Lab Experiment No. 01:
Traffic Light Controller Implementation on Arduino UNO

By: David Dalmeida
Instructor: Dr. Jafar Saniie
ECE 442
Lab Date: 05-22-2019
Due Date: 05-29-2019

Introduction:

Purpose: The task of this experiment is to be implement the traffic signal lights using Arduino Uno R3 board. It will be implemented on a breadboard using basic circuitry components like LEDs, jumper wires, push buttons etc. The goal of the project is to get familiar with the basics of the Arduino microcontroller board which will help us for our project. The programming in Arduino is very simple that can be easily done by downloading the Arduino software on the Arduino official website. It involves getting knowledge of that software, C programming the commands (digitalRead, digitalWrite, inputs and outputs).

Scope: The limitations of this project are that it only teaches us how to control lights and we are also limited to use C programming. Also we only have a week to complete and submit the report. But it was a great experience and I would not change anything but the time constraint.

a. Theory:

The experiment is divided into two part

1) Simple two-way intersection

Two directions namely north/south and east/west are considered. This first logic to be implemented is to control the traffic signal at a very simple intersection. Each traffic signal has three lamps green for proceed, yellow for changing and red for stop. A timer alone control these lights and no need for sensors. We also considered that the north/south is busier than the east/west road so the north/south road green light will stay on for 10 sec and the e/w green light will stay on for 5 seconds. When one direction is given the green light the other should turn red. A transition time of 1 unit is given by the yellow light to change from green to red.

2) Two-way intersection with traffic sensors

For the second circuit we consider the previous traffic light pattern. If the traffic on the north/south is heavy and there is no traffic coming from the east/west road it is going to create big traffic jam on the n/s road. If there are no cars waiting for the red light on the east/west street, there is no reason to change the north/south signal from green to red. Automobile sensors are inserted on the side of the roads and when it gets activated, it is assumed to have traffic on that road. Then, the corresponding traffic light will turn green and allow the flow to move on. This will allow the north/south street to have an unrestricted traffic flow if no cars are waiting at either of the cross-street traffic signals.

