



Department of Electrical and Computer Engineering

**Microcontroller System Design
EGC331**

Design of a Traffic Control System

Group Members	Department
Nikiforos Fokas	EE
Kyle Harkin	CE

Instructor: M. Otis

Abstract

Since the invention of the automobile, directing the flow of traffic has become of utmost importance. Therefore, having learned how to create traffic lights in EGC331, we wanted to tackle this issue as our final project. Our fictional road system is diverse and contains a railroad and overpass in addition to intersections and pedestrian crossings. Our road system mimics a very real possibility and one which may be seen in the real world. To control the flow of traffic, we used several microprocessors and knowledge learned from EGC331.

Table of Contents

Objective	3
Design Procedure	3
Road System.....	3
Components and Devices	6
Flowcharts.....	7
Implementation	12
Verification.....	14
Conclusions and Problems	15
References	15

Objective

The objective of this final design project was to create a functional traffic control system given a fictional road system.

Design Procedure

Road System

In order to design a method to control traffic flow, we needed a road system on which to base our design. The road system we devised was comprised of two intersections, a railroad with a road crossing, and an overpass. An illustration of our road system can be seen in Figure 1.

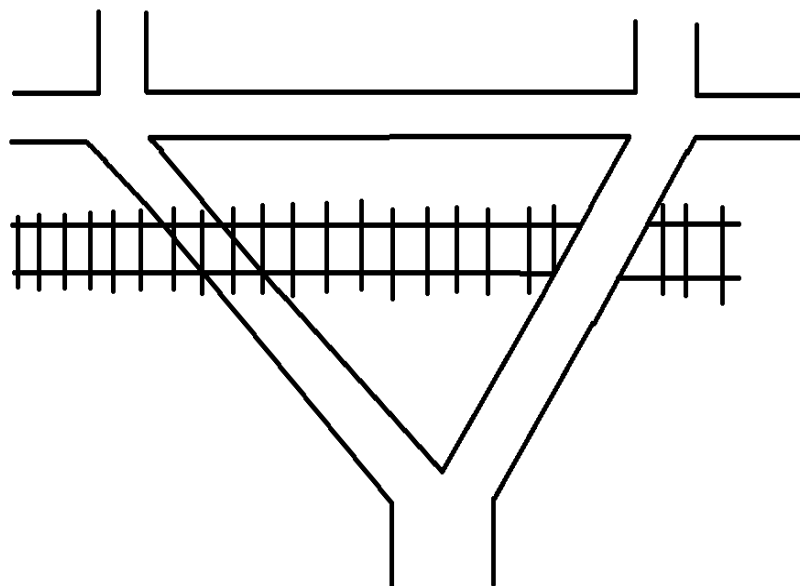


Figure 1. Road System

Further breaking down our road system, the two roads which interact with the railroad are one-way streets, while the rest are bidirectional. Figure 2 illustrates the directions of the roads shown in Figure 1.

