|  |  |
| --- | --- |
| Gerb-BMSTU_01 | Министерство науки и высшего образования Российской Федерации  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУК «Информатика и управление» \_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***к курсовой работе на тему:***

***Разработка системы автоматического управления шаговым двигателем на микроконтроллере***

***Вариант №9***

по дисциплине ***Элементы управления в АСОИиУ***

Студент гр. ИУК5-72Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Кузнецов Р.С. )

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_Пономарев С.В. \_\_\_)

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2022

***Министерство науки и высшего образования Российской Федерации***

*Калужский филиал  
 федерального государственного бюджетного   
образовательного учреждения высшего образования*

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***

***(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В. Вершинин)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине ***Элементы управления в АСОИиУ***

Студент **Кузнецов Р.С. ИУК5-72Б** \_.

(фамилия, инициалы, индекс группы)

Руководитель **Пономарев С.В.** \_\_\_\_\_\_

(фамилия, инициалы)

График выполнения работы: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

1. ***Тема курсовой работы***

***«Разработка системы автоматического управления шаговым двигателем на микроконтроллере»***

1. ***Техническое задание***

*«Написать программу управления шагового двигателя, в которой при нажатии кнопки скорость вращения вала двигателя будет снижаться. При этом диод должны извещать о скорости вращения вала двигателя по закону: включено 2 диода – скорость 45% от min скорости вращения вала.»*

1. ***Оформление курсовой работы***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О.)

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г.

(подпись) (Ф.И.О.)

**Содержание**

[1. Техническое задание 5](#_Toc90504051)

[1.1 Общие сведения 5](#_Toc90504052)

[1.1.1 Полное наименование системы и ее условное обозначение. 5](#_Toc90504053)

[1.1.2 Наименование разработчика системы. 5](#_Toc90504054)

[1.1.3 Основания для разработки. 5](#_Toc90504055)

[1.1.4 Плановые сроки начала и окончания работы по созданию системы: 5](#_Toc90504056)

[1.2 Назначение и цели создания системы 5](#_Toc90504057)

[1.2.1 Назначение системы. 5](#_Toc90504058)

[1.2.2 Цели создания системы. 6](#_Toc90504059)

[1.3 Требования к системе 6](#_Toc90504060)

[1.3.1 Требования к системе в целом 6](#_Toc90504061)

[1.4 Требования к видам обеспечения. 6](#_Toc90504062)

[1.4.1 Требования к программному обеспечению. 6](#_Toc90504063)

[1.4.2 Требования к техническому обеспечению. 7](#_Toc90504064)

[1.5 Состав и содержание работ по созданию системы 7](#_Toc90504065)

[1.6 Порядок контроля и приемки системы. 7](#_Toc90504066)

[1.7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие. 8](#_Toc90504067)

[1.8 Требования к документированию. 9](#_Toc90504068)

[2. НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ 10](#_Toc90504069)

[2.1 Постановка задачи проектирования 10](#_Toc90504070)

[2.2 Описание предметной области 10](#_Toc90504071)

[2.2.1Общие сведения о шаговых двигателях. 10](#_Toc90504072)

[2.2.2Архитектура приложения 11](#_Toc90504073)

[2.3 Анализ аналогов. 12](#_Toc90504074)

[2.3.1SMC\_Program 12](#_Toc90504075)

[2.3.2 SMC\_Program\_Modbus 13](#_Toc90504076)

[2.3.3 SMC\_PROGRAM LAN 13](#_Toc90504077)

[2.4 Перечень задач, подлежащих решению в процессе разработки 15](#_Toc90504078)

[2.5 Обоснование выбора платформы разработки 15](#_Toc90504079)

[3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ 17](#_Toc90504080)

[3.1. Разработка аппаратной и программной структуры системы автоматического управления шаговым двигателем на микроконтроллере 17](#_Toc90504081)

[3.2. Разработка алгоритмов системы автоматического управления шаговым двигателем на микроконтроллере 18](#_Toc90504082)

[3.3. Разработка аппаратной архитектуры системы автоматического управления шаговым двигателем на микроконтроллере. 19](#_Toc90504083)

[3.4. Разработка систем передачи информации системы автоматического управления шаговым двигателем на микроконтроллере. 22](#_Toc90504084)

[3.5. Реализация функционирующего программного обеспечения системы автоматического управления шаговым двигателем на микроконтроллере. 22](#_Toc90504085)

[3.6. Разработка интерфейса взаимодействия пользователя с системы автоматического управления шаговым двигателем на микроконтроллере. 22](#_Toc90504086)

[4. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 23](#_Toc90504087)

[4.1. Тестирование и отладка макета аппаратной и программной частей системы автоматического управления шаговым двигателем на микроконтроллере. 23](#_Toc90504088)

[4.2. Разработка руководства пользователя. 23](#_Toc90504089)

[4.3. Экспериментальные данные тестирования процедур и функциональных задач. 27](#_Toc90504090)

[ЗАКЛЮЧЕНИЕ 29](#_Toc90504091)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30](#_Toc90504092)

# Техническое задание

## 1.1 Общие сведения

### Полное наименование системы и ее условное обозначение.

Разработка системы автоматического управления шаговым двигателем на микроконтроллере.

### Наименование разработчика системы.

Исполнителем проекта является студент Калужского филиала МГТУ им. Н. Э. Баумана, факультета ИУК «Информатики и управления», кафедры ИУК5 «Системы обработки информации», группы ИУК5 – 72Б, Кузнецов Роман Сергеевич.

### Основания для разработки.

Основанием для разработки приложения является задание курсовой работы в соответствии с учебным планом дисциплины «Элементы управления в АСОИиУ»

### Плановые сроки начала и окончания работы по созданию системы:

Начало работы: 14.09.2022

Окончание работы: 30.12.2022

### Назначение системы.

Система предназначена для управления шаговым двигателем.

### Цели создания системы.

Целью создания системы является: упрощение работы с шаговым двигателем.

### Требования к системе в целом

Автоматизированная система должна представлять собой программу, управления шаговым двигателем.

Должна присутствовать кнопка, при нажатии которой скорость снижается. Диоды должны показывать какая сейчас скорость – если 45% от минимальной, загорается 2 светодиода.

Использовать порты микроконтроллера для включения двигателя на определенный промежуток времени, а также для включения светодиодов и реагирования на нажатие кнопки.

Управлять мощностью двигателя при помощи программной и аппаратной широтно-импульсной модуляции

### Требования к программному обеспечению.

Реализуемое решение требует для своей работы установки следующего ПО:

1. Операционная система: Microsoft Windows 7 или выше,
2. IDE Atmel Studio;
3. Виртуальный стенд

### Требования к техническому обеспечению.

Для функционирования ИС необходимо:

1. 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 гигагерц (ГГц) или выше;
2. 1 гигабайт (ГБ) (для 32-разрядной системы) или 2 ГБ (для 64-разрядной системы) оперативной памяти (ОЗУ);
3. 4 гигабайт (ГБ) (для 32-разрядной системы) или 8 ГБ (для 64-разрядной системы) пространства на жестком диске;

## 1.2 Состав и содержание работ по созданию системы

Расчетно-пояснительная записка:

* Титульный лист с подписями.
* Утвержденное задание на курсовую работу.
* Содержание (с постраничной разметкой).
* Техническое задание;
* Исследовательская часть;
* Проектно-конструкторская часть;
* Проектно-технологическая часть.
* Заключение.
* Список использованных источников.

