For hardware modules in Intelligent Traffic Light (ITL) we intend to use:**

* Arduino Uno (the other article Dynamic Traffic System Based On Real Time Detection Of Traffic Congestion states the use of Raspberry Pi 0 W micro controller instead)
* Camera Module
* LCD Display for output

**Software Modules:**

OpenCV for image Processing

**Other relevant materials for the project ITL:**

* 6 1kꭥ resistors
* Connecting wires
* Leds(red, yellow and green)
* Bread board

**Chapter Three**

**Methodology**

**Research Question – How to design an intelligent traffic light?**

**Description**

Under this chapter, we analyze the procedures used to achieve our objectives. The information gathered from research and observing other systems will be included. We also outline the experimental and simulation data.

**Research Methods Justification**

For our project, we could not fixate on one research method, because it proved limiting.

We used qualitative and quantitative research method which are outlined below. In qualitative research, our objective was collect information that might help us answer our research question.

In quantitative research our objective was to understand the physical data obtained from experiments and simulations to better understand our research question.

The research methods we selected are:

* Observational
* Simulation
* Derivation

**Qualitative Research**

**Observation**

Our objectives required intense observational data to give us insight, into how our project can meet real world applications.

The following are observations we made

**Abstract Observations**

Table 3.1

|  |  |
| --- | --- |
| Observation | Reason |
| Traffic flow at various T-junctions | To learn and understand how traffic flows and how to incorporate the data in our project |
| Traffic light delays(time between respective color interchanges | To scale the delays to our system |
| Traffic that passes at certain time intervals | To help our system in the optimum delay selection |

**In detail Observations**

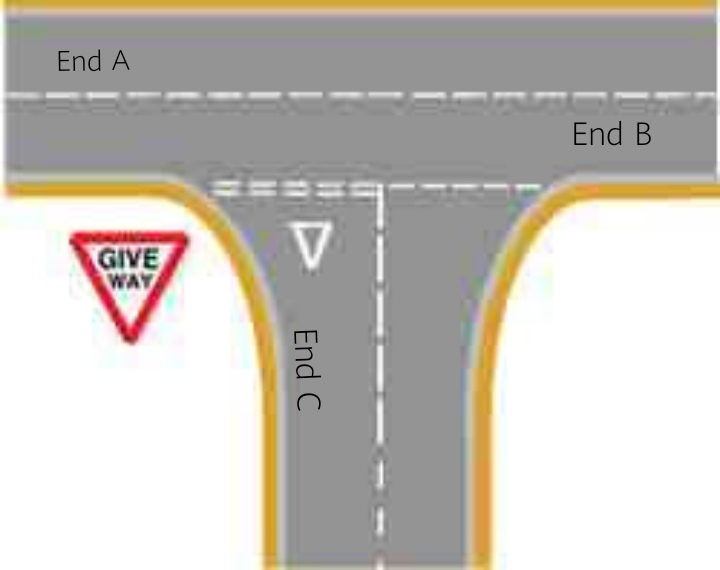
The below table contains data collected observing T-junctions

Table 3.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | End A | End B | End C | Average |
| Timing(Delays) |  |  |  |  |
| Traffic Count/Delay |  |  |  |  |
| Traffic Count/Hour |  |  |  |  |

We made observations of how traffic flows on T-junctions. Our effort was to understand the basic sequence in which traffic flows from one end to another and in each other.

Figure 3.1



As illustrated by the figure above, traffic flow from end C gives way to traffic from end A. We can also understand that the delay sequences used for End A and End B are the same.

This observation helped in our research, our fundamental design of the traffic light came down to controlling only two distinct sets of delay sequences.