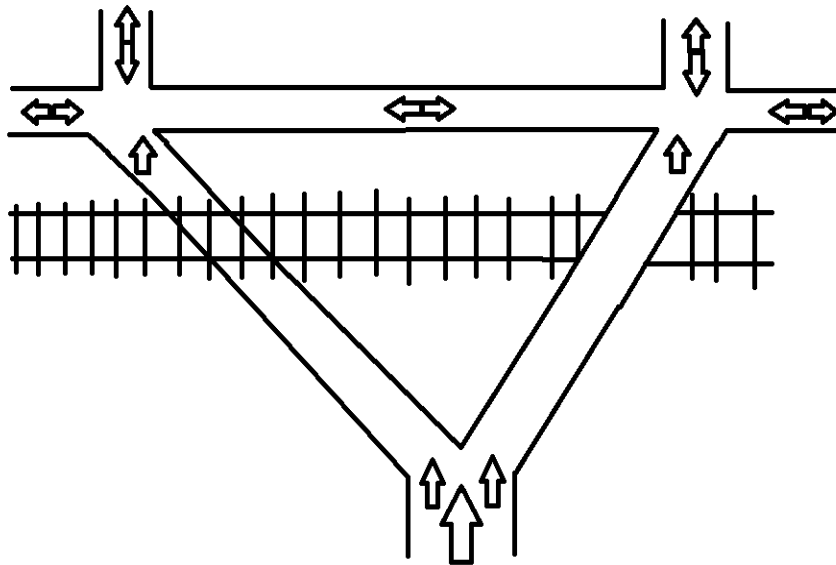


Figure 2. Road Directions

In Figure 2, the vertical arrows represent north-south (NS) traffic while horizontal arrows represent east-west (EW) respectively. To facilitate ease of reference, our road system is divided into several sections. The sections, which are numbered, can be seen in Figure 3 with their designated names in Table 1.

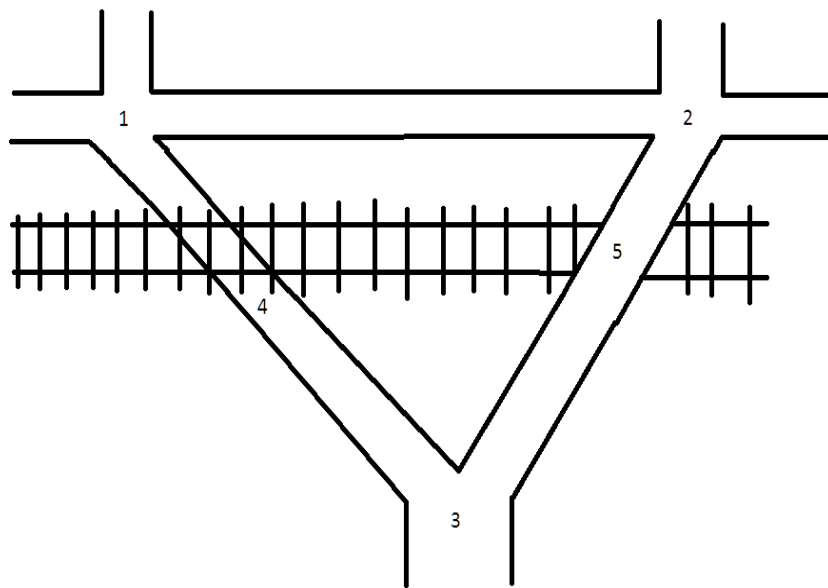


Figure 3. Road Sections

Section Significance

Number	Name
1	Intersection 1 (IS1)
2	Intersection 2 (IS2)
3	Fork
4	Rail Crossing
5	Overpass

Table 1.

Intersections will have their roads referred to as the direction which they travel followed by the corresponding number. For example, NS1 signifies the NS roads for IS1. Having broken down the road system, next comes the components of the traffic control system.

Components and Devices

To implement a traffic control system, we must understand the components for how the system will control traffic. Intersections IS1 and IS2 will be controlled via traffic lights. Additionally, each intersection will have pedestrian crossings, with times shown by seven segment displays, to allow foot-traffic. The intersections will have sensors embedded in each of their four respective roadbeds to allow intelligent control of the traffic light pattern. The rail crossing will contain sensors in the railbed to cause blockage of the crossing to traffic in the event of an oncoming train. Lastly, the fork will contain a sensor to determine if there is a backup due to a sustained NS1 red light. A liquid crystal display (LCD) screen will be used to divert traffic to either of the two road options at the fork, given the circumstances. Sensor locations are shown in Figure 4.

