Final Project

Title

Crosswalk Simulator

Course

CSC-11

Section

42129

Date

June 4th, 2020

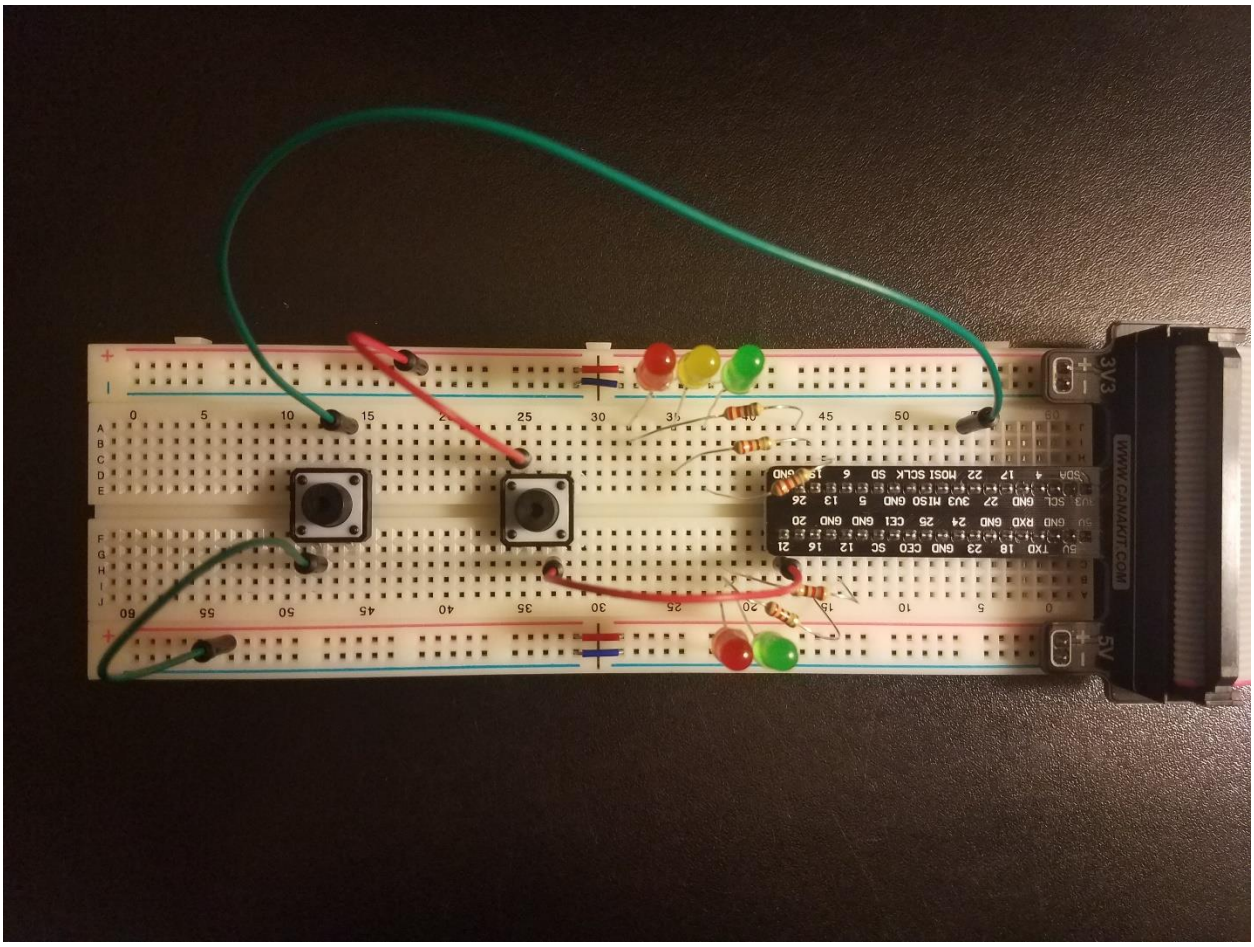
Author

Brandon Sanchez

Introduction

This crosswalk simulator program utilizes a breadboard with LED lights attached to a Raspberry Pi 3. The code is written in ARM assembly and calls the PIN based GPIO wiring library called Wiring Pi library to control the lights. Originally, I had a hard time choosing a program for the final project but decided on the crosswalk simulator because it was the most familiar to me.

