Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа №4

**Управление светодиодами с помощью кнопки и потенциометра**

Выполнил: Паршиков И.Н.

студент группы ИВТ-41-22

Проверил: Степанов С. В.

Чебоксары, 2025

Описание задачи: Создайте проект на Arduino Nano, который будет управлять светодиодами в зависимости от состояния кнопки и значения потенциометра. В зависимости от выбранного варианта, можно реализовать разные режимы работы. Необходимые компоненты: - Arduino Nano - 3 светодиода (красный, желтый, зеленый) - 3 резистора 220 Ом (для светодиодов) - Кнопка - Потенциометр - Соединительные провода - Макетная плата

19 номер = 19% 5 = 4

4 вариант:

Вариант 4: Управление яркостью светодиодов

1. Подключите светодиоды и потенциометр, как в варианте 1.

2. Реализуйте программу, которая:    - Управляет яркостью каждого светодиода с помощью ШИМ (PWM) в зависимости от значения потенциометра.

- Красный светодиод горит с яркостью, пропорциональной значению потенциометра.

 - Зеленый светодиод горит с яркостью, обратной значению потенциометра.

 - Желтый светодиод мигает с частотой, зависящей от значения потенциометра.

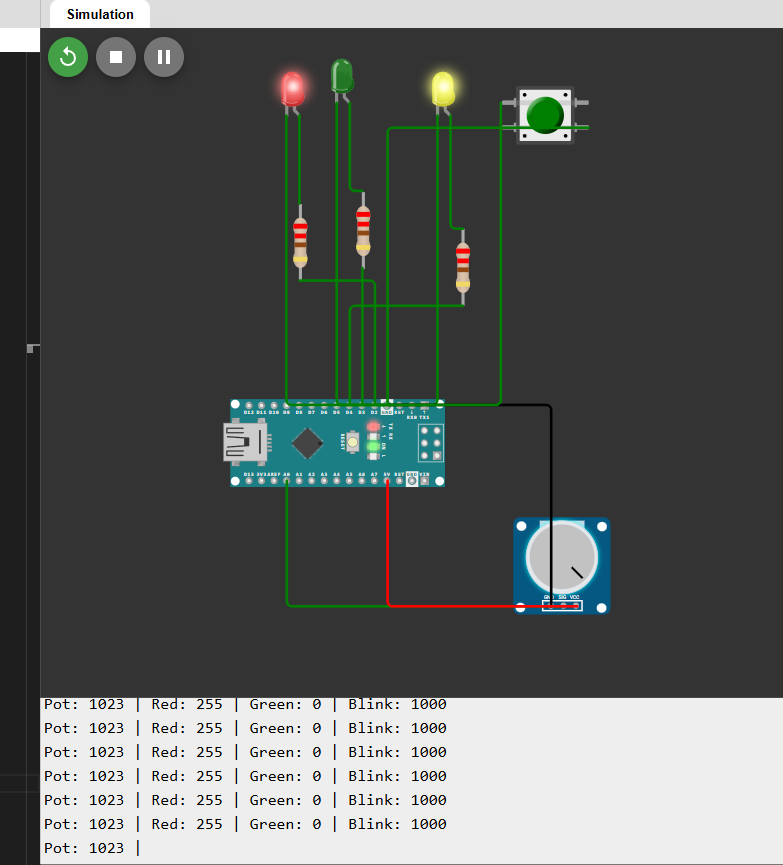
Схема из В1:

