# Experiment No.1

## Traffic Light Controller Implementation on Arduino UNO

By: Anand Baraguru Venkateshmurthy (A20429060)

Instructor: Dr. Won-Jae Yi

ECE 442

Lab Date: 03-01-2019

Due Date: 03-08-2019

Acknowledgment: I acknowledge all of the work (including figures and codes) belongs to me and/or persons who are referenced.

Signature: Anand BV

## I. Introduction

### A. Purpose

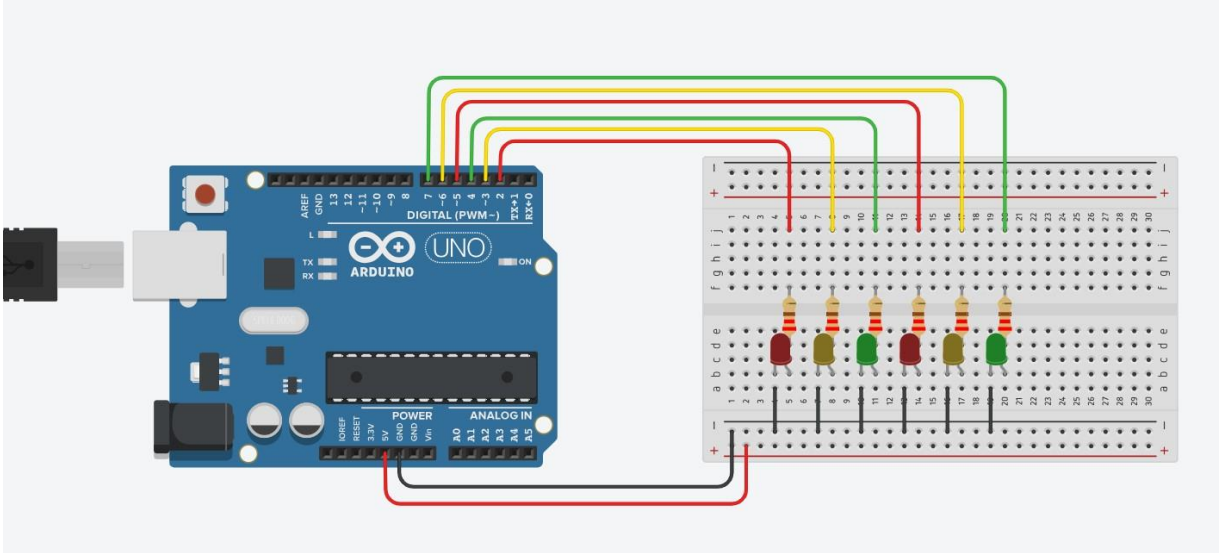
The main purpose of this experiment is to understand the basic operation of the Arduino UNO board, Arduino IDE and write a C program on Arduino. Traffic light controller is implemented on the Arduino board using C programming. A simple LED and pushbutton circuit must be built on the breadboard which is interfaced with the Arduino UNO board.

### Preliminary Assignment

Simulate traffic light controller system for two-way intersection and for a two-way intersection with traffic sensors on TinkerCAD and implement it using Arduino UNO R3 controller.