Приложения.

## Порядок контроля и приемки системы.

Первый этап 1 – 4-я недели.

Выбор и согласование темы с руководителем; проведение исследования по выбранной теме, обзор существующих программных продуктов; разработка технического задания.

Второй этап 5 – 7-я недели.

Разработка и реализация алгоритмов функционирования приложения, структуры, систем передачи информации, технологий обработки информации и интерфейса взаимодействия пользователя с системой. Выполнение и оформление проектно-конструкторской части работы.

Третий этап 9 – 10-я недели.

Тестирование и отладка программного приложения. Разработка руководства пользователя и программиста, спецификации необходимого оборудования. Написание проектно-технологической части расчетно-пояснительной записки.

Четвертый этап 11 – 14-я недели.

Завершающее оформление документации согласно требованиям ГОСТ и данного методического пособия. Подготовка доклада. Защита курсовой работы.

## 1.4 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.

По окончанию работы должны быть предоставлены:

1. Расчетно-пояснительная записка
2. Работоспособное и протестированное приложение.
3. Дополнительная документация (приложения, графики и т.п.)

## Требования к документированию.

Техническое задание было разработано на основе «Методических указаний по выполнению курсовой работы».

# 2. НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

## 2.1 Постановка задачи проектирования

Написать программу управления шагового двигателя, в которой при нажатии кнопки скорость вращения вала двигателя будет снижаться. При этом диод должны извещать о скорости вращения вала двигателя по закону: включено 2 диода – скорость 45% от min скорости вращения вала.

## 2.2 Описание предметной области

### 2.2.1 Общие сведения о шаговых двигателях.

Что такое шаговый двигатель и как он устроен?

Шаговый двигатель — это бесколлекторный двигатель, ротор которого вращается не плавно, а шагами (дискретно). Полный оборот ротора состоит из нескольких шагов. Меняя форму сигнала, количество импульсов, их длительность и фазовый сдвиг, можно задавать скорость вращения, направление вращения и количество оборотов ротора двигателя.

Шаговые двигатели состоят из ротора (подвижная часть) и статора (неподвижная часть). На статоре устанавливают электромагниты, а части ротора взаимодействующие с электромагнитами выполняются из магнитотвердого (двигатель с постоянными магнитами) или магнитомягкого (реактивный двигатель) материала.

Существует три основных вида шаговых двигателей:

Двигатель с постоянными магнитами (ротор из магнитотвердого материала). На роторе установлен один, или несколько, постоянных магнитов. Количество полных шагов в одном обороте таких двигателей, зависит от количества постоянных магнитов на роторе, и количества электромагнитов на статоре. Обычно в одном обороте от 4 до 48 шагов (один шаг от 7,5° до 90°).

Реактивный двигатель. Такие двигатели называют двигателями с переменным магнитным сопротивлением. Ротор не имеет постоянных магнитов, он выполнен из магнитомягкого материала в виде многоконечной звезды. Данные двигатели встречаются редко, так как у них наименьший крутящий момент, по сравнению с остальными, при тех же размерах. Количество полных шагов в одном обороте таких двигателей, зависит от количества зубцов на звезде ротора, и количества электромагнитов на статоре. Обычно в одном обороте от 24 до 72 шагов (один шаг от 5° до 15°.)

Гибридный двигатель совмещает технологии как реактивного, так двигателя так и двигатель с постоянными магнитами. Ротор выполнен из магнитотвердого материала (как у двигателя с постоянными магнитами), но имеет форму многоугольной звезды (как у реактивного двигателя). Количество полных шагов в одном обороте таких двигателей, зависит от количества постоянных магнитов на звезде ротора, и количества электромагнитов на статоре. Количество шагов в одном обороте таких двигателей может доходить до 400 (один шаг от 0,9°).

### 2.2.2 Архитектура приложения

Так как данная программа должна обеспечивать работу шагового двигателя в механизме и не должна связываться с сервером, приложение будет иметь одноуровневую архитектуру.

Микроконтроллер перепрошивают, после чего он выполняет свою работу.

## 2.3 Анализ аналогов.

Разрабатываемая программа не является уникальной и имеет несколько аналогов. Однако аналоги можно будет использовать в качестве примера.

### 2.3.1 SMC\_Program

Программа SMC\_Program предназначена для работы с программируемыми блокамим управления шаговыми двигателями SMSD‑4.2, SMSD‑1.5 и SMSD‑9.0. Программа подает команды для управления шаговыми двигателями через Com порт персонального компьютера или USB (интерфейс RS‑232).

Программа может управлять одним, двумя или тремя шаговыми двигателями, подключенными к контроллеру SMC‑3 или одним шаговым двигателем, подключенным к блоку SMSD‑1.5, SMSD‑3.0 или SMSD‑9.0, а также осуществлять запись перечня команд (исполняемую программу) для последующего автономного использования контроллера или блока серии SMSD (без помощи ПК), либо для запуска такой исполняемой программы при помощи ПК

Имеется возможность сохранять исполняемые программы на ПК в отдельные файлы и загружать готовые файлы в программу. Программа имеет простой графический интерфейс, ориентированный на неподготовленного пользователя. SMC\_Program не требует установки или каких-либо специальных требований к ПК. После копирования на жесткий диск ПК программа SMC\_Program сразу готова к работе. При запуске программы все настойки передачи по COM‑порту подставляются автоматически, остается только выбрать номер порта, к которому подключен контроллер или блок. Для справки все параметры открытого порта указываются внизу окна программы. В комплекте с программой SMC\_Program - подробное руководство пользователя и примеры управляющих программ.

### 2.3.2 SMC\_Program\_Modbus

Программа SMC\_Program\_Modbus предназначена для управления шаговыми приводами с использованием программируемых блоков управления SMSD‑1.5Modbus. Программа имеет простую панель управления, а также возможность составлять и записывать программы в память контроллера. Версия программы SMC\_Program\_Modbus распространяется бесплатно.

