

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Дата сдачи на проверку:

«__» _____ 2025 г.

Проверено:

«__» _____ 2025 г.

Отчёт по лабораторной работе №9
по дисциплине
«Информатика»
«Светодиодные индикаторы»

Разработал студент гр. ИВТб-1301-05-00

_____/Черкасов А. А./
(подпись)

Проверил преподаватель

_____/Шмакова Н.А./
(подпись)

Работа защищена

«__» _____ 2025 г.

Киров
2025

Цель

Закрепить на практике знания о светодиодных индикаторах на примере работы с arduino,

Задания

1. Бегущий огонек. На основе базовой схемы сделать движение светодиодов. Начальное состояние: вкл. Движение: четные/нечетные с 10.
2. Пульсар. На основе базовой схемы сделать движение светодиодов. Начальное состояние: вкл. Транзистор: биполярный. мена состои: увеличение до 1/3, до 2/3 и тах через сброс.
3. Ночной светильник. Начальное состояние: выкл. Входной сигнал: фоторезистор. Делитель напряжения: 220 Ом. Смена состояний: увеличение в зависимости от значений на датчике.
4. Кнопочный переключатель.
Л – левая кнопка;
П – правая кнопка;
ЛП – последовательное нажатие левой, а потом правой кнопки;
ПЛ – последовательное нажатие правой, а потом левой кнопки;
1/2 - количество кнопок, которые нужны для включения/выключения.
Включение: 2. Направление: ПЛ или ЛП. Выключение: 2. Направление: ПЛ.
5. RGB светодиод. Фиксация. Срабатывание по отпусканию. Смешивание цветов. Светодиод с общим катодом.

Решение

Задание 1

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 1.1.

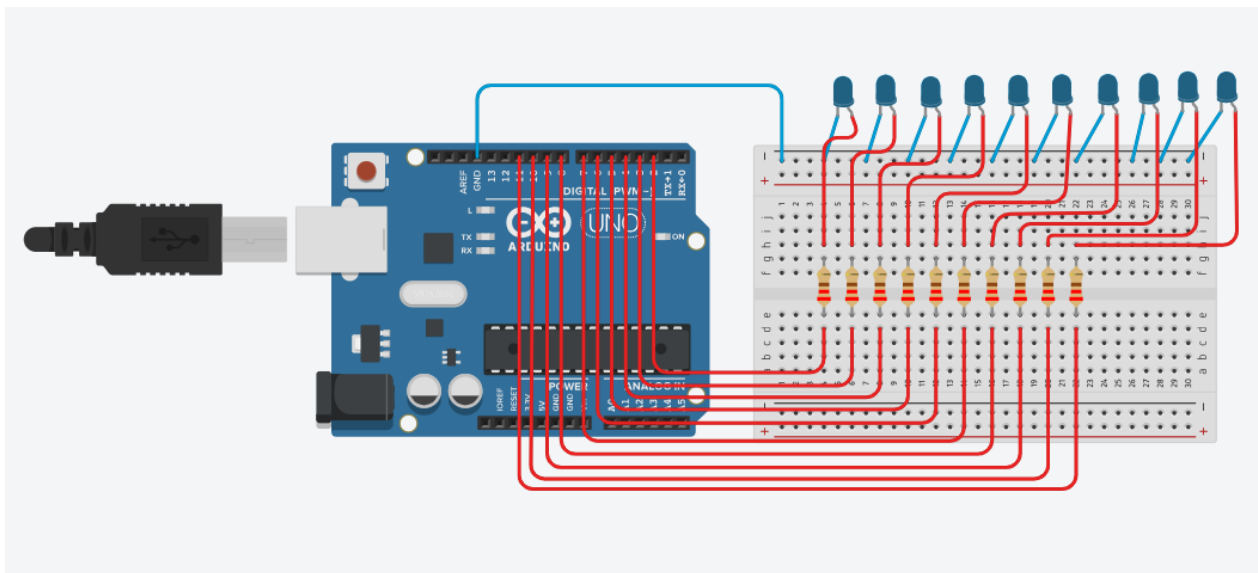


Рисунок 1.1 — Схема сборки задания №1

Принципиальная схема

Принципиальная схема представлена на рисунке 1.2.