**Quantitative Research**

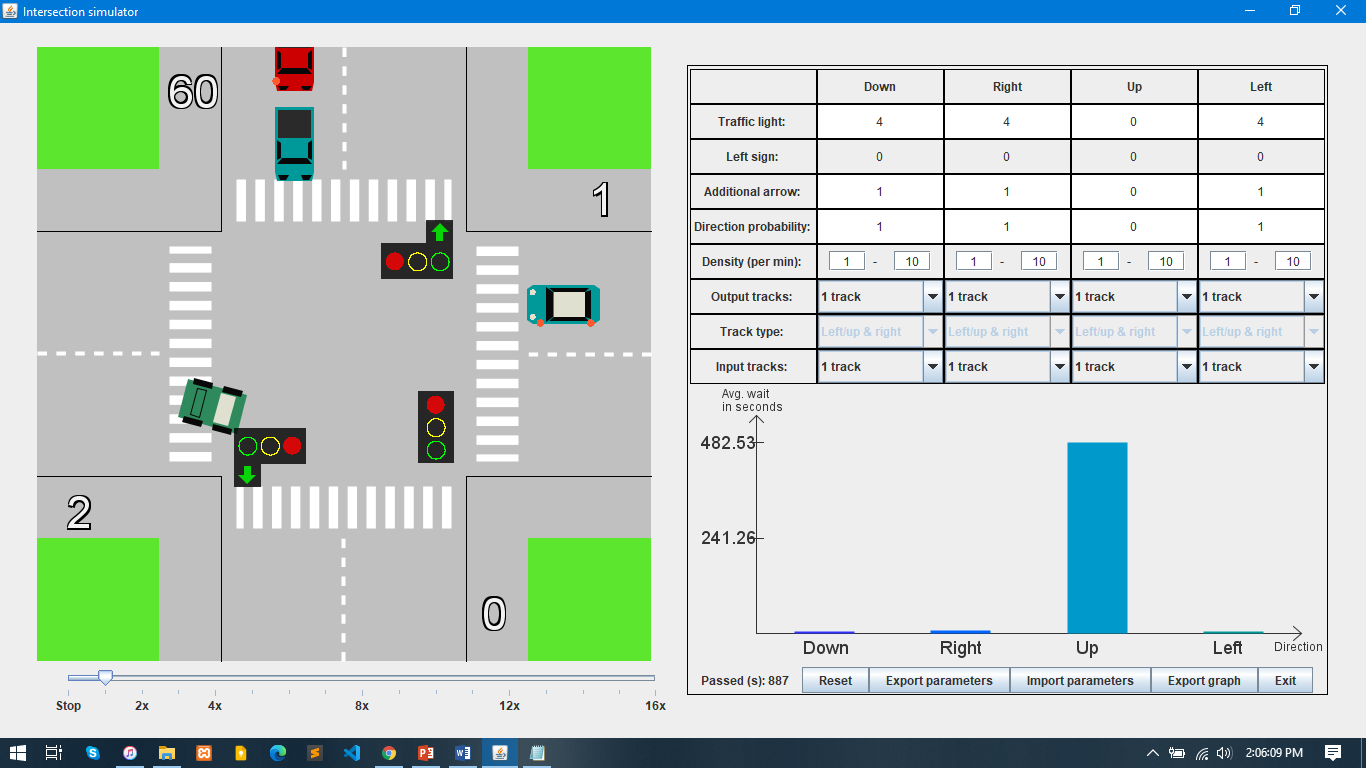
**Simulation**

A simulation’s underlying purpose is to shed light on the underlying mechanisms that control the behavior of system.

Our objective was to understand the flow of traffic, how delays affect the system and to determine the best way to design our system.

Traffic simulation efforts were made easier by using a software developed by Krešimir Kovačić (2015) called Intersection Simulator

The figure below shows an overview of the software.

 Figure 3.2

The Intersection Simulator had at its disposal tools to set traffic simulatory environments, like how much traffic flows from one end of a T-junction.

The following table contains data collected from simulations

Table 3.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | End A | End B | End C | Average |
| Timing(Delays) |  |  |  |  |
| Traffic Count/Delay |  |  |  |  |
| Traffic Count/Hour |  |  |  |  |
| Wait time(seconds) |  |  |  |  |

**Derivation**

To optimize our model system, we had to use derivative data.

According to Chandra Vennapoosa (2015), Data Derivation refers to the process of creating data values from contributing data values through some data derivation algorithm.

Although data derivation requires a lot of raw data, our data comprises of that collected from observations and simulations.

Comparing simulation and observational data, gave us an insight to optimum ways of designing our system.

One of the reasons we chose to use derived data is that original data is complex in analysis and doing system tests. The derived data help us scale our system to real world data values.

The following table outline the derived data

Table 3.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Original Data | Derivation Method | Derived Data | Scaled to System |
| Average Traffic Count/Delay |  |  |  |  |
| Average Traffic Count/Hour |  |  |  |  |
| Delays |  |  |  |  |

**Software Development Methodology**

After much research, we had sufficient data to start the design of our traffic light. As our software development methodology we selected Waterfall model.

