Polytechnic University of Puerto Rico
San Juan Campus

Electrical and Computer Engineering Faculty

Microprocessor Interfacing Laboratory

Basic I/O: Traffic Light System Design (North-South, East-West)

Kelvin Figueroa Figueroa - 108946
Rafael Gonzalez Cartagena - 90352

Professor: Dr. Roman Lopez

March 30, 2022

Contents

List of Tables

List of Figures

1 Objective

This laboratory aims to stop a program at times. To implement this application, it is simulated a traffic light where the directions are from north to south and east to west, refer to figure 1. When simulating the intersection, a function will be made to allow pedestrians to pass through the intersection, where the program will stop the sequence on one sight of the intersection.

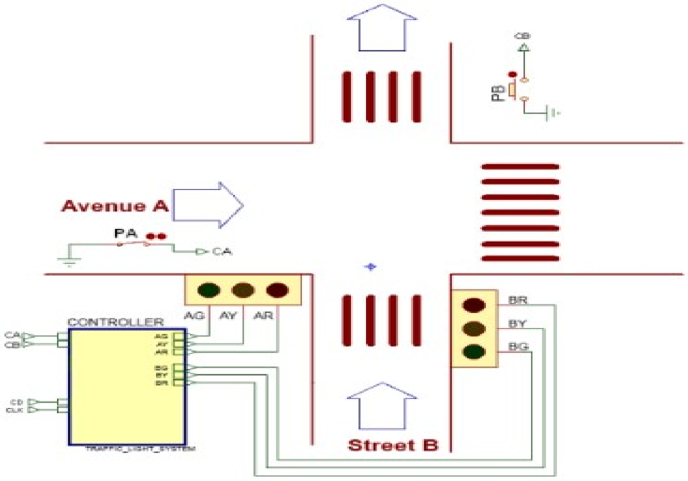


Figure 1: Intersection design

In order to make this application possible, make sure to have the following materials,

1. Arduino Mega ADK
2. 8 LED's (2 reds, 2 yellow, 2 greens and 2 blue)
3. 8 330 ohm's resistors
4. 2 push button switch

2 Procedures

This laboratory has two similar parts, where the first one implements one Traffic Light from North to South (refer to section 2.1 and the second part implements the first one and a Traffic Light from East to West (refer to section 2.2. To connect the buttons, we used the same laboratory board LED's Matrix, where the connections are in table 2

Table 1: Matrix Buttons Connection

Arduino Pins	Matrix Pins
5V	VCC
GND	GND
D2	S1
D3	S2

Before starting the experiments with the board, the following steps had to be followed to set up the Arduino IDE,

1. Assemble the circuit for two traffic lights into the laboratory board.

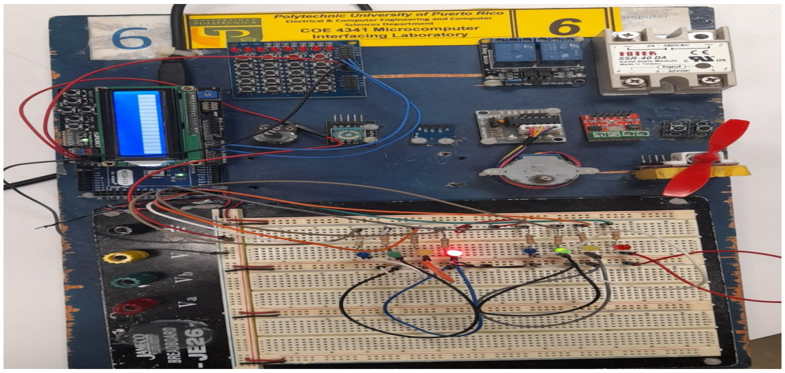
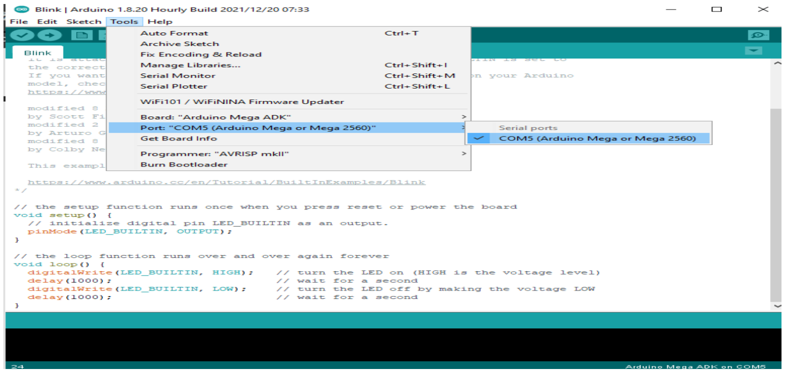


