

Electrical Circuit Construction

Electronics Engineering

[Task 1]

Mohamad Abdulmoula, R. Rama

ramamohameed9@gmail.com

Department of Software Engineer

Under the Supervision Of:

Eng Asmaa Duramae

July 5, 2025

Contents

Glossary of Terms	3
Project Overview	4
1. Planning	4
2. Requirements Analysis.....	4
Hardware Components.....	4
Functional Requirements	5
3. Design	5
4. Coding.....	7
5. Testing.....	7
Conclusion	8
References.....	9

Glossary of Terms

Term	Definition
Arduino UNO R3	A microcontroller board based on the ATmega328P, used for building electronic projects.
LED	Light Emitting Diode; a semiconductor device that emits light when current passes through it.
Push Button	A momentary switch used to allow user input by completing a circuit when pressed.
Breadboard	A board used for constructing and testing electronic circuits without soldering.
Resistor	An electronic component that limits the flow of current in a circuit.
Jumper Wires	Wires used to make electrical connections between components on a breadboard or Arduino.
INPUT_PULLUP	Arduino pin mode that uses an internal resistor to keep the input HIGH until pulled LOW.
Tinkercad	A web-based simulation tool used for 3D modeling and electronics prototyping.

Project Overview

In this project, we developed a simple interactive electronic circuit using an Arduino UNO R3, three LEDs (Yellow, Red, Blue), and three push buttons, where each button controls its corresponding LED. The system was designed and simulated in Tinkercad to ensure that each LED lights up only while its respective button is pressed, and turns off when released. The development process followed standard phases including Planning, Requirements Analysis, Design, Coding, and Testing.

1. Planning

We planned to build a multi-input interactive circuit using three push buttons, each mapped to control a separate LED (Yellow, Red, or Blue), with Arduino UNO R3 serving as the central controller. Before implementation, we simulated the entire system using Tinkercad to ensure accurate functionality and minimize wiring errors.

2. Requirements Analysis

Hardware Components

We used the following components:

- a) Arduino UNO R3
- b) 1 × Yellow LED
- c) 1 × Red LED
- d) 1 × Blue LED
- e) 3 × k Ω Resistors
- f) 3 × Push Buttons
- g) Small Breadboard
- h) Jumper Wires

Functional Requirements

- a) Each LED should turn ON only while its corresponding button is being pressed.
- b) Each LED should turn OFF immediately when its button is released.
- c) All operations must occur in real-time and independently.