**Discussion**

1. A schematic diagram of your hardware design and a fully commented listing of the traffic light controller program.

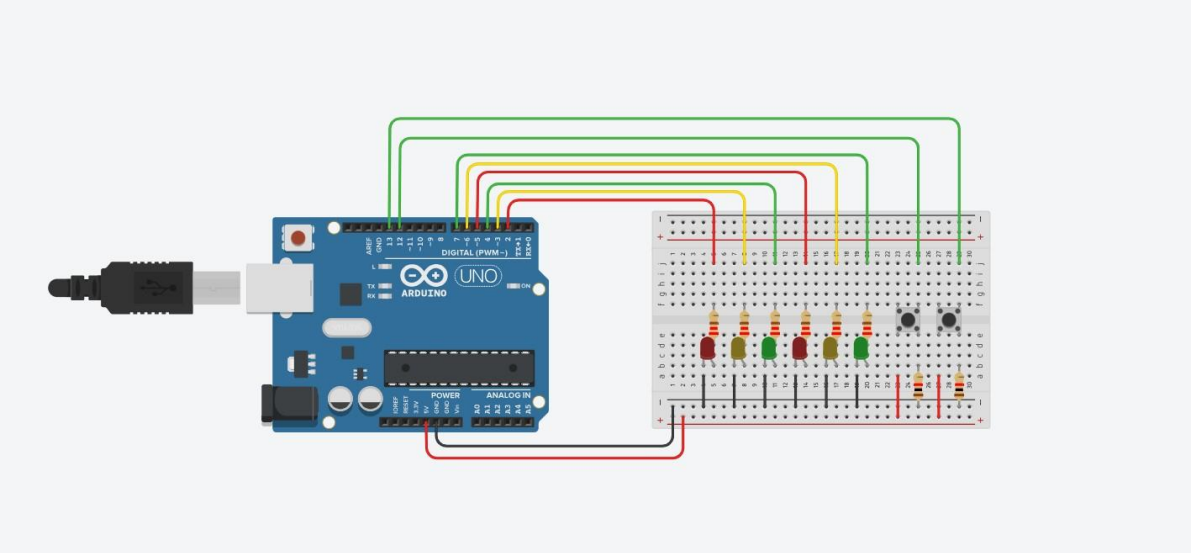



**Program:**

```
void setup()
{
  pinMode(5, OUTPUT); // digital pin 5 is set as output
  pinMode(6, OUTPUT); // digital pin 6 is set as output
  pinMode(7, OUTPUT); // sets digital pin 7 is set as output
  pinMode(2, OUTPUT); // sets digital pin 2 is set as output
  pinMode(3, OUTPUT); // sets digital pin 3 is set as output
  pinMode(4, OUTPUT); // sets digital pin 4 is set as output
  pinMode(13, INPUT); // sets digital pin 13 is set as input
  pinMode(12, INPUT); // sets digital pin 12 is set as input
}
void loop()
{
    digitalWrite(7, HIGH); //sets the digital pin 7 on
    digitalWrite(2, HIGH); //sets the digital pin 2 on
    digitalWrite(5, LOW); // sets the digital pin 5 off
    digitalWrite(4, LOW); // sets the digital pin 4 off
    digitalWrite(3, LOW); // sets the digital pin 3 off
```

```
        digitalWrite(6, LOW); // sets the digital pin 6 off
        delay(5000);              // waits for 5000 millisecond (s)
        digitalWrite(7, LOW); // sets the digital pin 7 off
        digitalWrite(6, HIGH); // sets the digital pin 6 on
        digitalWrite(2, HIGH); // sets the digital pin 2 on
        delay(2000);              // wait for 2000 millisecond (s)
        digitalWrite(6, LOW); // sets the digital pin 6 off
        digitalWrite(5, HIGH); // sets the digital pin 5 on
        delay(2000);              // wait for 2000 millisecond (s)
        digitalWrite(2, LOW); // sets the digital pin 2 off
        digitalWrite(4, HIGH); // sets the digital pin 4 on
        delay(2000);              // wait for 2000 millisecond (s)
        digitalWrite(4, LOW); // sets the digital pin 4 off
        digitalWrite(3, HIGH); // sets the digital pin 3 on
        delay(2000);              // wait for 2000 millisecond (s)
        digitalWrite(3, LOW); // sets the digital pin 3 off
        digitalWrite(2, HIGH); // sets the digital pin 2 on
        delay(2000);              // wait for 2000 millisecond (s)

    }
}
```