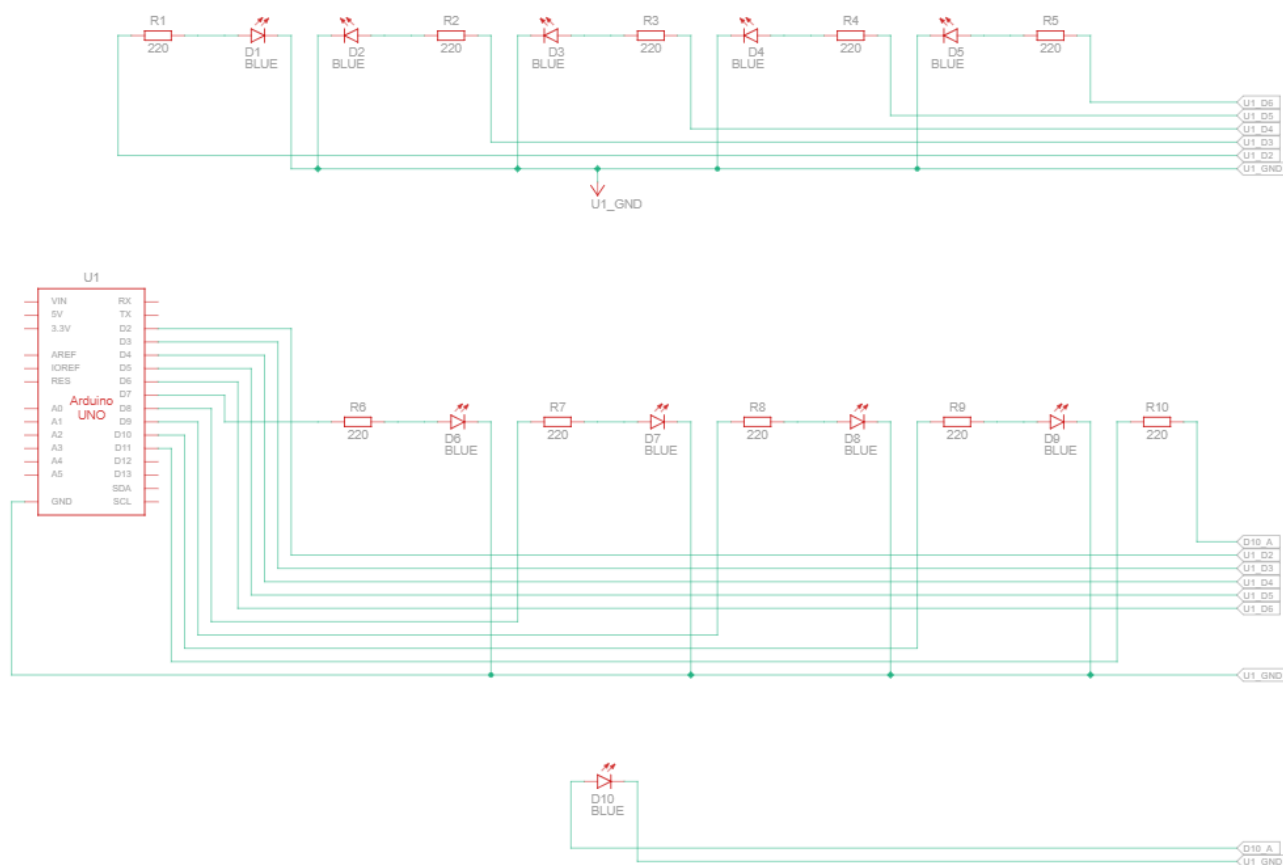


Рисунок 1.2 — Принципиальная схема задания №1

[Перейти на Tinkercad](#)

Исходный код решения задания представлен в приложении A1.

Задание 2

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 2.1.

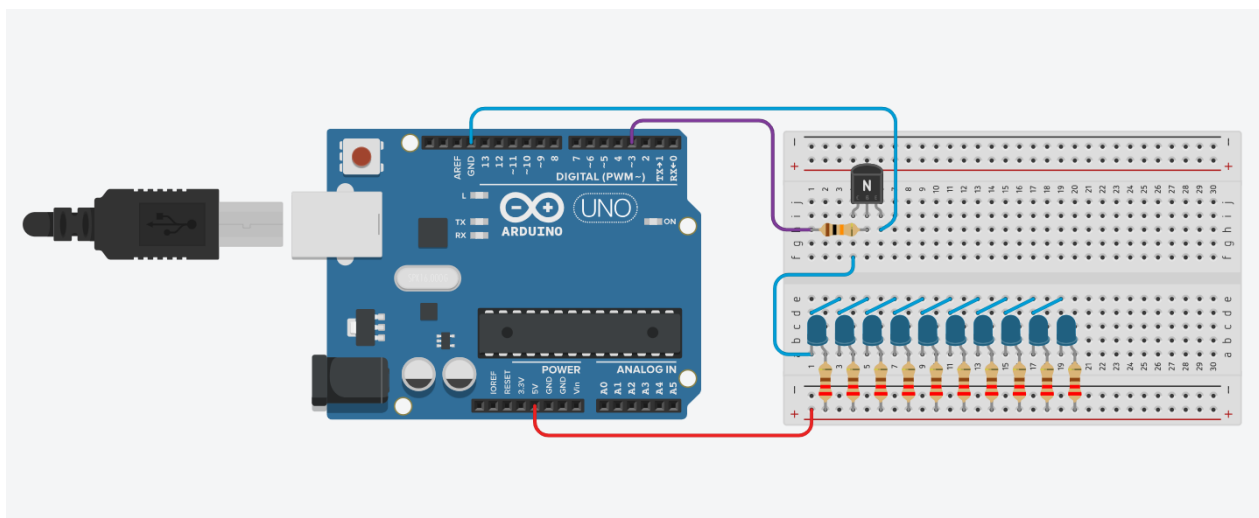


Рисунок 2.1 — Схема сборки задания №2

Принципиальная схема

Принципиальная схема представлена на рисунке 2.2.

