LAUNCH THE PROJECT - 02

Button Controlled LED

On AYNOP® UNO Launchpad Kit

Author: AYNOP Enterprises | Doc Version: 1.0 | Date: 03-9-2025

Table of Contents

1. Overview

This project demonstrates how to control an LED using a **push button** on the Arduino® UNO R4 Minima.

You will learn to:

- Use a push button as a **digital input**.
- Control an LED as a **digital output.**
- Understand the use of a **pull-down resistor** for stable input readings.

2. Components Required

- Arduino® UNO R4 Minima board
- USB Type-C data cable
- 1 × LED (any colour)
- 1 × 220 Ω resistor (current limiting)
- 1 × Push button
- 1 × 10 kΩ resistor (pull-down resistor for button)
- Breadboard
- Jumper wires

3. Software Required

- Arduino IDE (v2.3.6 or later recommended)
- No additional libraries required.

Note:
We assume the **Arduino UNO R4 Minima board package** is already installed on your machine, as explained in the 00_Getting_Started/00_GettingStarted_Arduino_R4_Minima guide. If it is not installed, please refer to that document and complete the installation before proceeding.

4. Hardware Setup

This section explains how to connect the components for the **Button Controlled LED** project.
It includes a **Wiring Diagram** and a **Circuit Schematic**.

4.1 Wiring Diagram

- The LED **anode (long leg)** is connected to digital pin **D13** through a **220 Ω resistor**, and the **cathode (short leg)** is connected to **GND**.
- The **push button** has 4 legs:
  - Leg 1 → Arduino **D2**
  - Leg 2 → **10 kΩ pull-down resistor → GND**
  - Leg 3 → **5V**
  - Leg 4 → **Not connected**

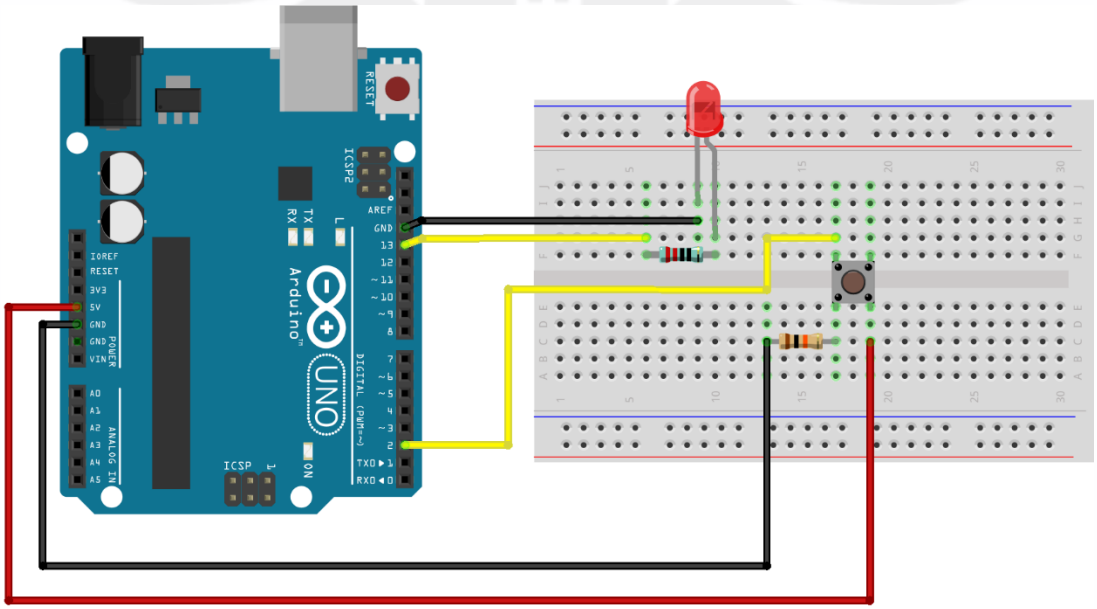

Figure 4.1 – Wiring diagram for Button Controlled LED project

Tip: Always disconnect the USB cable before making or changing hardware connections. This project is safe, but it's a good habit for advanced projects.

4.2 Circuit Schematic

- **R1 = 220 Ω** (current limiting resistor)
- **L1 = LED** (long leg = anode, short leg = cathode)
- **SW1 = Push button**
  - Pin 1 → Arduino D2
  - Pin 2 → 10 kΩ pull-down → GND
  - Pin 3 → 5V
  - Pin 4 → NC (not connected)
- **R2 = 10 kΩ** (pull-down resistor)
- Arduino **D13** → LED control
- Arduino **D2** → Button input