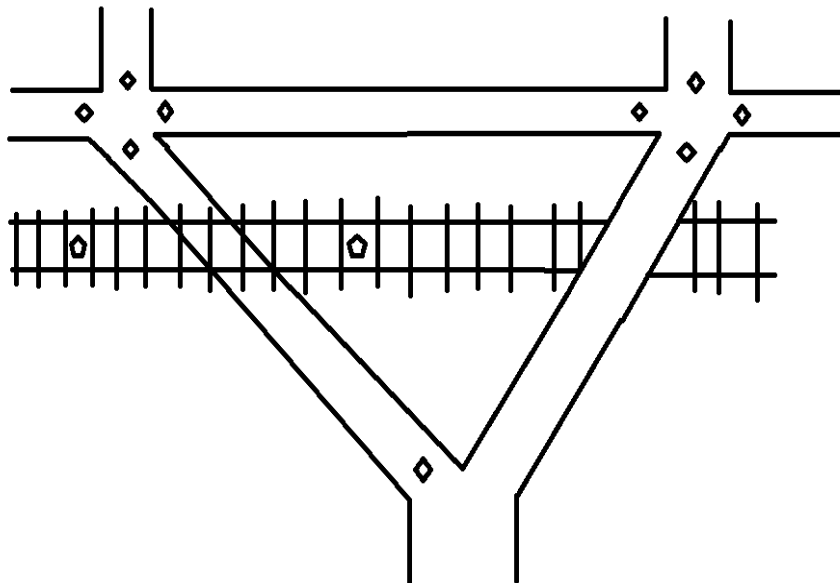


Figure 4. Sensor Locations

Sensors will be referred to as their location followed by “sensor.” For example, NS1 sensor. The NS, EW, and rail sensors will be chained together so that they work in tandem. For example, both NS sensors in IS1 are referred to as NS1 sensor. Likewise, both EW sensors in IS2 are referred to as EW2 sensor. Therefore, despite there being eleven sensors in total, they operate as six. Having described the sensor locations, the flowcharts which describe operation may now be understood.

Flowcharts

The flowchart for the intersections IS1 and IS2, pedestrian crossings, rail crossing, and traffic control system are shown in Chart 1, 2, 3 and 4 respectively.