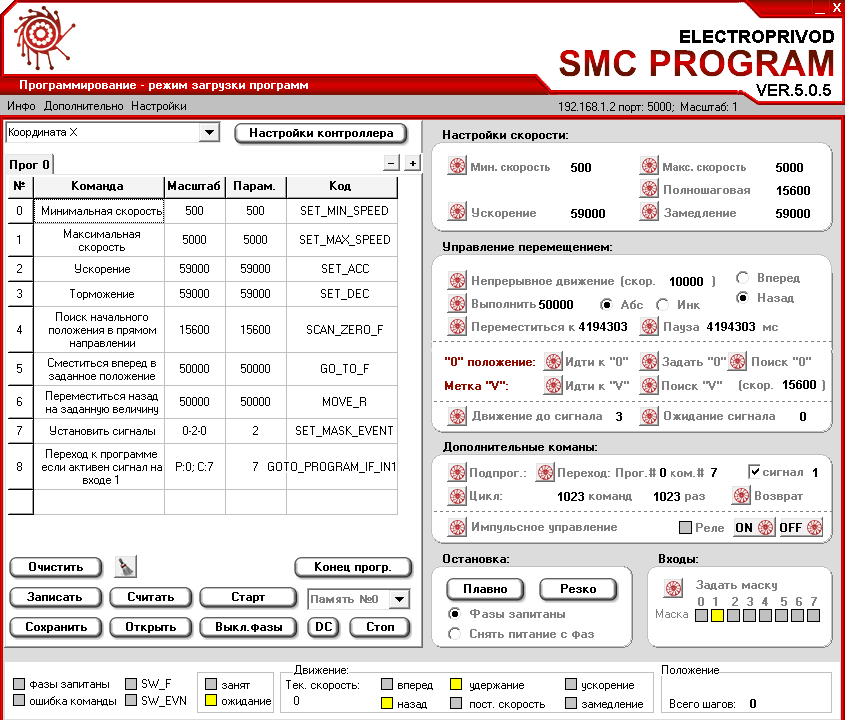


Рисунок 1 – Окно приложения SMC\_Program\_Modbus

### 2.3.3 SMC\_PROGRAM LAN

Программа SMC\_Program LAN предназначена для управления шаговыми приводами с использованием программируемых блоков управления серии SMSD LAN. Программа предусматривает подключение нескольких блоков через интерфейс USB или Ethernet, имеет простую панель управления и дружественный обновленный интерфейс. Программа предусматривает управление шаговым приводом, настройку параметров контроллера, составление и загрузку программ управления (в том числе нескольких программ одновременно), а также предоставляет возможность сохранять результаты работы в файл на ПК и загружать готовые программы из файлов.

## 2.4 Перечень задач, подлежащих решению в процессе разработки

1. Изучить предметную область и на основе полученной информации определить вид программы.

2. Разработать план поэтапного формирования

3. Создать или изучить готовые алгоритмы необходимых операций

4. При помощи виртуального стенда создать схему, в которой будет присутствовать микроконтроллер, шаговый двигатель и другие нужные детали

5. Написать код для микроконтроллера, удовлетворяющий условиям заданий.

## 2.5 Обоснование выбора платформы разработки

В качестве языка разработки был выбран C т. к. более предпочтителен на основе имеющихся навыков и знаний, а также имеет в своих библиотеках необходимые функции и методы. Этот язык удобней ассемблера и позволяет работать с портами микроконтроллера.

В качестве ПО для сборки виртуальной схемы и эмуляции работы был выбран Виртуальный Стенд. ВС - среда для проектирования и отладки электронных устройств, в т.ч. выполненных на основе микроконтроллеров различных семейств. Предоставляет возможности ввода схемы в графическом редакторе, моделирования её работы и разработки печатной платы, включая трехмерную визуализацию её сборки. Уникальной чертой среды ВС является возможность эффективного моделирования работы разнообразных микроконтроллеров (PIC, 8051, AVR, HC11, ARM7/LPC2000 и др.) и отладки микропрограммного обеспечения.

Виртуальный стенд имеет огромную библиотеку электронных компонентов, а недостающие - можно сделать самостоятельно. Предусмотрена поддержка SPICE-моделей, которые часто предоставляются производителями электронных компонентов.

Среда разработки Atmel Studio — основанная на Visual Studio бесплатная проприетарная интегрированная среда разработки (IDE) для разработки приложений для 8- и 32-битных микроконтроллеров семейства AVR и 32-битных микроконтроллеров семейства ARM от компании Atmel, работающая в операционных системах Windows NT/2000/XP/Vista/7/8/10. Atmel Studio содержит компилятор GNU C/C++ и эмулятор, позволяющий отладить выполнение программы без загрузки в микроконтроллер

Основные преимущества:

* Упрощается использование и освоение микроконтроллеров.
* Atmel Studio разработана для использования на любой стадии проектирования
* Atmel Studio бесплатна

# 3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

## 3.1. Разработка аппаратной и программной структуры системы автоматического управления шаговым двигателем на микроконтроллере

Первой задачей при создании схемы стал выбор подходящего микроконтроллера.

При выборе микроконтроллера было 2 варианта из семейства AVR. Микроконтроллер Atmega8 или микроконтроллер Atmega16. Программная память первого равна 8 кБ, цикл составляет 10000 раз. Программная память второго равна 16 кБ. Для поставленной задачи не нужно много свободных пинов и большой объём памяти, по этой причине был выбран микроконтроллер Atmega8.

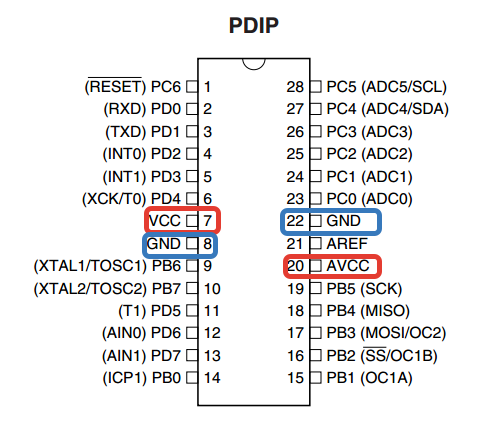


Рисунок 2 – Распиновка микроконтроллера Atmega8

Следующим шагом стал выбор шагового двигателя. Я выбрал «MOTOR-BISTEPPER» это биполярный шаговый, который относятся к виду бесколлекторных двигателей непрерывного (постоянного) тока.

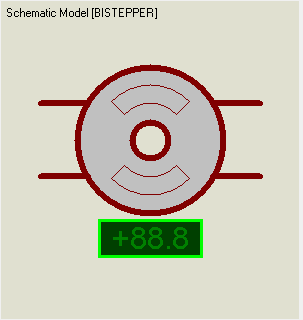


Рисунок 3 – Биполярный шаговый двигатель на схеме

## 3.2. Разработка алгоритмов системы автоматического управления шаговым двигателем на микроконтроллере

Так как задача является тривиальной нам будет достаточно подключить заголовочный файл avr/io.h. В этом файле находятся определения констант, имен регистров и всего прочего, что может понадобиться для выполнения базового ввода-вывода. Сюда включены определения набора регистра ввода / вывода и названия их разрядов, в соответствии с документацией Atmel. Без этого заголовочного файла нельзя работать с микроконтроллерами семейство AVR.

Следующий заголовочный файл, это <util/delay.h>. util/delay.h - библиотека, которая отвечает за задержки — это такие функции, как \_delay\_ms(), \_delay\_us().

Что бы пользоваться этой библиотекой, нужно в самом начале указать частоту используемого генератора. В моём случае это 1 мегагерц, то есть в программе указано это так: #define F\_CPU 1000000UL.

Вся следующая часть программы написана в main.c.

У микроконтроллера были задействованы 3 порта – B, C и D.

К первому подключается кнопка для переключения скоростей. Чтобы настроить данный порт на ввод было прописано DDRB=0x0. Командой PORTB=0xFE на указанные разъемы данного порта подается напряжение.

Второй связан со специальным драйвером, который уже управляет самим шаговым двигателем. Данный порт уже настроен на вывод с помощью DDRC = 0xFF и питание подается циклично.