b. Preliminary work

1. Arduino circuit to be simulate on TinkerCAD for traffic lights.

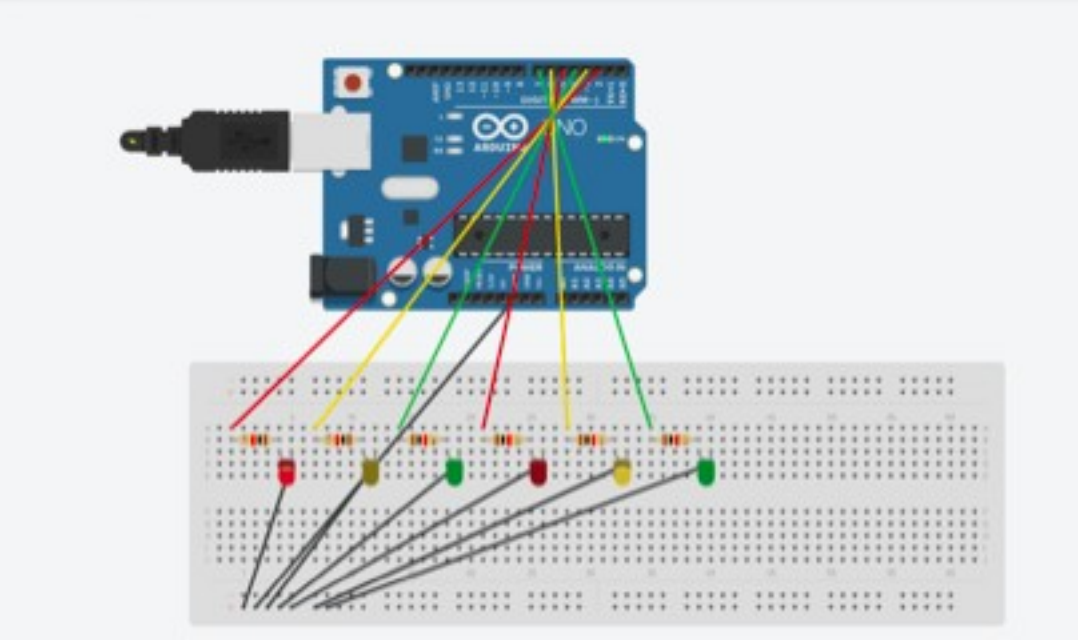


Fig.1 Part 1:Timer only
Refor to appendix for the code

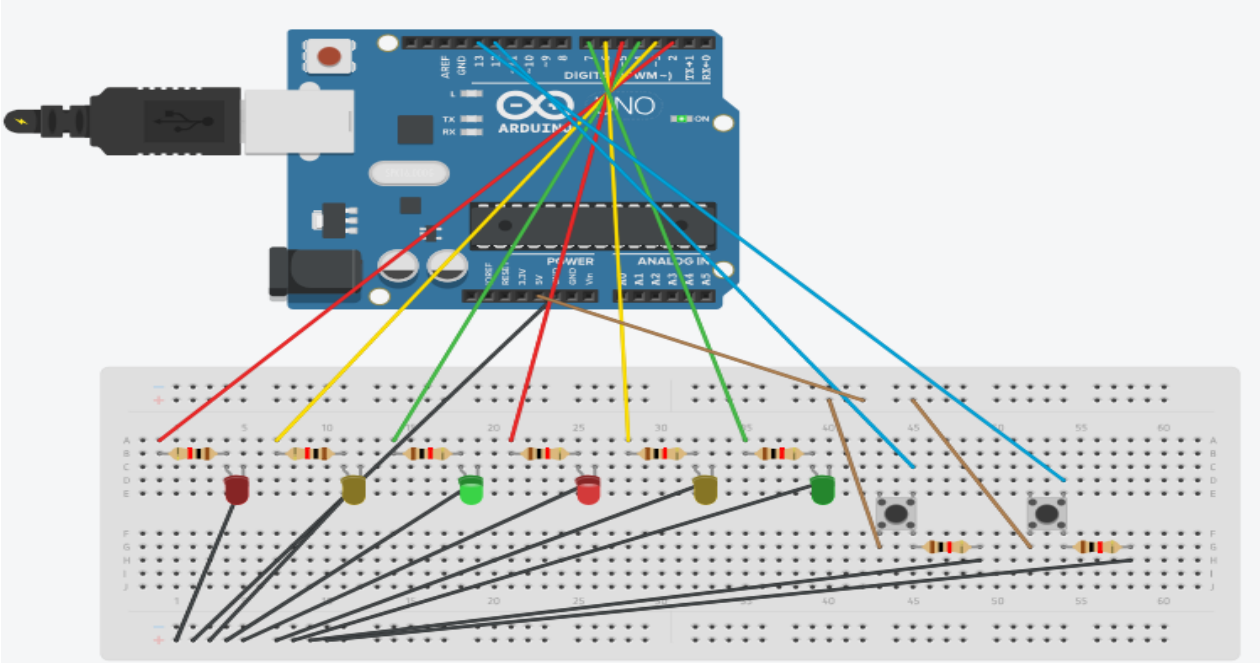
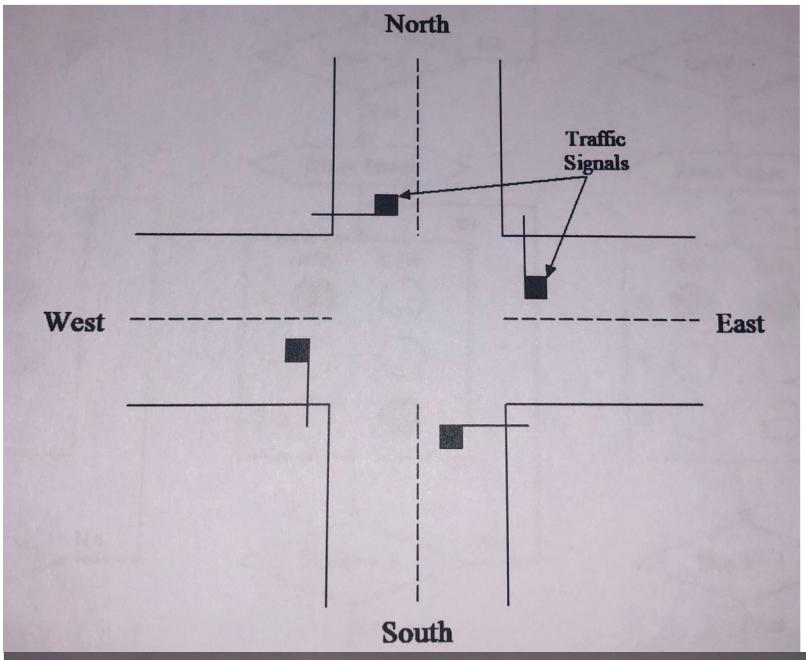


