The Traffic Light System

David Shim, Sean Yang, Sophia Yang
TEJ3MP
Mr. Wong
April 17, 2021

a) Introduction

This project is a traffic light system with a T-intersection, a street light, a pedestrian light, and a gate that opens when a car approaches it.

This traffic light system was created to simulate a real-life street intersection and to learn the different ways we can use arduinos and other components to mimic realistic systems. It also shows how computer engineering and coding can be used to help create and run structures that we use everyday.

This system was created by David Shim, Sean Yang, and Sophia Yang.

This system was created by simulation softwares called SolidWorks and TinkerCAD. SolidWorks is a computer program that can be used to create 3D models. It was used to create a plan and a visual representation of the intersection. TinkerCAD is a modelling program that can be used to create circuits. It was used to create the arduino, wiring, and other components. It was also used to write and test code.

b) Explanation

There are 3 traffic lights at the T intersection and the light is determined by the pedestrian walk button on the pole of the model. The timer on the traffic lights is naturally switching on its own. It switches every 6 seconds but can be changed easily.

The pedestrian light works by turning green when the traffic light parallel to it is also green. The light timing is the same as the traffic light, however there is no yellow light for the pedestrian walk. So the timing will also change along with the traffic light and is the same for all sides.

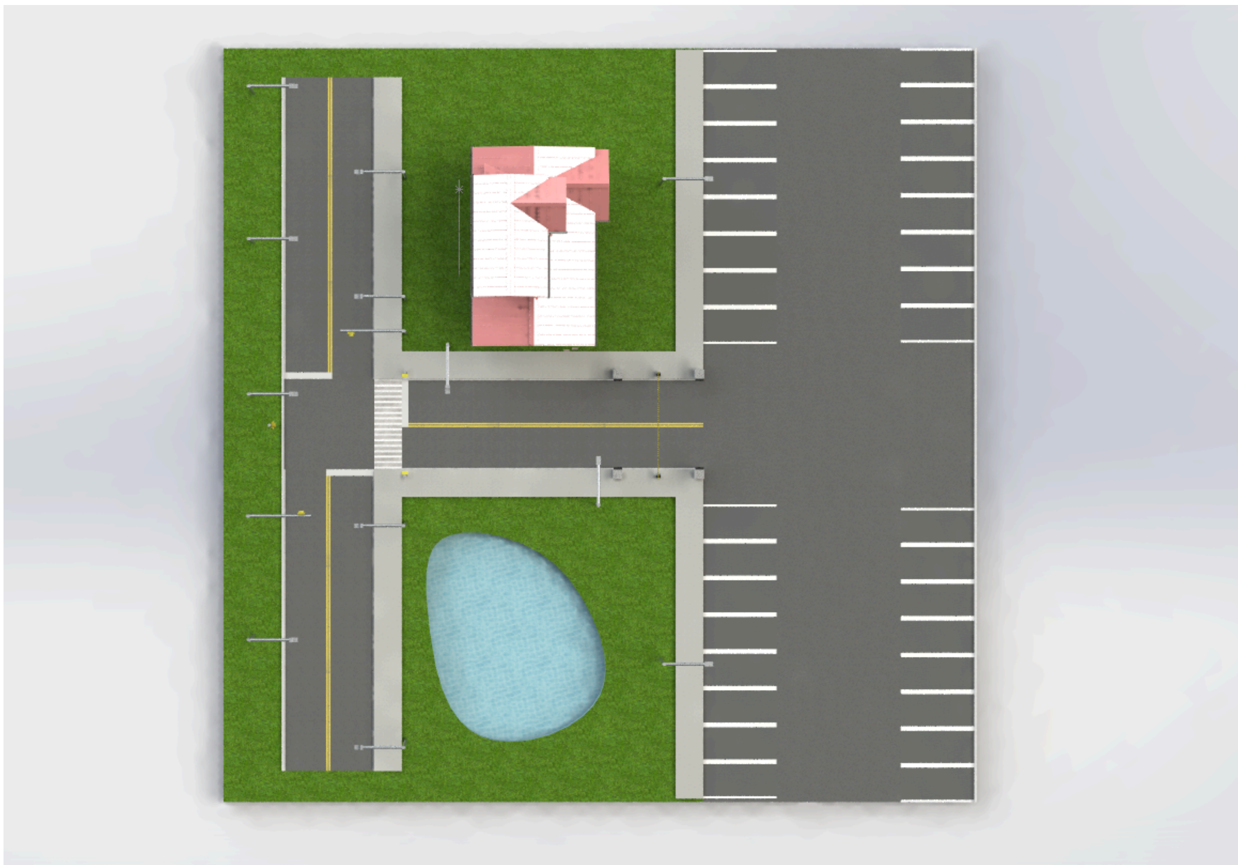
The manual switch tells the traffic and pedestrian lights that there are people waiting to cross. It then shortens the natural timer by half. Explanation of this would be if the normal time was 10 seconds till change, the button will shorten it to 5 seconds. It will then revert back to normal after the lights switch back to red.

