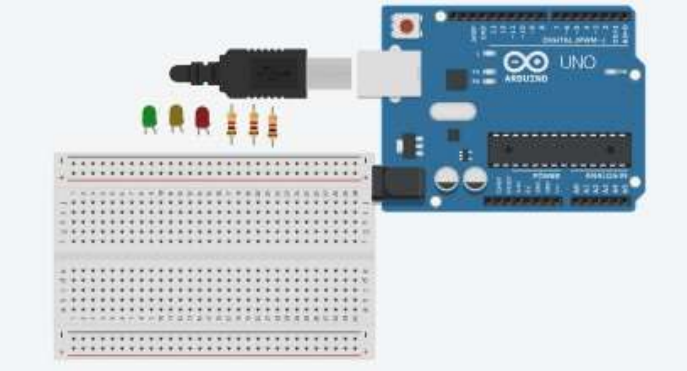
Experiment: 3

Student Name: Virat Samdarshi **UID: 22BCS12648**
Branch: B.E(CSE) **Section/Group: IOT_627-B**
Semester: Fifth **Date of Performance:02/08/24**
Subject Name: IOT LAB **Subject Code: 22CSP-329**

1. **Aim:** To Develop a smart traffic light management system with the help of IoT.

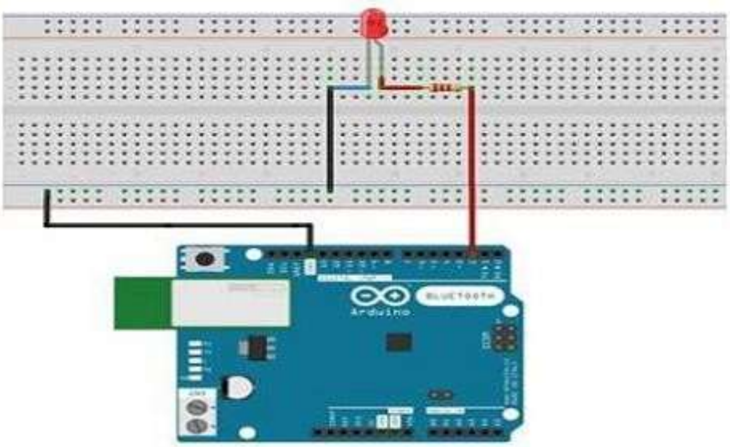
2. **Objective:**
   1. Learn about interfacing.
   2. Learn about IoT programming.

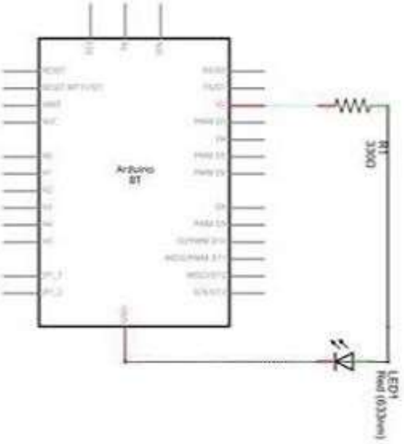
   Software Required:
   1. Tinker CAD
   2. LEDs (Red, Yellow, Green)
   3. Breadboard
   4. Arduino Uno
   5. Resistors

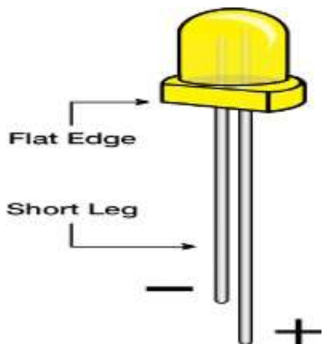


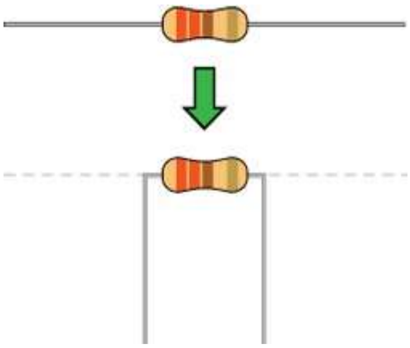
3. Procedure :

LEDs are small, powerful lights that are used in many different applications. To start, we will work on blinking an LED, the Hello World of microcontrollers. It is as simple as turning a light on and off. Establishing this important baseline will give you a solid foundation as we work towards experiments that are more complex. Follow the circuit diagram and hook up the components on the breadboard as shown in the image given below.









To find out the polarity of an LED, look at it closely. The shorter of the two legs, towards the flat edge of the bulb indicates the negative terminal.

