

СПРАВКА

о результатах проверки текстового документа на наличие заимствований

**Проверка выполнена в системе
Антиплагиат.Структура**

Автор работы	Шамраев Алексей Дмитриевич
Подразделение	Отделение информационных технологий и энергетических систем
Тип работы	Выпускная квалификационная работа
Название работы	BKP
Название файла	BKP.pdf
Процент заимствования	33.06 %
Процент самоцитирования	0.00 %
Процент цитирования	4.19 %
Процент оригинальности	62.75 %
Дата проверки	19:49:59 16 июня 2020г.
Модули поиска	Модуль поиска ИПС "Адилет"; Модуль выделения библиографических записей; Сводная коллекция ЭБС; Модуль поиска "Интернет Плюс"; Коллекция РГБ; Цитирование; Модуль поиска переводных заимствований; Модуль поиска переводных заимствований по elibrary (EnRu); Модуль поиска переводных заимствований по интернет (EnRu); Коллекция eLIBRARY.RU; Коллекция ГАРАНТ; Модуль поиска "КПФУ"; Коллекция Медицина; Диссертации и авторефераты НББ; Модуль поиска перефразирований eLIBRARY.RU; Модуль поиска перефразирований Интернет; Коллекция Патенты; Модуль поиска общеупотребительных выражений; Кольцо вузов
Работу проверил	Балабанов Игорь Петрович
Дата подписи	_____
	Подпись проверяющего

Чтобы убедиться
в подлинности справки,
используйте QR-код, который
содержит ссылку на отчет.



Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Отделение информационных технологий и энергетических систем
Кафедра автоматизации и управления

Направление подготовки 15.03.04 Автоматизация технологических процессов и

производств

Профиль _____

Допустить к защите
Заведующий кафедрой

_____ /Л.А. Симонова/
«____» _____ 2020г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(Бакалаврская работа)

БР 1.150304.20.2161113.10.00.00 ПЗ

на тему: _____ Разработка системы управления роботом-манипулятором с 5 степенями
свободы _____

Выпускник _____ / _____ Шамраев А. Д. _____ /
(подпись) (Ф.И.О.)

Руководитель ВКР _____ / _____ Сабиров И.С., к.т.н., доцент _____ /
(подпись) (Ф.И.О., уч. степень, звание, должность)

Набережные Челны 2020г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Отделение информационных технологий и энергетических систем
Кафедра автоматизации и управления

Направление подготовки 15.03.04 Автоматизация технологических процессов и

производств

Профиль _____

Допустить к защите
Заведующий кафедрой

_____ /Л.А. Симонова/
«____» _____ 2020г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(Бакалаврская работа)

БР 1.150304.20.2161113.10.00.00 ПЗ

на тему: Разработка системы управления роботом-манипулятором с 5 степенями свободы

Выпускник _____ / _____ Шамраев А. Д. _____ /
(подпись) (Ф.И.О.)

Руководитель ВКР _____ / _____ Сабиров И.С., к.т.н., _____ /
(подпись) (Ф.И.О., уч. степень, звание, должность)

Набережные Челны 2020г.

Предварительную защиту прошел

« » июня 2020г.

Председатель _____ А.А.Шабаев

Члены _____ Р.Р.Зиятдинов

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Отделение информационных технологий и энергетических систем
Кафедра автоматизации и управления

Направление подготовки 15.03.04 Автоматизация технологических процессов и производств

Профиль _____

УТВЕРЖДАЮ
Заведующий кафедрой

_____ / Л.А. Симонова/

«____»_____ 2020 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

1. Выпускник Шамраев Алексей Дмитриевич
2. Руководитель Сабиров Ильдар Салихзянович,
3. Степень к.т.н.
4. Тема ВКР Разработка системы управления роботом-манипулятором с 5 степенями свободы
утверждена приказом по институту от «15» апреля 2020 г. № 04/210
5. Исходные данные робот «Робко 01» без платы управления, с изношенными, сломанными частями
6. Содержание работы анализ предметной области, разработка аппаратной части системы управления, разработка программной части системы управления
7. Перечень отчетных материалов, 4 чертежа, 3 плаката

Дата выдачи задания
«____» апреля 2020г.

Дата представления ВКР к защите
«____» июня 2020 г.

Выпускник Шамраев А. Д. / _____ /

Руководитель Сабиров И. С. / _____ /

КАЛЕНДАРНЫЙ ПЛАН

выполнения основных и дополнительных разделов выпускной квалификационной работы

Основные разделы ВКР:

Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Ф.И.О. руководителя, подпись
1. Анализ предметной области	С 01.05.2020 по 06.05.2020	С 01.05.2020 по 05.05.2020	Сабиров И. С.
2. Разработка структурной и электрической принципиальной схемы платы управления	С 07.05.2020 по 08.05.2020	С 06.05.2020 по 07.05.2020	Сабиров И. С.
3. Выбор элементов СУ, изготовление платы	С 09.05.2020 по 10.05.2020	С 08.05.2020 по 09.05.2020	Сабиров И. С.
4. Разработка программной части системы управления	С 11.05.2020 по 06.06.2020	С 10.05.2020 по 06.06.2020	Сабиров И. С.

Дополнительные разделы ВКР:

Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Ф.И.О. консультанта, подпись
Нормоконтроль	08.06.2020 г.	_____.06.2020 г.	_____ Р.Р.Валиахметов

Руководитель выпускной квалификационной работы

Задание принял к исполнению

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**ОТЗЫВ РУКОВОДИТЕЛЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ
РАБОТЫ**

Выпускник Шамраев Алексей Дмитриевич
(Ф.И.О.)

Отделение информационных технологий и энергетических систем Группа 2161113
Направление подготовки 15.03.04 Автоматизация технологических процессов и производств

Профиль _____

Наименование темы: Разработка системы управления роботом-манипулятором с 5 степенями свободы

Руководитель Сабиров И.С., доцент кафедры автоматизации и управления, доцент, к.т.н.

(Ф.И.О., место работы, должность, ученое звание, степень)

ПОКАЗАТЕЛИ ОЦЕНКИ ВКР

№ п/п	Показатели	Оценка			
		5	4	3	0*
1	Новизна и актуальность темы работы				
2	Практическая значимость (ценность) работы				
3	Наличие элементов творчества				
4	Использование прикладных программ, проведение экспериментов				
5	Применение современной методологии исследования, расчетов, проведения экспериментов				
6	Оформление работы в соответствии с действующими стандартами				
7	Умение работать с различными источниками				
ИТОГОВАЯ ОЦЕНКА					

(*)- не оценивается, трудно оценить.

Характеристика в свободной форме, но не более страницы.

Руководитель Сабиров И.С. / _____ /
(подпись)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

РЕЦЕНЗИЯ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Выпускник Шамраев Алексей Дмитриевич
(*Ф. И.О.*)

Отделение информационных технологий и энергетических систем

Кафедра автоматизации и управления

Направление подготовки 15.03.04 Автоматизация технологических процессов и производств

Профиль _____

Наименование темы Разработка системы управления роботом-манипулятором с 5 степенями свободы

Рецензент _____

(*Фамилия И.О., место работы, должность, ученое звание, степень*)

ПОКАЗАТЕЛИ ОЦЕНКИ ВКР

№ п/ п	Показатели	Оценка				
		5	4	3	2	0*
1	Актуальность темы работы					
2	Практическая значимость (ценность) работы					
3	Наличие элементов творчества					
4	Применение современной методологии исследования, проведение экспериментов					
5	Использование прикладных программ, проведение экспериментов					
6	Практическая значимость (ценность) работы					
7	Оформление работы в соответствии с действующими стандартами					
Итоговая оценка						

(*) - не оценивается (трудно оценить) _____

В качестве недостатков и пожеланий для дальнейшей работы следует отметить

Рецензент _____ / _____ /
(подпись)

Дата _____ МП

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

АННОТАЦИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Выпускник _____

Отделение информационных технологий и энергетических систем Группа 2161113

Направление подготовки 15.03.04 Автоматизация технологических процессов и производств

Профиль _____

Тема ВКР: Разработка системы управления роботом-манипулятором с 5 степенями свободы

Руководитель Сабиров И.С., доцент кафедры автоматизации и управления, доцент, к.т.н.

(*Фамилия И.О., место работы, должность, ученое звание, степень*)

Консультант(ы) _____

(*Фамилия И.О., место работы, должность, ученое звание, степень*)

ХАРАКТЕРИСТИКА ВКР

1. Тема ВКР Разработка системы управления роботом-манипулятором с 5 степенями свободы

2. Цель работы для повышения производительности труда разработать систему управления роботом-манипулятором с 5 степенями свободы

3. Сведения об объеме, количестве иллюстраций, таблиц, использованных источников Количество страниц – 84, использованных источников – 15, иллюстраций – 79, таблиц – 6, формул – 19

4. Ключевые слова автоматизация, ОЗК, ПЗК, робот–манипулятор, 5 степеней свободы, система управления

5. Использование информационных ресурсов Internet да, 2

(*нет, да, число ссылок в списке литературы*)

6. Использование современных пакетов компьютерных программ и технологий Mathcad 15, Matlab R2016a, Wolfram Mathematica, AutoCAD 2017, Компас 2016, Visio 2016, Cambam, Visual Studio 2017, Visual Studio Code v.1.4, Arduino IDE 1.8.5, KiCad v.5.1.0

(*нет, да, число ссылок в списке литературы*)

7. Результаты работы система управления роботом-манипулятором с 5 степенями свободы

8. Наличие публикаций и выступлений на конференциях по теме выпускной работы да

a) 1. Статья в Web of Science – Analytical inverse kinematic solution for original robotic manipulator construction. Alexey D. Shamraev, Igor P. Balabanov, Rustem R. Ziyatdinov. Helix Vol. 9 (5): 5327- 5334. DOI 10.29042/2019-5327-5334 (WOS).

2. Разработка системы управления манипулятором "Робко 01" Шамраев А.Д. Научно-практические исследования. 2019. № 8-8 (23). С. 66-68.

3. Обзор методов решения обратной задачи кинематики для манипуляторов Шамраев А.Д. В сборнике: фундаментальные основы инновационного развития науки и образования сборник статей VI международной научно-практической конференции : в 3 ч.. Пенза, 2019. С. 51-53.

4. Решение обратной задачи кинематики методом обратных преобразований для манипулятора с пятью степенями свободы Шамраев А.Д. В сборнике: Современное научное знание: теория, методология, практика Сборник научных трудов по материалам XI Международной научно-практической конференции. 2019. С. 75-78.

5. Решение обратной задачи кинематики для манипулятора с замкнутой кинематической цепью Шамраев А.Д. В сборнике: Образование, наука и технологии: проблемы и перспективы сборник научных трудов по материалам II международной научно-практической конференции. Под общ. ред. А.В. Туголукова. Москва, 2019. С. 188-193.

(Список публикаций)

б) 1. «Фундаментальные основы инновационного развития науки и образования» VI международная научно-практическая конференция, Пенза, 2019.

2. «Современное научное знание: теория, методология, практика» XI Международная научно-практическая конференция. 2019.

3. «Образование, наука и технологии: проблемы и перспективы» II международной научно-практической конференции. Под общ. ред. А.В. Туголукова. Москва, 2019.

(Список выступлений на конференциях)

Выпускник

(подпись)

Руководитель

(подпись)

«____» _____ 2020 г.

СОДЕРЖАНИЕ

Введение	4
1 Анализ предметной области.....	6
1.1 Описание робота «Робко 01»	6
1.2 Обзор драйверов шаговых двигателей	8
1.3 Обзор датчиков	14
1.4 Обзор микроконтроллеров	18
1.5 Прямая задача кинематики.....	29
1.6 Обратная задача кинематики	30
1.7 Выводы по разделу.....	33
2 Разработка аппаратной части системы управления.....	35
2.1 IDEF0 для описания модели процесса разработки и реализации системы управления	35
2.2 Разработка структурной схемы платы управления.....	37
2.3 Выбор драйвера для управления шаговым двигателем	38
2.4 Выбор датчиков	40
2.5 Выбор микроконтроллера.....	41
2.6 Разработка схемы электрической принципиальной	42
2.7 Изготовление платы	44
2.8 Установка датчиков.....	50
2.9 Выводы по разделу.....	51
3 Разработка программной части системы управления	52
3.1 Разработка архитектуры программной части системы управления	52
3.2 Решение прямой задачи кинематики методом Денавита-Хартенберга....	56
3.3 Решение обратной задачи кинематики методом обратных преобразований.....	60
3.4 Расчет операционного пространства робота-манипулятора «Робко 01» .	64

3.5 Разработка алгоритма работы системы управления	68
3.6 Разработка линейного и кругового интерполяторов	71
3.7 Структура кода программы	73
3.8 Интерфейс программного обеспечения	76
3.9 Логгирование сессии программного обеспечения	77
3.10 Результат работы системы управления.....	78
3.11 Выводы по разделу.....	80
Заключение.....	81
Список использованных источников	83

ВВЕДЕНИЕ

В представленной работе показана разработка системы управления для робота-манипулятора «Робко 01». Указанный робот не имеет современной системы управления, которая бы выполняла ряд таких задач, как: позиционное, контурное управления. В связи с такой проблемой требуется разработать современную, гибкую, многофункциональную систему управления.

Таким образом сформулируем цель работы: для повышения производительности труда разработать систему управления роботом-манипулятором с 5 степенями свободы.

Система управления для робота-манипулятора с 5 степенями свободы, далее СУ, необходима для задания необходимых положений звеньев робота в каждый момент времени. Например, с помощью схвата переместить объект в операционном пространстве. Зная начальное и конечное положения (x,y,z) объекта необходимо найти углы поворота каждого звена, то есть решить обратную задачу кинематики (ОЗК). Далее контроллер по найденным углам перемещения звеньев дает задание соответствующим драйверам шаговых двигателей сделать определенное количество шагов в определенном направлении.

Функционал: управление движением роботом «Робко 01», решение прямой задачи кинематики (ПЗК), решение обратной задачи кинематики (ОЗК).

Для достижения поставленной цели необходимо выполнить ряд задач:

- 1) Разработка архитектуры СУ
- 2) Разработка схемы платы системы управления
- 3) Выбор драйвера ШД
- 4) Выбор датчиков
- 5) Выбор МК
- 6) Разработка протокола обмена данными
- 7) Решение прямой задачи кинематики

8) Решение обратной задачи кинематики

9) Планирование пути

10) Разработка прошивки для МК

При разработке десктопного приложения:

1) Архитектура приложения (классы и пр.)

2) Решатели ПЗК и ОЗК

3) Интерпретаторы G-кода

4) Интерполяторы (линейные и круговые)

1 Анализ предметной области

1.1 Описание робота «Робко 01»

Объектом разрабатываемой системы управления является манипуляционный робот (МР) «Робко 01».

«Робко 01» – болгарский обучаемый антропоморфный робот (робот-рука) с 5 степенями подвижности. Он был разработан в ИТКР-БАН с целью обучения робототехнике и кибернетике. Разработан для работы с „ИМКО-1“ и последующие его «ИМКО-2» и «Правец-8». Он производился на фабрике медицинской техники в Софии. Было сделано около 17 000 штук.

Робот «Робко 01» изображен на рисунке 1.



Рисунок 1 – Робот-манипулятор «Робко 01»

Установленный в «Робко 01» шаговый двигатель [1] можно видеть на рисунке 2.



Рисунок 2 – Установленный в «Робко 01» шаговый двигатель

Шаговые двигатели робота «Робко 01» – униполярные, с 6 выводами, могут управляться в биполярном режиме.

У этого робота есть коробка для размещения платы управления (рисунок 3) размером 210x150x25.



Рисунок 3 – Коробка для размещения платы управления в «Робко 01»

Общие характеристики:

- 1) он управляется 8-битным компьютером «Правец 8», а также современными 32 и 64-битными компьютерами независимо от оси;
- 2) привод шаговых электродвигателей;
- 3) конструкция робота обеспечивает наглядность кинематики и привода;
- 4) обладает богатыми возможностями программного обеспечения в различных системах координат;
- 5) правила работы относительно просты и легко усваиваются даже новичкам;
- 6) обеспечивает безопасность работы;

Технические характеристики:

- 1) количество степеней подвижности-5;
- 2) радиус рабочего объема-356 мм;
- 3) рассеяние при многократном позиционировании (повторяемость) - 1 мм;
- 4) точность повторения обученной точки-5 мм;

- 5) грузоподъемность-1500 г
- 6) напряжение питания-12 В;
- 7) потребление-до 2,5 А;

1.2 Обзор драйверов шаговых двигателей

Драйвер шагового двигателя – электронное устройство, которое заставляет шаговый двигатель "шагать" по сигналам управления. Стандартом де-факто в области управления ШД являются сигналы STEP/DIR/ENABLE. STEP это сигнал шага, DIR это сигнал направления вращения, ENABLE это сигнал включения драйвера.

Управлять ШД намного сложнее чем обычным коллекторным двигателем – нужно в определенной последовательности переключать напряжения в обмотках с одновременным контролем тока. Поэтому для управления ШД разработаны специальные устройства – драйверы ШД. Драйвер ШД позволяет управлять вращением ротора ШД в соответствии с сигналами управления и электронным образом делить физический шаг ШД на более мелкие дискреты.

К драйверу ШД подключается источник питания, сам ШД (его обмотки) и сигналы управления. Стандартом по сигналам управления является управление сигналами STEP/DIR или CW/CCW и сигнал ENABLE.

Драйвер ULN2003а. Микросхема ULN2003 (ULN2003а на рисунке 4) по сути своей является набором мощных составных ключей для применения в цепях индуктивных нагрузок. Может быть применена для управления нагрузкой значительной мощности, включая электромагнитные реле, двигатели постоянного тока, электромагнитные клапаны, в схемах управления различными шаговыми двигателями и другие.



Рисунок 4 – Драйвер ШД ULN2003а

Микросхема ULN2003а – это транзисторная сборка Дарлингтона с выходными ключами повышенной мощности, имеющая на выходах защитные диоды, которые предназначены для защиты управляющих электрических цепей от обратного выброса напряжения от индуктивной нагрузки.

Каждый канал (пара Дарлингтона) в ULN2003 рассчитан на нагрузку 500 мА и выдерживает максимальный ток до 600 мА. Входы и выходы расположены в корпусе микросхемы друг напротив друга, что значительно облегчает разводку печатной платы.

ULN2003 относится к семейству микросхем ULN200X. Различные версии этой микросхемы предназначены для определенной логики. В частности, микросхема ULN2003 предназначена для работы с TTL логикой (5В) и логических устройств CMOS. Широкое применение ULN2003 нашло в схемах управления широким спектром нагрузок, в качестве релейных драйверов, драйверов дисплея, линейных драйверов и т. д. ULN2003 также используется в драйверах шаговых двигателей. Схема подключения показана на рисунке 5.

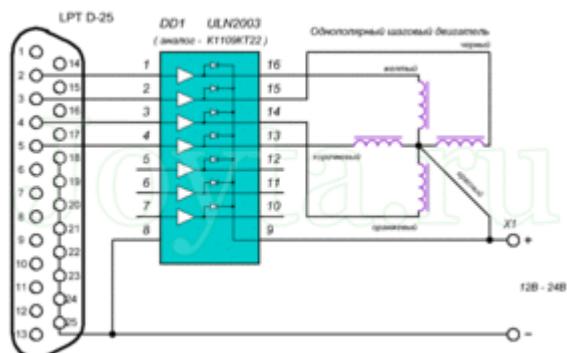


Рисунок 5 – Схема подключения ULN2003

Важнейшие технические характеристики:

- 1) Максимальное напряжение 50 В
- 2) Максимальный ток 0.5 А
- 3) Управление

Драйвер A4988. Основная микросхема модуля это драйвер от Allegro – A4988 (рисунок 6), которая имеет небольшие размеры (всего 8 мм х 6 мм), хоть микросхема и маленькая, но она может работать с выходным напряжением до 35 В с током до 1 А на катушку без радиатора и до 2 А с радиатором (дополнительным охлаждением). Для управления шаговым двигателем, необходимо всего два управляющих контакта (по сравнению с L298N необходимо четыре), один используется для управления шагами, второй для управления вращения двигателем.

Драйвер позволяет использовать пять вариантов шага, полный шаг, полшага, четверть шага, восьмой шаг и шестнадцатый шаг.

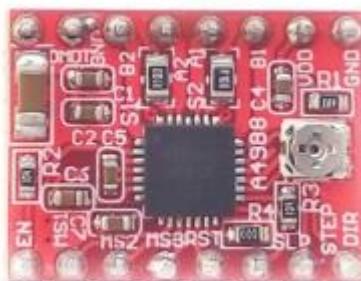


Рисунок 6 – Драйвер ШД A4988

Технические параметры:

- 1) Напряжение питания: от 8 до 35 В
- 2) Установка шага: 1, 1/2, 1/4, 1/8, 1/16
- 3) Напряжение логики: 3 В или 5.5 В
- 4) Защита от перегрева: Есть
- 5) Максимальный ток на фазу: 1 А без радиатора, 2 А с радиатором.
- 6) Габариты модуля: 20 мм x 15 мм x 10 мм
- 7) Габариты радиатора: 9 мм x 5 мм x 9 мм

Режимы микрошага показаны на рисунке 7.

MS1	MS2	MS3	Resolution
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Half Step
LOW	HIGH	LOW	Quarter Step
HIGH	HIGH	LOW	Eighth step
HIGH	HIGH	HIGH	Sixteenth Step

Рисунок 7 – Режим микрошага A4988

Назначение контактов драйвера A4988:

- 1) ENABLE – включение/выключение драйвера
- 2) MS1, MS2, MS3 – контакты для установки микрошага
- 3) RESET - сброс микросхемы
- 4) STEP - генерация импульсов для движения двигателей (каждый импульс – шаг), можно регулировать скорость двигателя
- 5) DIR – установка направление вращения
- 6) VMOT – питание для двигателя (8 – 35 В)
- 7) GND – общий
- 8) 2B, 2A, 1A, 1B – для подключения обмоток двигателя
- 9) VDD – питание микросхемы (3.5 – 5В)

Для работы в режиме микрошага необходим слабый ток. На модуле A4988 имеется потенциометр для тонкой настройки по току. Драйвер очень чувствителен к скачкам напряжения по питанию двигателя, поэтому производитель рекомендует устанавливать электролитический конденсатор большой емкости по питанию VMOT для сглаживания скачков. Стоит обратить особое внимание на то, что подключение или отключение шагового двигателя при включённом драйвере может привести к выходу двигателя из строя.

Схема подключения драйвера A4988 для управления биполярным шаговым двигателем показана на рисунке 8. Вывод RESET подключен к выводу SLEEP, чтобы на нем был высокий уровень HIGH.

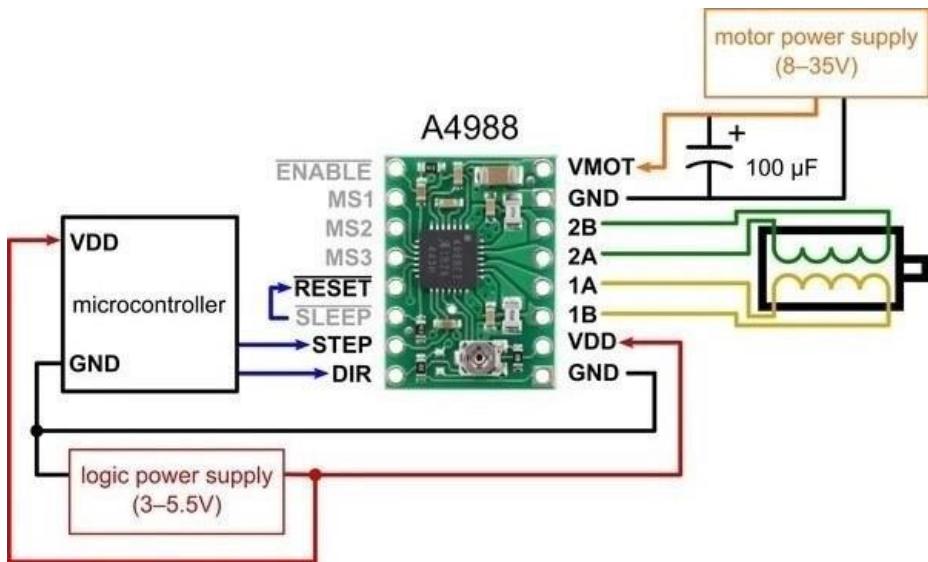


Рисунок 8 – Схема подключения A4988

Драйвер DRV8852. DRV8825 (показан на рисунке 9) полностью взаимозаменяемый с драйвером A4988 и может работать с микрошагом до 1/32, напряжением до 45 В и током до 2.5 А.