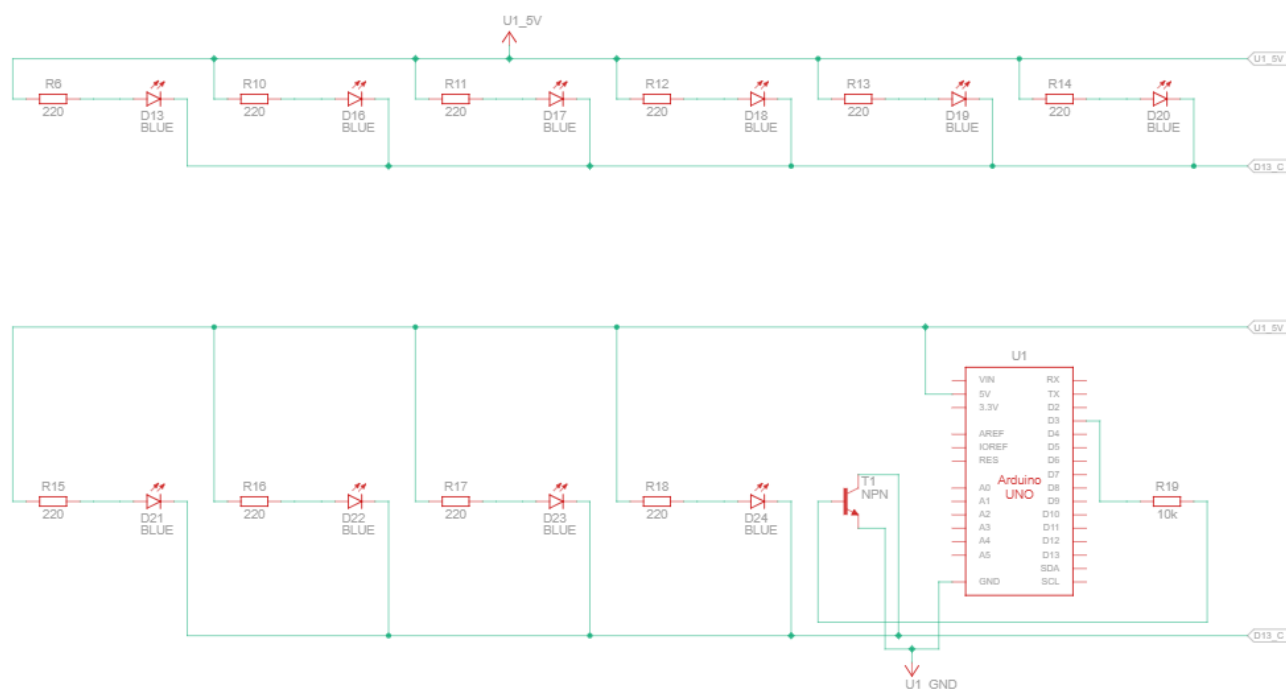


Рисунок 2.2 — Принципиальная схема задания №2

[Перейти на Tinkercad](#)

Исходный код решения задания представлен в приложении А2.

Решение

Задание 3

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 3.1.

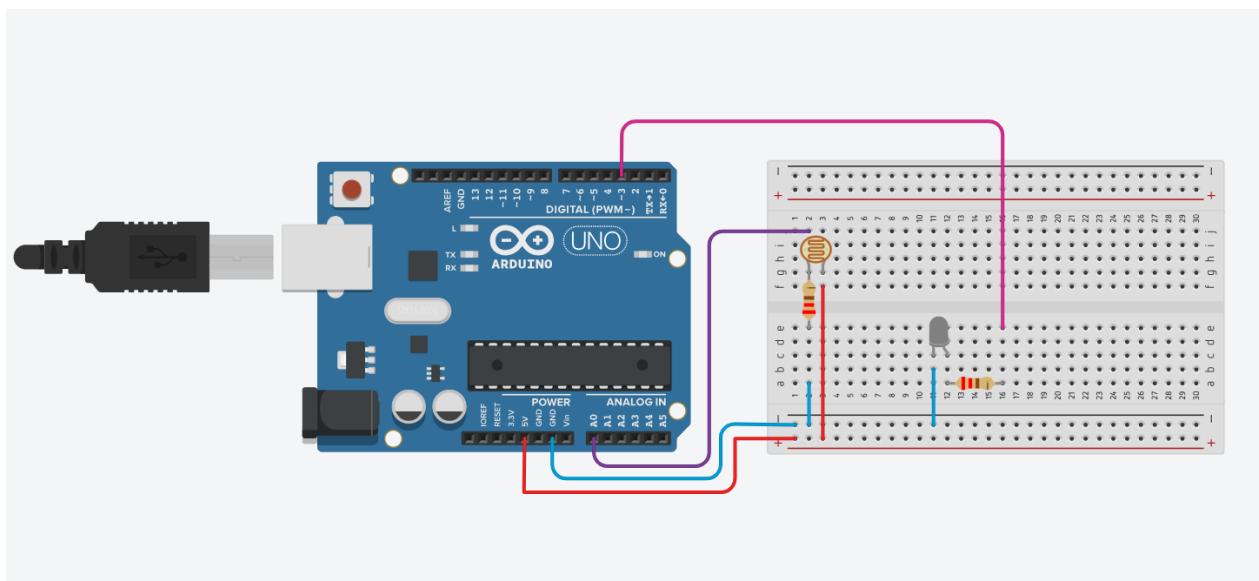


Рисунок 3.1 — Схема сборки задания №3

Принципиальная схема

Принципиальная схема представлена на рисунке 3.2.

