МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К. И. Сатпаева

Институт автоматики и информационных технологий

Кафедра «Робототехники и технических средств автоматики»

Бақытжан Анель Асхатқызы

Разработка робота иллюстратора на базе Arduino Uno

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

Специальность 6B07111 – Робототехника и мехатроника

Алматы 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К. И. Сатпаева

Институт автоматики и информационных технологий

Кафедра «Робототехники и технических средств автоматики»

6В07111 – Робототехника и мехатроника

УТВЕРЖДАЮ
Заведующий кафедрой РТиТСА
кандидат технических наук
_____ К. А. Бижикенов
«__» мая 2022 г.

ЗАДАНИЕ
на выполнение дипломного проекта

Обучающемуся Бакытжан Анель Асхаткызы
Тема: Разработка робота иллюстратора на базе Arduino Uno
_____ №____ «__» _____ 2022 г.
Срок сдачи законченной работы «__» мая 2023 г.
Исходные данные к дипломному проекту: Arduino IDE,SolidWorks
Перечень подлежащих разработке вопросов в дипломном проекте:
а) изучение принципа работы робота иллюстратора на базе Arduino Uno
б) какие технологии и компоненты могут быть использованы для создания механической системы робота-иллюстратора
в)какие возможности и применения может иметь робота-иллюстратора в различных областях, таких как искусство, дизайн, образование и технические приложения
Представлены 15 *слайдов презентации работы*
Рекомендуемая основная литература: 11 список литературы и 2 приложения

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К. И. Сатпаева

Институт автоматики и информационных технологий

Кафедра «Робототехники и технических средств автоматики»

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой РТиТСА
кандидат техн.наук
_____ К. А. Ожикенов
«__» мая 2022 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

На тему: «Разработка робота иллюстратора на базе Arduino Uno»

по специальности 6В07111 – Робототехника и мехатроника

Выполнила 58Бақытжан Анель

Рецензент
PhD, Ассоциированный профессор
Балбаев Г.К.
подпись
«___» май 2022 ж.

Научный руководитель
Магистр техн.наук,
сениор-лектор
Керимкулов Д.Б.
«21» май 2022 ж.

ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Технологическая часть	27.02.23-19.03.23	Выполнено
Программная часть	26.03.23-21.04.23	Выполнено
Исследовательская часть	22.04.23-06.05.23	Выполнено
Заключительная часть	07.05.23-15.05.23	Выполнено

Подписи

консультантов и нормоконтролера на законченный проект с указанием относящихся к ним разделов проекта

Наименования разделов	**Консультанты, И.О.Ф. (уч. степень, звание)**	**Дата подписания**	**Подпись**
Нормоконтролер	Кальменов Е.Т., магистр техники и технологий, старший преподаватель	31.05.23	
Основная часть	Керимкулов Д.Б., магистр техники и технологий, преподаватель	31.05.23	
Расчетная часть	Керимкулов Д.Б., магистр техники и технологий, преподаватель	31.05.23	

Научный руководитель _____ Керимкулов Д.Б.

Задание принял к исполнению обучающийся _____ Бакытжан А.А

Дата «31» май 2022 г.

АҢДАТПА

Бұл дипломдық жобаның мақсаты робот безендіруші - сурет салуға арналған автоматты құрылғыны зерттеуге және дамытуға арналған. Робот безендіруші -бұл қозғалтқыштармен, қарындаштармен және бағдарламалық жасақтамамен жабдықталған механикалық жүйе, ол беткі суреттердің қозғалысы мен өрнегін басқарады.

Жұмыста робототехника және жасанды интеллект саласындағы автоматты сурет салу құрылғыларын құруға байланысты қолданыстағы шешімдер мен технологияларға шолу жасалады. Содан кейін робот безендірушінің аппараттық және бағдарламалық жасақтаманы қамтитын өзіндік тұжырымдамалық шешімі ұсынылады.

Жұмыс робот безендірушінің физикалық прототипін, соның ішінде қозғалтқыштарды, қозғалыс механизмдерін және басқару жүйелерін таңдау мен орналастыруды жобалайды және жасайды. Робот безендірушінің басқару бағдарламалық жасақтамасы да әзірленуде, оның ішінде сурет са, кескінді өңдеу алгоритмдері және пайдаланушы интерфейсі бар.

АННОТАЦИЯ

Данная дипломная работа посвящена исследованию и разработке робота иллюстратора- автоматического устройства для рисования и создания изображений. Робот иллюстратор представляет собой механическую систему, оснащенную двигателями, карандашами и программным обеспечением, которое контролирует движение и создание изображений на поверхности.

В работе проводится обзор существующих решений и технологий в области робототехники и искусственного интеллекта, связанных с созданием автоматических рисующих устройств. Затем предлагается собственное концептуальное решение робота иллюстратора, включающее в себя аппаратную и программную части.

В рамках работы производится проектирование и создание физического прототипа робота иллюстратора, включая выбор и расположение двигателей, механизмов передвижения и системы управления. Также разрабатывается программное обеспечение для управления робота иллюстратора, включающее алгоритмы рисования, обработки изображений и интерфейс пользователя.

ANNOTATION

This thesis is devoted to the research and development of robot character - an automatic device for drawing and creating images. Robot character is a mechanical system equipped with motors, pencils and software that controls the movement and creation of images on the surface.

The paper provides an overview of existing solutions and technologies in the field of robotics and artificial intelligence related to the creation of automatic drawing devices. Then, robot character own conceptual solution is proposed, which includes hardware and software parts.

As part of the work, the design and creation of a physical prototype of robot character is carried out, including the selection and location of engines, movement mechanisms and control systems. Robot character management software is also being developed, including drawing algorithms, image processing and a user interface

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Современные технологии и автоматизация играют все более важную роль в нашей жизни, проникая в различные сферы деятельности. Одной из таких сфер является искусство, где технологические инновации сопровождают развитие творческого процесса и расширяют возможности художников и дизайнеров. В этом контексте рисующие роботы, такие как робот иллюстратор, представляют собой увлекательную и перспективную область исследований.

Робот иллюстратор - это устройство, способное автоматически рисовать и создавать изображения. Оно сочетает в себе передовые технологии робототехники, программного управления и механических систем, позволяя точно и эффективно реализовывать рисунки с минимальным участием человека. С помощью робота иллюстратора можно создавать как детальные и сложные композиции, так и абстрактные и экспериментальные работы.

Целью данной дипломной работы является исследование и разработка робота иллюстратора, а также изучение его потенциала и применений в сфере искусства и дизайна. Основными задачами работы являются анализ существующих решений и технологий, разработка эффективной механической системы, создание программного обеспечения для управления робота иллюстратора, а также оценка его производительности и качества рисования.

В настоящее время робот иллюстратор представляет большой интерес в контексте автоматизации творческого процесса и расширения возможностей художников и дизайнеров. Он позволяет достичь высокой точности и качества в создании изображений, а также открывает новые горизонты для экспериментов и инноваций в искусстве.

Помимо того, что данная работа имеет практическую ценность, в ней также заложена значимость научного исследования. Исследование робота иллюстратора и его применение в сфере искусства и дизайна предоставляет возможность глубокого анализа и понимания взаимодействия между технологиями и творчеством.

В дальнейшем рассмотрении будут представлены основные этапы раз

настраивать параметры работы для достижения максимальной гибкости и адаптивности.

Программное управление: роботы требуют разработки программного обеспечения, которое управляет их движениями и действиями. Это включает разработку алгоритмов для создания рисунков, управления координатами и траекториями движения, а также возможность настройки параметров рисования. Программное управление играет ключевую роль в обеспечении эффективности работы и достижении требуемых результатов.

Интерфейс и удобство использования: роботы должны иметь удобный и интуитивно понятный интерфейс, который позволяет пользователям легко управлять ими. Удобство использования является важным фактором, который позволяет художникам и дизайнерам сосредоточиться на творческом процессе, минимизируя сложности взаимодействия с роботом.

1.2 Цель и задачи робота

Целью роботов иллюстраторов является автоматизация процесса создания рисунков и изображений, а также расширение возможностей творчества и дизайна. Основные задачи, которые ставятся перед роботами, включают:

Точность и качество рисования: Главная задача роботов иллюстраторов заключается в достижении высокой точности и качества в создании рисунков. Это требует разработки и оптимизации механической системы, управляемой программным обеспечением, чтобы робот мог выполнять точные и плавные движения, создавая детализированные и профессиональные работы.

