LAB 4 – MAKERSPACE LABS – TRAFFIC CONTROL

Use Lab 0 as a reference to this Lab

Click hyperlinks (blue underlined text) to learn more about a particular topic or concept.

Objectives

- Connect 7 LEDs & a pushbutton to the breadboard
- Create a traffic light system

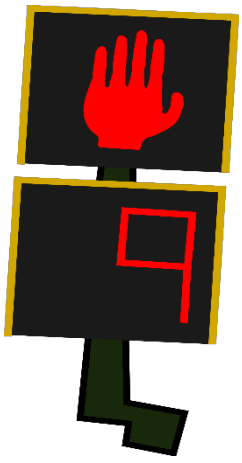
Parts/Equipment Required

- Arduino Uno
- Breadboard
- Laptop
- USB Cable
- Several LEDs
- Several 470Ω Resistors
- Pushbutton
- Various Wires

Traffic Light Graphic

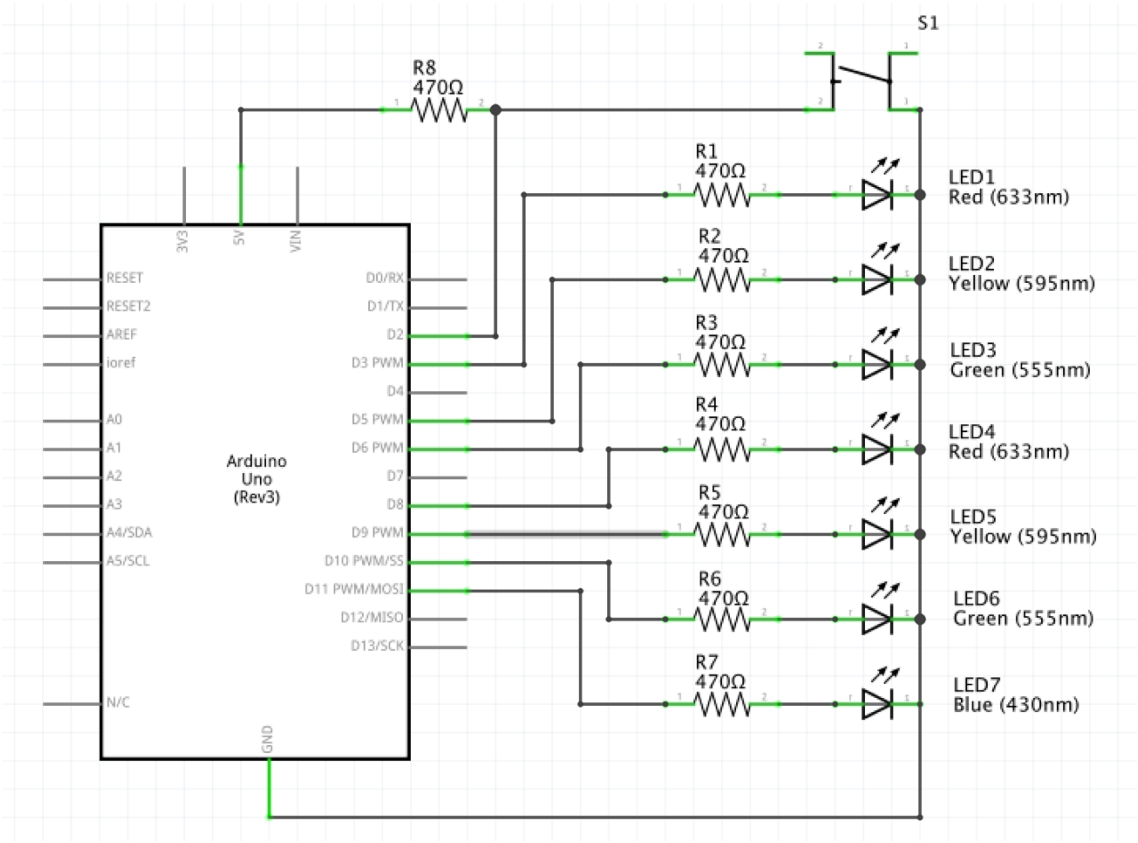
We will be mimicking a Traffic Light for this Lab. Using 2 sets of Red, Yellow and Green LED's to represent the symbol colours of a set of Traffic Lights.