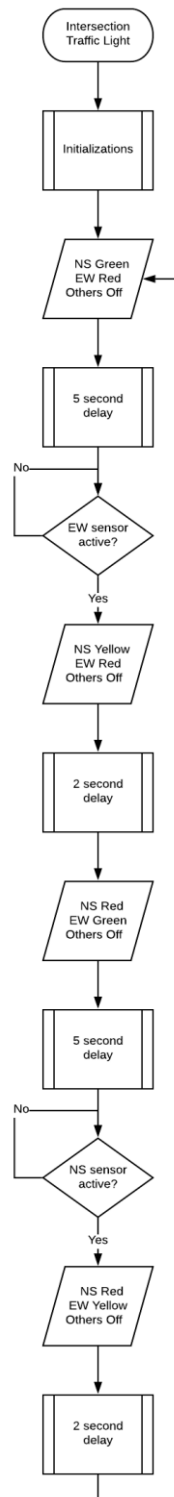


Chart 1. Intersection Flowchart

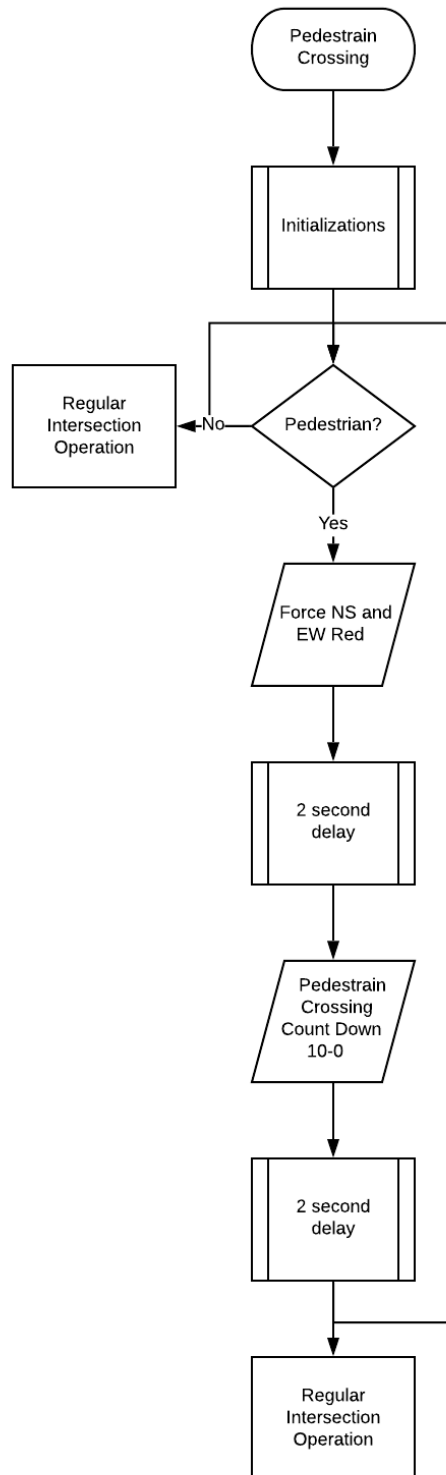


Chart 2. Pedestrian Crossing Flowchart

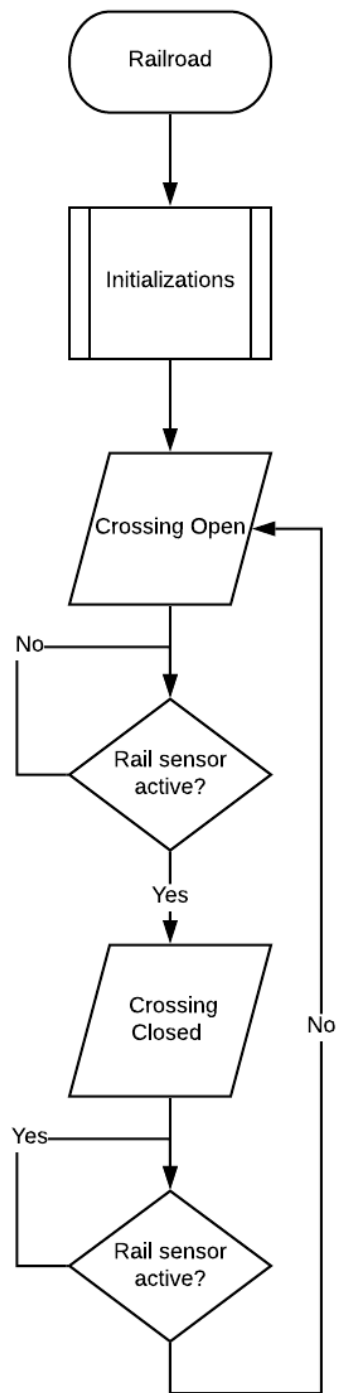


Chart 3. Rail Crossing Flowchart

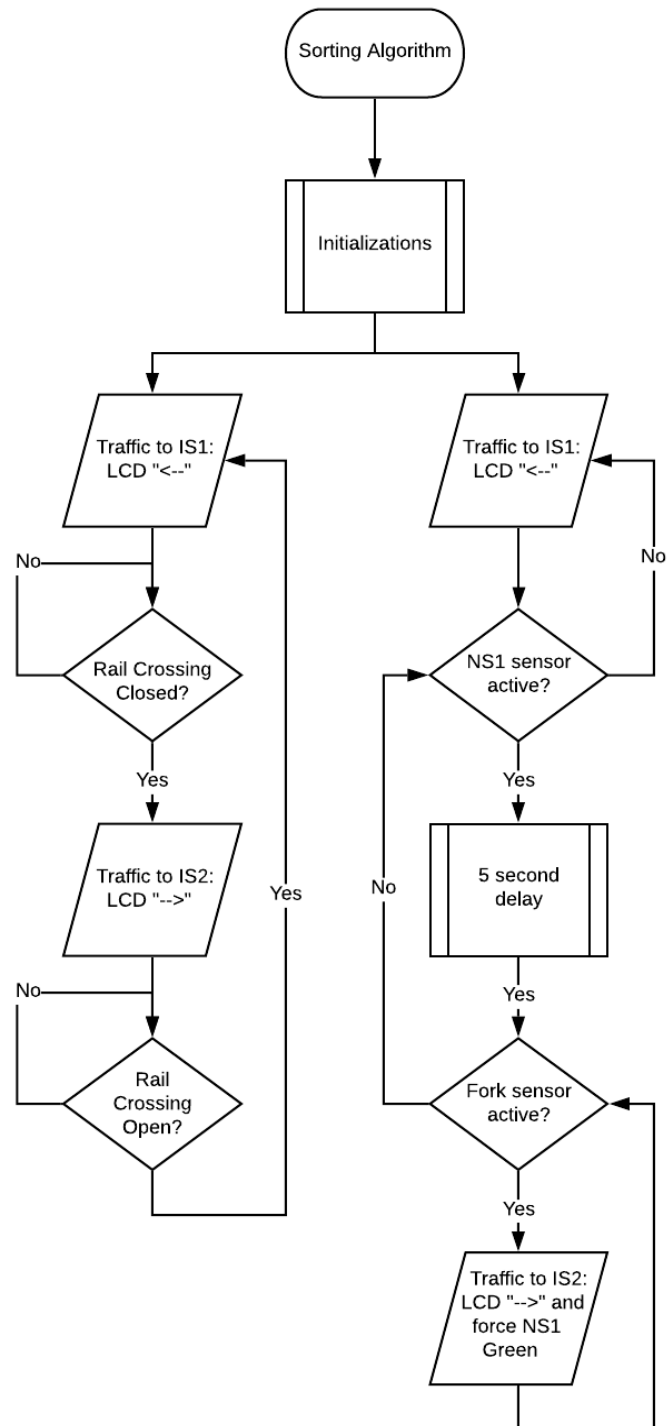


Chart 4. Traffic Control System Flowchart

An important thing to note is that the sorting algorithm for the traffic control system gives a weight to each branch. In other words, the left (rail) branch is given higher priority than the right (backup condition) branch. For example, if a train were to come while there was a backup, the backup condition would be ignored, and the rail condition would be executed.