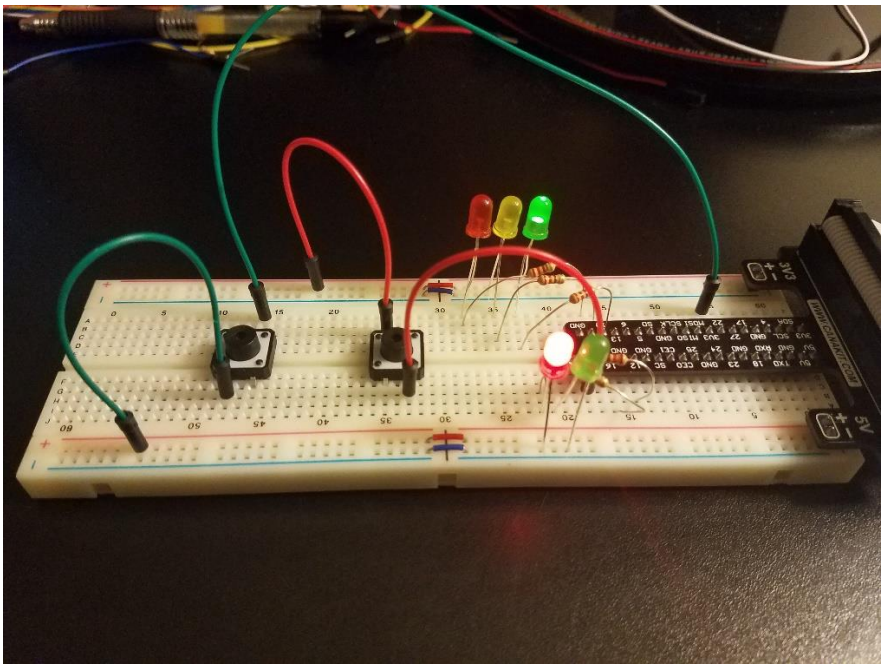
Top-Down View



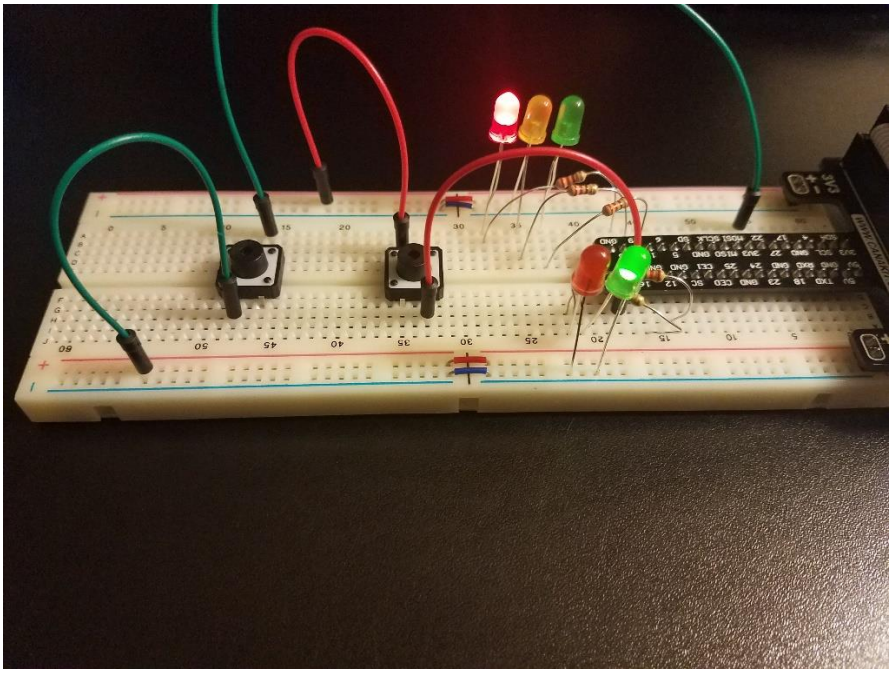
The black T-shaped GPIO breakout board on the right connected to the Raspberry Pi allows the program to control the lights and buttons. The red and green lights at the bottom are the crosswalk lights for pedestrians. The top 3 lights are the intersection lights for cars. The right button connected to the red wires can be pressed by pedestrians to cross. The button on the left connected by green wires allows the user to stop the program. All lights are connected to resistors to prevent burn out.

