МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

"Ижевский государственный технический университет имени М.Т.Калашникова"

(ФГБОУ ВО «ИжГТУ имени М.Т.Калашникова)

Кафедра «Программное обеспечение»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломной работе на тему:

Разработка программного комплекса для управления человекоподобным роботом на платформе Arduino.

Дипломник А.Е. Волков

студент гр. Д19-191-1

Руководитель К.С. Ермолин

Нормоконтролер В.П. Соболева

Ижевск 2023г.

# РЕФЕРАТ

Сведения об объеме пояснительной записки:

Количество страниц – 37;

Количество иллюстраций – 8;

Количество таблиц – 1;

Современные технологии и наука развиваются с каждым годом все быстрее и быстрее, что приводит к появлению новых устройств и механизмов, которые помогают людям в повседневной жизни. Одним из таких устройств является человекоподобный робот, который позволяет выполнять различные задачи, как в промышленности, так и в быту.

Целью данной дипломной работы является разработка программного комплекса для управления человекоподобным роботом на платформе Arduino.

Для управления человекоподобным роботом на платформе Arduino необходимо разработать программный комплекс, который будет обеспечивать управление роботом. Данный комплекс должен содержать несколько модулей, которые будут отвечать за различные функции робота.

Для достижения поставленной цели были поставлены следующие задачи:

1. Разработать модуль передачи данных с ПК на Arduino.
2. Разработать модуль настройки поворотов сервоприводов.
3. Разработать модуль работы с файлами, хранящими позиции сервоприводов.
4. Разработать программу для микроконтроллера Arduino.

На первом этапе работы был разработан модуль передачи данных с компьютера на микроконтроллер Arduino. Данный модуль является основным в программе, с его помощью компьютер передает положения робота на микроконтроллер, а тот, в свою очередь, даёт обратную связь, успешно ли была выполнена команда.

На втором этапе был разработан модуль поворотов сервоприводов. Данный модуль обрабатывает действия пользователя и, в зависимости от сценария, который указал пользователь, отправляет модулю передачи данных положения робота по отдельности или друг за другом циклически.

На третьем этапе был разработан модуль работы с файлами, хранящими позиции сервоприводов. Данный модуль позволяет импортировать в программу положения робота, перезаписать их или сохранить в новый файл. Благодаря этому, появляется возможность хранить лучшие наборы положений робота. Положения робота хранятся в текстовых файлах, что позволяет открыть их или скопировать содержимое используя даже предустановленный в системе текстовый редактор.

На четвертом этапе работы был разработан программный код для управления роботом на платформе Arduino. Для этого был использован язык программирования C++ и Arduino IDE. В результате была разработана программа, которая позволяет управлять всеми движениями робота.

В целом, разработка программного комплекса для управления человекоподобным роботом на платформе Arduino является достаточно сложной задачей, которая требует знаний в области программирования, электроники и механики. Однако, благодаря современным технологиям и инструментам, эта задача может быть успешно решена, что позволит создать уникальное устройство, способное выполнять различные задачи в автоматическом режиме.

# СОДЕРЖАНИЕ

Оглавление

[РЕФЕРАТ 2](#_Toc137120327)

[СОДЕРЖАНИЕ 4](#_Toc137120328)

[ВВЕДЕНИЕ 6](#_Toc137120329)

[1. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА 8](#_Toc137120330)

[1.1. Обоснование целесообразности разработки ПП 8](#_Toc137120331)

[1.2. Аналитический обзор существующих программных продуктов-аналогов 10](#_Toc137120332)

[1.3. Требования к ПП(техническое задание) 12](#_Toc137120333)

[1.3.1. Требования к функциональным характеристикам 12](#_Toc137120334)

[1.3.2. Требования к надежности 15](#_Toc137120335)

[1.3.3. Условия эксплуатации 15](#_Toc137120336)

[1.3.4. Требования к составу и параметрам технических средств 15](#_Toc137120337)

[1.3.5. Требования к информационной и программной совместимости 16](#_Toc137120338)

[1.4. Технические решения проекта ПП 16](#_Toc137120339)

[Описание информационной базы данных: 16](#_Toc137120340)

[1.5. Планирование тестирования программного продукта 17](#_Toc137120341)

[2. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА 21](#_Toc137120342)