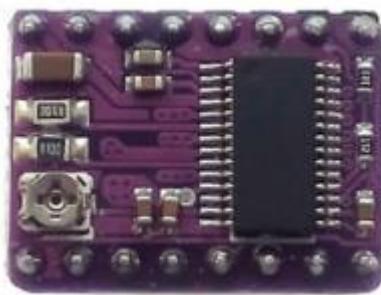


Рисунок 9 – Драйвер ШД DRV8852

Основная микросхема модуля это драйвер от TI (Texas Instruments Inc.) DRV8825, которая способна управлять одним биполярным шаговым двигателем. Как говорил ранее данный драйвер полностью взаимозаменяемый с драйвером A4988. Микросхема DRV8825 может работать с выходным напряжением до 45 В и током до 1.5 на катушку без радиатора и до 2.5 А с радиатором (дополнительным охлаждением). Так же, модуль имеет внутренний стабилизатор напряжение, который питает логическую часть модуля напряжение 3.3 В от источника шагового питания двигателя.

Драйвер позволяет использовать шесть вариантов шага: 1, 1/2, 1/4, 1/8, 1/16, 1/32.

Технические параметры:

- 1) Напряжения питания: от 8,2 до 45 В
- 2) Установка шага: 1, 1/2, 1/4, 1/8, 1/16, 1/32
- 3) Напряжение логики: 3.3 В
- 4) Защита от перегрева: Есть
- 5) Максимальный ток на фазу: 1.5 А без радиатора, 2.5 А с радиатором.
- 6) Габариты модуля: 20 мм x 15 мм x 10 мм
- 7) Габариты радиатора: 9 мм x 5 мм x 9 мм

Схема подключения аналогична A4988.

Драйвер L298N. Основной чип модуля это микросхема L298N (показан на рисунке 10), состоящая из двух Н-мост (H-Bridge), один для выхода А, второй для выхода В.



Рисунок 10 – Драйвер ШД L298N

Н-мост широко используется в электронике и служит для изменения вращения двигателем, схема Н-моста содержит четыре транзистора (ключа) с двигателем в центре, образуя Н-подобную компоновку. Принцип работы прост, при одновременном закрытие двух отдельных транзистора изменяется полярность напряжения, приложенного к двигателю. Это позволяет изменять направление вращения двигателя. На рисунке 11 показана работа Н-мостовой схемы.

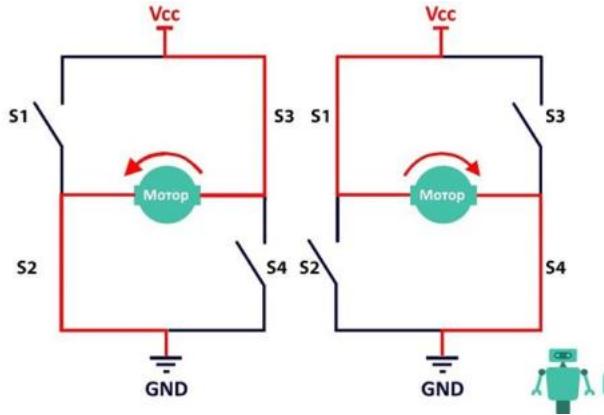


Рисунок 11 – работа Н-мостовой схемы

Для управления скоростью двигателя постоянного тока используется метод PWM (Широтно-импульсной модуляции).

Технические параметры:

- 1) Напряжение питания логики модуля: 5 В
- 2) Потребляемый ток встроенной логики: 36 мА
- 3) Напряжение питания драйвера: 5 В – 35 В
- 4) Рабочий ток драйвера: 2 А (пиковый ток 3 А)
- 5) Габариты: 43.5 мм x 43.2мм x 29.4мм

1.3 Обзор датчиков

Для ограничения движения разных агрегатов и механизмов применяются концевые выключатели. К этим устройствам предъявляются требования по их надежности, долговечности и безопасности при работе.

Имеется много разных видов концевиков, разделяющихся по принципу действия и особенностям. В каждый вид могут входить дополнительные виды. Это зависит от места применения устройства. Перечислим виды датчиков: механические, микровыключатели, герконы, индуктивные выключатели, оптические сенсоры, емкостные выключатели.

Датчики KIPPRIBOR серии KLS-A5.xxxx. Концевые выключатели (концевики) KIPPRIBOR серии KLS – это универсальные устройства, которые имеют очень широкую область применения. Они могут использоваться в качестве конечных, путевых выключателей, устройств блокировки, коммутационных и переключающих элементов в самом различном промышленном оборудовании.

Схема выключателей KIPPRIBOR серии KLS-A5.xxxx показана на рисунке 12.

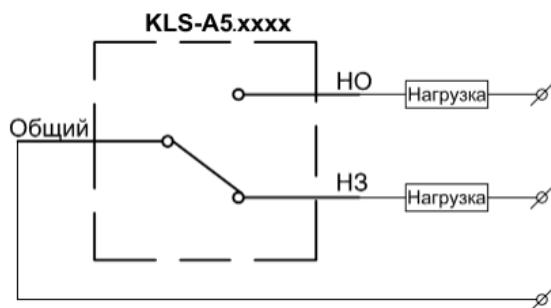


Рисунок 12 – Схема выключателей KIPPRIBOR серии KLS-A5.xxxx

Технические характеристики концевых выключателей KIPPRIBOR серии KLS изображены в таблице 1.

Таблица 1 – Технические характеристики концевых выключателей KIPPRIBOR серии KLS

Параметр	Значение параметра
1	2
Скорость срабатывания	0,01 мм/с...1м/с
Частота срабатывания	механическая: 240 переключений/мин электрическая: 20 переключений/мин
Тип контакта	1НО+1НЗ
Номинальный ток	1.5 A / 380VAC (AC-15(1)) 0,3A / 220 VDC (DC-13(1))
Термический ток(2)	15 A
Сопротивление контактов	<15 мОм
Сопротивление изоляции	100 мОм (500 VDC)

Продолжение таблицы 1

1	2
Диэлектрическая прочность	1890 VAC в течение минуты между контактами 2500 VAC в течение минуты между контактом и корпусом 2500 VAC в течение 1 минуты между корпусом и заземлением
Виброустойчивость	Отказы в работе: 10...55 Гц (удвоенная амплитуда 1,5 мм)
Ударопрочность	Разрушение: 1000 м/с ² Отказы в работе: 300 м/с ²
Рабочая температура	-10...+80°C без обледенения и образования конденсата
Влажность	<90%
Механическая износостойкость	10 млн. циклов
Электрическая износостойкость	200 000 циклов
Степень защиты	Корпус: IP40 Контакты: IP00

На рисунках 13–15 показаны микровыключатели KIPPRIBOR серии KLS-A5.xxxx.

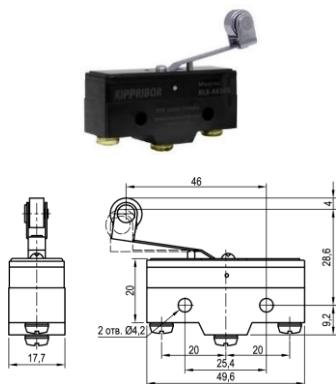


Рисунок 13 – Вид и схема KLS-A5.15S

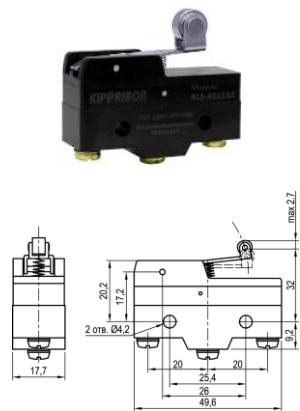


Рисунок 14 – Вид и схема KLS-A5.11G2

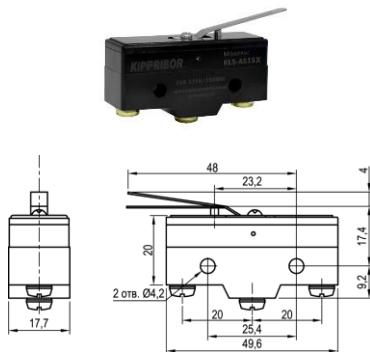


Рисунок 15 – Вид и схема KLS-A5.15X

Электромеханический микровыключатель EziUsin изображен на рисунке 16.



Рисунок 16 – Электромеханический микровыключатель EziUsin

Технические характеристики EziUsin:

- 1) Максимальный ток – 5А;
 - 2) Максимальное напряжение – 250В;
 - 3) Габаритные размеры – 25x20x6 мм;
 - 4) Механическая износостойкость – 1 млн. циклов;

Датчики Холла TLE4905LHALA1 изображен на рисунке 17.



Рисунок 17 – Датчики Холла TLE4905LHALA1

Технические характеристики TLE4905LHALA1 показаны в таблице 2.

Таблица 2 – Технические характеристики TLE4905LHALA1

Параметр	Значение параметра
Корпус	P-SSO-3-2
Корпус (размер)	3-SIP, SSO-3-02
Рабочая температура	-40°C ~ 150°C
Тип выхода	Digital, Open Collector
Current - Output (Max)	100mA
Ток выходной	8mA
Напряжение питания	3.8 V ~ 24 V
Тип	Unipolar Switch
Sensing Range	18mT Trip, 5mT Release
Lead Free Status / RoHS Status	Lead free / RoHS Compliant
Output Type	Digital, Open Collector
Габаритные размеры	4x5x1.5

1.4 Обзор микроконтроллеров

Atmel AVR представляет собой семейство универсальных 8-разрядных микроконтроллеров на основе общего ядра с различными встроенными периферийными устройствами [2]. МК этого семейства могут изготавливаться в разных корпусах, с разными габаритами, как показано на рисунке 18.

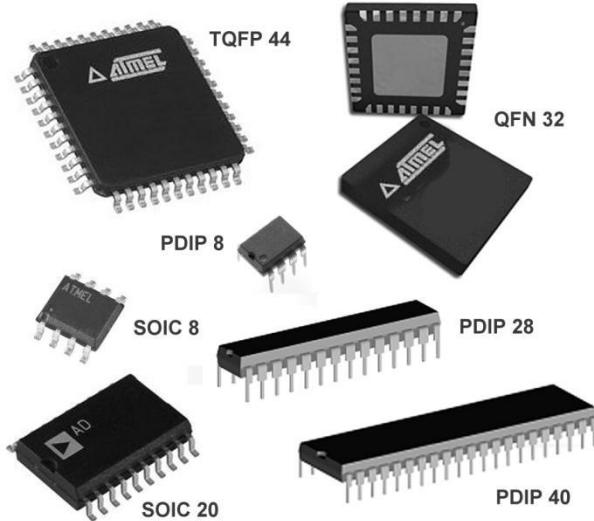


Рисунок 18 – Примеры различных типов корпусов для МК AVR

На сегодняшний день семейство МК Atmel AVR делится на 2 подсемейства – Tiny и Mega. Можно сказать, что Tiny менее “навороченные” чем Mega.

Подсемейство Mega оснащено Flash-ПЗУ программ объемом 8–256 кбайт и корпусами с 28–100 выводами, в то время как микросхемы Tiny имеют Flash-ПЗУ программ объемом 1–8 кбайт и размещаются в основном в корпусах с 8–20 выводами.

Все семейства могут иметь две модификации: буква "L" в обозначении говорит о расширенном диапазоне питания 2,7–5,5 В, отсутствие такой буквы означает диапазон питания 4,5–5,5 В [2].

Рассмотрим архитектуру МК AVR. В структуре AVR (рисунок 19) имеются три разновидности памяти: flash-память программ, ОЗУ (SRAM) для временных данных и энергонезависимая память (EEPROM) для долговременного хранения констант и данных [2].

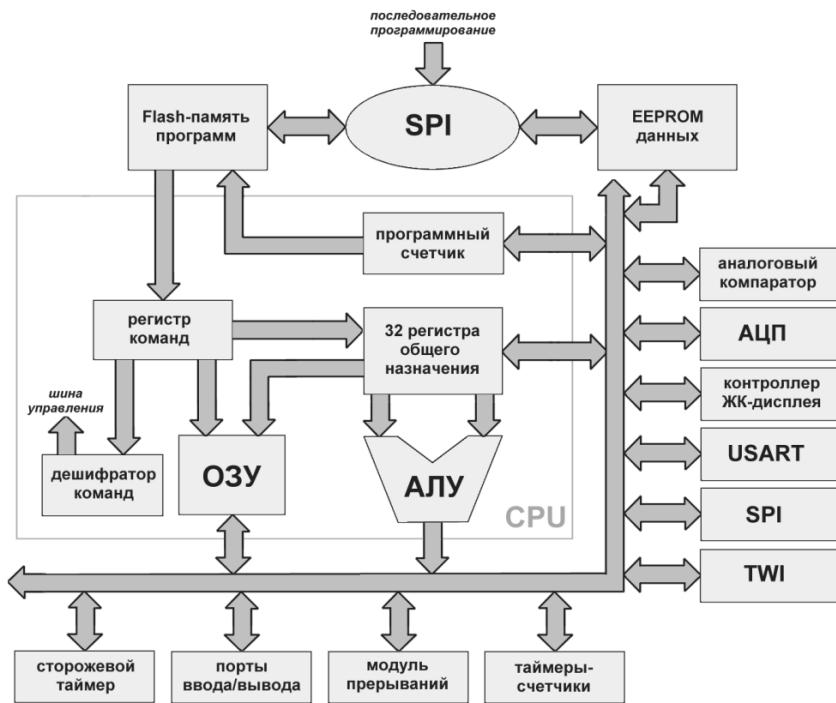


Рисунок 19 – Общая структурная схема микроконтроллеров AVR

Как показано, на рисунке 20 память данных МК разделена на регистры общего назначения (РОН), регистры ввода-вывода (РВВ), и статическую память данных (SRAM) [3].

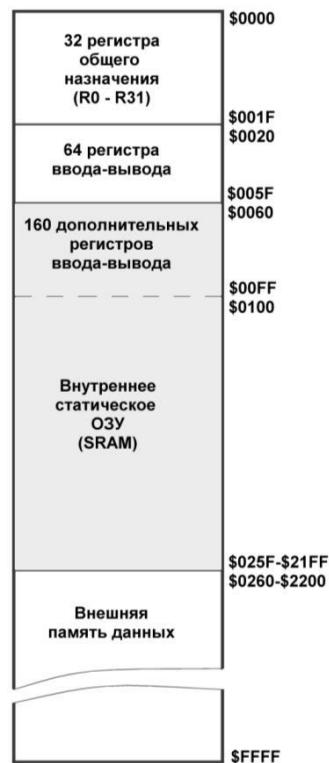


Рисунок 20 – Адресное пространство статической памяти данных (SRAM) микроконтроллеров AVR

STM от STMicroelectronics. Восьмиразрядные микроконтроллеры STM8 (рисунок 21) были выпущены в 2008 году и позиционировались для задач, требующих высокой надежности при низком энергопотреблении.

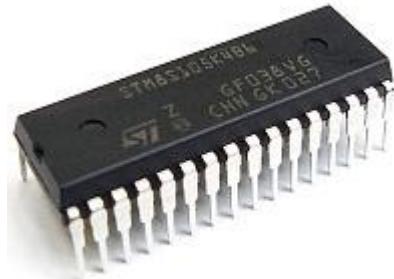


Рисунок 21 – Микроконтроллер STM8

Они сразу обратили на себя внимание невысокой ценой и широким модельным рядом. Работа с микроконтроллерами STM8 не вызвала никаких сложностей как у тех, кто уже освоил работу с микроконтроллерами PIC или AVR, так и у начинающих, и они быстро обрели широкую популярность. STM8 имеют хороший набор периферии и развитые средства программирования. Небольшим сдерживающим фактором распространения этих микроконтроллеров в России поначалу стало отсутствие достаточного количества учебной литературы на русском языке, но сегодня уже существует множество сайтов с подробными руководствами по освоению этих микроконтроллеров.

Чуть раньше, в 2004 году, Acorn разработали 32-разрядное процессорное ядро ARM Cortex-M3 и компания STMicroelectronics стала одной из первых, кто вывел на рынок семейство микроконтроллеров на этом ядре, получивших название STM32. Сегодня ARM Cortex-M3 становится, возможно, самым популярным 32-разрядным процессорным ядром для встраиваемых систем, а микроконтроллеры на его базе имеют универсальную, хорошо сбалансированную архитектуру. Многие специалисты считают STM32 одним из самых распространенных, дешевых и перспективных 32-разрядных микроконтроллеров.

Arduino Uno. Arduino Uno контроллер построен на ATmega328 (рисунок 22). Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, квадратный генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

В отличие от всех предыдущих плат, использовавших FTDI USB микроконтроллер для связи по USB, новый Ардуино Uno использует микроконтроллер ATmega8U2.

"Uno" переводится как один с итальянского и разработчики тем самым намекают на грядущий выход Arduino 1.0. Новая плата стала флагманом линейки плат Ардуино. Для сравнения с предыдущими версиями можно обратиться к полному списку плат Arduino.

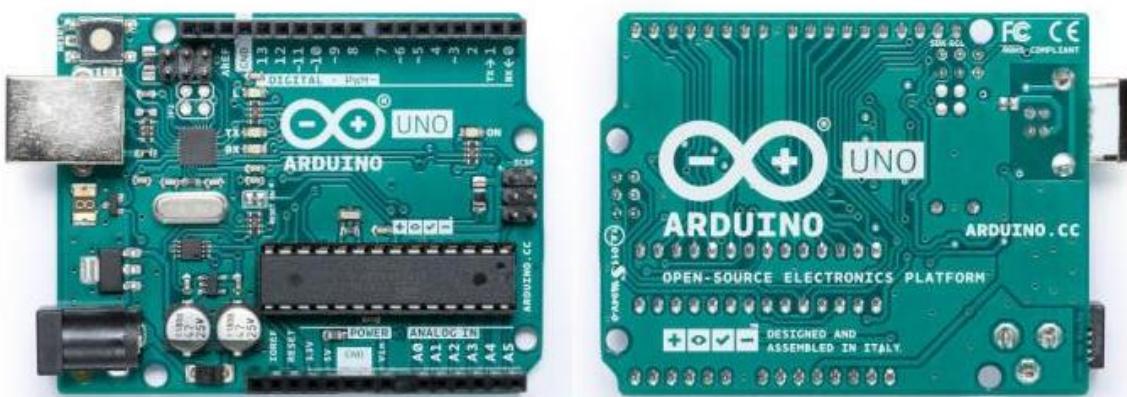


Рисунок 22 – Arduino Uno

Технические характеристики Arduino Uno показаны в таблице 3.

Таблица 3 – Технические характеристики Arduino Uno

Параметр	Значение параметра
Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флеш-память	32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика
ОЗУ	2 Кб (ATmega328)
EEPROM	1 Кб (ATmega328)
Тактовая частота	16 МГц

Arduino Mega 2560. Arduino Mega (рисунок 23) построена на микроконтроллере ATmega2560. Плата имеет 54 цифровых входа/выходов (14 из которых могут использоваться как выходы ШИМ), 16 аналоговых входов, 4 последовательных порта UART, квадцевый генератор 16 МГц, USB коннектор, разъем питания, разъем ICSP и кнопка перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB или подать питание при помощи адаптера AC/DC, или аккумуляторной батареи. Arduino Mega 2560 совместима со всеми платами расширения, разработанными для платформ Uno или Duemilanove.

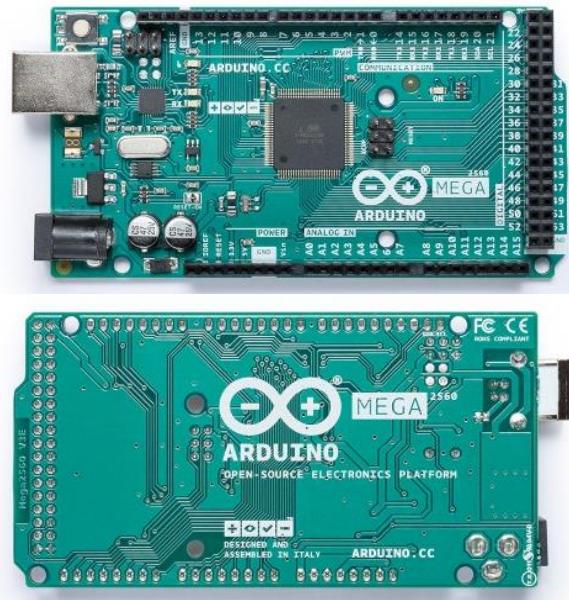


Рисунок 23 – Arduino Mega

Технические характеристики Arduino Uno показаны в таблице 4.

Таблица 4 – Технические характеристики Arduino Mega

Параметр	Значение параметра
Микроконтроллер	ATmega2560
Рабочее напряжение	5В
Входное напряжение (рекомендуемое)	7-12В
Входное напряжение (предельное)	6-20В
Цифровые Входы/Выходы	54 (14 из которых могут работать также как выходы ШИМ)
Аналоговые входы	16
Постоянный ток через вход/выход	40 mA
Постоянный ток для вывода 3.3 В	50 mA
Флеш-память	256 KB (из которых 8 KB используются для загрузчика)
ОЗУ	8 KB
Энергонезависимая память	4 KB
Тактовая частота	16 MHz

Входы/выходы Arduino Mega показаны на рисунке 24.

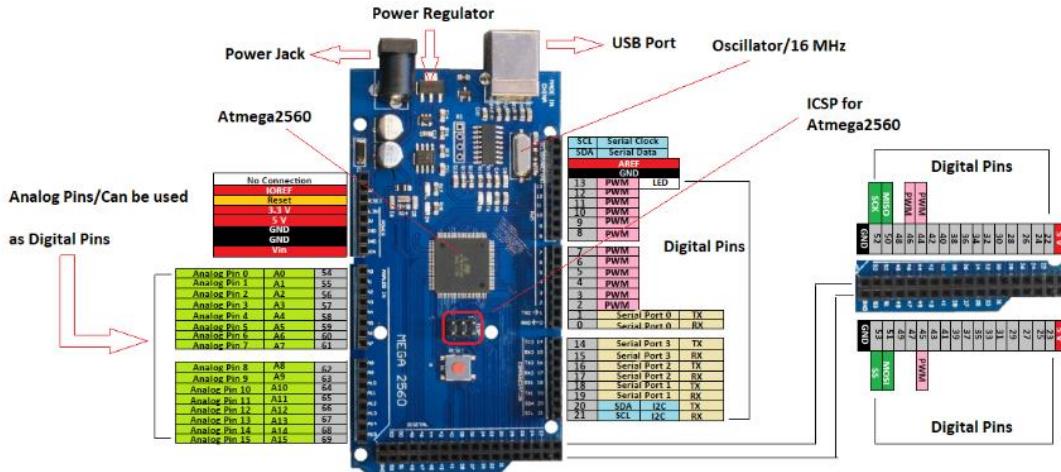
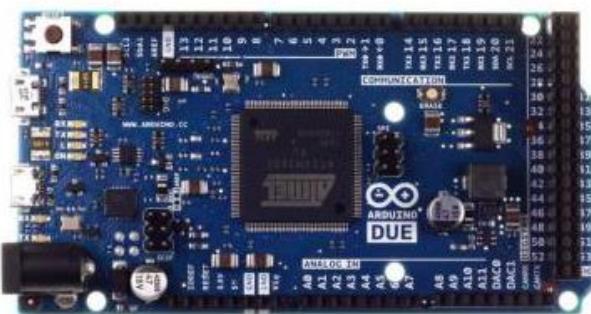


Рисунок 24 – Входы/выходы Arduino Mega

Arduino Due. Arduino Due (рисунок 25) – плата микроконтроллера на базе процессора Atmel SAM3X8E ARM Cortex-M3. Это первая плата Arduino на основе 32-битного микроконтроллера с ARM ядром. На ней имеется 54 цифровых вход/выхода (из них 12 можно задействовать под выходы ШИМ), 12 аналоговых входов, 4 UARTa (аппаратных последовательных порта), а генератор тактовой частоты 84 МГц, связь по USB с поддержкой OTG, 2 ЦАП (цифро-аналоговых преобразователя), 2 TWI, разъем питания, разъем SPI, разъем JTAG, кнопка сброса и кнопка стирания.