Fig 2 Part 2: Push buttons
Refer to appendix for code

Experimental procedure

a. Schematic

1. Simple two way intersection



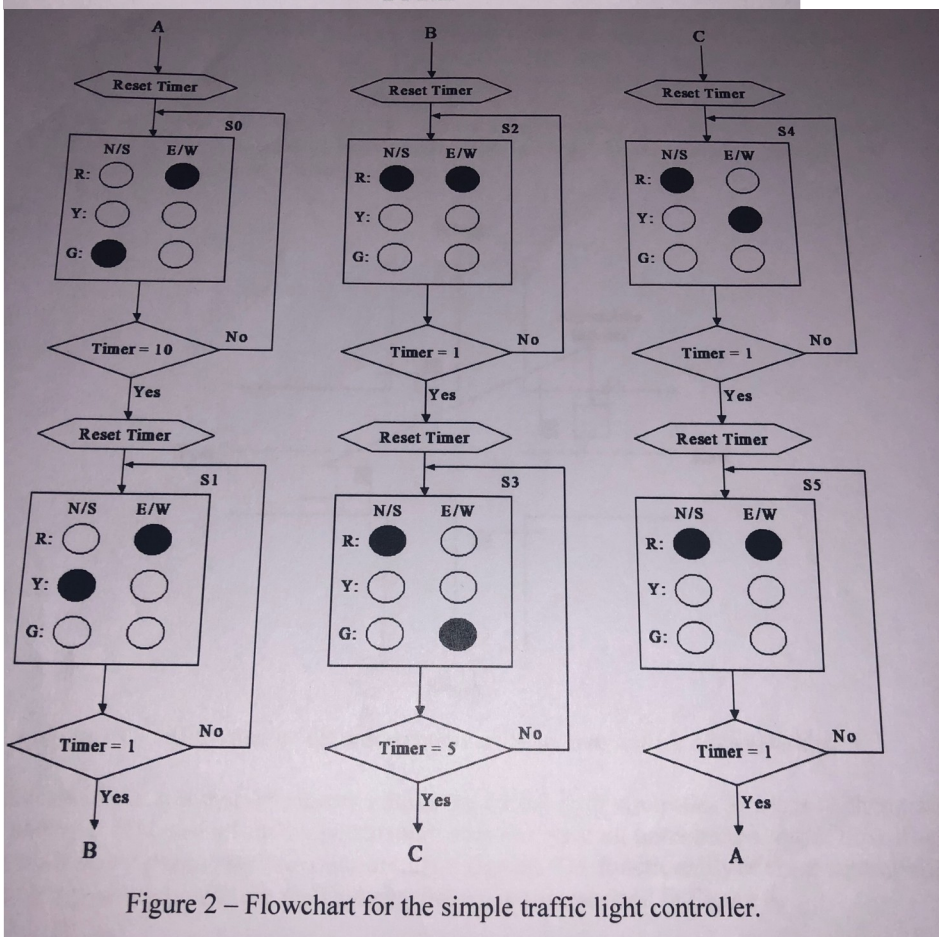


Figure 2 – Flowchart for the simple traffic light controller.