This type of software development method is used to improve the effectiveness of software development.

The figure below outlines the Waterfall Model

Figure 3.3



**Requirements**

At this stage we collected the software requirements of the traffic light. In the next chapter (System Analysis and Design), we have all requirements used outlined.

**Design**

At this stage we used data collected from observations and experimental data, which helped in designing a system that works.

**Implementation**

At this stage our aim was to verify if the system works from the work done using the data from the design stage. We also looked at the errors that might come with implementation.

**Verification**

At this stage we verified the system functionality with the objectives, making sure that the requirements are being fully met.

**Maintenance**

This is a further stage, but in time after the real world implementation of our traffic light, we can then maintain its performance. This helps in making sure that the system still works as intended.

Reasons for choosing the Waterfall model

* It’s simple and easy to use
* Stages are processed one at a time
* We could easily understand our milestones and progress

**Chapter Four**

**System Analysis and Design**

**Description**

This chapter’s role seeks to give an understanding and appreciation for the whole system. We will cover the analysis of data flow systematically, the system development cycle – outlining how our system flows from analysis to design and implementation.

**System Analysis**

According to Ben Davis (2021), System analysis is conducted for the purpose of studying a system in order to identify its objectives. It’s a problem solving technique that improves the system and ensures that all components of the system work efficiently.

**System objectives**

* Object Detection
* Object Counting
* Variable Delays

**System Analysis Tools**

* Process Flow Diagram

**Process Flow Diagram**

According to Nancy Tague (2005), the purpose of the flow diagram is to develop an understanding as to how a process is done and to study a process for improvement.

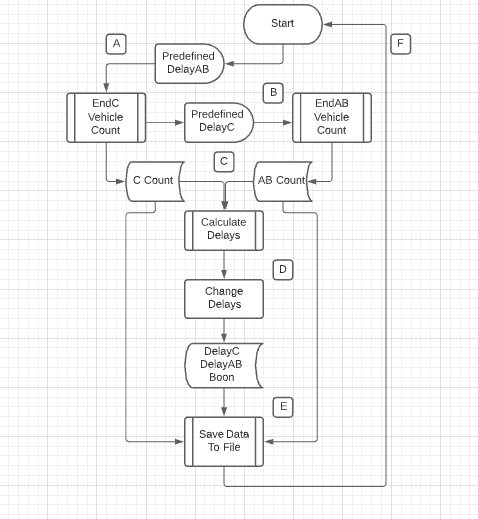
The process to be diagrammed involves the major objectives of the system.

**Process Activities**

* Process uses predefined variables for delays
* Counts vehicles from each T-junction end
* Calculates delays using predefined function
* Change predefined delays with new calculated delays
* Save data to file

The figure below illustrates an abstract process flow of the system

Figure 4.1

**Detailed Explanation of the Process**

**Steps**

1. A

On system start up, there are predefined values that represent the delays for the T-junction ends.

When traffic is idle on EndAB using the delay DelayC, traffic is flowing through EndC and a predefined method counts the traffic passing through.

A value corresponding to the traffic count is stored (C Count).

1. B

After DelayC expires, traffic is idle on EndC using delay DelayAB, traffic flows through EndAB and a predefined method counts the traffic passing through.

A value corresponding to the traffic count is stored (AB Count).

1. C

Taking values C Count and AB Count, the system uses another predefined method to calculate the respective junction ends delays using the traffic counts.

A value called Boon is defined based on which junction end has more delay time. Basically the value represents preference.

1. D

Using the delay values calculated in the step above, the system changes the previously predefined delays with the new values.

1. E

The data from the above steps is collected and stored to file using a predefined method. The data collected over time will be used to model a machine learning model.

1. F

After the data is saved, our system repeats the loop using new delay values calculated for the respective junction ends.

**System Evaluation**

We will evaluate the system to see if it aligns with the objectives and find ways to improve it.

This will be achieved using a preloaded videos of traffic flow because the right equipment costs. From these videos our aim is to extract vehicle count to be used in calculating the delays.

**System Design**

According to Nancy Tague (2005), system design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified objectives.

**Overview**

Wikipedia encapsulated system design down to the act of taking system analysis information and designing a system that checks all project requirements. We broke down or system design into three sections, Architectural, Logical and Physical design.