Figure 2: Assembled circuit

2. Then select the port where our board is connected, Tools > Port > COM.


Blink | Arduino 1.8.20 Hourly Build 2021/12/20 07:33
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Mega ADK"
Port: "COM5 (Arduino Mega or Mega 2560)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM5 (Arduino Mega or Mega 2560)
Blink
24
Arduino Mega ADK on COM5

```
  the correct
  If you want                                          on your Arduino
  model, chec
  https://www

  modified 8
  by Scott Fi
  modified 2
  by Arturo G
  modified 8
  by Colby Ne

  This exampl

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Figure 3: Selecting Port inside IDE

2.1 One Traffic Light(N-S)

For this first stage of the laboratory, we are looking to implement a traffic light from north to south with the option of implementing a pedestrian crossing.

1. Open a new Sketch on Arduino IDE and write the code given by the professor. The code can be found in presentation 3.6 from MEGA.

2. Load the code to the Arduino Mega and simulate a traffic light from North to South.

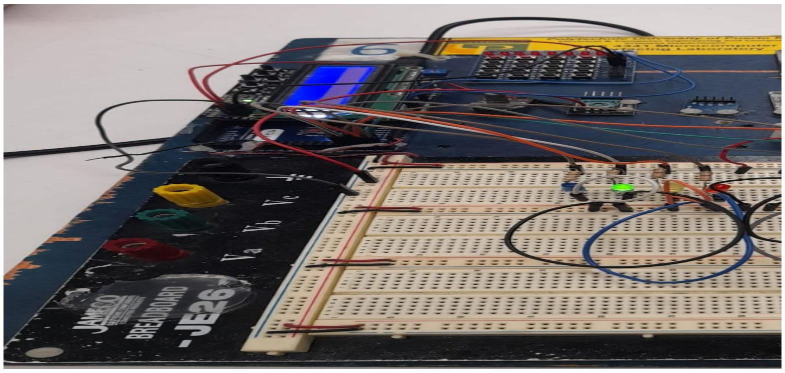


Figure 4: One Traffic Light

2.2 Two Traffic Lights (N-S and E-W

For this second stage of the laboratory, we implement a complete intersection with pedestrian crossing when one of the button's are pushed.

1. Having the code for a traffic light, it had to be modified so that it could work with two traffic lights. The Arduino Pins used for this stage are from pin 22 to 29. Refer to code 3.1

2. After writing the program, we load it into the Arduino Mega and waited for 3 cycles to make sure that the program was running good. Once waited for the three cycles, we tested the pedestrian functions by pressing one of the two buttons in the laboratory board. Refer to figure 5

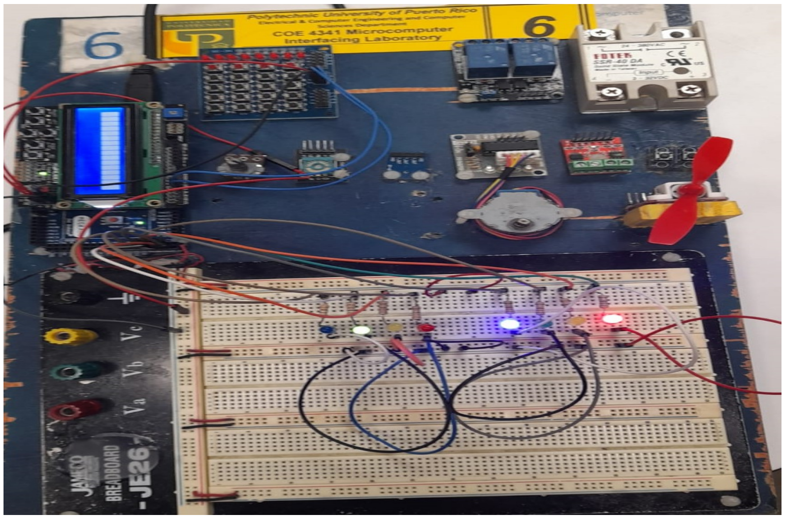


Figure 5: Pedestrian Blue Light Test

It should be noted that the descriptions of all the steps provided in this section were obtained with the laboratory's documentation [1].

3 Results

This section describes the results obtained in the experiment related with this report.

3.1 The Embedded C Programs

The *Embedded C* programs that were created, modified and documented for the two (2) stages of the Basic I/O: Traffic Light System Design (North-South, East-West) are shown below:

Embedded C Code for Stage 1: Single Traffic Light System (Test Stage)