Using a Blue LED to represent the Walking Light, being activated by a push-button. The Walk Light will blink for the desired time and return back to the normal functionality after.

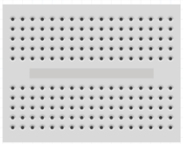


Hardware

In this lab, we are going to be wiring a circuit that looks like this:

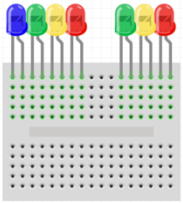


1. For this lab, it would be a good idea to start with a fresh, blank breadboard.

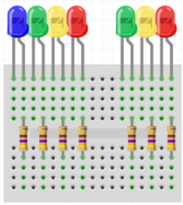


2. Start by adding two red, two yellow, two green, and one blue LED to the breadboard in this pattern.

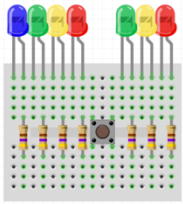
   Remember that the blue LED may look clear and not blue.



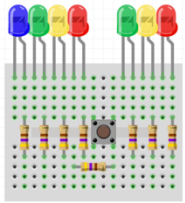
3. Now add 470Ω resistors for each LED. Make sure to add them to the anode of the LED.

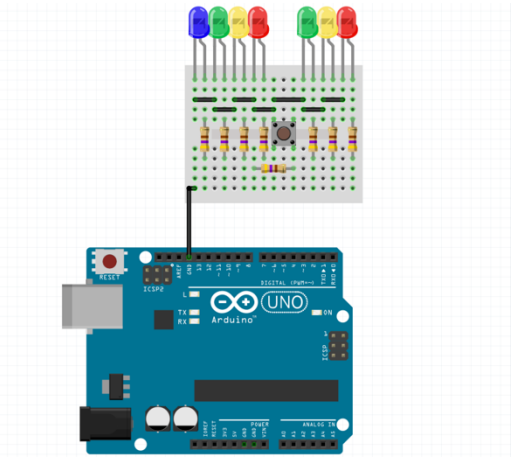


4. Now connect a switch in the space provided in the middle of the board.



5. A 470Ω resistor should be added to pull-up the button.

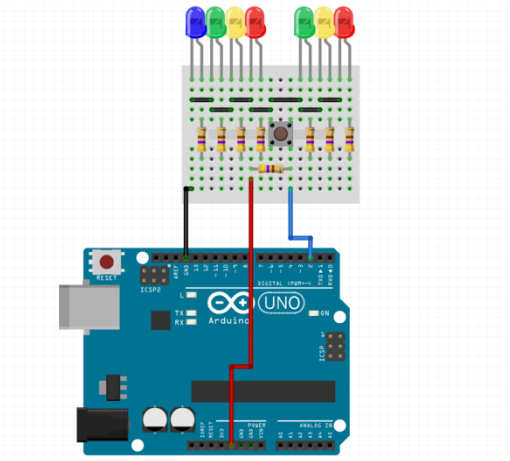


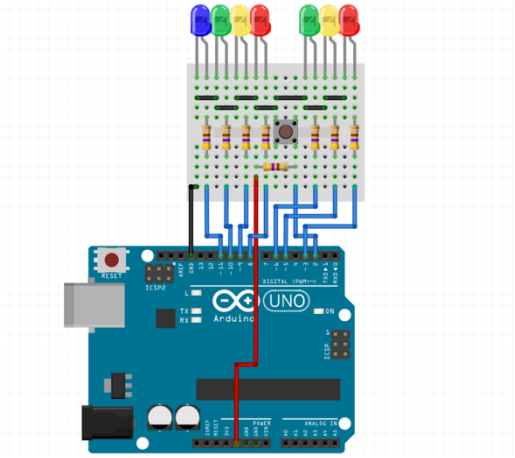


6. All of the cathodes on the LEDs and one side of the switch can be connected to ground on the Arduino.

7. The 470 Ω resistor on the Arduino can be connected to +5V.

   The side of the switch with the resistor can be connected to pin 2 on the Arduino.





