МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3

**«**Управление устройствами вывода информации. Управление семисегментным индикатором.**»**

по дисциплине «Микропроцессорные системы»

студента 3 курса группы ИВТ-б-о-222(2)

Гоголев Виктора Григорьевича

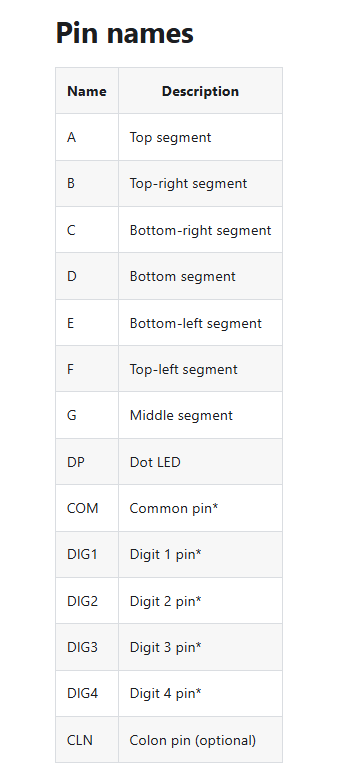
09.03.01 «Информатика и вычислительная техника»

Симферополь, 2025

Цель: получить практические навыки работы с аппаратно- программным комплексом Arduino, использовать его для управления семисегментым индикатором

Оборудование: Персональный компьютер с выходом в Интернет.

Ход работ:

На сайте WINWIE создаем новый проект и ставим Arduino uno r3 и малую плату.

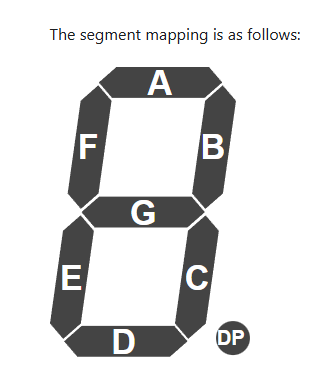


Рисунок 1 – подключение семисегментника

Подключение резисторов к сегментам семисегментинка осуществлялось на основе информации с документации сайта WINWI.

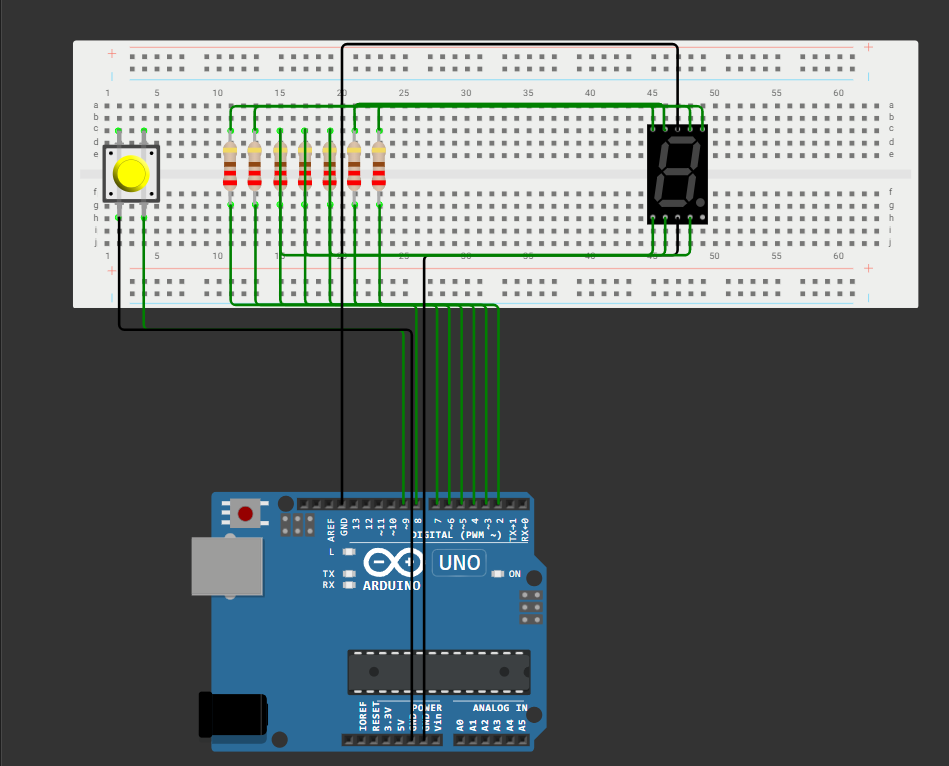


Рисунок 2 – итоговый вариант микросхемы

Потребовалось:

- 1 контактная плата;

- 1 Arduino Uno;

- 1 полноразмерная кнопка;

- 1 семисегментный экран;

- 7 токоограничивающих резисторов резисторов;

- провода для подключения контактов.