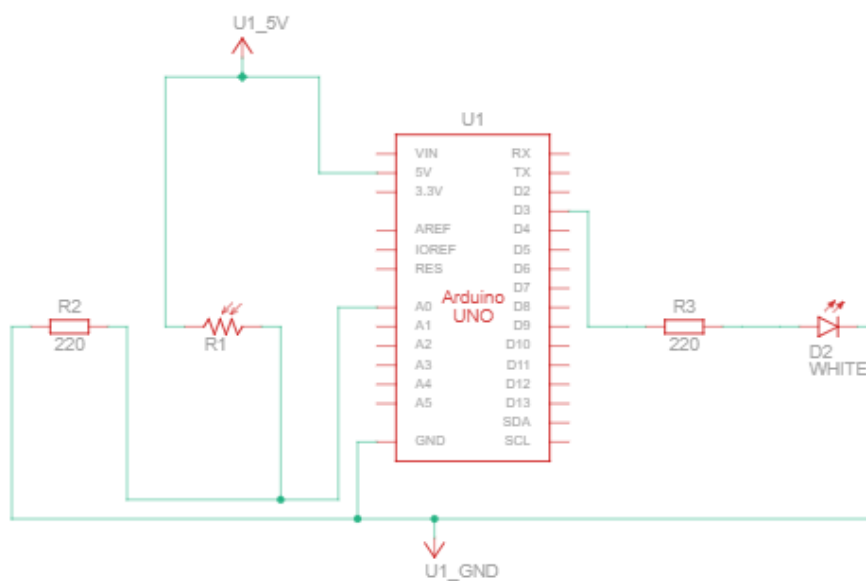


Рисунок 3.2 — Принципиальная схема задания №3

[Перейти на Tinkercad](#)

Исходный код решения задания представлен в приложении А3.

Решение

Задание 4

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 4.1.

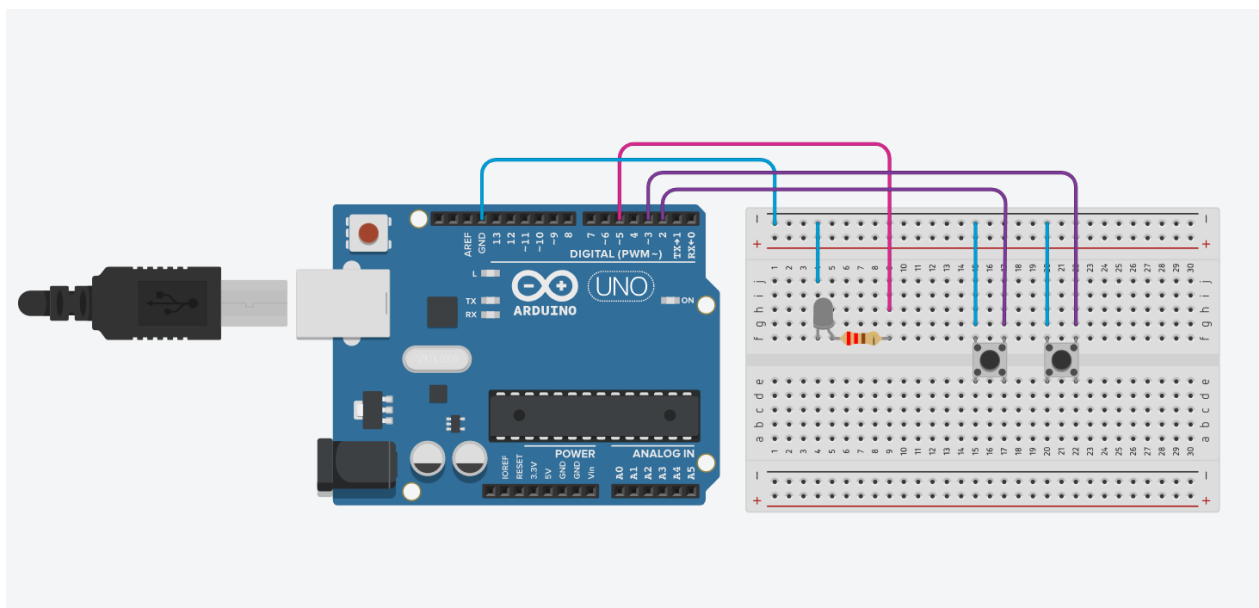


Рисунок 4.1 — Схема сборки задания №4

Принципиальная схема

Принципиальная схема представлена на рисунке 4.2.