[2.1. Разработка модуля передачи данных с ПК на Arduino 21](#_Toc137120343)

[2.1.1. Описание алгоритма 21](#_Toc137120344)

[2.1.1.1. Назначение и характеристика алгоритма 21](#_Toc137120345)

[2.1.1.2. Математическое описание алгоритма 22](#_Toc137120346)

[2.1.1.3. Алгоритм 22](#_Toc137120347)

[2.2. Разработка модуля настройки поворотов сервоприводов. 23](#_Toc137120348)

[2.2.1. Описание алгоритма 23](#_Toc137120349)

[2.2.1.1. Назначение и характеристика алгоритма 23](#_Toc137120350)

[2.2.1.2. Алгоритм 23](#_Toc137120351)

[2.3. Разработка модуля работы с файлами, хранящими позиции сервоприводов. 24](#_Toc137120352)

[2.3.1. Описание алгоритма 24](#_Toc137120353)

[2.3.1.1. Назначение и характеристика алгоритма 24](#_Toc137120354)

[2.3.1.2. Алгоритм 25](#_Toc137120355)

[2.4. Разработка программного кода для управления роботом на платформе Arduino. 26](#_Toc137120356)

[2.4.1. Описание алгоритма 26](#_Toc137120357)

[2.4.1.1. Назначение и характеристика алгоритма 26](#_Toc137120358)

[2.4.1.2. Математическое описание алгоритма 27](#_Toc137120359)

[2.4.1.3. Алгоритм 27](#_Toc137120360)

[СПИСОК ЛИТЕРАТУРЫ 29](#_Toc137120361)

[ПРИЛОЖЕНИЕ А 30](#_Toc137120362)

[ПРИЛОЖЕНИЕ Б 36](#_Toc137120363)

# ВВЕДЕНИЕ

В последние годы робототехника стала одной из наиболее перспективных областей развития технологий. Одним из наиболее интересных объектов в этой области являются человекоподобные роботы, которые способны имитировать движения и поведение человека. Для управления такими роботами необходимо использовать специальные программные комплексы, которые позволяют управлять всеми его функциями и движениями.

Актуальность данной дипломной работы заключается в том, что робототехника является одной из наиболее перспективных областей развития технологий. Человекоподобные роботы могут быть использованы в различных сферах, начиная от медицины и промышленности, и заканчивая развлекательной индустрией. Управление такими роботами требует специальных программных комплексов, которые позволяют управлять всеми их функциями и движениями. Разработка программного комплекса для управления человекоподобным роботом на платформе Arduino является актуальной задачей, которая упрощает процесс настройки робота.

Выбор данной темы обоснован ее практической значимостью. Материалы, представленные в практической части, могут служить основой приложения, помогающего в частных и образовательных целях.

Новизна дипломной работы заключается в том, что данная тема, в интересующем нас аспекте, является применимой для платформы Arduino, в то время как другие разрабатывались под определенные платформы с закрытым исходным кодом.

Цель курсовой работы - упростить настройку робота на платформе Arduino , используя программу с интуитивно наглядным интерфейсом.

Данная система позволяет изучить основы взаимодействия с роботом и настроить его движения под различные задачи, в том числе ходьба и различные сложные положения тела. Данная система помогает быстро задать необходимые положения сервоприводов, не используя сложных математических расчетов и знаний обратной кинематики. Чтобы быстро освоить функционал системы, она имеет наглядный, интуитивный интерфейс.

Для достижения поставленной цели необходимо решить следующие задачи:

* Разработать модуль передачи данных с ПК на Arduino.
* Разработать модуль настройки поворотов сервоприводов.
* Разработать модуль работы с файлами, хранящими позиции сервоприводов.
* Разработать программу для микроконтроллера Arduino.

# ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

# Обоснование целесообразности разработки ПП

* + 1. Назначение ПП

Программный продукт (ПП) предназначен для изучения основ взаимодействия с роботом и настройки его движения под различные задачи, в том числе ходьба и различные сложные положения тела. Данный ПП помогает быстро задать необходимые положения сервоприводов, не используя сложных математических расчетов и знаний обратной кинематики.