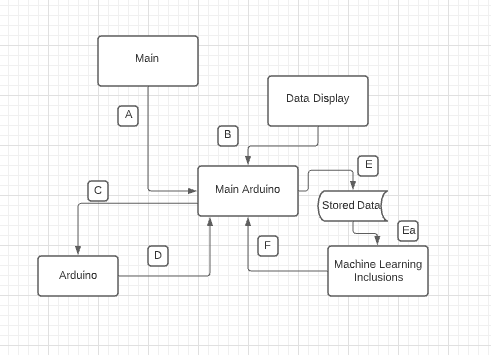
**Architectural Design**

Pranathi Badugu (2018) define architectural design as the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.

We used call and return architecture method, this allows us to create a program that is easy to scale and modify.

The figure below illustrates the systems architectural design

**Figure 4.2**



The figure above outlines the main modules of the system, how they are connected. Letters indicated follow the data flow.

**Modules Overview and Behavior**

The process flow in Figure 4.1 takes place in the Main Arduino module. The predefined functions are imported from the modules Main and Data Display.

Arduino module is an Arduino hardware component that is that functions with the data input it obtains from the Main Arduino module.

The Main Arduino module generated data and stores it externally. The stored data is used to train a machine learning model. Models require a lot of data for accuracy. When our model becomes accurate, the Machine Learning Inclusions module with weigh in the delay decision making process.

**Logical Design**

The logical design of a system pertains to an abstract representation of data flow, inputs and outputs of the system, Pranathi Badugu (2018). Using Figure 4.2 above, we are going to explain the logic behind our system.

**Main Module**

The main module contains four major methods that are implemented by the Main Arduino module.

**Methods**

* + Vehicle Counter

The purpose of this method is to activate the video stream and utilize the opencv library to detect objects passing through. This returns a traffic count value.

* + Delay Calculator

This method takes in arguments of the traffic counts obtained from the method above and using a constant value (object per second) calculate the respective delays of the T-junction.

* + Board Controller

This method controls the output of the system through an Arduino board. The delays calculated from the above method are used to toggle the leds representing traffic lights.

* + Traffic Logs

This method logs every value created and calculated during the process flow. The data will be later used to train a machine learning model.

**Data Display Module**

This module’s only task is to display the acquired statistics to the console screen. This helps us debug if an error occurs or if some functionality requires improvements.

**Main Arduino Module**

The system’s process flow revolves around this module. It imports all the methods from the described modules above. One in particular is the method for logging traffic stats, the data is stored in a csv file. The document will be later used in the module below.

**Machine Learning Inclusions Module**

This module’s only purpose is to use the stored collected traffic stats in conjunction with a saved trained model to predict the probability of prioritizing a junction end at a particular hour.

Machine learning requires lots of data for accuracy, our system using a model trained with insufficient data will produce unreliable results.

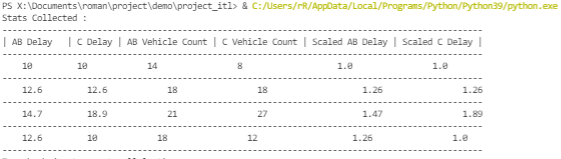
**Physical Design**

Physical Design relates to the actual input and output processes of the system. To further illustrate the concept, we divided physical design into two subtasks. The subtasks are Data Design and Process design.

**Data Design**

We are looking at how data is represented and stored. The Data Display module, Vehicle counter method and Traffic Log method can help present the case of data design.

**Data Display Module**

Figure 4.3

The figure above shows how data is displayed using the data display method defined in the data display module.

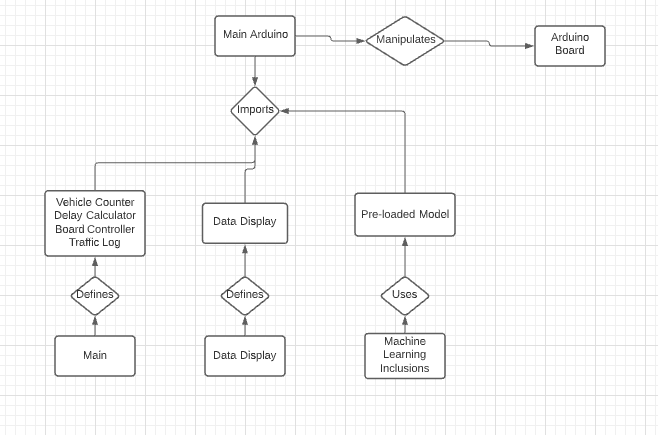
AB Vehicle and C Vehicle Count as illustrated above are values returned from the Vehicle Counter method.

