

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

**ИНСТИТУТ ЭНЕРГЕТИКИ И ЭЛЕКТРОТЕХНИКИ**

(институт)

**Промышленная электроника**

(кафедра)

**11.03.04 Электроника и наноэлектроника**

(код и наименование направления подготовки, специальности)

**Промышленная электроника**

(профиль)

**БАКАЛАВРСКАЯ РАБОТА**

на тему **Электронный демонстрационный стенд с использованием  
шагового двигателя**

Студент(ка)

И.В. Гончаров

(И.О. Фамилия)

(личная подпись)

Руководитель

А.В. Прядилов

(И.О. Фамилия)

(личная подпись)

Консультанты

М.В Емелина

(И.О. Фамилия)

(личная подпись)

**Допустить к защите**

Заведующий кафедрой, к.т.н., доцент А.А. Шевцов

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

Тольятти 2018

## Аннотация

Тема бакалаврской работы: Электронный демонстрационный стенд с использованием шагового двигателя

Объем бакалаврской работы 48 страниц, на которых размещены 39 рисунков и 2 таблицы, графическая часть состоит из 6 плакатов формата А1. При написании выпускной квалификационной работы использовалось 26 источников.

Ключевые слова работы: Шаговый двигатель, Arduino, электромагнит, программирование, плата, контролер.

Цель работ - стенд на основе Arduino, позволяющий управлять режимом работы шагового двигателя и электромагнитом. Источником питания служит адаптер 220В.

Данная работа состоит из трех частей:

1. Состояние вопроса. В этой части формируется цель и задачи проекта, а так же анализируются исходные данные и известные решения
2. Проектный раздел. В проектном разделе происходит разработка электрической схемы, конструкции устройства и программной части устройства. Так же проводится выбор необходимых компонентов и краткое описание устройства и инструкция по работе с ним.
3. Практическая реализация проекта. В заключительной части происходит проверка и отладка программной части устройства.

В качестве вывода можно отметить, что областью применения разрабатываемого стенда является научные выставки для детей, школьников и абитуриентов.

## **Abstract**

The given graduation work is devoted to the electronic demonstration stand with a stepper motor.

The graduation project consists of an explanatory note on 48 pages, including 39 figures, 2 tables, the list of 26 references including 6 foreign sources and 9 appendices, and the graphic part on 6 A1 sheets.

The goal of this graduation work is to develop the demonstration stand showing the action of the stepper motor in various modes: single-step / half step, manual / automatic, reverse.

We begin with the statement of the problem and then logically pass over to its possible solutions.

The practical work is divided into three main parts:

1. State of the issue. This article analyzes the initial data and the existing solutions.

2. Project part. In the design chapter, the circuits, the device designs and the software part of the devices are developed. Also, the necessary components are selected and a short description of the device as well as the corresponding manual for it are presented.

3. Practical implementation of the project. In the last chapter, the software part of the device is checked and debugged.

We come to the conclusion that the application area of the demonstration stand under investigation is scientific exhibitions for children, schoolchildren and entrants.

## Содержание

Введение.....	5
1. Состояние вопроса .....	6
1.1 Формулирование цели и задачи проекта .....	6
1.2 Анализ исходных данных и известных решений.....	6
2. Проектный раздел .....	12
2.1 Разработка электрической схемы и выбор необходимых компонентов	12
2.2 Разработка конструкции устройства.....	27
2.3 Расчет магнитного поля.....	29
2.4 Разработка программной части устройства.....	29
2.5 Краткое описание устройства и инструкция по работе с ним.....	39
3. Практическая реализация проекта.....	43
3.1 Изготовление устройства.....	43
3.2 Проверка и отладка программной части устройства.....	44
Заключение .....	45
Список используемой литературы .....	46
ПриложениеА .....	49

## **Введение**

Актуальность выпускной работы заключается в простоте использования разработанного стенда. Работа со стендом не требует специальных навыков и знаний от человека, работающего с ним. Поэтому с разработанным стендом легко справятся дети, школьники и абитуриенты, пришедшие на день открытых дверей.

Объектом исследования является стенд показывающий работу шагового двигателя.

Предметом исследования является биполярный шаговый двигатель. Шагового двигатель имеет хорошие перспективы, так как использование шагового двигателя позволяет разрабатывать приложения с простым разомкнутым контуром системы контроля. В этом случае механическая приводная система имеет низкое ускорение со статическими нагрузками. Шаговый двигатель имеет преимущество производить большой крутящий момент на низкой скорости. Шаговый двигатель может сохранять крутящий момент, что способствует устойчивому положению двигателя, когда он не вращается. Удержание его положения полезно для предполагаемого применения, когда двигатель будет запущен и остановлен. В это время на двигатель действует противоположная сила. В результате будет не нужен тормозной механизм. Для шагового двигателя требуется генератор сигналов, называемый драйвером.

В разрабатываемом стенде будет возможность выполнения различных режимов работы. Это возможно осуществить с помощью использования микроконтроллера. Микроконтроллер, используемый для управления шаговым двигателем, используют в кружках по робототехнике и с недавних пор начали изучать в школах. Исходя из этого, стенд будет привлекать внимание подростков на выставках.

## **1. Состояние вопроса**

### **1.1 Формулирование цели и задачи проекта**

Цель: Разработка стенда, показывающего возможности шагового двигателя.

Задачи:

1. Изучить литературу по электронике, в которой отображена проблема исследования.
2. Рассмотреть уже существующие варианты стендов, показывающие работу шагового двигателя.
3. Разработать электрическую схему и подобрать подходящие элементы.
4. Придумать конструкцию устройства и написать программу устройства.
5. Разработать собственный электронный демонстрационный стенд.

### **1.2 Анализ исходных данных и известных решений**

Существует множество стендов, показывающих работу шагового двигателя. Все они являются лабораторными и могут выполнять работу не только с шаговым двигателем.

При рассмотрении рынка существующих аналогов стенда, было выделено несколько вариантов. Например, учебный лабораторный стенд «Классик - 1» Рисунок 1.1.



Для объекта исследования, внешние устройства могут отличаться.

Доступные внешние устройства:

1. Цифровая клавиатура;
2. 8-и и 7-и сегментный индикатор;
3. Цифровой и аналоговый вольтметры;
4. Излучатель звука;
5. FLASH и FRAM память;
6. Датчик угла поворота;
7. Жидкокристаллический индикатор 16x2;
8. Драйвер шагового двигателя;
9. Интерфейсы RS-232 и RS-485;
10. MMC/SD - карты;
11. DIP переключатель;
12. Индикаторные СИД модули;
13. Набор кнопок имитации событий;
14. Входной, выходной регистры.

На рисунке 1.2 изображен фрагмент платы стенда, на которой размещены средства для переключения и управления работой периферийных устройств микропроцессора.

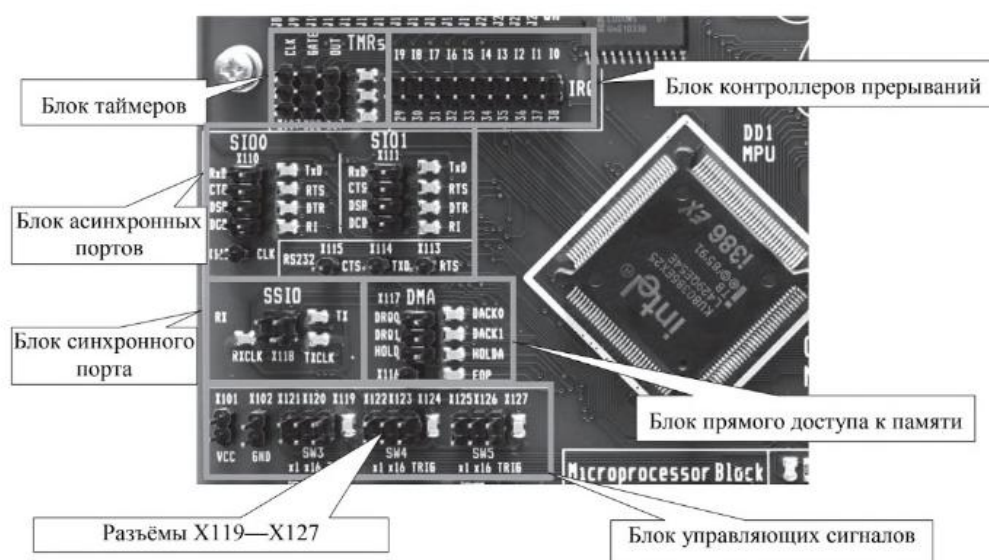


Рисунок 1.2 - Средства управления периферийными устройствами



Плюсом данного стенда является то, что работа с микропроцессором и подключенными с ним дополнительные устройства ведется без ПК. Применение этого стенда позволяет овладеть детальными умениями работы практически со всеми нынешними цифровыми технологиями.

Так как «Классик - 1» представляет собой модульный учебный комплекс, имеющий большое количество возможных опций. Он имеет высокую стоимость, что является его главным недостатком.

Второй вариант аналога ИШД-МРЦ. Он представляет собой лабораторный стенд, предназначенный для обучения студентов высших учебных заведений, изучающих дисциплины «Электрические микромашины», «Электрические машины», «Системы управления электроприводов».

#### **Технические характеристики:**

Напряжение питания	220 В.
Частота питающего напряжения	50 Гц.
Потребляемая мощность, не более	500 Вт.
Габариты	500x200x400 мм.
Масса, не более	35 кг.

Преимущество данного стенда в его многофункциональности и относительно не больших габаритах. Так же к плюсам можно отнести его комплектацию.

#### **Комплектация стенда ИШД-МРЦ:**

Моноблок системы управления.

Осциллограф.

Электромашинный агрегат.

Комплект силовых кабелей и соединительных проводов.

Техническое описание стенда.

Методические указания к проведению лабораторных работ.



Рисунок 1.3 - Лабораторный стенд ИШД-МРЦ

Перечень лабораторных работ:

1. Изучение принципа действия, режимов работы и способов коммутации фаз шагового двигателя.
2. Снятие зависимости момента двигателя от углового положения вала двигателя в различных режимах работы.
3. Исследование характеристик шагового двигателя в режиме регулирования скорости.
4. Исследование характеристик шагового двигателя в режиме регулирования положения.
5. Исследование шагового двигателя и его переходных процессов.
6. Исследование в скоростном режиме шагового двигателя.
7. Исследование в позиционном режиме работы шагового двигателя

Главным недостатком стенда является его цена 233220 рублей. Так же недостатком является его масса и габаритные размеры.

Рассмотрев пару аналогов, выяснилось, что все представленные на рынке стенды имеют высокую стоимость. В выпускной работе мы создадим электронный стенд, имеющий гораздо меньшую стоимость, массу и габариты.

## 2. Проектный раздел

### 2.1 Разработка электрической схемы и выбор необходимых компонентов

В разрабатываемом проекте управление шаговым двигателем осуществляется с помощью платформы Arduino Mega 2560 и драйвера L298n.



Рисунок 2.1 - Платформа Arduino Mega 2560

Проект Arduino был разработан в Италии в 2005 году. Как написано на их веб-сайте «Arduino - это электронная платформа с открытым исходным кодом, основанная на простом в использовании аппаратном обеспечении и программном обеспечении». Платформы Arduino были созданы для решения ряда проблем в других устройствах с микроконтроллером. Платформа Arduino вызывает большой интерес у любителей, студентов и преподавателей. Arduino представляет собой электронную платформу с открытым исходным кодом, основанную на простом в использовании аппаратном и программном обеспечении. Платы Arduino способны считывать входы - свет на датчике, палец на кнопке или сообщение Twitter.