Последний порт задействован для диодов и также настроен на вывод: DDRD=0xFF. Аналогично с предыдущем питание циклично.

После настройки портов был создан бесконечный цикл while(1). В цикле задействован оператор множественного выбора switch-case. Всего 4 случая работы приложения и для каждого прописан case.

Case1 – двигатель остановлен. Здесь прописано PORTD = 0x4, то есть подача питания на красную лампочку.

Case2 – питание с красной лампочки убирается и подается на желтую и зеленую PORTD = 0x3. Также выполняется 8 смен питания на порте С для совершения полного круга поворота вала: PORTC = 0x8, PORTC = 0xC, PORTC = 0x4, PORTC = 0x6, PORTC = 0x2, PORTC = 0x3, PORTC = 0x1. Для придания нужной скорости используется \_delay\_ms(300) после каждой из 8 смены питания. На 2 скорости задержка 300, то есть максимальная задержка между полным циклом вращения – 2,4 сек. Это минимальная выбранная скорость вращения.

Case3 – питание убирается с желтой и остается только на зеленой: PORTD = 0x1, движение двигателя продолжается аналогично Case2, но с задержкой 145, что является 45% от максимальной скорости, т.к. максимальная становится при задержке 100.

Case3 – не горит ни один диод, все 8 операций выполняются с задержкой 100 миллисекунд.

Сами переключатели зависят от переменной int speed, начальное значение которой 1, а конечное 4, она и является переключением скорости.

В цикле также проверяется нажатие кнопки, если она нажата и скорость вращения не максимальная (4), переменная speed инкрементируется на 1.

## 3.3. Разработка аппаратной архитектуры системы автоматического управления шаговым двигателем на микроконтроллере.

Силы тока микроконтроллера недостаточно для надёжной работы шагового двигателя. Для управления двигателями робота необходимо устройство, которое бы преобразовывало управляющие сигналы малой мощности в токи, достаточные для управления моторами. Такое устройство называют драйвером двигателей. достаточно много самых различных схем для управления электродвигателями. Они различаются как мощностью, так и элементной базой, на основе которой они выполнены.

Я выбрал один из самых простых драйверов управления двигателями. Он выполнен в виде полностью готовой к работе микросхемы. Эта микросхема называется L293D и является одной из самых распространенных микросхем, предназначенных для этой цели. L293D содержит сразу два драйвера для управления электродвигателями небольшой мощности (четыре независимых канала, объединенных в две пары). Имеет две пары входов для управляющих сигналов и две пары выходов для подключения электромоторов. Кроме того, у L293D есть два входа для включения каждого из драйверов. Эти входы используются для управления скоростью вращения электромоторов с помощью широтно-модулированного сигнала (ШИМ).

L293D обеспечивает разделение электропитания для микросхемы и для управляемых ею двигателей, что позволяет подключить электродвигатели с большим напряжением питания, чем у микросхемы. Разделение электропитания микросхем и электродвигателей может быть также необходимо для уменьшения помех, вызванных бросками напряжения, связанными с работой моторов.

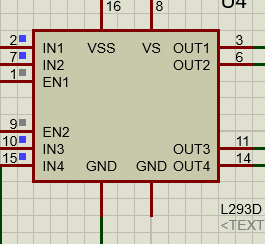


Рисунок 4 – Драйвер управления шаговым двигателем.

Для выполнения требований из ТЗ было размещено 4 синих светодиода. Первый горит, когда вал двигателя вращается со скоростью от 0 до 25% от максимальной скорости вращения. Когда горят первые два светодиода – скорость вращения составляет от 25% до 50%, горят первые три светодиода – скорость 50-75% от максимальной. Если же горят все четыре – вал вращается со скоростью 75%-100%.

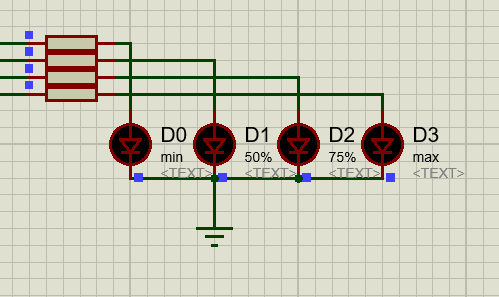


Рисунок 5 – Схема светодиодов

Так же были добавлены резисторы с сопротивлением в 100 Ом.

Конечный вид схемы разрабатываемой программы представлен ниже:

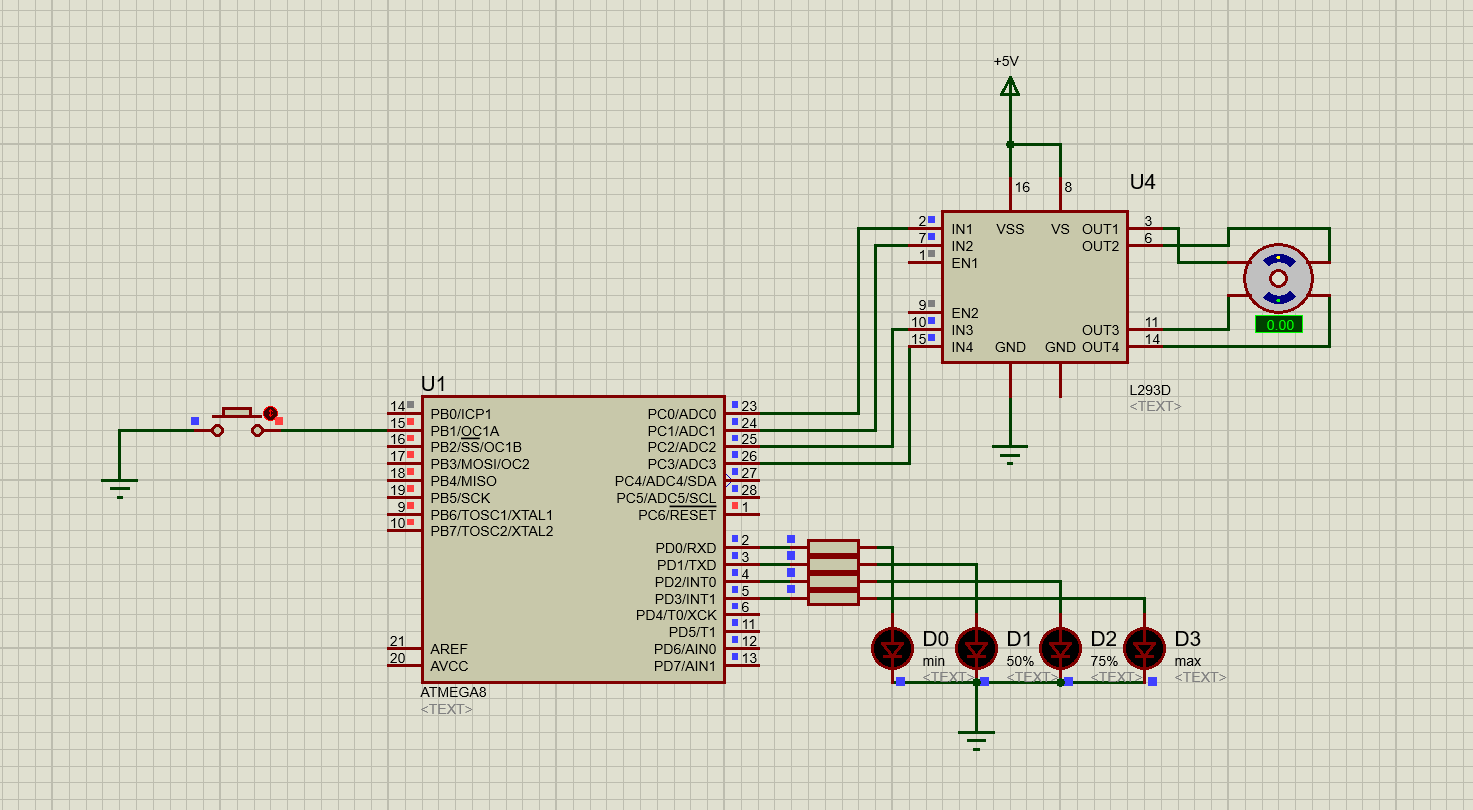


Рисунок 6 – Полная схема устройства.

## 3.4. Разработка систем передачи информации системы автоматического управления шаговым двигателем на микроконтроллере.

Передача информации на настоящий микроконтроллер осуществляется при помощи прошивки.

Для этого необходимо специальное устройство-программатор и программа, которая производит прошивку. В этом проекте будут использованы уже готовые решения.

## 3.5. Разработка интерфейса взаимодействия пользователя с системы автоматического управления шаговым двигателем на микроконтроллере.

В этом приложении будет использован стандартный интерфейс. Интерфейс взаимодействие представляет из себя запуск программы в виртуальном стенде. Приложение в стенде берет код из Atmel автоматически.

# 4. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## 4.1. Тестирование и отладка макета аппаратной и программной частей системы автоматического управления шаговым двигателем на микроконтроллере.

Для проверки отказоустойчивости будет выполнено тестирование на виртуальном стенде.

Порядок выполнения:

1. «Виртуальная прошивка микроконтроллера»,

2. Начало симуляции

Таблица 1 — Ожидаемые результаты тестирования работы микроконтроллера

|  |  |
| --- | --- |
| Действие | Ожидаемый результат |
| Симуляция не активна | Светодиоды не горят, двигатель отключен |
| Активация симуляции | Светодиоды не горят, двигатель не вращается |
| Нажатие на кнопку | Шаговый двигатель начинает вращение, и загорается первый светодиод со скоростью 25% от максимальной, после полного оборота скорость двигателя увеличивается на 25% и загорается следующий светодиод. После достижения скорости 100% загораются все четыре светодиода, вал делает полный оборот и двигатель останавливается |

## 4.2. Разработка руководства пользователя.

Чтобы программа начала работу необходимо прошить контроллер Atmega 8. Для прошивки микроконтроллеров Atmel AVR необходим программатор, например для прошивки через USB-порт такой - «USB ASP ПРОГРАММАТОР МИКРОКОНТРОЛЛЕРОВ ATMEGA AVR».

После чего нужно собрать схему из микроконтроллера, драйверов управления двигателями L293D, шагового двигателя и 4 светодиодов. А также добавить источник тока и резисторы в нужные места.

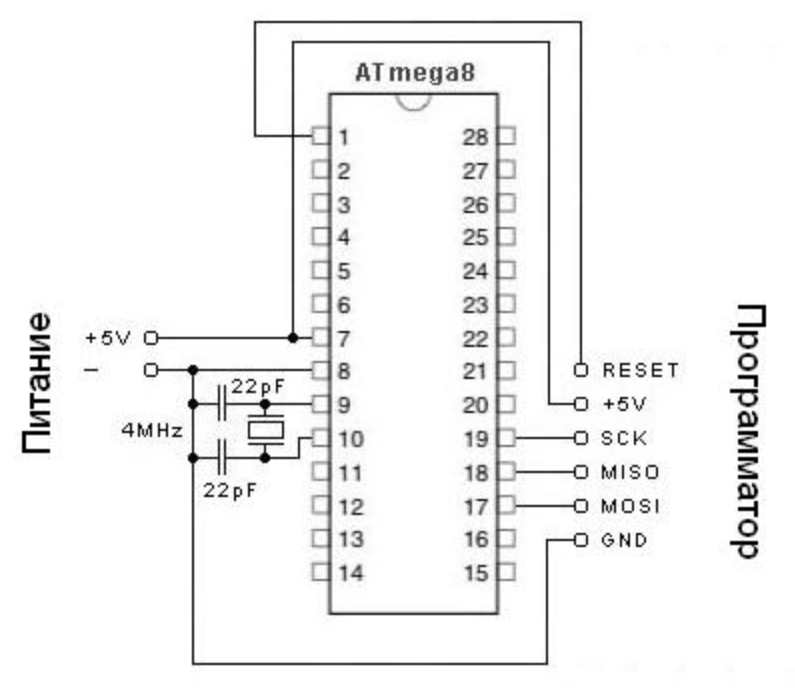


Рисунок 7 – Схема подключения программатора

Для прошивки контроллера необходимо подключить программатор, затем необходимо подключить питание на программатор.

Запускаем программу для прошивки контроллеров, заходим настройки и запускаем проверку. Если появится сообщение “Тест завершен”, значит программатор работает корректно.

Выбираем тип и модель контроллера, в данном случае это AVR micro и ATMEGA8.

Для прошивки контроллера выбираем файл прошивки, который имеет расширение hex. Нажимаем кнопку “Записать устройство”, данное действие запишет в память микроконтроллера выбранный код.

## 4.3. Экспериментальные данные тестирования процедур и функциональных задач.

Для проведения итогового теста вернёмся в окно виртуального стенда. Мы увидим полностью собранную схему