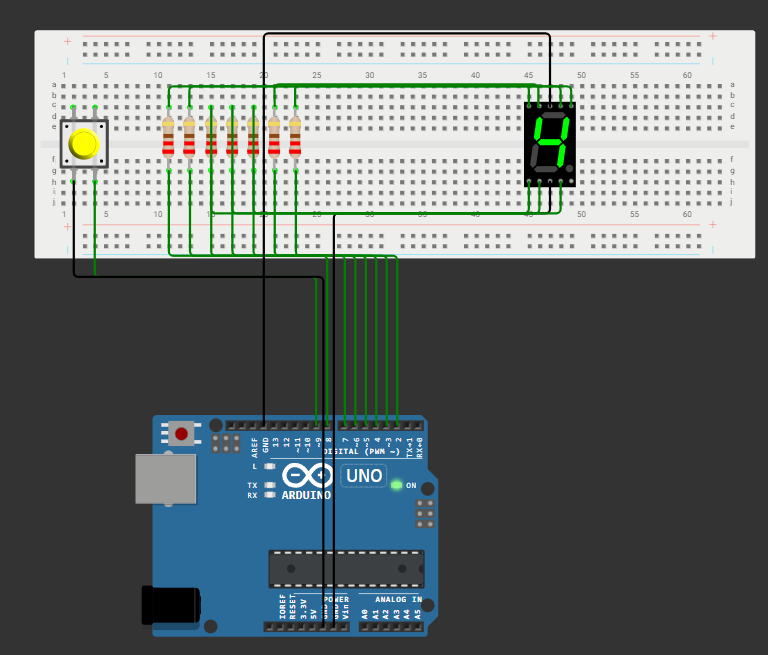


Рисунок 3 - результат работы программы

ВЫВОД

Целью данной лабораторной работы было изучение принципов работы с семисегментным индикатором и микроконтроллером Arduino, а также разработка программы для управления индикатором с помощью кнопки.

Программа была написана на языке C++ для Arduino. Она использует:

* массив для хранения шаблонов отображения цифр (0–9);
* функцию showDigit, которая активирует соответствующие сегменты для отображения текущей цифры;
* логику обработки нажатия кнопки с защитой от дребезга.

Результаты:

* программа успешно отображает цифры от 0 до 9 на семисегментном индикаторе;
* каждое нажатие кнопки увеличивает число на 1, после 9 происходит сброс на 0;
* защита от дребезга кнопки обеспечивает стабильную работу программы.

Цель работы достигнута: разработана и реализована программа для управления семисегментным индикатором с помощью кнопки. В процессе работы были изучены принципы работы с семисегментным индикатором, обработка сигналов с кнопки и защита от дребезга контактов. Полученные навыки могут быть применены в более сложных проектах, связанных с управлением внешними устройствами с помощью Arduino.

ПРИЛОЖЕНИЕ

const int segmentPins[7] = {2, 3, 4, 5, 6, 7, 8}; // A, B, C, D, E, F, G

const int buttonPin = 9;

int currentNumber = 0;

//таблица соотвествия сегментов и цифр

const byte digitPatterns[10][7] = {

  {1, 1, 1, 1, 1, 1, 0}, // 0

  {0, 1, 1, 0, 0, 0, 0}, // 1

  {1, 1, 0, 1, 1, 0, 1}, // 2

  {1, 1, 1, 1, 0, 0, 1}, // 3

  {0, 1, 1, 0, 0, 1, 1}, // 4

  {1, 0, 1, 1, 0, 1, 1}, // 5

  {1, 0, 1, 1, 1, 1, 1}, // 6

  {1, 1, 1, 0, 0, 0, 0}, // 7

  {1, 1, 1, 1, 1, 1, 1}, // 8

  {1, 1, 1, 1, 0, 1, 1}  // 9

};

// Функция для отображения цифры на индикаторе

void showDigit(int number) {

  for (int i = 0; i < 7; i++) {

    digitalWrite(segmentPins[i], digitPatterns[number][i]);

  }

}

// Функция для перехода к следующей цифре

void nextDigit() {

  currentNumber = (currentNumber + 1) % 10; // Увеличиваем число, сбрасывая на 0 после 9

  showDigit(currentNumber); // Отображаем новое число

}

void setup() {

  // Настраиваем пины сегментов как выходы

  for (int i = 0; i < 7; i++) {

    pinMode(segmentPins[i], OUTPUT);

  }

  // Настраиваем пин кнопки как вход с подтяжкой

  pinMode(buttonPin, INPUT\_PULLUP);

  // Показываем начальное число (0)

  showDigit(currentNumber);

}

void loop() {

  // Если кнопка нажата (LOW, так как используется INPUT\_PULLUP)

  if (digitalRead(buttonPin) == LOW) {

    delay(100); // Задержка для защиты от дребезга

    nextDigit(); // Переходим к следующей цифре

    while (digitalRead(buttonPin) == LOW); // Ждем, пока кнопку отпустят

  }

}

Листинг программного кода