Arduino Due, вид спереди



Arduino Due, вид сзади

Рисунок 25 – Arduino Due

Плата содержит все, что необходимо для поддержки микроконтроллера. Чтобы начать работу с ней, достаточно просто подключить её к компьютеру кабелем микро-USB, либо подать питание с AC/DC преобразователя или батарейки. Due совместим со всеми платами расширения Arduino, работающими от 3,3 В, и с цоколевкой Arduino 1.0.

Расположение выводов Due повторяет цоколевку Arduino 1.0:

- 1) TWI: Выводы SDA и SCL расположены рядом с выводом AREF.
- 2) Вывод IOREF, который позволяет с помощью правильной конфигурации адаптировать присоединенную плату расширения к напряжению, выдаваемому Arduino. Благодаря этому платы расширения могут быть совместимы и с 3,3-вольтовыми платами типа Due и с платами на базе AVR, работающими от 5 В.
- 3) Неподключенные выводы, зарезервированные для использования в будущем.

Преимущества ядра ARM.

На Due установлено 32-битное ARM ядро, превосходящее по производительности обычные 8-битные микроконтроллеры. Наиболее значимые отличия:

32-битное ядро, позволяющее выполнять операции с данными шириной 4 байта за 1 такт (более подробную информацию смотри на странице int type).

- 1) частота процессора (CPU) 84 МГц.
- 2) 96 КБ ОЗУ.
- 3) 512 КБ флеш-памяти для хранения программ.
- 4) контроллер DMA, который разгружает центральный процессор от выполнения интенсивных операций с памятью.

Технические характеристики Arduino Due показаны в таблице 5.

Таблица 5 – Технические характеристики Arduino Due

Параметр	Значение параметра
Микроконтроллер	AT91SAM3X8E
Рабочее напряжение	3,3 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	54 (на 12 из которых реализуется выход ШИМ)
Аналоговые входы	12
Аналоговые выходы	2 (ЦАП)
Общий выходной постоянный ток на всех выходах/выходах	50 мА
Постоянный ток через вывод 3,3 В	800 мА
Постоянный ток через вывод 5 В	800 мА
Флеш-память	512 КБ доступно всего для пользовательских приложений
ОЗУ	96 КБ (два банка: 64 КБ и 32 КБ)
Тактовая частота	84 МГц

STM32F103RE. STM32 – это платформа, в основе которой лежат микроконтроллеры STMicroelectronics на базе ARM процессора, различные модули и периферия, а также программные решения (IDE) для работы с железом. Решения на базе stm активно используются благодаря производительности микроконтроллера, его удачной архитектуре, малом энергопотреблении, небольшой цене. В настоящее время STM32 состоит уже из нескольких линеек для самых разных назначений.

Семейство STM32 имеет широкий ассортимент изделий, различающихся по объему памяти, производительности, потреблению энергии и другим характеристикам.

Серии STM32F-1, STM32F-2 и STM32L полностью совместимы. Каждая из серий имеет десятки микросхем, которые можно без труда поменять на другие изделия. STM32F-1 была первой линейкой, ее производительность

была ограничена. Из-за этого по характеристикам контроллеры быстро догнали изделия семейства Stellaris и LPC17. Позднее была выпущена STM32F-2 с улучшенными характеристиками – тактовая частота достигала 120 МГц. Отличается высокой процессорной мощностью, которая достигнута благодаря новой технологии производства 90 нм. Линейка STM32L представлена моделями, которые изготовлены по специальному технологическому процессу. Утечки транзисторов минимальны, благодаря чему приборы показывают лучшие значения.

Важно отметить, что контроллеры линейки STM32W не имеют pin-to-pin совместимости с STM32F-1, STM32F-2 и STM32L. Причина заключается в том, что линейку разрабатывала компания, которая предоставила радиочастотную часть. Это наложило ограничения на разработку для компании ST.

STM32F103RE показан на рисунке 26.

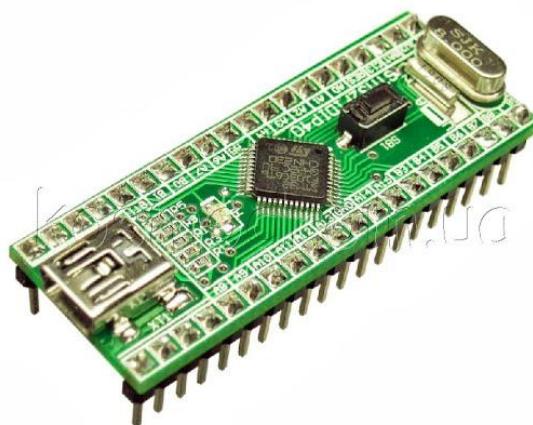


Рисунок 26 – STM32F103RE

Технические характеристики STM32F103RE показаны в таблице 6.

Таблица 6 – Технические характеристики STM32F103RE

Параметр	Значение параметра
Раздел	Микроконтроллеры широкого назначения
Ядро	ARM-Cortex-M3
Максимальная частота ядра	72 МГц
Объём памяти программ	512 кБайт
Тип памяти программ	Flash
Объём оперативной памяти	64 кБайт
Количество входов/выходов	51
Интерфейсы	CAN, I ² C, IrDA, LIN, SPI, UART/USART, USB
Периферия	DMA, Motor Control PWM, PDR, POR, PVD, PWM, Temp Sensor, WDT
АЦП/ЦАП	A/D 16x12b; D/A 2x12b
Напряжение питания	2...3.6 В

1.5 Прямая задача кинематики

Кинематика изучает движение тел без учета сил или моментов, которые вызывают движение. Кинематика робота относится к аналитическому исследованию движения робота-манипулятора. Разработка подходящих кинематических моделей для робота очень важна при анализе поведения промышленных манипуляторов.

Сформулируем прямую задачу кинематики. Прямая задача кинематики (ПЗК) – это вычисление положения и ориентации (X, Y, Z) рабочего органа манипулятора по его кинематической схеме и заданной ориентации ($q_1, q_2 \dots q_n$) его звеньев (n — число степеней свободы манипулятора, q — обобщенные

координаты (углы поворота)). Следует отметить, что решение для ПЗК единственно-возможное.

На практике гораздо чаще приходится решать обратную задачу кинематики.

1.6 Обратная задача кинематики

Среди задач управления роботом-манипулятором выделяется обратная задача кинематики (ОЗК) [4].

Для простоты понимания сути сформулируем ее упрощенно на конкретном примере. На рисунке 27 видна кинематическая схема манипулятора с 3 степенями свободы и 3 обобщенными координатами (q_1 , q_2 , q_3). Для решения ОЗК необходимо найти углы поворота каждого звена, чтобы схват (конец робота) манипулятора оказался в заданных положении и ориентации.

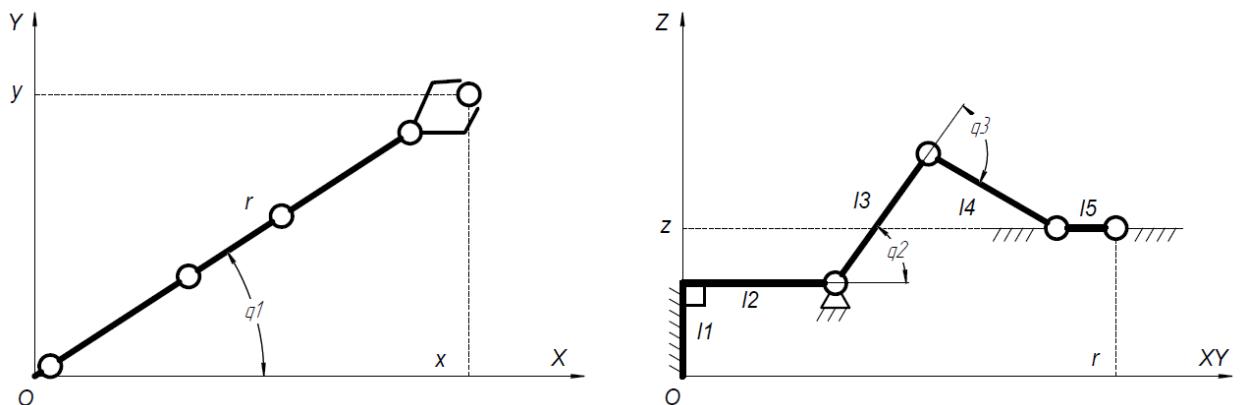


Рисунок 27 – Кинематическая схема манипулятора с 3 степенями свободы

Например, необходимо перемещать схват робота по прямой. В этом случае в каждый момент времени происходит решение ОЗК. Эта задача очень важна в робототехнике.

Теперь же сформулируем задачу более точно. Для этого введем понятие обобщенных координат. Ведь вместо угла поворота может быть изменение длины звена.

Для N-степенного манипулятора с кинематическими парами пятого класса вектором обобщенных координат \mathbf{q} называют вектор $(q_1, q_2, \dots, q_n)^T$, каждая компонента q_i которого является обобщенной координатой и представляет собой либо угол поворота, либо перемещение i-го звена относительно (i-1)-го звена [4].

Заметим, что при одном положении и ориентации схвата может быть несколько решений ОЗК. Выбор решения зависит прежде всего от ограничений, наложенных конструкцией робота на диапазон углов поворота или изменения длины звеньев робота. Например, по нашему примеру (рисунок 27) звено 13 не может поворачиваться как бы «вниз» от горизонтальной оси. Если остались варианты, то вводятся дополнительные ограничения. Например, на минимальное перемещение звеньев от предыдущего состояния.

Для манипуляторов без избыточности, то есть со степенями свободы схвата ≤ 6 , множество решений ОЗК конечно. С избыточностью множество решений ОЗК бесконечно много.

Как видно, обратная задача кинематики – это сложная задача, причем многое в ее решении зависит от степеней свободы манипулятора. Чем их больше, тем сложнее найти решение. Для решения ОЗК было разработано колоссальное количество методов. Все эти методы можно разделить на две группы:

- 1) Точные методы.
- 2) Приближенные (численные) методы.

Точные методы дают решение с абсолютной точностью, приближенные – с заранее заданной. Преимущество точных методов состоит в том, что решение ОЗК по этим методам находится быстрее, чем по приближенным.

Предпочтение отдают точным методам. Но иногда это невозможно. Например, для манипулятора кинематики с избыточностью, в таком случае применяют приближенные методы. Обычно промышленные манипуляторы изготавливают таким образом, чтобы их кинематика была без избыточности.

Для того, чтобы можно было применить точный метод и таким образом относительно быстро найти решение ОЗК. Поэтому в большинстве случаев можно видеть манипуляторы с максимальным количеством степеней свободы без избыточности – с 6 степенями свободы, где-то пишут с 6 осями.

Точные методы. Среди них выделяют: тригонометрический подход, метод обратных преобразований и другие. Суть тригонометрического подхода состоит в том, что по кинематической схеме находят зависимости и посредством тригонометрических формул находят решение ОЗК. По нашему примеру: звенья 13, 14 можно дополнить до треугольника и таким образом из треугольника с известными сторонами по теореме косинусов найти нужные углы. Заметим, что для манипуляторов с простой кинематикой, как в нашем примере, ОЗК по этому методу находится очень просто, интуитивно понятно.

Метод обратных преобразований. Для каждого звена вводится своя система координат. Для этого используют представление Денавита-Хартенберга [4]. Составляются матрицы перехода из системы координат одного звена в систему координат соседнего звена. И последовательно перемножая эти матрицы находят матрицу перехода из начальной (абсолютной) системы координат в СК конечного звена. Для решения ОЗК будем двигаться как бы наоборот.

Матрица T_N определяет положение и ориентацию схвата:

$$T_N = A_1 \cdot A_2 \cdot \dots \cdot A_N, \quad (1)$$

где $A_i = A_i(q_i)$ – матрицы перехода от i-й к (i-1)-й системе координат манипулятора.

Умножив на $A_i^{-1}(q_i)$ слева и справа получим:

$$A_i^{-1}(q_i) \cdot T_N = A_2 \cdot \dots \cdot A_N. \quad (2)$$

Из полученных соотношений находят q_1 , и так далее для q_2, q_3, \dots, q_N .

Приближенные методы. Среди них выделяют: метод ньютона [1], метод конечного поворота и смещения [5] Бикватернионное решение [6], Mesh-based Inverse Kinematics [7], метод FABRIK [8].

Не будем останавливаться здесь на этих методах. Рассмотри метод FABRIK. Этот метод великолепно работает для манипуляторов с избыточностью и его довольно просто понять. Имеется какое-то начальное положение манипулятора. Считываются обобщенные координаты когда манипулятор как бы «пришел» в нужное положение. Его конец находится в нужном положении, но начальное звено сдвинулось. Далее манипулятор как бы «переходит обратно». Однако теперь его конечное звено ближе к заданному положению чем изначально. И так до тех пор, пока это несоответствие, эта ошибка, не будет удовлетворять заданному. То есть мы делаем переходы прямо и назад. Отсюда и название FABRIK – Forward Backward Inverse Kinematics.

Подведя итог, можем сказать, что для простых манипуляторов предпочтительно использовать тригонометрический подход, для сложных без избыточности – метод обратных преобразований. Для манипуляторов с избыточностью хорошо подходит метод FABRIK.

1.7 Выводы по разделу

Таким образом, в данном разделе провели анализ предметной области. Сделаем следующие выводы:

Объектом разрабатываемой системы управления является манипуляционный робот «Робко 01», имеющий 5 степеней подвижности. Питание для его униполярных шаговых двигателей имеет следующие требования: напряжение 12 В, сила тока до 2.5 А.

Биполярные ШД существенно превосходят униполярные благодаря на 40% большей мощности (используется не пол обмотки, а целая часть), однако за это достоинство приходится расплачиваться сложностью драйвера управления ШД.

Драйверы шаговых двигателей в широком потреблении относятся к с ШИМ-драйверам и имеют интерфейс управления STEP/DIR. В разделе приведено описание приближенно подходящих для системы управления драйверов: ULN2003a, A4988, DRV8852, L298N.

На рынке представлены следующие типы датчиков: оптический, индукционный, емкостный, датчик холла, электромеханический. Самые малые размеры имеют датчики холла и электромеханические микровыключатели. В разделе приведено описание приближенно подходящих для системы управления датчиков: KLS-A5.15S, KLS-A5.11G2, KLS-A5.15X, EziUsin, TLE4905LHALA1.

Широко распространены микроконтроллеры: Atmel AVR, PIC (Microchip), STM8 (STMicroelectronics). В разделе приведено описание приближенно подходящих для системы управления микроконтроллеров: Arduino Uno, Arduino Mega 2560, Arduino Due, STM32F103RE.

2 Разработка аппаратной части системы управления

2.1 IDEF0 для описания модели процесса разработки и реализации системы управления

Для определения плана работы воспользуемся нотацией IDEF0.

Определим входные и выходные составляющие всего проекта (рисунок 28).

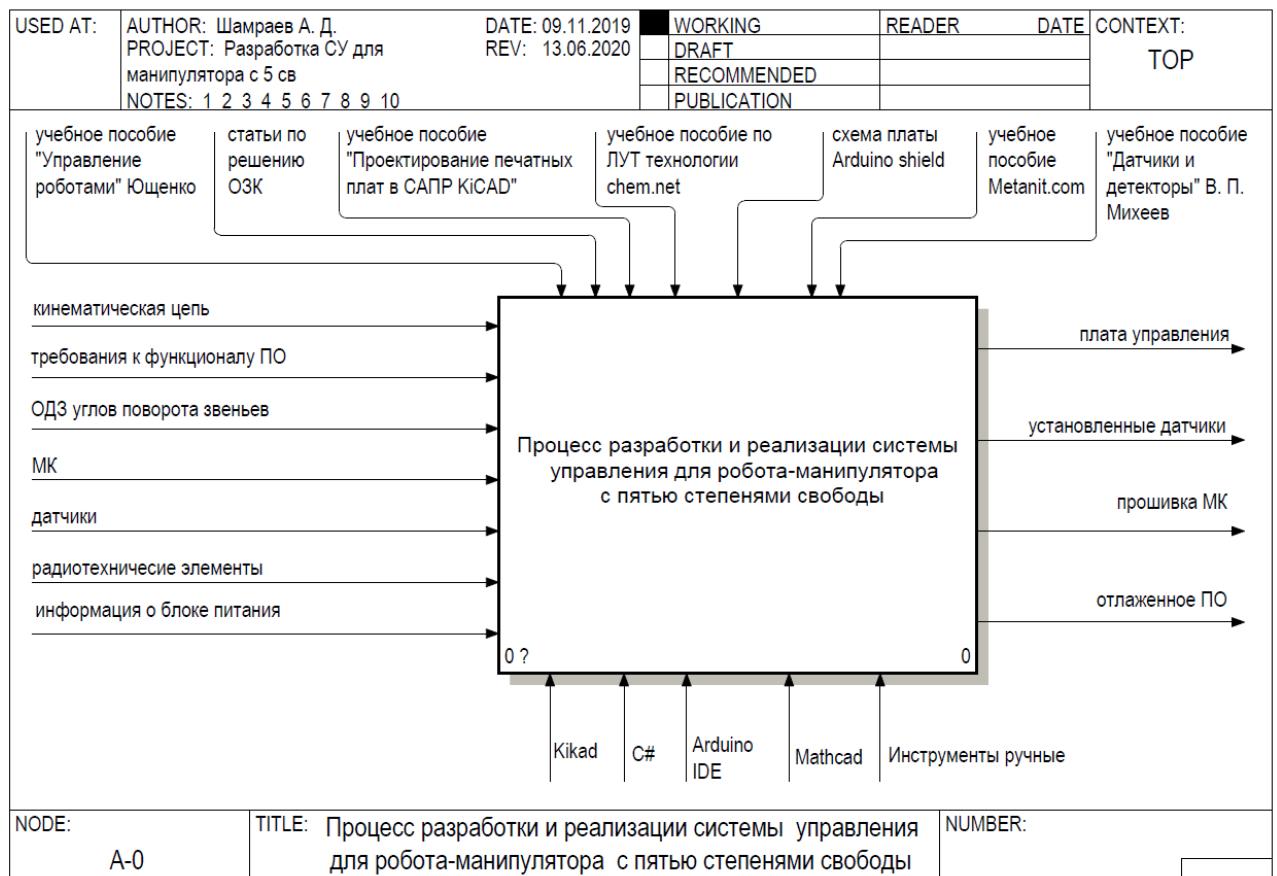


Рисунок 28 – IDEF0 уровень 0

Декомпозирируем уровень, как показано на рисунках 29, 30.

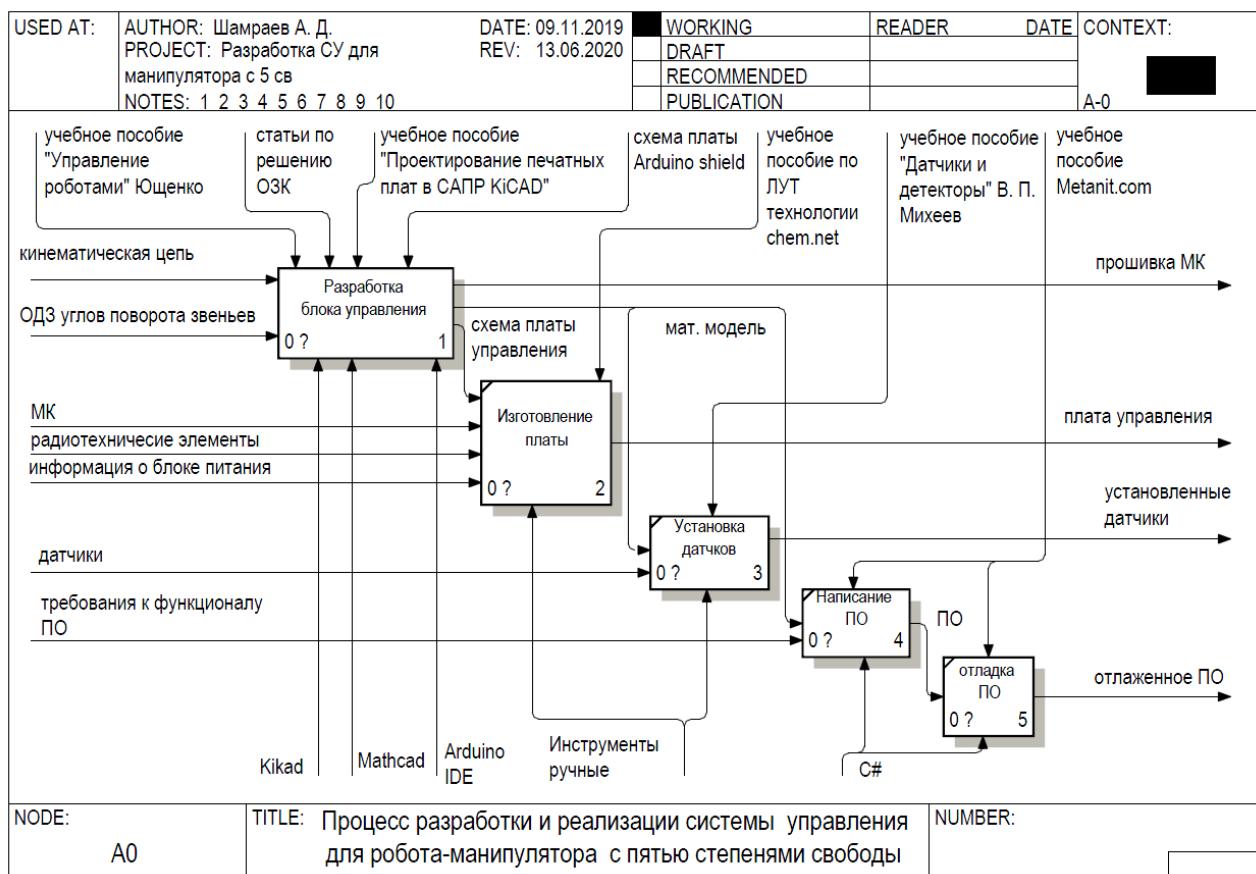


Рисунок 29 – IDEF0 уровень 1

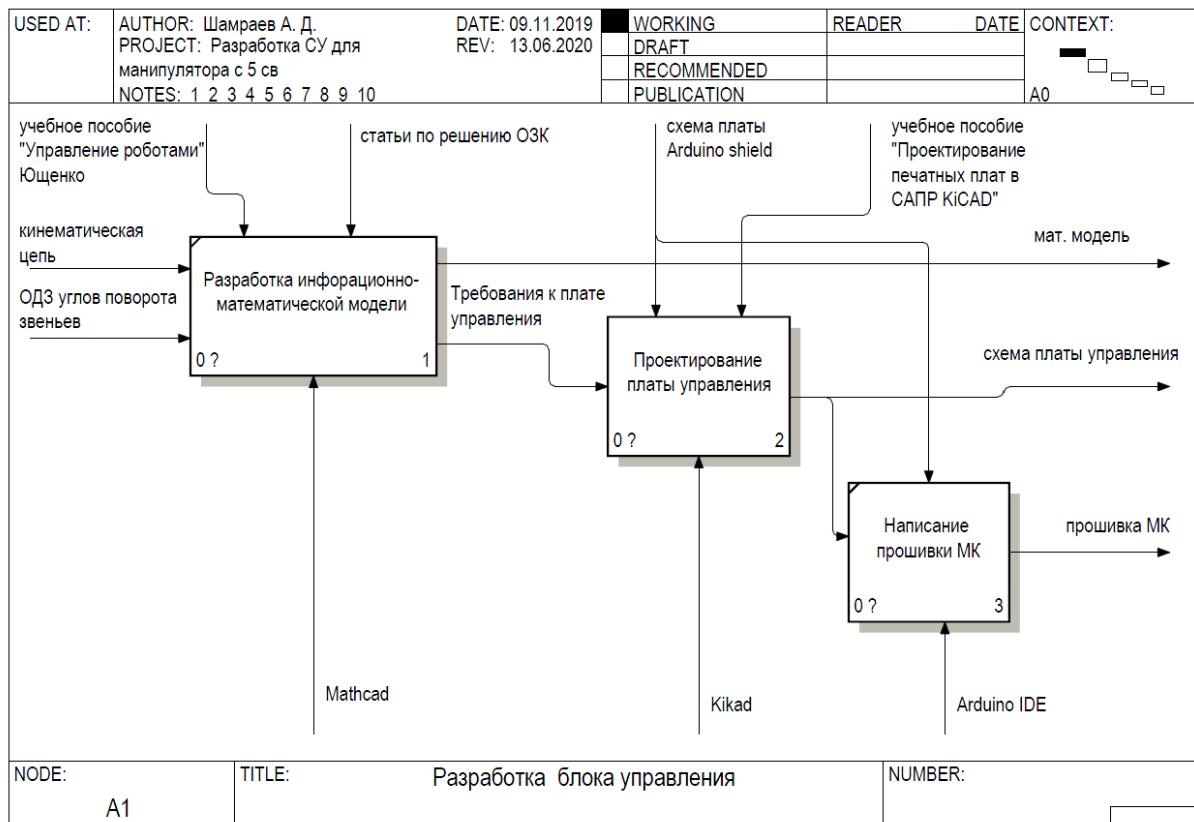


Рисунок 30 – IDEF0 уровень 2

В работе используется САЕ Kicad на этапе проектирования платы, среды разработки Visual Studio C# на этапе написания ПО, Arduino IDE на этапе написания прошивки микроконтроллера.