2. Two way Intersection with traffic Sensors

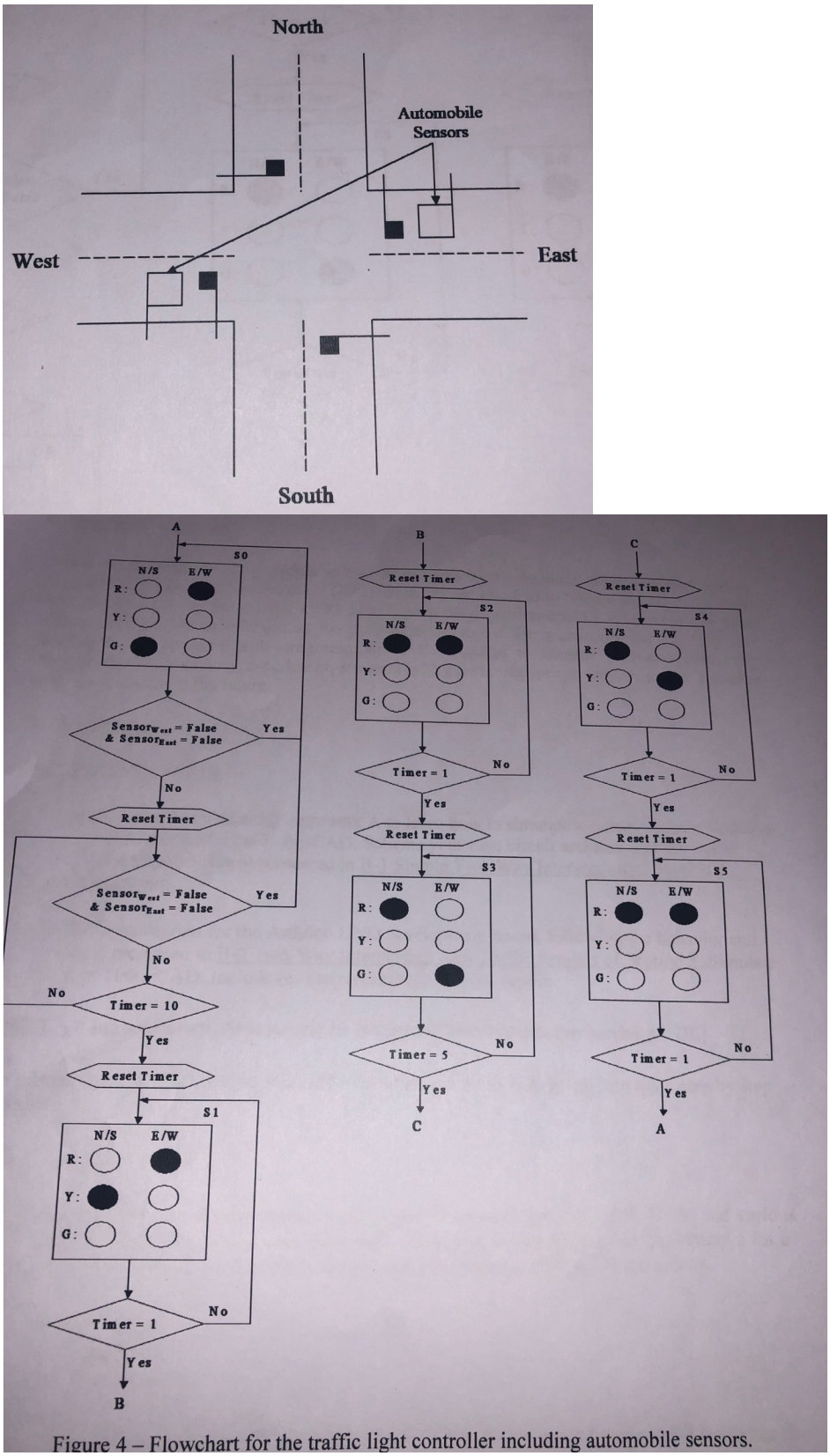


Figure 4 – Flowchart for the traffic light controller including automobile sensors.

b. Procedure

Simple two way intersection

light on the same direction are identical, we use 6 LEDs to build the circuit for the t
light controller. Use the following table for pinout setup of these LEDs.

Digital Pin 2	OUTPUT	Red North/South Direction
Digital Pin 3	OUTPUT	Yellow North/South Direction
Digital Pin 4	OUTPUT	Green North/South Direction
Digital Pin 5	OUTPUT	Red East/West Direction
Digital Pin 6	OUTPUT	Yellow East/West Direction
Digital Pin 7	OUTPUT	Green East/West Direction

Load the program in below Sample Program No.1 to the Arduino UNO. Show your

Traffic sensors

above part 1, you can add two push buttons to the circuit. With the modified circuit,
must write the C code to implement the program logic that is described in Figure 4.
the following table for pinout setup for the push buttons.

Digital Pin 12	INPUT	Automobile Sensor on West
Digital Pin 13	INPUT	Automobile Sensor on East

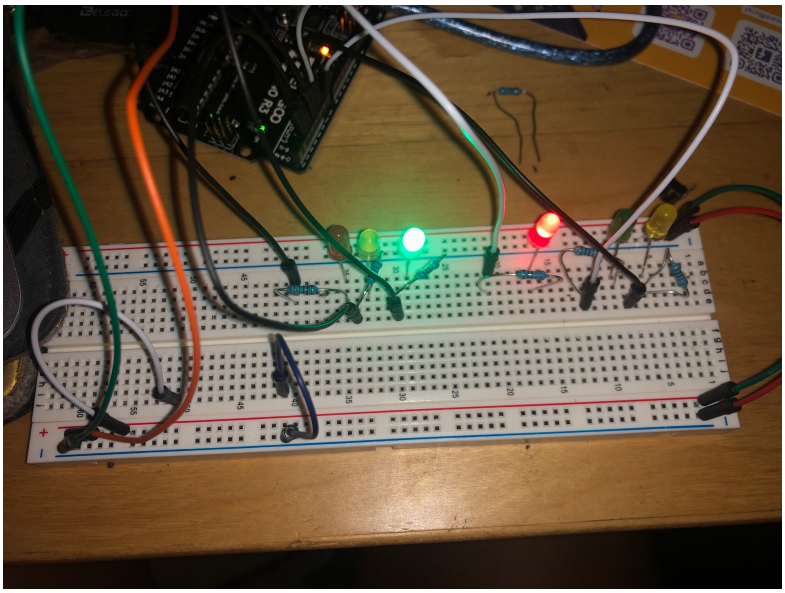
Finish the code, load it to the Arduino UNO board. Show your result in your report. Y
may record a short video and include it with your submission to the Blackboard.