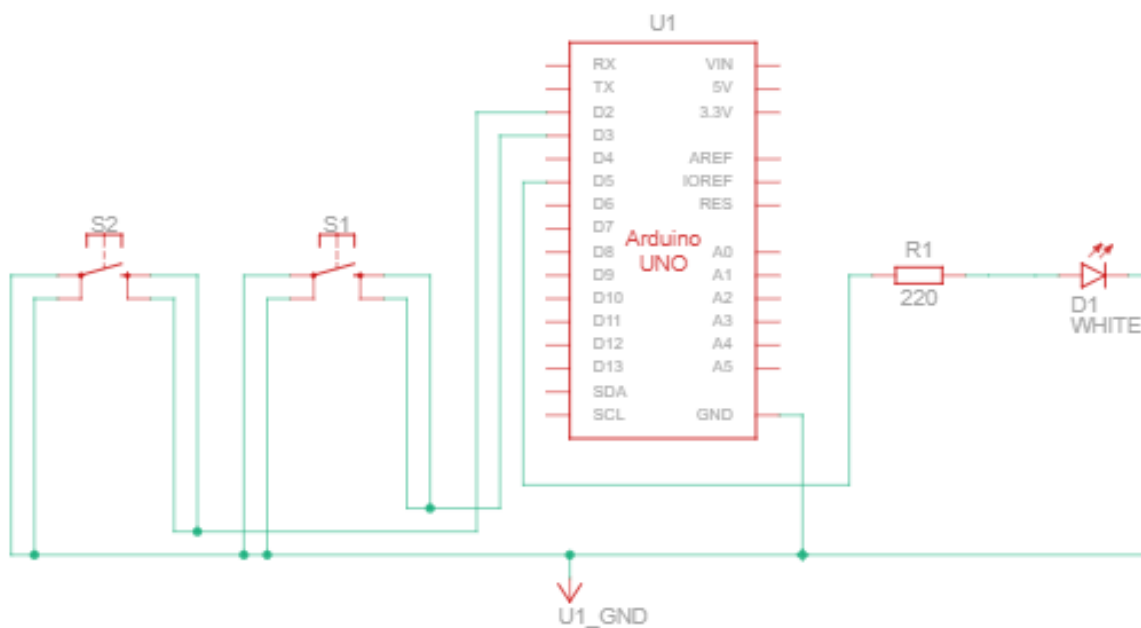


Рисунок 4.2 — Принципиальная схема задания №4

[Перейти на Tinkercad](#)

Исходный код решения задания представлен в приложении А4.

Решение

Задание 5

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 5.1.

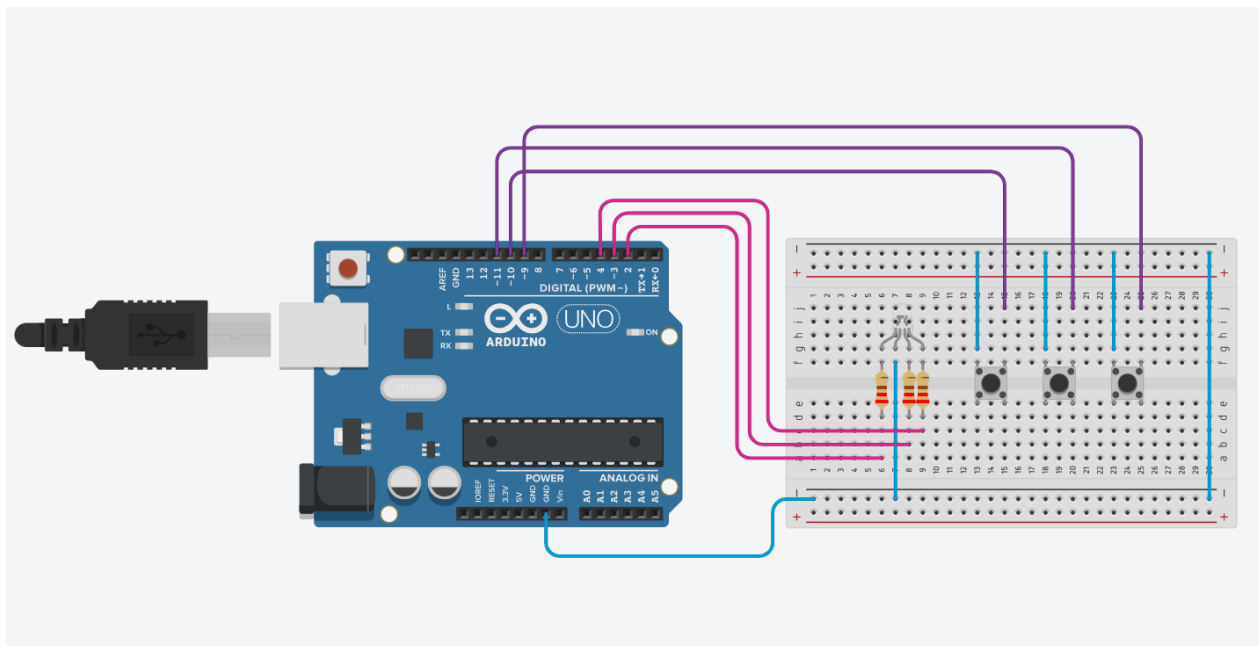


Рисунок 5.1 — Схема сборки задания №5

Принципиальная схема

Принципиальная схема представлена на рисунке 5.2.