2.2 Разработка структурной схемы платы управления

Разрабатываем структурную схему системы управления роботом манипулятором. Она состоит из следующих элементов: шаговых двигателей (ШД), драйверов шаговых двигателей, микроконтроллера (посыпает сигналы драйверам), контроллера (ПК посыпает необходимые углы поворота ШД), концевых датчиков. Схема показана на рисунке 31.

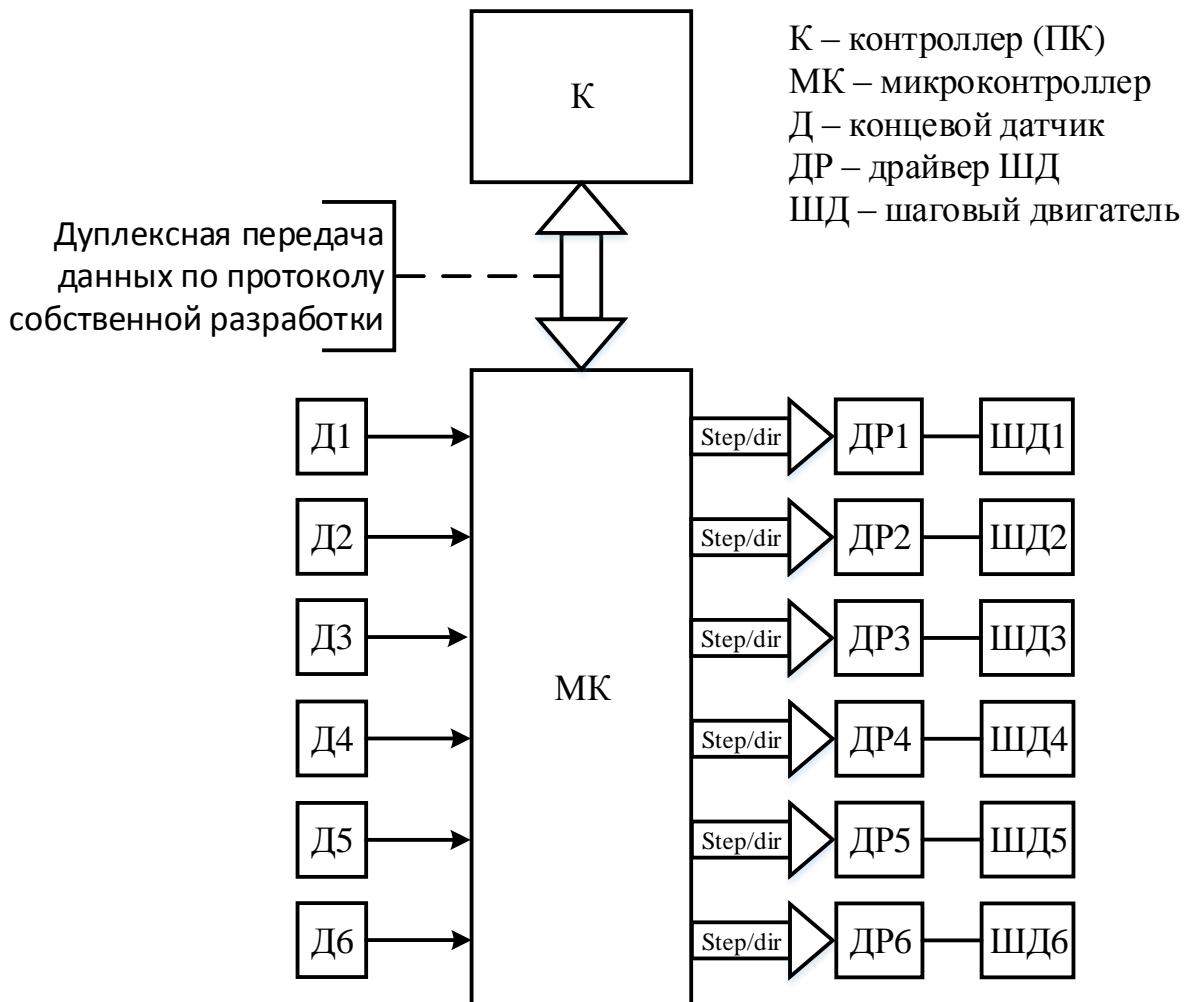


Рисунок 31 – Структурная схема системы управления роботом манипулятором

2.3 Выбор драйвера для управления шаговым двигателем

Среди технических характеристик «Робко 01», показанных в разделе 1, выделим следующие:

- 1) напряжение питания – 12 В;
- 2) потребление – до 2.5 А;

«Робко 01» разработан в 1986 году, в то время, когда не было широкой возможности управления ШД в биполярном режиме. Поэтому питание шаговых двигателей (ШД) рассчитано на управление ими в униполярном режиме.

Однако на сегодняшний день драйверы ШД для управления ими в биполярном режиме широко распространены. Учитывая большие преимущества (мощность выше на 40%) биполярных ШД перед униполярными, изложенными в 1 разделе, примем решение об управлении ШД в биполярном режиме.

Раз ШД управляются в биполярном режиме, то вместо половины (как раньше) обмотки используется целая, следовательно, напряжение питания увеличится в 2 раза – 24 В.

Замерим ток, проходящий через обмотку ШД при напряжении питания 24 В – 0.44 А. Следовательно, ток протекающий через драйвер (с запасом) 1 А.

Плата управления будет располагаться в самом «Робко 01», как показано на рисунке 3.

Поэтому драйверы должны быть подходящих размеров – не более 50x50x20.

В итоге к драйверу ШД следующие требования:

- 1) Максимальный ток на фазу – 1 А (>1А допустимо);
- 2) Максимальное напряжение – 24 В (>24 В допустимо);
- 3) Управление биполярных ШД;
- 4) Максимальные габаритные размеры – 50x50x20;

Произведём выбор драйвера для системы управления исходя из заданных требований. Имеем для выбора драйверы: ULN2003a, A4988, DRV8852, L298N.

Уберем из рассмотрения ULN2003a, так как ULN2003a производит управление только униполярных ШД. Также не выполняются требования у L298N – габаритные размеры (43.5x 43.2x 29.4) больше требуемых.

Окончательный выбор сделаем на основе критерия минимальной цены.

Цена на сайте iarduino.ru на 26.03.2020: A4988 – 185 р., DRV8852 – 286 р.

Таким образом для разрабатываемой системы управления будем использовать драйвер A4988.

2.4 Выбор датчиков

Для определения начального положения звеньев робота необходимо использование датчиков. Необходим концевой датчик.

Существуют следующие типы датчиков: оптический, индукционный, емкостный, датчик холла, электромеханический.

Рассмотрим оптимальные типы датчиков для разрабатываемой системы управления.

Не подойдет оптический тип датчика по причине его высокой чувствительности к освещенности. Ведь робот будет находиться в различных помещениях (выставках и пр.) с различными параметрами освещенности.

В итоге к концевому датчику следующие требования:

- 1) Максимальные габаритные размеры – 30x30x20;
- 2) Цена <100 р.

Емкостные датчики имеют высокую стоимость (>100 р.), поэтому уберем их из рассмотрения.

Индуктивные датчики имеют габариты больше требуемых.

Остаются следующие типы датчиков: датчики холла, электромеханические.

Для электромеханических датчиков введем дополнительное требование:

- 3) Вид наконечника – круглый.

Произведём выбор датчика для системы управления исходя из заданных требований. Имеем для выбора следующие датчики: KLS-A5.15S, KLS-A5.11G2, KLS-A5.15X, EziUsin, TLE4905LHALA1.

Кроме EziUsin, TLE4905LHALA1 ни один из датчиков не выполняет требования максимальных габаритов.

Окончательный выбор сделаем на основе критериев минимальной цены и простоты монтажа.

Цена на сайте <https://aliexpress.ru/> на 26.03.2020: EziUsin 5A 125 250V SPDT – 15 р.

Цена на сайте <https://www.chipdip.ru/> на 26.03.2020: TLE4905LHALA1 – 50 р.

Электромеханический датчик значительно проще монтировать, чем датчик Холла, так как для последнего требуется, помимо всего прочего, еще и магнит.

Таким образом для разрабатываемой системы управления будем использовать электромеханический датчик EziUsin.

2.5 Выбор микроконтроллера

Драйвер A4988 управляет по протоколу STEP/DIR, логические сигналы 5 В. Следовательно, необходимо 2 пина для управления A4988. В «Робко 01» всего 6 двигателей, следовательно, необходимо 6 драйверов. Также нужен 1 pin для включения всех драйверов (Enable). Тогда пинов для управления драйверами: $2 * 6 + 1 = 13$.

Для электромеханического датчика требуется питание (любое, т.к. работает по принципу кнопки) и 1 pin для приема сигнала. Всего 6 двигателей, следовательно, необходимо 6 датчиков. Для всех датчиков питание будем осуществлять с помощью 2 пинов (хватает и 1, но для разгрузки линии выберем 2). Тогда пинов для работы с датчиками: $1 * 6 + 2 = 8$.

Для возможности расширения функционала системы управления, а, следовательно, и увеличения количества устройств управления, выделим еще 10 пинов.

В итоге к плате с микроконтроллером следующие требования:

- 1) Минимальное количество пинов – 31;
- 2) Максимальные габаритные размеры – 130x100x20;
- 3) Логическое питание – 5 В;

Произведём выбор МК для системы управления исходя из заданных требований. Имеем для выбора следующие МК: Arduino Uno, Arduino Mega 2560, Arduino Due, STM32F103RE.

Не выполняется требование к минимальному количеству пинов у Arduino Uno (их всего 14). Также не выполняются требования у STM32F103RE – логическое питание [3.6 В] меньше требуемого [5 В].

Окончательный выбор сделаем на основе критерия минимальной цены.

Цена на сайте iarduino.ru на 26.03.2020: Arduino Mega 2560 – 1370 р., Arduino Due – 1821 р.

Таким образом для разрабатываемой системы управления будем МК Arduino Mega 2560.

2.6 Разработка схемы электрической принципиальной

Разработанная схема электрическая принципиальная системы управления роботом-манипулятором показана на рисунке 32. На ней изображены драйверы ШД A4988, конденсаторы 100 мкФ (рекомендовано производителем драйверов, как показано в разделе 1) для сглаживания напряжения питания к драйверам. Так же на схеме показаны стягивающие резисторы 10 кОм для предотвращения ложного срабатывания датчиков, то есть защиты от наводок. На схеме показана и сама плата Arduino Mega.

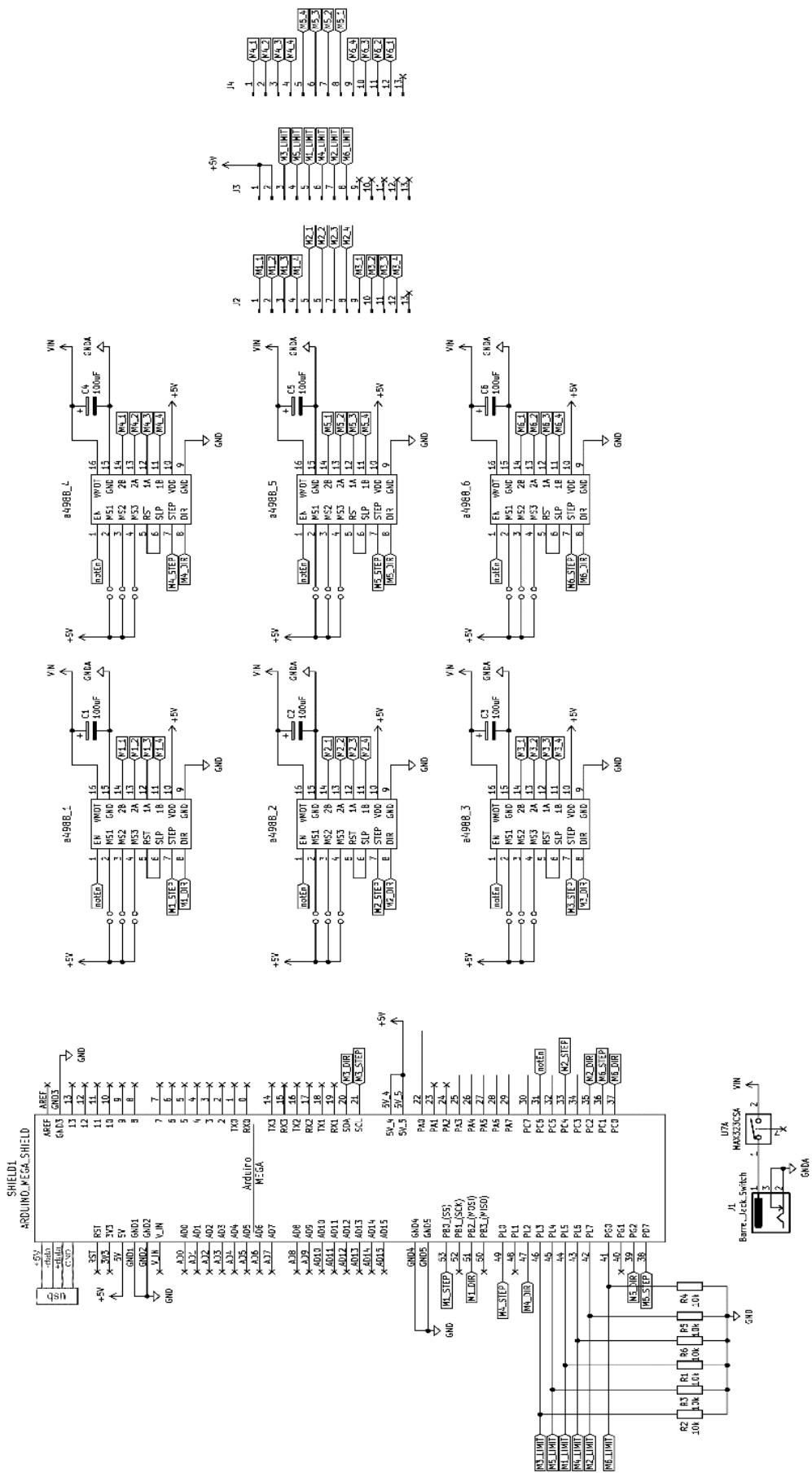


Рисунок 32 – Схема электрическая принципиальная системы управления роботом-манипулятором

2.7 Изготовление платы

Произведем разводку дорожек платы управления в KiCad. Ввиду значительного количества выводов на плате будем разводить на два слоя.

Прототип будет изготавливаться вручную, поэтому необходимо развести так, чтобы неточность позиционирования верхнего и нижнего слоев платы не повлияла на ее работоспособность. Для этого все дорожки, соединяющие драйверы и МК, будут на одном слое, а силовые линии (соединяющие драйверы и ШД) на другом.

Разведенная плата управления показана на рисунках 33, 34.

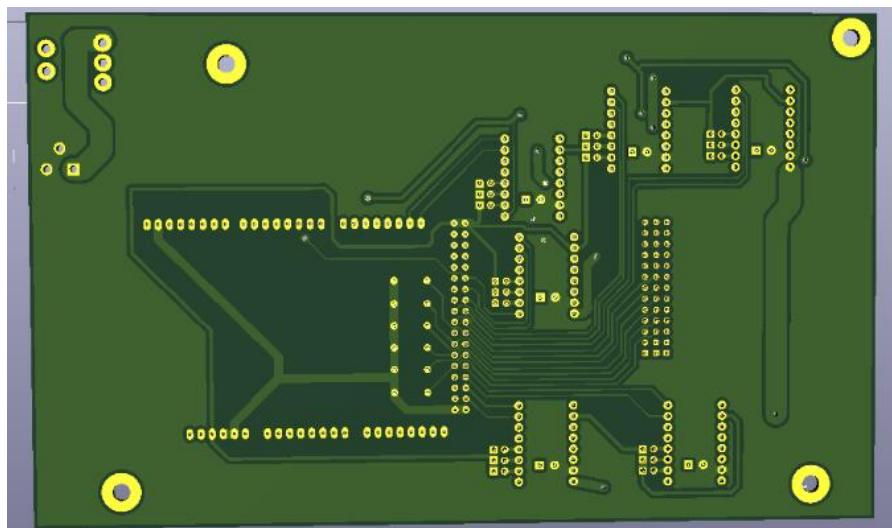


Рисунок 33 – Плата управления в KiCad вид снизу

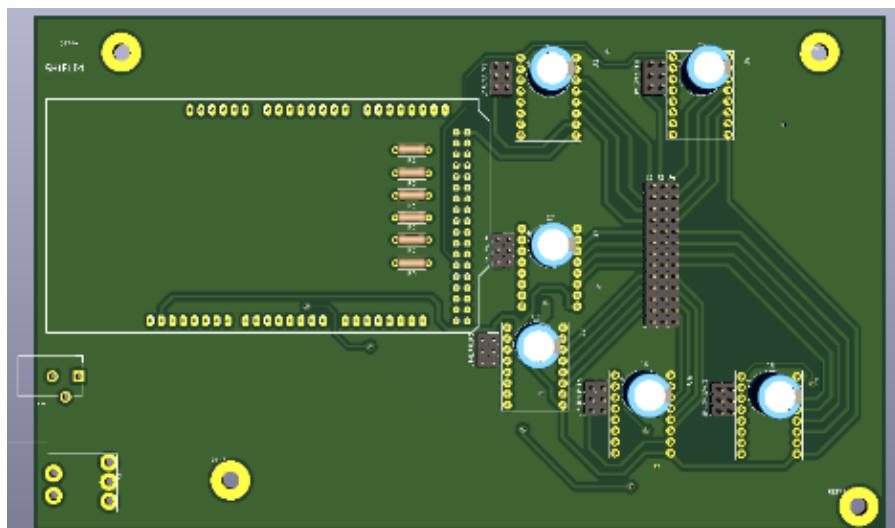


Рисунок 34 – Плата управления в KiCad вид сверху

На рисунках 35-48 показано изготовление платы.