c. Equipment

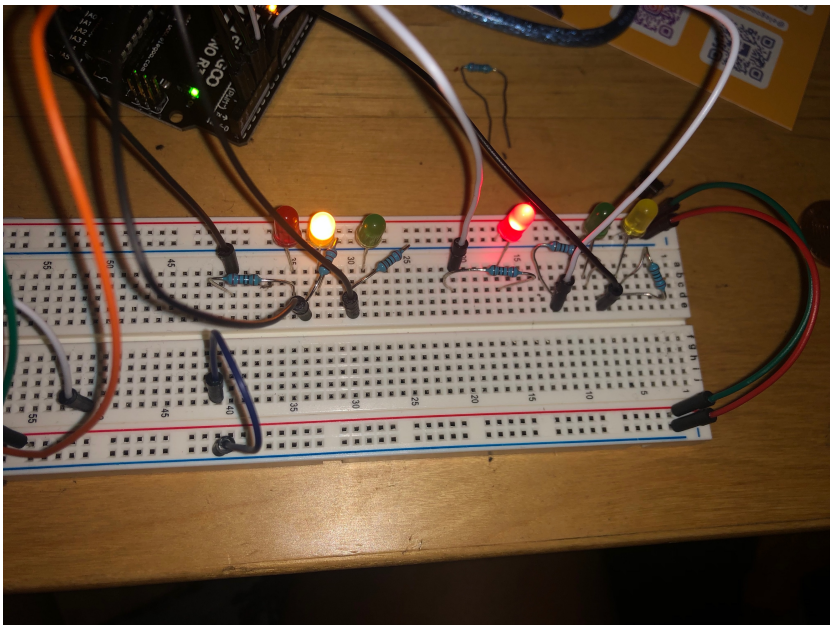
1) Arduino Uno R3
2) Red, green, yellow LEDs
3) Resistors 220 ohms.
4) Push buttons *2
5) wires