```c
/*
 * This is the original code of the professor for one traffic light case!
 * ---------------------------------------------
 * Created by Dr. Roman Lopez.
 * ---------------------------------------------
 */

#define cross 28    // Pin for pedestrain light
#define red 22      // Pin for red light
#define yellow 24   // Pin for yellow light
#define green 26    // Pin for green light
#define button 2    // Pin for pedestrian button
#define delayTime 500 // The delay times to detect the button
// The time-constants for each of the lights
#define greenTime 5
#define yellowTime 3
#define redTime 5

volatile int buttonState = 0; // I think this was not used throughout the code!
int i = 0; // Varibale to act as the i of the FORs
int flag;  // Flag for interrupt
int lightState; // The state of the lights of the traffic light

void setup()
{
  pinMode(cross, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(button, INPUT);

  attachInterrupt(digitalPinToInterrupt(button), setFlag, LOW); //Attached the Interrupt subroutine
  Serial.begin(9600);
  flag = 0;
}

void loop()
{
LOOP:
  //Set the state flag, 1=green, 2=yellow, 3=red
  //Set green ON, ref and Yellow OFF
  //Green state
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
  digitalWrite(green, HIGH);
  digitalWrite(cross, LOW);
  for (i = 0; i <= greenTime; i++)
  {
    delay(delayTime);
    if (flag == 1)
    {
      lightState = 1;
      Pedestrian();
      goto LOOP;
    }
  }

  digitalWrite(green, LOW);
  digitalWrite(yellow, HIGH);
  for (i = 0; i <= redTime; i++)
  {
    delay(delayTime);
    if (flag == 1)
    {
      lightState = 2;
      Pedestrian();
      goto LOOP;
    }
  }

  digitalWrite(yellow, LOW);
  digitalWrite(red, HIGH);
  for (i = 0; i <= yellowTime; i++)
  {
    delay(delayTime);
    if (flag == 1)
    {
      lightState = 3;
      Pedestrian();
      goto LOOP;
    }
  }
}
/*
 * Function to set the interrupt flag of the traffic light to 1.
 */
void setFlag(void)
{
  flag = 1;
}
void Pedestrian(void)
{
  switch (lightState)
  {
    case 1: // Case for when traffic light is GREEN.
      {
        digitalWrite(green, LOW);
        digitalWrite(yellow, HIGH);
        delay(2000);
        digitalWrite(yellow, LOW);
        digitalWrite(red, HIGH);
        digitalWrite(cross, HIGH);
        delay(5000);
        break;
      }

    case 2: // Case for when traffic light is YELLOW.
      {
        delay(1500);
        digitalWrite(yellow, LOW);
        digitalWrite(red, HIGH);
        digitalWrite(cross, HIGH);
        delay(5000);
        break;
      }

    case 3: // Case for when traffic light is RED.
      {
        digitalWrite(red, HIGH);
        digitalWrite(cross, HIGH);
        delay(3000);
        break;
      }

    default : {};
  }

  for (i = 0; i <= 5; i++) // For to turn on and off intermitently the pedestrian light for 1 second each time.
  {
    digitalWrite(cross, LOW);
    delay(1000);
    digitalWrite(cross, HIGH);
    delay(1000);
  }
  flag = 0; // Reset the interrupt flag of the predestrian.
}
```

Embedded C Code for Stage 2: Two Traffic Lights System