Рисунок 35 – Подготовленная часть стеклотекстолита для печатной платы СУ

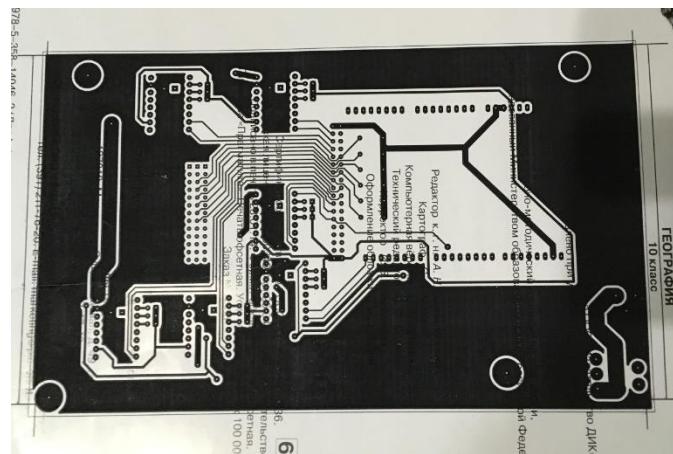


Рисунок 36 – Результат печати нижнего слоя платы управления на глянцевой бумаге

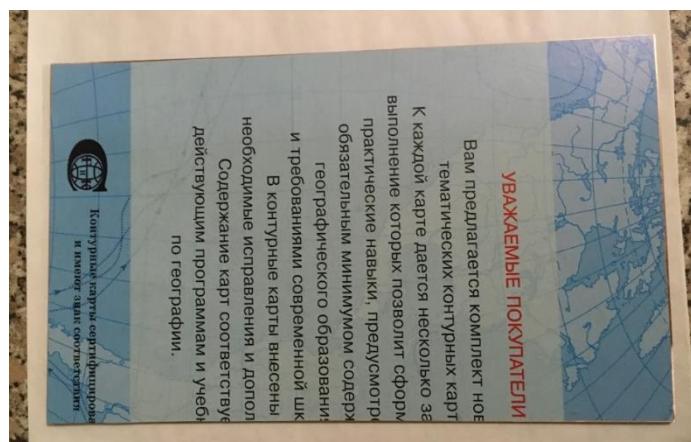


Рисунок 37 – Перевод тонера на стеклотекстолит



Рисунок 39 – Вымачивание бумаги на стеклотекстолите

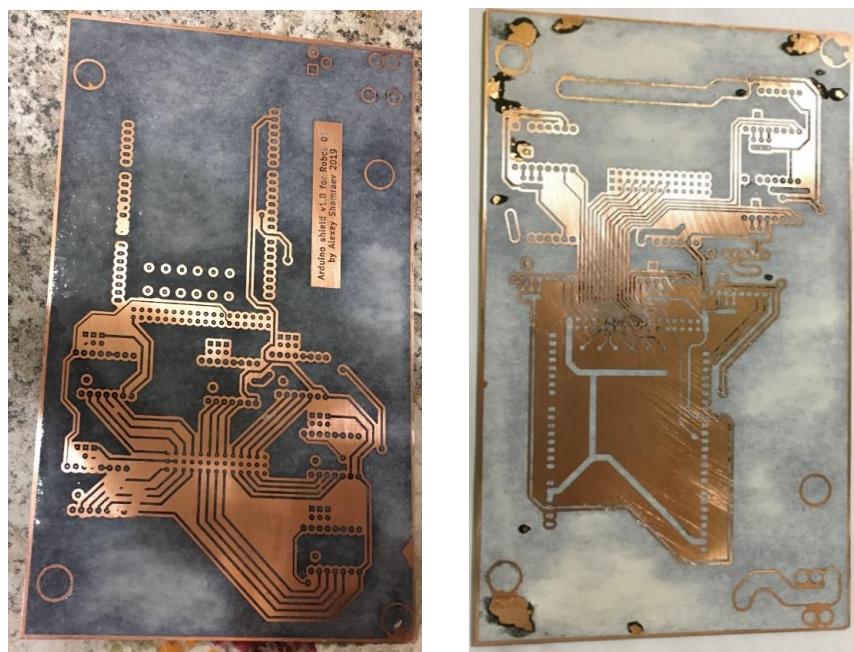


Рисунок 38 – Результат перевода тонера верхнего (слева) и нижнего (справа) слоев платы на стеклотекстолит

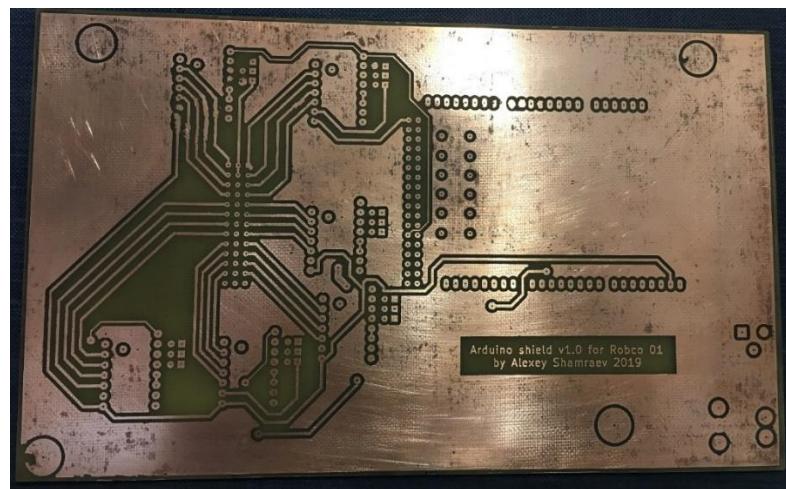


Рисунок 40 – Результат травления верхнего слоя платы

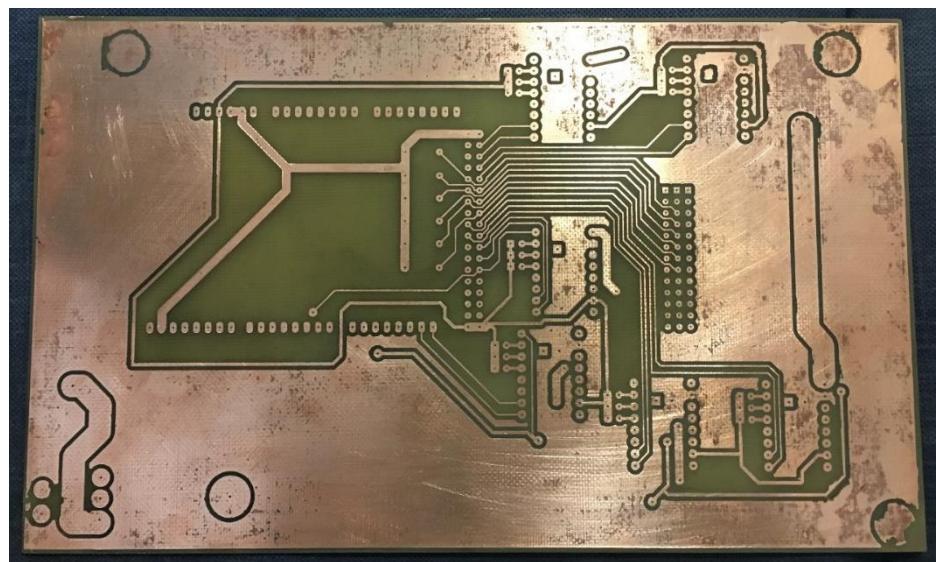


Рисунок 41 – Результат травления нижнего слоя платы

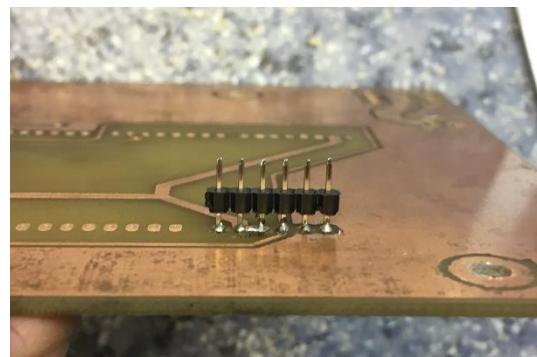


Рисунок 42 – Пайка контакта с двух слоев (вид с нижнего слоя)

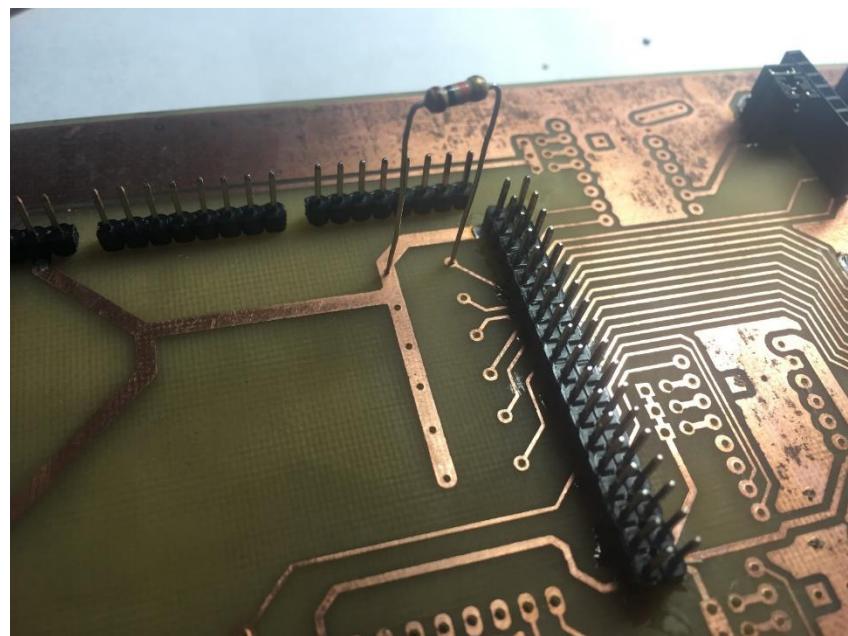


Рисунок 43 – Монтируем стягивающего резистора

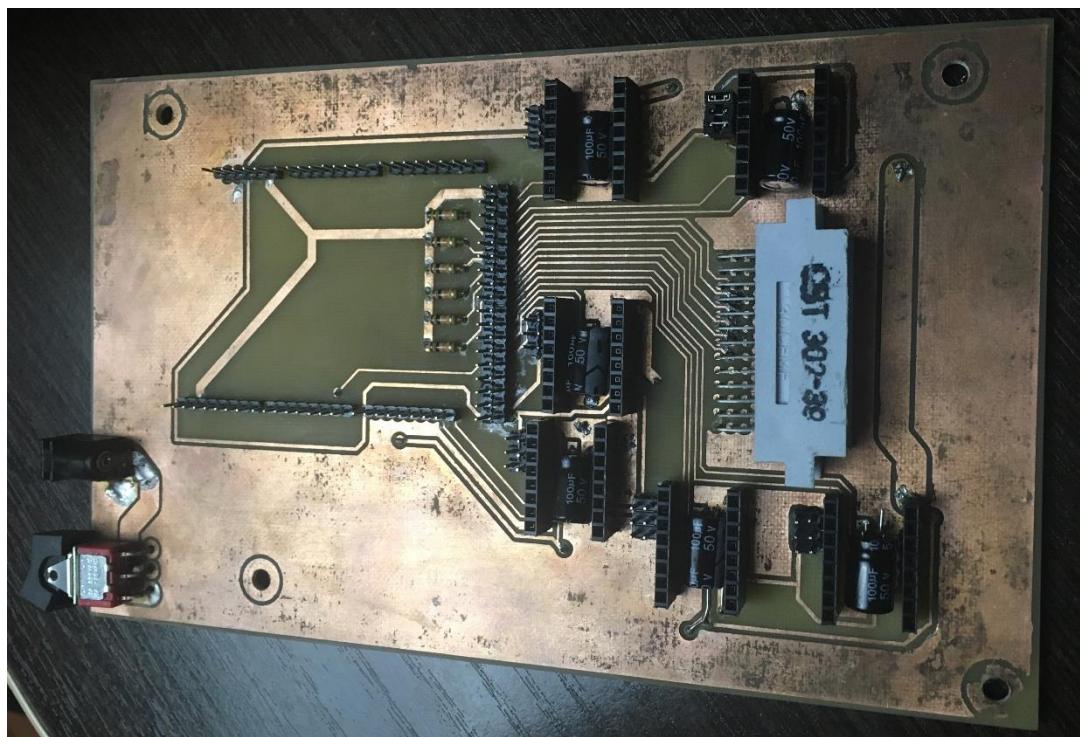


Рисунок 44 – Результат пайки элементов: вид нижнего слоя платы

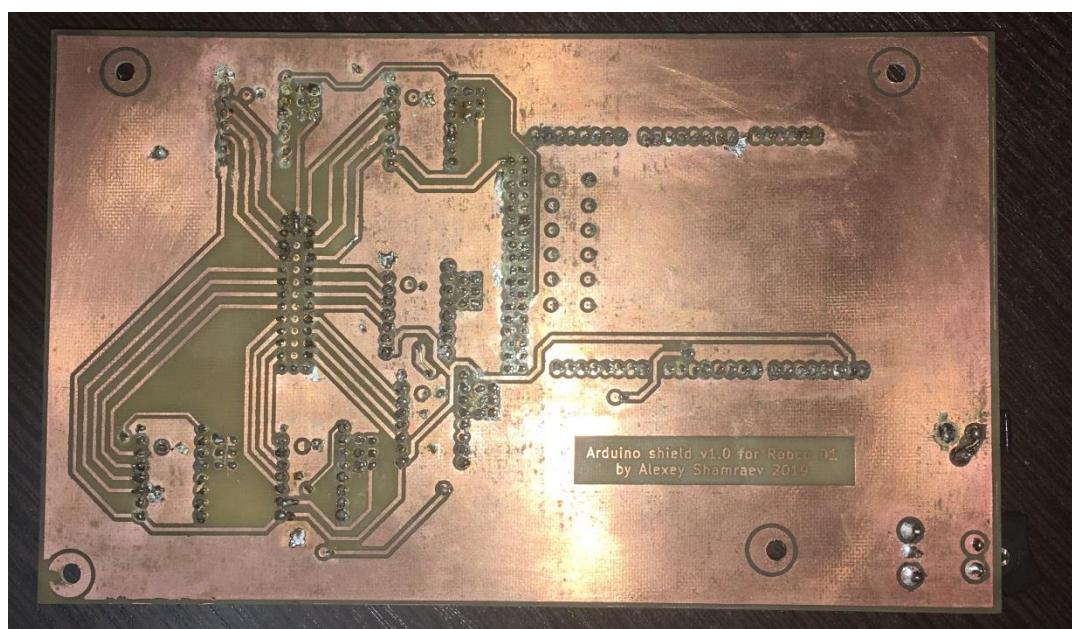


Рисунок 45 – Результат пайки элементов: вид верхнего слоя платы

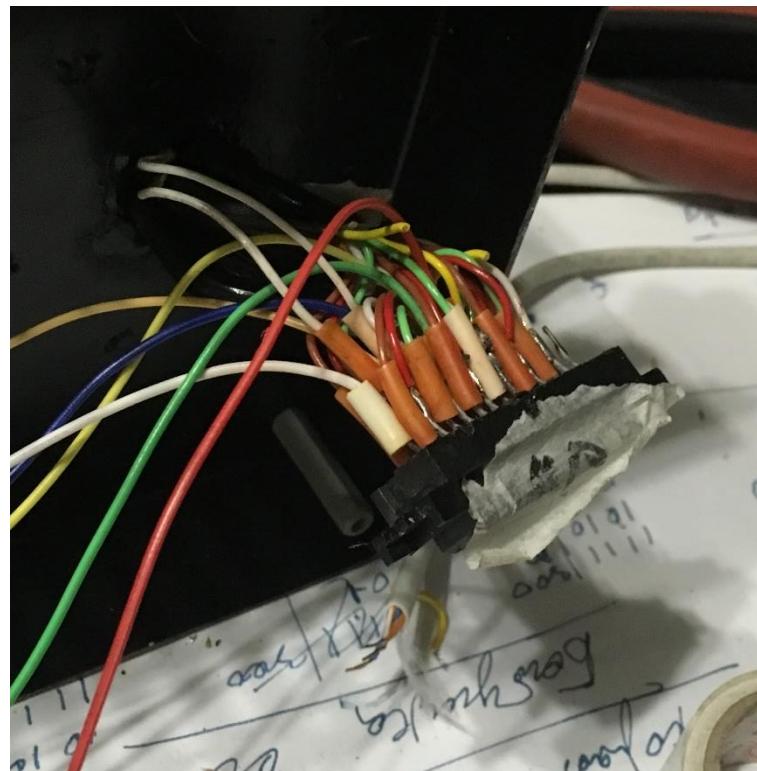


Рисунок 46 – результат пайки проводов для датчиков в контакт робота

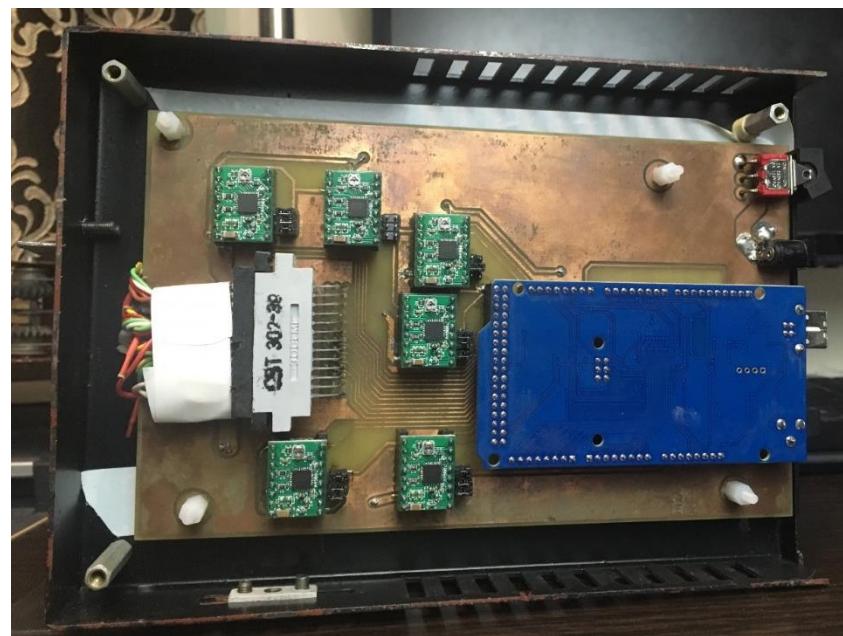


Рисунок 47 – Монтированная плата в «Робко 01», вид снизу



Рисунок 48 – Монтированная плата в «Робко 01», вид спереди

2.8 Установка датчиков

На рисунках 49, 50 показан процесс монтажа датчиков – микроподвижных выключателей.



Рисунок 49 – Нарезание резьбы под крепежные винты датчиков

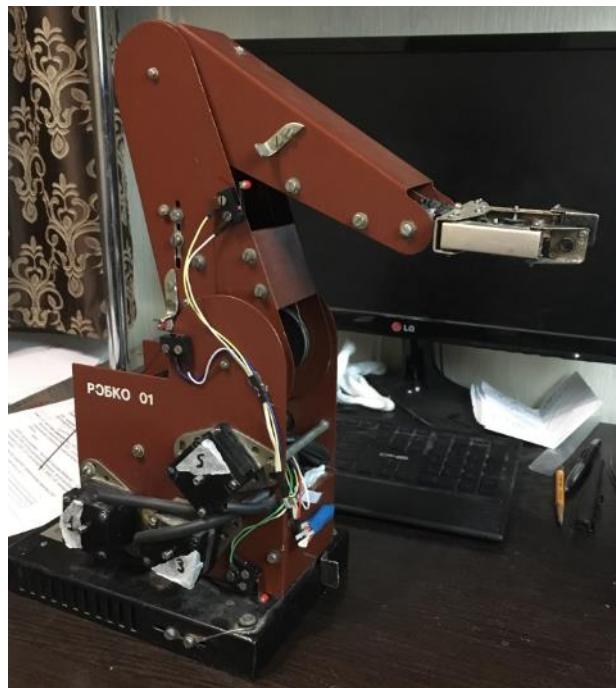


Рисунок 50 – «Робко 01» с установленными датчиками

2.9 Выводы по разделу

Таким образом, в данном разделе привели описание разработки аппаратной части системы управления. Сделаем следующие выводы:

Разработанная структурная схема состоит из следующих элементов: шаговых двигателей (ШД), драйверов шаговых двигателей, микроконтроллера (посыпает сигналы драйверам), контроллера (ПК посыпает необходимые углы поворота ШД).

Выбор элементов разрабатываемой системы управления имеет следующий результат: драйвер шагового двигателя A4988, электромеханический микровыключатель EziUsin, микроконтроллер Arduino Mega 2560.

Также в этом разделе приводится само изготовление платы управления, установка датчиков в «Робко 01».

3 Разработка программной части системы управления

3.1 Разработка архитектуры программной части системы управления

Язык верхнего уровня системы управления. Существует три основных подхода к описанию синтаксиса языка программирования для системы управления:

- 1) Использование команд с параметрами для задания траекторий и точек перемещения (ABB, Yaskawa, Motoman),
- 2) Использование структурированного представления данных (KUKA),
- 3) Использование кодовых команд (Motorola).

Для задания перемещений используются следующие команды:

- произвольное перемещение между двумя точками (point, ptp);
- перемещение между двумя точками по линейной траектории (linear);
- перемещение по окружности с заданием определённого набора точек

(начальная координата, центр окружности, конечная координата или радиус) (circle) [9];

Самый емкий по памяти из представленных трех подходов – это кодовые команды (Motorola). Однако самый понятный – подход структурированного представления данных (KUKA). В разрабатываемой системе будет применен этот подход. Для удобства работы с СУ будут обрабатываться G-коды. Их легко получить с помощью бесплатных CAM–систем, что немаловажно для образовательных целей.

Блок управления. В блок управления входят такие компоненты: планировщик траектории, блок расчета прямой и обратной задач кинематики, система обработки управляющих программ [10].

Задачей планировщика траектории является расчёт основных видов траекторий, используемых в работе МР: линейных, круговых [11]. Для их реализации требуются соответствующие интерполяторы, взаимодействующие с блоком расчёта кинематики МР [9].

Контроллер верхнего уровня обрабатывает управляющую программу как показано на рисунке 51.



Рисунок 51 – Структура модуля управляющей программы

Протокол обмена данными между контроллером верхнего уровня и микроконтроллером.

При разработке системы управления (СУ) манипулятором появляется задача разработки и протокола. Система управления разрабатывается для «Робко 01». Протокол нужен для обмена данными между контроллером верхнего уровня (персональным компьютером) и микроконтроллером mega 2560. На компьютере считается обратная задача кинематики, планируется путь и другие задачи. Компьютер отправляет задание, в котором содержатся углы поворота каждого звена манипулятора, через определенный промежуток времени, а также иную информацию. Микроконтроллер эту информацию получает, обрабатывает и посыпает соответствующие сигналы для выполнения задания (импульсы для драйверов шаговых двигателей и прочее). Так как большую часть вычислений берет на себя контроллер верхнего уровня, то ему необходимо очень часто посылать задания микроконтроллеру. Например, для движения схвата по прямой линии контроллер верхнего уровня считает контрольные точки через необходимый интервал, углы, соответствующие этим

точкам. Далее контроллер верхнего уровня эти углы посыпает микроконтроллеру, а тот поворачивает звенья манипулятора, выдерживая эти углы. Промежутки времени между этими посыпаемыми углами может быть довольно короткими (<0.01 с.). Поэтому становится важным передавать данные尽可能 быстро, минимизировав задержку между заданиями. За основу разрабатываемого протокола был взят протокол Modbus [12] [13].

Перед тем, как приступить к описанию нового протокола обозначим данные которые по нему будут передаваться. Рассматриваемый Манипулятор «Робко 01» имеет 5 степеней свободы схватка, способен сжимать и разжимать схват. В текущей версии ПО управление производится заданием трех углов: q1, q2, q3. Всего 3 числа. Максимальное число: 360,99 °, минимальное: -360,99 °. Необходимо 4 байта на одно число типа float. Получается 3*4=12 байт полезной информации, payload.

Посмотрим на формат пакета нашего протокола (рисунок 52).

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Рисунок 52 – Формат пакета разрабатываемого протокола

Обозначим содержание ячеек пакета (рисунок 52):

- 1 – begin byte, стартовый байт, 1 байт;
- 2 – request byte, байт запроса (ответа), 1 байт
- 3 – length byte, байт длины payload, 1 байт;
- 4 – operation code, байт кода операции, 1 байт;
- 5 – status code, байт кода статуса, 1 байт;
- 6 – payload, байты полезной информации, 12 байт;
- 7 – crc, байты контрольной суммы, 2 байта.

Разберем формат этих ячеек пакета.

- 1 – begin byte = 1;
- 2 – request byte:
REQUEST = 0,
- RESPONSE = 1;
- 4 – operation code:

NONE = 0,
MOVE_TO_ABSOLUTE_ANGLES_Q1Q2Q3 = 1,
FIND_AND_GO_TO_ZEROS = 2,
GRIPPER_GRIP = 3 (схват сжать до срабатывания датчика),
GRIPPER_UNGRIP = 4 (схват разжать до срабатывания датчика),
GRIPPER_OPEN_TO_ABSOLUTE_DISTANCE = 5, (схват сжать губки на определенное расстояние между ними);

5 – status code:

NONE = 0,
ERROR = 1,
DONE = 2;

В ячейку 6 – Payload заносится полезная информация, а в ячейку 7 – crc заносятся байты контрольной суммы (исходя из ранее сформированных ячеек).

Контрольная сумма (crc) считается следующим образом: в цикле производят операцию логического исключающего ИЛИ (XOR) над двумя байтами – байтом пакета данных (без crc) и байтом контрольной суммы (если четный байт пакета данных – то crc[0], иначе crc[1]). То есть имеем 17 байтовый пакет frame (так как без crc) и 2 байтовую crc:

$crc[0] = crc[0] \text{ XOR } frame[index]$, если четный index;
 $crc[1] = crc[1] \text{ XOR } frame[index]$, если нечетный index.

Стоит отметить, что очень важно проверять целостность пакета данных.

При скорости обмена данными 115200 бод (бит/с) ошибки при передаче данных могут возникать. Контрольная сумма (crc) помогает определить целостен ли пакет, если произошла ошибка при передаче данных, то МК запрашивает повторное отправление данных.

Всего пакет занимает 19 байт. При скорости 115200 бод (бит/с) передача одного пакета займет 1,3 мс.

Приведем пример пакета данных – задание роботу переместиться в абсолютные углы (-2.23, 0.07, 0.56):

(1 0 19 1 0 82 184 14 192 41 92 143 61 41 92 15 63 104 221).

Суммируя вышеизложенное скажем, что разрабатываемая система управления роботом «Робко 01» имеет: язык верхнего уровня – структурный, внедрены G-коды. Блок управления содержит: планировщик траектории, блок расчета прямой и обратной задач кинематики, систему обработки управляющих программ. Протокол: на основе Modbus RTU, 7 секторов, 19 байта в одном пакете.

3.2 Решение прямой задачи кинематики методом Денавита-Хартенберга

Кинематическая схема изображена на рисунках 53, 54.

Введем системы координат в каждом сочленении и в определенной точке схваты.

