# Experiment – 2

## Aim: Create a circuit with code to flash an LED connected to and Arduino board.

### Components

- Arduino Uno R3
- Small Breadboard
- 1 LED
- 1 Jumper Cable

### Theory

1. Arduino Uno R3

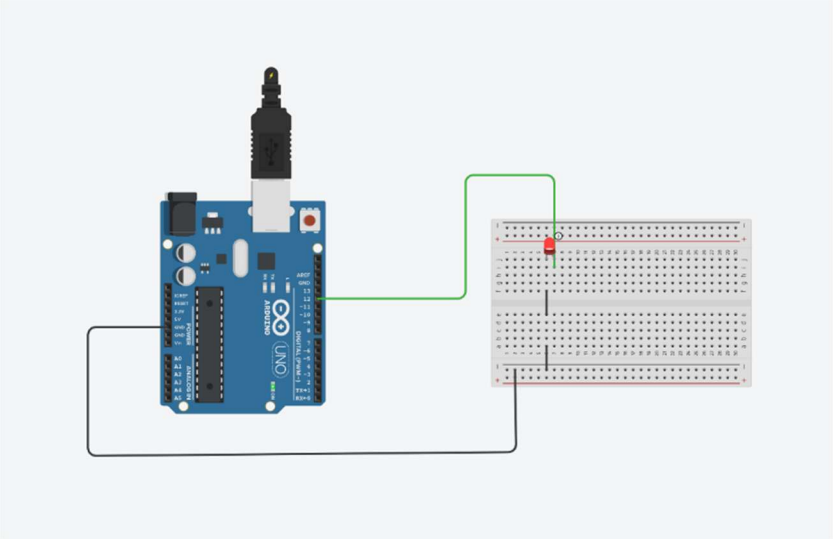
   The Arduino Uno R3 is the main microcontroller in this circuit. It acts as the brain of the setup, executing the programmed code to control the LED. It provides both power and signal to the circuit, making it an essential component for automation and control projects.
2. Small Breadboard

   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. The LED

   The LED (Light Emitting Diode) is the primary output component in this circuit. It emits light when current flows through it, and its blinking pattern is controlled by the Arduino using programmed delays.
4. Jumper Cable

   The cables (jumper wires) are used to establish connections between the Arduino, the LED, and the breadboard. They ensure proper electrical pathways for power and signals, making the circuit functional and organized.

Circuit Design



Procedure to Flash an LED using Arduino on Tinkercad

Step 1: Add Components to the Workspace

1. In the components panel, search for Arduino Uno R3 and drag it to the workspace.
2. Search for a breadboard (small) and place it on the workspace next to the Arduino.
3. Search for an LED and place it on the breadboard.
4. Use jumper wires to connect the components correctly.

Step 2: Build the Circuit Connections

1. Connect the longer leg (anode +) of the LED to pin 12 of the Arduino.
2. Connect the shorter leg (cathode -) of the LED to one side of the resistor.
3. Connect the other side of the resistor to GND (ground) on the Arduino using a jumper wire.
4. Ensure all connections are secure and properly placed.

Step 3: Code for LED Blinking

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

Step 4: Simulate the Circuit

Click on “Start Simulation” at the top of the screen.

Observe the LED turning on and off every second, indicating the code is working properly.

## Conclusion

Using Tinkercad Circuits to simulate an Arduino LED blinking project is an effective way to learn about basic electronics and coding. The platform allows users to design, test, and debug circuits without needing physical components, making it ideal for beginners. Through this project, we learned how to add components, connect circuits, and write code to control an LED. This foundational knowledge can be expanded to build more complex projects using additional sensors, motors, and modules. Tinkercad provides a cost-effective and accessible way to experiment with Arduino programming and circuit design.