Traffic Light System

Miguel Antonio Rodriguez Delgado,
Marwa Mohammed Nabawey Hassan, Vincent Chinedu Obigwe,
Abeeb Opeyemi Nureni, Amit Chakma, Charles Arsenal Okere,
Hochschule Hamm-Lippstadt

November 2, 2020

1 Introduction

How would we able to design a spaceship if we cannot design a simple project first? The first step is always the most important along our lives, starting when we are babies; our fist day on school or even our first microcontroller project. During this Traffic Light System we will be able to understand the basics of microcontrollers, the way that we interact with it, how we can input and output information, and how we can realize and sketch a project in a microcontroller system such as the Arduino Uno. To help us with this we will make use of all the knowledge that we have acquire during the different subjects of our course studies up to now.

2 Basic Elements

To develop our Traffic Light System, we need to get familiar with the different components that we are going to use.

2.1 Microcontroller

A microcontroller is an integrated circuit that will allow us to execute some specific operations in a system that we will design. The main features that a microcontroller includes are a processor, memory, AD/DA converters, and input/output in a single device. [2]

The processor is the core of the microcontroller. It is the responsible to make all the arithmetic operations, and the logic of our device. The memory allocates all the information that the processor needs to execute the different task. The inputs allow us to provide new information from the user or from the ambience to the microcontroller. The outputs grant us a way to show the different states in our device or to give information to the user. The AD/DA converters transforms the signals from the ambience to digital signals that can be processed by the microcontroller and vice versa. [1]

2.1.1 Arduino Uno

The Arduino Uno is going to be the main element during this project, since is the “heart” of the Traffic Light System that we will develop. The Arduino Uno is a simple microcontroller that is based on a ATmega328P microprocessor. It also provides us 14 digital pins that we will be able to use [3]

The Arduino Uno provide us 12 digital pins the ones we can use for general purposes as inputs or outputs via the functions pinMode(), digitalRead() and digitalWrite. [4] When the pins are configured as an output in digitalWrite mode, the voltage output is 5 volts. [5] Moreover, we have de power pins that deliver 5 or 3.3 volts depending on out requirements. [4] Finally, we have the External Interrupts in pins 2 and 3. We can configure this pins as inputs connected to the attachInterrupt() function to read a input automatically while we are in any moment of the program. [6]

2.2 Light Emitting Diodes (LED)

According with the datasheet, the LEDs work at an absolute maximum of 25 mA of continues forward current. This value is going to be important to calculate the resistors that we must use for the project.

2.3 Resistors

According with the maximum current on our LEDs we calculate the value of our resistors based on Ohms Law in the following way:

$$Voltage = v = 5V$$

$$Current = i = 25mV$$

$$R = v/i$$

$$R = 5v/25mA$$

$$R = 200\Omega$$

To ensure that we are not going to damage the LEDs we decide to use 220Ω resistors for.

To connect the buttons we need to ensure that the voltage is going to flow through the wire connected to the input. To achieve this we decide to use $10k\Omega$ resistors.

2.4 Buttons

Another important element that we are going to use are buttons. They will be representing both the buttons that can be pressed by the pedestrians, as well as the internal buttons for the bus driver.

3 Circuit

To implement this circuit, we started with the electrical circuit (Figure 1), and then we took base in the pins that the Arduino offer us, and we assigned the pins according the information in section 2.1.1 and in the following order:

Pin 2: Pedestrian button.
Pin 3: Bus driver interrupt.
Pin 6: Notification LED for the driver.
Pin 7: Notification LED for the pedestrian.
Pin 8: Red traffic light.
Pin 9: Yellow traffic light.
Pin 10: Green traffic light.
Pin 11: Green pedestrian light.
Pin 12: Red pedestrian light.

Also, we use connect the project to the ground and with the 5 volts output of the Arduino (Figure 2).