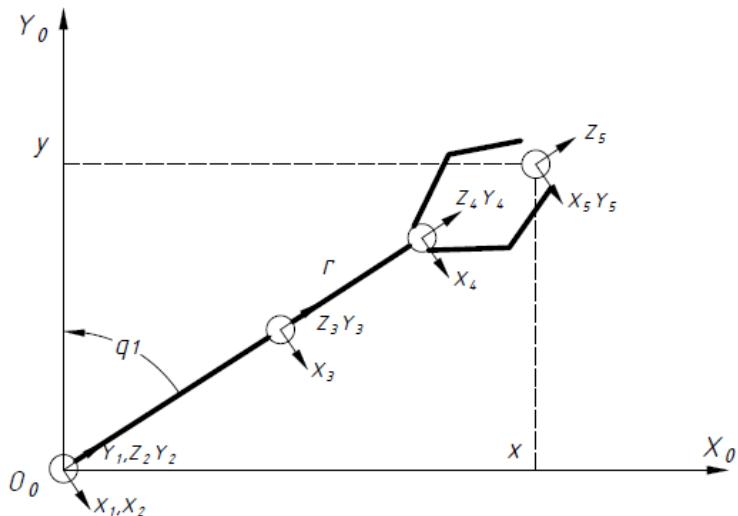


Рисунок 53 – Кинематическая схема робота – вид сверху

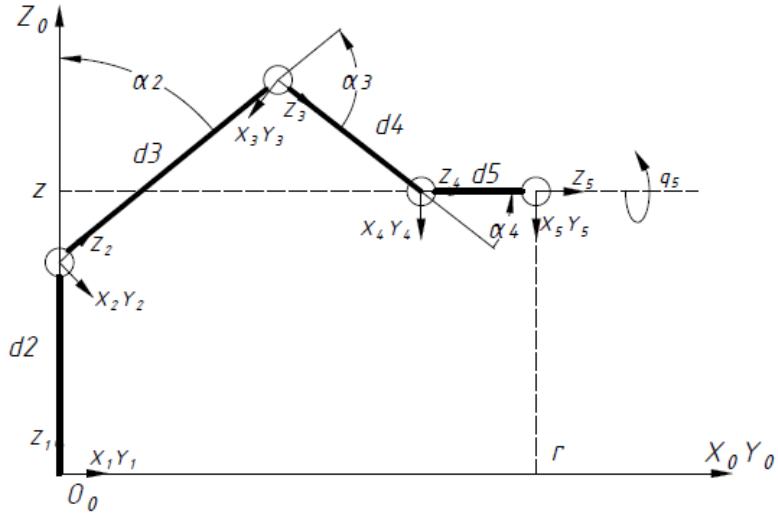


Рисунок 54 – Кинематическая схема робота – вид слева

При решении прямой задачи кинематики имеем обобщенные координаты $(q_1, q_2, \dots, q_5)^T = (q_1, \alpha_2, \alpha_3, \alpha_4, q_5)^T$, необходимо найти положение и ориентацию схвата. То есть по заданным углам поворота каждого звена нужно найти положение и ориентацию схвата.

Положение и ориентация схвата описываются матрицей T_5 :

$$T_5 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

где r_{ij} – элементы матрицы поворота системы координат $O_5X_5Y_5Z_5$ относительно $O_0X_0Y_0Z_0$, эта матрица описывает ориентацию схвата. Положение схвата определяется вектором $(x, y, z)^T$.

Опишем связи между звеньями с помощью представления Денавита-Хартенберга:

$$\begin{aligned}
A_1 &= \begin{pmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}; A_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_2 & -S_2 & 0 \\ 0 & S_2 & C_2 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \\
A_3 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_3 & -S_3 & 0 \\ 0 & S_3 & C_3 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \\
A_4 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_4 & -S_4 & 0 \\ 0 & S_4 & C_4 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}; A_5 = \begin{pmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix},
\end{aligned} \tag{4}$$

где A_i – матрица перехода из системы координат $O_0X_0Y_0Z_0$ в $O_1X_1Y_1Z_1$; $C_1 = \cos(q_1), S_1 = \sin(q_1); C_2 = \cos(\alpha_2), S_2 = \sin(\alpha_2)$ и т.д.; d_i – расстояние между центрами систем координат $O_0X_0Y_0Z_0$ и $O_1X_1Y_1Z_1$, для d_2, \dots, d_5 – это длины звеньев.

Перемножая матрицы перехода A_1, \dots, A_5 найдем положение и ориентацию схвата в абсолютной системе координат:

$$T_5 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5. \tag{5}$$

Проверим разработанную математическую модель решения прямой задачи кинематики (ПЗК) в Mathcad (рисунок 55).

$d1 := 0 \quad d2 := 190 \quad d3 := 178 \quad d4 := 178 \quad d5 := 82$ $q1 := \frac{\pi}{2} \quad al2 := \frac{-\pi}{2} \quad al3 := \frac{-\pi}{2} \quad al4 := \frac{\pi}{2} \quad q5 := 0$ $q := (q1 \ al2 \ al3 \ al4 \ q5)^T$	$al2 - \text{это } \alpha_2$ $(ortX5 \ ortY5 \ ortZ5)^T$ – это ориентация схвата $(px \ py \ pz)^T$ – это положение схвата	
$A1 := \begin{pmatrix} \cos(q1) & -\sin(q1) & 0 & 0 \\ \sin(q1) & \cos(q1) & 0 & 0 \\ 0 & 0 & 1 & d1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $A4 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(al4) & -\sin(al4) & 0 \\ 0 & \sin(al4) & \cos(al4) & d4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$A2 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(al2) & -\sin(al2) & 0 \\ 0 & \sin(al2) & \cos(al2) & d2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $A5 := \begin{pmatrix} \cos(q5) & -\sin(q5) & 0 & 0 \\ \sin(q5) & \cos(q5) & 0 & 0 \\ 0 & 0 & 1 & d5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$A3 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(al3) & -\sin(al3) & 0 \\ 0 & \sin(al3) & \cos(al3) & d3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$T5 := A1 \cdot A2 \cdot A3 \cdot A4 \cdot A5$		
$T5 = \begin{pmatrix} 0 & 0 & -1 & -260 \\ 1 & 0 & 0 & 1.59 \times 10^{-14} \\ 0 & -1 & 0 & 12 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$ortX5 := (T5_{0,0} \ T5_{1,0} \ T5_{2,0})^T$ $ortY5 := (T5_{0,1} \ T5_{1,1} \ T5_{2,1})^T$ $ortZ5 := (T5_{0,2} \ T5_{1,2} \ T5_{2,2})^T$	$px := T5_{0,3}$ $py := T5_{1,3}$ $pz := T5_{2,3}$
$ortX5 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad ortY5 = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad ortZ5 = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$ $(px \ py \ pz)^T = \begin{pmatrix} -260 \\ 1.59 \times 10^{-14} \\ 12 \end{pmatrix}$	$T5 = \begin{pmatrix} 0 & 0 & -1 & -260 \\ 1 & 0 & 0 & 1.59 \times 10^{-14} \\ 0 & -1 & 0 & 12 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	

Рисунок 55 – Проверка решения ПЗК в Mathcad

Как показано на рисунке при обобщенных координатах:

$$(q_1, q_2, \dots, q_5)^T = \left(\frac{\pi}{2}, \frac{-\pi}{2}, \frac{-\pi}{2}, \frac{\pi}{2}, 0 \right)^T. \quad (6)$$

Схват манипулятора окажется в точке $(x, y, z)^T = (-260, 0, 12)^T$.

3.3 Решение обратной задачи кинематики методом обратных преобразований

При решении обратной задачи кинематики имеем положение и ориентацию схвата, необходимо найти обобщенные координаты $(q_1, q_2, \dots, q_5)^T = (q_1, \alpha_2, \alpha_3, \alpha_4, q_5)^T$.

Положение и ориентация схвата описываются матрицей T_5 , которая задана:

$$T_5 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (7)$$

где r_{ij} – элементы матрицы поворота системы координат $O_5X_5Y_5Z_5$ относительно $O_0X_0Y_0Z_0$, эта матрица описывает ориентацию схвата. Положение схвата определяется вектором $(x, y, z)^T$. Необходимо найти обобщенные координаты $(q_1, q_2, \dots, q_5)^T = (q_1, \alpha_2, \alpha_3, \alpha_4, q_5)^T$.

Имеем:

$$T_5 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5. \quad (8)$$

Матрица T_5 известна, A_1, \dots, A_5 – неизвестны. Тогда, умножая обе стороны уравнения на A_1^{-1} :

$$A_1^{-1} \cdot T_5 = A_2 \cdot A_3 \cdot A_4 \cdot A_5. \quad (9)$$

Из полученных соотношений находим q_1, q_5 . Умножая на A_2^{-1} находим все остальные углы.

Получаем следующие уравнения:

$$\left\{ \begin{array}{l} q_1 = \text{atan}2(-x, y); \\ a_1 = d_5 C_1 r_{23} - d_5 S_1 r_{13} - C_1 y + S_1 x; \\ a_2 = d_5 r_{33} + d_1 + d_2 - z; \\ \alpha_2 = \text{atan}2(-2a_1 d_3, 2a_2 d_3) \pm \\ \pm \text{atan}2\left(\sqrt{(-2a_1 d_3)^2 + (2a_2 d_3)^2 - (d_4^2 - d_3^2 - a_1^2 - a_2^2)^2}, d_4^2 - d_3^2 - a_1^2 - a_2^2\right); \\ \alpha_3 = \text{atan}2\left(\frac{C_2 a_1 + S_2 a_2}{d_4}, \frac{-C_2 a_2 + S_2 a_1 - d_3}{d_4}\right); \\ \alpha_4 = \text{atan}2(-S_3, C_3) \pm \\ \pm \text{atan}2\left(\sqrt{(-S_3)^2 + (C_3)^2 - (C_2 r_{33} - C_1 S_2 r_{23} + S_1 S_2 r_{13})^2}, C_2 r_{33} - C_1 S_2 r_{23} + S_1 S_2 r_{13}\right); \\ q_5 = \text{atan}2(-(C_1 \cdot r_{12} + S_1 \cdot r_{22}), C_1 \cdot r_{11} + S_1 \cdot r_{21}). \end{array} \right.$$

(10)

Проверим разработанную математическую модель решения обратной задачи кинематики (ОЗК) в Mathcad (рисунки 56, 57).

$d1 := 0 \quad d2 := 190 \quad d3 := 178 \quad d4 := 178 \quad d5 := 82$

$(px \quad py \quad pz) := (-260 \quad 0 \quad 12)$

$ortX5 := (0 \quad 1 \quad 0)$

$ortY5 := (0 \quad 0 \quad -1)$

$ortZ5 := (-1 \quad 0 \quad 0)$

$$T5 := \begin{pmatrix} ortX5_{0,0} & ortY5_{0,0} & ortZ5_{0,0} & px \\ ortX5_{0,1} & ortY5_{0,1} & ortZ5_{0,1} & py \\ ortX5_{0,2} & ortY5_{0,2} & ortZ5_{0,2} & pz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$al2 = \text{atan2}(a, b)$

$(ortX5 \quad ortY5 \quad ortZ5)^T$ – это ориентация схвата

$(px \quad py \quad pz)^T$ – это положение схвата

$$\text{atan2}(a, b) := \begin{cases} \text{atan}\left(\frac{a}{b}\right) & \text{if } b > 0 \\ \text{atan}\left(\frac{a}{b}\right) + \pi & \text{if } b < 0 \wedge a \geq 0 \\ \text{atan}\left(\frac{a}{b}\right) - \pi & \text{if } b < 0 \wedge a < 0 \\ \frac{\pi}{2} & \text{if } b = 0 \wedge a > 0 \\ -\frac{\pi}{2} & \text{if } b = 0 \wedge a < 0 \end{cases}$$

$q1 := \text{atan2}(-px, py) = 1.57 \quad \text{решение!!}$

$c1 := \cos(q1)$

$s1 := \sin(q1)$

$q5 := \text{atan2}[-(c1 \cdot ortY5_{0,0} + s1 \cdot ortY5_{0,1}), c1 \cdot ortX5_{0,0} + s1 \cdot ortX5_{0,1}]$

$c5 := \cos(q5)$

$s5 := \sin(q5)$

$A11 := d5 \cdot c1 \cdot ortZ5_{0,1} - d5 \cdot s1 \cdot ortZ5_{0,0} - c1 \cdot py + px \cdot s1 = -178$

$A22 := d5 \cdot ortZ5_{0,2} + d1 + d2 - pz = 178$

+

$aa := -2 \cdot A11 \cdot d3$

$cc := d4^2 - d3^2 - A11^2 - A22^2 = -6.34 \times 10^4$

$bb := 2 \cdot A22 \cdot d3$

$al22 := \text{atan2}(aa, bb) + \text{atan2}\left(\sqrt{aa^2 + bb^2 - cc^2}, cc\right) = 3.14$

$al2 := \text{atan2}(aa, bb) - \text{atan2}\left(\sqrt{aa^2 + bb^2 - cc^2}, cc\right) = -1.57$

$c2 := \cos(al2)$

$s2 := \sin(al2)$

Рисунок 56 – Проверка решения ОЗК методом обратных преобразований в Mathcad – часть 1

$$\begin{aligned}
s3 &:= \frac{(c_2 \cdot A_{11} + s_2 \cdot A_{22})}{d_4} = -1 \\
c3 &:= \frac{[c_2 \cdot (-A_{22}) + s_2 \cdot A_{11} - d_3]}{d_4} = 0 \\
a13 &:= \text{atan22}(s3, c3) = -1.57 \\
aa4 &:= -s3 \quad bb4 := c3 \\
cc4 &:= c_2 \cdot \text{ortZ}_{5,2} - (c_1 \cdot s_2 \cdot \text{ortZ}_{5,1} - s_1 \cdot s_2 \cdot \text{ortZ}_{5,0}) \\
a144 &:= \text{atan22}(aa4, bb4) + \text{atan22}\left(\sqrt{aa4^2 + bb4^2 - cc4^2}, cc4\right) = 1.57 \\
a14 &:= \text{atan22}(aa4, bb4) - \text{atan22}\left(\sqrt{aa4^2 + bb4^2 - cc4^2}, cc4\right) = 1.57 \\
\begin{pmatrix} q1 \\ a12 \\ a13 \\ a14 \\ q5 \end{pmatrix} &= \begin{pmatrix} 1.57 \\ -1.57 \\ -1.57 \\ 1.57 \\ 0 \end{pmatrix} \\
q1_deg &:= q1 \cdot \frac{180}{\pi} = 90 \\
a12_deg &:= a12 \cdot \frac{180}{\pi} = -90 \\
a13_deg &:= a13 \cdot \frac{180}{\pi} = -90 \\
a14_deg &:= a14 \cdot \frac{180}{\pi} = 90 \\
q5_deg &:= q5 \cdot \frac{180}{\pi} = 0
\end{aligned}$$

Рисунок 57 – Проверка решения ОЗК методом обратных преобразований в Mathcad – часть 2

Как показано на рисунках для достижения положений схвата манипулятора:

$$(x, y, z)^T = (-260, 0, 12)^T. \quad (11)$$

Необходимо повернуть каждое звено на следующие углы в радианах:

$$(q_1, q_2, \dots, q_5)^T = \left(\frac{\pi}{2}, \frac{-\pi}{2}, \frac{-\pi}{2}, \frac{\pi}{2}, 0 \right)^T. \quad (12)$$

3.4 Расчет операционного пространства робота-манипулятора «Робко 01»

Манипулирование объектом манипулятором может происходить лишь в определенной области пространства. Такое пространство называется операционным [4]. Рассчитаем его для робота-манипулятора «Робко 01», учитывая, что принадлежащей роботу точкой, по которой будет строится операционное пространство будет конец схвата.

Робот «Робко 01» имеет ограничения на диапазон углов поворота звеньев (по рисункам 53, 54) – $(q_1, q_2, \dots, q_5)^T = (q_1, \alpha_2, \alpha_3, \alpha_4, q_5)^T$:

$$\begin{cases} -\frac{\pi}{3} \leq q_1 \leq \frac{\pi}{3}; \\ -\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{2}; \\ 0 \leq q_2 + q_3 - \frac{\pi}{2} \leq \frac{3\pi}{4}. \end{cases} \quad (13)$$

Углы q_4, q_5 не рассматриваем, так как они не влияют на операционное пространство.

С такими ограничениями операционное пространство (ОП) будет иметь вид как показано на рисунках (в Wolfram) 58-61.

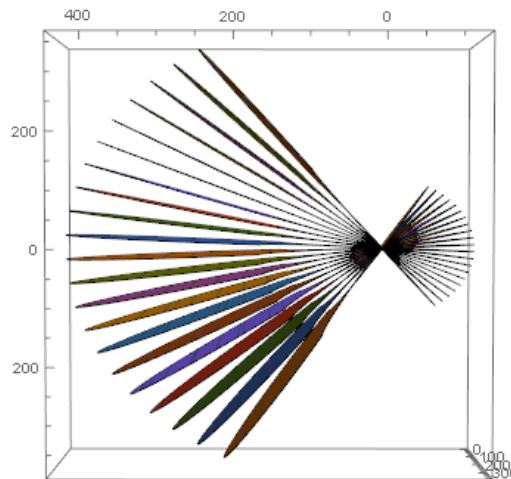


Рисунок 58 – Операционное пространство «Робко 01» в Wolfram без ограничения на пересечение звеньев, вид сверху

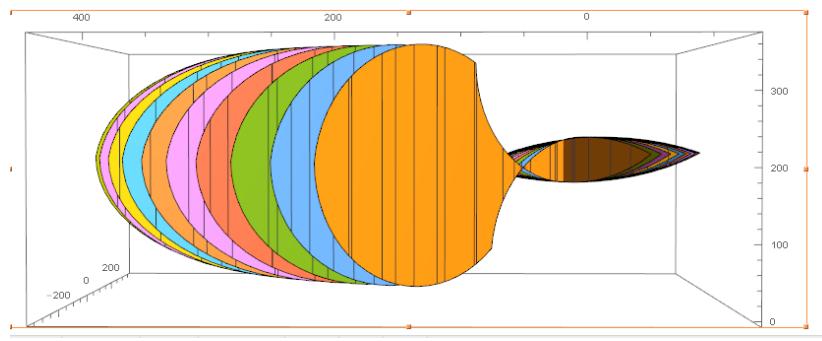


Рисунок 59 – Операционное пространство «Робко 01» в Wolfram без ограничения на пересечение звеньев, вид слева

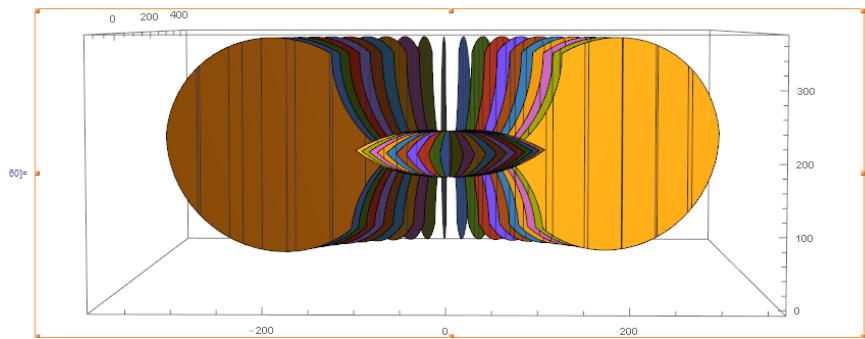


Рисунок 60 – Операционное пространство «Робко 01» в Wolfram без ограничения на пересечение звеньев, вид сзади

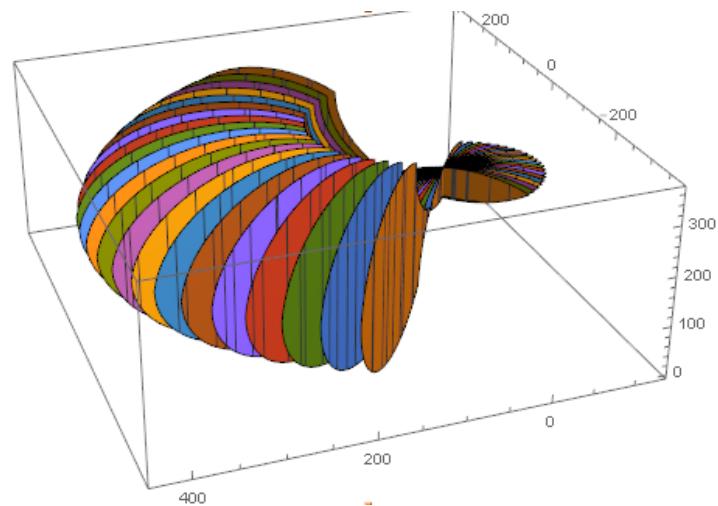


Рисунок 61 – Операционное пространство «Робко 01» в Wolfram без ограничения на пересечение звеньев, общий вид

Мы не учли самопересечение звеньев (рисунки 62 и 63), поэтому замечен как бы «хвост рыбы». Точка-конец схватка робота в этой части пространства не может находиться.

Введем дополнительное ограничение на самопересечение звеньев.



Рисунок 62 – Ограничение на пересечение звеньев d3 и d4

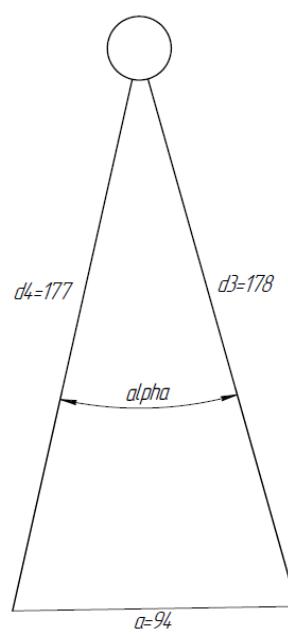


Рисунок 63 – Ограничение на пересечение звеньев d3 и d4, схематический рисунок

Найдем угол между звеньями d3 и d4 (из рисунка 63):

$$\begin{aligned} \alpha = \arccos \left(\frac{d4^2 + d3^2 - a^2}{2d4 \cdot d3} \right) &= \arccos \left(\frac{177^2 + 178^2 - 94^2}{2 \cdot 177 \cdot 178} \right) = \\ &= 0.536(\text{rad}) = 30.7^\circ. \end{aligned} \quad (14)$$

Возьмем с запасом в 4 градуса:

$$\alpha_{\min} = 35^\circ = 0.6 \text{ rad}, \quad (15)$$

где α_{\min} - минимальный угол между звеньями d3, d4.

Таким образом, ограничения на допустимые значения углов поворота звеньев примут вид

$$\begin{cases} -\frac{\pi}{3} \leq q_1 \leq \frac{\pi}{3}; \\ -\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{2}; \\ 0 \leq q_2 + q_3 - \frac{\pi}{2} \leq \frac{3\pi}{4}; \\ \pi - q_3 > 0.6 \end{cases} \quad (16)$$

ОП В Matlab, со всеми ограничениями, можно видеть на рисунках 64 и 65.

