PROJECT TITLE : TRAFFIC MANAGEMENT

PHASE 3 : IoT devices and developing a script on IoT devices as per the project requirements .

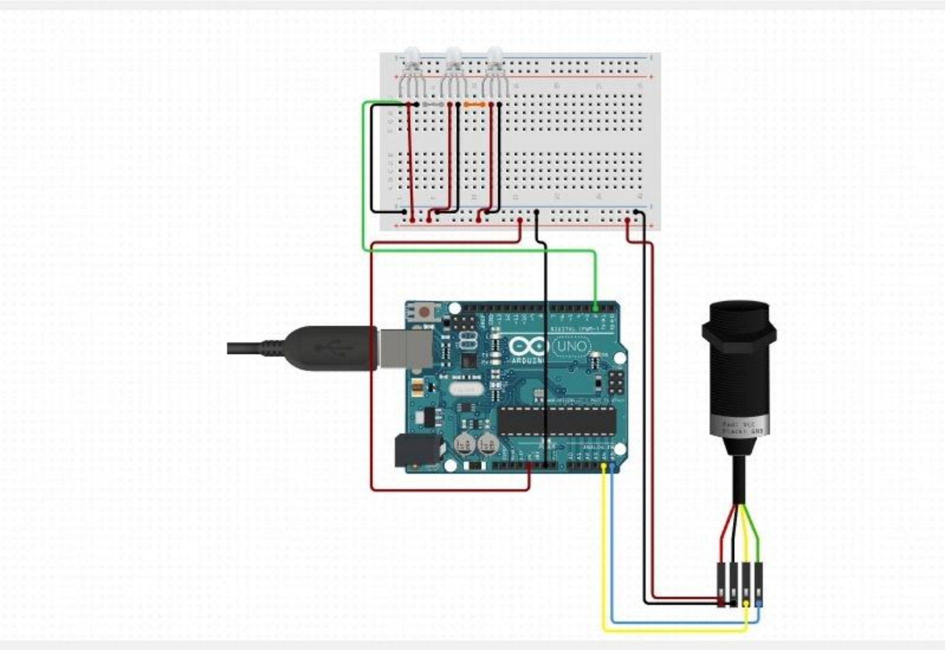
<u>COMPONENTS USED :</u>

Microcontroller : Arduino UNO .

LED lights – 3 NOS .

IR Positioning camera .

CIRCUIT DIAGRAM :