The data presented in the figure above is stored in csv file using the Traffic Log method. The same csv file is used in training the machine learning model for predictions.

**Process Design**

This section acts as an optimization for the improvement hardware components and software tools.

We are going to focus on how the process flows from the software specifications to how the Arduino will behave based on the data that it receives from the main Arduino module.

Figure 4.4

The figure above shows how the modules interact from imports, data manipulations to hardware manipulations.

Our process has 4 different relationships. Looking at them will help explain how the process unfolds.

Note – For the modules to be mentioned in the following section, full explanation available in the Logic Design Section.

**Relationship – Defines**

This relationship is used twice. Firstly with the Main Module to define the methods shown in the figure above.

The Data Display Module defines a data display method.

**Relationship – Uses**

Machine Learning Inclusions Module uses a pre-loaded model to do predictions.

**Relationship – Imports**

The Main Arduino module imports all the shown methods. The pre-loaded model is still in an experimental phase, so physical right now it’s not being implemented.

After the imports, the module executes all the methods. This execution brings about a number of values, all of which are shown in Figure 4.3 (Data Design).

**Relationship – Manipulates**

This relationship occurs physical when the Main Arduino module is in execution. The values mentioned in the above relationship are used to manipulate the behavior of the Arduino.

The values manipulates the delay for all junction ends, adds a row to the stored data file.

**Chapter 5**

**Implementation and Testing**

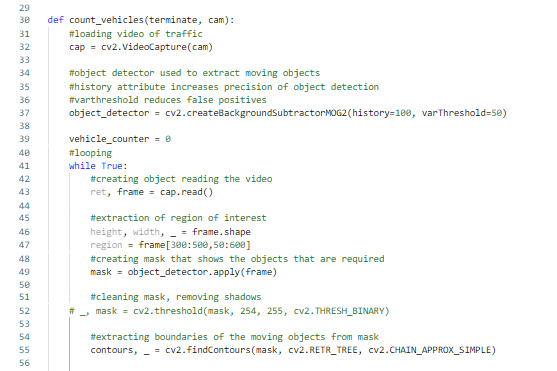
**Description**

This chapter focuses on the implementation of our system, which is ensuring that it is fully operational. Using the designs outlined in the chapter above, we are going to show how our system modules are working.

We are describing implementation using an abstract system flow that has its tasks executing in the following order

1. Count Vehicles
2. Calculate Delays
3. Manipulate Delays

A

Figure 5.1

The figure above is a code snippet that shows part of a method (Vehicle Counter)

The function count\_vehicles takes in two arguments labeled terminate and cam.

Terminate is a value that controls how long a camera is performing object detection.

Cam is a value that references the camera used for object detection.

From this function a value representing traffic count is returned.

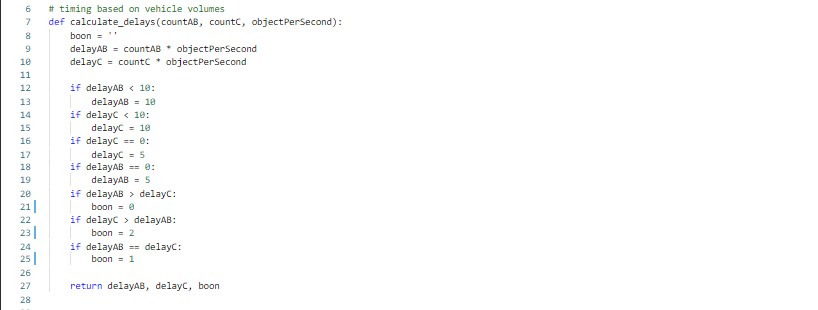
B

Figure 5.2

The figure above illustrates a function that calculate delays based on the vehicle volume

After the vehicle counts have been collected from respective junctions, the values are used in calculating their respective delays.

The function calculate\_delays takes in three positional arguments. CountAB and countC values are the vehicle counts from the T-junction end, AB and C.

The objectPerSecond argument is a value that is calculated from the traffic statistics that are collected. Basically its purpose is to provide a constant value that defines rate of flow of traffic per second.

Three values are returned, two of which are the respective delays for the junction ends. The boon value however is used to keep track of how many times a junction end is prioritized at a particular how.