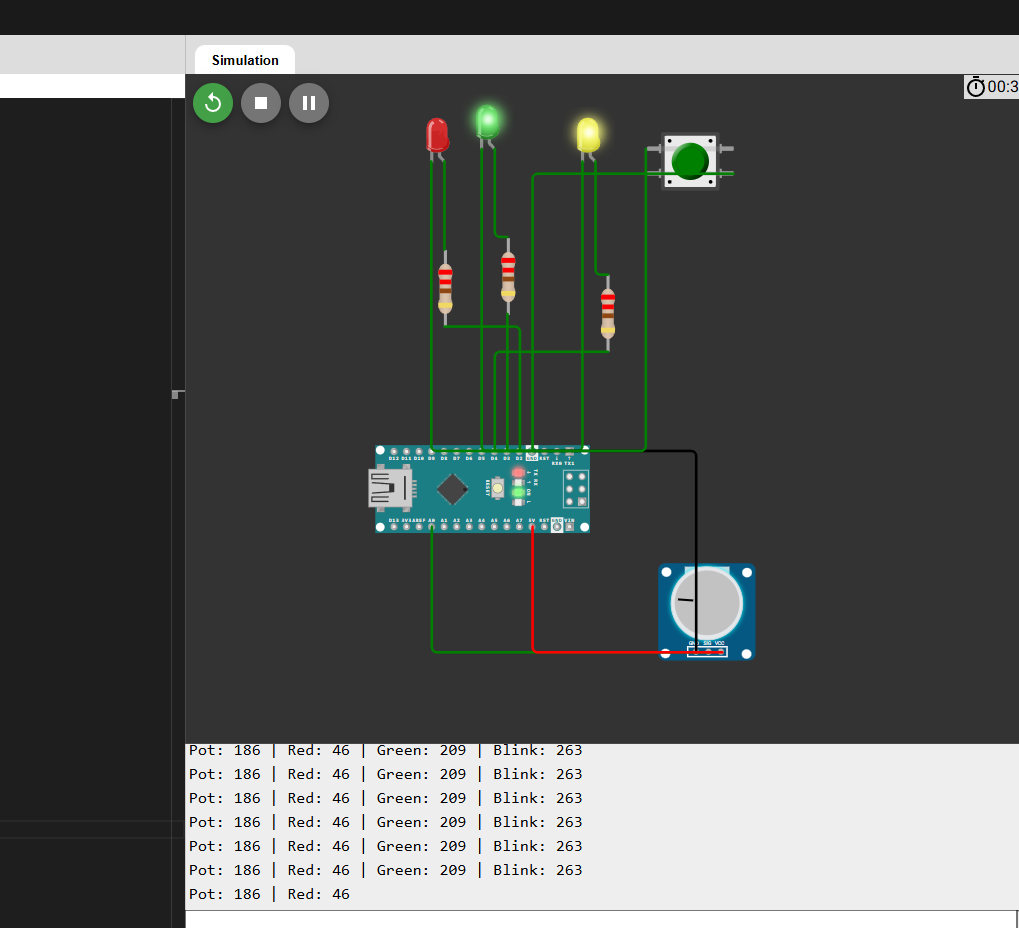
1. Подключите три светодиода к цифровым выходам Arduino (например, D2, D3, D4).

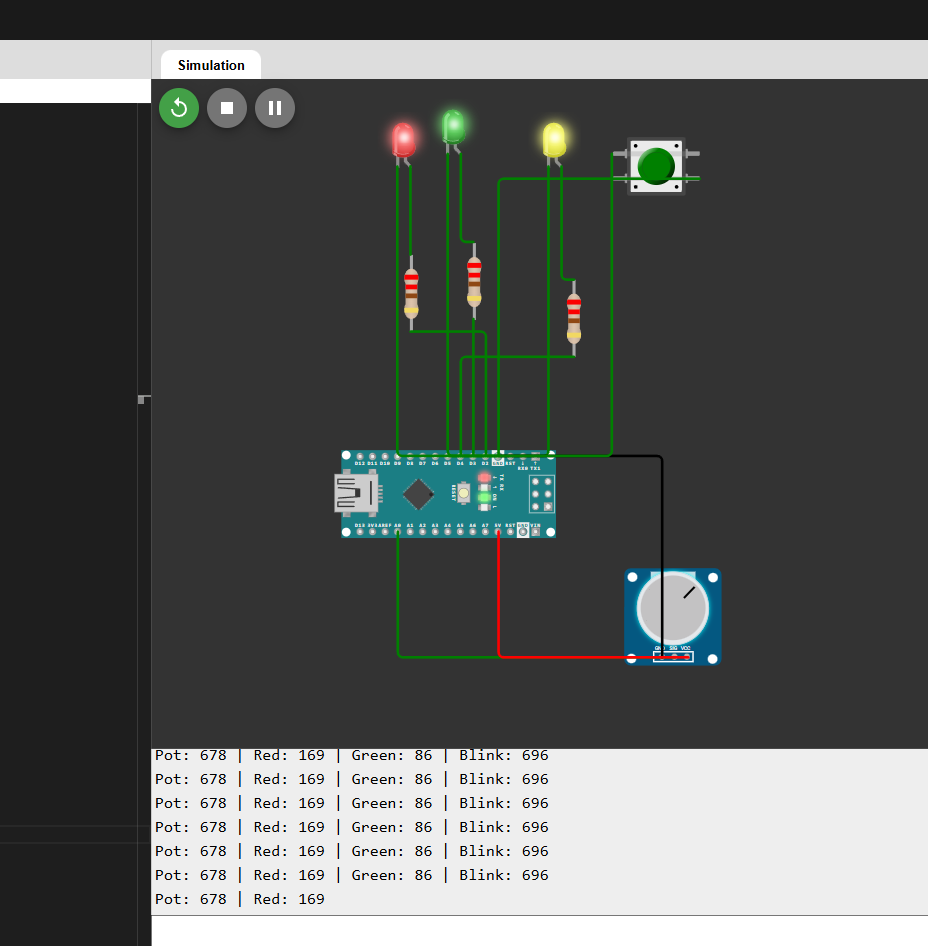
2. Подключите кнопку к цифровому входу (например, D5).