Функциональным назначением ПП является предоставление возможностей:

Всем пользователям:

1. Выбор модели робота в зависимости от количества сервоприводов
2. Настройка подключения робота к компьютеру
   1. Выбор COM-порта
   2. Выбор скорости передачи данных,
   3. Открытие и закрытие передачи в COM-порт
3. Настройка положений сервоприводов:
   1. Через изменение положения слайдеров
   2. Через задание числового значения
   3. Через открытие файла с массивом положений сервоприводов
   4. Через функциональные кнопки
4. Сохранение массива положений в текстовом файле.

Ученику:

1. Отправка файла с массивом положений сервоприводов на сервер для проверки преподавателем

Преподавателю:

1. Скачивание файлов с массивами положений сервоприводов с сервера и использование в программе
   * 1. Обоснование цели ПП

На сегодняшний день в открытом доступе находится лишь одно приложение российского разработчика, которое позволяет управлять положением робота и сохранять модели движения. Оно ограничено в функциональности, имеет устаревший интерфейс и не поддерживается разработчиком. Другие приложения поставляются только к определенным моделям готовых роботов и несовместимы с роботами на платформе Arduino.

# Аналитический обзор существующих программных продуктов-аналогов

* + 1. RoboPlus Motion

Программа, поставляемая с моделью робота BIOLOID китайской компании ROBOTIS (см. рис. 1). RoboPlus Motion имеет обширный функционал для настройки робота, а также большое количество готовых движений; программа включает в себя возможность настройки робота как через указание углов поворота сервопривода, так и с использованием обратной кинематики. В программе присутствует 3D визуализация робота, что позволяет настроить робота, не имея его рядом. Не поддерживает русский язык.



Рис.1.1 Главное окно программы RoboPlus Motion

* + 1. RoboPlus Motion 2.0

Вторая версия программы RoboPlus Motion, имеет обновленный интерфейс (см. рис. 2), благодаря которому проще пользоваться программой. Имеет схожую функциональность, что и первая версия, поставляется с последними моделями роботов компании ROBOTIS. Так же не поддерживает русский язык.



Рис.1.2 Главное окно программы RoboPlus Motion 2.0

* + 1. Hiwonder RoboSoul

Программа, поставляемая с роботами китайской компании Hiwonder. Имеет удобный и красивый интерфейс (см. рис. 3). Благодаря наглядности, можно быстро разобраться, как работать с программой. С ее помощью можно настроить положение отдельного сервопривода или сразу нескольких, используя сохранение массива положений сервоприводов. С программой идет множество готовых шаблонов движений для робота. Имеет возможность запуска движений в цикличном исполнении. Так же не поддерживает русский язык.



Рис.1.3 Главное окно программы Hiwonder RoboSoul

# Требования к ПП(техническое задание)

# Требования к функциональным характеристикам

* Управление сервоприводами робота: разработанный программный комплекс должен обеспечивать управление сервоприводами робота.
* Управление режимами работы: ПП должен позволять переключаться между различными режимами работы, включая отправка массивов позиций в реальном времени, отправка массивов позиций друг за другом, отправка одной выбранной позиции.
* Интерфейс управления: программный комплекс должен иметь интуитивно понятный интерфейс управления, который позволяет пользователю легко настраивать параметры работы робота и управлять им.
* Поддержка различных типов роботов: разработанный комплекс должен поддерживать работу с различными типами человекоподобных роботов, чтобы пользователь мог выбирать наиболее подходящую модель для своих задач.
* Окно настройки робота должно содержать следующие элементы интерфейса:

1. Названия сервоприводов в виде поля ввода, их количество равно количеству сервоприводов в выбранной конфигурации
2. Кнопки “Min”, “Max”, “Res” рядом с названиями сервоприводов.
3. Слайдеры и цифровые значения положения слайдеров справа от них.
4. Кнопки “A”(циклическая отправка массивов значений сервоприводов), “1”(отправка выделенной строки в поле сохраненных положений сервоприводов), “Up”(ставит выделенную строку на одну позицию выше), “Down”(ставит выделенную строку на одну позицию ниже).
5. Поле сохранения положений робота, где отображаются записанные позиции робота, взятые со слайдеров и/или из файла положений.
6. Кнопки “Загрузить”(загрузка файла с массивами значений сервоприводов), “ Сохранить”(сохранение записанных позиций робота), “Записать”(запись положений сервоприводов, которые выставлены на слайдерах), “Удалить”(удаляет из записанных выделенную строку с позициями сервоприводов), “Очистить”(Удалить все записанные позиции сервоприводов).
7. Выпадающий список, в котором должны отображаться COM-порты, на которых подключен робот, поля ввода для ограничения максимального и минимального значения сервоприводов, кнопка открытия передачи в COM-порт(если подключение не установлено – красная, если удалось подключиться к роботу – зеленая).