```
/*
 * The following LEDs and pins are related to one-another:
 * RED1     ->      22
 * RED2     ->      23
 * YELLOW1  ->      24
 * YELLOW2 ->     25
 * GREEN1  ->     26
 * GREEN2  ->     27
 * CROSS1  ->     28
 * CROSS2  ->     29
 */

const int button1 = 2;                            // Pin of
    button for traffic light 1
const int button2 = 3;                            // Pin of
    button for reaffuc light 2
const int tSize = 4;                              // The size
    of each of the arrays for the number of lights of each
    traffic light
const int trafficA[] = {22, 24, 26, 28};          // Traffic
    Light A lights
const int trafficB[]= {23, 25, 27, 29};           // Traffic
    Light B Lights
const int timmings[] = {5, 3, 5};                 // The
    constant of timmings for each of the lights: RED, YELLOW
    and GREEN, respectively.

int state = 1;                                    // Variable
    to know which light is the program currently on
bool atFirst = true;                              // Variable
    to know which of the two (2) traffic lights is currently
    being used; i.e.: changing colors.
int flag1 = 0;                                    // Interrupt
    Flag for traffic 1
int flag2 = 0;                                    // Interrupt
    Flag for traffic 2

void setup() {
  /*
   * FOR to set the pin mode of all the pins of the two (2)
       traffic lights LEDs. With that being said, since they
       are LEDs, they are set to be OUTPUTS.
   */
  for(int i = 0; i<tSize;i++){
    pinMode(trafficA[i],OUTPUT);
    pinMode(trafficB[i],OUTPUT);
  }
  pinMode(button1, INPUT);                            // Setting pin mode of the pin of button 1 to INPUT since its, well, a button.
  pinMode(button2, INPUT);                            // Setting pin mode of the pin of button 2 to INPUT.
  attachInterrupt(digitalPinToInterrupt(button1),setFlag1,LOW); // Attached the Interrup subroutine. It is LOW per professor's instructions; hence, oposite button will activate this interrupt.
  attachInterrupt(digitalPinToInterrupt(button2),setFlag2,LOW); // Same as above, but for button 2.
  Serial.begin(9600);
  construct();                                        // Call of function to set all LEDS to OFF state. (Initial State)
}

void loop() {
  changeStateTA();                                    // Call of function to tackle with the changing of traffic 1 lights. This does not include when the pedestrian light is on.
  changeStateTB();                                    // Call of function to tackle with the changing of traffic 2 lights. This does not include when the pedestrian light is on.
  waitIntv();                                         // Call of function to wait some interval of time depending on the current light-states of the traffic lights.
  /*
   * If condition to check if state is not equal to zero; that is, an interrupt hasn't occured. Don't want to jump states!
   */
  if(state != 0){
    state = state + 1;
  }
  /*
   * If state is greater than three (3); that is, all three states have already been completer for one of the traffic lights (or a Interrupt has occured), then reset
```

```
    the state variable to 1 and invert atFirst.
  */
 if(state > 3){
  state = 1;
  atFirst = !atFirst;
 }
}