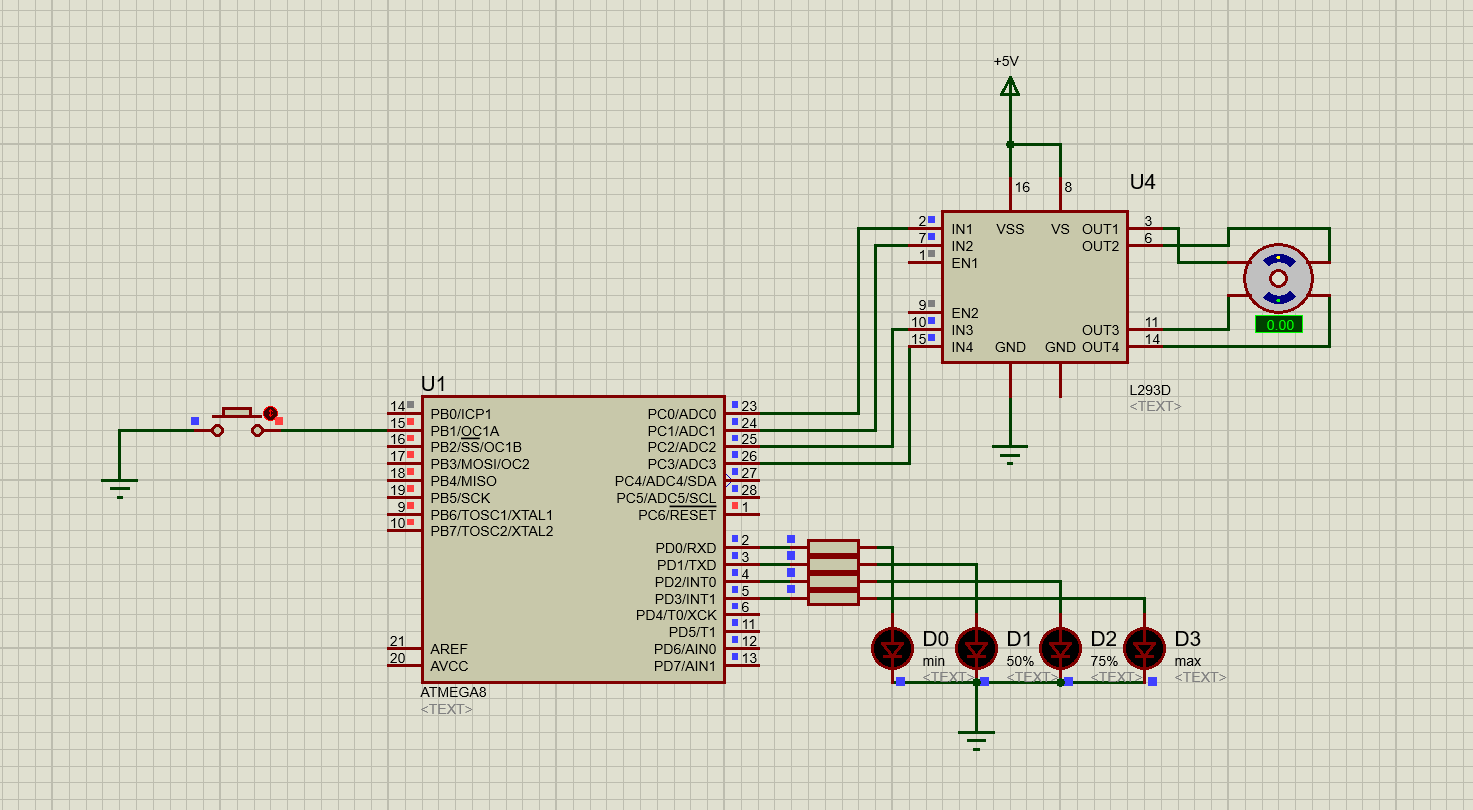


Рисунок 8 – Двигатель не вращается

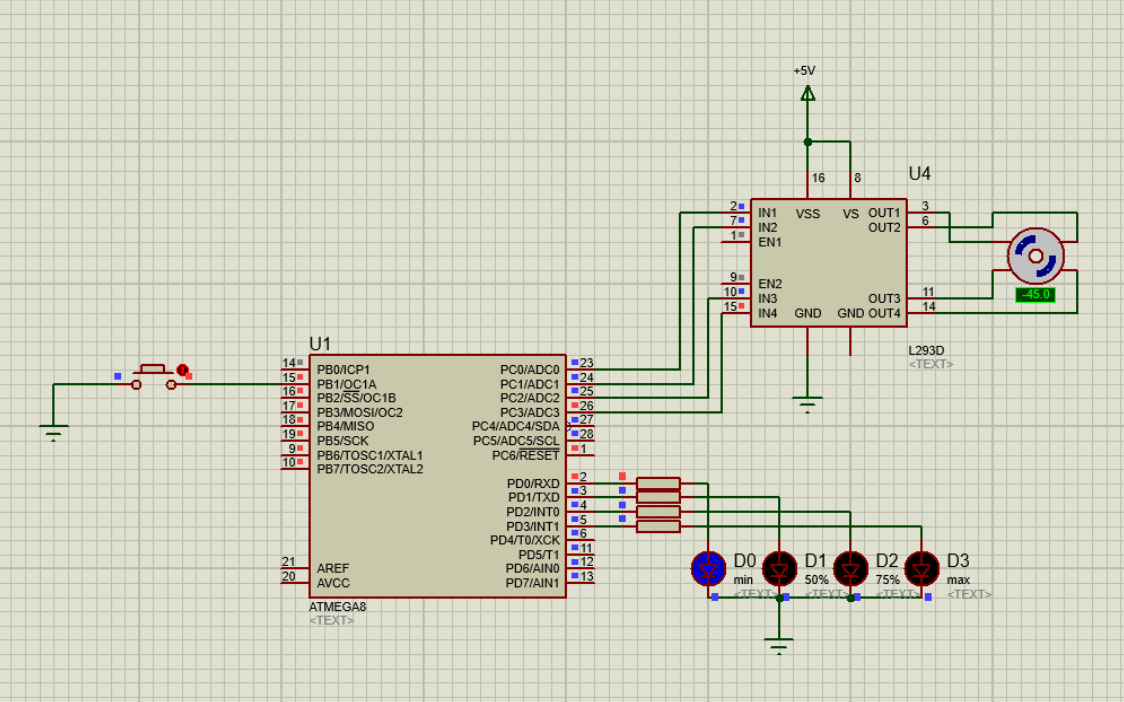


Рисунок 9 – Скорость вращения 25% от максимальной

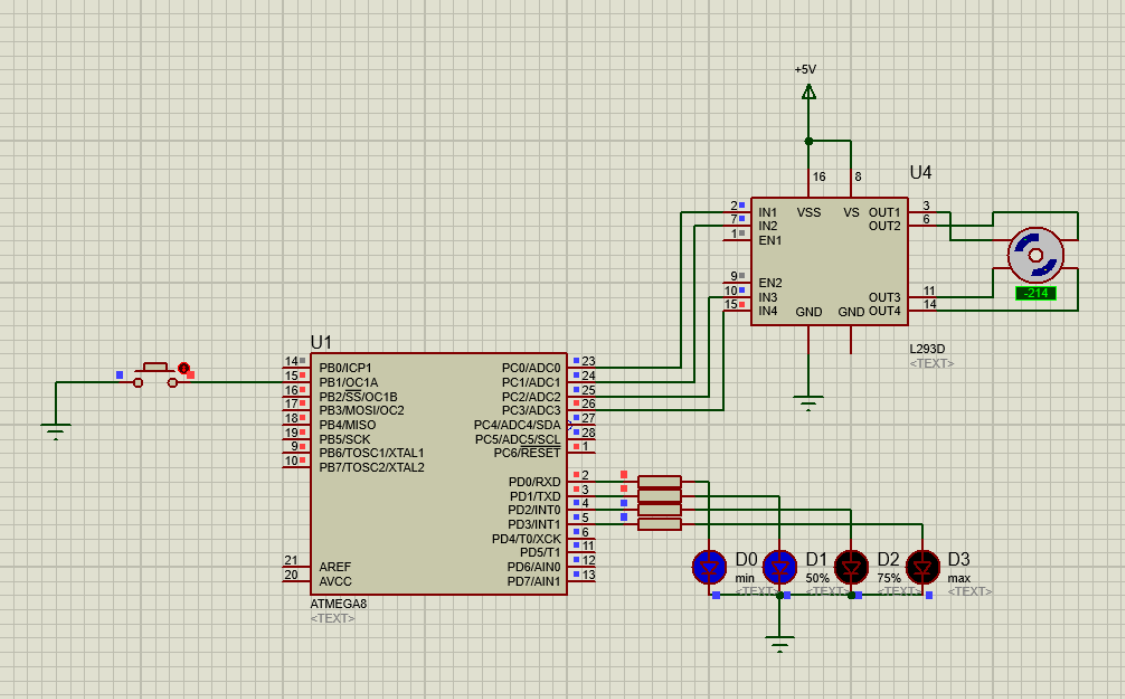


Рисунок 10 – Скорость вращения 50% от максимальной