Вы можете сообщить своей схеме, что делать, отправив набор инструкций на микроконтроллер на плате. Для этого вы используете программное обеспечение Arduino (на основе проводки) и программное обеспечение Arduino (IDE) на основе обработки. Продукты Arduino основаны на 8-битных микроконтроллерах ATmega.

Платы оснащены большим количеством цифровых и аналоговых выводов, последовательными коммуникационными модулями, USB-соединением и возможностями программирования в готовом устройстве по последовательному протоколу. Платы Arduino легко взаимодействуют с внешними компонентами для приложений по сбору и управлению данными. Все платы Arduino могут быть подключены к нескольким экранам, разработаны для разных приложений. Технические характеристики Arduino Mega 2560.

Таблица 1.1 - Технические характеристики Arduino Mega 2560.

Рабочее напряжение	5В
Входное напряжение (рекомендуемое)	7-12В
Входное напряжение (предельное)	5.5-16В
Цифровые Входы/Выходы	54 шт
Аналоговые входы	16 шт
Постоянный ток через вход/выход	40 mA
Постоянный ток для вывода 3.3 В	50 mA
Флеш-память	256 KB
ОЗУ	8 KB
Энергонезависимая память	4 KB
Тактовая частота	16 MHz
Длина	101.52 мм
Ширина	53.3 мм
Вес	37 г

Платформа Arduino была выбрана, потому что имеет преимущества, которые заключаются в низкой стоимости, целевой платформенности, простоте программирования и расширяемом программном и аппаратном обеспечении с открытым исходным кодом

Контролер шагового двигателя L298n- это специализированная интегральная схема L298n генерирует четырехфазные сигналы для двухфазного биполярного и четырехфазного униполярного шагового двигателя в приложениях с микроконтроллером. Мотор можно приводить в действие на половину шага, при нормальном и волновом режиме, а на чиповых цепях PWM-чипа позволяют переключаться в режиме управления током в обмотках. Блок-схема контроллера шагового двигателя L298n показана на рисунок 2.3.

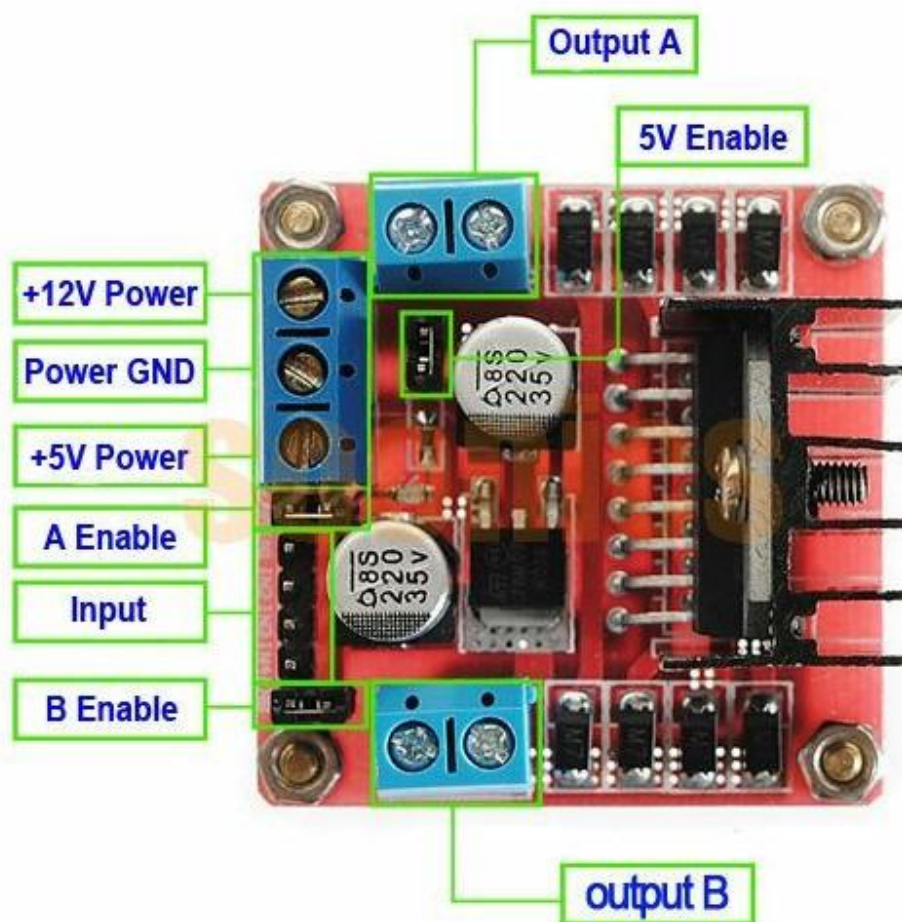


Рисунок 2.2 - Контроллер шагового двигателя L298

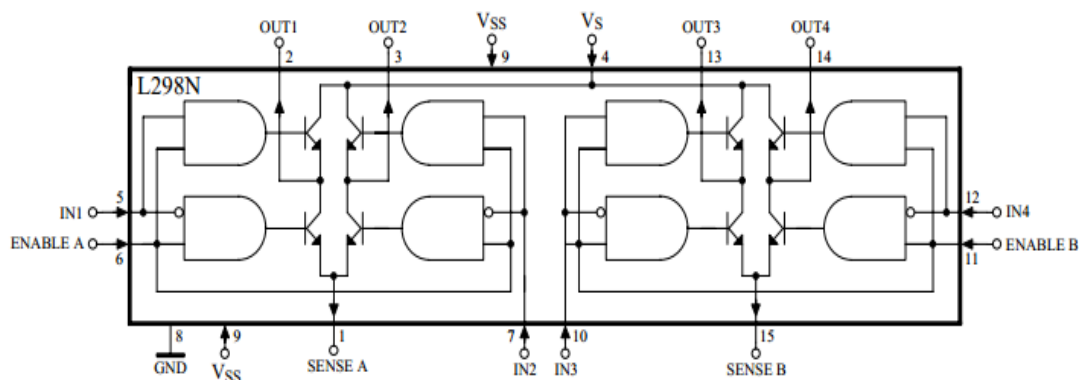


Рисунок 2.3 - Блок-схема контроллера шагового двигателя L298

Основными функциями являются транслятор, который генерирует последовательности фаз двигателя и схему двойного прерывателя ШИМ, которая регулирует ток в обмотках двигателя (SGS-Thomson, 1988). Транслятор генерирует три различные последовательности, выбранные входом HALF / FULL. Они могут быть обычными (с двухфазным напряжением), волновым приводом (с одной фазой) и полушагом (поочередно одна фаза под напряжением / две фазы под напряжением). Два сигнала подавления, INH1 и INH2 также генерируются L297 в режимах с полушагом и волновым режимом. Эти сигналы, которые передаются непосредственно к входам разрешения драйвера, предназначены для ускорения пропадания тока при обесточивании обмотки. Когда L297 используется для управления униполярным шаговым двигателем, на этих линиях действует прерыватель. Вход, называемый CONTROL, определяет, будет ли прерыватель действовать на фазовые линии A, B, C, D или блокировки INH1, INH2. Когда фазовые линии прерываются, активируется неактивная фазовая линия каждой пары (AC или BD) (Phillips, 1981). Обычный встроенный генератор управляет двойным прерыванием. Внешняя сеть RTCT (от RT до VS и CT к земле), подключенная к протоколу OSC, определяет частоту генератора прерывателя : RT должен быть более 10 кОм. Он подает импульсы на частоте прерывателя, которые устанавливают два триггера FF1 и FF2. Когда ток в обмотке достигает запрограммированного пикового значения, напряжение на резисторе чувствительности равно



опорному напряжению ( $V_{REF}$ ), и соответствующий компаратор сбрасывает его триггер, прерывая ток возбуждения до тех пор, пока не поступит следующий импульс осциллятора. Пиковый ток для обеих обмоток программируется делителем напряжения на входе опорного напряжения ( $V_{REF}$ ). Другие сигналы: - вход управления по часовой стрелке / против часовой стрелки - асинхронный сброс ввода RESET (активный низкий импульс на этом входе восстанавливает переводчик в исходное положение  $ABCD = 0101$ ); - выход HOME с открытым коллектором, который указывает, что L298m находится в исходном состоянии ( $ABCD = 0101$ ); - входной сигнал разрешения чипа ENABLE (низкий уровень на этом входе приводит к низким  $INH1$ ,  $INH2$ , A, B, C и D, отключает обмотки двигателя).

Драйвер шагового двигателя должен решать две основные задачи. Это формирование временных последовательных сигналов и обеспечения необходимого тока в обмотках.

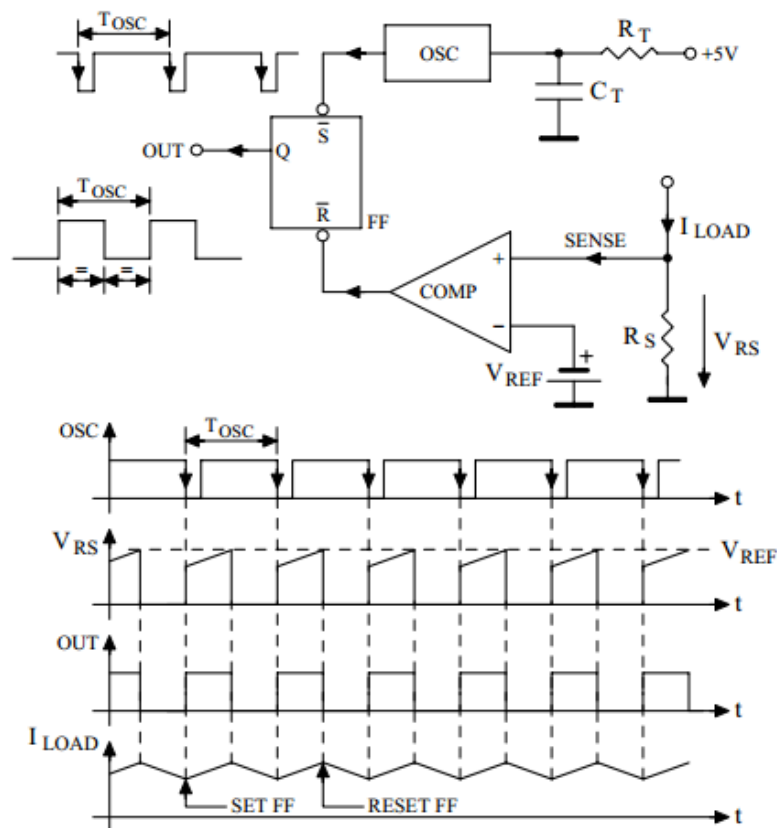


Рисунок 2.4 - Контроль тока нагрузки



Прерыватель может воздействовать либо на линии фаз (A, B, C, D), либо на линии ингибирования INH1 и INH2. Вход с именем CONTROL определяет, какой. Запрет на измельчение используется для униполярных двигателей, но вы можете выбирать между прерыванием фазы и подавлением прерывания для биполярных двигателей.

Фазовое изменение показано на рисунке 2.5. Энергия, хранящаяся в обмотке, рассеивается током, рециркулирующим через T1 и D3. Текущий спад по этому пути довольно медленный, потому что напряжение на обмотке низкое ( $V_{CEsat} T1 + V_{D3}$ ).