/*
 * Function to set the value of flag1 to 1
 */
void setFlag1(void){
 flag1 = 1;
}
/*
 * Function to set the value of flag2 to 1
 */
void setFlag2(void){
 flag2 = 1;
}
/*
 * Function to do the initial case for all the traffic lights;
    that is, to deactivate all of the LEDS if activated
    previously.
 */
void construct(void){
 for(int i = 0; i < tSize; i++){
  digitalWrite(trafficA[i],LOW);
 }
 for(int i = 0; i < tSize; i++){
  digitalWrite(trafficB[i],LOW);
 }
}
/*
 * Function to change the current light of traffic light 1;
    hence:
 *  If atFirst = true, then the light state (the current
    activated light) will change for the traffic light 1.
 *  If the opposite is true, the traffic light will have the
    RED light on.
 */
void changeStateTA(void){
  if(atFirst==true){
    if(state != 1){
      digitalWrite(trafficA[tSize-state],LOW); // Deactivate the previous activated light. (If green is on in state 1, then in state 2 it must be deactivated.
    }
    else{
      digitalWrite(trafficA[0],LOW); // Deactivate RED light since it will be on by state 1.
    }
    digitalWrite(trafficA[tSize-state-1],HIGH); // Activate current state light.
  }
  else{
    digitalWrite(trafficA[0], HIGH); // Activate RED light if the if does not hold.
  }
}
/*
 * This function does the same thing but for traffic light 2. More over, the only difference is that atFirst must be false since only one of the two traffic lights
 * must be able to change its light state at a time.
 */
void changeStateTB(void){
  if(atFirst==false){
    if(state != 1){
      digitalWrite(trafficB[tSize-state],LOW);
    }
    else{
      digitalWrite(trafficB[0],LOW);
    }
    digitalWrite(trafficB[tSize-state-1],HIGH);
  }
  else{
    digitalWrite(trafficB[0], HIGH);
  }
}
```

```

/*
 * Function to enable each of the traffic lights' LEDS be on for a certain interval of time. For instance,
 *  RED and GREEN should be on for 5 seconds.
 *  YELLOW should be on for 3 seconds.
 *  NOTE: This is their normal case; not in their pedestrian alternative!
 */
void waitIntv(void){
  for(int i = 0; i < timmings[state-1];i++){
    delay(1000);                      // 1 second
    if(flag1 == 1 || flag2 == 1){  // If to check if one of the buttons was pressed; that is, if any of the interrupts occurred.
      pedestrianBehavior();           // Pedestrian Function to do when interrupt occurs.
      break;                          // This was the bane of my existance, since I forgot to put it.
    }
  }
}

/*
 * This function does all of the steps necesary for a pedestrain to be able to walk the crosswalk for one of the traffic lights.
 */
void pedestrianBehavior(void){
  if (atFirst == true){           // If to check if the first traffic light was changing its lights colors when one of the buttons was pressed.
    switch(state){
      case 1:                     // Case for when Traffic Light 1 is at GREEN and Traffic Light 2 is at RED
        if(flag2 == 1){
          digitalWrite(trafficB[tSize-1], HIGH);
          delay(timmings[state]*1000);
          break;
        }
```

```
     for(int i = tSize-state-1; i>0; i--){
       digitalWrite(trafficA[i],LOW);
       digitalWrite(trafficA[i-1],HIGH);
       if(i == tSize-state-1){
        delay(2000);
       }
     }
     digitalWrite(trafficB[0],LOW);
     digitalWrite(trafficB[tSize-state-1],HIGH);
     digitalWrite(trafficA[tSize-1],HIGH);
     delay(timings[state]*1000);
     break;
    case 2:         // Case for when Traffic Light 1 is at YELLOW and Traffic Light 2 is at RED
     delay(timings[state]*1000/2);
     digitalWrite(trafficA[tSize-state-1],LOW);
     digitalWrite(trafficA[0],HIGH);
     if(flag2 == 1){
      digitalWrite(trafficB[0],LOW);
      digitalWrite(trafficB[tSize-state-1],HIGH);
      digitalWrite(trafficB[tSize-1],HIGH);
      delay(2000);
      digitalWrite(trafficB[tSize-state], LOW);
      digitalWrite(trafficB[tSize-state-1],HIGH);
      delay(2000);
      digitalWrite(trafficB[tSize-state-1],LOW);
      digitalWrite(trafficB[0],HIGH);
      digitalWrite(trafficA[0],LOW);
      digitalWrite(trafficA[tSize-state],HIGH);
      break;
     }
     digitalWrite(trafficA[tSize-1],HIGH);
     digitalWrite(trafficB[0],LOW);
     digitalWrite(trafficB[tSize-state],HIGH);
     delay(5000);
     break;
    case 3:      // Case for when Traffic Light 1 is at RED and Traffic Light 2 is at RED.
     if(flag2 == 1){
      digitalWrite(trafficB[tSize-1],HIGH);
```

