INTERNATIONAL UNIVERSITY OF SARAJEVO

Lab Report *VIII*

Project:

Arduino Traffic lights

ENS203 Electrical Circuits I

Sarajevo, Spring 2022

Table of Contents

Table of figures

Objectives

The ojective of this lab was to build a project of our choice, plan and implement it during our lab.

PART I – *Arduino traffic light*

1. Statement of the Problem

We will be implementing a street model traffic lights simulation using the Arduino Uno microcontroller.

2. Materials

1. Arduino uno microcontroller
2. LED: 2 x green, 2x red, 2x yellow
3. Resistors: 6x 220 Ohms
4. Arduino USB 2.0 data cable
5. 2x Breadboard
6. Jumper wires

3. Diagrams and Figures

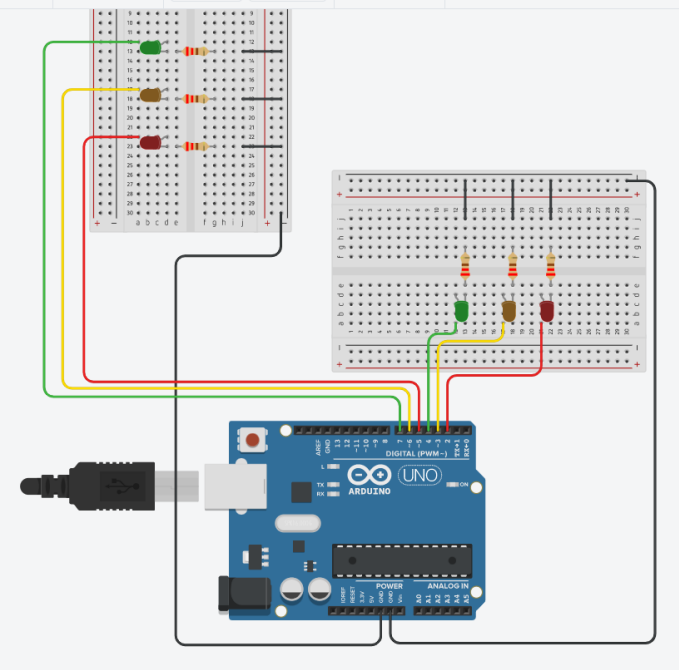

Figure 1: Tinkercad simulation guide

4. Procedure

First, we will get introduced to the microcontroller we will be using today. Arduino Uno is a programmable microcontroller board used for multiple purposes. It includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, etc. The USB connector is used to connect it to a laptop on which we will use the Arduino IDE. We will write our code in the programming language C++, and through our USB connection to the board will be supplied with voltage.[1] For our project we will be using the USB connector and the digital input/output pins.

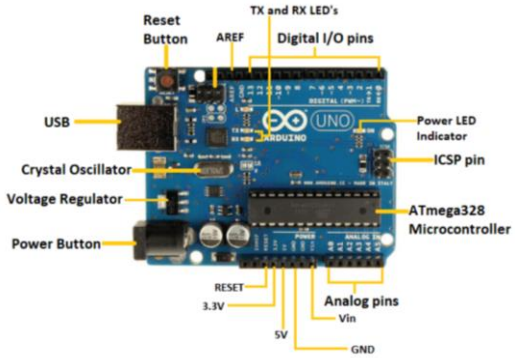

Figure 2:Arduino uno pinout ("Arduino UNO - JavaTpoint")

Now, on to the idea behind the traffic light. We imagined a crossroad with 4 traffic lights- since opposite directions have the same signals on, we will focus on 2 lights- on perpendicular streets.

Our main design ideas are based on state machines, therefore we implemented states in our design which will be a repeating sequence. The easiest way to describe our work is through a table.

Table 1: traffic light states

state	Green One	Yellow One	Red One	Green Two	Yellow Two	Red Two
one	OFF	OFF	ON	ON	OFF	OFF
two	OFF	OFF	ON	OFF	ON	OFF
three	OFF	OFF	ON	OFF	OFF	ON
four	ON	OFF	OFF	OFF	OFF	ON
five	OFF	ON	OFF	OFF	OFF	ON
six	OFF	OFF	ON	OFF	OFF	ON
seven	OFF	OFF	ON	OFF	OFF	ON

[1] ("Arduino UNO - JavaTpoint")

Since there are some countries that have traffic lights that go from red to green immidiately, we decided to go with that approach to simplify our code and lives.

To differentiate the traffic lights we decided to use two breadboards with 3 LED's on each one. The three LED's correspond to the colors of a traffic light, so we have a red , yellow and a green one- six in total.