The street lights will be turned on/off when the sensor detects that the room is dark and will proceed to activate the lights. If the room is bright then the lights will proceed to turn off.

The ultrasonic sensor is used to measure the wind speed and direction, As well as tank or channel fluid levels, and speed through air or water. It uses detectors to measure the units that we are trying to calculate. The system uses a transducer which generates ultrasonic sound waves. It turns the electrical energy into sound which can be measured. It can also be used to make point to point distance measurements.

The IR sensor works with the gate by measuring the amount of light radiating from the car light in the sensor's field of view. The IR will detect that there is a general movement, which in this case it will sense the car moving and will cause the gate to open by sending information to the gate that there is something moving.

c) Layout



Challenge

One of the biggest challenges we faced had a lot to do with organization issues. Sometimes we would overlook or forget about a certain instruction and have to spend time going back and changing our work to meet the requirements. This was not only time consuming, but also very confusing and overall not a good way to approach any problem.

This happened because we didn't spend enough time at the beginning going through every instruction and outlining what needed to be done. Without organized lists, it's easy to misread or forget about a certain requirement. Realizing at the very end that we missed something generally induces quite a bit of panic and confusion.

Another challenge we faced was being unfamiliar with many of the components in TinkerCAD. It would have helped if we had a brief introduction of the uses and purpose of the components that we would need. This way, we wouldn't have to spend time trying to figure out what each component did and what we needed to accomplish something.

Investigation

We started our project by creating our models in Sketchup and Solidworks. While creating our model, we made many changes such as adding decorations or deleting parts that were unnecessary. Many revisions had to be made during this step as we constantly messed things up as we tried to learn our way through the programs. Eventually we were able to create well polished models that contained all of the required components and design features.

Next, we created our circuit in TinkerCAD. We first began with a lot of experimentation that helped us learn how to use and code Arduino as well as the TinkerCAD environment itself. As we created our circuit, we focused heavily on making sure our circuit met all of the requirements and guidelines. We made many modifications in this step as we constantly adapted our ideas to fit the guidelines. For example, much of the wiring needed to be moved or replaced in order to fit the additional feature that we had to implement in our circuit. Finally, when we were finished with the basic design and coding of our circuit, we put it through rigorous testing to catch and fix any bugs that remained. This part also necessitated some modifications of both the circuit and code that we found were causing too many problems.

Evaluation

The next time we do a project like this, we would start by planning out the process to prevent mistakes and better organize our tasks. We should also take more time to understand the instructions. This way, we could avoid misunderstandings later on. This would really help speed up the process and we could have saved ourselves a lot of stress.

Another thing we could have done differently was better manage our time. Without a general outline of the tasks ahead of us, it's difficult to plan ahead and figure out how much work we had to do before we could hand it in. Although we didn't leave everything to the last minute, we still should have worked to finish a little earlier to leave room for corrections and final touch ups.