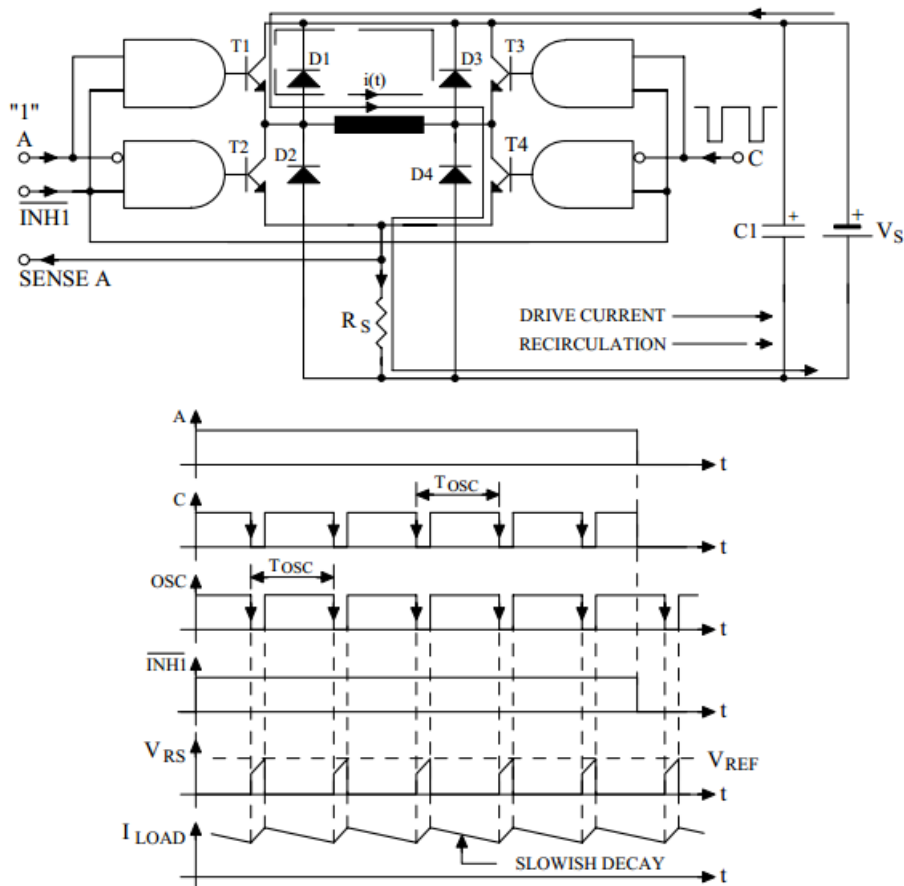


Рисунок 2.5 - Фазовое изменение

Альтернативой является привязать вход CONTROL к земле так, чтобы прерыватель действовал на INH1 и INH2. Рассматривая тот же пример, A

высокий, а С низкий. Таким образом, транзисторы T1 и T4 являются проводящими и проводят ток через T1, обмотку, T4 и RS. В этом случае напряжение на RS достигает VREF, триггер прерывания сбрасывается и активируется INH1. Сигнал INH1 отключает все четыре транзистора, поэтому ток рециркулирует из земли, через D2, обмотку и D3 в VS. Разряженный через источник питания, который может быть до 46 В, ток очень быстро падает.

Плюсами выбранного контроллера является большой диапазон напряжения подключаемых двигателей. Недостатком является отсутствие нормальной Grbl библиотеки для управления и высокая температура при работе микросхемы.

Для установки был выбран шаговый двигатель QH4-4401.



Рисунок 2.6 - Шаговый двигатель QH4-4401

QH4-4401 является биполярным двигателем с постоянным магнитом, т.е. имеет одну обмотку в каждой фазе. Для изменения направления вращения, обмотка должна менять полярность с помощью контроллера. Для такого типа двигателя требуется мостовой драйвер, или полумостовой с

двухполярным питанием. Биполярный двигатель имеет четыре вывода и две обмотки(рисунки 2.7.).



Рисунок 2.7 - Схема биполярного шагового двигателя

Шаговый двигатель можно рассматривать как цифровую электромеханическую систему, в которой, каждый электрический входной импульсный вращает ротор, на дискретный угол, называемый шагом, следовательно, отсюда и название шагового двигателя. Шаговые двигатели используются в коммерческих и промышленных целях из-за их высокой надежности, низкой стоимости, высокого крутящего момента на низких скоростях и простой конструкции, которая работает, практически, в каждой среде.

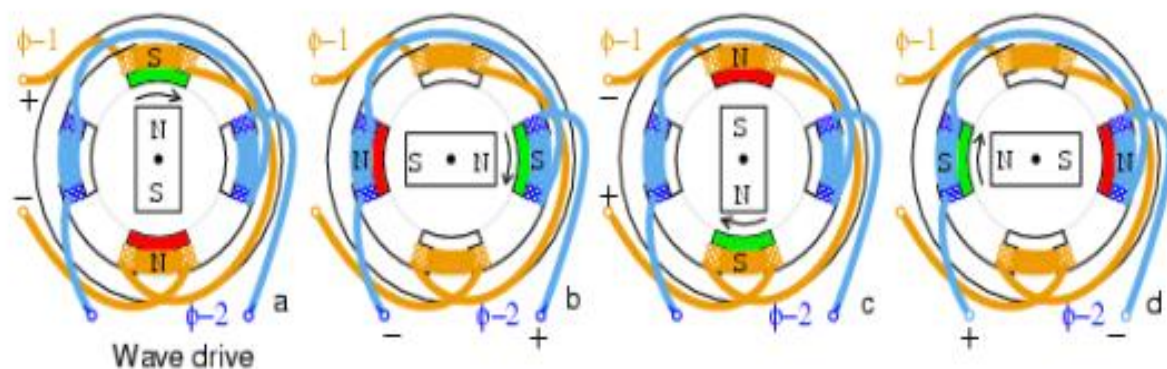


Рисунок 2.8 - Волновой способ управления шаговым двигателем

Концептуально простейшим является волновой метод показанный на рисунке 2.8. Слева на справа находится последовательность вращения положительные  $\phi-1$  точки ротора северный полюс вверх, (+)  $\phi-2$  точки ротор северный правый, отрицательный  $\phi-1$  привлекает ротор на север вниз, (-)  $\phi-2$  точки ротора влево. Волновой режим ниже как показано, что только одна

катушка под напряжением за один раз. Хотя это простой способ управления, но это не производят столько крутящего момента, как другие методы.

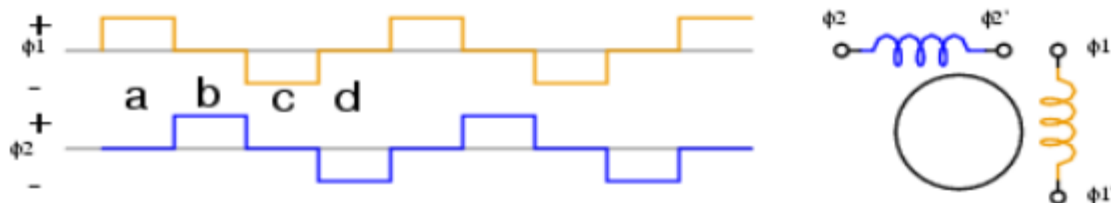


Рисунок 2.9 - Биполярные сигналы в волновом режиме

Сигналы, показанные на рисунке 2.9, являются биполярными, поскольку обе полярности, (+) и (-) управляют шаговым двигателем. Катушка меняет магнитное поле, поскольку полярность тока привода меняется. Полный привод обеспечивает больше чем волновой привод, поскольку оба катушки в то же время. Это привлекает ротор полюсов на полпути между двумя полями

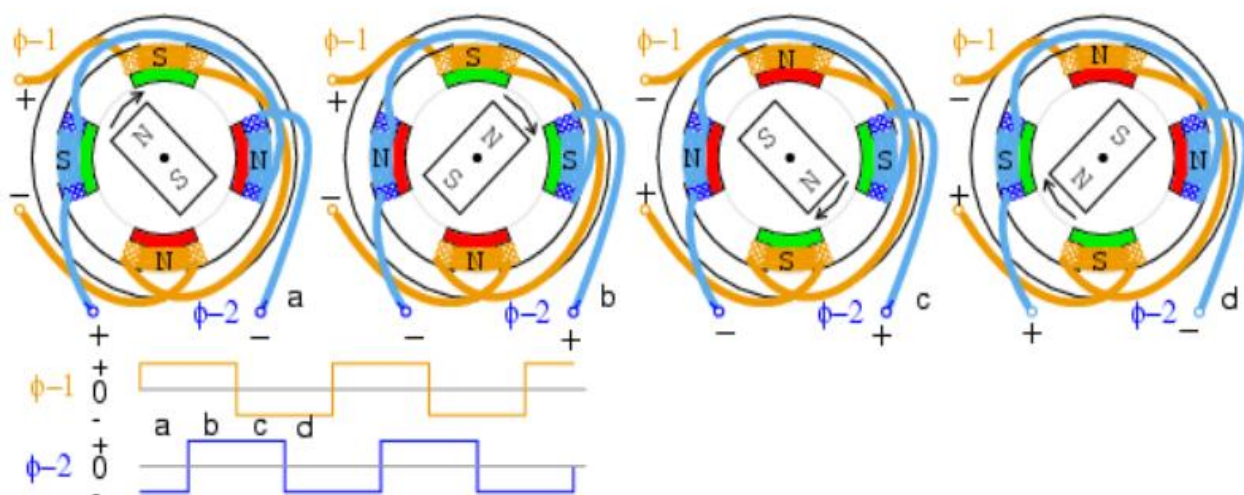


Рисунок 2.10 - Шаговый режим шагового двигателя

Режим полного шага биполярного двигателя, как показано на рисунке 2.10, имеет тот же шаг, что и волновой режим. Дополнительная подключенная обмотка биполярного привода оправдана, когда требуется больше крутящего момента. Угол шага для данной геометрии шагового

двигателя разделяется пополам с помощью полушагового режима. Это соответствует удвоенному количеству импульсов на оборот. На рисунке 2.11 показано, что половина ступеней обеспечивает большее количество устойчивых положений вала двигателя. Например, полушаговый режим двигателя, позволяет перемещать печатающую головку через бумагу струйного принтера, с удвоенной плотностью точек.

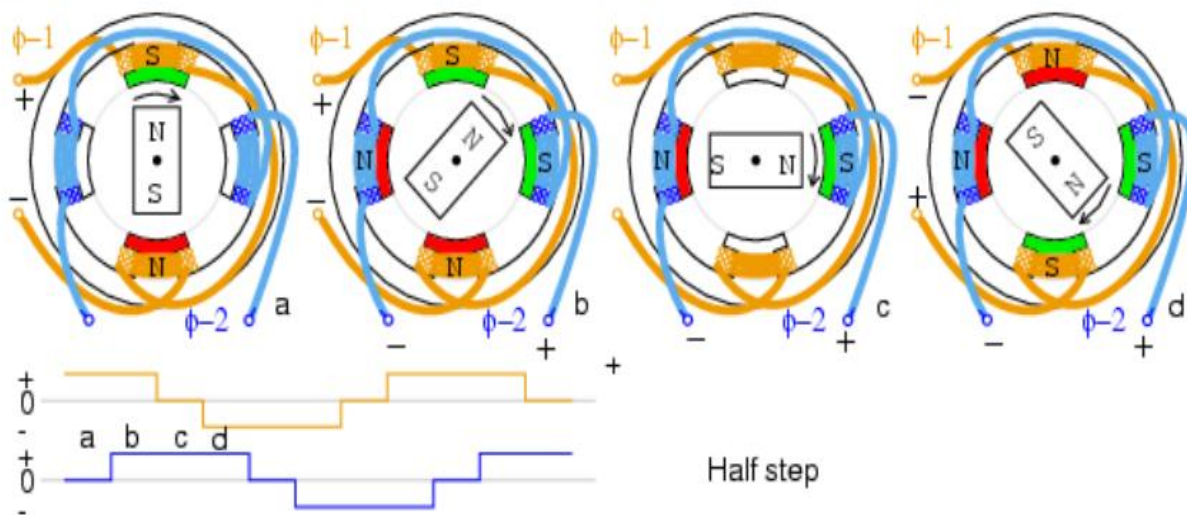


Рисунок 2.11 - Полушаговый режим шагового двигателя

Полушаговый режим представляет собой комбинацию волнового режима и шагового режима с одной подключенной обмоткой, за которой следуют две обмотки, подключенные вместе. Ротор выравнивается с полюсами поля, как для волнового режима, так и между полюсами, как для полного шагового режима.