1. Connect the circuit using breadboard according to the schematic below for traffic light control system using sensors.




**Program:**

```
// 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 defines the running speed
#define time_base 1000
// North_South Sensor
#define sensor_ns 12
// East_West Sensor
#define sensor_ew 13
void setup() // initialize LEDs as output and sensors/push-buttons as input
{
  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(sensor_ns,INPUT);
  pinMode(sensor_ew,INPUT);
}
void ChangeLedValue(byte number) //logic to represent each LED with a bit number of the byte sequence
    {
 digitalWrite(r_ns,bitRead(number, 5)); //reads bit 5(MSB) of the byte only where bit 5 represents LED r_ns
 digitalWrite(y_ns,bitRead(number, 4)); //reads bit 4 of the byte only where bit 4 represents LED y_ns
 digitalWrite(g_ns,bitRead(number, 3)); //reads bit 3 of the byte only where bit 3 represents LED g_ns
 digitalWrite(r_ew,bitRead(number, 2)); //reads bit 2 of the byte only where bit 2 represents LED r_ew
 digitalWrite(y_ew,bitRead(number, 1)); //reads bit 1 of the byte only where bit 1 represents LED y_ew
 digitalWrite(g_ew,bitRead(number, 0)); //reads bit 0(LSB) of the byte only where bit 0 represents LED g_ew
}
void LightSequence() //to define which LEDs have to be set to HIGH or LOW according the traffic light signal sequence
{
  ChangeLedValue(B001100); //Green LED g_ns and Red LED r_ew is HIGH
  delay(time_base * 10); //delay for transition
  ChangeLedValue(B010100); //Yellow LED y_ns and Red LED r_ew is HIGH
  delay(time_base * 1); //delay for transition
  ChangeLedValue(B100100); //Red LED r_ns and Red LED r_ew is HIGH
  delay(time_base * 1); //delay for transition
  ChangeLedValue(B100001); //Red LED r_ns and Green LED g_ew is HIGH
  delay(time_base * 10); //delay for transition
```

```
  ChangeLedValue(B100010);
  delay(time_base * 1); //delay for transition
  ChangeLedValue(B100100);
  delay(time_base * 1); //delay for transition
}
void LightSequence_ns()
{
  ChangeLedValue(B001100);
  delay(time_base * 10);
  ChangeLedValue(B010100);
  ChangeLedValue(B100100);
  delay(time_base * 1);
}
void LightSequence_ew()
{
  ChangeLedValue(B100001);

      ChangeLedValue(B100010);
      delay(time_base * 1);
  ChangeLedValue(B100100);
  delay(time_base * 1);
    }
    void loop() //main code runs repeatedly
{
  if(digitalRead(sensor_ns==HIGH)) //when pushbutton representing sensor on North_South road
is pressed
   {
    delay(time_base * 1); //wait and confirming if push button is pressed
    if(digitalRead(sensor_ns==HIGH))
    {

       }
      }
```

```
 else if(digitalRead(sensor_ew==HIGH)) //when pushbutton representing sensor on
East_West road is pressed
       {
         delay(time_base * 1); //wait and confirming if push button is pressed
         if(digitalRead(sensor_ew==HIGH))
         {
          ();         }

  }

else

  {

   LightSequence();

  }

}
```