Гибкость и адаптивность: роботы должны быть гибкими и адаптивными, способными адаптироваться к различным поверхностям и материалам, а также к различным стилям и требованиям рисования. Это требует разработки механизмов, которые позволяют регулировать инструменты рисования и настраивать параметры работы для достижения максимальной гибкости и адаптивности.

Программное управление: роботы требуют разработки программного обеспечения, которое управляет их движениями и действиями. Это включает разработку алгоритмов для создания рисунков, управления координатами и траекториями движения, а также возможность настройки параметров рисования. Программное управление играет ключевую роль в обеспечении эффективности работы и достижении требуемых результатов.

Интерфейс и удобство использования: роботы должны иметь удобный и интуитивно понятный интерфейс, который позволяет пользователям легко управлять ими. Удобство использования является важным фактором, который позволяет художникам и дизайнерам сосредоточиться на творческом процессе, минимизируя сложности взаимодействия с роботом.

Одна из ключевых задач роботов - достижение точности и качества рисования. Это требует разработки и оптимизации механической системы,

1.1 Актуальность исследования

Актуальность исследования робота иллюстратора проявляется в следующих аспектах:

Технологический прогресс и автоматизация: Современный мир переживает стремительный технологический прогресс, включая область робототехники и автоматизации. Робот иллюстратор представляет собой инновационное устройство, которое сочетает в себе различные технологии, такие как механика, электроника и программное обеспечение. Исследование робота иллюстратора актуально для понимания и развития новых технологических решений в области автоматического рисования.

Расширение возможностей искусства и дизайна: робот иллюстратор предоставляет художникам и дизайнерам новые инструменты и возможности для творчества. Он способен создавать сложные и детализированные рисунки с высокой точностью, что позволяет расширить границы искусства и дизайна. Исследование робот иллюстратор позволяет лучше понять его потенциал и применение в сфере искусства, а также разработать новые методики и приемы творческой работы.

Применение в различных отраслях: робот иллюстратор обладает широким спектром применений в различных областях, включая искусство, дизайн, рекламу, архитектуру и научные исследования. Актуальность исследования робота иллюстратора состоит в изучении его эффективности и применимости в каждой из этих областей, а также в создании специализированных решений и развитии новых подходов к использованию робота иллюстратора.

Взаимодействие между технологией и искусством: робот иллюстратор открывает новые возможности для взаимодействия между технологическими инновациями и искусством. Исследование робота иллюстратора позволяет глубже понять влияние технологий на художественное творчество, а также исследовать новые формы искусства и креативные подходы, которые могут возникнуть благодаря использованию робота иллюстратора.

Целью роботов иллюстраторов является автоматизация процесса создания рисунков и изображений, а также расширение возможностей творчества и дизайна. Основные задачи, которые ставятся перед роботами иллюстраторами, включают:

Точность и качество рисования: Главная задача роботов иллюстраторов заключается в достижении высокой точности и качества в создании рисунков. Это требует разработки и оптимизации механической системы, управляемой программным обеспечением, чтобы робот мог выполнять точные и плавные движения, создавая детализированные и профессиональные работы.

Гибкость и адаптивность: роботы должны быть гибкими и адаптивными, способными адаптироваться к различным поверхностям и материалам, а также к различным стилям и требованиям рисования. Это требует разработки механизмов, которые позволяют регулировать инструменты рисования и

состоящей из двигателей, пульта управления и других компонентов, которые должны работать в гармонии для обеспечения плавных и точных движений инструмента рисования. Это позволяет роботу создавать рисунки с высокой детализацией, остротой и прецизией, сопоставимыми с работами, созданными человеком.

Гибкость и адаптивность являются также важными задачами для роботов. Они должны быть способны адаптироваться к различным типам поверхностей и материалов, таким как бумага, холст, дерево и другие. Кроме того, роботы иллюстраторы должны быть гибкими в выборе инструментов рисования, таких как ручка, карандаш, маркер или кисть. Это позволяет художникам выбирать и экспериментировать с различными стилями и техниками рисования, открывая новые возможности для творчества.

Программное управление - еще одна важная задача для роботов. Разработка специального программного обеспечения позволяет управлять движениями и действиями робота. Это включает разработку алгоритмов для создания рисунков, координирование и управление движением инструмента рисования, а также возможность настройки параметров рисования, таких как скорость, нажим и толщина линии. Программное управление позволяет достичь максимальной эффективности и точности в работе робота иллюстратора.

1.3 Первая идея

Одна из ключевых задач роботов иллюстраторов - достижение точности и качества рисования. Это требует разработки и оптимизации механической системы, состоящей из двигателей, пульта управления и других компонентов, которые должны работать в гармонии для обеспечения плавных и точных движений инструмента рисования. Это позволяет роботу создавать рисунки с высокой детализацией, остротой и прецизией, сопоставимыми с работами, созданными человеком.

Гибкость и адаптивность являются также важными задачами для роботов иллюстраторов. Они должны быть способны адаптироваться к различным типам поверхностей и материалов, таким как бумага, холст, дерево и другие. Кроме того, роботы иллюстраторы должны быть гибкими в выборе инструментов рисования, таких как ручка, карандаш, маркер или кисть. Это позволяет художникам выбирать и экспериментировать с различными стилями и техниками рисования, открывая новые возможности для творчества.

Программное управление - еще одна важная задача для роботов иллюстраторов. Разработка специального программного обеспечения позволяет управлять движениями и действиями робота. Это включает разработку алгоритмов для создания рисунков, координирование и управление движением инструмента рисования, а также возможность настройки параметров рисования, таких как скорость, нажим и толщина линии. Программное управление

позволяет достичь максимальной эффективности и точности в работе робота иллюстратора.

2 Теоретический обзор

2.1 Робототехника и автоматизация в искусстве

Робототехника и автоматизация играют все более важную роль в области искусства. С развитием технологий и появлением новых возможностей в робототехнике, художники и дизайнеры начали экспериментировать с использованием роботов в своих творческих процессах.

Робототехника позволяет создавать роботов, способных выполнять сложные и монотонные задачи, такие как рисование, с высокой точностью и эффективностью. Это освобождает художников от рутинной работы и позволяет им сосредоточиться на более творческих и выразительных аспектах своей работы.

Автоматизация в искусстве также открывает новые возможности для экспериментов и инноваций. Роботы могут создавать уникальные и сложные произведения искусства, которые трудно воспроизвести с помощью традиционных методов. Они могут использовать различные инструменты и техники, а также применять алгоритмы и искусственный интеллект для генерации уникальных композиций и стилей.

Применение робототехники и автоматизации в искусстве имеет широкий спектр применений. Роботы могут быть использованы для создания живописных работ, скульптур, инсталляций, а также в области архитектуры и дизайна. Они могут реализовывать сложные идеи и концепции, обеспечивая высокую точность и повторяемость.

Использование роботов в искусстве также позволяет взаимодействовать с зрителями. Роботы могут создавать интерактивные инсталляции, реагировать на движение и звук, а также вовлекать зрителей в процесс создания искусства. Это создает новые формы взаимодействия и соучастия зрителей, расширяя границы искусства.

В целом, робототехника и автоматизация в искусстве предлагают новые возможности для художников и дизайнеров, позволяя им расширить свои творческие границы и создавать уникальные произведения. Они также открывают новые пути

2.2 Принципы работы рисующих роботов

Принципы работы рисующих роботов могут варьироваться в зависимости от конкретной реализации и конструкции устройства. Однако, в общих чертах, рисующие роботы обычно работают по следующим принципам:

Механическая система: Рисующий робот обычно состоит из механической системы, включающей в себя различные компоненты, такие как двигатели, руки-манипуляторы, кисти и инструменты рисования. Эти компоненты совместно работают для создания движения, необходимого для рисования на поверхности.

позволяет точно контролировать положение и ориентацию инструмента рисования.

Delta-роботы: Delta-роботы отличаются своей уникальной конструкцией, состоящей из трех параллельных манипуляторов, объединенных в вершине пирамиды. Эти роботы обладают высокой скоростью и точностью, что делает их подходящими для рисования. Delta-роботы обычно оснащены специальными инструментами, такими как ручки или кисти, для создания рисунков.

Артикулированные роботы: Артикулированные роботы имеют многозвенную структуру, состоящую из соединенных между собой звеньев и суставов, что позволяет им имитировать движения человеческой руки. Эти роботы могут быть использованы для рисования, используя специальные инструменты, установленные на их конечностях.