C

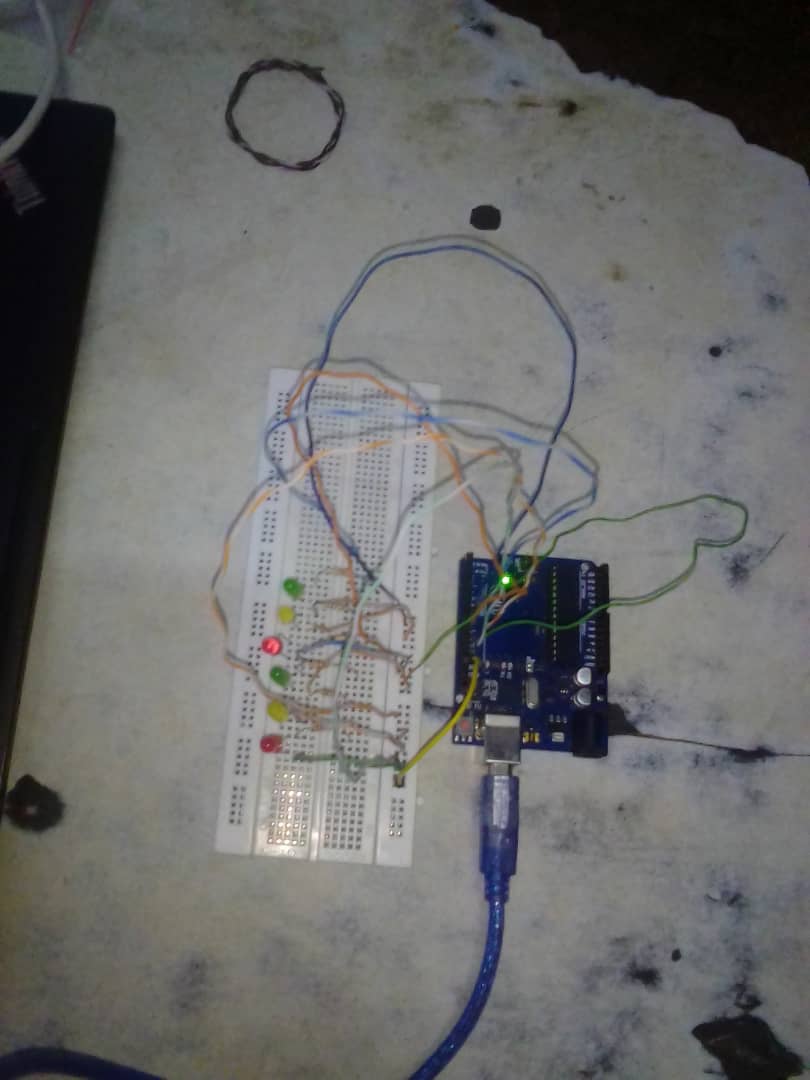
The delay values that are obtained are used to update the predefined variable values.