```
        digitalWrite(trafficB[0],LOW);
        digitalWrite(trafficB[state-1],HIGH);
        delay(2000);
        digitalWrite(trafficB[state-1],LOW);
        digitalWrite(trafficB[tSize-state],HIGH);
        delay(2000);
        digitalWrite(trafficB[tSize-state],LOW);
        digitalWrite(trafficB[0],HIGH);
        digitalWrite(trafficA[0],LOW);
        digitalWrite(trafficA[state-1],HIGH);
        delay(2000);
        break;
      }
      digitalWrite(trafficA[tSize-1],HIGH);
      delay(2000);
      digitalWrite(trafficB[0],LOW);
      digitalWrite(trafficB[state-1],HIGH);
      digitalWrite(trafficA[tSize-1],HIGH);
      delay(2000);
      break;
    default:
    {};
  }
}
else{ // ELSE activate is the second traffic light is changing colors.
  switch(state){
    case 1:      // Case for when Traffic Light 1 is at RED and Traffic Light 2 is at GREEN.
      if(flag2 == 1){
        delay(2000);
        digitalWrite(trafficB[tSize-state-1],LOW);
        digitalWrite(trafficB[state],HIGH);
        delay(2000);
        digitalWrite(trafficB[state],LOW);
        digitalWrite(trafficB[0],HIGH);
        digitalWrite(trafficA[0],LOW);
        digitalWrite(trafficA[tSize-state-1],HIGH);
        digitalWrite(trafficB[tSize-1],HIGH);
        delay(2000);
      break;
     }
     digitalWrite(trafficA[tSize-1],HIGH);
     delay(2000);
     break;
    case 2: // Case for when Traffic Light 1 is at RED and Traffic Light 2 is at YELLOW.
     if(flag2 == 1){
      digitalWrite(trafficB[tSize-1],HIGH);
      delay(2000);
      digitalWrite(trafficB[state-1],LOW);
      digitalWrite(trafficB[0],HIGH);
      digitalWrite(trafficA[0],LOW);
      digitalWrite(trafficA[state],HIGH);
      delay(2000);
      break;
     }
     digitalWrite(trafficA[tSize-1],HIGH);
     delay(1000);
     digitalWrite(trafficB[state-1],LOW);
     digitalWrite(trafficB[0],HIGH);
     digitalWrite(trafficA[0],LOW);
     digitalWrite(trafficA[state],HIGH);
     delay(2000);
     digitalWrite(trafficA[state],LOW);
     digitalWrite(trafficA[state-1],HIGH);
     delay(20000);
     digitalWrite(trafficA[state-1],LOW);
     digitalWrite(trafficA[0],HIGH);
     digitalWrite(trafficB[0],LOW);
     digitalWrite(trafficB[state],HIGH);
     delay(1000);
     break;
    case 3: // Case for when Traffic Light 1 is at RED and Traffic Light 2 is at RED.
     if(flag2 ==1){
      digitalWrite(trafficB[state],HIGH);
      delay(1000);
      digitalWrite(trafficA[0],LOW);
      digitalWrite(trafficA[state-1],HIGH);
```