Components like resistors need to have their terminals bent into 90° angles in order to fit the breadboard sockets properly. You can also cut the terminals shorter.

Connections:

This is the circuit diagram for the traffic light controller by using Arduino.

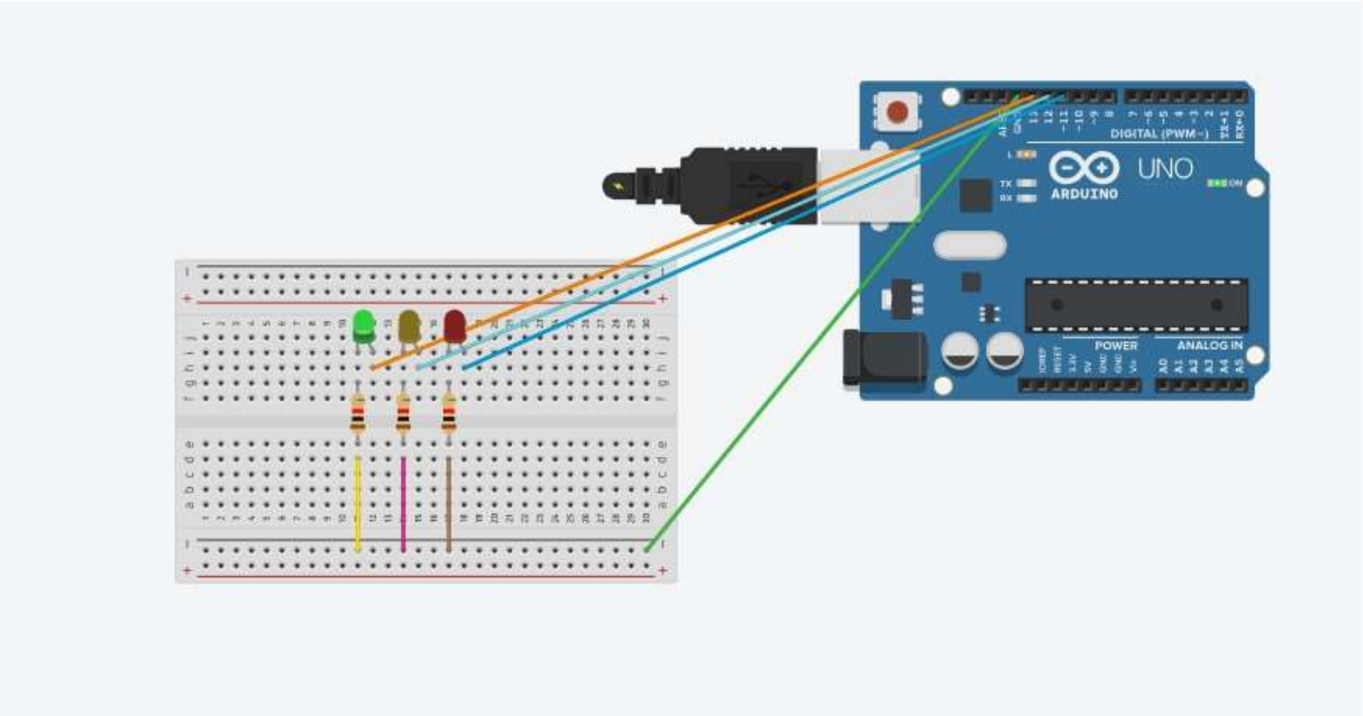
- Connect LEDs on the breadboard as Red, Yellow, Green, respectively.
- Connect the negative terminal of the LED and connect the 220 Ohm resistor in series.
- Connect these negative terminals to the ground.
- Connect the positive terminal of the LEDs to the pins 2 to 10, respectively.
- Power the breadboard by using 5V and GND on the Arduino.

4. Code:

```
int red=13;
int yellow=12;
int green=11;
void setup()
{
pinMode (red, OUTPUT); pinMode (yellow, OUTPUT); pinMode (green, OUTPUT);
void loop(){
```

```
digitalWrite(red, HIGH); delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(red, LOW); delay(1000); // Wait for 1000 millisecond(a)
digitalWrite(yellow, HIGH); delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(yellow, LOW); delay(1800); // Wait for 1000 millisecond(s)
digitalWrite(green, HIGH);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(green, LOW);
delay (1400); Wait for 1000 millisecond(s)
}
```

5. Output:



6. Learning Outcomes:

- Learn the basics of IoT through practical simulations.
- Gain experience interfacing LEDs and sensors with Arduino Uno.
- Enhance skills in writing and debugging Arduino code.
- Design and test circuits using the Tinkercad simulation platform.
- Improve problem-solving and troubleshooting skill.