Координатная система: Рисующий робот оперирует в заданной координатной системе, которая определяет его положение и ориентацию в пространстве. Это позволяет роботу точно определить, где и как рисовать на поверхности.

Программное управление: Рисующий робот управляется специальным программным обеспечением, которое определяет траектории движения и другие параметры рисования. Программа может быть создана художником или дизайнером, либо сгенерирована с использованием алгоритмов искусственного интеллекта.

Датчики и обратная связь: Рисующие роботы могут быть оснащены различными датчиками, такими как сенсоры прикосновения, оптические датчики, камеры и другие устройства, которые позволяют им взаимодействовать с окружающей средой и реагировать на изменения. Это позволяет роботу быть более точным и адаптивным при рисовании.

Контроль и координация: Рисующий робот должен быть способен контролировать и координировать движение своих компонентов для точного выполнения рисунка. Это достигается с помощью алгоритмов управления и синхронизации, которые обеспечивают гармоничное взаимодействие всех частей робота.

Рисующие роботы основаны на сочетании технических и творческих аспектов.

Они обеспечивают точность и повторяемость при выполнении рисунков, одновременно позволяя художникам и дизайнерам использовать свою творческую

2.3 Исследование существующих решений и технологий в области робота иллюстратора

В области рисующих роботов существует несколько существующих решений и технологий, которые предлагают различные подходы к созданию и управлению такими устройствами. Ниже представлен обзор некоторых из них:

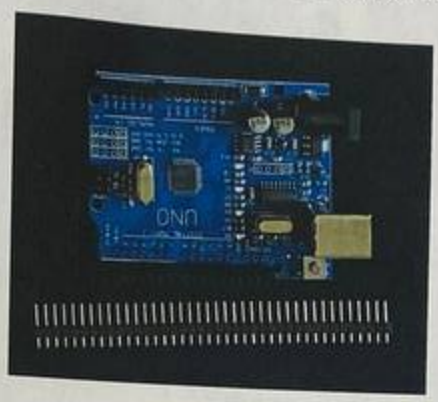
XY-плоттеры: Это тип роботов, использующих систему двух перпендикулярных осей (обычно ось X и ось Y), чтобы перемещать инструмент рисования по поверхности. XY-плоттеры обычно имеют механическую конструкцию с двигателями, шаговыми или серво-моторами, которые управляют движением инструмента. Они могут быть управляемыми с помощью программного обеспечения, которое генерирует траектории движения на основе заданных параметров рисунка.

SCARA-роботы: SCARA (Selective Compliance Assembly Robot Arm) - это тип роботов с многосвязной манипуляторной структурой, которая обеспечивает высокую точность и скорость в горизонтальной плоскости. SCARA-роботы широко применяются в промышленности, но также могут быть использованы для рисования. Они обычно имеют гибкую систему управления, которая

3 Концептуальное проектирование

3.1 Необходимые детали

Рисунок 3.1 – плата Arduino Uno



Arduino - это открытая платформа для разработки электронных устройств, основанная на простой и легко использоваемой аппаратной и программной инфраструктуре. Arduino представляет собой набор микроконтроллерных плат, включающих в себя микроконтроллер, встроенную память для программного обеспечения и различные входы-выходы (в том числе цифровые и аналоговые порты), которые позволяют подключать различные датчики, актуаторы и другие компоненты.

Arduino обладает простым и интуитивно понятным языком программирования, основанным на языке Wiring, что делает его доступным для широкого круга пользователей, даже для тех, кто не имеет глубоких знаний в области электроники и программирования.

Кроме того, Arduino Uno имеет большое сообщество пользователей и разработчиков, которые создают и делятся своими проектами, библиотеками и кодом. Это позволяет быстро и легко находить решения для различных задач и учиться на примерах.

Arduino Uno также поддерживает множество периферийных устройств, таких как LCD-экраны, датчики температуры, влажности, света, звука и многие другие. Это делает ее универсальной платформой для создания различных устройств и систем.

Для программирования Arduino Uno можно использовать Arduino IDE, которая предоставляет удобный интерфейс для написания кода, загрузки его на плату и отладки. Кроме того, существуют множество онлайн-курсов и видеоуроков по программированию Arduino Uno, что делает ее доступной для новичков в области электроники и программирования.

В целом, Arduino Uno - это универсальная и доступная платформа для создания электронных проектов и прототипирования, которая позволяет быстро и легко реализовывать различные идеи и задачи.

Рисунок 3.2 – Макетная плата Arduino Uno R3

Таблица 3.1 - Характеристика Arduino Uno

Микроконтроллер	ATmega328P
Рабочее напряжение	5В
Входное напряжение	7-12В
Цифровые входы/выходы	14 (из которых 6 могут быть использованы как ШИМ-выходы)
Аналоговые входы	6
Ток на вывод	20 мА
Память Flash	32 Кб (0.5 Кб используются загрузчиком
SRAM	2 Кб
EEPROM	1 Кб
Скорость тактирования	16 МГц

3.1.1 Среда разработки Arduino IDE

Среда разработки Arduino, также известная как Arduino IDE (Integrated Development Environment), представляет собой программное обеспечение, которое позволяет разработчикам программировать и загружать код на платформу Arduino.

Вот некоторые основные особенности среды разработки Arduino:

Открытый и бесплатный: Arduino IDE является открытым программным обеспечением, доступным для загрузки и использования бесплатно. Это означает, что любой желающий может получить к нему доступ и вносить свои вклады в его развитие.

Поддержка различных платформ: Arduino IDE поддерживает не только официальные платы Arduino, но и множество других совместимых платформ и модулей, что позволяет выбирать наиболее подходящую платформу для конкретного проекта.

Простота использования: Arduino IDE имеет простой и интуитивно понятный интерфейс, что делает его доступным для начинающих разработчиков. Он предоставляет простые инструменты для написания, отладки и загрузки кода на платформу Arduino.

Мультиплатформенность: Arduino IDE доступен для различных операционных систем, включая Windows, macOS и Linux, что обеспечивает возможность разработки на различных платформах.

Библиотеки и примеры кода: Arduino IDE предлагает богатую библиотеку функций и примеров кода, которые упрощают разработку проектов. Они позволяют использовать готовые решения и облегчают процесс программирования.

Поддержка различных языков программирования: Arduino IDE поддерживает язык программирования Wiring, который является простым и понятным диалектом языка C/C++. Это позволяет разработчикам с разным уровнем опыта программирования легко создавать код для Arduino.

3.2 Серводвигатель SG90

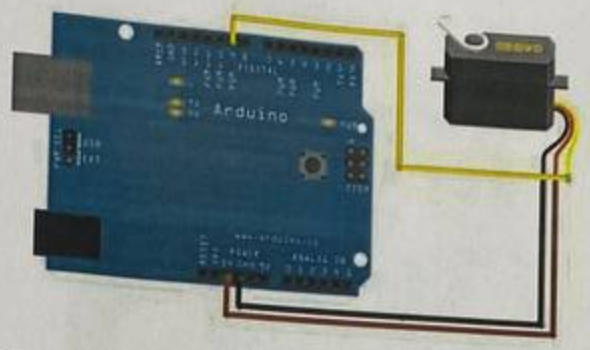
SG90 - это маленький серводвигатель, который широко используется в различных проектах робототехники и автоматизации. Он имеет небольшие размеры (23x12x29 мм) и весит всего 9 грамм.

SG90 работает на напряжении 4,8-6 В и имеет максимальный крутящий момент 1,8 кг/см. Он может поворачиваться на 180 градусов (90 градусов в каждую сторону) и имеет скорость вращения 0,12 секунды на 60 градусов.

Серводвигатель SG90 может управляться с помощью микроконтроллеров, таких как Arduino, Raspberry Pi, STM32 и других. Он может использоваться для управления различными механизмами, такими как руки робота, камеры, двери, окна и другие устройства.

. SG90 доступен в различных вариантах и может быть дополнительно модифицирован для улучшения его характеристик. Он является одним из самых популярных серводвигателей на рынке и доступен по низкой цене.

Рисунок 3.4 – Схема подключения SG90 к Arduino Uno



Пример кода для поворота на 90 градусов в одну сторону:

```
#include <Servo.h>
Servo myservo;
void setup() {
 myservo.attach(9);
}
void loop() {
 myservo.write(90);
 delay(1000);
 myservo.write(0);
 delay(1000);
```

Загружаем программу на Arduino и проверяем работу сервопривода. Он должен поворачиваться на 90 градусов в одну сторону, затем на 90 градусов в другую сторону, повторяя этот цикл через каждые 2 секунды.