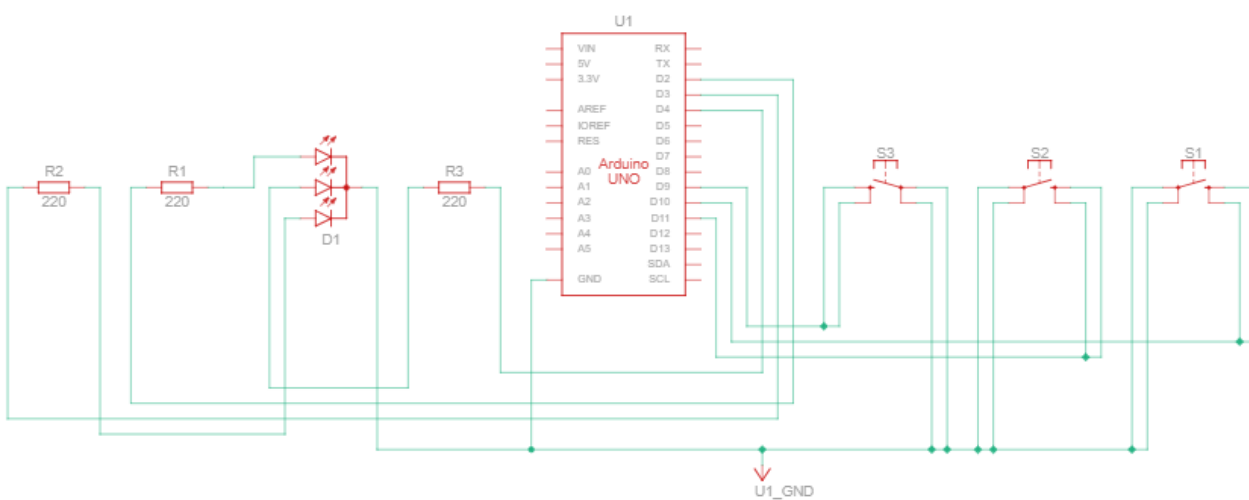


Рисунок 5.2 — Принципиальная схема задания №5

[Перейти на Tinkercad](#)

Исходный код решения задания представлен в приложении А5.

Решение

Дополнительное задание

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 6.1.

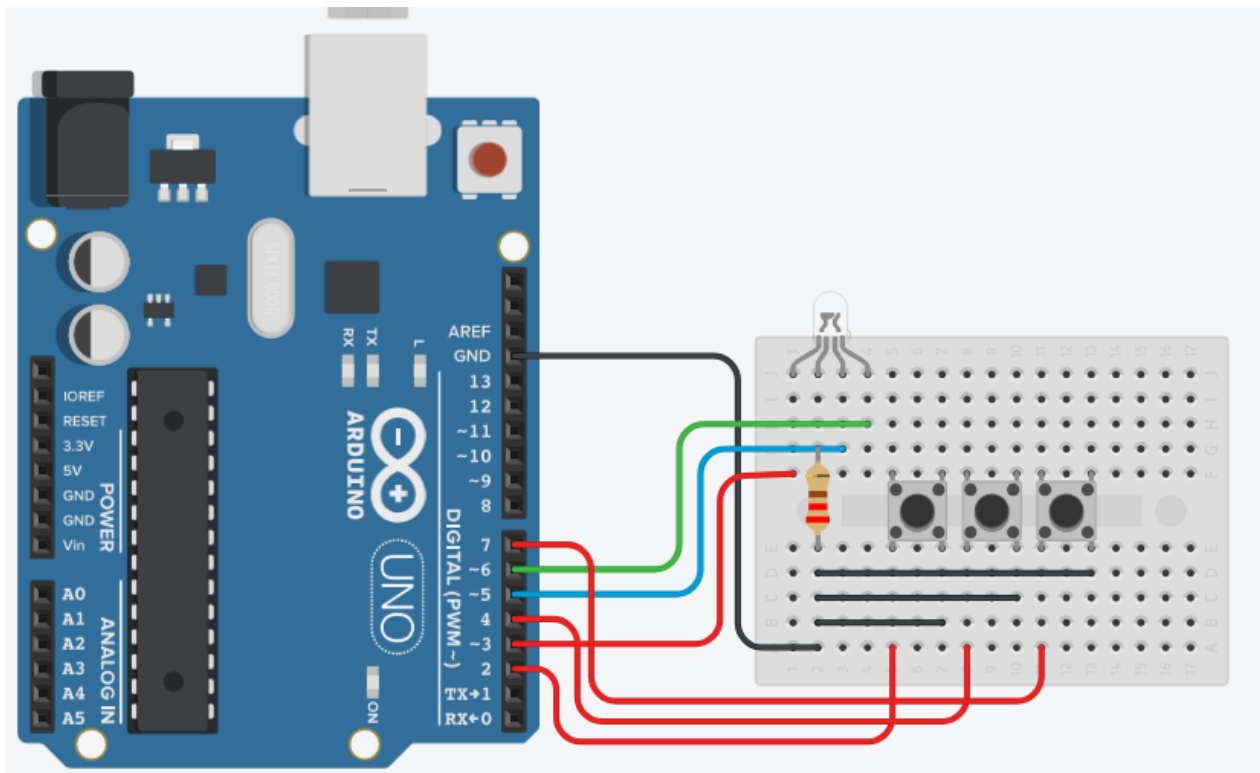


Рисунок 6.1 — Схема сборки задания №5

Принципиальная схема

Принципиальная схема представлена на рисунке 6.2.