2. Write a brief discussion on (a) what is Arduino, (b) what Arduino can achieve and what is the shortcoming.

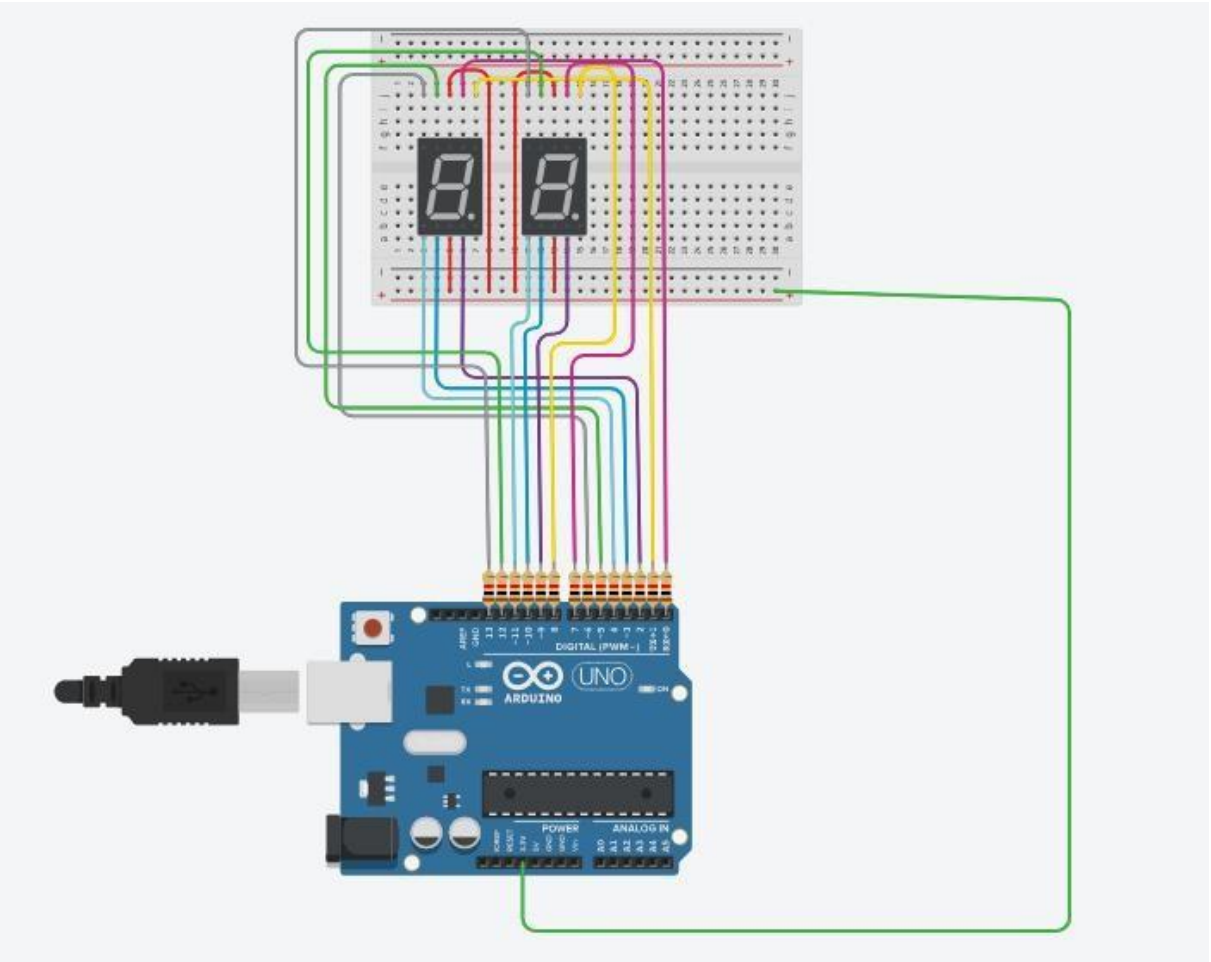
Solution: Arduino is an open source computing platform that is used for general purpose programming. Arduino is based on ATmegaP328P microcontroller board and is equipped with 14 digital I/O pins. Variety of microcontrollers and microprocessors are used in designing of Arduino board and the microcontrollers are programmed using C and C++ languages. For loading programs from personal computer, Arduino board provides serial communication interface like Universal Serial Bus (USB). To add functionality to the Arduino board "shields" are used such as TFT touchscreen shield, Motor/Servo Shield, Ethernet Shield etc. Arduino consists of a programmable board and Integrated Development Environment (IDE) that is upload code from the computer to the board.

Arduino is used in many IoT applications such as Smart home, Irrigation Controller, Heartbeat monitoring, Structural health monitoring system, Smart lighting, Robotic arm.