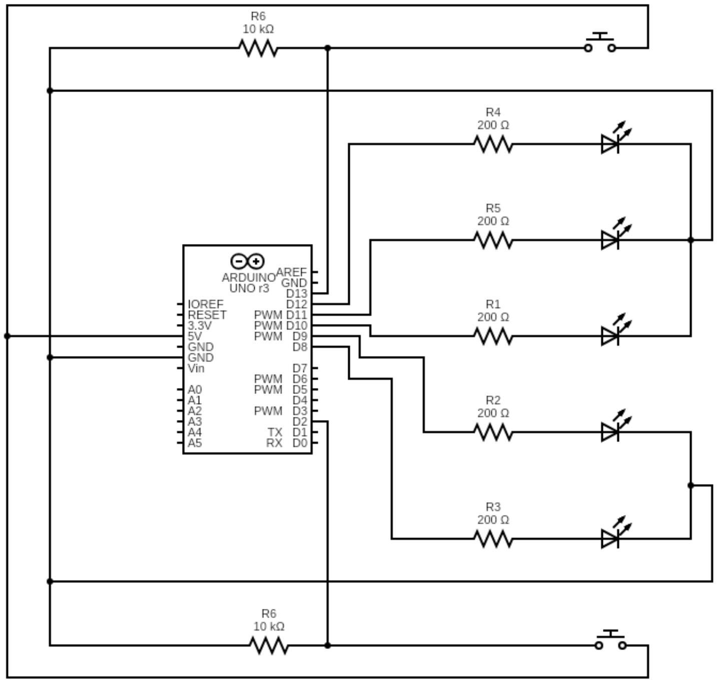

Figure 1: Electric Circuit

4 UML

The next step in the design of our Traffic Light System was to develop the Class Diagram and the State Machine Diagram based in the knowledge acquired in the last semester (Figures 3 and 4).

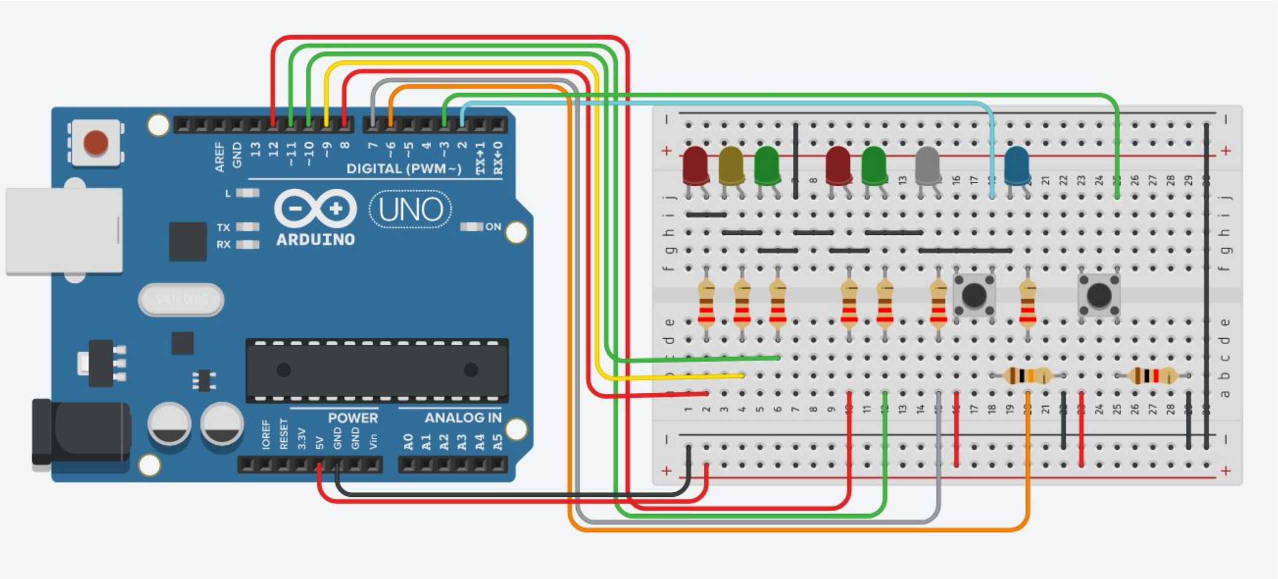

Figure 2: Arduino Uno connections

5 Coding

To finalize our project we mapped the code to the Arduino using the different functions that are available for us. For instance, *void setup()*, function that is used to initialize the Arduino and the pins that we are going to use, and *void loop()* that is the function that will execute the program and change the lights of our system.

Moreover, we use the function *attachInterrupt()* to read the inputs from the pedestrian and the bus driver at any moment during the execution of the program.

The *void loop()* function was programmed with a *switch case* that evaluate the status of the Traffic Light System based in a new type of variable that we created via a enumeration.

6 Conclusion

To develop a micro-controller project we need to pay attention to many details. If we miss one of the steps, even if it looks small or insignificant, can guide us to an error or a misunderstanding during the execution of the project. Is very important that we stick to each part of the process, starting with the understanding of the elements that we are going to use, going to a correct design on paper of the electronic circuit, creating a detailed Class Diagram and State Machine Diagram to be able to map all the ideas that we have from the beginning to a final result that can represent all the requirements, and the ideas that we have to develop it.

