

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САПР**

**ОТЧЕТ по лабораторной**  
**работе №4**  
**по дисциплине «Проектирование цифровых устройств»**  
**Тема: Цифровой ввод-вывод**

Студент гр. 3364

Преподаватель

\_\_\_\_\_ Юсфи А.

\_\_\_\_\_ Михайлов А.А

Санкт-Петербург

2025

## Цель работы.

Реализовать управление уровнем на светодиодной линейке с вводом по кнопкам с применением прерываний и программного антидребезга. Вывод попытки задать нереализуемые уровни (когда светодиоды не горят или все зажжены) на звуковую индикацию. Используется плата ПрЦУ-шилд.

## Ход работы

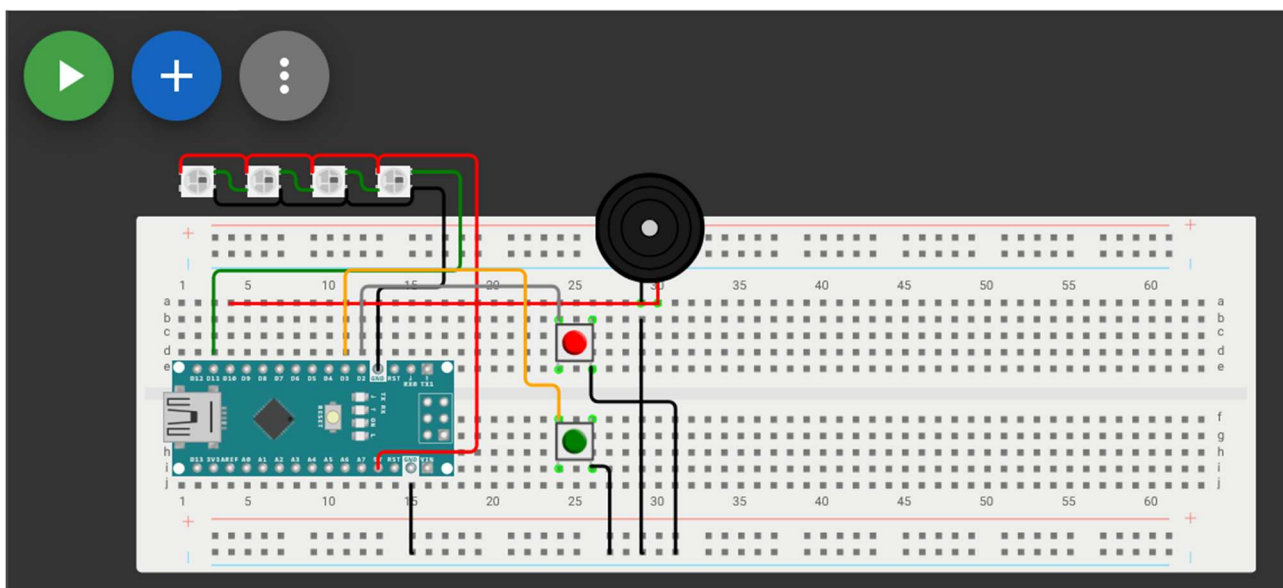


Рисунок 1 – Принципиальная схема устройства.

Рисунок 2 – Прототип на макетной плате.

## Принцип работы устройства

При включении устройства светодиодная лента остаётся в выключенном состоянии. Далее, при нажатии кнопок, в зависимости от кнопки, изменяется уровень светодиодной линейки, то есть загорается или гаснет один светодиод. При выходе за допустимый уровень звучит buzzer.

## Вывод

В результате выполнения работы мы научились работать со светодиодной лентой, а точнее научились управлять её светодиодами по-отдельности. Также мы научились выводить звуковой сигнал на buzzer.

## Листинг кода

```
#define Btn1 2
#define Btn2 3
#define Bip 10
#define PIN 11 // Pin для подключения ленты
#define NUMPIXELS 4 // Number светодиодов в ленте
#include "Adafruit_NeoPixel.h"

int Level = 0;
unsigned long Time1 = 0;
unsigned long Time2 = 0;
unsigned long BipTime = 0;
Adafruit_NeoPixel Leds (NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
void setup() {
    Leds.begin(); // инициализация ленты
    Leds.setBrightness(255);
    Serial.begin(9600);
    pinMode(Btn1, INPUT_PULLUP);
    pinMode(Btn2, INPUT_PULLUP);
    attachInterrupt(0, Btn_P, CHANGE);
    attachInterrupt(1, Btn_M, CHANGE);
}
void Btn_P(){
    if (millis() - Time1 > 200 && digitalRead(Btn1)){
        Time1 = millis();
        Level++;
        Leds.setPixelColor(Level-1, Leds.Color(255, 0, 0));
        Leds.show();
    }
}
void Btn_M(){
    if (millis() - Time2 > 200 && digitalRead(Btn2)){
        Time2 = millis();
        Level--;
        Leds.setPixelColor(Level, Leds.Color(0, 0, 0));
        Leds.show();
    }
}
```

```

    }
}

void NotBip(){
    int BipCount = 0;
    while (BipCount <= 6){
        if (millis()- BipTime > 100){
            BipTime = millis();
            BipCount++;
        }
        if (BipCount % 2 == 0) tone(Bip,1000);
        if (BipCount % 2 == 1) noTone(Bip);
    }
}

void loop() {
    if(Level < 0){
        NotBip();
        Level = 0;
    }
    if(Level > NUMPIXELS){
        NotBip();
        Level = NUMPIXELS;
    }
    Serial.println(Level);
}

```