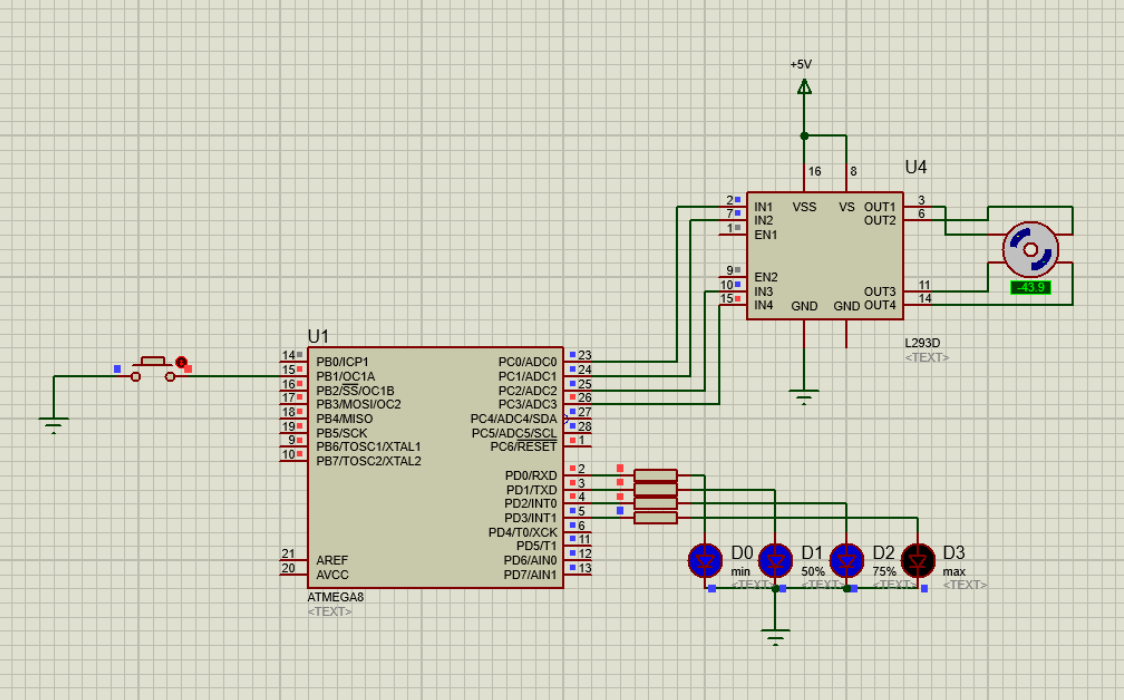


Рисунок 11 – Скорость вращения 75% от максимальной

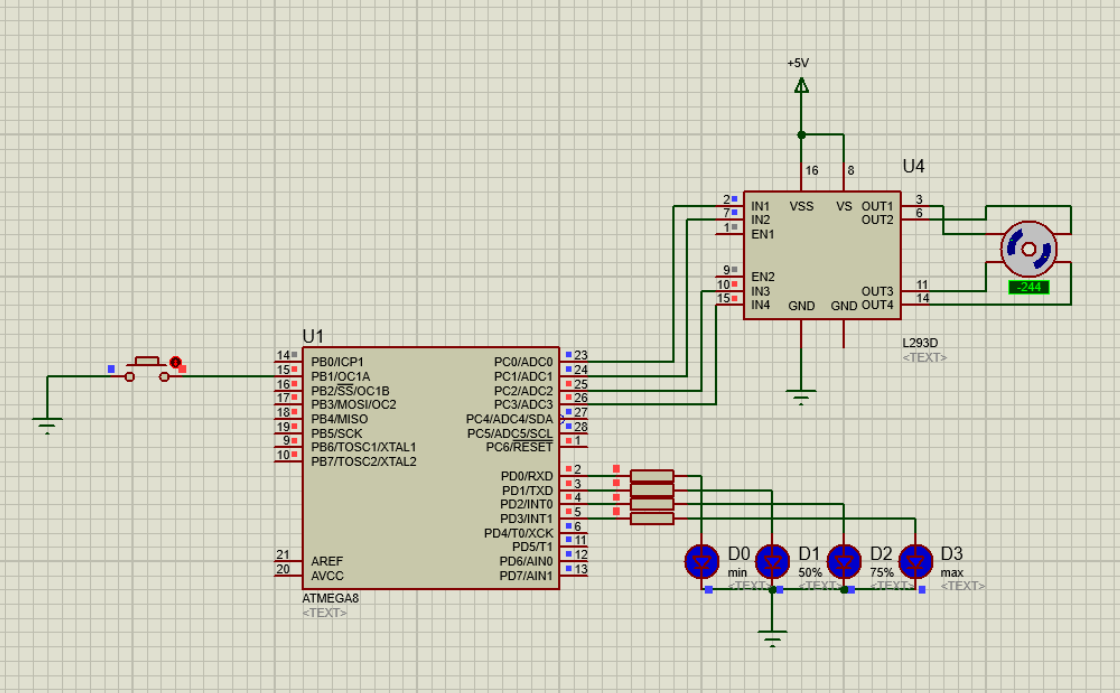


Рисунок 12 – Скорость вращения максимальная

После завершения работы, двигатель останавливается и светодиоды гаснут.