The resistors we will be using in this experiment are there to ensure our LED's survive.

Before the lab itself, we created the circuit in TinkerCad software and tried it out, and according to that schema we created the one in class. We got all the supplies and assembled the circuit.

Our final circuit version looked like this:

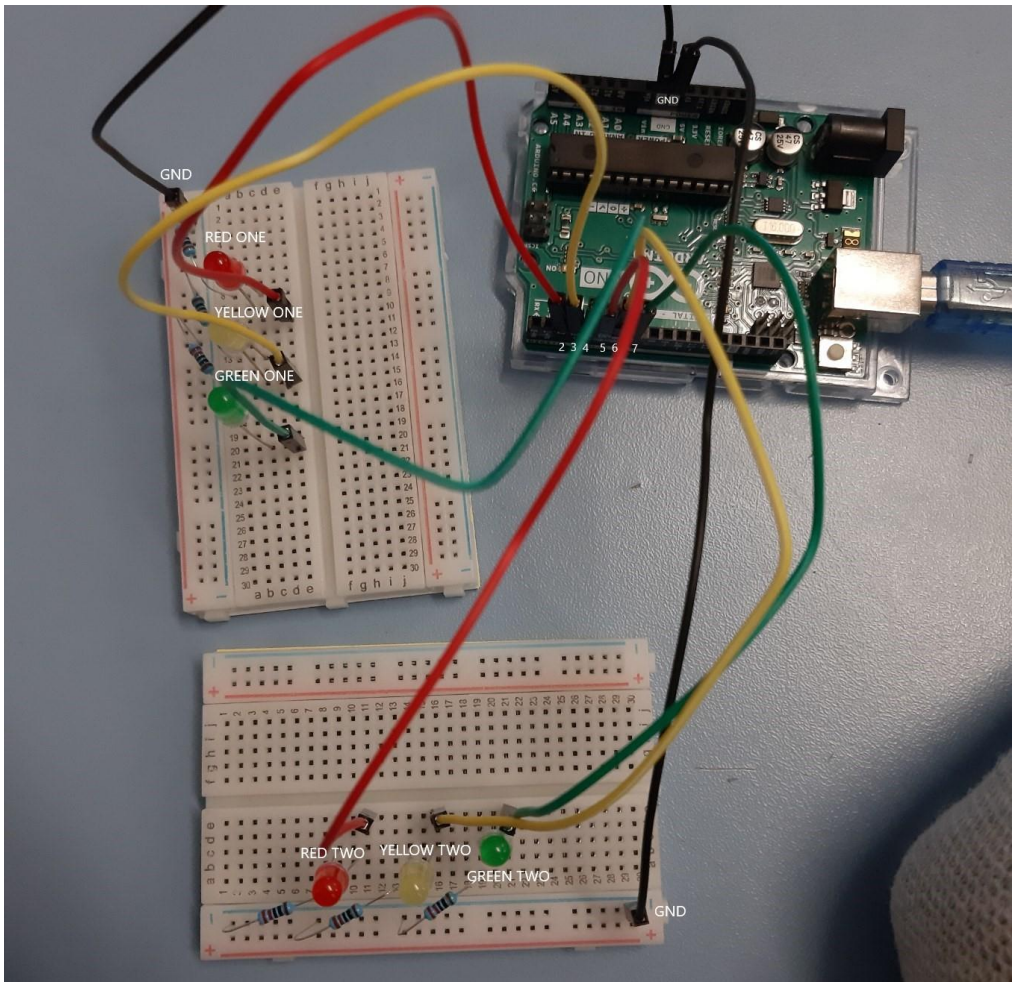

Figure 3: Circuit

The code we will be using to program the board:

```
// declaring our pins, we could also keep them unnamed (and use their numbers instead) but for the sake of simplicity we will name them according to their colors and numbers

int FirstRed = 2;

int FirstYellow = 3;

int FirstGreen = 4;

int SecondRed = 5;

int SecondYellow = 6;

int SecondGreen = 7;

//so the pins 2-4 are 'street one' and pins 5-7 are 'streer' two.


void setup() {

// sets pins as outputs, since all our LED's are outputs.

   pinMode(FirstGreen, OUTPUT);

   pinMode(SecondGreen, OUTPUT);

   pinMode(FirstYellow, OUTPUT);

   pinMode(SecondYellow, OUTPUT);

   pinMode(FirstRed, OUTPUT);

   pinMode(SecondRed, OUTPUT);

}


void loop() {

// the function that will loop itself until we cancel it


 // state one
```

```
 digitalWrite(FirstGreen, LOW);
 digitalWrite(SecondGreen, HIGH);
 digitalWrite(FirstYellow, LOW);
 digitalWrite(SecondYellow, LOW);
 digitalWrite(FirstRed, HIGH);
 digitalWrite(SecondRed, LOW);
 delay(5000); //in miliseconds

// state two
 digitalWrite(FirstGreen, LOW);
 digitalWrite(SecondGreen, LOW);
 digitalWrite(FirstYellow, LOW);
 digitalWrite(SecondYellow, HIGH);
 digitalWrite(FirstRed, HIGH);
 digitalWrite(SecondRed, LOW);
 delay(2000);
// state three
 digitalWrite(FirstGreen, LOW);
 digitalWrite(SecondGreen, LOW);
 digitalWrite(FirstYellow, LOW);
 digitalWrite(SecondYellow, LOW);
 digitalWrite(FirstRed, HIGH);
 digitalWrite(SecondRed, HIGH);
 delay(2000);
//state four
```

```
  digitalWrite(FirstGreen, HIGH);
  digitalWrite(SecondGreen, LOW);
  digitalWrite(FirstYellow, LOW);
  digitalWrite(SecondYellow, LOW);
  digitalWrite(FirstRed, LOW);
  digitalWrite(SecondRed, HIGH);