```
          delay(1000);
          break;
        }
        digitalWrite(trafficA[state],HIGH);
        delay(1000);
        digitalWrite(trafficA[0],LOW);
        digitalWrite(trafficA[state-1],HIGH);
        delay(2000);
        digitalWrite(trafficA[state-1],LOW);
        digitalWrite(trafficA[tSize-state],HIGH);
        delay(2000);
        digitalWrite(trafficA[tSize-state],LOW);
        digitalWrite(trafficA[0],HIGH);
        digitalWrite(trafficB[0],LOW);
        digitalWrite(trafficB[state-1],HIGH);
        break;
      default:
      {};
    }
  }
  /*
   * If and else statments to activate intermitently the pedestrian lights of traffic light 1 and 2, respectively.
   */
  if(flag1 ==1){
    for(int i = 0; i<5;i++){
      digitalWrite(trafficA[tSize-1],LOW);
      delay(500);
      digitalWrite(trafficA[tSize-1],HIGH);
      delay(500);
    }
  }
  else{
    for(int i = 0; i<5;i++){
      digitalWrite(trafficB[tSize-1],LOW);
      delay(500);
      digitalWrite(trafficB[tSize-1],HIGH);
      delay(500);
    }
  }
  /*
   * IF to reset traffic light 1 whenever its button was pressed.
   */
  if(flag1 == 1){
    digitalWrite(trafficA[tSize-1],LOW);
    digitalWrite(trafficB[2],LOW);
    digitalWrite(trafficB[1],HIGH);
    delay(1000);
    digitalWrite(trafficB[1],LOW);
    digitalWrite(trafficB[0],HIGH);
  }
  else{
    digitalWrite(trafficB[tSize-1],LOW); // This else statment may be deprecated, but there seems to be no change in the functionality of the program.
  }
  flag1 = 0;    // Reset flag 1
  flag2 = 0;    // Reset flag 2
  state = 4;    // Set state to be greater than 3 so to reset it later on in the loop function
  atFirst = false; // Set atFirst to false to invert it later on.
}
```

With all of the developed code already shown, it should be noted that the code of the first state, i.e.: **Embedded C Code for Stage 1: Single Traffic Light System (Test Stage)**, was acquired from the professor; hence, such lines of code are not our own [2].

3.2 The Videos of the Results

In conjunction with the previous subsection, this one indicates the appointed names of the video files which illustrate the results of each of the already described stages, as shown in the Procedures section of this report. With that being said, the names of such videos are found in the following table, i.e., Video Names of Laboratory Results:

Table 2: Video Names of Laboratory Results

Video Names of Laboratory Results
stage1_Video_1traffic_light
stage2_Video_2traffic_light

4 Analysis of Results

4.1 Stage 1: Single Traffic Light System (Test Stage)

From the steps and results that were respectively undertaken and gathered, this stage was success; more over, as once stated, because the program that was to be used was of the professor.

4.2 Stage 2: Two Traffic Lights System

Now, this stage; that is, **Stage 2: Two Traffic Lights System**, was, to some extent, most complicated, albeit not impossible. This remark is stated given the following reasoning:

- The Embedded C program that was created for this phase was rewritten in different ways, since it was never understood if it was possible to utilize the program created by the professor of the previous stage; that is: Stage 1: Single Traffic Light System (Test Stage). With that being said, it should be noted that because of the constant re-writings and whatnot, a portion of the code malfunctioned for some time, after any of the interruptions occurred; however, this was fixed.
- The instructions presented on the laboratory documentation for this stage were not properly understood with respect to the timings of each of the LED's.

No matter these shortcomings, it is believed that, as stated in the previous subsection of this experiment, this one was fulfilled successfully.

5 Conclusion

Overall, the creation and implementation of a basic Traffic Light System with an Arduino MEGA ADK, as stated previously, was a success. Furthermore, it was found to be possible to implement:

- Various ways to implement an interrupt system for any Arduino Embedded C program with the aid of functions, such as: **attactchInterrupt()** and **digitalPinToInterrupt()**.
- Use functions, like: **delay()** or **micro()**, to create in one form or another waiting systems.

References

The following *References* related with this document are in the IEEE Format, as shown below:

[1] Roman Lopez, Phd., "Lab 2 Traffic Light System". Mega. 2022.
[2] Roman Lopez, Phd., "3.6 Pedestrian Traffic Light". Mega. 2022.