3. Design

Circuit Diagram Description

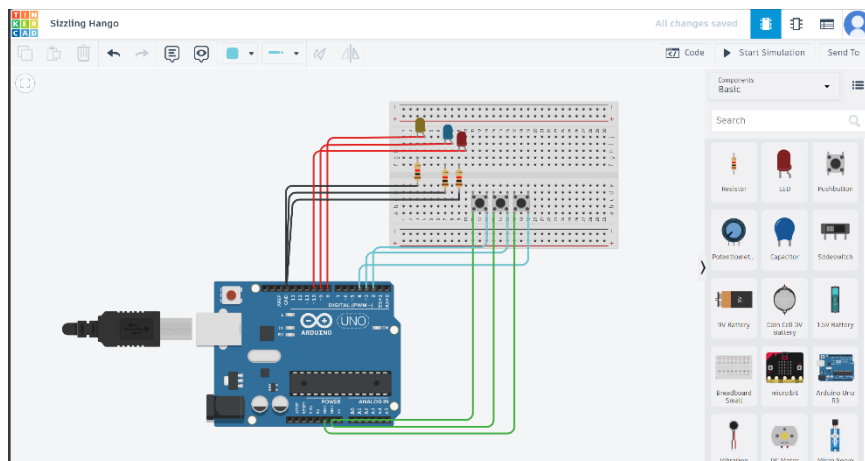


Figure 1 All LEDs are OFF – no buttons are currently pressed

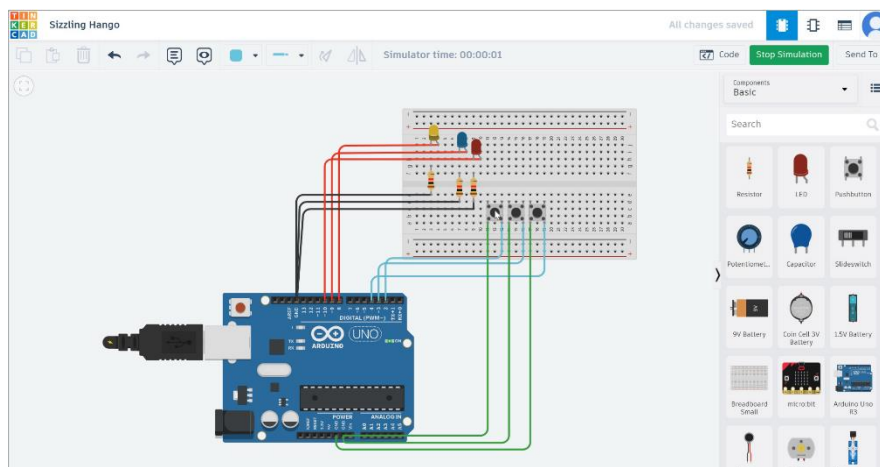


Figure 2 Yellow LED is ON – Yellow button is being pressed

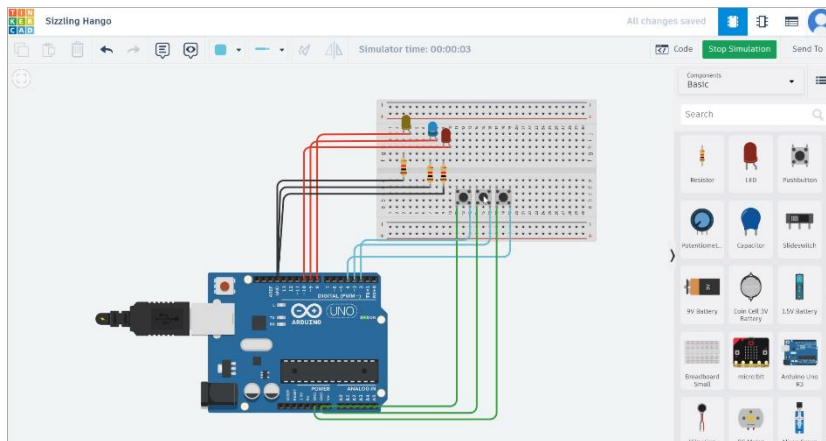


Figure 3 Blue LED is ON – Blue button is being pressed

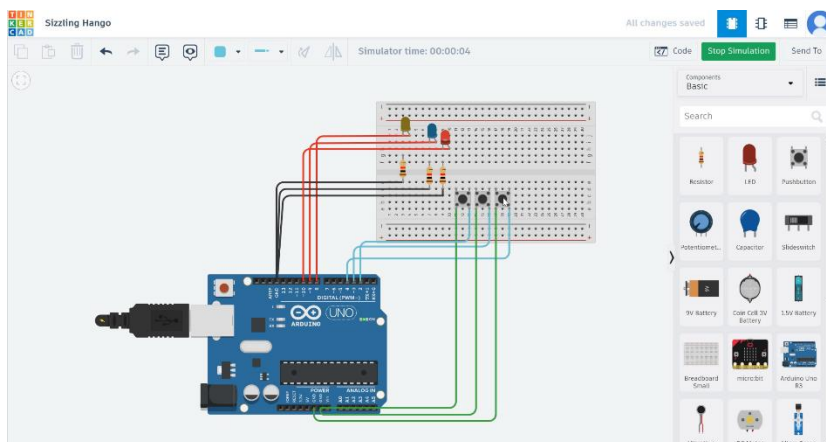


Figure 4 Red LED is ON – Red button is being pressed

We designed the circuit as follows:

Each LED anode (long leg) is connected to a specific digital output pin on the Arduino (e.g., 10 for Green, 11 for Red, 12 for Blue). The cathodes are connected to GND through a $1k\Omega$ resistor. Each push button is connected to a separate digital input pin (e.g., 2 for Green, 3 for Red, 4 for Blue), and configured using Arduino's INPUT_PULLUP feature. When pressed, the input reads LOW, activating the corresponding LED.