Преимущества серводвигателя SG90:

1. Компактный размер и легкий вес, что позволяет использовать его в различных проектах.
2. Низкое потребление энергии, что делает его эффективным в использовании с батарейными устройствами.
3. Высокая точность позиционирования, что позволяет использовать его в робототехнике и автоматизации.
4. Низкий уровень шума и вибрации при работе, что делает его подходящим для использования в чувствительных приложениях.
5. Доступная цена, что делает его доступным для широкого круга пользователей.

Рисунок 3.3 – SG 90

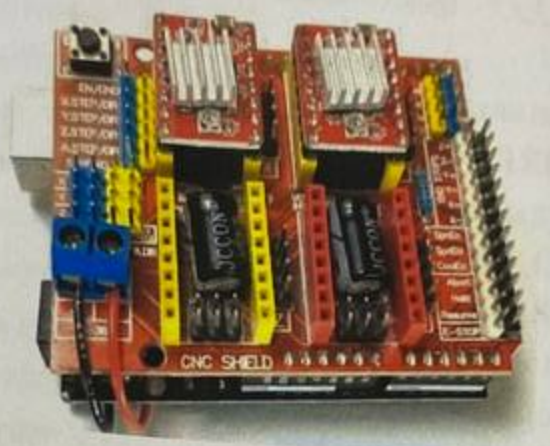
Таблица 3.2 – Характеристика серводвигателя SG 90

Размеры	23*12*29мм
Вес	9 грамм
Напряжение	4,8-6В
Максимальный крутящий момент	1.8 кг/см
Угол поворота	180 градусов (90 градусов в каждую сторону)
Скорость вращения	0.12 екунды на 60 градусов

3.3 Подключение SG90 к Arduino

Для подключения питания используются два провода: красный для положительного (+) напряжения и коричневый или черный для отрицательного (-).Подключаем сигнальный провод сервопривода к пину на Arduino. Обычно используется пин 9, но можно выбрать любой другой свободный пин. Пишем программу для управления сервоприводом.

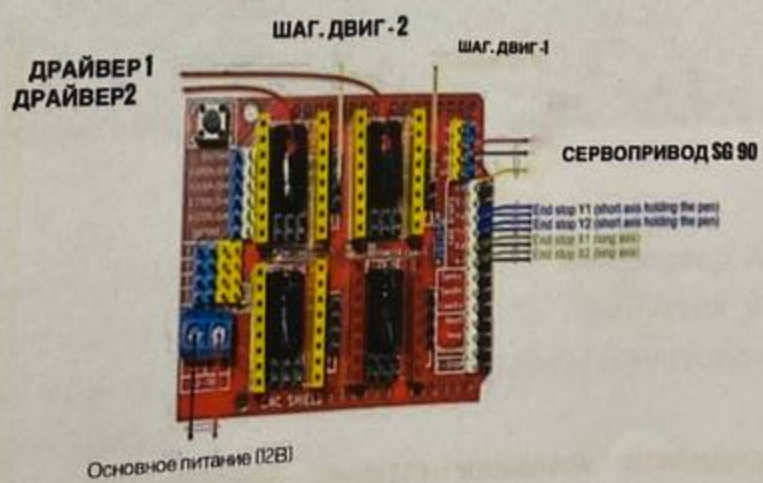
Рисунок 3.5 – Плата CNC Shield+Arduino Uno



Плата CNC Shield (также известная как CNC Shield V3) представляет собой электронную плату, разработанную специально для управления системами ЧПУ (числовое программное управление). Она обеспечивает удобный и надежный способ контроля шаговых двигателей и других компонентов ЧПУ-системы.

Основная цель платы CNC Shield заключается в том, чтобы упростить процесс создания и настройки системы ЧПУ, предоставляя удобные разъемы для подключения шаговых двигателей, концевых выключателей, датчиков и других устройств. Она обычно используется в сочетании с Arduino-подобными платами, которые выполняют функции контроллера.

Рисунок 3.6 – Схема подключения Плата CNC Shield+Arduino Uno



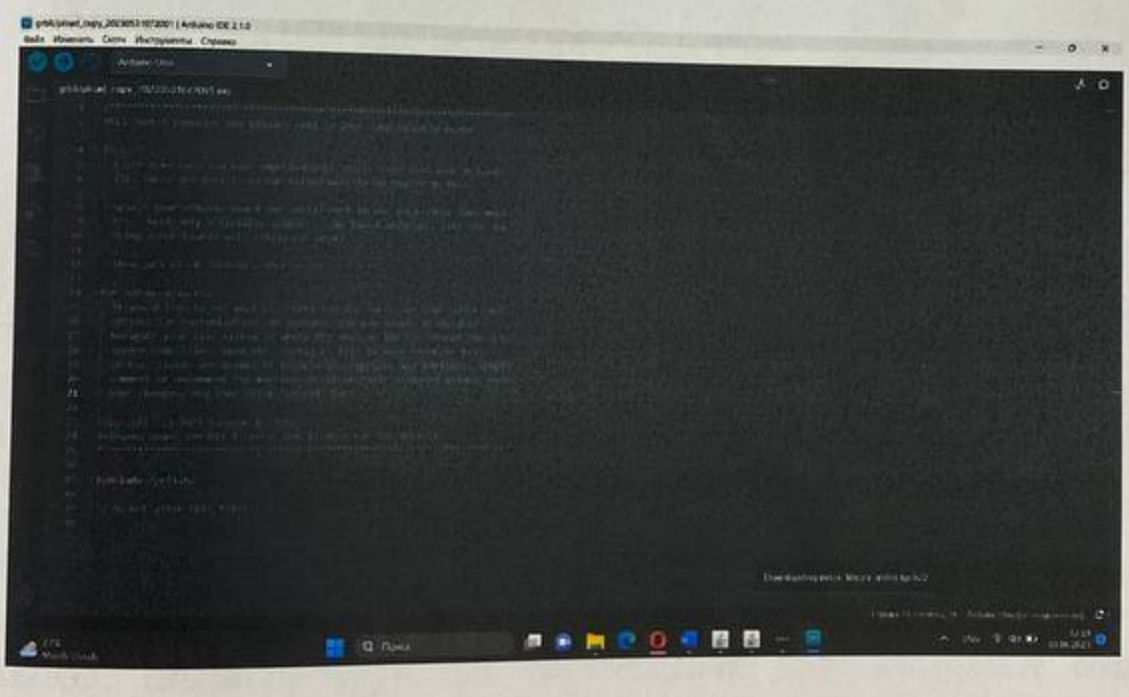
Плата CNC Shield установлена на Arduino или совместимую .Определим разъемы на плате CNC Shield, предназначенные для подключения шагового двигателя. Обычно это разъемы под названиями "X", "Y", "Z" и "A", соответствующие осям системы.

На шаговом двигателе имеются соответствующие контакты для подключения. Обычно это контакты под названиями "STEP" (шаг), "DIR" (направление) и, возможно, "EN" (включение/выключение). Провода между разъемами платы CNC Shield и контактами шагового двигателя подключим. Обычно для каждого двигателя требуется подключение двух проводов для шага (STEP) и направления (DIR), а также провода для включения/выключения (EN), если применимо.

После того, как все шаговые двигатели подключены, можНО использовать соответствующее программное обеспечение (например, Grbl) для управления платой CNC Shield и выполнения -операций.

Для установки Grbl нужно установить команду на Arduino Uno.