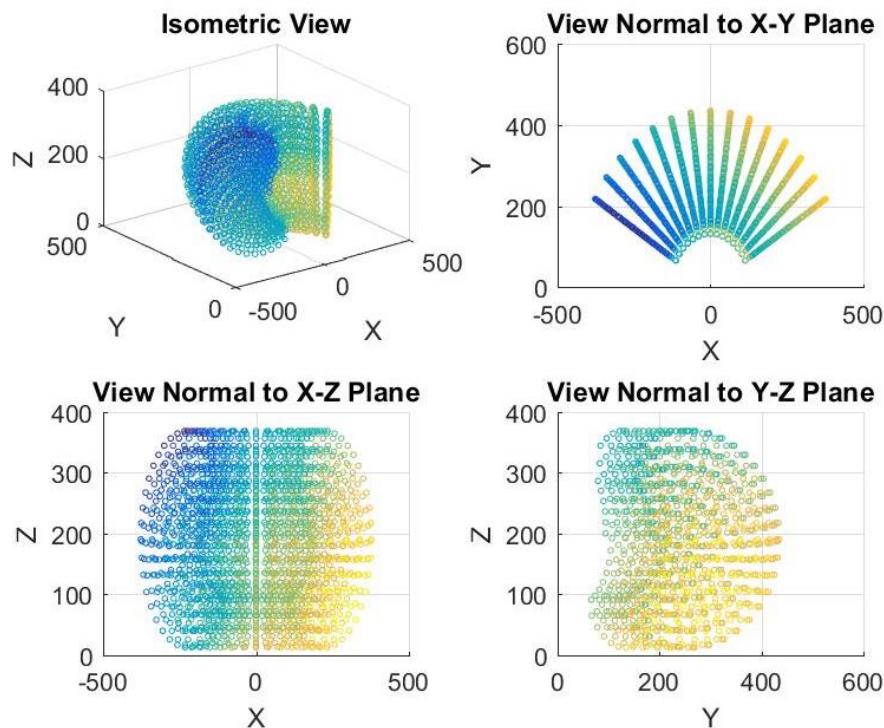


Рисунок 64 – Операционное пространство «Робко 01» в Matlab с ограничениями на пересечение звеньев, полученные точки достижимости

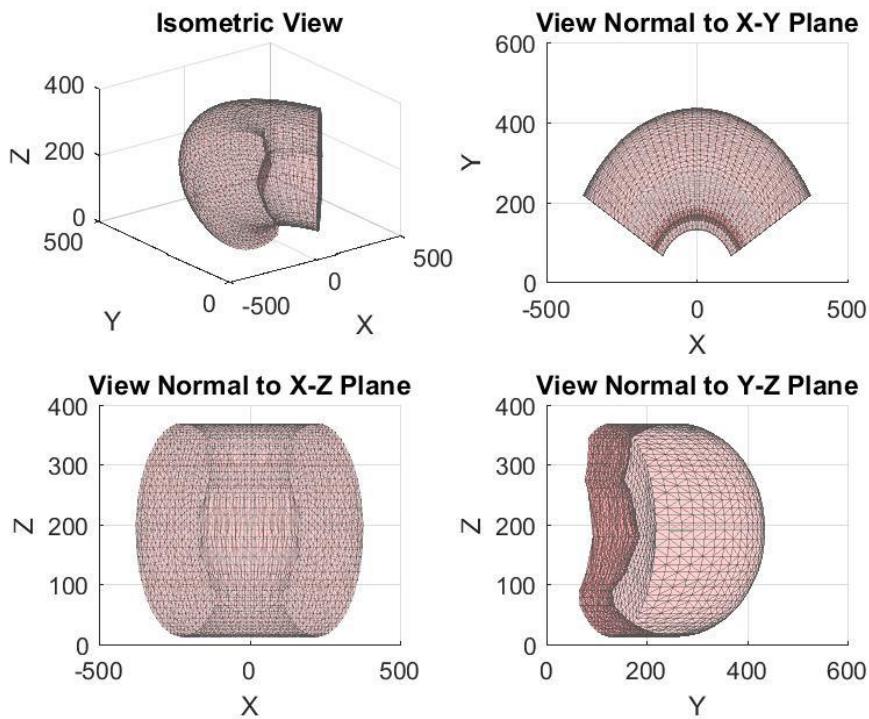


Рисунок 65 – Операционное пространство «Робко 01» в Matlab с ограничениями на пересечение звеньев, граничная поверхность по ранее найденным точкам достижимости

Как видно из рисунка 65 параллельная плоскости XY плоскость, сечение ОП по которой имеет наибольшую площадь, проходит через координату Z = 190. Стол для рисования (или других операций) нужно располагать, имея это в виду.

3.5 Разработка алгоритма работы системы управления

Разработаем алгоритм работы системы управления роботом манипулятором. Схема верхнего уровня показана на рисунке 66. Подпрограммы обработки G01, G02/G03 представлены на рисунках 67, 68.

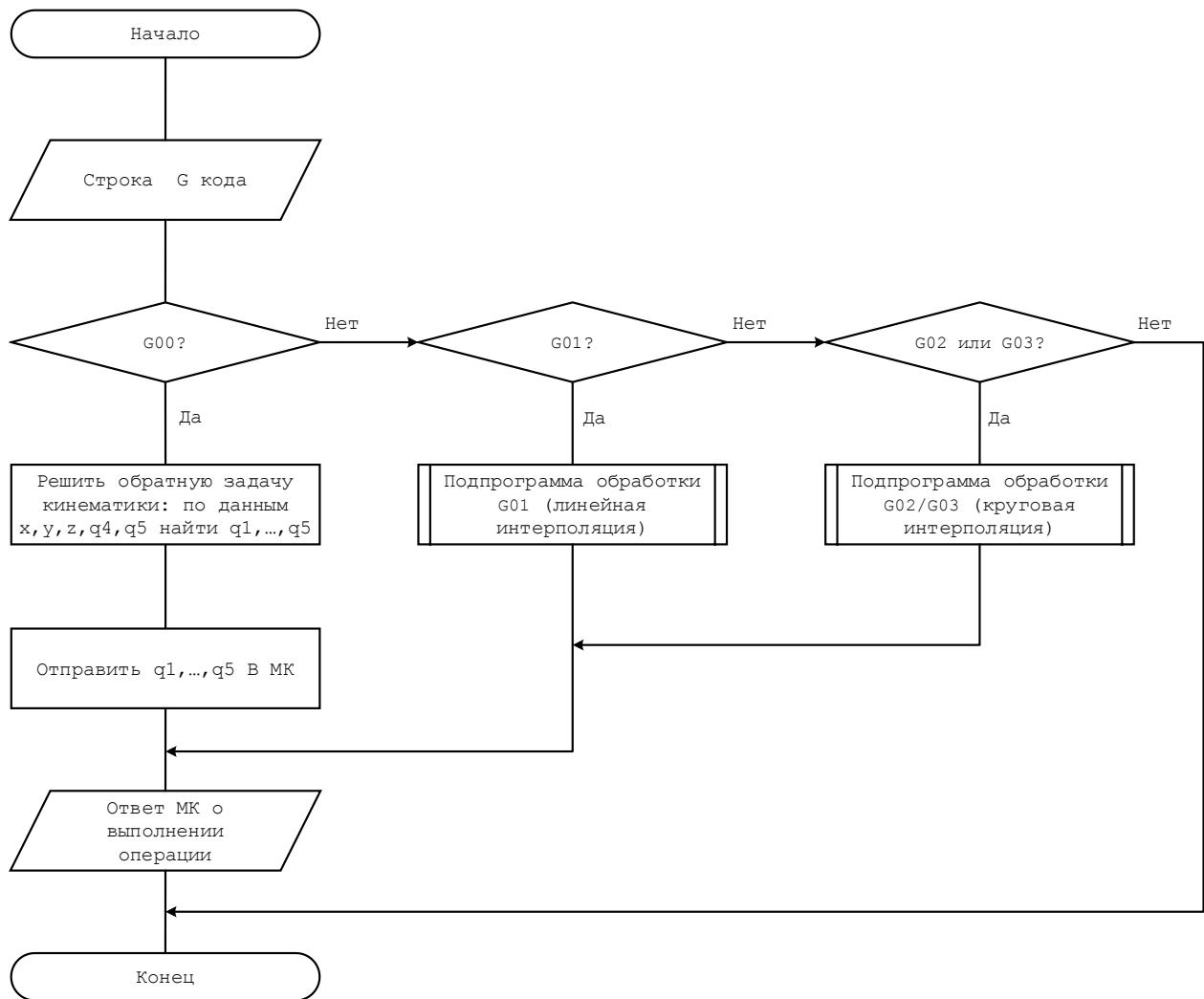


Рисунок 66 – Блок-схема алгоритма работы системы управления роботом-манипулятором

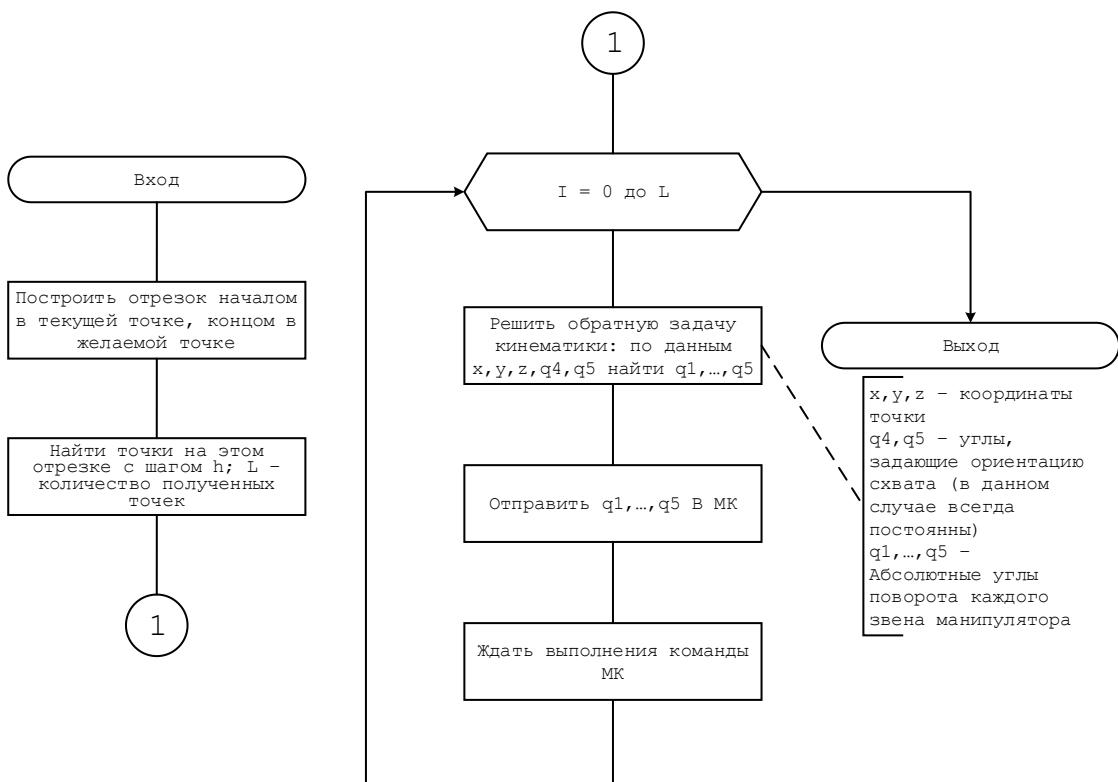


Рисунок 67 – Блок-схема алгоритма подпрограммы обработки G01 (линейная интерполяция)

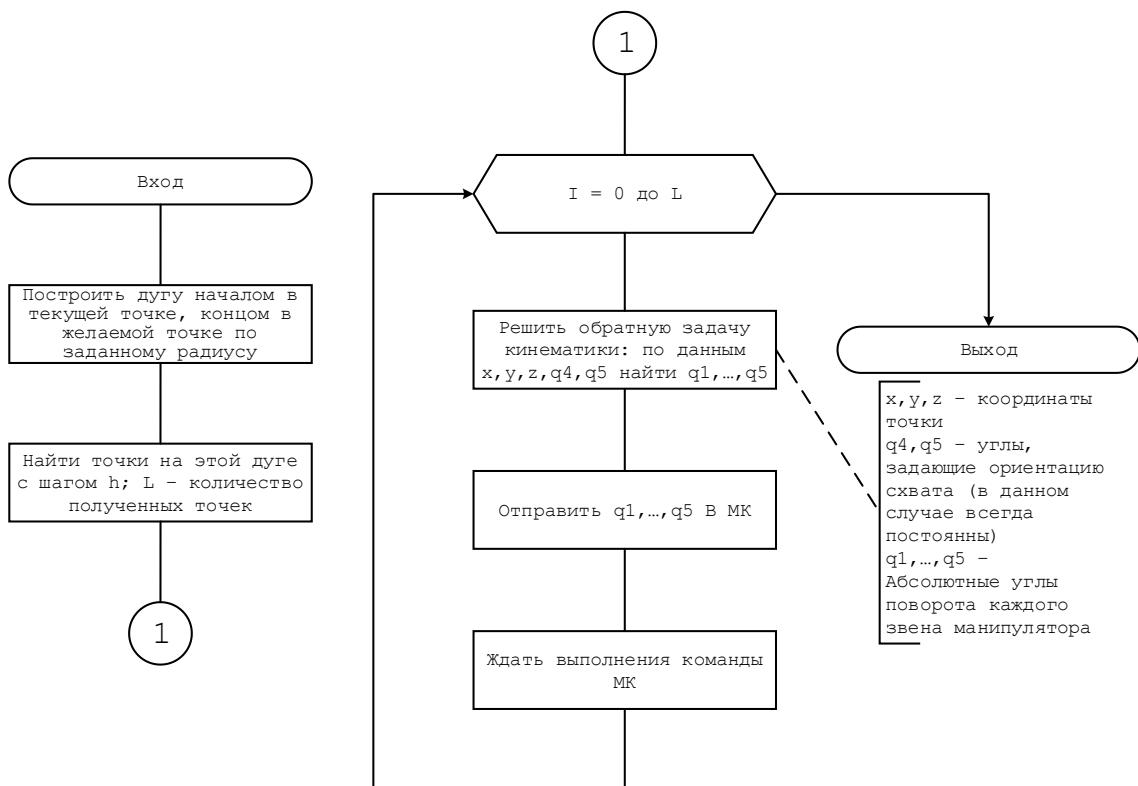


Рисунок 68 – Блок-схема алгоритма подпрограммы обработки G02/G03 (круговая интерполяция)

Согласно разработанному алгоритму работы системы управления было написано десктопное программное обеспечение на языке программирования C#. Прошивка написана на Arduino – C++ подобном языке программирования МК. Исходники находятся по ссылке [14].

3.6 Разработка линейного и кругового интерполяторов

После определения команд G01, G02/G03 необходимо производить интерполяцию.

В случае команды G01 – траекторией движения является отрезок с началом в текущей точке и концом в указанных координатах X Y Z команды G01.

Пусть P1 – точка начала отрезка, P2 – точка конца отрезка, тогда можем найти точку P этого отрезка изменяя параметр t от 0 до 1 (0 – начало, 1 – конец, 0.5 – середина):

$$\begin{cases} \bar{V}_{cut} = P_1 - P_0; \\ P = P_0 + t \cdot \bar{V}_{cut}; \\ t \in [0;1]. \end{cases} \quad (17)$$

В случае команд G02/G03 – траекторией движения является дуга с началом в текущей точке и концом в указанных координатах X Y Z команд G02/G03. Направление дуги: G02 – по часовой стрелке, G03 – против часовой стрелки. Могут указываться либо радиус R дуги, либо смещение I J K ее центра относительно точки начала дуги. В любом случае необходимо найти радиус R дуги и ее центр СР для удобной работы с дугой.

Пусть P1 – точка начала дуги, P2 – точка конца дуги, СР – центр дуги, R – радиус дуги. Тогда можем найти точку P этой дуги изменяя параметр t от 0 до 1 (0 – начало, 1 – конец, 0.5 – середина):

$$\begin{cases} \bar{V}_1 = P_1 - CP \\ \bar{V}_2 = P_2 - CP \\ \theta = \angle(\bar{V}_1, \bar{V}_2) \\ \theta_0 = \angle(\bar{V}_1, \overline{Xaxes}) \\ P = (R \cdot \cos(t \cdot \theta + \theta_0) + CP_x; R \cdot \sin(t \cdot \theta + \theta_0) + CP_y) \\ t \in [0;1] \end{cases}, \quad (18)$$

где θ, θ_0 – центральный угол дуги, угол между вектором \bar{V}_1 и вектором $\overline{Xaxes} = (1;0)$ (сонарвленным с осью OX) соответственно (показано на рисунке 69). Эти углы знаковые: если угол положителен, то направление угла против часовой стрелки, иначе – по часовой стрелке. Поясняющее изображение дуги можно видеть на рисунке 69.

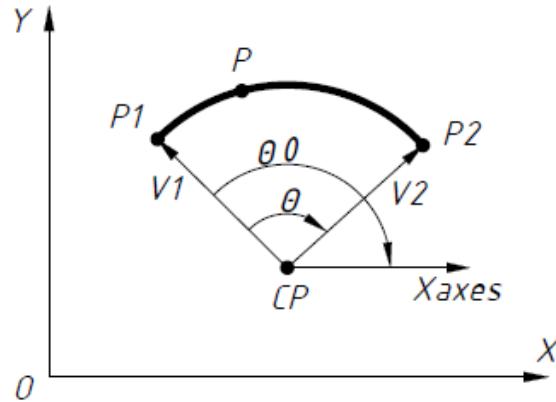


Рисунок 69 – Дуга при круговой интерполяции

Если нужно получить точку на расстоянии h от точки начала отрезка/дуги, то параметр t найдем таким образом:

$$t = \frac{h}{L}, \quad (19)$$

где h – расстояние от точки начала до искомой точки (при траектории отрезка/дуги), L – длина отрезка/дуги.

3.7 Структура кода программы

На рисунке 70 показана диаграмма существующих классов программы.

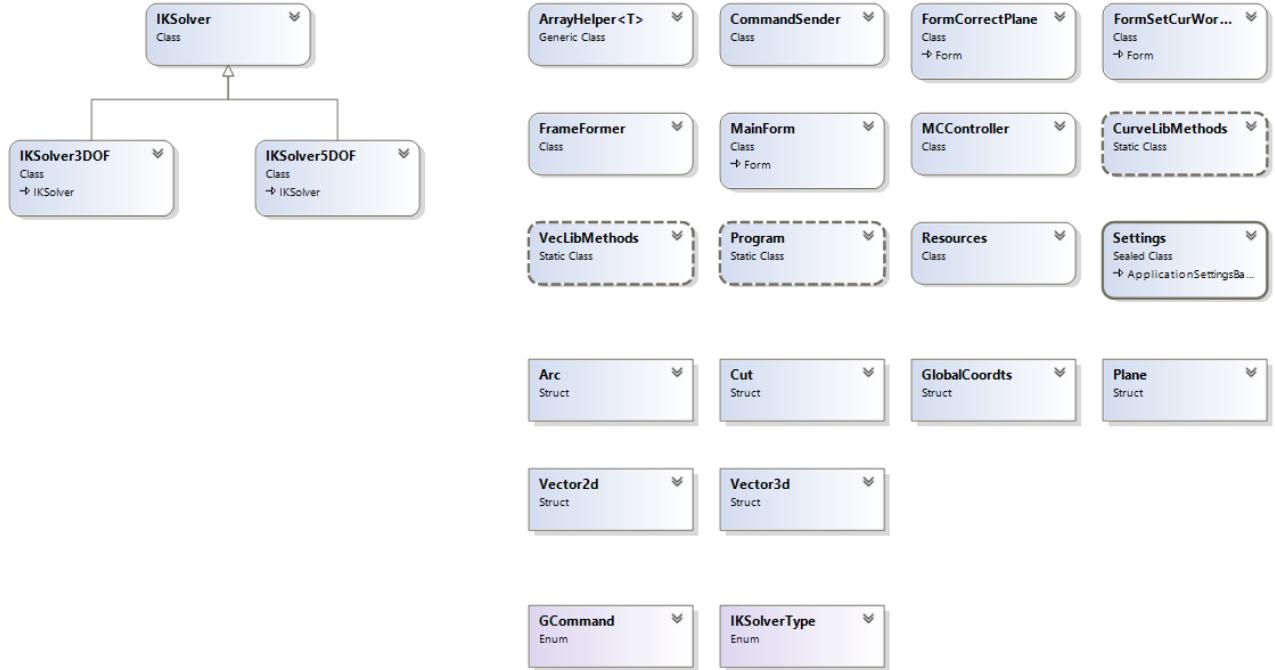


Рисунок 70 – Диаграмма классов программы

На рисунке 71 изображена карта кода с пространствами имен.

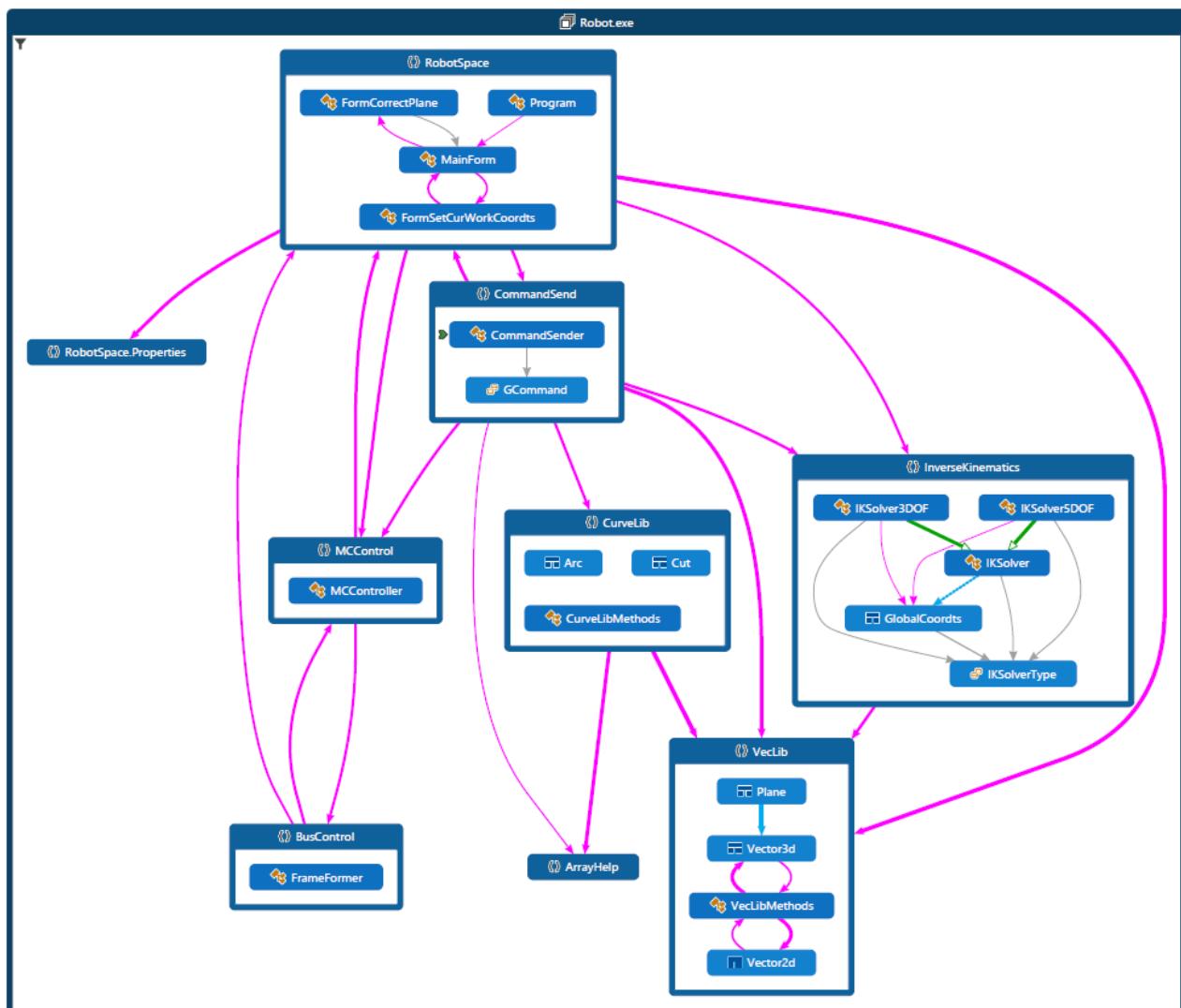


Рисунок 71 – Карта кода с пространствами имен

Для понимания взаимодействия внутренних элементов программы покажем лишь ключевые элементы (рисунки 72 и 73).