Demo

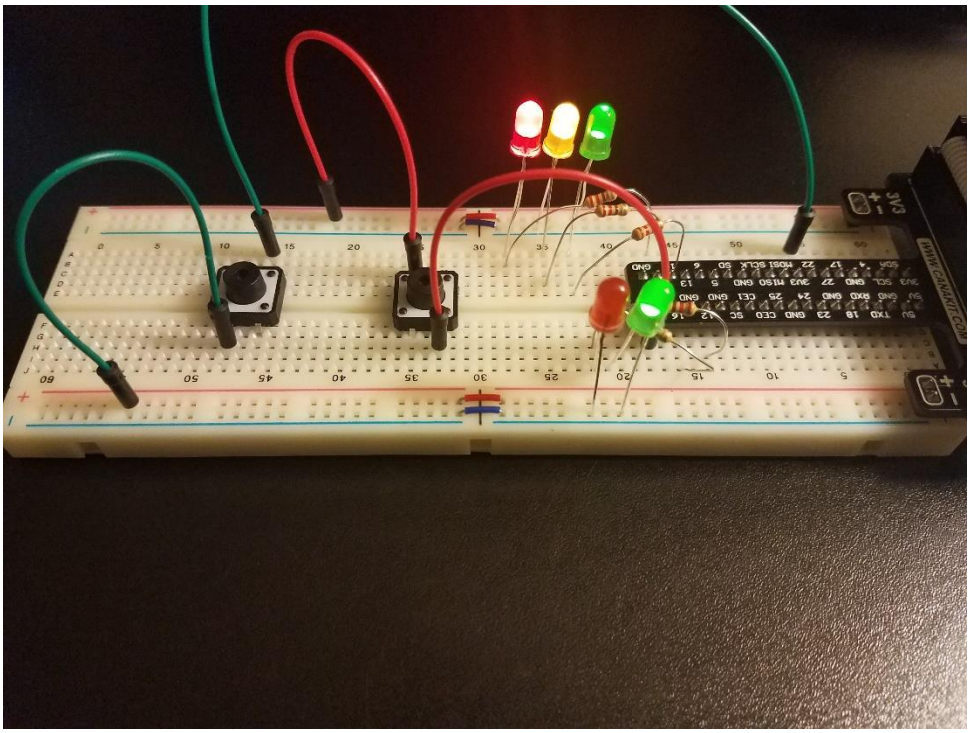
Video: https://www.youtube.com/watch?v=RNZ_fOvOdK4



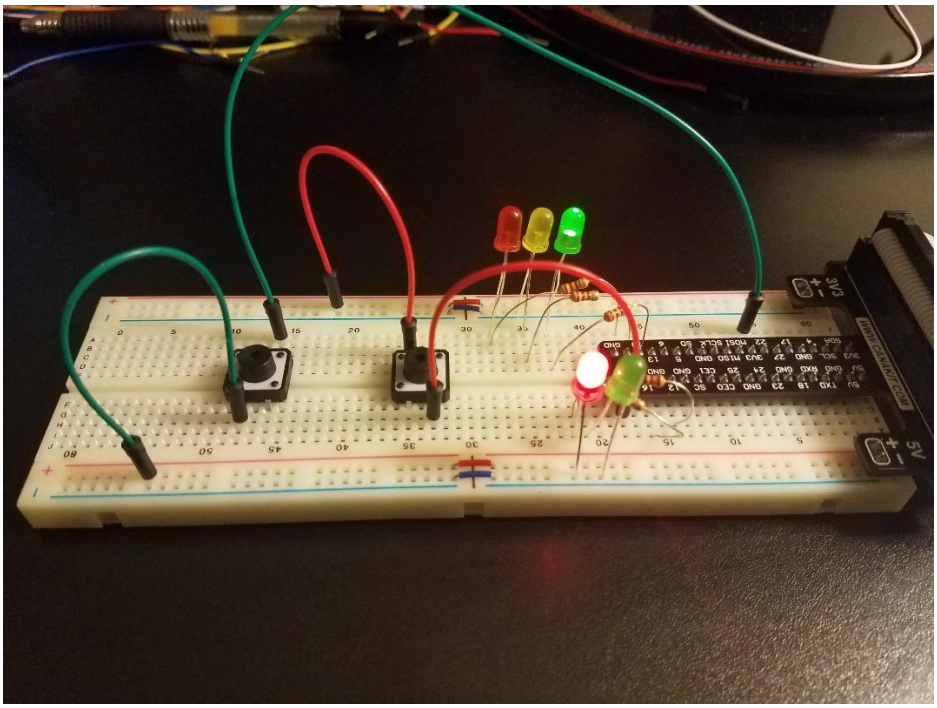
Once the program starts, the bottom red crosswalk light turns on to signal pedestrians not to cross. The intersection light at the top is green for cars to go.