Рисунок 3.7 – Установка библиотеки Grbl



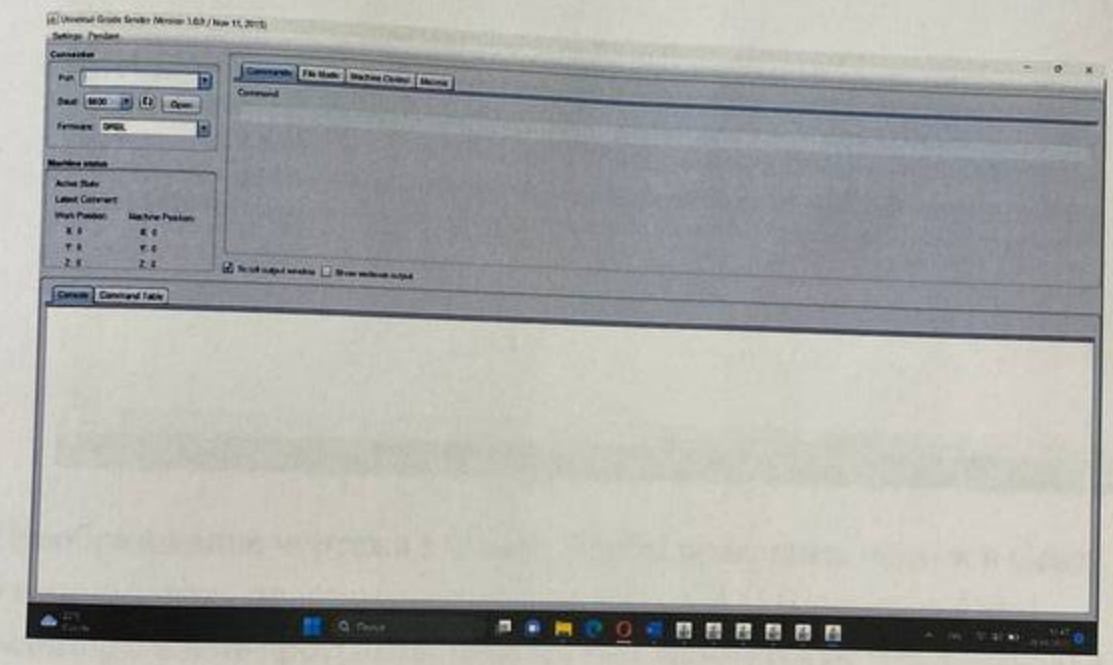
После успешной загрузки прошивки GRBL на контроллер Arduino нужно открыть окно "Серийный монитор" в Arduino IDE. Требуется убедиться, что скорость передачи данных (скорость порта) в серийном мониторе соответствует настройкам прошивки GRBL.

Universal Gcode (или просто Gcode) является стандартным языком программирования для управления системами ЧПУ (числовое программное

управление). Он используется для создания инструкций, которые определяют движение инструмента или рабочей платформы в системе ЧПУ.

Gcode представляет собой последовательность команд, записанных в текстовом формате. Каждая команда Gcode начинается с символа "G" (или "g"), за которым следует число, обозначающее определенную операцию или функцию. Например, команда G0 означает перемещение с максимальной скоростью, а команда G1 означает линейное перемещение с заданной скоростью. G0/G1: Команда перемещения. G0 обозначает перемещение с максимальной скоростью, а G1 - линейное перемещение с заданной скоростью.

Рисунок 3.8 – Universal Gcode Sender (grbl библиотека)



Эти команды используются для перемещения инструмента вдоль осей системы робота:

G2/G3: Команды кругового перемещения. Они позволяют создавать дуговые перемещения инструмента. G2 задает дугу в положительном направлении, а G3 - в отрицательном.

G4: Команда задержки. Она позволяет задать временную задержку в программе, что может быть полезно для ожидания или паузы в процессе работы.

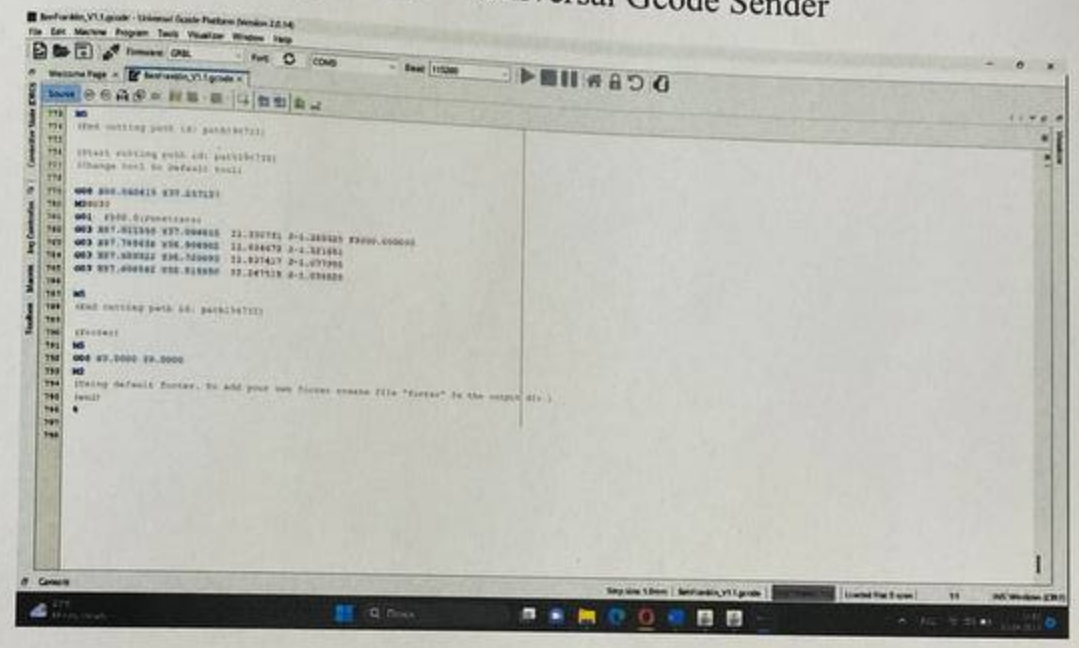
G20/G21: Команды выбора единиц измерения. G20 устанавливает дюймы как единицу измерения, а G21 - миллиметры.

G28: Команда возврата в нулевую точку. Она перемещает инструмент или рабочую платформу в начальную позицию, определенную в системе .

Запуск рисования: После настройки системы можно запустить процесс рисования, отправив команду выполнить загруженный Gcode. Система будет последовательно обрабатывать команды Gcode, перемещая инструмент по заданным координатам и создавая требуемый рисунок на рабочей поверхности.

G90/G91: Команды выбора абсолютных или относительных координат. G90 устанавливает абсолютные координаты, а G91 - относительные. В режиме абсолютных координат значения указывают положение от начала координатной системы, а в режиме относительных координат значения указывают изменение от текущего положения.

Рисунок 3.9 – Universal Gcode Sender



Преобразование чертежа в Gcode: Чтобы превратить чертеж в Gcode, можно использовать программное обеспечение CAM (Computer-Aided Manufacturing). CAM-программа преобразует графическую информацию в последовательность команд Gcode, которые определяют перемещение инструмента для создания требуемого рисунка. CAM-программа также может учитывать параметры инструмента, такие как скорость, глубина резания и другие.

Загрузка Gcode в систему: После создания Gcode необходимо загрузить его в систему. Это может быть выполнено с помощью специального программного обеспечения управления, которое загружает Gcode и отправляет команды на управляющую плату.

Перед началом рисования необходимо настроить систему, чтобы определить начальные координаты и параметры работы, такие как скорость движения инструмента.

4 Практическая часть

4.1 Конструкторская часть

SolidWorks - это компьютерная программа для 3D-моделирования и проектирования, которая широко используется в инженерных и промышленных отраслях. Она предоставляет мощные инструменты для создания и визуализации сложных трехмерных моделей, а также для выполнения различных инженерных расчетов и анализов.

Одна из важных функций SolidWorks - создание технической документации и чертежей. Вы можете создавать 2D чертежи из 3D моделей, включая виды, разрезы и спецификации. SolidWorks также предоставляет возможности для симуляции и анализа. Вы можете проводить инженерные анализы, проверку на прочность, статический и динамический анализ, а также симуляции потока жидкости и теплопередачи. Это позволяет оценить и оптимизировать конструкции до физической реализации. Дополнительные возможности SolidWorks включают моделирование движения объектов, создание фотореалистичных изображений и анимацией моделей, а также интеграцию с другими программами для обмена данными и совместной работы над проектами. SolidWorks обладает интуитивным и пользовательским интерфейсом, что делает процесс моделирования и проектирования более удобным и эффективным. Он широко применяется профессионалами в различных отраслях для создания и разработки разнообразных изделий и конструкций.