Implementation

Implementation of the traffic control system was undertaken using several microprocessors which are as follows, two TI Tivas, one Arduino, and two AT Tiny85s. The Arduino was used for analog to digital conversion for the sensors, which gave analog outputs, such as our fork sensor. An overall summary, or mind-map if you will, of how the boards were implemented with various routines can be seen in Figure 5.

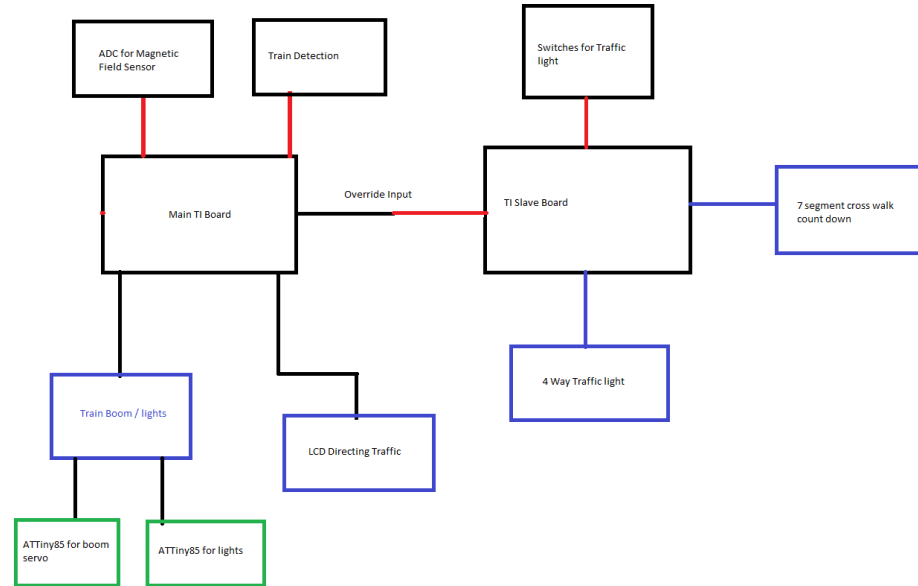


Figure 5. The Mind Map

Verification

The physical construction of Figure 5 is shown in Figure 6.

