

Project Report

DISCLAIMER: This project was inspired by Ordercube and was completed with educational purposes only. The actual product which inspired this project is the property of this company: <https://ordercube.ca/>

Introduction

This project was completed with the goal of using finite state machines in a way an engineer would. The original goal was to complete this without a microcontroller, but due to time constraints and lack of knowledge in electrical engineering, I resorted to using a microcontroller. The user in this case would be a restaurant, which could install this device to each table.

User Needs

- Simple
- User-friendly
- Can easily catch waiters' attention without distracting guests

The Idea

The Order Cube is made to be used in restaurants to prevent awkward interactions with waiters, like failing to get their attention. The cube is simple and **user-friendly**, consisting of only 2 buttons. One button is the **pay button** and sets the Cube to the color green. This alerts surrounding waiters that the table is ready to pay for their meal. The other button is the **service button**, which alerts the surrounding waiters that the table requires service. This could be a water refill or simply to order. Once the service button is touched, the Cube lights up blue and over a set and **customizable** period of time, gradually becomes purple to alert waiters that the clients have been waiting longer than they should have.

Solutions

The 2 buttons, situated on the top of the device, are simple. They are made of thin conductive metal that are used as capacitive sensors with the help of 10M ohms resistors and the *capacitivesensor* Arduino library.

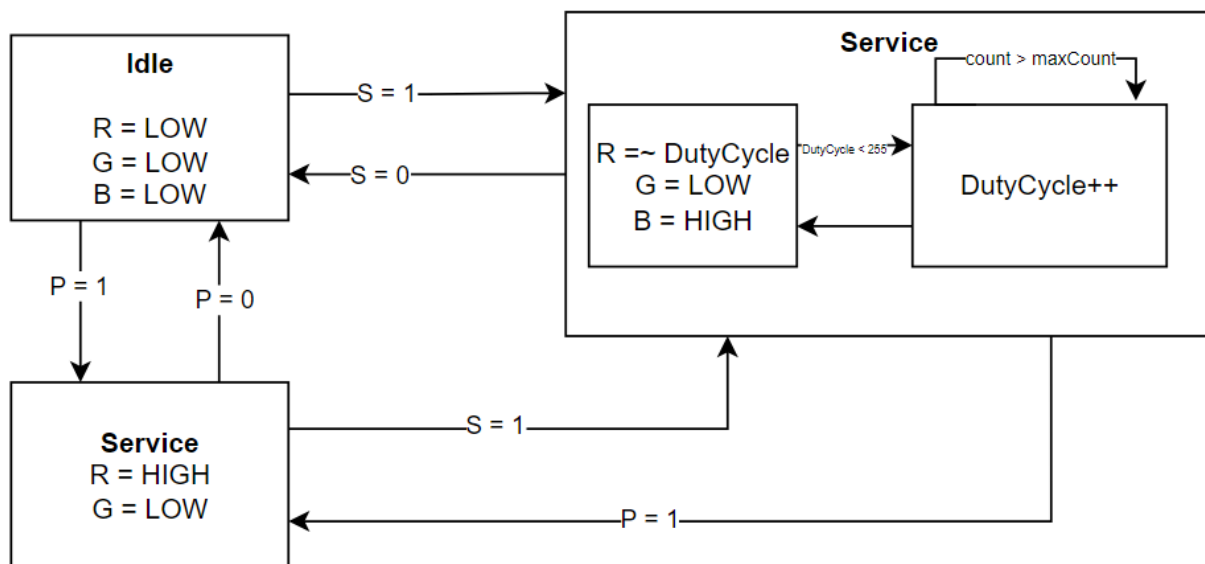
A 12 volt RGB LED strip of 5050 LEDs is used to light up the Cube. The R, G, and B pins are all controlled via PN2222 transistors and controlled by setting a voltage to the gate with the Arduino. Since the only color that is needed to gradually increase in brightness is red to produce the blue to purple color change that happens when the service button is pressed, it is the only one that is connected to an analog output.

The gradual red light is made by slowly increasing the duty cycle used in the AnalogWrite. The speed at which the duty cycle is increased is calculated with the following formula:

$$\text{maxCount} = (\text{blue_to_purple_time} / 255) / \text{avg_exec_time}$$

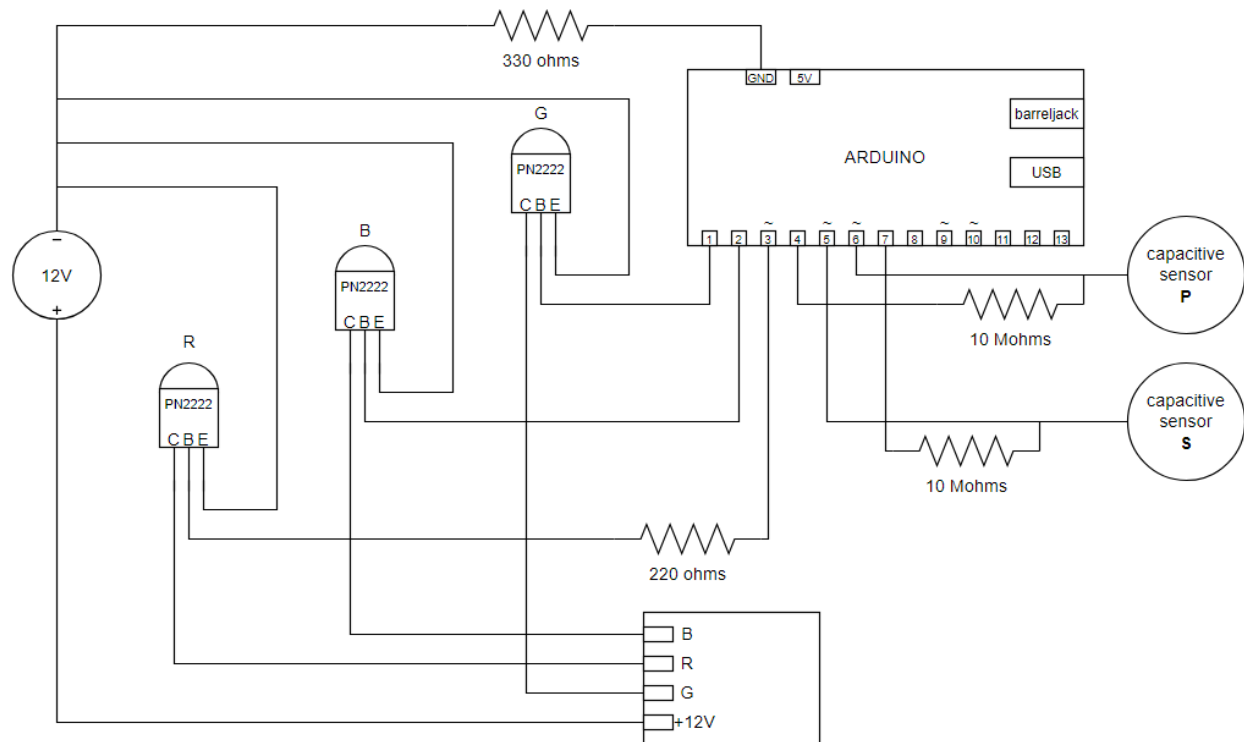
Where `blue_to_purple_time` can be set by the user (restaurant).

The FSM's state diagram is as follows:



Prototype 1

Due to time constraints, the prototype was done with materials found at home, which included translucent plastic and cardboard. It is also round,, but the final shape would preferably be a cube.



Conclusion

This was a fun and short project that made practical use of an FSM.