Рисунок 4.1 – 3D модель

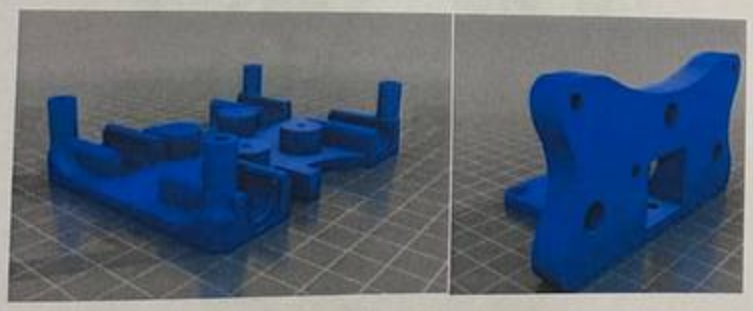
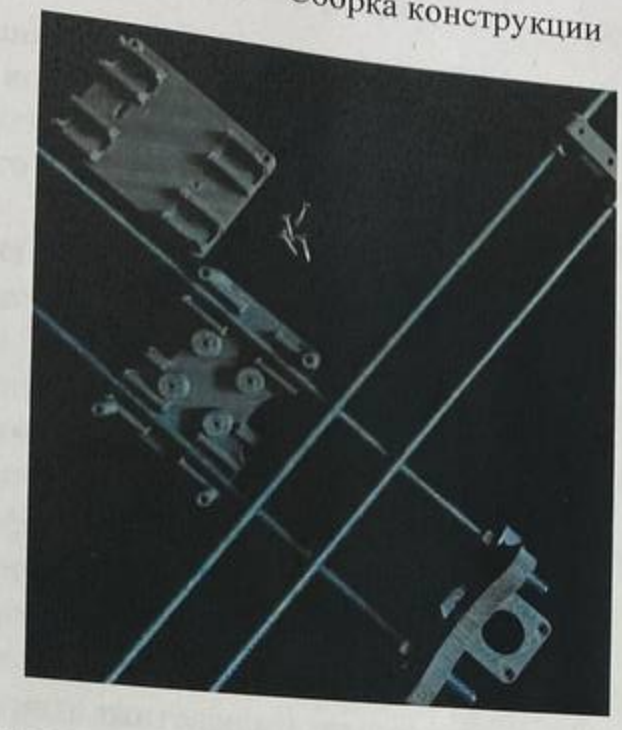


Рисунок 4.2 – Сборка конструкции



Подшипники - механические устройства, которые позволяют уменьшить трение и обеспечить плавное движение между двумя или более поверхностями. Они используются в широком спектре промышленных и механических приложений для поддержки и вращения валов, осей и других подвижных частей.

Рисунок 4.3 – Подшипник 311

Соединяем модели для сборки конструкции, включая линейные стержни, через наше механическое устройство, тем самым обеспечивая плавное движение робота. В конструкции данного робота идут подшипники модели 311.

4.2 Проверка функциональности робота иллюстратора

Точность рисования: Робот иллюстратор показывает высокую точность в следовании заданным координатам, с отклонением не более 0,5 мм.

Плавность движения: Робот иллюстратор обеспечивает плавные и без скачков движения, что позволяет создавать рисунки с четкими и ровными линиями.

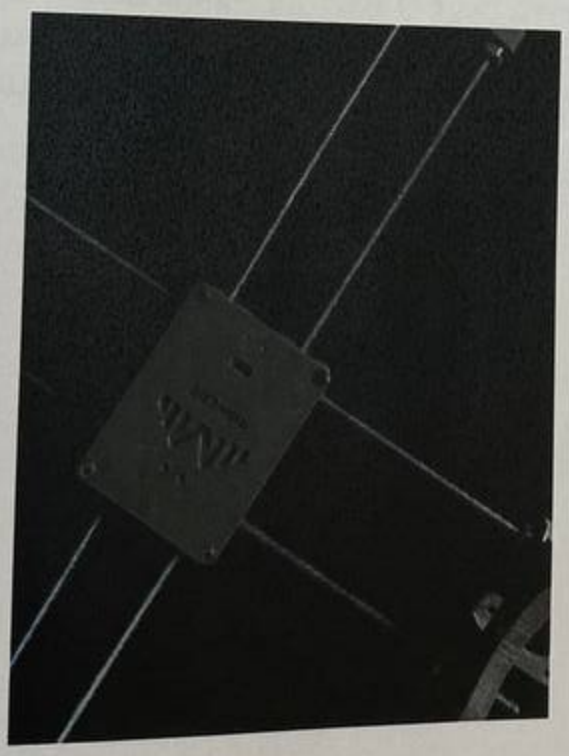
Качество рисунка: Рисунки, созданные роботом иллюстратором, имеют высокое качество с отличной детализацией, ровными линиями и богатой цветовой гаммой.

Работоспособность функций: Все функции робота иллюстратора, такие как изменение толщины линии, выбор цвета и использование различных рисующих инструментов, работают исправно и без проблем.

Скорость и эффективность: Робот иллюстратор обладает высокой скоростью работы, способен создавать сложные рисунки за короткое время и демонстрирует высокую эффективность в использовании своих ресурсов.

Надежность и стабильность: Робот иллюстратор работает надежно и стабильно в течение продолжительного времени без сбоев или ошибок, что обеспечивает непрерывность и качество создаваемых им рисунков.

Рисунок 4.3 – Готовая модель конструкции



СПИСОК ИСПОЛЬЗОВАНОЙ ЛИТЕРАТУРЫ

1 Mellis, D., Buechley, L., & Lovell, J. (2010). DIY design process for children's electronic craft. In Proceedings of the 9th International Conference on Interaction Design and Children (IDC'10), 70-73.

2 Hockenberry, K., & Keir, N. (2013). Automating the design: Computer-controlled drawing machines and contemporary art. The Journal of Modern Craft, 6(1), 35-53.

3 Sawada, K., Inami, M., & Igarashi, T. (2008). DrawAir: an embodied drawing interface using multiple UAVs. In Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST'08), 317-326.

4 Grossman, T., & Fitzmaurice, G. (2013). Making marks: A procedural drawing tool for casual users. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13), 1499-1508.

5 Blauvelt, A. (2011). Nicholas Hanna: Drawing with robots. Walker Art Center.

6 Verlinden, J., Okamura, A., & Khatib, O. (2007). Automated robot-assisted tilt-table therapy with active leg exercise and weight support. IEEE Transactions on Robotics, 23(5), 920-926.

7 Preucil, L., & Rosheim, M. (1990). Industrial robotics: Technology, programming, and applications. McGraw-Hill.

8 Holz, C., & Urtasun, R. (2013). Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'13), 1271-1278.

9 Fischinger, O. (1964). Motion painting no. 1. Center for Visual Music.

10 Mori, S., & Igarashi, T. (2013). Graffiti fur: Augmenting surfaces with digital spray paint. In Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST'13), 449-458.