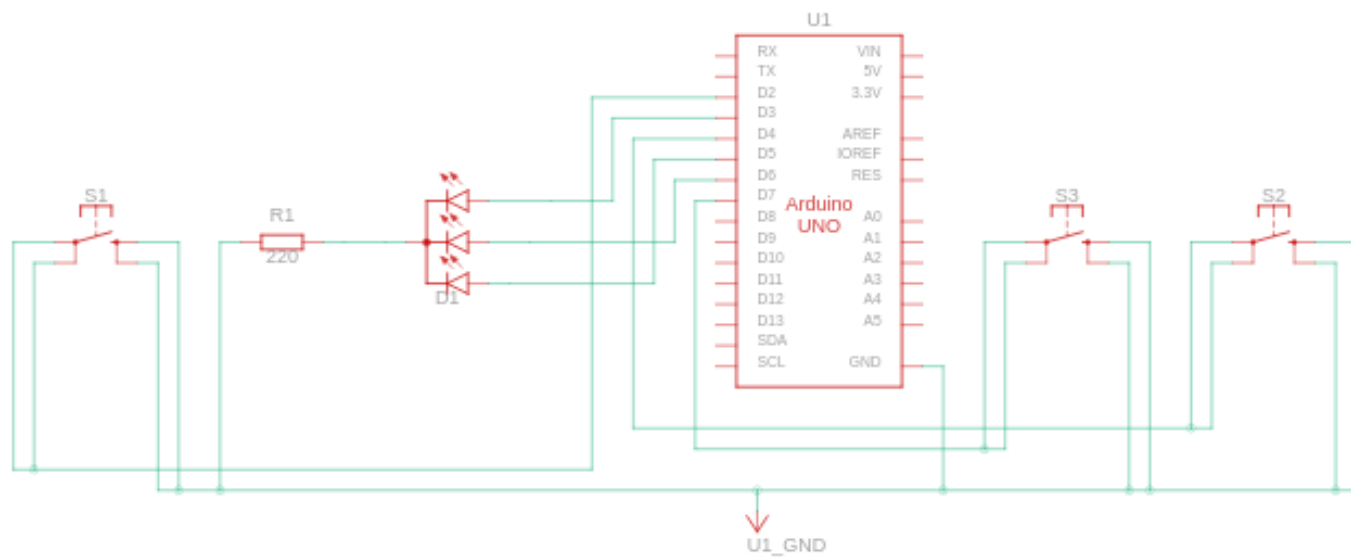


Рисунок 6.2 — Принципиальная схема задания №5

[Перейти на Tinkercad](#)

Исходный код решения задания представлен в приложении А6.

Вывод

В результате выполнения лабораторной работы была достигнута поставленная цель — закрепление на практике знаний о светодиодных индикаторах на примере работы с платформой Arduino. Были успешно реализованы пять различных заданий, демонстрирующих разнообразные способы управления светодиодами:

1. Бегущий огонек — последовательное включение светодиодов с чередованием четных и нечетных позиций;
2. Пульсар — плавное изменение яркости светодиода с использованием ШИМ-модуляции;
3. Ночной светильник — автоматическое управление яркостью в зависимости от освещенности с помощью фоторезистора;
4. Кнопочный переключатель — управление включением/выключением с использованием последовательности нажатий кнопок;
5. RGB-светодиод — управление цветом путем смешивания трех основных цветов.

В ходе работы были изучены основные принципы работы со светодиодными индикаторами, освоены методы цифрового и аналогового управления выводами Arduino, а также получены практические навыки программирования микроконтроллера. Были использованы следующие ключевые функции Arduino:

`pinMode(pin, mode)` – настройка режима работы пина (вход/выход)

`digitalWrite(pin, value)` – установка цифрового уровня сигнала (HIGH/LOW)

`analogWrite(pin, value)` – генерация ШИМ-сигнала для управления яркостью

`digitalRead(pin)` – чтение состояния цифрового входа

`analogRead(pin)` – чтение аналогового значения с датчика

`delay(ms)` – задержка выполнения программы

`Serial.begin(speed)` – инициализация последовательного порта для отладки

Приложение А1. Исходный код решения для задания №1

```
#include <Arduino.h>

#define FIRST_PIN 2
#define LAST_PIN 11

void setup() {
    for (int pin = FIRST_PIN; pin <= LAST_PIN; pin++) {
        pinMode(pin, OUTPUT);
        digitalWrite(pin, HIGH);
    }
}

void loop() {
    int pin;
    unsigned int ms = millis();
    ms = ms % 3000;
    if (ms < 1501) {
        pin = -1 + LAST_PIN - ((ms / 300) % 5) * 2;
        digitalWrite(pin, LOW);
        delay(100);
        digitalWrite(pin, HIGH);
    } else {
        pin = FIRST_PIN + ((ms / 300) % 5) * 2 + 1;
        digitalWrite(pin, LOW);
        delay(100);
        digitalWrite(pin, HIGH);
    }
}
```