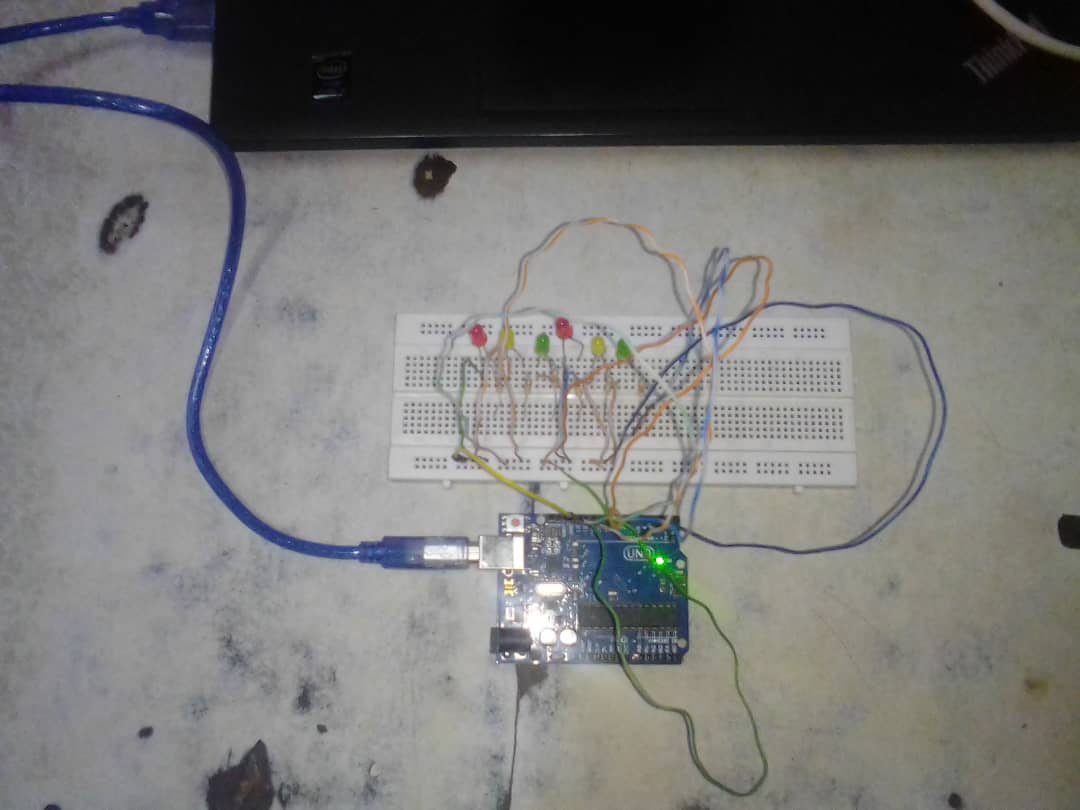
**System Testing**

System testing is done to verify that the system’s requirements are met. We performed functional tests. These tests focus on whether if the system is actually fully operational tests.

**Functional Test**

The basis of these tests is to make certain that the system is producing expected results.

Figure 5.3



The figures above shows the leds responding to the delay change

The figure cannot actually prove that its working but we are still going to outline the results we collected.

|  |  |  |
| --- | --- | --- |
| Test | Method Used | Status |
| Object detection and counting | Defined a function that uses the openCV library to detect objects from video (live or not). | Fully Operational |
| Calculating delays based on object count | Defined a function that calculated delays from the object counts and a constant | Fully Operational |
| Predictions using machine learning | Trained a model from the data collected from tests | Fully Operational (However prediction accuracy depends on data. The more the data the more accurate the prediction becomes) |

**References**

A. Kanungo, A. Sharma, and C. Singla, “Smart traffic lights switching and traffic density calculation using video processing,” 2014 Recent Adv.

Eng. computer. Sci., pp. 1–6, 2014.

H. Zhao, “Intelligent Traffic Control System Based on DSP and Nios II,”

pp. 90–94, 2009.

Prashant Jadhav, Pratiksha Kelkar, Kunal Patil, Snehal Thorat Smart Traffic Control using Image Processing, Volume 3,International Research Journal of Engineering and Technology (IRJET)

<https://www.mdpi.com/journal/sensors>

K. Vidhya and A. B. Banu, “‘Density Based Traffic Signal System,’” Int.

J. Innov. Res. Sci. Technol., vol. 3, no. 3, pp. 1–3, 2014.

M. Z. Talukder, S. S. Towqir, A. R. Remon, and H. U. Zaman, “An IoT

Based automated traffic control system with real-time update capability,”

2017 8th Int. Conf. computer. Communication, Network. Technologies, pp. 1–6, 2017.

G. S. Khekare, “A Smart City Framework for Intelligent Traffic System

Using VANET,” pp. 302–305, 2013.

S. Badura and A. Lieskovsky, “Intelligent traffic system: cooperation of

MANET and Image processing,” 2010.

A. S. Salama, “Intelligent Cross Road Traffic Management System

(ICRTMS),” no. Icctd, pp. 27–31, 2010.

H. Zhao, “Intelligent Traffic Control System Based on DSP and Nios II,”

pp. 90–94, 2009.

A. Kanungo, A. Sharma, and C. Singla, “Smart traffic lights switching

And traffic density calculation using video processing,” 2014 Recent Adv.

Eng. computer. Sci., pp. 1–6, 2014.