d. Results

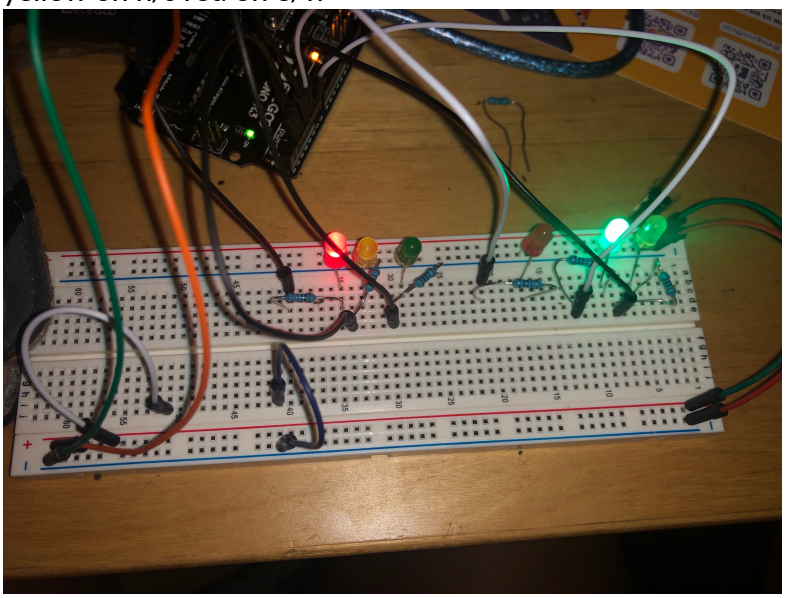
TWO WAY SIMPLE INTERSECTION



First case green on n/s red on e/w

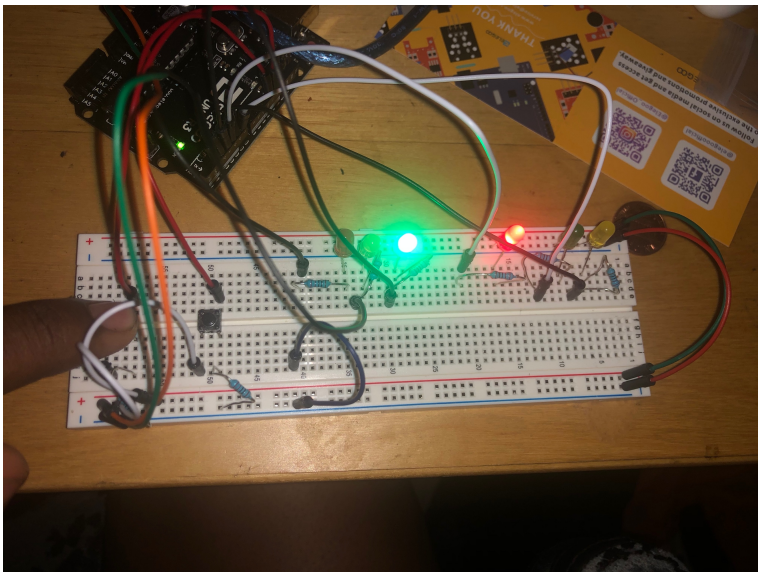


yellow on n/s red on e/w

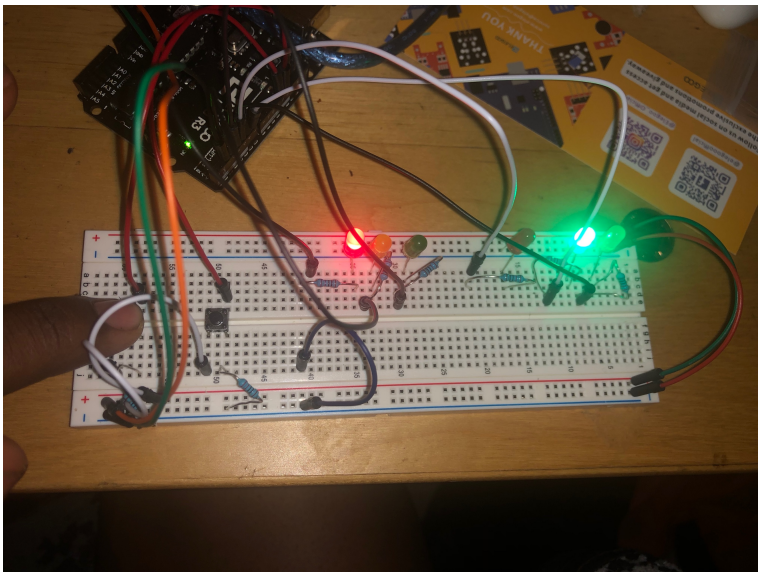


green on e/w and red on n/s

2WAY INTERSECTION WITH TRAFFIC SENSORS



Green on n/s and red on e/w then push button



Goes red on n/s and green on e/w

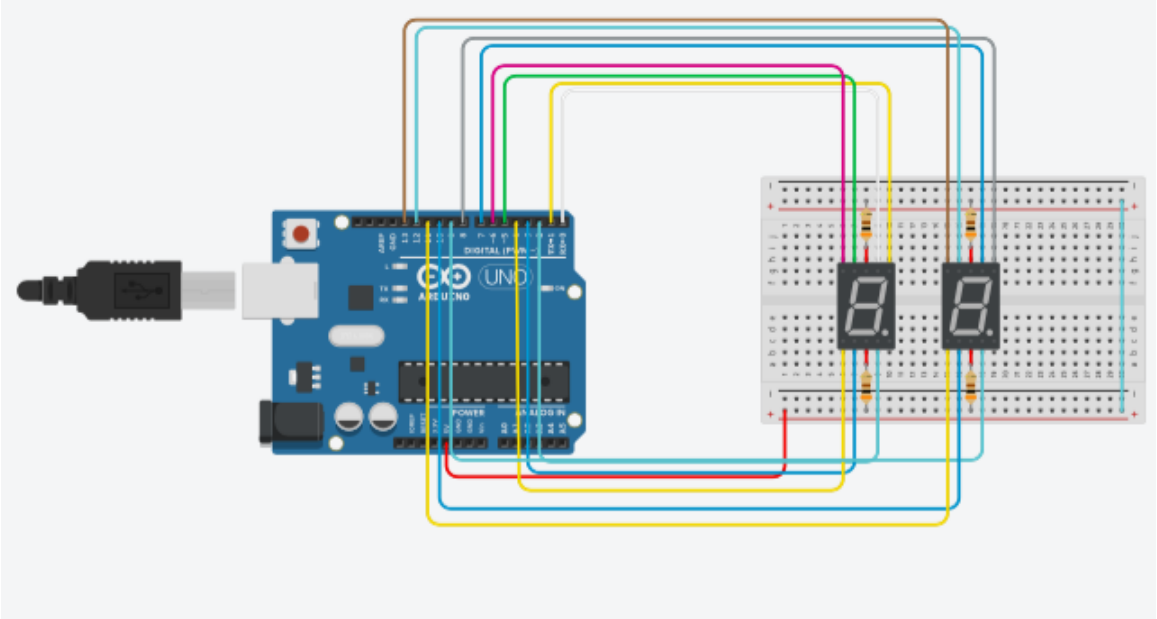
e. <u>Discussion</u>

1. See the schematic in procedure
2.
    a. Arduino is an open source computer hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. The board itself is a computer on a single integrated chip that has a CPU, Control Unit, Memory, Clock, I/O. The digital and analog input/output pins may be interfaced with various expansion boards and