  delay(5000);
 // state five
  digitalWrite(FirstGreen, LOW);
  digitalWrite(SecondGreen, LOW);
  digitalWrite(FirstYellow, HIGH);
  digitalWrite(SecondYellow, LOW);
  digitalWrite(FirstRed, LOW);
  digitalWrite(SecondRed, HIGH);
  delay(2000);

 // state six
  digitalWrite(FirstGreen, LOW);
  digitalWrite(SecondGreen, LOW);
  digitalWrite(FirstYellow, LOW);
  digitalWrite(SecondYellow, LOW);
  digitalWrite(FirstRed, HIGH);
  digitalWrite(SecondRed, HIGH);
  delay(2000);
} //end of loop, end of code
```

5. Results and Conclusion

Like we said, we already tried out the simulation prior to the lab and tried out similar projects in Digital Design class, we already knew what to expect. We also took some inspiration from multiple Arduino projects found online, we will reference one such project. After plugging the Arduino and starting it, it worked with no errors. We only played with the delays a bit, we are still unsure if we got them right. Here are a few results of our simulation to show a few states.

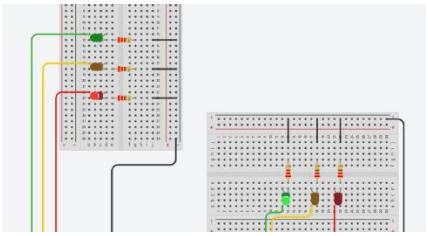

Figure 4: simulation results 1

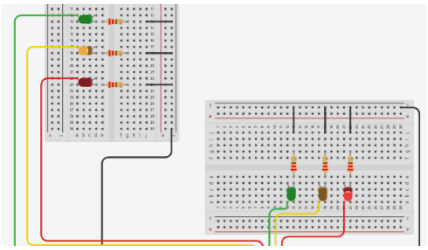

Figure 5: simulation results 2

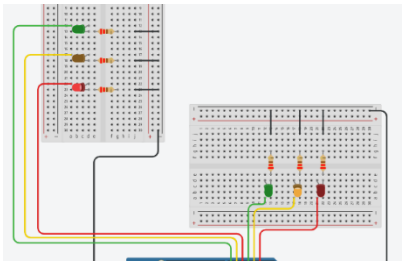

Figure 6: simulation results 3

To conclude, Arduino is a very small but powerful microcontroller that can be used in hundreds of simple and many more complex projects, and also in real life implementations. We have enjoyed using it during this project a lot. It would have also been better if we had the time to explore different components we have in the lab, since we couldn't do many projects because we were unsure of what we could get and actually use for this project. All in all, it was a fun experience.

References and software

Software:

Tinkercad, https://www.tinkercad.com/

Adruino Uno IDE

References:

"Arduino UNO - JavaTpoint." *Www.javatpoint.com*, 2021, www.javatpoint.com/arduino-uno.

"Arduino Traffic Light Simulator." *Arduino Project Hub*, Arduino Project Hub, 2017, create.arduino.cc/projecthub/techno_z/arduino-traffic-light-simulator-2ec9f7.