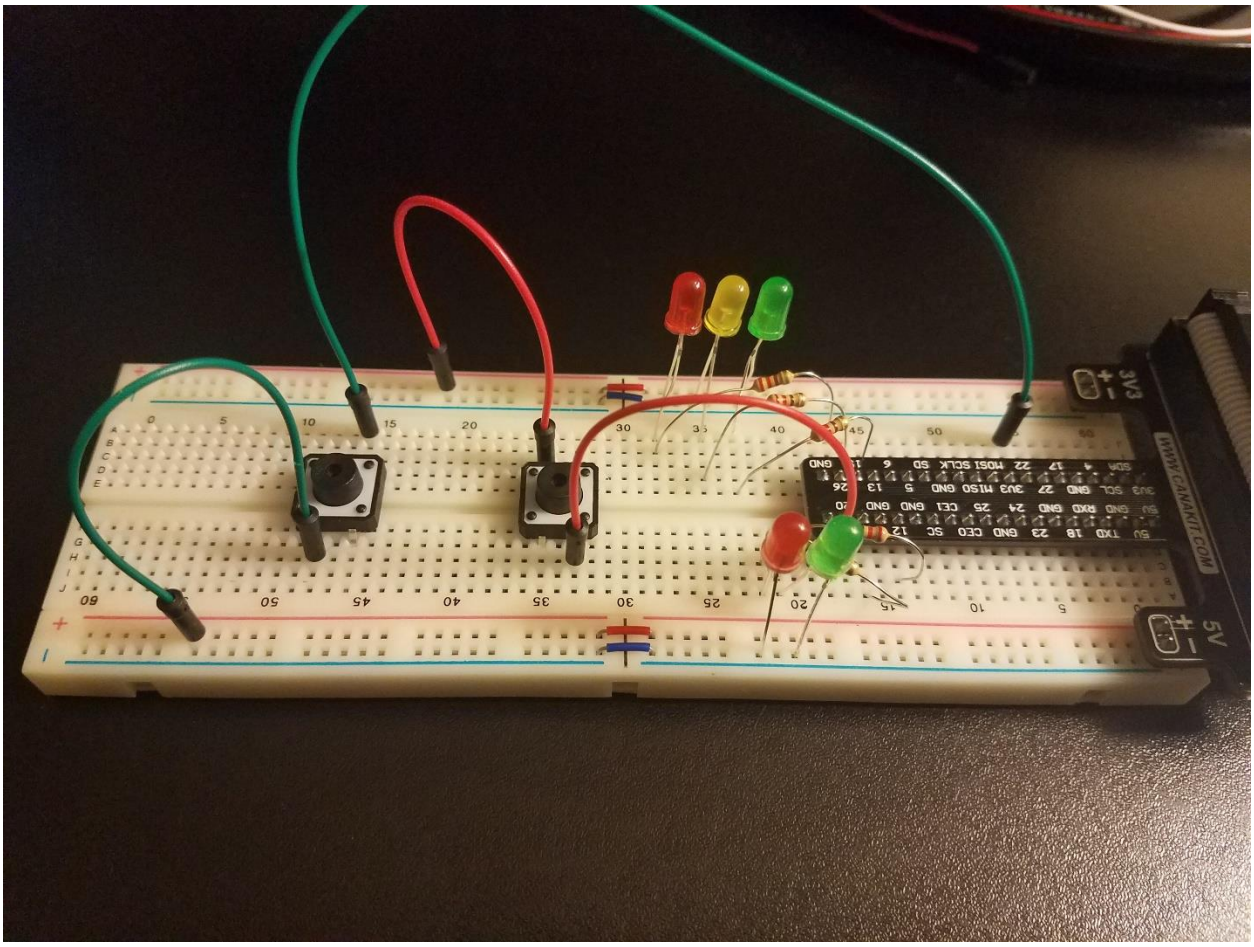
Once the right button is pressed, the pedestrian light turns green to cross and the intersection light is red. This lasts for a total of 10 seconds.



After the 10 seconds has passed, the yellow and green intersection lights flash to signal to pedestrians that the pedestrian light will turn to red soon. This continues for 5 seconds.



After the total of 15 seconds since the right button was pressed, the program returns to its original state. The user can press the right button however many times they want to repeat the cycle.



Once the left button is pressed, the program terminates, and all the lights turn off.

Programming Process

When programming, it is always a good idea to start out small to spot any bugs. I started out with only the bottom 2 crosswalk lights and the crosswalk button. I faced a challenge with the button not changing the lights properly. After implementing Professor Conrad's action function that makes use of the ctime library it was fixed. Then, the intersection lights were implemented with the right button changing the lights. Lastly, the terminate button on the left was placed to finish off the program.

I gained a better understanding of ARM assembly through this project. At the start, I was making many syntax and logic errors. Progress was slow but it got quicker as I became more skilled with the language. Overall, I enjoyed this project since I am new to the hardware side.

Pseudocode

Set respective pins for input/output

Turn crosswalk red light on

Turn intersection green light on

If user presses right crosswalk button:

Crosswalk red light off, crosswalk green light on

Intersection green light off, intersection red light on

After 10 seconds, flash intersection yellow and green lights

After 5 seconds of flashing, reset lights back to original state

If user presses left terminate button:

Turn off all lights

End program