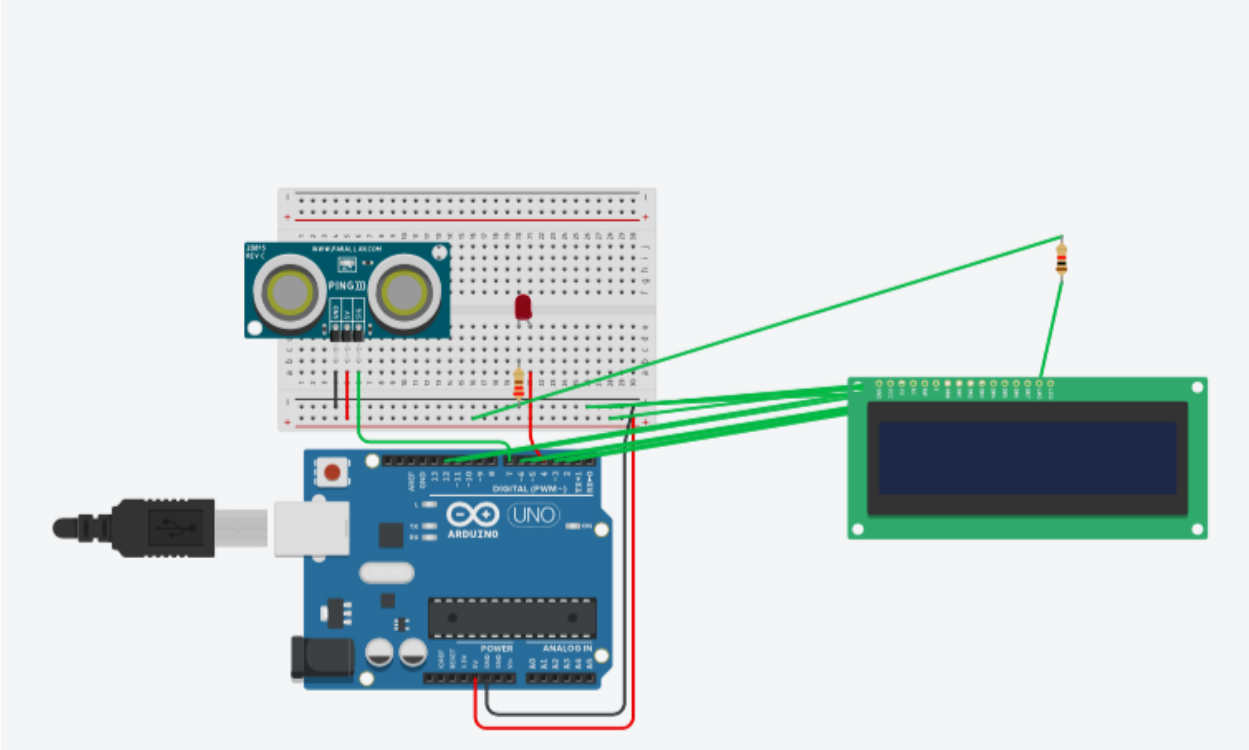
other circuits. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. It can support for SPI, I2C interfaces and also Wifi, Ethernet and USB connections. The Arduino uno that we are using is for general purpose development with electronics and coding. It has a ATmega328, 5V, 20 digital I/O's with 6 PWM channels.

b. Arduino can be used for a lot of different applications like home appliances, mobile devices, health temperature controller, internet of things. A lot of sensors like IR sensors or temperature sensors can be combined with the Arduino to create a smart device. The Arduino is really good for small project because it can only draw 20mA of power and anything over that would require a shield to amplify the power.

3. Picture of the circuit below , find code in appendix A3



4. Schematic of the implementation
   See appendix for code



Appendix code part B without sensors

```
// North_South LEDs
#define r_ns 2
#define y_ns 3
#define g_ns 4

// East_West LEDs
#define r_ew 5
#define y_ew 6
#define g_ew 7

//time_base define the running speed
//Set it to 1000 is in real time
//Set it to a smaller value for debugging
#define time_base 500

#define ns 12
#define ew 13

void setup() {
```

```
  // put your setup code here, to run once:
  // initialize all LEDs as output
  pinMode(r_ns,OUTPUT);
  pinMode(y_ns,OUTPUT);
  pinMode(g_ns,OUTPUT);
  pinMode(r_ew,OUTPUT);
  pinMode(y_ew,OUTPUT);
  pinMode(g_ew,OUTPUT);
  pinMode(ns,INPUT);
  pinMode(ew,INPUT);

}


void ChangeLedValue(byte number)
{
  digitalWrite(r_ns,bitRead(number,5));
  digitalWrite(y_ns,bitRead(number,4));
  digitalWrite(g_ns,bitRead(number,3));
  digitalWrite(r_ew, bitRead(number,2));
  digitalWrite(y_ew,bitRead(number,1));
  digitalWrite(g_ew, bitRead(number,0));

  }

void LightSequence_ns()
{
  ChangeLedValue(B001100);
  delay(time_base *10);
  ChangeLedValue(B010100);
  delay(time_base *1);
  ChangeLedValue(B100100);
  delay(time_base *1);
}
void LightSequence_ew()
{
  ChangeLedValue(B100001);
  delay(time_base *5);
  ChangeLedValue(B100010);
  delay(time_base *1);
  ChangeLedValue(B100100);
  delay(time_base *1);

}
```

```
void LightSequence()
{
  ChangeLedValue(B001100);
  delay(time_base*10);
  ChangeLed


}

void loop()
{
  // put your main code here, to run repeatedly:

  LightSequence();

}
```

