Heavens' Light is Our Guide

Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

CSE 3106

Computer Interfacing & Embedded Systems

Controlling an LED using Pushbutton

Submitted by

Kaif Ahmed Khan

Roll: 2103163

Submitted to

Nasif Osman Khansur
Lecturer

May 13, 2025

Contents

List of Listings

List of Figures

1 Controlling an LED using Pushbutton

1.1 Introduction

Controlling an LED using a pushbutton is a fundamental experiment that demonstrates digital input and output interaction in microcontrollers. In this setup, an external pull-down resistor is used with the pushbutton to ensure a stable and defined logic level when the button is not pressed. When the button is open, the resistor keeps the input pin at a LOW (0V) state, and when the button is pressed, the pin reads HIGH (3.3V or 5V, depending on the microcontroller logic).

The STM32F103C6 microcontroller allows us to easily read the state of a pushbutton using digital input pins and control an LED output pin accordingly. In the Arduino framework, the `digitalRead` function is used to detect the button press, and the `digitalWrite` function is used to turn the LED ON or OFF based on the button's state. This experiment helps in understanding how external components like switches interface with microcontroller GPIOs and how pull-down resistors prevent floating inputs, ensuring reliable operation of the circuit.

1.2 Procedure

1. I have collected a microcontroller STM32 blue pill, STLink programmer, a breadboard, an LED, a 220Ω resistor, a pushbutton, a $10k\Omega$ resistor (for pull-down), and some jumper wires.
2. Connected the LED's anode pin to GPIO pin PB0 through the 220Ω resistor, and connected the LED's cathode to ground.
3. Connected one terminal of the pushbutton to 3.3V (V_{cc}), and the other terminal to GPIO pin PC15.
4. Connected a $10k\Omega$ resistor between PC15 and ground to serve as the pull-down resistor.
5. Written the code to turn the LED ON when the button is pressed and OFF when it is released (see code 1), and uploaded it using the STLink programmer.

1.3 Diagram

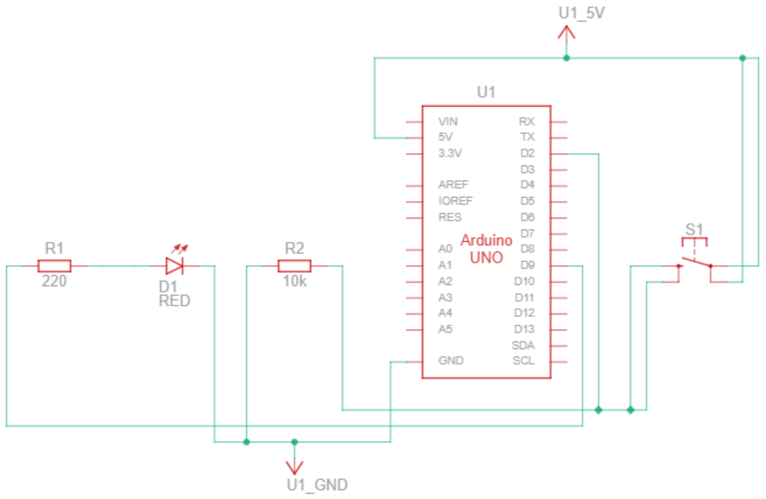

Figure 1: *Schematic diagram of the circuit using Arduino in TinkerCAD*

1.4 Source Code

Listing 1: *Controlling an LED using pushbutton*

```
#include <Arduino.h>
const int LED = PB0;
// The LED is connected to pin 9
const int BUTTON = PC15;
// The Button is connected to pin 2
void setup(){
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}
void loop(){
  if (digitalRead(BUTTON) == HIGH){
    digitalWrite(LED, LOW);
  } else {
    digitalWrite(LED, HIGH);
  }
}
```

2 Discussion

Through this project, I have learned how to interface a pushbutton with the STM32F103C6 microcontroller using an external pull-down resistor. I understood the purpose of the pull-down resistor in preventing floating inputs and ensuring reliable detection of button presses. By reading the pushbutton state and controlling the LED accordingly, I gained practical experience with digital input and output in embedded systems.