# Experiment – 3

## Aim: Create the circuit to connect the Arduino board to two RGB LEDs / Normal LEDs that blinks alternatively.

### Components

- Arduino Uno R3
- Small Breadboard
- 3 LED RGB
- Jumper cable
- Resistor (200 ohms)

### Theory

1. Arduino Uno R3

   The Arduino Uno R3 is the main microcontroller in this circuit. It acts as the brain of the setup, executing the programmed code to control the LED. It provides both power and signal to the circuit, making it an essential component for automation and control projects.
2. Small Breadboard

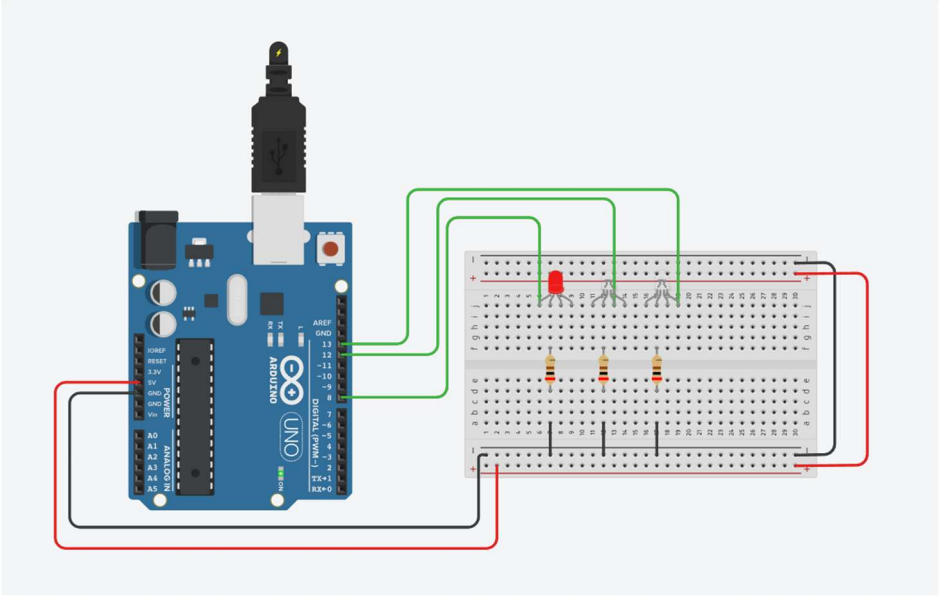
   The small breadboard is used for easy and temporary circuit connections without soldering. It allows the LED and other components to be connected securely while maintaining flexibility for modifications.
3. LED RGB:

   An RGB LED is a single LED that contains red, green, and blue diodes, allowing it to produce a wide range of colors by adjusting the brightness of each. It typically has four pins—one common (anode or cathode) and three control pins for color mixing.
4. Jumper Cable:

   Jumper cables are flexible wires with male or female connectors used to create electrical connections between components on a breadboard or with an Arduino. They are essential for linking circuits without the need for soldering.
5. Resistor (200 Ohms):

   A 200Ω resistor is used to limit current flow and protect components like LEDs from burning out. It ensures safe operation by reducing excess voltage and current in the circuit.

Circuit Design



Procedure to Connect Two LEDs to an Arduino and Make Them Blink Alternately on Tinkercad

Step 1: Add Components to the Workspace

1. Open Tinkercad Circuits and click “Create New Circuit.”
2. Search for Arduino Uno R3 and add it to the workspace.
3. Search for a small breadboard and place it next to the Arduino.
4. Search for two LEDs (either normal or RGB LEDs).
5. Search for two resistors (220Ω or 330Ω) and place each in series with an LED.
6. Use jumper wires to connect the components correctly.

Step 2: Build the Circuit Connections

1. Place the first LED1 on the breadboard.
2. Connect the anode (+) of LED1 to digital pin 8 on the Arduino.
3. Connect the cathode (-) of LED1 to one side of a 220Ω resistor and the other side of the resistor to GND.
4. Repeat the same process for LED2, but connect the anode to digital pin 9 on the Arduino.

Step 3: Code for Alternate Blinking

```
void setup() {
  pinMode(8, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}
void loop() {
  digitalWrite(8, HIGH);
  delay(500); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(500); // Wait for 1000 millisecond(s)
  digitalWrite(13, HIGH);
  delay(500); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(500); // Wait for 1000 millisecond(s)
```

```
  digitalWrite(12, HIGH);
 delay(500); // Wait for 1000 millisecond(s)
 digitalWrite(12, LOW);
 delay(500); // Wait for 1000 millisecond(s)
}
```

Step 4: Simulate the Circuit

1. Click “Start Simulation” at the top of the screen.
2. Observe the LEDs blinking alternately every second.

## Conclusion

In this project, we successfully connected an Arduino Uno R3 to three RGB LEDs using a breadboard, jumper wires, and resistors to create an alternating blinking effect. The Arduino controls the LEDs by sending signals through digital pins, while the resistors help regulate the current to prevent damage. This experiment demonstrates the fundamental principles of controlling LEDs with microcontrollers, which can be expanded into more advanced lighting projects, such as color-changing effects, multiple LED patterns, or interactive lighting systems. Using Tinkercad Circuits, we can test and modify the design before implementing it with real hardware.