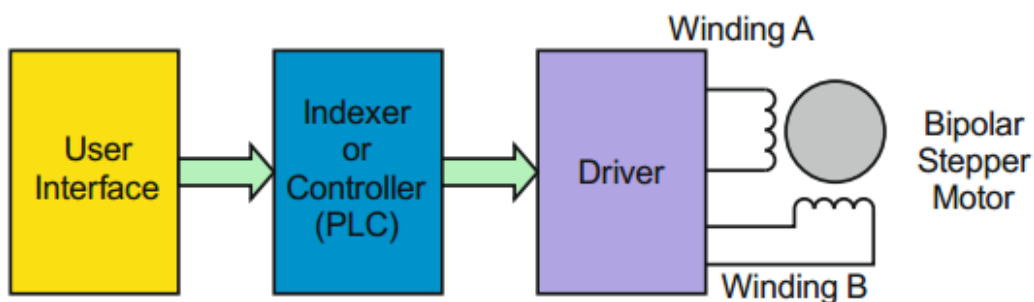


Рисунок 2.12 - Блок-схема имитационной модели шагового двигателя

На рисунке 2.12 представлена блок-схема имитационной модели шагового двигателя с постоянным магнитом, построенной из базовых блоков: контроллера, драйвера и ведущего двигателя.

Переключение режимов и управление в ручном режиме в стенде осуществляется с помощью тактовой кнопки



Рисунок 2.13 - Тактовая кнопка

В разрабатываемом стенде для визуализации вращения двигателя используется диск из органического стекла. Изначально материалом для создания диска являлся ДВП (древесноволокнистая плита). Выбор ДВП был обоснован его легкостью механической обработки и ценой. Но после экспериментального запуска стенда было выявлен его главный недостаток. Он не обладал достаточной прочностью поверхности. По краям диска происходило расслаивание. Заменой диска из древесноволокнистой плиты, стал диск из органического стекла.



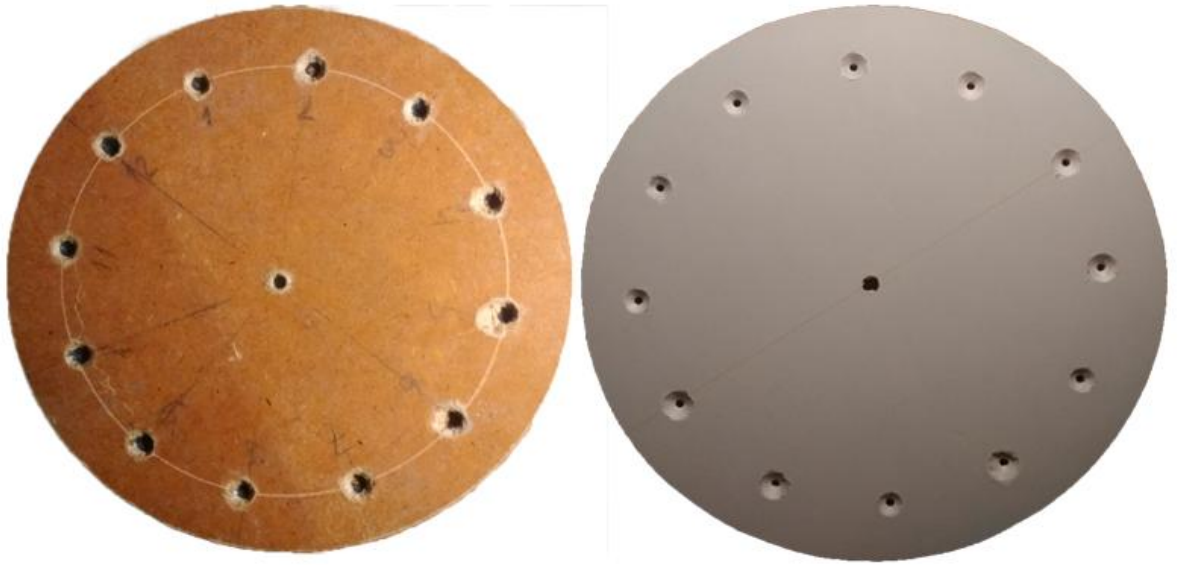


Рисунок 2.14 -Диск из ДВП и оргстекла

Поверхность, диска из оргстекла является более гладкой и плотной. В диске просверлены двенадцать отверстий для устойчивого размещения металлического шара. Стальной шар можно перемещать в ручном режиме с помощью электромагнита. Магнит подключен по схеме представленной на рисунке 2.15.

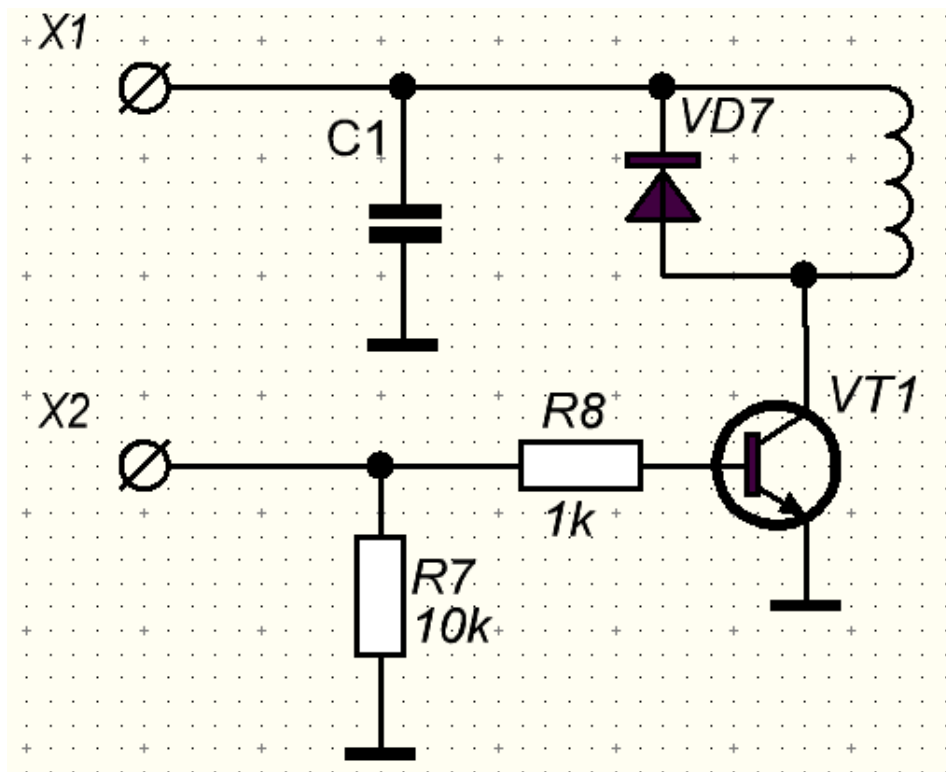


Рисунок 2.15 - Схема подключения электромагнита

В данной схеме вывод 1 подключается к питанию, а вывод 2 подключается к выводу Arduino и программно открывает или закрывает ключ. В качестве ключа используется кремниевый n-p-n транзистор КТ817 выполненный в корпусе TO126.

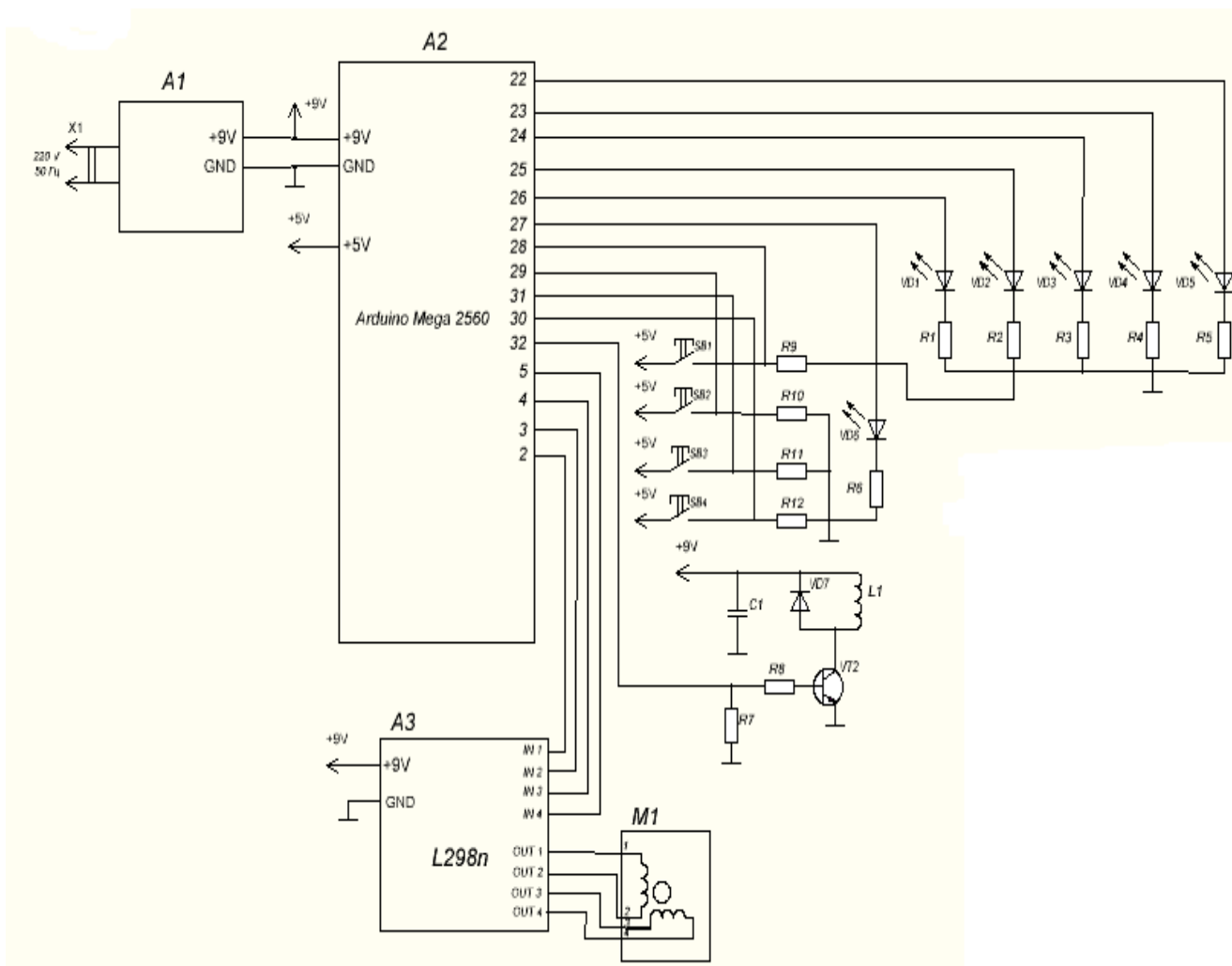


Рисунок 2.16 - Схема электрическая принципиальная

Перечень выбранных элементов представлен в таблице 2.1.



