Task 1: Push Button Controlled LEDs using Arduino

Abdo Wael

Objective

Build a circuit using Arduino UNO with two LEDs (Red and Green) and one push button. The circuit should behave as follows:

- On the **first press**, turn on the Green LED.
- On the **second press**, turn on the Red LED.
- On the **third press**, turn off both LEDs.
- Repeat the cycle.

Tinkercad Link

Click here to view the project in Tinkercad

Components Required

- Arduino Uno R3 – 1
- Breadboard – 1
- LED (Red, Green) – 2
- Push Button – 1
- Resistors 220Ω – 3
- Jumper Wires – several

Circuit Diagram

Circuit Explanation

- Green LED anode is connected to pin 11 through a 220Ω resistor.
- Red LED anode is connected to pin 9 through a 220Ω resistor.
- The push button is connected between pin 4 and 5V. A pull-down resistor is used to connect the button leg to GND.

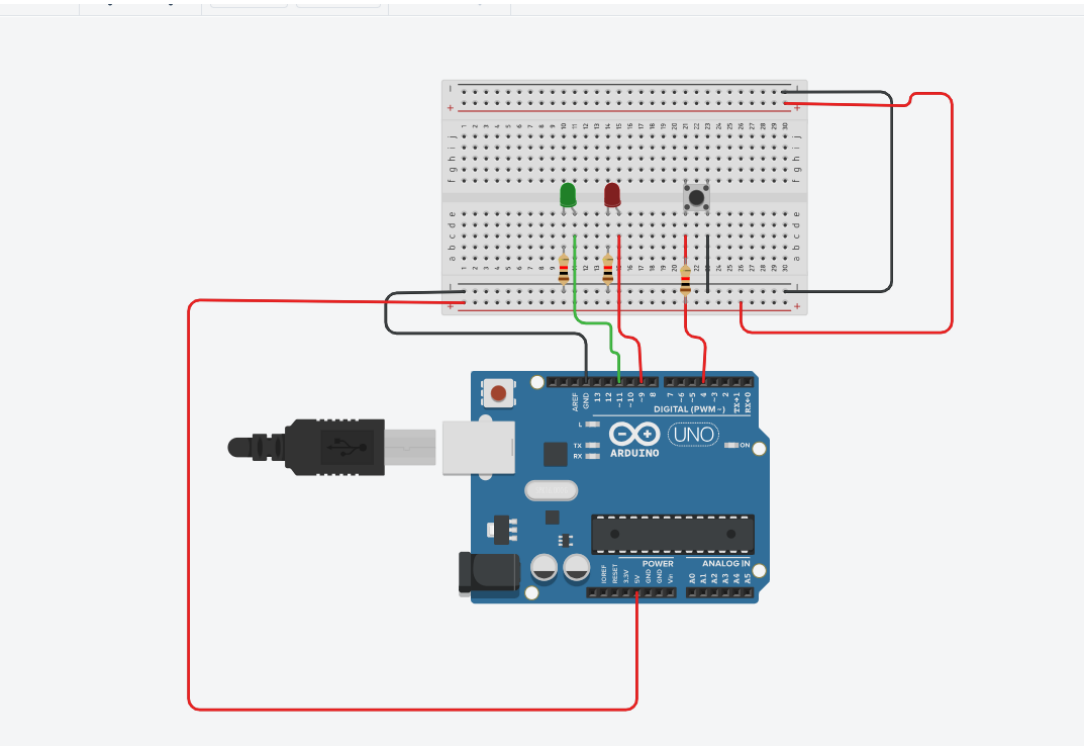

Figure 1: Breadboard design showing two LEDs, a push button, and Arduino UNO wiring

Working Principle

1. The push button acts as a digital input to track presses.
2. A counter is used to determine how many times the button has been pressed.
3. Depending on the count, different LEDs are activated or deactivated.
4. After the third press, the cycle resets.

Arduino Code

```
int led1 = 11;
int led2 = 9;
int button = 4;

int pressCount = 0;
bool lastButtonState = HIGH;

void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(button, INPUT_PULLUP);
}

void loop() {
  bool currentButtonState = digitalRead(button);
```

```
  if (lastButtonState == HIGH && currentButtonState == LOW) {
    pressCount++;

    if (pressCount == 1) {
      digitalWrite(led1, HIGH);
      digitalWrite(led2, LOW);
    } else if (pressCount == 2) {
      digitalWrite(led1, LOW);
      digitalWrite(led2, HIGH);
    } else if (pressCount == 3) {
      digitalWrite(led1, LOW);
      digitalWrite(led2, LOW);
      pressCount = 0;
    }

    delay(200);
  }

  lastButtonState = currentButtonState;
}
```

Steps to Build the Project

1. Place the LEDs and button on the breadboard.
2. Connect the LEDs to Arduino digital pins 11 and 9 with resistors.
3. Connect the button to pin 4 and GND using a pull-down resistor.
4. Upload the code to Arduino and test.

Similar Project Ideas

- **Traffic Light Simulation:** Add a third (yellow) LED and simulate real traffic signal timing.
- **LED Pattern Game:** Implement game logic with LED sequence memory.
- **Reaction Timer:** Use the button to measure user's reaction time with LED prompts.
- **RGB Color Changer:** Use a single RGB LED and cycle through colors with button presses.

How to Make This Project More Advanced

- Add an LCD screen to display button press count or status messages.
- Use hardware or software debouncing for more reliability.

- Store button state in EEPROM to retain state after power loss.
- Replace digital control with analog input (e.g., a potentiometer).
- Add Bluetooth module to control LEDs from a mobile app.