Figure 1 All LEDs are OFF – no buttons are currently pressed

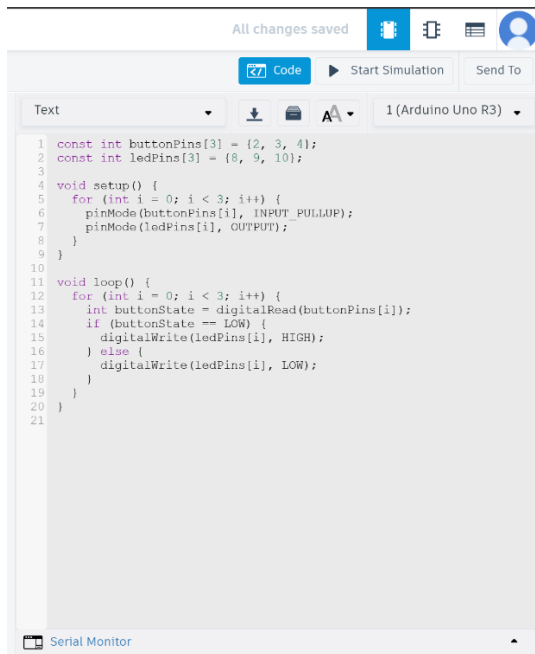
Figure 2 Yellow LED is ON – Yellow button is being pressed

Figure 3 Blue LED is ON – Blue button is being pressed

Figure 4 Red LED is ON – Red button is being pressed

4. Coding

Here is the code we uploaded to the Arduino:



```
1 const int buttonPins[] = {2, 3, 4};
2 const int ledPins[] = {8, 9, 10};
3
4 void setup() {
5   for (int i = 0; i < 3; i++) {
6     pinMode(buttonPins[i], INPUT_PULLUP);
7     pinMode(ledPins[i], OUTPUT);
8   }
9 }
10
11 void loop() {
12   for (int i = 0; i < 3; i++) {
13     int buttonState = digitalRead(buttonPins[i]);
14     if (buttonState == LOW) {
15       digitalWrite(ledPins[i], HIGH);
16     } else {
17       digitalWrite(ledPins[i], LOW);
18     }
19   }
20 }
21
```

In this code, we control three LEDs using three push buttons connected to an Arduino. The buttons are connected to pins 2, 3, and 4, and the LEDs are connected to pins 8, 9, and 10. We use INPUT_PULLUP for each button, which means the pin normally reads HIGH and goes LOW when the button is pressed. In the loop(), the Arduino checks each button. If a button is pressed (reads LOW), the corresponding LED is turned ON by sending HIGH to its pin. If the button is not pressed, the LED is turned OFF by sending LOW to the same pin.

5. Testing

We ran simulation tests in Tinkercad and verified the following:

- a) Pressing each button turned only the matching LED ON.
- b) Releasing the button immediately turned that LED OFF.
- c) There was no interference between the buttons.
- d) The system responded instantly and reliably.

Note: A recorded test demonstration is provided below to validate the correct operation of the circuit as designed.



testing1.wmv

Conclusion

Through this project, we successfully:

- a) Built an interactive system to control 3 different LEDs using 3 separate buttons.
- b) Simulated the design on Tinkercad for validation.
- c) Ensured real-time response using Arduino and basic electronic components.
- d) Demonstrated core embedded systems concepts including digital I/O, pull-up configuration, and LED control.

References

1. Arduino Official Documentation – <https://www.arduino.cc>
2. Tinkercad by Autodesk – <https://www.tinkercad.com>
3. Arduino Reference: <https://www.arduino.cc/reference/en/>
4. Electronics Tutorials – <https://www.electronics-tutorials.ws>