Code scripting :

```
// Include Libraries

#include "Arduino.h"

#include "Adafruit_NeoPixel.h"
```

```
// Pin Definitions
#define LEDRGB_1_PIN_DIN	2



// Global variables and defines
#define LedRGB_1_NUMOFLEDS 3
// object initialization
Adafruit_NeoPixel LedRGB_1(LEDRGB_1_PIN_DIN);


// define vars for testing menu
const int timeout = 10000;	//define timeout of 10 sec
char menuOption = 0;
long time0;

// Setup the essentials for your circuit to work. It runs first every time your circuit is powered with electricity.
void setup()
{
  // Setup Serial which is useful for debugging
  // Use the Serial Monitor to view printed messages
  Serial.begin(9600);
  while (!Serial) ; // wait for serial port to connect. Needed for native USB
  Serial.println("start");

  LedRGB_1.begin(); // This initializes the NeoPixel library.
  LedRGB_1.show(); // Initialize all leds to 'off'
  menuOption = menu();

}
```

```
// Main logic of your circuit. It defines the interaction between the components you selected. After setup, it runs over and over again, in an eternal loop.
void loop()
{



    if(menuOption == '1')
    {
    // Disclaimer: The IR Positioning Camera is in testing and/or doesn't have code, therefore it may be buggy. Please be kind and report any bugs you may find.
    }
    else if(menuOption == '2') {
    // LED - RGB Addressable, PTH, 5mm Diffused (5 Pack) #1 - Test Code
    for(int i=0 ; i <= LedRGB_1_NUMOFLEDS ; i++){
    for (int k = 0 ; k <= 255 ; k++) {
    // set leds Color to RGB values, from 0,0,0 up to 255,255,255
    LedRGB_1.setPixelColor(i, LedRGB_1.Color(255-k,k,100)); // turn on green color on led #i.
    if (i > 0)
    LedRGB_1.setPixelColor(i-1, LedRGB_1.Color(0,0,0)); // turn off last led
    LedRGB_1.show(); //update led color to the hardware.
    delay(1);
    }
    }

    }

    if (millis() - time0 > timeout)
    {
        menuOption = menu();
    }
```

```
}




// Menu function for selecting the components to be tested
// Follow serial monitor for instrcutions
char menu()
{

   Serial.println(F("\nWhich component would you like to test?"));
   Serial.println(F("(1) IR Positioning Camera"));
   Serial.println(F("(2) LED - RGB Addressable, PTH, 5mm Diffused (5 Pack) #1"));
   Serial.println(F("(menu) send anything else or press on board reset button\n"));
   while (!Serial.available());

   // Read data from serial monitor if received
   while (Serial.available())
   {
      char c = Serial.read();
      if (isAlphaNumeric(c))
      {

         if(c == '1')
                                    Serial.println(F("Now Testing IR Positioning Camera - note that this
component doesn't have a test code"));
                        else if(c == '2')
                                    Serial.println(F("Now Testing LED - RGB Addressable, PTH, 5mm Diffused (5
Pack) #1"));
         else
         {
            Serial.println(F("illegal input!"));
            return 0;
            }
            time0 = millis();
            return c;
        }
    }
}

#include "Arduino.h"
#include "Adafruit_NeoPixel.h"



// Pin Definitions
#define LEDRGB_1_PIN_DIN	2




// Global variables and defines
#define LedRGB_1_NUMOFLEDS 3
// object initialization
Adafruit_NeoPixel LedRGB_1(LEDRGB_1_PIN_DIN);



// define vars for testing menu
const int timeout = 10000;       //define timeout of 10 sec
char menuOption = 0;
long time0;

void setup()
{
    // Setup Serial which is useful for debugging
    // Use the Serial Monitor to view printed messages
```

```
 Serial.begin(9600);
 while (!Serial) ; // wait for serial port to connect. Needed for native USB
 Serial.println("start");

 LedRGB_1.begin(); // This initializes the NeoPixel library.
 LedRGB_1.show(); // Initialize all leds to 'off'
 menuOption = menu();

}
void loop()
{


 if(menuOption == '1')
 {
 }
 else if(menuOption == '2') {
 // LED - RGB Addressable, PTH, 5mm Diffused (5 Pack) #1 - Test Code
 for(int i=0 ; i <= LedRGB_1_NUMOFLEDS ; i++){
 for (int k = 0 ; k <= 255 ; k++) {
 // set leds Color to RGB values, from 0,0,0 up to 255,255,255
 LedRGB_1.setPixelColor(i, LedRGB_1.Color(255-k,k,100)); // turn on green color on led #i.
 if (i > 0)
 LedRGB_1.setPixelColor(i-1, LedRGB_1.Color(0,0,0)); // turn off last led
 LedRGB_1.show(); //update led color to the hardware.
 delay(1);
 }
 }

 }
```

```
  if (millis() - time0 > timeout)
  {
    menuOption = menu();
  }

}



char menu()
{

  Serial.println(F("\nWhich component would you like to test?"));
  Serial.println(F("(1) IR Positioning Camera"));
  Serial.println(F("(2) LED - RGB Addressable, PTH, 5mm Diffused (5 Pack) #1"));
  Serial.println(F("(menu) send anything else or press on board reset button\n"));
  while (!Serial.available());
  while (Serial.available())
  {
    char c = Serial.read();
    if (isAlphaNumeric(c))
    {

      if(c == '1')
                      Serial.println(F("Now Testing IR Positioning Camera - note that this component doesn't have a test code"));
              else if(c == '2')
                      Serial.println(F("Now Testing LED - RGB Addressable, PTH, 5mm Diffused (5 Pack) #1"));
      else
      {
        Serial.println(F("illegal input!"));
```

```
            return 0;
          }
          time0 = millis();
          return c;
        }
    }
}
```