Experiment – 4

Aim: Design the circuit using a push button. Activate and deactivate the LED attached to the Arduino Board.

Components

- Arduino Uno R3
- Small Breadboard
- Jumper cable
- LED
- Resistors
- Push Button

Theory

1. Arduino Uno R3

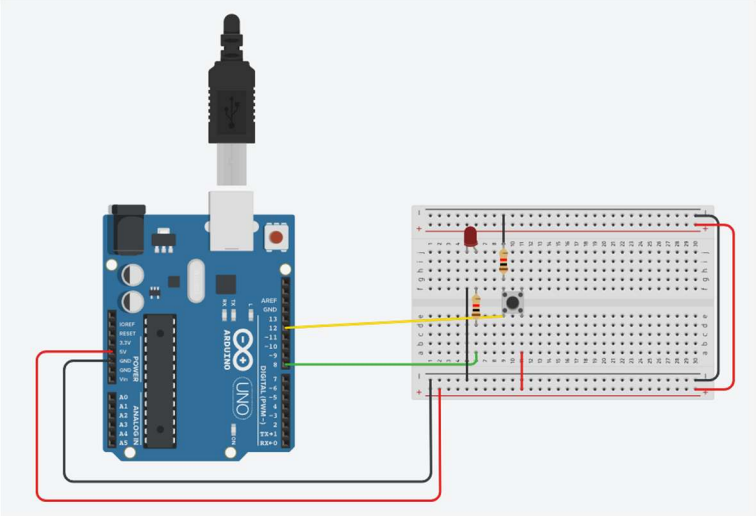
   The Arduino Uno R3 is the main microcontroller in this circuit. It acts as the brain of the setup, executing the programmed code to control the LED. It provides both power and signal to the circuit, making it an essential component for automation and control projects.
2. Small Breadboard

   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. Jumper Cable:

   Used to make electrical connections between the Arduino, breadboard, and components.
4. 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.
5. Resistors: The 220Ω resistor protects the LED, while the 10kΩ resistor acts as a pull-down resistor to prevent false button presses.
6. Push Button: A switch that allows user interaction by providing an ON/OFF signal to the Arduino.

Circuit Design



Procedure to Design a Push Button Controlled LED Circuit using Arduino on Tinkercad

Step 1: Add Components to the Workspace

1. Search for Arduino Uno R3 and add it to the workspace.
2. Search for a small breadboard and place it next to the Arduino.
3. Search for an LED and place it on the breadboard.
4. Search for a push button and position it on the breadboard.
5. Search for resistors (220Ω for the LED and 10kΩ for the push button) and place them in the circuit.
6. Use jumper cables to establish connections between the components.

Step 2: Build the Circuit Connections

LED Connections:

1. Anode (+) of LED → Connect to digital pin 8 on the Arduino.
2. Cathode (-) of LED → Connect to one side of a 220Ω resistor.
3. Other side of the resistor → Connect to GND.

Push Button Connections:

1. One side of the push button → Connect to 5V on the Arduino.
2. The other side of the push button → Connect to digital pin 7 on the Arduino.
3. A 10kΩ resistor → Connect between the button output (digital pin 7) and GND (pull-down resistor to prevent floating signals).

Step 3: Code for LED Control

```
int push_btn_state = 0;

void setup() {
        pinMode(12, INPUT);
        pinMode(8, OUTPUT);
}
void loop() {
        push_btn_state = digitalRead(12);
  if(push_btn_state == HIGH) {
        digitalWrite(8, HIGH);
  } else if(push_btn_state == LOW){
```

```
        digitalWrite(8, LOW);
  }
}
```

Step 4: Simulate the Circuit

1. Click **"Start Simulation"** in Tinkercad.
2. Press the **push button** to see the LED turn on and release it to turn it off.

Conclusion

This project demonstrates how to control an LED using a push button with an Arduino, which is a fundamental concept in digital input handling. The push button serves as an input device, and the LED provides visual feedback based on button presses. This setup can be expanded into more complex circuits such as button-controlled relays, home automation switches, or interactive systems. Using Tinkercad, we can simulate and refine the circuit before implementing it with real components.