Appendix code for part B with sensors

```
// North_South LEDs
#define r_ns 2
#define y_ns 3
#define g_ns 4

// East_West LEDs
#define r_ew 5
#define y_ew 6
#define g_ew 7

//time_base define the running speed
//Set it to 1000 is in real time
//Set it to a smaller value for debugging
#define time_base 500

#define ns 12
#define ew 13

void setup() {
  // put your setup code here, to run once:
  // initialize all LEDs as output
  pinMode(r_ns,OUTPUT);
  pinMode(y_ns,OUTPUT);
```

```
  pinMode(g_ns,OUTPUT);
  pinMode(r_ew,OUTPUT);
  pinMode(y_ew,OUTPUT);
  pinMode(g_ew,OUTPUT);
  pinMode(ns,INPUT);
  pinMode(ew,INPUT);

}


void ChangeLedValue(byte number)
{
  digitalWrite(r_ns,bitRead(number,5));
  digitalWrite(y_ns,bitRead(number,4));
  digitalWrite(g_ns,bitRead(number,3));
  digitalWrite(r_ew, bitRead(number,2));
  digitalWrite(y_ew,bitRead(number,1));
  digitalWrite(g_ew, bitRead(number,0));

  }

void LightSequence_ns()
{
  ChangeLedValue(B001100);

}
void LightSequence_ew()
{
  delay(time_base *10);
  ChangeLedValue(B010100);
  delay(time_base *1);
  ChangeLedValue(B100100);
  delay(time_base *1);
  ChangeLedValue(B100001);
  delay(time_base *5);
  ChangeLedValue(B100010);
  delay(time_base *1);
  ChangeLedValue(B100100);
  delay(time_base *1);

}

void LightSequence()
{
```

```
  ChangeLedValue(B001100);
  delay(time_base*10);
  ChangeLedValue(B010100);
  delay(time_base*1);
  ChangeLedValue(B100100);
  delay(time_base*1);
  ChangeLedValue(B100001);
  delay(time_base*5);
  ChangeLedValue(B100010);
  delay(time_base*1);
  ChangeLedValue(B100100);
  delay(time_base*1);
}

void carDetection()
{

 if(digitalRead(12)|| digitalRead(13))
  {
    LightSequence_ew();
  }
 else
 {
  LightSequence_ns();

 }

}


void loop()
{
  // put your main code here, to run repeatedly:

  carDetection();

}
```

Appendix count 0-99

```
void setup()
{
  for(int i =0;i<=13;i++)
    pinMode(i,OUTPUT);
```

```
}
const int number[11] = {
    0b1000000,
    0b1111001,
    0b0100100,
    0b0110000,
    0b0011001,
    0b0010010,
    0b0000010,
    0b1111000,
    0B0000000,
    0B0010000
  };

void loop()
{
  for(int numb =0;numb <10;numb++){
  display_numb(numb);
  }
}

void display_numb(const int numb){
int pin1,a,post;
  for(pin1=0,a=0;pin1<7;pin1++,a++){
          digitalWrite(pin1,bitRead(number[numb],a));
  }
  for (post=0;post<10;post++){
  display_post(post);
  delay(300);
  }
}

void display_post(const int x){
          int pin2,b;
  for(pin2=7,b=0;pin2<=13;pin2++,b++){
          digitalWrite(pin2,bitRead(number[x],b));
  }
}
```

Appendix for ultrasonic sensor controlling led

```
#include <LiquidCrystal.h>


// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
int distanceThreshold = 0;

int cm = 0;

int inches = 0;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setup()
{
  Serial.begin(9600);

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
}

void loop()
{
  lcd.clear();

  cm = 0.01723 * readUltrasonicDistance(7,12);
  inches = (cm / 2.54);

  lcd.setCursor(0, 0);
  // print the number of seconds since reset:
  lcd.print("cm: ");
  lcd.setCursor(4,0);
  lcd.print(cm);

  lcd.setCursor(0,1);
  lcd.print("inch: ");
```

```
  lcd.setCursor(6,1);
  lcd.print(inches);

  delay(1000);


  // set threshold distance to activate LEDs
  distanceThreshold = 350;
  // measure the ping time in cm
  cm = 0.01723 * readUltrasonicDistance(7, 7);
  // convert to inches by dividing by 2.54
  inches = (cm / 2.54);
  Serial.print(cm);
  Serial.print("cm, ");
  Serial.print(inches);
  Serial.println("in");
  if (cm > distanceThreshold) {
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }
  if (cm <= distanceThreshold && cm > distanceThreshold - 100) {
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }
  if (cm <= distanceThreshold - 100 && cm > distanceThreshold - 250) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
  }
  if (cm <= distanceThreshold - 250 && cm > distanceThreshold - 350) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
  }
  if (cm <= distanceThreshold - 350) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
  }
  delay(100); // Wait for 100 millisecond(s)
}
```