**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**

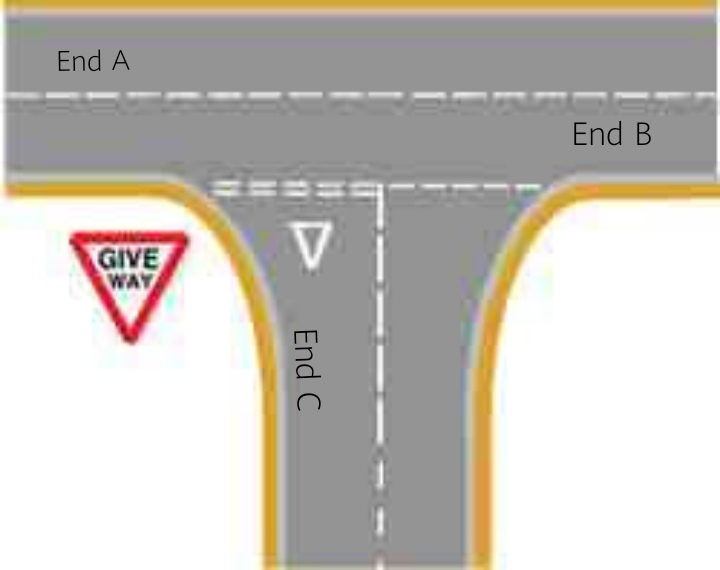
The below table contains data collected observing T-junctions