Таблица 2.1 - Перечень элементов

Дет. обознач.	Наименование	Кол.	Примечание
<i>Устройства</i>			
A1	Блок питания	1	
A2	Плата Arduino	1	
A3	Драйвер L298n	1	
<i>Конденсатор</i>			
C1	ЕСАР (К50-35), 220 мкФ, 25 В	1	
<i>Электромагнит</i>			
L1	Электромагнит	1	
<i>Двигатель</i>			
M1	QH4-4401	1	
<i>Резисторы</i>			
R1-R6	МО-50 (С2-23) 0.5 Вт, 300 Ом, 5%	6	
R7,R9-R12	MF-25 (С2-23) 0.25 Вт, 10 кОм, 1%	4	
R8	CF-25 (С1-4) 0.25 Вт, 1 кОм, 5%	1	
<i>Тактовая кнопка</i>			
SB1-SB4	KLS7-TS6601-13.0-180	4	
<i>Диоды</i>			
VD1-VD6	GNL-3014GT	6	
VD7	КД522	1	
<i>Транзистор</i>			
VT1	КТ817	1	



Рисунок 2.17 - Транзистор КТ817 в корпусе ТО126

Для удобства сборки схемы используется макетная плата

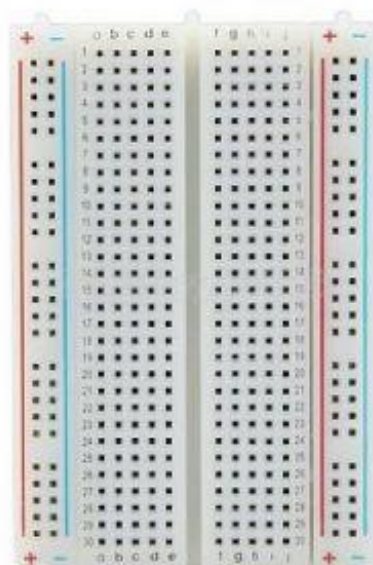


Рисунок 2.18 - Макетная плата

Схема подключения кнопки и светодиода к Arduino представлена на рисунке 2.18. К гнезду X3 подключается питание 5В, X1 и X2 подключается к контролеру. Резистор R1 является подтягивающим. Он необходим для задания известного состояния порту контроллера. Резистор R2 является токоограничивающим для светодиода.

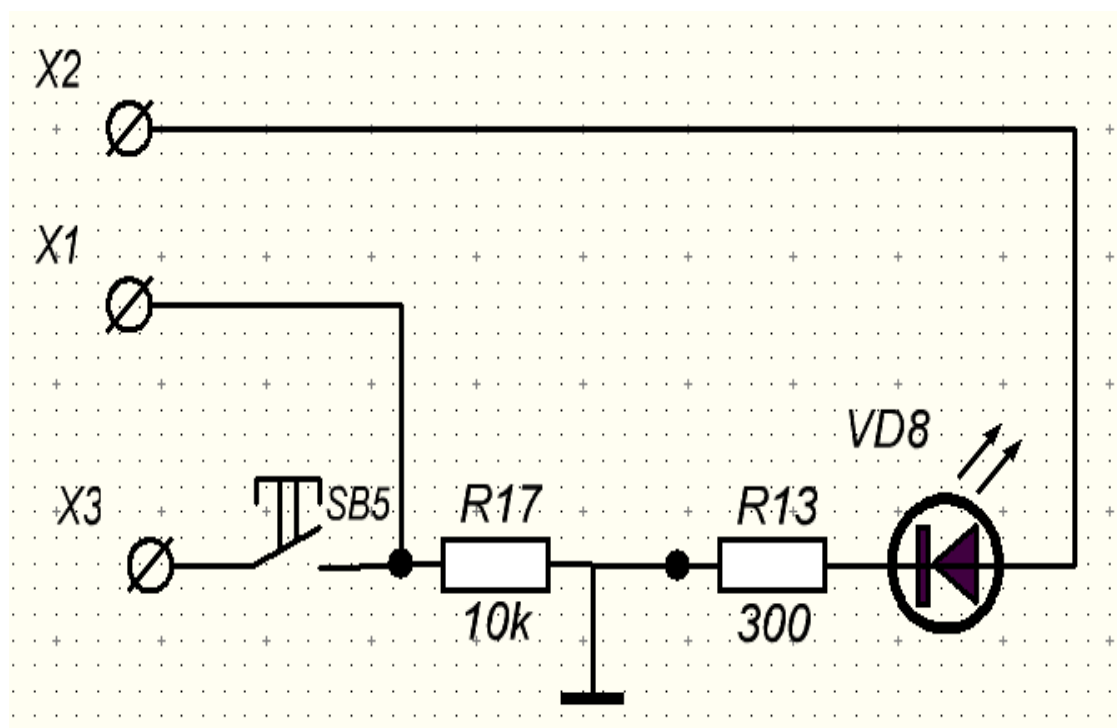


Рисунок 2.19 - Схема подключения кнопки и светодиода к Arduino

## 2.2 Разработка конструкции устройства.

Конструкция стенда представляет собой деревянную площадку, с прикрученным к ней шаговым двигателем, на вал которого надет диск с отверстиями. Электромагнит прикреплен к деревянной планке, которая надета на металлическую шпильку, вставленную в площадку. Планка регулируется по высоте с помощью болтов.

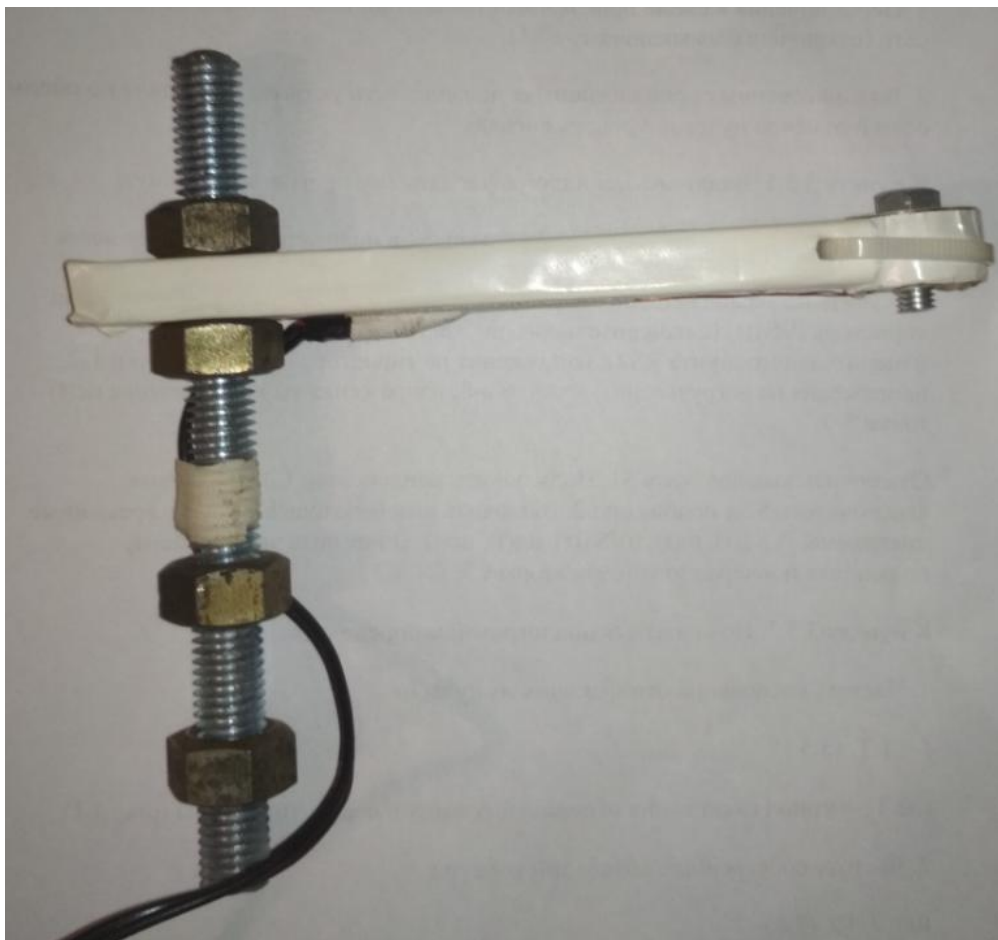


Рисунок 2.20 - Устройство крепления электромагнита



Рисунок 2.21 - Разрабатываемый стенд(Вид сверху)

Так же на площадке расположен пульт для изменения режима шагового двигателя, управления ручным режимом и включение/выключение электромагнита. Пульт крепится четырьмя болтами к основанию. Индикация режимов осуществляется с помощью светодиодов.



Рисунок 2.22 - Светодиод 5мм

Платформа Arduino, драйвер для шагового двигателя и схема подключения электромагнита расположены под площадкой. Провода необходимые для питания и управления электроникой проходят через отверстия в площадке.

### 2.3 Расчет магнитного поля

Электромагнит выполнен в виде катушки, намотанной на нее медный одножильный эмалированный провод. В центре катушке расположен стальной сердечник из стали. Диаметр провода 0.27мм., средний диаметр катушки равен 12мм. Удельное электросопротивление меди равно  $1.75e-8$ . Зная все эти параметры, можно рассчитать количество витков, для создания электромагнита с подходящей магнитной силой . Для электромагнита на ток 1А, сопротивление катушки должно быть 9 Ом, это 724 витка. Моделирование электромагнита проводилось в программе Elcut.

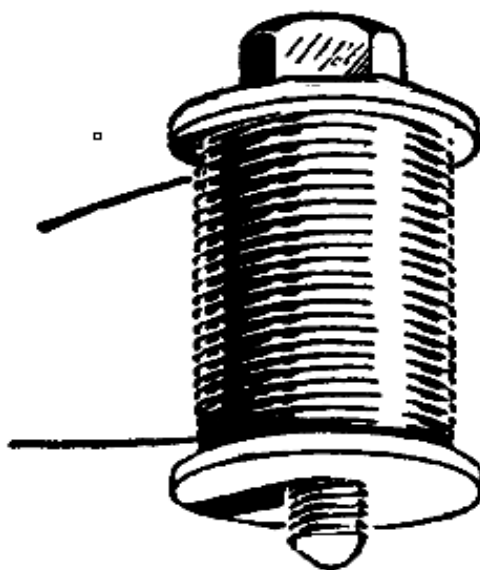


Рисунок 2.23 - Электромагнит

### 2.4 Разработка программной части устройства

Для работы платы Arduino требуется специальная среда разработки. Среда разработки Arduino состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста(консоли),

панели инструментов с кнопками часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino. Программа написанная в среде ардуино называется скетч.