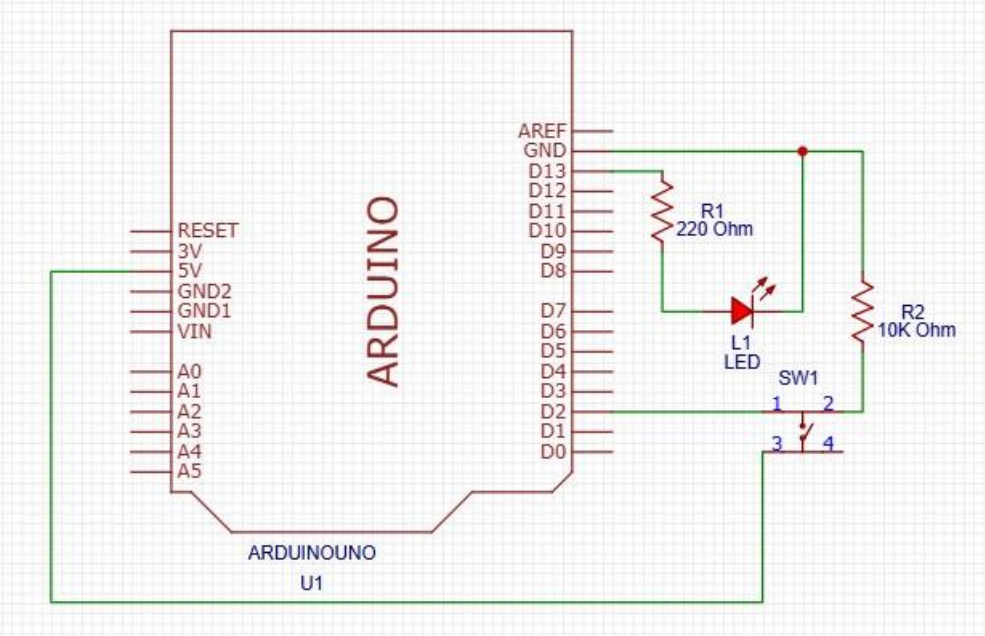

Figure 4.2 – Circuit schematic for Button Controlled LED project

5. Principle – How It Works

A push button is a simple **digital input device**.

- When the button is **not pressed**, the pull-down resistor keeps pin D2 at LOW (0V).
- When the button is **pressed**, pin D2 is connected to 5V, and the Arduino reads it as HIGH**.**
- The Arduino checks this input and controls the LED:
  - HIGH → LED ON
  - LOW → LED OFF

6. Procedure – Steps to Run

1. **Build the Circuit**
   a. Assemble the circuit as shown in the **Wiring Diagram (Figure 4.1)**.

2. **Connect the Board**
   a. Use a USB Type-C data cable to connect your UNO R4 Minima to your computer.

3. **Open the Project Code**
   a. Simply **double-click** the file *02_Button_Controlled_LED.ino* in the project folder, and it will open directly in the Arduino IDE (if installed).

4. **Confirm Board Selection**
   a. The IDE usually auto-detects the UNO R4 Minima if the package is installed.
   b. If not installed, refer to the *00_Getting_Started/00_GettingStarted_Arduino_R4_Minima* document to install the necessary board package.
   c. Verify that *Arduino UNO R4 Minima* is selected in the IDE's board selector (top toolbar).

5. **Upload the Code**
   a. Click the **Upload** button (arrow icon) in the top-left corner of the IDE.
   b. Wait until the console displays "**Done uploading**.".

6. **Observe the Behaviour**
   a. Press the button → the LED turns ON.
   b. Release the button → the LED turns OFF.

7. Expected Output

Once the code is uploaded successfully:

- When the **button is pressed**, the LED lights up.
- When the **button is released**, the LED turns off.
- The behaviour repeats reliably for each press and release.

8. Code

The source code for this project is included in the downloaded folder:

📂 *uno-launchpad-kit/01_Basic_Projects/02_Button_Controlled_LED/02_Button_Controlled_LED.ino*

👉 **Tip:**

- Simply **double-click** this .ino file, and it will automatically open in the Arduino IDE (if the Arduino IDE is installed on your computer).
- If the Arduino UNO R4 Minima board is **not yet installed** in your IDE, refer to the guide:
  📂 *uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima*

8.1 Function References

- setup() – runs once when the board is powered on or reset.
- loop() – runs continuously after setup() finishes.
- pinMode(pin, mode) – configures a pin as either **INPUT** or **OUTPUT**.
- digitalWrite(pin, value) – writes a digital value (**HIGH** or **LOW**) to a pin.
- digitalRead(pin) – reads the state of a digital pin (HIGH/LOW)

📚 **Arduino Reference:**

- setup()
- loop()
- pinMode()
- digitalWrite()
- digitalRead()

📚 **For more details and advanced usage, visit:**

🔗 Arduino Language Reference — The official guide for all Arduino functions.

9. Troubleshooting Tips

- **LED not responding?**
  - Check polarity: the **anode (long leg)** must connect to pin D13 (via resistor), and the **cathode (short leg)** must go to GND.
  - Ensure the resistor is in series with the LED, not parallel.

- **Button press not detected?**
  - Verify push button wiring from **Wiring Diagram (Figure 4.1)**.
  - Check that the pull-down resistor is firmly connected between D2 and GND.

- **Upload error in Arduino IDE?**
  - Verify that the correct board (**Arduino UNO R4 Minima**) is selected in the IDE.
  - Check that the correct **COM port** is chosen.

- **Board not detected?**
  - Ensure you are using a **data-capable USB Type-C cable** (some cables only provide charging).
  - Try reconnecting the cable or using a different USB port.

- **Board not listed in Arduino IDE?**
  - If you don't see **Arduino UNO R4 Minima** in the board selector, the **board package is not installed**.
  - To fix this, follow the installation steps in:
    uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima

Tip: If nothing works, press the **RESET** button on the UNO R4 Minima and try uploading the code again.

10. License

11. Support & Feedback

We value your feedback and are happy to assist with any questions, troubleshooting, or suggestions you may have.

Email: *support@aynop.com*

When sending an email, please include your kit name (AYNOP® UNO Launchpad Kit) and, if applicable, the project name in the subject line. This will help our team respond faster and more accurately. We aim to respond to all queries within 2–3 business days. Your feedback helps us improve our products and create even better learning experiences for all Arduino beginners.