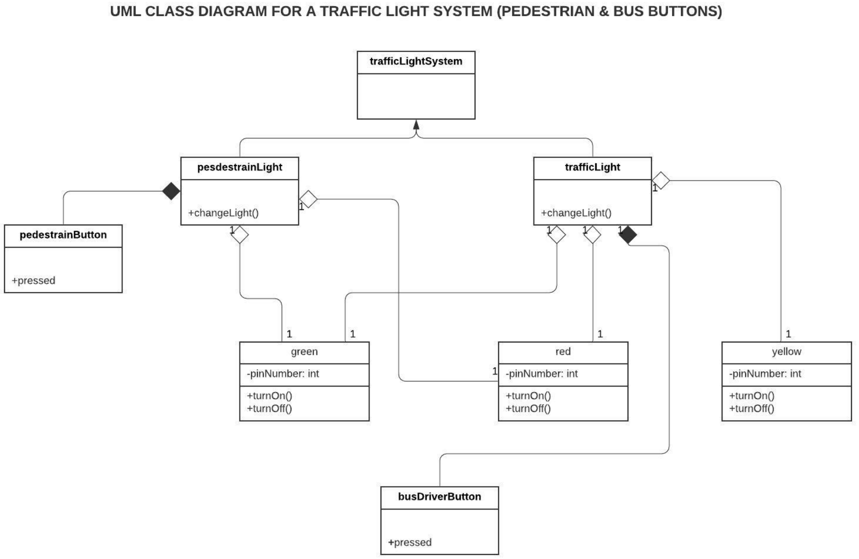

Figure 3: Class Diagram

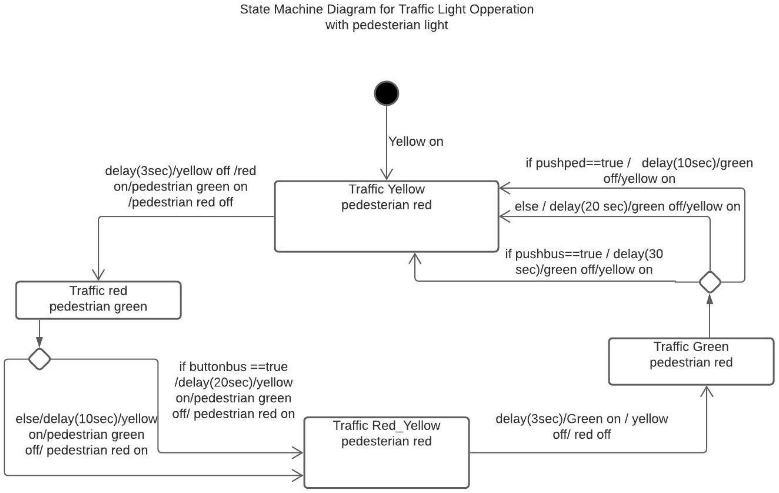
State Machine Diagram for Traffic Light Opperation
with pedesterian light
Yellow on
Traffic Yellow
pedesterian red
delay(3sec)/yellow off /red on/pedestrian green on /pedestrian red off
if pushped==true / delay(10sec)/green off/yellow on
else / delay(20 sec)/green off/yellow on
if pushbus==true / delay(30 sec)/green off/yellow on
Traffic red
pedestrian green
Traffic Green
pedestrian red
if buttonbus ==true /delay(20sec)/yellow on/pedestrian green off/ pedestrian red on
else/delay(10sec)/yellow on/pedestrian green off/ pedestrian red on
Traffic Red_Yellow
pedesterian red
delay(3sec)/Green on / yellow off/ red off

Figure 4: State Machine Diagram

References

[1] Stefan Henkler *Microcontrollers* Hochschule Hamm-Lippstadt. 2020

[2] Margaret Rouse, Sharon Shea, Ben Lutkevich. *microcontroller (MCU).* https://internetofthingsagenda.techtarget.com/definition/microcontroller November 2019

[3] Arduino Uno Rev. https://store.arduino.cc/arduino-uno-rev3.

[4] Introduction to the Arduino Board https://www.arduino.cc/en/Reference/Board

[5] digitalWrite(). https://www.arduino.cc/en/Reference.digitalWrite

[6] attachInterrupt(). https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/