Shortcomings of Arduino are:

1) Size: Size of Arduino board is big. Because of which even the shields which are used to increase the functionality of the Arduino are also big in structure.
2) Shields and other modules should be used to enhance the functionality of the Arduino.

3. Implement a system that will count from 0 to 99 and display on two 7 segment LED displays in TinkerCAD. A schematic diagram of your hardware design and fully commented listing of the program need to be included in the lab report.



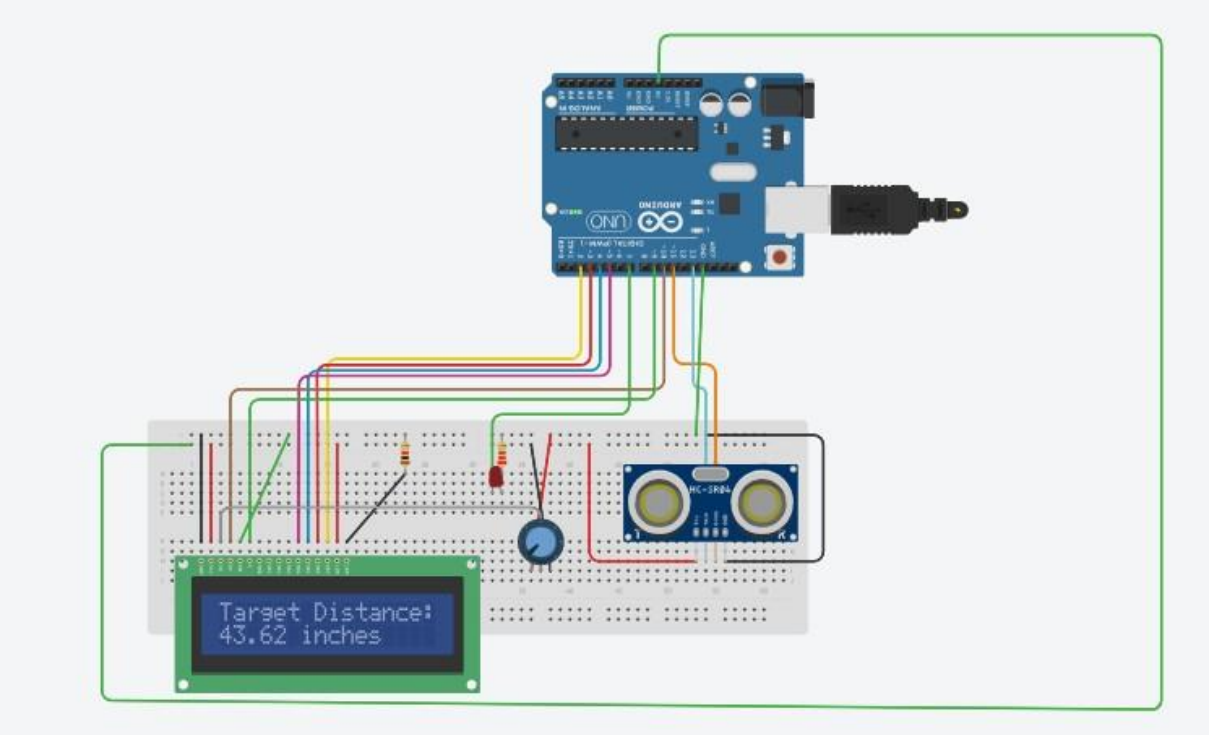

Program:

```
void setup()
{
  for (int i = 0; i <= 13; i++)
    pinMode(i, OUTPUT); //pins from 0 to 13 are set as output pins
 }
// The line below is the array containing all the binary numbers for the digits on a seven segment display from 0 to 9
const int number[11] = {0b1000000, 0b1111001, 0b0100100, 0b0110000, 0b0011001, 0b0010010, 0b0000010, 0b1111000, 0b0000000, 0b0010000};


void loop()
{
```

```
  for (int tens = 0; tens < 10; tens++)


  {
    display_tens(tens);
  }
}

void display_tens(const int tens)
{
  int pin1, a, ones;
  //pin1 is used to deal with pins of the 1st Seven Segment Display which desplays the tens digits

  for (pin1 = 0, a = 0; pin1 < 7; pin1++, a++)
  {
    digitalWrite(pin1, bitRead(number[tens], a));
  }
  for (ones = 0; ones < 10; ones++)
  {
    display_ones(ones);
    delay(300); // wait for 300 milliseconds
  }
}

void display_ones(const int x)
{ int pin2, b;
//pin2 is used to deal with pins of the 2nd Seven Segment Display which desplays the ones digit

  for (pin2 = 7, b = 0; pin2 <= 13; pin2++, b++)
  {
    digitalWrite(pin2, bitRead(number[x], b));


  }
```

}

4. Implement a proximity recognition system using an LED light and an ultrasonic distance sensor with LCD display in TinkerCAD. Your program should print object's distance from the ultrasonic sensor on the LCD display. Also, an LED display should be used to indicate object's proximity where the LED light gets brighter as the object gets closer to the sensor, and LED light gets dimmer as the object moves away from the sensor. A scheatic diagram of your hardware design and a fully commented listing of the program need to be included in the lab report.




**Program:**

```
#include <LiquidCrystal.h> //Load Liquid Crystal Headerfile
#define bulb 7