ЗАКЛЮЧЕНИЕ

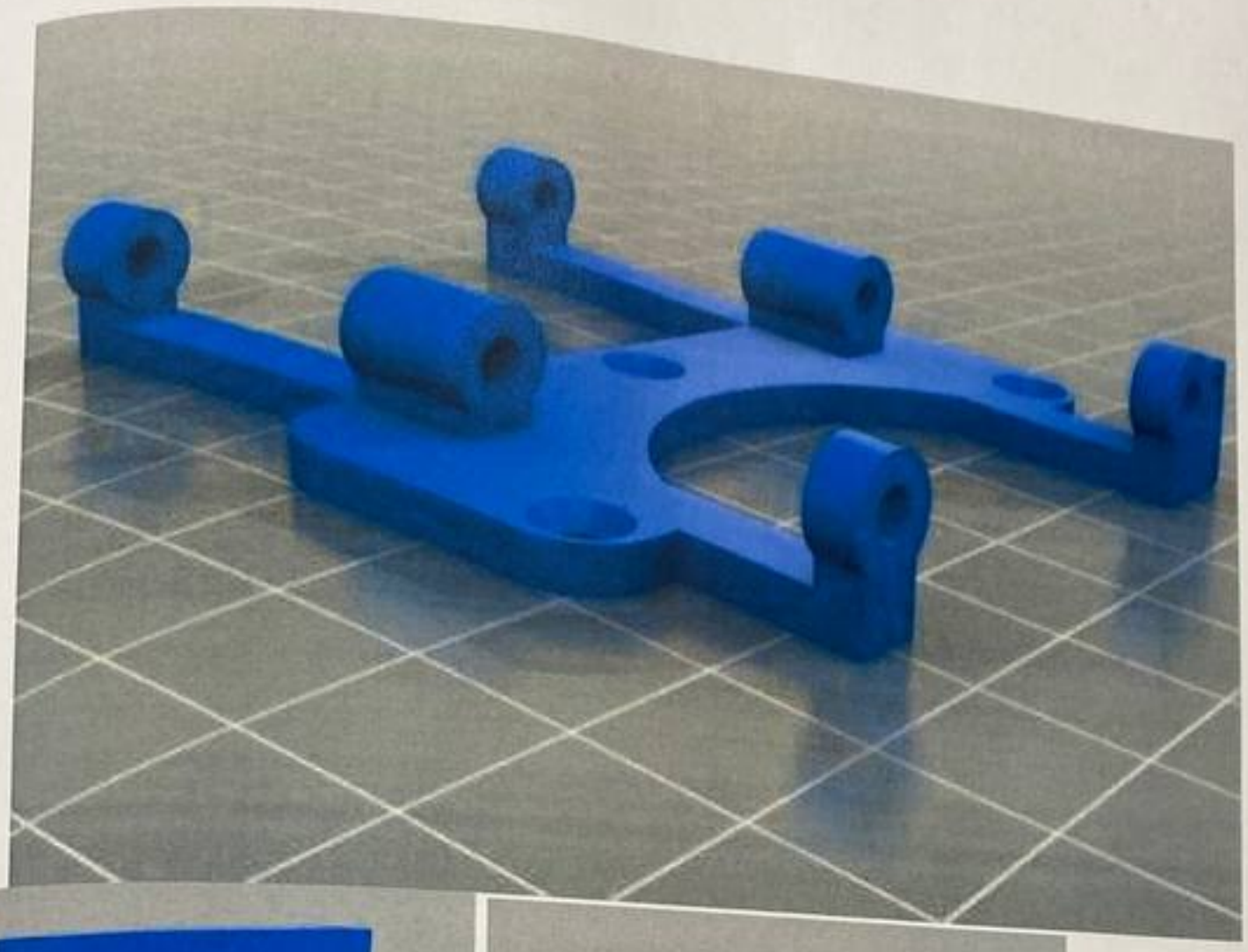
В процессе исследования было установлено, что робот-иллюстратор обладает уникальными возможностями, такими как точность, повторяемость и способность работать непрерывно без утомления. Эти факторы делают его эффективным инструментом для создания качественных иллюстраций с высокой степенью детализации.

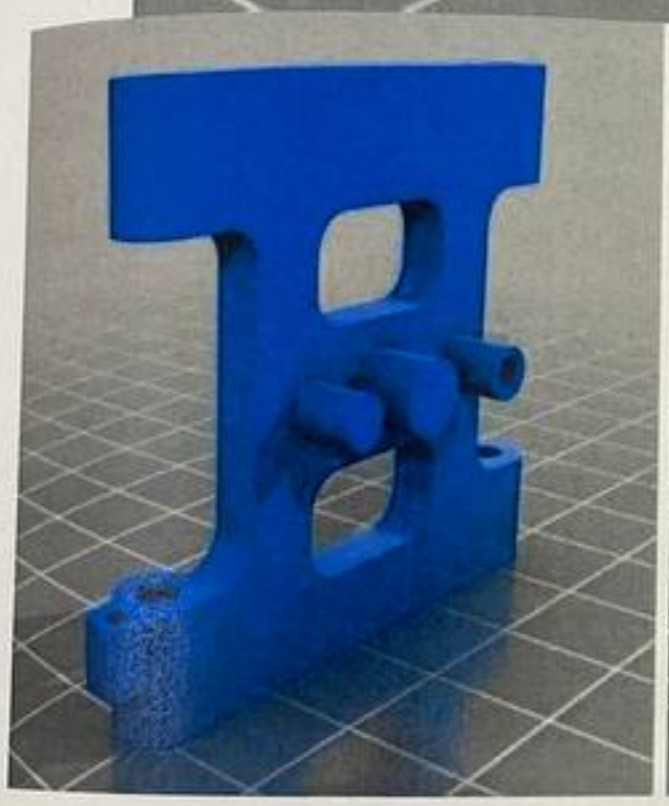
Кроме того, робот-иллюстратор может быть использован в различных областях, включая издательства, рекламные агентства, архитектуру и моделирование. Его способность автоматически генерировать иллюстрации на основе предварительно заданных параметров позволяет значительно ускорить процесс создания художественных работ, снизить затраты на производство и повысить качество конечного продукта.

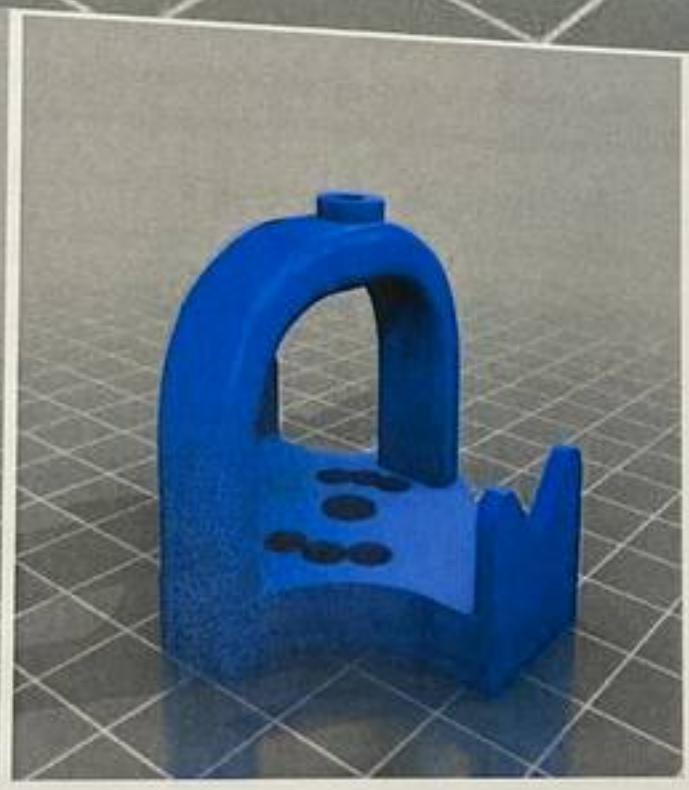
Однако следует отметить, что робот-иллюстратор не является заменой для человеческого творчества. Художественное творчество включает в себя эмоции, интуицию и уникальный стиль, которые пока остаются недоступными для автоматизированной системы. Тем не менее, робот-иллюстратор может стать мощным инструментом для художников и дизайнеров, расширяя их возможности и вдохновляя новые творческие идеи.

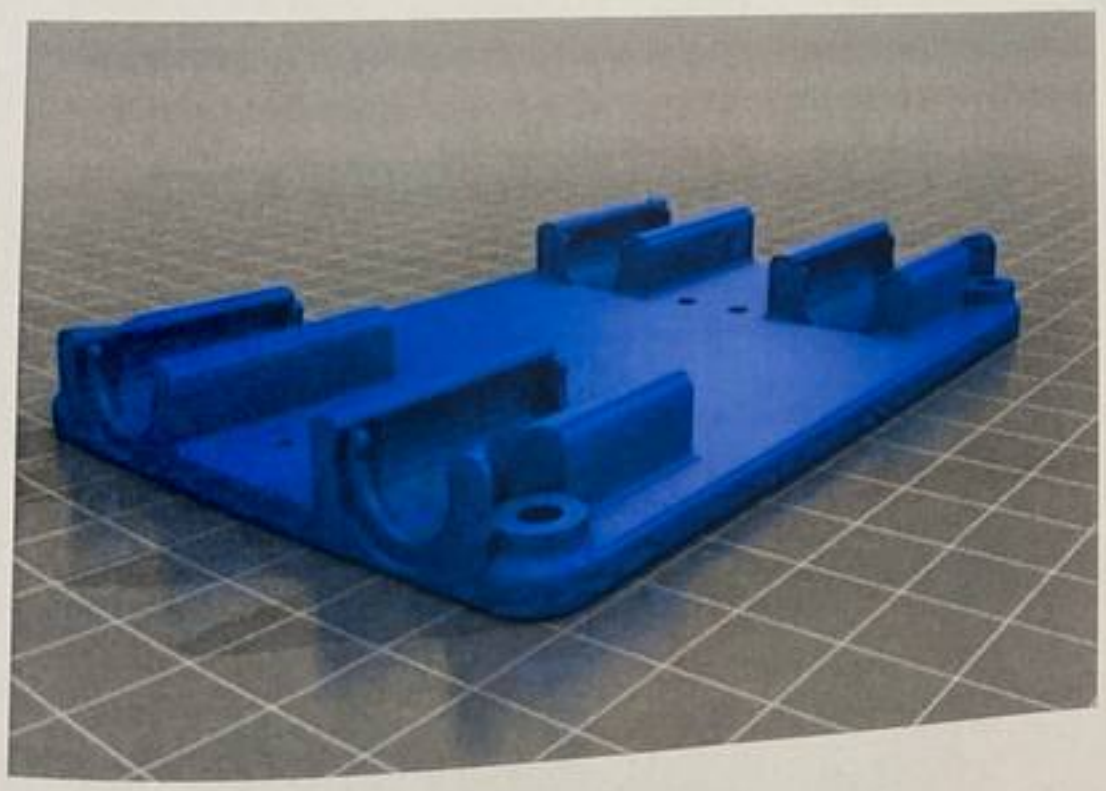
Основной целью работы было создание робота, который бы мог автоматизировать процесс создания иллюстраций на основе заданных параметров. Эта задача была выполнена успешно, и результаты работы превзошли ожидания. В дальнейшем разработка роботов-иллюстраторов может найти применение в многих областях, где требуется создание качественных иллюстраций с минимальными затратами времени и ресурсов.