3. Подключите потенциометр к аналоговому входу (например, A0).

Горение светодиодов в зависимости от значения потенциометра:







Код программы:

// Определяем пины согласно вашей схеме

const int POT\_PIN = A0;      // Потенциометр подключен к A0

const int RED\_LED = 2;       // Красный светодиод на D2 (led1)

const int GREEN\_LED = 3;     // Зеленый светодиод на D3 (led2)

const int YELLOW\_LED = 4;    // Желтый светодиод на D4 (led3)

const int BUTTON\_PIN = 5;    // Кнопка на D5 (btn1)

// Переменные для мигания желтого светодиода

unsigned long previousMillis = 0;

bool yellowLedState = LOW;

void setup() {

  pinMode(RED\_LED, OUTPUT);

  pinMode(GREEN\_LED, OUTPUT);

  pinMode(YELLOW\_LED, OUTPUT);

  pinMode(BUTTON\_PIN, INPUT\_PULLUP);

  // Инициализация Serial для отладки

**Serial**.begin(9600);

}

void loop() {

  // Чтение значения потенциометра (0-1023)

  int potValue = analogRead(POT\_PIN);

  // Масштабируем значение для ШИМ (0-255)

  int pwmValue = map(potValue, 0, 1023, 0, 255);

  // Управление красным светодиодом (прямая пропорция)

  analogWrite(RED\_LED, pwmValue);

  // Управление зеленым светодиодом (обратная пропорция)

  analogWrite(GREEN\_LED, 255 - pwmValue);

  // Управление желтым светодиодом (мигание с изменяемой частотой)

  unsigned long currentMillis = millis();

  // Частота мигания зависит от положения потенциометра

  // Минимальный интервал 100ms, максимальный 1000ms

  int blinkInterval = map(potValue, 0, 1023, 100, 1000);

  if (currentMillis - previousMillis >= blinkInterval) {

    previousMillis = currentMillis;

    yellowLedState = !yellowLedState;

    digitalWrite(YELLOW\_LED, yellowLedState);

  }

  // Опционально: вывод значений в Serial Monitor для отладки

**Serial**.print("Pot: ");

**Serial**.print(potValue);

**Serial**.print(" | Red: ");

**Serial**.print(pwmValue);

**Serial**.print(" | Green: ");

**Serial**.print(255 - pwmValue);

**Serial**.print(" | Blink: ");

**Serial**.println(blinkInterval);

  // Проверка состояния кнопки (дополнительный функционал)

  if (digitalRead(BUTTON\_PIN) == LOW) {

    // Кнопка нажата (можно добавить нужную логику)

    delay(50); // Защита от дребезга

  }

  // Небольшая задержка для стабильности

  delay(10);

}

Доп задание:

Код:

#include <LiquidCrystal.h>

#include <EEPROM.h>

// Определяем пины согласно схеме

const int POT\_PIN = A0;      // Потенциометр

const int RED\_LED = 2;       // Красный светодиод (D2)

const int GREEN\_LED = 3;     // Зеленый светодиод (D3)

const int YELLOW\_LED = 4;    // Желтый светодиод (D4)

const int BUTTON\_PIN = 5;    // Кнопка (D5)

const int BUZZER\_PIN = 6;    // Пьезоизлучатель (D6)

// Пины для LCD дисплея (16x2)

const int RS = 7, EN = 8, D4 = 9, D5 = 10, D6 = 11, D7 = 12;

LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);

// Переменные для работы системы

unsigned long previousMillis = 0;

bool yellowLedState = LOW;

int lastMode = 0;            // 0 - обычный, 1 - альтернативный

int eepromAddress = 0;       // Адрес в EEPROM для сохранения режима

void setup() {

  // Инициализация пинов

  pinMode(RED\_LED, OUTPUT);

  pinMode(GREEN\_LED, OUTPUT);

  pinMode(YELLOW\_LED, OUTPUT);

  pinMode(BUTTON\_PIN, INPUT\_PULLUP);

  pinMode(BUZZER\_PIN, OUTPUT);

  // Инициализация LCD

  lcd.begin(16, 2);

  lcd.print("Mode: Normal");

  lcd.setCursor(0, 1);

  lcd.print("Pot: ");

  // Загрузка последнего режима из EEPROM

  lastMode = **EEPROM**.read(eepromAddress);

  if (lastMode > 1) lastMode = 0; // Защита от некорректных значений

  // Звуковой сигнал при запуске

  tone(BUZZER\_PIN, 1000, 200);

  delay(200);

  tone(BUZZER\_PIN, 1500, 200);

  // Инициализация Serial для отладки

**Serial**.begin(9600);

}

void loop() {

  // Чтение значения потенциометра

  int potValue = analogRead(POT\_PIN);

  int pwmValue = map(potValue, 0, 1023, 0, 255);

  // Обновление LCD дисплея

  updateDisplay(potValue);

  // Обработка кнопки (смена режима)

  if (digitalRead(BUTTON\_PIN) == LOW) {

    delay(50); // Защита от дребезга

    if (digitalRead(BUTTON\_PIN) == LOW) {

      lastMode = !lastMode;

**EEPROM**.write(eepromAddress, lastMode);

      tone(BUZZER\_PIN, 2000, 100);

      lcd.clear();

      lcd.print("Mode: ");

      lcd.print(lastMode ? "Alt" : "Normal");

      delay(300); // Задержка для визуального подтверждения

    }

  }

  // Управление светодиодами в зависимости от режима

  if (lastMode == 0) {

    // Нормальный режим

    analogWrite(RED\_LED, pwmValue);

    analogWrite(GREEN\_LED, 255 - pwmValue);

    blinkYellowLed(potValue);

  } else {

    // Альтернативный режим (можно настроить по-другому)

    analogWrite(GREEN\_LED, pwmValue);

    analogWrite(RED\_LED, 255 - pwmValue);

    blinkYellowLed(1023 - potValue);

  }

  // Небольшая задержка для стабильности

  delay(10);

}

void blinkYellowLed(int potValue) {

  // Мигание желтого светодиода с переменной частотой

  unsigned long currentMillis = millis();

  int blinkInterval = map(potValue, 0, 1023, 100, 1000);

  if (currentMillis - previousMillis >= blinkInterval) {

    previousMillis = currentMillis;

    yellowLedState = !yellowLedState;

    digitalWrite(YELLOW\_LED, yellowLedState);

    // Короткий звуковой сигнал при каждом переключении

    if (yellowLedState) {

      tone(BUZZER\_PIN, 500 + potValue/4, 50);

    }

  }

}

void updateDisplay(int potValue) {

  static unsigned long lastUpdate = 0;

  static int lastPotValue = -1;

  // Обновляем показания потенциометра не чаще чем раз в 200мс

  if (millis() - lastUpdate > 200 || abs(potValue - lastPotValue) > 50) {

    lastUpdate = millis();

    lastPotValue = potValue;

    lcd.setCursor(5, 1);

    lcd.print("    "); // Очистка предыдущего значения

    lcd.setCursor(5, 1);

    lcd.print(map(potValue, 0, 1023, 0, 100));

    lcd.print("%");

  }

}