Heavens' Light is Our Guide

Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

CSE 3106
Computer Interfacing & Embedded Systems

Dimming an LED using PWM

Submitted by

Kaif Ahmed Khan

Roll: 2103163

Submitted to

Nasif Osman Khansur
Lecturer

May 13, 2025

Contents

List of Listings

List of Figures

1 Dimming an LED using PWM

1.1 Introduction

Pulse Width Modulation (PWM) is a technique that allows analog like control over digital output by controlling the duty cycle of a digital signal. The level of power depends on the duty cycle, which is the percentage of time the signal stays on during each cycle. PWM is commonly used for dimming LEDs and controlling motors.

In STM32F103C6 microcontroller there are several GPIO pins that can work as PWM pins. For example the PB0 pin can work as a PWM pin. Using the Arduino framework the `analogWrite` function provides the PWM feature. The function takes the pin number and voltage level as the input and sends the output voltage according to it.

1.2 Procedure

1. I have collected a microcontroller STM32 blue pill, STLink programmer, a breadboard, an LED, a 220Ω resistor and some jumper wire.
2. Connected the LED's anode pin to GPIO pin PB0 through the resistor, and connected the cathode pin to the ground.
3. Written the code for dimming the LED (see code 1) and uploaded it using STLink programmer.

1.3 Diagram

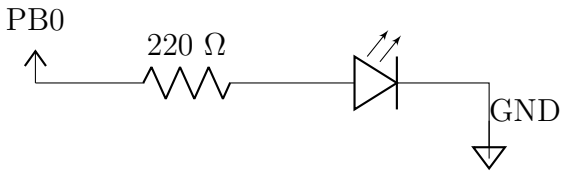

Figure 1: *Circuit diagarm for dimming an LED using PWM.*

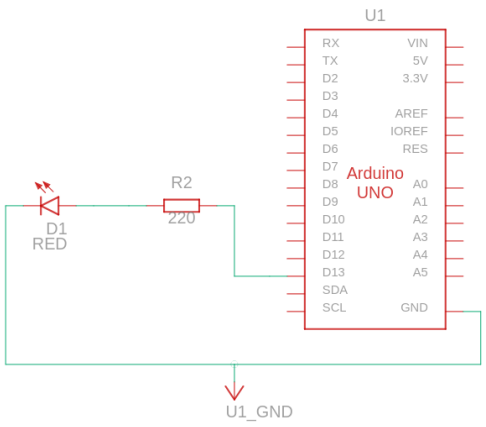

Figure 2: *Schematic diagram of the circuit using Arduino in TinkerCAD*

1.4 Source Code

Listing 1: *Dimming an LED using PWM in Arduino Framework*

```
#include <Arduino.h>
#define LED_PIN PB0

void setup() { pinMode(LED_PIN, OUTPUT); }

void loop() {
  for (int i = 0; i <= 255; i++) {
    analogWrite(LED_PIN, i);
    delay(10);
  }
  for (int i = 255; i >= 0; i--) {
    analogWrite(LED_PIN, i);
    delay(10);
  }
}
```

2 Discussion

Through this project, I have learned how Pulse Width Modulation (PWM) can be used to control the brightness of an LED using the STM32F103C6 microcontroller and the Arduino framework. By varying the duty cycle with the `analogWrite` function, I understood how digital signals could simulate analog output effectively. I also gained hands-on experience with setting up hardware connections, programming with PlatformIO, and working with microcontroller GPIO pins for PWM functionality.