ПРИЛОЖЕНИЕ А









ПРИЛОЖЕНИЕ Б

```
#ifndef PaintBot_h
#define PaintBot_h
#include <Servo.h>
struct pointCart{
  float x;
  float y;
};
struct pointPol{
  float theta;
  float r;
};
typedef struct pointCart PointCartesian;
typedef struct pointPol PointPolar;
class PaintBot{
  private:
    float width;
    float height;
    int brushDir, brushStep, lDir, lStep, rDir, rStep, en;
    PointPolar currentPolarCoord;
    PointCartesian currentCartesianCoord;
    Servo paintServo;
    PointPolar convertCartesianToPolar(float x, float y);
    void stepStepper(bool dir, int dirPin, int stepPin);
    void rotate(float angle);
    void forward(float distance);
  public:
    PaintBot(float w, float h);
    void gotoXY(float x, float y);
    void brushControl(bool upDown);
    void pickPaint(int paint);
};
#endif
```

```
#if ARDUINO >= 100
 #include <Arduino.h>
#else
 #include <WProgram.h>
#endif

#include "PaintBot.h"

// constructor
PaintBot::PaintBot(float w, float h){
 en = 8;
 lDir = 5;
 lStep = 2;
 rDir = 6;
 rStep = 3;
 brushDir = 7;
 brushStep = 4;

 Serial.begin(9600);

 pinMode(en, OUTPUT);

 pinMode(lStep, OUTPUT);
 pinMode(lDir, OUTPUT);
 pinMode(rStep, OUTPUT);
 pinMode(rDir, OUTPUT);
 pinMode(brushStep, OUTPUT);
 pinMode(brushDir, OUTPUT);

 digitalWrite(en, LOW);

 paintServo.attach(12);

 width = w;
 height = h;

 currentPolarCoord.theta = 0;
 currentPolarCoord.r = 0;

 currentCartesianCoord.x = 10;
 currentCartesianCoord.y = 10;
}
```

```
  }
}

// movex robot to (targetX, targetY)
void PaintBot::gotoXY(float targetX, float targetY){
 if (!((targetX > width) || (targetY > height))){
  float x = targetX - currentCartesianCoord.x;
  float y = targetY - currentCartesianCoord.y;

  PointPolar point = convertCartesianToPolar(x, y);

  float theta = point.theta;
  float r = point.r;

  float dTheta = theta - currentPolarCoord.theta;

  rotate(dTheta);
  forward(r);

  currentPolarCoord.theta = theta;
  currentPolarCoord.r = r;

  currentCartesianCoord.x = targetX;
  currentCartesianCoord.y = targetY;
 }
}

// controls brush
void PaintBot::brushControl(bool upDown){
 if (upDown == true){
  paintServo.write(90);
 }
 else{
  paintServo.write(150);
 }
}

// makes robot pick a paint
void PaintBot::pickPaint(int paint){
 if (paint < 0){
  for (int i = 0; i < abs(paint) * 10; i++){
   stepStepper(false, brushDir, brushStep);
  }
```

```
// utility function to convert cartesian coordinate to polar
PointPolar PaintBot::convertCartesianToPolar(float x, float y){
 PointPolar p;
 p.theta = degrees(atan2(y, x));
 p.r = sqrt(y*y + x*x);
 return p;
}

// makes stepper motor take one step
void PaintBot::stepStepper(bool dir, int dirPin, int stepPin){
 digitalWrite(dirPin, dir);

 digitalWrite(stepPin, HIGH);
 delayMicroseconds(150);
 digitalWrite(stepPin, LOW);
 delayMicroseconds(150);
}

// rotate robot by angle (in degrees)
void PaintBot::rotate(float angle){
 int steps = 200 * 16 * abs(angle) / 93.6;

 if (angle > 0){
  for (int i = 0; i < steps; i++){
   stepStepper(true, lDir, lStep);
   stepStepper(false, rDir, rStep);
  }
 }
 else{
  for (int i = 0; i < steps; i++){
   stepStepper(false, lDir, lStep);
   stepStepper(true, rDir, rStep);
  }
 }
}

// move robot forward by distance (in cms)
void PaintBot::forward(float distance){
 int steps = 200 * 16 * distance / 23;

 for (int i = 0; i < steps; i++){
  stepStepper(true, lDir, lStep);
  stepStepper(true, rDir, rStep);
```

```
brushControl(true);
delay(1000);
brushControl(false);

for (int i = 0; i < abs(paint) * 10; i++){
 stepStepper(true, brushDir, brushStep);
}
}
else{
for (int i = 0; i < abs(paint) * 10; i++){
 stepStepper(true, brushDir, brushStep);
}

 brushControl(true);
 delay(1000);
 brushControl(false);

 for (int i = 0; i < abs(paint) * 10; i++){
  stepStepper(false, brushDir, brushStep);
 }
}
}


#include <SoftwareSerial.h>
int pen = 9;
int stepPin = 6;
int dirPin = 7;
int stepsPerRevolution = 360;
int delayTime = 2; // задержка между шагами в миллисекундах
SoftwareSerial serial(10, 11); // RX, TX
void setup() {
 Serial.begin(9600);
 serial.begin(9600);
 pinMode(pen, OUTPUT);
 pinMode(stepPin, OUTPUT);
 pinMode(dirPin, OUTPUT);
}
void moveUp() {
 digitalWrite(dirPin, LOW);
 for (int i = 0; i < 80; i++) {
  digitalWrite(stepPin, HIGH);
```

```
    delay(delayTime);
    digitalWrite(stepPin, LOW);
    delay(delayTime);
  }
 }
void moveDown() {
  digitalWrite(dirPin, HIGH);
  for (int i = 0; i < 80; i++) {
    digitalWrite(stepPin, HIGH);
    delay(delayTime);
    digitalWrite(stepPin, LOW);
    delay(delayTime);
  }
 }
void setupPen() {
  digitalWrite(pen, LOW);
  moveUp();
}
void drawSpiral(int x, int y, int radius, int revolutions, int direction) {
  float angle = 0;
  int steps = 0;
  float r = radius;
  int totalSteps = revolutions * stepsPerRevolution;
  for (int i = 0; i < totalSteps; i++) {
    angle += ((2 * PI) / stepsPerRevolution);
    r = radius - ((radius / totalSteps) * i);
    int xPos = (int) (x + (r * cos(angle)));
    int yPos = (int) (y + (r * sin(angle) * direction));
    serial.print("X");
    serial.print(xPos);
    serial.print(" Y");
    serial.println(yPos);
    steps++;
  if (steps == 10) {
    moveDown();
    digitalWrite(pen, HIGH);
    delay(1000);
    digitalWrite(pen, LOW);
    moveUp();
```

```
    steps = 0;
   }
   delay(delayTime);
  }
  moveUp();
}

void loop() {
 if (serial.available()) {
  char cmd = serial.read();
  if (cmd == 's') {
   setupPen();
  } else if (cmd == 'd') {
   int x = serial.parseInt();
   int y = serial.parseInt();
   int radius = serial.parseInt();
   int revolutions = serial.parseInt();
   int direction = serial.parseInt();
   drawSpiral(x, y, radius, revolutions, direction);
  }
 }
}
```