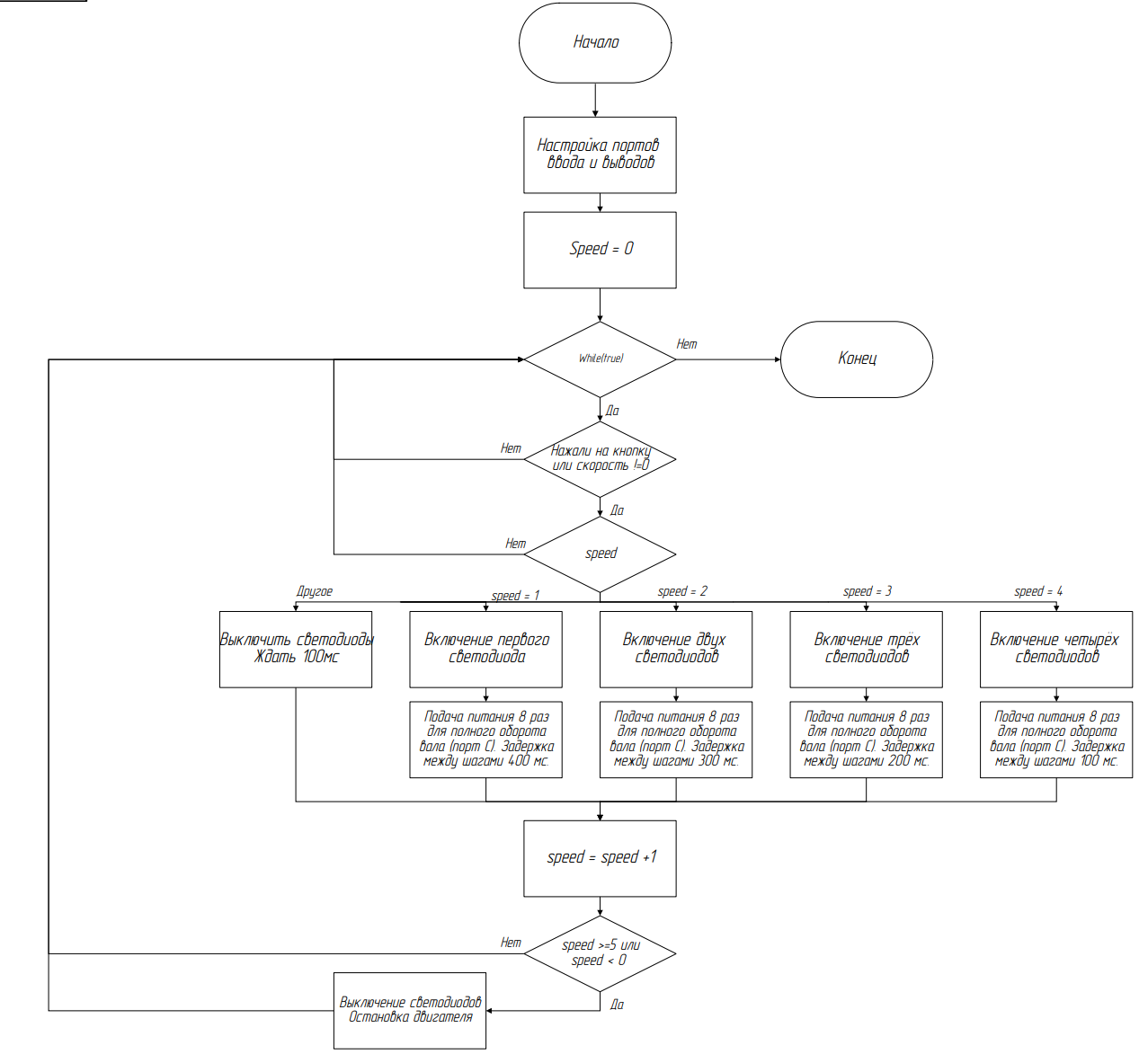


Рисунок 13 – Алгоритм работы приложения

Приложение – листинг программы

#define F\_CPU 1000000UL

#include <avr/io.h>

#include <util/delay.h>

void step(int speed) {

switch(speed){

case 1:

PORTD = 0x01;

PORTC = 0x9;

\_delay\_ms(400);

PORTC = 0x8;

\_delay\_ms(400);

PORTC = 0xC;

\_delay\_ms(400);

PORTC = 0x4;

\_delay\_ms(400);

PORTC = 0x6;

\_delay\_ms(400);

PORTC = 0x2;

\_delay\_ms(400);

PORTC = 0x3;

\_delay\_ms(400);

PORTC = 0x1;

\_delay\_ms(400);

break;

case 2:

PORTD = 0x03;

PORTC = 0x9;

\_delay\_ms(300);

PORTC = 0x8;

\_delay\_ms(300);

PORTC = 0xC;

\_delay\_ms(300);

PORTC = 0x4;

\_delay\_ms(300);

PORTC = 0x6;

\_delay\_ms(300);

PORTC = 0x2;

\_delay\_ms(300);

PORTC = 0x3;

\_delay\_ms(300);

PORTC = 0x1;

\_delay\_ms(300);

break;

case 3:

PORTD = 0x07;

PORTC = 0x9;

\_delay\_ms(200);

PORTC = 0x8;

\_delay\_ms(200);

PORTC = 0xC;

\_delay\_ms(200);

PORTC = 0x4;

\_delay\_ms(200);

PORTC = 0x6;

\_delay\_ms(200);

PORTC = 0x2;

\_delay\_ms(200);

PORTC = 0x3;

\_delay\_ms(200);

PORTC = 0x1;

\_delay\_ms(200);

break;

case 4:

PORTD = 0x0F;

PORTC = 0x9;

\_delay\_ms(100);

PORTC = 0x8;

\_delay\_ms(100);

PORTC = 0xC;

\_delay\_ms(100);

PORTC = 0x4;

\_delay\_ms(100);

PORTC = 0x6;

\_delay\_ms(100);

PORTC = 0x2;

\_delay\_ms(100);

PORTC = 0x3;

\_delay\_ms(100);

PORTC = 0x1;

\_delay\_ms(100);

break;

default:

PORTD = 0x00;

\_delay\_ms(100);

break;

}}

int main(void) {

DDRB = 0x0; // все разъемы на вход

DDRC = 0xFF; //все разъемы PC на выход

DDRD = 0xFF; //все разъемы PD на выход

PORTB = 0xFE; //питание для кнопки

int speed = 0;

while (1) {

if (PINB == 0xFD || speed !=0)

{

step(speed);

speed++;

}

if (speed>=5 || speed<=0)

{

speed=0;

step(0);

} } }

# ЗАКЛЮЧЕНИЕ

При выполнении курсовой работы «Разработка системы автоматического управления шаговым двигателем на микроконтроллере» по дисциплине «Элементы управления в автоматизированных системах обработки информации и управления» была выбрана и описана предметная область, а также выделена цель применения программы.

Выполнен анализ аналогов и обоснован выбор платформы для разработки.

Составлена Научно-Исследовательская часть для введения в тему работы и описания подходов к решению, Проектно-Конструкторская часть для понимания структуры и архитектуры разработанного приложения, и Проектно-Технологическая часть для демонстрации интерфейса полученного приложения, и обоснования способов взаимодействия с ним, приведения его спецификации и проведения тестирования

Параллельно написанию РПЗ было разработано приложение, соответствующее представленным в ТЗ требованиям. В процессе работы получены знания и навыки программирования микроконтроллеров, создание схем и навыки прошивки микроконтроллеров.

Созданное приложение можно улучшить в будущем.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Ачилов Р.Н. Самоучитель разработчика устройств на микроконтроллерах AVR:. М: ДМКпресс, 2013. <https://e.lanbook.com/book/66472?category_pk=1547#book_name>

2) Васильков А.В., Васильков А.А., Васильков И.А. Микроконтроллеры AVR: практикум для начинающих. М.: Форум, 2013.

3) Е.В. Смирнова, А.В. Пролетарский, Е.А. Ромашкина и др. Практика программирования микроконтроллеров AVR. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2013.

4) Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификационной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.

5) Каталог библиотек для работы с микроконтроллерами. [Электронный ресурс] – URL:// <https://metanit.com/sharp/wpf/6.php>

6) Полное руководство по языку программирования С 5. [Электронный ресурс] – URL:// <https://metanit.com/sharp/tutorial/>