На рисунке 72 показана диаграмма ключевых классов программы.

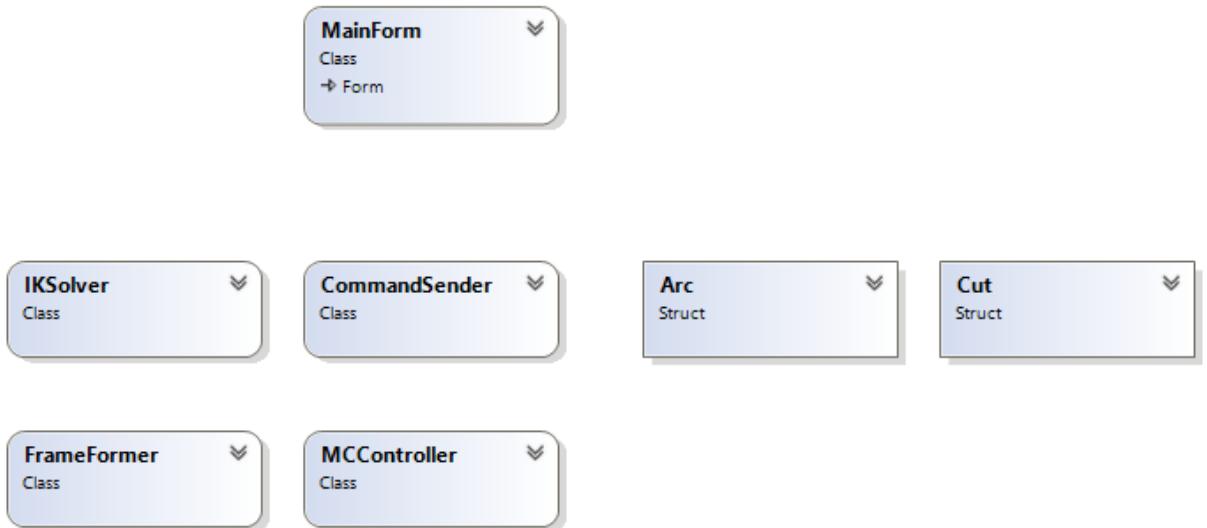


Рисунок 72 – Диаграмма ключевых классов программы

На рисунке 73 показана карта кода ключевых классов программы.

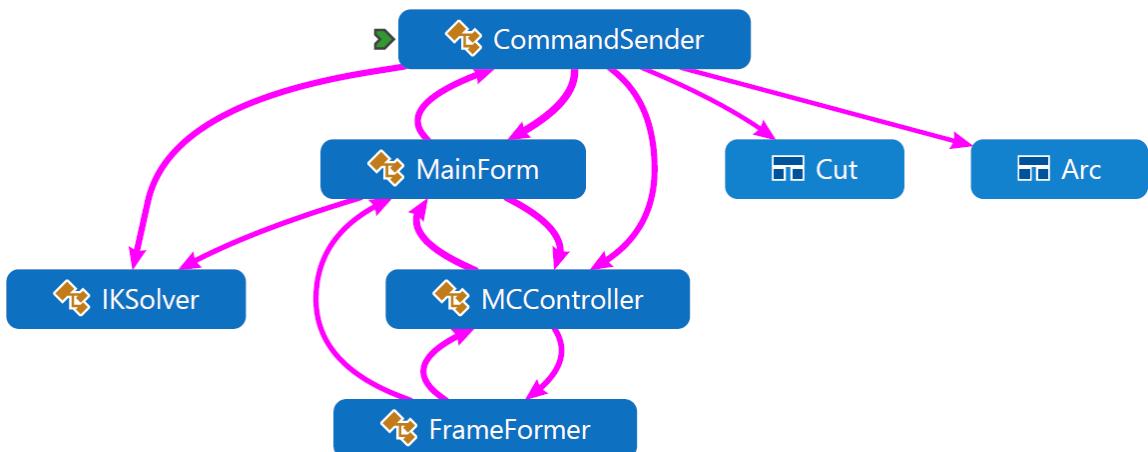


Рисунок 73 – Карта кода вызова ключевых классов программы

Внутреннее взаимодействие ключевых элементов программы происходит следующим образом:

- Класс MainForm получает управляющую программу (УП) в виде G кода. Затем по команде «Старт» (кнопка «play» на главной форме – рисунок 74) передает УП классу CommandSender. В любой момент времени можно приостановить выполнение УП по команде «Пауза» (кнопка «pause» на главной форме, появляется при нажатии «play» – рисунок 74) или полностью остановить по команде «Стоп» (кнопка «stop» на главной форме – рисунок 74).

2) Класс CommandSender распознает тип команды (G00, G01, G02/G03, и пр.), и после соответствующей интерполяции с помощью структур Cut (отрезок), Arc (дуга) получает точку (на определенном расстоянии – шаге от текущей точки) (x,y,z). Затем полученную точку отправляет в класс IKSolver для решения ОЗК. Полученные обобщенные координаты (q1, q2, q3) отправляет классу MCController.

3) Класс MCController формирует пакет данных, задание микроконтроллеру, с помощью класса FrameFormer. А затем полученный пакет данных отправляет микроконтроллеру (МК). Если МК отправляет ответ о выполнении задания, то класс MCController посылает классу CommandSender команду продолжать выполнение УП, иначе – класс MCController отправляет повторно пакет данных.

3.8 Интерфейс программного обеспечения

Интерфейс программы представлен на рисунке 74.

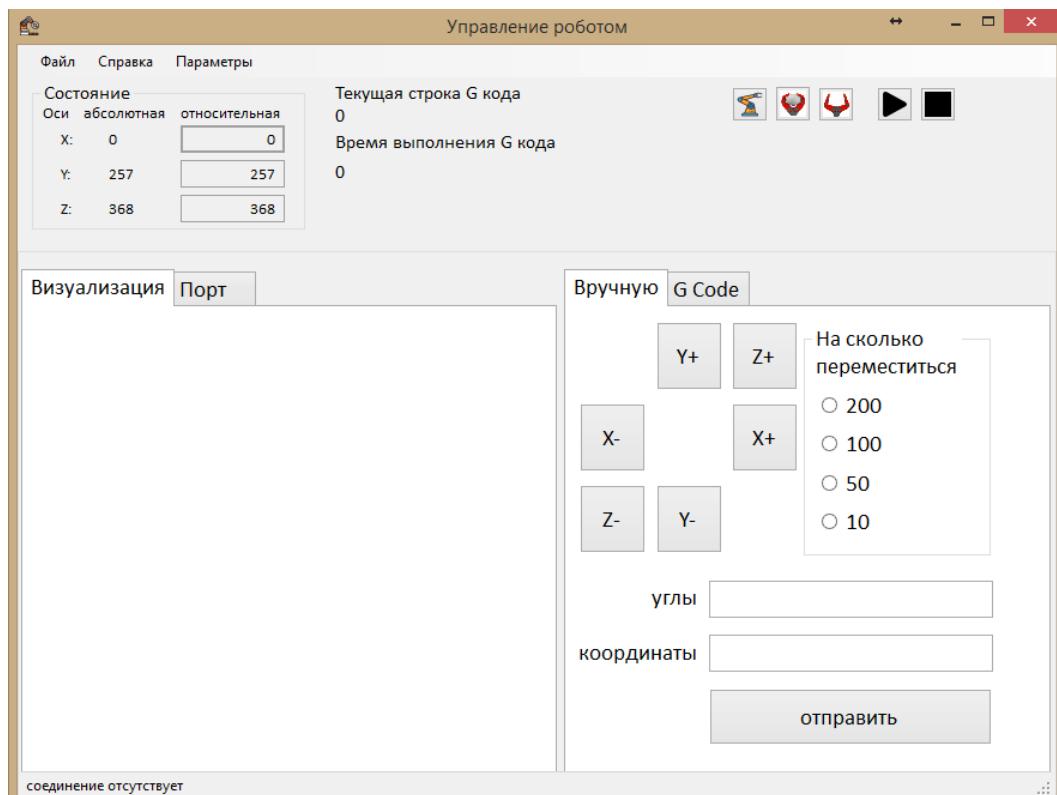


Рисунок 74 – Интерфейс программы

3.9 Логгирование сессии программного обеспечения

При отладки программного обеспечения очень помогает логгирование.

Пример логгирования показан на рисунке 75.

```
Log file Date: 10.06.2020 14:44:44
*****
***** соединено с COM7 *****
***** выполнение G команд: старт *****
***** выполнение G команд *****
-----
CommandSender.DoCommand(): Command: 1 : G02 X4 R4
-----
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (0, 0, 0)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (0, 0.00234603512708436, 0.519958195011937)
FrameFormer.DoFrame_Request_MoveToAbsoluteAngles(): floatArray = 0 0 0.52
MCControl.Send(): curByteArr = 1 0 19 1 0 0 0 0 128 0 0 0 0 184 30 5 63 179 188
-----
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (0.919207611423778, 0.387117388346168, 0)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (-0.204619769860721, 0.127521811489926, 0.524318311359249)
FrameFormer.DoFrame_Request_MoveToAbsoluteAngles(): floatArray = -0.2 0.13 0.52
MCControl.Send(): curByteArr = 1 0 19 1 0 205 204 76 190 184 30 5 62 184 30 5 63 97 128
-----
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (1.90561365883636, 0.53478463213182, 0)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (-0.423949050575247, 0.176833062572367, 0.528476837142383)
FrameFormer.DoFrame_Request_MoveToAbsoluteAngles(): floatArray = -0.42 0.18 0.53
MCControl.Send(): curByteArr = 1 0 19 1 0 61 10 215 190 236 81 56 62 20 174 7 63 88 44
-----
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (2.89788819178969, 0.433820497353033, 0)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (-0.644941484074294, 0.147350437633894, 0.532371668789409)
FrameFormer.DoFrame_Request_MoveToAbsoluteAngles(): floatArray = -0.64 0.15 0.53
MCControl.Send(): curByteArr = 1 0 19 1 0 10 215 35 191 154 153 25 62 20 174 7 63 76 184
-----
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (3.83433638582831, 0.0905024448985139, 0)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (-0.854465657108736, 0.0408244959688576, 0.535642304808775)
FrameFormer.DoFrame_Request_MoveToAbsoluteAngles(): floatArray = -0.85 0.04 0.54
MCControl.Send(): curByteArr = 1 0 19 1 0 154 153 89 191 10 215 35 61 113 61 10 63 220 144
-----
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (4, 0, 0)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (-0.891691107521886, 0.012513559500432, 0.536142804943994)
FrameFormer.DoFrame_Request_MoveToAbsoluteAngles(): floatArray = -0.89 0.01 0.54
MCControl.Send(): curByteArr = 1 0 19 1 0 10 215 99 191 10 215 35 60 113 61 10 63 147 58
*****
G код выполнен за такое время: 00:00:00.4005202
```

Рисунок 75 – Пример логгирования сессии программы при выполнении команды G02 X4 R4

Благодаря логированию была выявлена неустойчивая ошибка (рисунок 76). Это было в старой версии протокола без контрольной суммы в пакете данных. Она появлялась случайным образом. Персональный компьютер (ПК) отправлял задание микроконтроллеру (МК), но МК неправильно выполнял задание (робот ехал в недоступную зону). На рисунке 76 изображены логи: слева

– правильное выполнение УП, справа – с ошибкой. Как видно, отличие логов состоит в том, что ПК не получил ответ от МК. Ошибка заключается в том, что при передаче терялись данные. Для выявления таких случаев в протокол обмена данными были внесены изменения: добавлена контрольная сумма в пакет данных. Теперь МК сам считает контрольную сумму полученного пакета данных и если она отличается от той, что была в пакете данных, то МК отправляет ответ ПК: ошибка. В таком случае ПК отправляет пакет данных повторно. Таким образом, ошибка была решена.

```
-----|-----|
CommandSender.DoCommand(): Command: 242 : G2 X59.3749 Y17.1673 I0.4002 J12.3845
----|----|
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (61.4283, 16.9284, 0)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (-2.45157018651404, 32.6354690257)
MCControl.TaskAngles(): floatArray = -2.45 32.64 58.99
MCControl.Send(): curByteArr = 205 204 28 192 92 143 2 66 195 245 107 66
----|----|
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (60.4312088800473, 17.00097)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (-2.23729690613133, 32.6376425434)
MCControl.TaskAngles(): floatArray = -2.24 32.64 58.98
MCControl.Send(): curByteArr = 41 92 15 192 92 143 2 66 133 235 107 66
----|----|
CommandSender.TaskGoToRelativeCoorts(): owner.CurWorkCoorts: (59.4432135609947, 17.15368)
CommandSender.TaskGoToRelativeCoorts(): iKSolver.QDeg: (-2.02441939892568, 32.6555870230)
MCControl.TaskAngles(): floatArray = -2.02 32.66 58.96
MCControl.Send(): curByteArr = 174 71 1 192 215 163 2 66 10 215 107 66
-----|-----|
```

Рисунок 76 – Логи выполнения УП без ошибки – слева, с ошибкой – справа

3.10 Результат работы системы управления

Результат рисования прямоугольника 50x50 показан на рисунке 77.

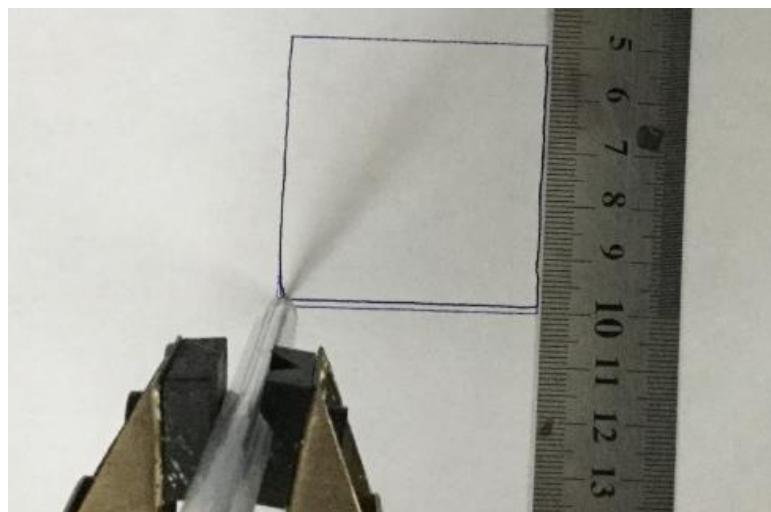


Рисунок 77 – Результат рисования прямоугольника 50x50

По более сложному заданию в виде различных G кодов (рисунок 78) результат рисования показан на рисунке 79. Видео находится по ссылке [15].

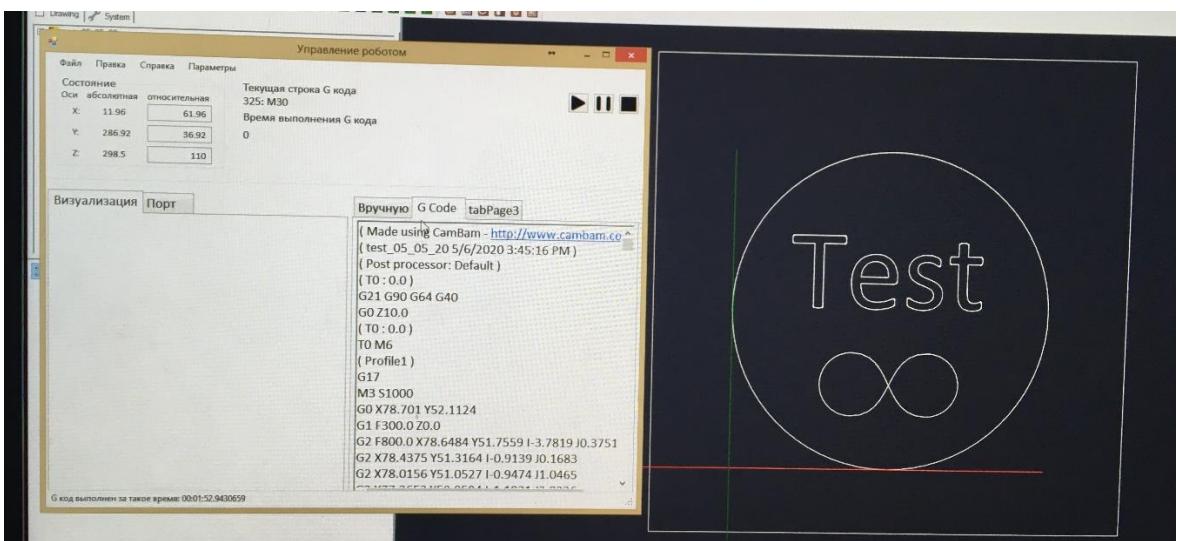


Рисунок 78 – Задание для тестирования обработки различных G кодов (создано в CamBam)

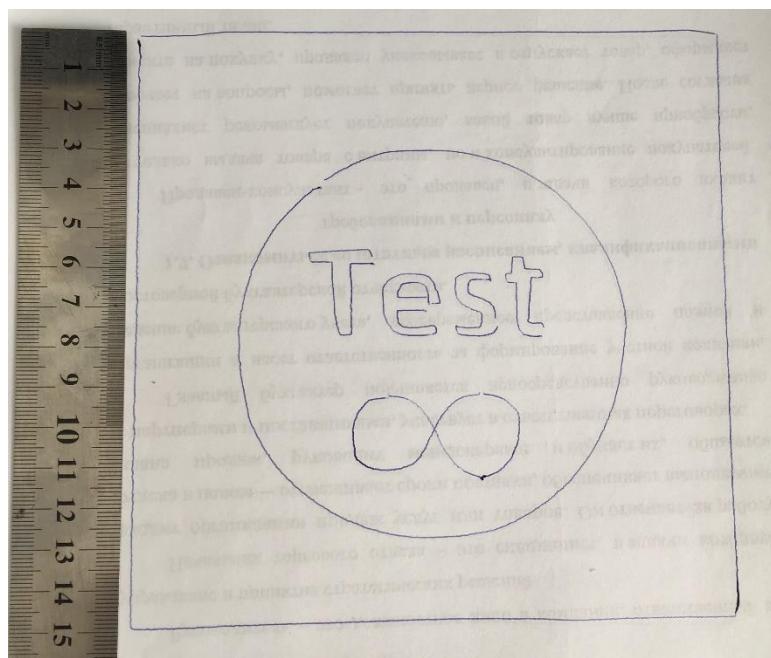


Рисунок 79 – Результат рисования задания для тестирования обработки различных G кодов

3.11 Выводы по разделу

Таким образом, в данном разделе описание разработки программной части системы управления. Сделаем следующие выводы:

Разрабатываемая система управления роботом «Робко 01» имеет: язык верхнего уровня – структурный, внедрены G-коды. Блок управления содержит: планировщик траектории, блок расчета прямой и обратной задач кинематики, систему обработки управляющих программ. Протокол: на основе Modbus RTU, 7 секторов, 19 байтов в одном пакете.

Также приводится решение прямой задачи кинематики методом Денавита-Хартенберга; решение обратной задачи кинематики методом обратных преобразований; алгоритм работы системы управления.

Исходники десктопного приложения, прошивки МК, схемы платы управления находятся по ссылке [14]. Видео работы системы управления находится по ссылке [15].

ЗАКЛЮЧЕНИЕ

В данной работе показана разработка системы управления для роботоманипулятора «Робко 01».

Стоит отметить, что результат работы системы управления, показанный на рисунке 79, является результатом различных корректировок. Дело в том, что робот имеет существенный недостаток: схват сжимается/разжимается в зависимости от положения звеньев d3, d4 (рисунок 54). Необходимо было ввести корректировку зжатия/разжатия схвата по положению звеньев d3, d4. Кроме того, ось вращения звена d2 (рисунок 54) не перпендикулярна плоскости OXY (такова конструкция робота), из-за чего ошибка по Z достигала 2.5 мм. В связи с чем была внесена корректировка по Z в зависимости от координат (x,y) и ошибка по Z уменьшилась до 0.5 мм.

Однако, как видно из рисунка 79, результат рисования не идеален. Причиной этому является в первую очередь зазор между зубьями шестерен, а также недостаточная жесткость робота.

Разработана современная, гибкая, многофункциональная система управления.

Реализован функционал: управление движением роботом «Робко 01», решение прямой задачи кинематики (ПЗК), решение обратной задачи кинематики (ОЗК).

Выполнены следующие задачи:

- 1) Разработка архитектуры СУ
- 2) Разработка схемы платы системы управления
- 3) Выбор драйвера ШД
- 4) Выбор датчиков
- 5) Выбор МК
- 6) Разработка протокола обмена данными
- 7) Решение прямой задачи кинематики

- 8) Решение обратной задачи кинематики
- 9) Планирование пути
- 10) Разработка прошивки для МК

При разработке десктопного приложения:

- 1) Архитектура приложения (классы и пр.)
- 2) Решатели ПЗК и ОЗК
- 3) Интерпретаторы G-кода
- 4) Интерполяторы (линейные и круговые)

Исходники десктопного приложения, прошивки МК, схемы платы управления находятся по ссылке [14]. Видео работы системы управления находится по ссылке [15].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Емельянов А. В Ш.А.Н. Шаговые двигатели: учеб. пособие. – Волгоград, 2005. – 48 с.
2. Ревич Ю. В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. — 2-е изд., испр. — СПб.: БХВ-Петербург, 2011. — 352 с.: ил. — (Электроника) ISBN 978-5-9775-0657-1.
3. Евстифеев А.В. Микроконтроллеры AVR семейств Tiny и Mega фирмы.
4. Зенкевич С. Л. Ю.А.С. Управление роботами. Основы управления манипуляционными роботами. Москва: МГТУ им. Н. Э. Баумана, 2000. 400 с.
5. Данилов А.В. К.А.Н..Т.О.В. Общий подход к решению обратной задачи кинематики для манипулятора последовательной структуры с помощью конечного поворота и смещения // Препринты ИПМ им. М.В.Келдыша. 2018. № 81. 15 с. doi:10.20948/prepr-2018
6. Y. C. Solution to the Problems of Direct and Inverse Kinematics of the Robots-Manipulators Using Dual Matrices and Biquaternions on the Example of Stanford Robot Arm 2015.
7. Sumner R.W. Z.M..G.C..P.J. Mesh-based inverse kinematics. ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2005, 2005, vol. 24, iss. 3, pp. 488-495. DOI: 10.1145/1073204.1073218
8. Aristidou A. L.J. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. Graphical Models, 2011, vol. 73, iss. 5, pp. 243-260. DOI: 0.1016/j.gmod.2011.05.003
9. Варков А.А. Разработка и исследование системы управления манипуляционным промышленным роботом на базе контроллера движения: Диссертация – Иваного, 2015.

10. В.В. Рябченко Н.Н.Д. Программно-аппаратный комплекс управления роботами-манипуляторами фирмы NEURONICS AG [Электронный ресурс] – Режим доступа: <http://ea.donntu.edu.ua>.
11. А.Н. Горитов С.М.А. Сглаживание траекторий перемещения рабочего органа робота манипулятора// Известия Томского политехнического университета. – Томск, 2006 г. № 8. – с. 176-179.
12. Modbus RTU protocol and arduino IO package: A real time implementation of a 3 finger adaptive robot gripper. Amirul Syafiq Sadun, Jamaludin Jalani and Jumadi Abdul Sukor. MATEC Web Conf., 108 (2017) 05004. DOI: <https://doi.org/10.1051/matecconf/2017108>.
13. Exploiting Modbus Protocol in Wired and Wireless Multilevel Communication Architecture. By: Guarese, Giuliano B. M.; Sieben, Felipe G.; Webber, Thais; et al. 2012 BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEM ENGINEERING (SBESC 2012) Book Series: Brazilian.
14. https://github.com/Shamraev/Robko_01_source [Электронный ресурс]
15. https://youtu.be/lId4_X3kYyYM [Электронный ресурс]