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**

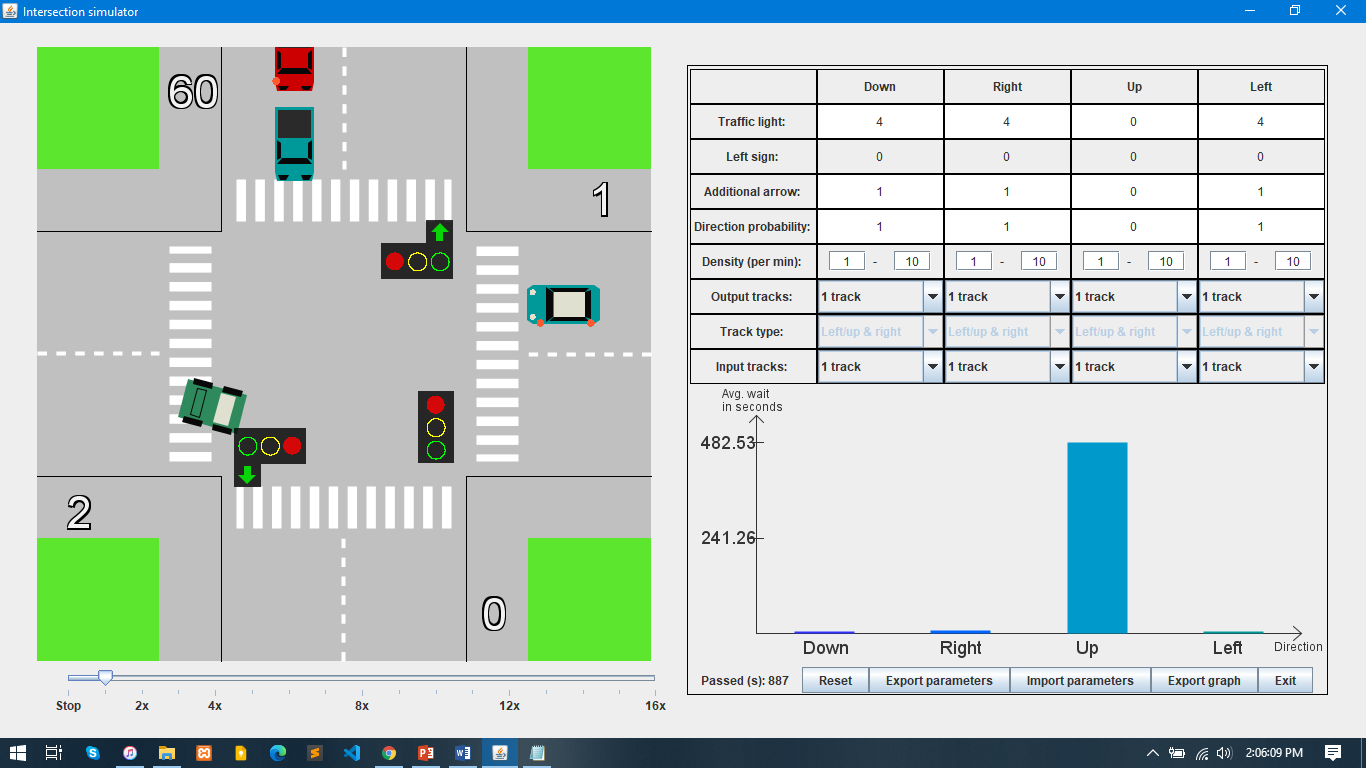
**Simulation**

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

**Experimental Data**

The following data was collected from the traffic light algorithm tests

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

* X
* Y
* Z
* A
* C

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.

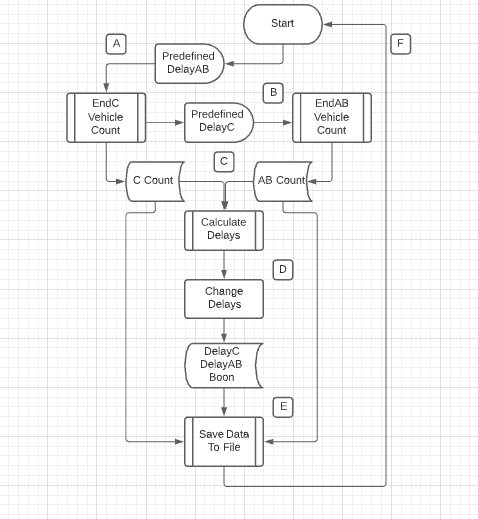
The process to be diagrammed involves the major objectives of the system, and is abstractly called process\_nameX.

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

* 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).

* 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).

* 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.

* D

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

* 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.

* 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.

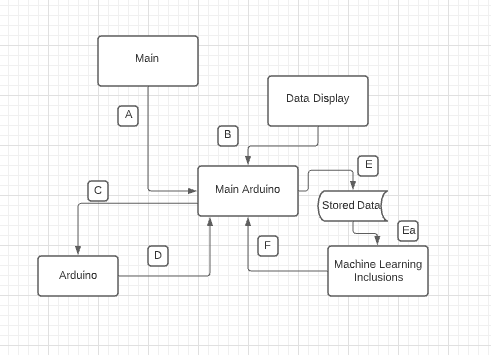
Architectural Design

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.

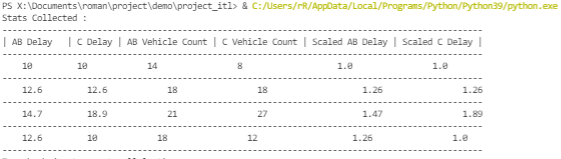
Physical Design

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.

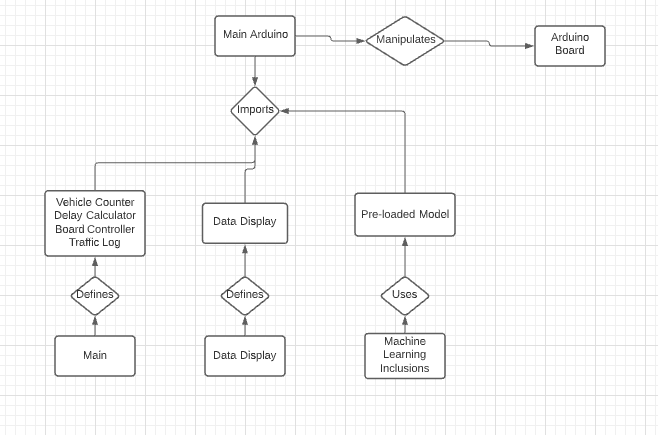
AB Vehicle and C Vehicle Count as illustrated above are values returned from the Vehicle Counter method.

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.