LiquidCrystal LCD(10, 9, 5, 4, 3, 2); //Create Liquid Crystal Object called LCD

int trigPin=13; //Sensor Trip pin connected to Arduino pin 13
int echoPin=11;  //Sensor Echo pin connected to Arduino pin 11
int myCounter=0; //declare your variable myCounter and set to 0
int servoControlPin=6; //Servo control line is connected to pin 6
float pingTime; //time for ping to travel from sensor to target and return
float targetDistance; //Distance to Target in inches
float speedOfSound=776.5; //Speed of sound in miles per hour when temp is 77 degrees.

void setup() {

Serial.begin(9600);// Serial communication baud rate set to 9600 bps
pinMode(trigPin, OUTPUT);// Sensor Trip pin set as output
pinMode(echoPin, INPUT); //Sensor echopin set as input
pinMode (bulb, OUTPUT);
```

```
LCD.setCursor(0,0); // LCD cursor is set to upper left corner, column 0, row 0
LCD.print("Target Distance:"); //Message is printed on the first row
}
 //void Light() // Light subroutine is started

//{ digitalWrite(bulb, HIGH); //light is turned on

//delay (15000); //wait 15 seconds

//digitalWrite(bulb, LOW); //light is turned off

//}
void loop() {

  digitalWrite(trigPin, LOW); //trigger pin is set low
  delayMicroseconds(15); //Delay in high state
  digitalWrite(trigPin, LOW); //trigger pin has now been sent
  delayMicroseconds(10); //Delay in high state


  pingTime=pingTime/1000000; //convert pingTime to seconds by dividing by 1000000
(microseconds in a second)
  pingTime=pingTime/3600; //convert pingtime to hourse by dividing by 3600 (seconds in an hour)
targetDistance=targetDistance/2; //Remember ping travels to target and back from target, so you
must divide by 2 for actual target distance.
  targetDistance= targetDistance*63360;    //Convert miles to inches by multipling by 63360
   // if (targetDistance < 100) //if the distance is less than 13 cm
//{
// Light(); //execute the Light subroutine
//}
  LCD.print("               "); //Print blanks to clear the row
  LCD.setCursor(0,1); //Set Cursor again to first column of second row
  LCD.print(targetDistance); //Print measured distance
  LCD.print(" inches"); //Print units
  delay(250); //pause to let things settle


  }
```

References:

1). https://www.makeuseof.com/tag/arduino-technology-explained/

2). http://engineerexperiences.com/advantages-and-disadvatages.html

3). Lab manual

4) https://www.tinkercad.com