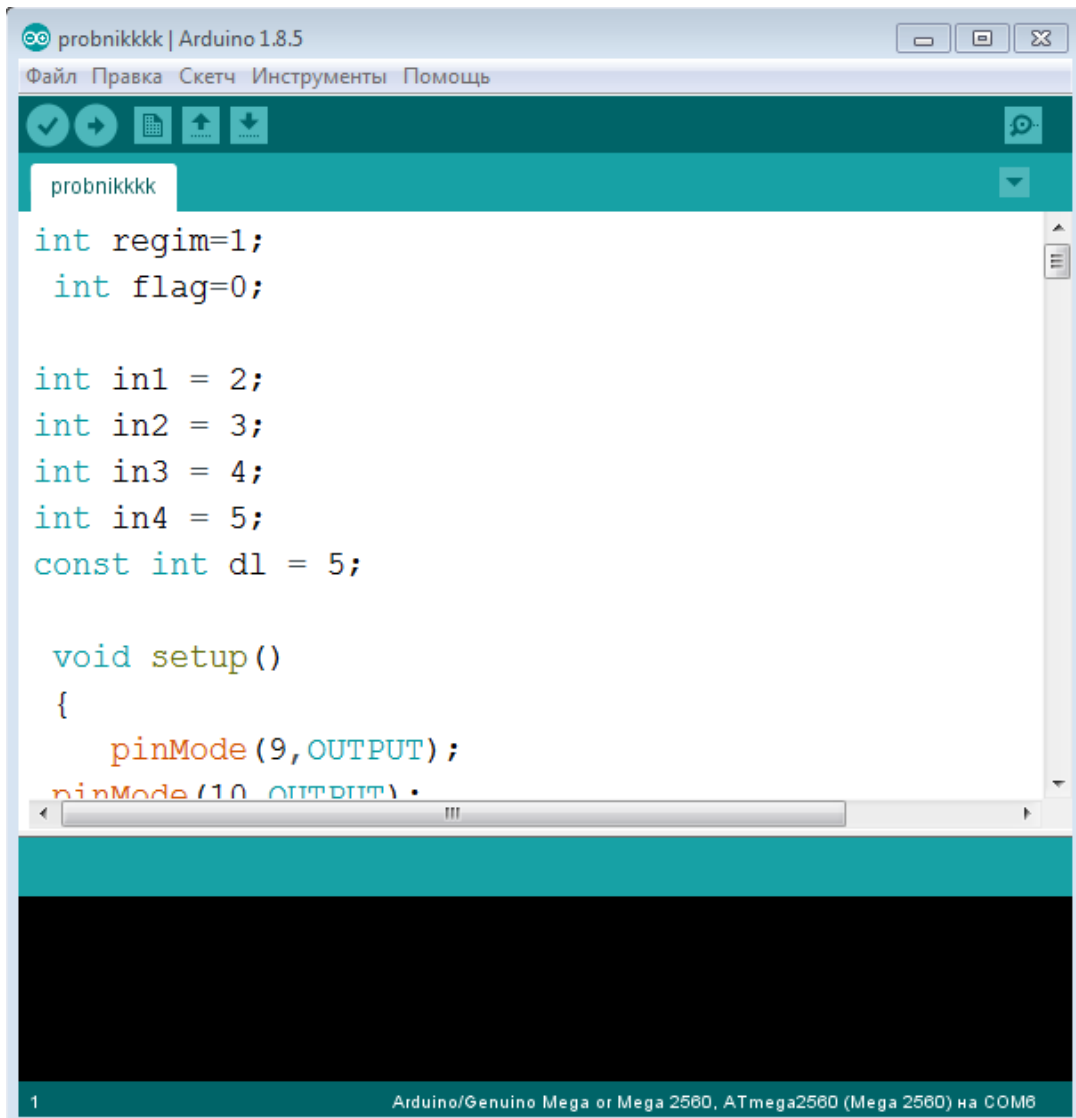


Рисунок 2.24 - Среда разработки Arduino

Написанная программа, по нажатию контактной кнопки, выполняет пять режимов. В овсех режимах используется стандартная библиотека «Stepper.h». Библиотека «Stepper.h» предоставляет простой и понятный интерфейс для использования шагового двигателя. Функция **constintstepsPerRevolution** показывает сколько шагов должен сделать двигатель, чтобы его ротор прокрутился на 360 градусов. Функция

**myStepper.setSpeed(70)** устанавливает скорость вращения вала двигателя. Функция **myStepper.step** говорит программе сколько нужно сделать шагов и в какую сторону.

В начале программы объявляем переменные и постоянные.

```
#include <Stepper.h>
int flag1=0;
int flag2=0;
int flag3=0;
int regim=1;
  int flag=0;
  long randNumber;
  //Кол-во шагов двигателя для оборота на 360
const int stepsPerRevolution = 48;
//подключение к пинам на Ардуино
Stepper myStepper(stepsPerRevolution, 3,5,2,4);
int Mag = 32;
int in1 = 2;
int in2 = 3;
int in3 = 4;
int in4 = 5;
const int d2 = 100;
void setup()
{
  pinMode(22,OUTPUT);
pinMode(23,OUTPUT);
  pinMode(24,OUTPUT);
pinMode(25,OUTPUT);
  pinMode(26,OUTPUT);
pinMode(27,OUTPUT);
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
randomSeed(analogRead(0));
pinMode(Mag, OUTPUT);
myStepper.setSpeed(15); //установка скорости вращения ротора
Serial.begin(9600);

}
```

Рисунок 2.25 - Постоянные и переменные программы

```

if(digitalRead(28)==HIGH&&flag==0)//если кнопка нажата
// и переменная flag равна 0 , то ...
{
    regim++;
    flag=1;

    //это нужно для того что бы с каждым нажатием кнопки
    //происходило только одно действие
    // плюс защита от "дребезга" 100%
    if(regim>5)//ограничим количество режимов
    {
        regim=1;//так как мы используем только одну кнопку,
        // то переключать режимы будем циклично
    }
}
if(digitalRead(28)==LOW&&flag==1)//если кнопка НЕ нажата
//и переменная flag равна - 1 ,то ...
{
    flag=0;//обнуляем переменную "кнопка"
}
if(regim==1) //первый режим
{
    digitalWrite(22,HIGH);//включаем 1 светодиода
    digitalWrite(23,LOW);
    digitalWrite(24,LOW);
    digitalWrite(25,LOW);
    digitalWrite(26,LOW);
    digitalWrite(27,LOW);
digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    digitalWrite( 32, LOW );
    digitalWrite(22,LOW);
}

```

Рисунок 2.26 - Программа первого режима



Первый режим включается при подаче питания на плату Arduino. Загорается первый светодиод и включается режим ожидания. Напряжение на обмотки шагового двигателя не подается и индикаторные светодиоды не горят. Функция digitalWrite показывает в каком состоянии находятся обмотка двигателя и светодиоды. HIGH-светодиод горит, через обмотку проходит ток. LOW- светодиод не горит, через обмотку не проходит ток. Программа первого режима представлена на рисунке 2.26.

Второй режим выполняет выключение первого светодиода и включение второго. В этом режиме шары последовательно перемещаются по часовой стрелке. При включении третьего режима гаснет первый светодиод и загорается второй. Программа второго режима представлена на рисунке 2.27.

В момент включения третьего режима гаснет второй светодиод и загорается третий. В этом режиме диск вращается на случайное количество шагов по часовой и против часовой стрелки. Диапазон этих шагов от -48 до 48.

```

if(digitalRead(30)==HIGH&&flag3==0)//если кнопка нажата
// и переменная flag равна 0 , то ...
{
    flag3=1;
    //это нужно для того что бы с каждым нажатием кнопки
    //происходило только одно действие
    // плюс защита от "дребезга" 100%
}
if(digitalRead(30)==LOW&&flag3==1)//если кнопка НЕ нажата
//и переменная flag3 равна - 1 ,то ...
{
    digitalWrite(32,HIGH);
    delay(400);
myStepper.step(-4);
digitalWrite(32,HIGH);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
myStepper.step(8);
    delay(800);
    digitalWrite(32,HIGH);
    delay(400);
myStepper.step(-4);
delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(8);
    delay(800);
digitalWrite(32,HIGH);
delay(400);
    myStepper.step(-4);
    delay(800);
digitalWrite(32,LOW);
delay(400);
    myStepper.step(-8);
    delay(800);
    flag3=0;//обнуляем переменную flag3
}

delay(300);

digitalWrite(23,LOW);

```

Рисунок 2.27 - Программа второго режима

```

        if(regim==3) //третий режим
        {
            digitalWrite(24,HIGH); //включается 3 светодиод
digitalWrite(in1, LOW);
            digitalWrite(in2, LOW);
            digitalWrite(in3, LOW);
            digitalWrite(in4, LOW);
            if(digitalRead(30)==HIGH&&flag3==0) //если кнопка нажата
                // и переменная flag равна 0 , то ...
            {
                flag3=1;
                //это нужно для того что бы с каждым нажатием кнопки
                //происходило только одно действие
                // плюс защита от "дребезга" 100%
            }
            if(digitalRead(30)==LOW&&flag3==1) //если кнопка НЕ нажата
                //и переменная flag3 равна - 1 ,то ...
            {
                // выводим случайное число в диапазоне от -48 до 48
                randomNumber = random(-48, 49);
                myStepper.step(randomNumber);
                delay(1000);

                flag3=0; //обнуляем переменную flag3
            }

            delay(300);

            digitalWrite(24,LOW);

```

Рисунок 2.28- Программа третьего режима

При включении четвертого режима гаснет третий светодиод, загорается четвертый. В этом режиме первый шарик перемещается против часовой стрелки, а второй по часовой. Каждый шарик после пяти перемещений возвращается в исходную точку.

```

if(regim==4)
{
    digitalWrite(25,HIGH); //включаем 6 светодиод
}

if(digitalRead(30)==HIGH&&flag3==0) //если кнопка нажата
// и переменная flag равна 0 , то ...
{
    flag3=1;
    //это нужно для того что бы с каждым нажатием кнопки
    //происходило только одно действие
    // плюс защита от "дребезга" 100%
}
if(digitalRead(30)==LOW&&flag3==1) //если кнопка НЕ нажата
//и переменная flag3 равна - 1 ,то ...
{
    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(4);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(-12);
    delay(800);

    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(-4);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(16);
    delay(800);

    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(4);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(-20);
    delay(800);

    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(40);
    delay(800);

    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(-4);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(24);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(-24);
    delay(800);

    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(4);
    delay(800);
    digitalWrite(32,HIGH);
    delay(1000);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(16);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(8);
    delay(800);
    flag3=0; //обнуляем переменную flag3
}

    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(-4);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(32);
    delay(800);
    delay(300);
    digitalWrite(25,LOW);

    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(4);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
    myStepper.step(-36);
    delay(800);

    digitalWrite(32,HIGH);
    delay(1000);
    myStepper.step(-4);
    delay(800);
    digitalWrite(32,LOW);
    delay(600);
}

```

Рисунок 2.29 - Программа четвертого режима.