8. Now all the LEDs can be connected to the Arduino. Depending on what pins you connect the LEDs to you may have to modify the code.

Software

Type the code on the next page into the Arduino IDE. Because there is a lot of code, it might be a good idea to copy and paste it from https://arudino.zaugg.ca /code/Lab4_Code.html instead. Once you've copied the code, flashed the Arduino, and verified that things are working correctly, feel free to mess around with different parts of the code to see what they do. There are a lot of interesting and advanced components to this code.

```
/** Advanced Traffic Light
 *  @author Torin Zaugg
 *  @date 02/12/2018
 *  @comments Working with Arduino UNO
 */


int lamps[] = {3, 5, 6, 8, 9, 10, 11}; //This array contains the pins of all the LEDs

//Define the pins for all of the LEDs.
//NS = North/South. EW = East/West
#define GreenNS 3
#define YellowNS 5
#define RedNS 6
#define GreenEW 8
#define YellowEW 9
#define RedEW 10
#define Walk 11

//Define the pin for the switch
#define Button 2

//Variables for time information
int cycleTime = 5000;             //This is how long the light cycle is. Setting it larger makes the lights longer.
int yellowTime = (cycleTime/3);   //This is how long the yellow light is on for. Set 3 to 4 to make yellow lights longer.

boolean doWalkCycle = false;      //This variable is set to true when when the switch is pressed.

void setup() {

  //This attaches an interrupt to the pin the switch is on. When the pin is connected to ground (by pressing the switch)
  //the function toggleButton will run.
  attachInterrupt(digitalPinToInterrupt(Button), toggleButton, FALLING);

  //Set the direction for all the LEDs by cycling though the array with a for loop.
  for (int i = 0; i < 7; i = i + 1){
    pinMode(lamps[i], OUTPUT);
  }

  //Set the direction of the switch.
  pinMode(Button, INPUT);


  //Turn all the LEDs off by cycling though the array with a for loop.
  for (int i = 0; i < 7; i = i + 1){
    digitalWrite(lamps[i], LOW);
  }

}

//This function is ran when the button is pressed.
//Makes the variable doWalkCycle true.
void toggleButton(){
  doWalkCycle = true;
}

void loop() {

  //In this condition, the walk light should be turned on with the NS green LED.
  if(doWalkCycle){
    digitalWrite(RedNS, LOW);
    digitalWrite(YellowEW, LOW);
    digitalWrite(GreenNS, HIGH);
    digitalWrite(RedEW, HIGH);
    digitalWrite(Walk, HIGH);

    delay(cycleTime/2);

    //This flashes the walk light when it is about to change to yellow
    for(int i = 0; i < 4; i = i + 1){
      digitalWrite(Walk, LOW);
      delay(250);
      digitalWrite(Walk, HIGH);
      delay(250);
    }

    //Change to yellow and delay for the yellowTime
    digitalWrite(GreenNS, LOW);
    digitalWrite(Walk, LOW);
    digitalWrite(YellowNS, HIGH);
    delay(yellowTime);

    doWalkCycle = false;  //Set doWalkCycle back to false so you have to press the switch
                          //to run the walk cycle again.
```

```
84
85    //In this condition the walk light is not turned on. Besides that it is the same as above.
86    }else{
87      digitalWrite(RedNS, LOW);
88      digitalWrite(YellowEW, LOW);
89      digitalWrite(GreenNS, HIGH);
90      digitalWrite(RedEW, HIGH);
91
92      delay(cycleTime);
93
94      digitalWrite(GreenNS, LOW);
95      digitalWrite(YellowNS, HIGH);
96
97      delay(yellowTime);
98    }
99
100   digitalWrite(YellowNS, LOW);
101   digitalWrite(RedEW, LOW);
102   digitalWrite(RedNS, HIGH);
103   digitalWrite(GreenEW, HIGH);
104
105   delay(cycleTime);
106
107   digitalWrite(GreenEW, LOW);
108   digitalWrite(YellowEW, HIGH);
109
110   delay(yellowTime);
111 }
```