AnyLogic (2021) Road Traffic Simulation Software. Available at: <https://www.anylogic.com/road-traffic/> (Accessed: 23 June 2021)

Geek Interview (2008) What is Data Derivation. Available at: <http://www.learn.geekinterview.com/data-warehouse/dw-basics/what-is-data-derivation.html> (Accessed: 12 July 2021)

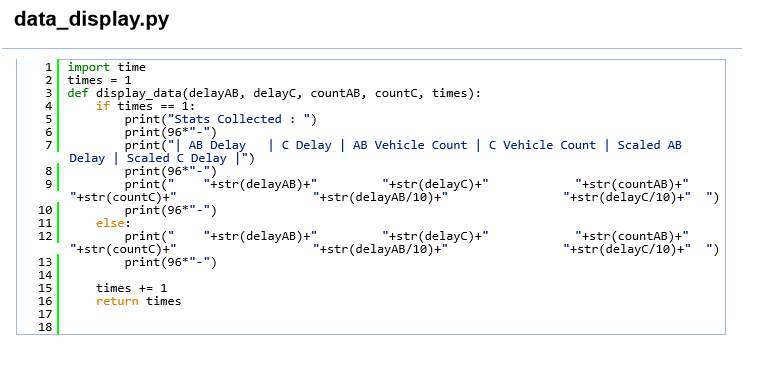
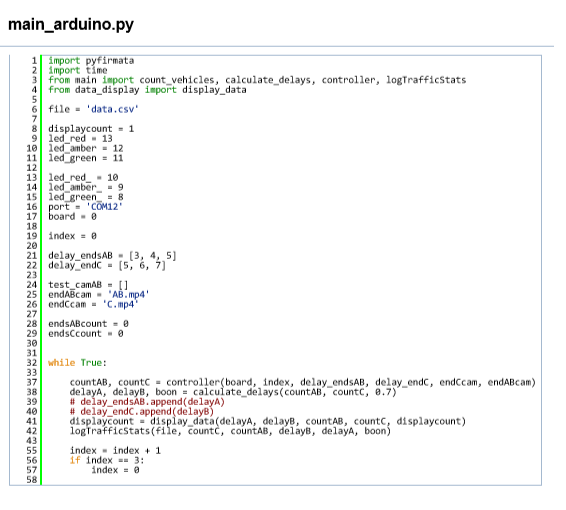
Ramakrishman (2012) System Analysis and Design. Available at: <https://www.longdom.org/open-access/system-analysis-and-design-2165-7866.S8-e001.pdf> (Accessed: 26 June 2021)

Ben Davis (2021) What is the role of system analysis and design? Available at: <https://www.mvorganizing.org/what-is-the-role-of-system-analysis-and-design/#What_is_the_main_purpose_of_system_analysis> (Accessed: 3 August 2021)

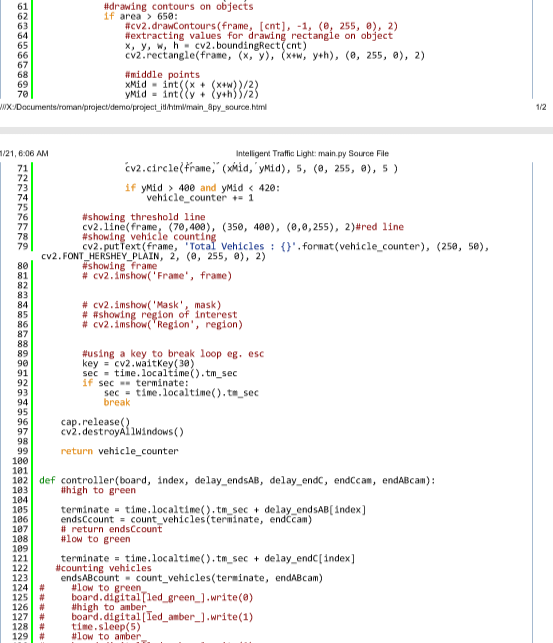
Nancy R. Tague (2005) The Quality Toolbox, Second Edition. Available at: <https://asq.org/quality-press/display-item?item=H1224> (Accessed: 3 August 2021)

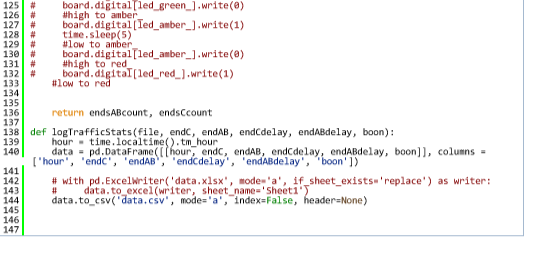
Ulrich & Eppinger (200). Product Design & Development. Irwin McGraw-Hill. Available at: <https://en.wikipedia.org/wiki/Systems_design> (Accessed: 12 August 2021)

**Appendix**

****

****

****

****