При включении пятого режима загорается пятый светодиод и тухнет четвертый. В этом режиме используются три дополнительные кнопки. При нажатии на первую кнопку ротор двигателя выполняет поворот против часовой стрелки на четыре шага.

```

if(digitalRead(28)==HIGH&&flag2==0)//если кнопка нажата
  // и переменная flag2 равна 0 , то ...
  {
    digitalWrite( in1, HIGH ); //против часовой
digitalWrite( in2, HIGH );
digitalWrite( in3, LOW );
digitalWrite( in4, LOW );
delay(d2);

digitalWrite( in1, LOW );
digitalWrite( in2, HIGH );
digitalWrite( in3, HIGH );
digitalWrite( in4, LOW );
delay(d2);

digitalWrite( in1, LOW );
digitalWrite( in2, LOW );
digitalWrite( in3, HIGH );
digitalWrite( in4, HIGH );
delay(d2);

digitalWrite( in1, HIGH );
digitalWrite( in2, LOW );
digitalWrite( in3, LOW );
digitalWrite( in4, HIGH );
delay(d2);
  flag2=1;
  //это нужно для того что бы с каждым нажатием кнопки
  //происходило только одно действие
  // плюс защита от "дребезга" 100%
  }

```

Рисунок 2.30 - Программа для поворота ротора против часовой стрелки на 4 шага

При нажатии на третью кнопку ротор двигателя выполняет поворот по часовой стрелке на четыре шага. В выбранном режиме возможно

```
if(digitalRead(26)==HIGH&&flag1==0)//если кнопка нажата
  // и переменная flag1 равна 0 , то ...
  {
    digitalWrite( in1, HIGH ); //ПО ЧАСОВОЙ
digitalWrite( in2, LOW );
digitalWrite( in3, LOW );
digitalWrite( in4, HIGH );
delay(d2);

digitalWrite( in1, LOW );
digitalWrite( in2, LOW );
digitalWrite( in3, HIGH );
digitalWrite( in4, HIGH );
delay(d2);

digitalWrite( in1, LOW );
digitalWrite( in2, HIGH );
digitalWrite( in3, HIGH );
digitalWrite( in4, LOW );
delay(d2);

digitalWrite( in1, HIGH );
digitalWrite( in2, HIGH );
digitalWrite( in3, LOW );
digitalWrite( in4, LOW );
delay(d2);
flag1=1;
//это нужно для того что бы с каждым нажатием кнопки
//происходило только одно действие
// плюс защита от "дребезга" 100%
  }
```

Рисунок 2.31 - Программа для поворота ротора по часовой стрелки на 4 шага

При нажатии на вторую контактную кнопку, включается электромагнит. При повторном нажатии магнит отключается.

```

if(digitalRead(30)==HIGH&&flag3==0)//если кнопка нажата
// и переменная flag равна 0 , то ...
{
    flag3=1;
    //это нужно для того что бы с каждым нажатием кнопки
    //происходило только одно действие
    // плюс защита от "дребезга" 100%
}
if(digitalRead(30)==LOW&&flag3==1)//если кнопка НЕ нажата
//и переменная flag3 равна - 1 ,то ...
{
    digitalWrite(32,!digitalRead(32));
    flag3=0;//обнуляем переменную flag3
}

```

Рисунок 2.32- Программа для включения электромагнита

При последующих нажатиях на кнопку «Режим», программа возвращается к первому режиму и повторяет цикл.

## 2.5 Краткое описание устройства и инструкция по работе с ним

Стенд представляет собой электронное устройство, работающее от сети 220В через адаптер.



Рисунок 2.33 - Адаптер 220В

Постоянное выпрямленное напряжение питает плату Arduino. При подаче питания на плату, включается первый режим работы. Подробное описание стенда представлено в пункте 2.2.

Руководство по эксплуатации электронного демонстрационного стенда.

Подключить стенд к сети. На пульте управления выбрать нужный режим нажатием кнопки «Режим». После подключения питания, автоматически включается режим ожидания.

При однократном нажатии на кнопку «Режим» выбирается режим «Змейка». В этом режиме используется три шарика. Поставить шарики в положения 12,11,10. Магнит должен находиться над положением 12. При нажатии на «Выбор» запускается выбранный режим. Для переключения режима, необходимо дождаться завершения цикла и только после этого нажать на кнопку «Режим».





Рисунок 2.34 - Блок управления

При втором нажатии на кнопку «Режим», включается режим «Рандом». После переключения режима, для начала его работы, необходимо нажать на кнопку «Выбор». Для переключения режима, необходимо дождаться завершения цикла и только после этого нажать на кнопку «Режим».

При третьем нажатии на кнопку «Режим», включается режим «Капельки». В этом режиме используется два шарика. Их необходимо поставить в положения 1 и 11. Магнит должен находиться над положением 11. Далее необходимо нажать на кнопку «Выбор», для запуска выбранного режима. Для переключения режима, необходимо дождаться завершения цикла и только после этого нажать на кнопку «Режим».

При третьем нажатии на кнопку «Режим», включается режим «Ручной». В этом режиме используется один шарик и три дополнительные кнопки: «По часовой», «Против часовой», «Магнит». Шарик можно расположить в любом положении. Магнит должен находиться над шариком. Для переноса шарика в другое положение, необходимо в момент, когда магнит находится над шариком нажать кнопку «Магнит». После этого прокрутить диск по часовой или против часовой стрелки, нажатием кнопки «По часовой» или «Против часовой».

При пятом нажатии на кнопку «Режим», включается режим ожидания.

### 3. Практическая реализация проекта.

#### 3.1 Изготовление устройства.

При реализации устройства, было решено перекрасить его в белый цвет. Диск из органического стекла, планка для крепления электромагнита и внешняя часть площадки, где установлен электромагнит, обклеены белой самоклеющейся бумагой. Более подробная конструкция проекта представлена в пункте 2.2.



Рисунок 3.1 - Разработка стенда

Во время сборки устройства возникла проблема с вылетом соединительных проводов из макетной платы и контроллера Arduino. Для устранения данной проблемы, было решено закрепить макетную плату и микроконтроллер с помощью термоклея к основанию.

Для транспортировки стенда, была склеена коробка с нужными геометрическими размерами.

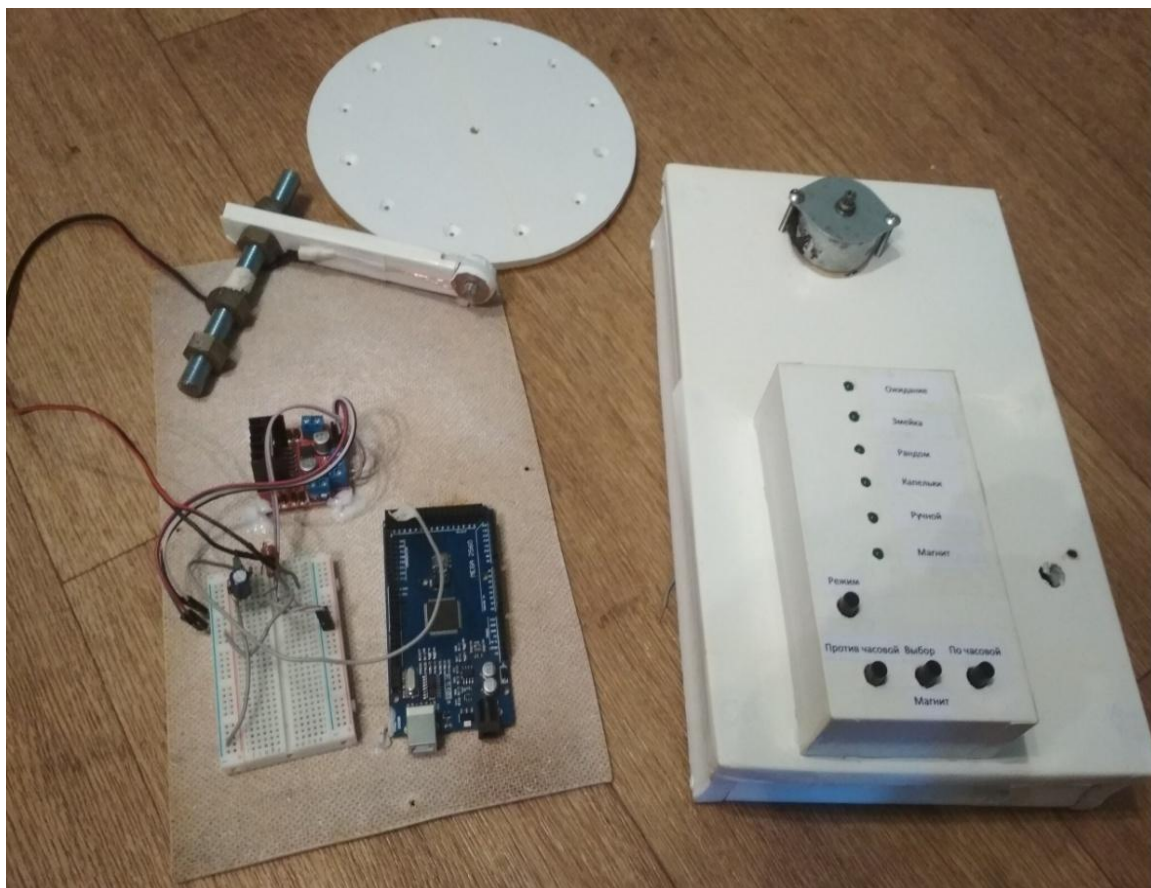


Рисунок 3.2 - Окончательный вариант стенда (в разобранном виде)

### **3.2 Проверка и отладка программной части устройства.**

После проверки программы, исправлены некоторые недочеты, объединены программы. В программе дописаны комментарии, для того чтобы любой пользователь не писавший программу смог легко в ней разобраться. Выходы и входы платы Arduino подобраны так, чтобы была возможность управлять шаговым двигателем с помощью платы Arduino UNO. Написанная программа работает стабильно. Подробное описание программы написано в пункте 2.3. Программа представлена в Приложении А.

## **Заключение**

В ходе выполнения бакалаврской работы была изучена литература по данной теме. Изучен рынок аналогов стендов с использованием шагового двигателя. Была разработана электрическая схема и подобраны подходящие элементы. Подобраны материалы и собрана конструкция устройства. Написана программа для микроконтроллера на языке программирования C/C++. Пошагово описан процесс выполнения программы. Был рассчитан и намотан электромагнит. Разработана система, позволяющая управлять шаговым двигателем и электромагнитом. Решены задачи, позволяющие достичь поставленной цели.

Цель работы заключалась в создании электронного демонстрационного стенда с использованием шагового двигателя.

В результате выполнения выпускной работы, был создан электронный демонстрационный стенд с использованием шагового двигателя. Разработанный стенд имеет меньшую стоимость, нежели его аналоги.