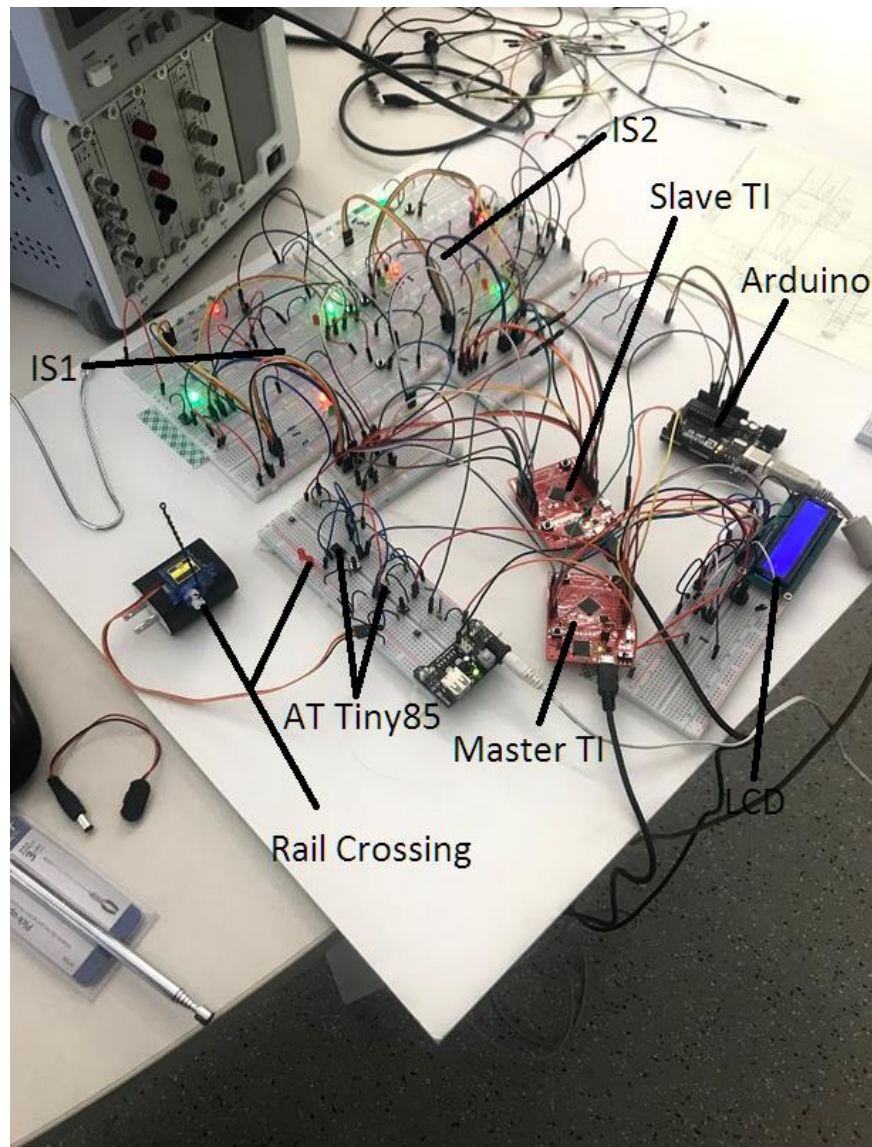


Figure 6. Constructed Circuit

Conclusions and Problems

Having been faced with the task of using a traffic light, seven segment display, and LCD screen together as a final project, we decided to create a traffic control system. We created a fictional, yet realistic, road system on which to base our project design. Using several microprocessors and analog components, we used our knowledge from EGC331 to implement our design. One problem which we faced, yet overcame, while implementing our design was the communication between different microprocessors. To solve this issue, we used a few pins on each board to act as a communication line. This way, the boards could communicate with each other and use outputs from one as inputs for the other. Another problem that we face, yet were unable to overcome, was the issue of power. Having such a plethora of devices, such as capacitors, sensors, servo motors, and light emitting diodes, our TI Tivas were simply not able to power them all. We tried to rectify this issue by using a separate “power board” to boost the voltage and current available. However, this did not prove to be a capable way of powering the entire circuit. Although the entire circuit was not able to be fully powered, it was powered enough to demonstrate the success of our design. Improvements to our design would include finding a way to power the entire circuit from just the TI Tivas and minimizing power usage.

References

[1] M. A. Mazidi, S. A. Chen, S. A. Naimi, and S. A. Naimi, *TI Tiva ARM Programming For Embedded Systems*. 2014.