Приложение А2. Исходный код решения для задания №2

```
#include <Arduino.h>

#define control_pin 3

int i = 0; //текущее состояние
int brightness = 255; //яркость
int levels[3] = {170, 85, 0}; //уровни яркости
void setup(){
    pinMode(control_pin, OUTPUT); //настройка пина на выход
    analogWrite(control_pin, brightness); //начальная яркость 255
    Serial.begin(9600); //монитор порта
}

void loop(){
    if (brightness > levels[i])
    {
        brightness--; //уменьшение яркости (шаг 1)
        analogWrite(control_pin, brightness); //применение значения
        Serial.println(brightness); //для графика
        delay(20); //задержка
    }
    else //переход к след. состоянию
    {
        i = (i + 1) % 3; //изменение состояния
        brightness = 255; //сброс яркости
        Serial.println(brightness); //для графика
        delay(500); //задержка
    }
}
```

Приложение А3. Исходный код решения для задания №3

```
#include <Arduino.h>

#define led_pin 3 //имя для пина со светодиодом
#define ldr_pin A0 //имя для пина с фоторезистором
int light;
void setup(){
    pinMode(led_pin, OUTPUT); //пин светодиода на выход
    Serial.begin(9600); //монитор порта
}
void loop(){
    light = analogRead(ldr_pin); //значения фоторезистора
    Serial.println(light); //вывод значений фоторезистора
    analogWrite(led_pin, map(light, 1, 310, 0, 255)); //масшт.
    delay(20); //задержка
}
```

Приложение А4. Исходный код решения для задания №4

```
#include <Arduino.h>

#define led_pin 5 //светодиод
#define left_button 2 //левая кнопка
#define right_button 3 //правая кнопка
bool led_state = false, right_first = false; //состояние св.
int right_flag = 0, left_flag = 0; //правый и левый флаги
void setup() {
    pinMode(led_pin, OUTPUT);
    pinMode(left_button, INPUT_PULLUP);
    pinMode(right_button, INPUT_PULLUP);
    Serial.begin(9600);
}
void loop() {
    bool left_state = digitalRead(left_button), right_state =
digitalRead(right_button); //получение значений
    Serial.println(String("Left pressed: " +
        left_flag + " | Right pressed: " + right_flag)); //вывод
    if (!led_state && right_flag && left_flag) { //вкл свет.
        digitalWrite(led_pin, HIGH); //включение светодиода
        led_state = true; //изменение статуса светодиода
        right_flag = left_flag = 0; //сброс флагов
    }
    right_first = (right_flag && !left_flag); //П нажата 1-ой?
    Serial.println(right_first); //вывод прав. кнопки
    left_flag = (left_state==0) ? (left_flag + 1)%2: left_flag;
    right_flag = (right_state == 0) ?
        (right_flag + 1)%2: right_flag;
    if (led_state && right_first && left_flag) {
        digitalWrite(led_pin, LOW);
        led_state = right_first = left_flag = right_flag = 0;
    }
    delay(100);
}
```

Приложение А5. Исходный код решения для задания №5

```
#include <Arduino.h>

boolean led[] = {false, false, false}; //состояния св.
byte ledik[] = {9, 10, 11}; //пины для кн.
byte color[] = {2,3,4}; //массив с пинами
byte l = 0;
void setup() {
    for (byte i = 9; i < 12; i++) pinMode(i, INPUT_PULLUP); //кн
    for (byte j = 2; j < 5; j++) pinMode(j, OUTPUT); //св
}
void loop() {
    buttonClick(ledik[l % 3], l % 3); //пин из массива ledik
    analogWrite(color[l], led[l] ? 255 : 0); //вкл выкл
    delay(50); //задержка
    l = (l + 1) % 3; //увеличение индекса
}
bool prevButtonState[] = {true, true, true}; //массив для пред.
void buttonClick(byte buttonPin, byte colorIndex) {
    bool currentButtonState = digitalRead(buttonPin);
    if (!prevButtonState[colorIndex] && currentButtonState) {
        led[colorIndex] = !led[colorIndex]; //переключение свет.
    }
    prevButtonState[colorIndex] = currentButtonState; //обновление
}
```

Приложение А6. Исходный код решения для дополнительного задания

```
#include <Arduino.h>

boolean led[] = {false, false, false}; //состояния св.
byte ledik[] = {9, 10, 11}; //пины для кн.
byte color[] = {2,3,4}; //массив с пинами
byte l = 0;
void setup() {
    for (byte i = 9; i < 12; i++) pinMode(i, INPUT_PULLUP); //кн
    for (byte j = 2; j < 5; j++) pinMode(j, OUTPUT); //св
}
void loop() {
    buttonClick(ledik[l % 3], l % 3); //пин из массива ledik
    analogWrite(color[l], led[l] ? 255 : 0); //вкл выкл
    delay(50); //задержка
    l = (l + 1) % 3; //увеличение индекса
}
bool prevButtonState[] = {true, true, true}; //массив для пред.
void buttonClick(byte buttonPin, byte colorIndex) {
    bool currentButtonState = digitalRead(buttonPin);
    if (!prevButtonState[colorIndex] && currentButtonState) {
        led[colorIndex] = !led[colorIndex]; //переключение свет.
    }
    prevButtonState[colorIndex] = currentButtonState; //обновление
}
```