## Список используемой литературы

1. Хоровиц, Хилл : Искусство схемотехники. М.: Издательство БИНОМ 2014. -704 с.,ил.
2. РевичЮ.В. ; Занимательная электроника. Издательство: ВHV 2018 г. -672 с.,
3. Улли Соммер ; Программирование микроконтроллерных плат Arduino/Freeduino: Издательство БХВ-Петербург 2016 г. -256 с.
4. Виктор Петин; Просты с использованием контроллера Arduino: Издательство БХВ-Петербург 2015 г. -448 с.
5. А. П. Кашкаров.: Занимательная электроника: нешаблонная энциклопедия полезных схем; Феникс 2015 г. 230 с.
6. Электрические конденсаторы и конденсаторные установки: Справочник / Под общ. ред. Г.С.Кучинского. – М.: Энергоатомиздат, 1987.
7. Полупроводниковые приборы. Диоды выпрямительные, стабилитроны, тиристоры: Справочник – 2-е изд. стереотип. /А. Б. Гитцевич, А. А. Зайцев, В. В. Мокряков и др. Под ред. А. В. Голомедова. – М.: Радио и связь. КУБК-а, 1994. – 528с., ил.
8. Полупроводниковые приборы. Транзисторы большой и средней мощности: Справочник– 2-е изд. стереотип./ А. А. Зайцев, А. И. Миркин., В. В. Мокряков и др. Под ред. А. В. Голомедова. – М.: - Радио и связь. КУБК-а, 1994. – 640с., ил.
9. Гусев В.Г., Гусев .М. Электроника: Учебник для вузов. – 2-е изд., перераб. и доп. – М.: Высш. школа, 1991. – 622 с., ил.
10. Осваиваем Arduino[Электронный ресурс]. URL: <http://www.radiomagic.ru/arduino-projects>, (дата обращения: 27.05.2018)
11. Каталог электронных компонентов [Электронный ресурс]. URL: [http:// www.chip-dip.ru](http://www.chip-dip.ru) (дата обращения: 23.05.2018)
12. Все проекты с Arduino [Электронный ресурс]. URL: <http://arduinoprojects.ru/projects/cool/> (дата обращения: 23.05.2018)

13. Подключение кнопки к Arduino[Электронный ресурс]. URL: <http://arduino.ru/> (дата обращения: 23.05.2018)
14. Шаговые двигатели и моторы Ардуино[Электронный ресурс]. URL: <https://arduinomaster.ru/> (дата обращения: 23.05.2018)
15. Драйвер шагового двигателя и двигателя постоянного тока L298n [Электронный ресурс]. URL: <http://arduino-diy.com/> (дата обращения: 23.05.2018)
16. Амперка[Электронный ресурс]. URL: <http://amperka.ru/> (дата обращения: 22.05.2018)
17. Программирование Arduino [Электронный ресурс]. URL: <http://arduino.ru/Reference> (дата обращения: 23.05.2018)
18. Блок-схемы алгоритмов [Электронный ресурс]. URL: <https://pro-prof.com/archives/1462> (дата обращения: 25.05.2018)
19. Шаговый двигатель [Электронный ресурс]. URL: <http://engineering-solutions.ru/motorcontrol/stepper/> (дата обращения: 21.05.2018)
20. Arduino[Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Arduino> (дата обращения: 24.05.2018)
21. Patrick T. Murphy, Evolution Motor Control Design, Design News, Vol. 64, Issue 7, Jul 1, 2009, p. 14.
22. Subrata Cahattopadhyay, Utpal Chakraborty, Arindan Bhakta and Sagarika Pal, Microcontroller Based Closed Loop PMDC Motor Position Control System, Sensors & Transducers, Vol. 102, Issue 3, March 2009, pp. 62-70.
23. Scientific Bulletin of the „Petru Maior” University of Tîrgu Mureş Vol. 11 (XXVIII) no. 2, 2014 ISSN-L 1841-9267 (Print), ISSN 2285-438X (Online), ISSN 2286-3184 (CD-ROM)
24. THE ANNALS OF ”DUNAREA DE JOS” UNIVERSITY OF GALATI FASCICLE III, 2004 ISSN 1221-454X

25. T. Chandrasekhar, Nagabhushan K Raju, V.V.C.H. Ramana, NagabushanaKatte, Mani C Kuma, Embedded Based DC Motor Speed Control System, Sensors & Transducers, Vol. 121, Issue 10, Oct. 2010, pp. 95-104.

26. S. A. Prasad, B. S. Kariyappa, R. Nagary, S. K. Thakur, Microcontroller Based AC Power Controller, Wireless Sensor Network, Vol. 1, Issue 2, Jul 2009, pp. 76-81.



## Приложение А

Приложение А

```
#include<Stepper.h>
int flag1=0;
int flag2=0;
int flag3=0;
int regim=1;
int flag=0;
longrandNumber;
//Кол-во шагов двигателя для оборота на 360
const int stepsPerRevolution = 48;
//подключениекпинамнаАрдуино
Stepper myStepper(stepsPerRevolution, 3,5,2,4);
int Mag = 32;
int in1 = 2;
int in2 = 3;
int in3 = 4;
int in4 = 5;
const int d2 = 100;
void setup()
{
pinMode(22,OUTPUT);
pinMode(23,OUTPUT);
pinMode(24,OUTPUT);
pinMode(25,OUTPUT);
pinMode(26,OUTPUT);
pinMode(27,OUTPUT);
pinMode(in1,OUTPUT);
pinMode(in2,OUTPUT);
pinMode(in3,OUTPUT);
pinMode(in4,OUTPUT);
randomSeed(analogRead(0));
pinMode(Mag, OUTPUT);
myStepper.setSpeed(15); //установка скорости вращения ротора
Serial.begin(9600);
}
void loop()
{
if(digitalRead(28)==HIGH&&flag==0)//если кнопка нажата
// и переменнаяflag равна 0 , то ...
{
regim++;
flag=1;
//это нужно для того что бы с каждым нажатием кнопки
//происходило только одно действие
// плюс защита от "дребезга" 100%
```

```

if(regim>5)//ограничим количество режимов
{
regim=1;//так как мы используем только одну кнопку,
// то переключать режимы будем циклично
}
}
if(digitalRead(28)==LOW&&flag==1)//есликнопкаНЕнажата
//и переменная flag равна - 1 ,то ...
{
flag=0;//обнуляем переменную "кнопка"
}
if(regim==1) //первый режим
{
digitalWrite(22,HIGH);//включаем 1 светодиода
digitalWrite(23,LOW);
digitalWrite(24,LOW);
digitalWrite(25,LOW);
digitalWrite(26,LOW);
digitalWrite(27,LOW);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
digitalWrite( 32, LOW );
digitalWrite(22,LOW);
}
if(regim==2) //второй режим
{
digitalWrite(23,HIGH);//включае 2 светодиод
if(digitalRead(30)==HIGH&&flag3==0)//если кнопка нажата
// и переменнаяflag равна 0 , то ...
{
flag3=1;
//это нужно для того что бы с каждым нажатием кнопки
//происходило только одно действие
// плюс защита от "дребезга" 100%
}
if(digitalRead(30)==LOW&&flag3==1)//если кнопка НЕ нажата
//и переменная flag3 равна - 1 ,то ...
{
digitalWrite(32,HIGH);
delay(400);
myStepper.step(-4);
digitalWrite(32,HIGH);
delay(800);
digitalWrite(32,LOW);
}
}
}

```

```

delay(600);
myStepper.step(8);
delay(800);
digitalWrite(32,HIGH);
delay(400);
myStepper.step(-4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(8);
delay(800);
digitalWrite(32,HIGH);
delay(400);
myStepper.step(-4);
delay(800);
digitalWrite(32,LOW);
delay(400);
myStepper.step(-8);
delay(800);
flag3=0;//обнуляем переменную flag3
}
delay(300);
digitalWrite(23,LOW);
//здесь может быть любое ваше действие
}
if(regim==3) //третий режим
{
digitalWrite(24,HIGH);//включается 3 светодиода
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
if(digitalRead(30)==HIGH&&flag3==0)//если кнопка нажата
// и переменная flag равна 0 , то ...
{
flag3=1;
//это нужно для того что бы с каждым нажатием кнопки
//происходило только одно действие
// плюс защита от "дребезга" 100%
}
if(digitalRead(30)==LOW&&flag3==1)//если кнопка НЕ нажата
//и переменная flag3 равна - 1 ,то ...
{
// выводим случайное число в диапазоне от -48 до 48
randNumber = random(-48, 49);
myStepper.step(randNumber);
}
}

```

```

delay(1000);
flag3=0;//обнуляем переменную flag3
}
delay(300);
digitalWrite(24,LOW);
//здесь может быть любое ваше действие
}
if(regim==4) //четвертый режим
{
digitalWrite(25,HIGH);//включаем б светодиод
if(digitalRead(30)==HIGH&&flag3==0)//если кнопка нажата
// и переменнаяflag равна 0 , то ...
{
flag3=1;
//это нужно для того что бы с каждым нажатием кнопки
//происходило только одно действие
// плюс защита от "дребезга" 100%
}
if(digitalRead(30)==LOW&&flag3==1)//если кнопка НЕ нажата
//и переменная flag3 равна - 1 ,то ...
{
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(-12);
delay(800);
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(-4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(16);
delay(800);
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(-20);
delay(800);
digitalWrite(32,HIGH);

```

```
delay(1000);
myStepper.step(-4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(24);
delay(800);
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(-28);
delay(800);
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(-4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(32);
delay(800);
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(-36);
delay(800);
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(-4);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(40);
delay(800);
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(-16);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(-24);
```

```

delay(800);
digitalWrite(32,HIGH);
delay(1000);
myStepper.step(16);
delay(800);
digitalWrite(32,LOW);
delay(600);
myStepper.step(8);
delay(800);
flag3=0;//обнуляем переменную flag3
}
delay(300);
digitalWrite(25,LOW);
//здесь может быть любое ваше действие
}
if(regim==5) //пятый режим
{
digitalWrite(26,HIGH);//включаем 6 светодиод
if(digitalRead(31)==HIGH&&flag1==0)//есликнопканажата
// и переменная flag1 равна 0 , то ...
{
digitalWrite( in1, HIGH ); //ПОЧАСОВОЙ
digitalWrite( in2, LOW );
digitalWrite( in3, LOW );
digitalWrite( in4, HIGH );
delay(d2);
digitalWrite( in1, LOW );
digitalWrite( in2, LOW );
digitalWrite( in3, HIGH );
digitalWrite( in4, HIGH );
delay(d2);
digitalWrite( in1, LOW );
digitalWrite( in2, HIGH );
digitalWrite( in3, HIGH );
digitalWrite( in4, LOW );
delay(d2);
digitalWrite( in1, HIGH );
digitalWrite( in2, HIGH );
digitalWrite( in3, LOW );
digitalWrite( in4, LOW );
delay(d2);
flag1=1;
//это нужно для того что бы с каждым нажатием кнопки
//происходило только одно действие
// плюс защита от "дребезга" 100%
}

```

```

if(digitalRead(31)==LOW&&flag1==1)//если кнопка НЕ нажата
//и переменная flag равна - 1 ,то ...
{
flag1=0;//обнуляем переменную flag1
}
if(digitalRead(29)==HIGH&&flag2==0)//если кнопка нажата
// и переменная flag2 равна 0 , то ...
{
digitalWrite( in1, HIGH ); // против часовой
digitalWrite( in2, HIGH );
digitalWrite( in3, LOW );
digitalWrite( in4, LOW );
delay(d2);
digitalWrite( in1, LOW );
digitalWrite( in2, HIGH );
digitalWrite( in3, HIGH );
digitalWrite( in4, LOW );
delay(d2);
digitalWrite( in1, LOW );
digitalWrite( in2, LOW );
digitalWrite( in3, HIGH );
digitalWrite( in4, HIGH );
delay(d2);
digitalWrite( in1, HIGH );
digitalWrite( in2, LOW );
digitalWrite( in3, LOW );
digitalWrite( in4, HIGH );
delay(d2);
digitalWrite( in2, HIGH );
digitalWrite( in3, LOW );
digitalWrite( in4, LOW );
delay(d2);
flag2=1;
//это нужно для того что бы с каждым нажатием кнопки
//происходило только одно действие
// плюс защита от "дребезга" 100%
}
if(digitalRead(29)==LOW&&flag2==1) // если кнопка НЕ нажата
//и переменная flag2 равна - 1 ,то ...
{
flag2=0;//обнуляем переменную flag1
}
if(digitalRead(30)==HIGH&&flag3==0) //если кнопка нажата
// и переменная flag равна 0 , то ...
{
flag3=1;

```

```
//это нужно для того что бы с каждым нажатием кнопки
//происходило только одно действие
// плюс защита от "дребезга" 100%
}
if(digitalRead(30)==LOW&&flag3==1) // если кнопка НЕ нажата
//и переменная flag3 равна - 1 ,то ...
{
digitalWrite(32,!digitalRead(32));
digitalWrite(27,!digitalRead(27));
flag3=0; //обнуляем переменную flag3
}
digitalWrite(26,LOW);
}
}
```