* При нажатии кнопки “Min” у соответсвующего ей слайдера левая граница становится равной текущему значению, ползунок слайдера переносится влево. При нажатии кнопки “Max” у соответсвующего ей слайдера правая граница становится равной текущему значению, ползунок слайдера переносится вправо. При нажатии кнопки “Res” у соответсвующего ей слайдера обе границы вернутся к начальному состоянию.
* При нажатии кнопки “1” на робота отправляется выбранный двойным кликом массив значений положений сервоприводов через COM-порт.
* При нажатии кнопки “А” на робота отправляются друг за другом массивы значений положений сервоприводов через COM-порт, с задержкой, необходимой для того, чтобы робот успел перейти из одной позиции в другую.
* При нажатии кнопки “Загрузить” открывается окно проводника windows, в котором пользователь может открыть файл с массивом значений положений сервоприводов. Значения из файла устанавливаются в поле сохранённых положений робота. Слайдеры устанавливаются в положения, указанные в первой строке массива.
* При нажатии кнопки “Сохранить” открывается окно проводника windows, в котором пользователь может выбрать файл, в который будут записаны массивы значений положений сервоприводов, созданные пользователем с помощью кнопки “ Записать”.

Показатели качества программного продукта отображены в таблице 1.1

Таблица 1.1

“Показатели качества ПП для каждого уровня тестирования”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Характеристика** | **Атрибут** | | **Метрика** | **Описание метрики** | **Вес метрики** |
| **Уровни тестирования бизнес требований** | | | | | |
| Завершенность | | Область выбора модели робота. | Наличие основных кнопок выбора модели. | Присутствуют кнопки “6 сервоприводов”, “8 сервоприводов”. | 0.5 |
| Область выбора иной модели робота. | Наличие поля ввода и кнопки выбора для выбора иной модели. | Присутствуют поле ввода количества сервоприводов,  кнопка выбора модели на указанное в поле ввода количество сервоприводов. | 0.5 |
|  | |  |  | Итог: | 1 |
| Недвусмыслен  ность | | Область работы с сохраненными положениями робота | Название элементов управления  соответствует их функциям | При нажатии на кнопку “1” отправляется одно выбранное  положение робота, на “A” – друг за другом все записанные и т.д. | 0.5 |
| **Характеристика** | | **Атрибут** | **Метрика** | **Описание метрики** | **Вес метрики** |
|  | | Область работы с файлами | Всплывающие окна с уведомлениями об ошибках отражают суть ошибки. | При попытке открыть файл с неподходящим содержанием выводится сообщение ”Открыт неверный файл” | 0.5 |
|  | |  |  | Итог: | 1 |
| **За характеристику: 0,5** | | | | | |
| **Уровни тестирования пользовательских требований** | | | | | |
| Завершенность | | Область управления сервопривода  ми | Удобство интерфейса | Интерфейс программы интуитивен и легко читаем | 0.5 |
| Корректность | | Область работы с файлами | Реализация открытия файла с массивами положений робота. | При открытии файла в область записанных значений вводятся только валидные значения. | 0.5 |
|  | |  |  | Итог: | 1 |
| **За характеристику: 0,5** | | | | | |
| **Уровни тестирования продуктных требований** | | | | | |
| Завершенность | | Область выбора модели робота | Работоспособ  ность кнопки “6 сервоприво  дов” | При нажатии на кнопку “6 сервоприводов” открывается  основное окно, в котором 6 слайдеров. | 0.2 |
| **Характеристика** | | **Атрибут** | **Метрика** | **Описание метрики** | **Вес метрики** |
| Завершенность | |  | Работоспособ  ность кнопки “8 сервоприво  дов” | При нажатии на кнопку “8 сервоприводов” открывается основное окно, в котором 8 слайдеров. | 0.2 |
| Выполнимость | | Область выбора пользовательской модели | Валидация поля “количество сервоприво  дов” | Для ввода допускаются числа от 1 до 20 включительно | 0.2 |
| Выполнимость | | Область управления сервопривода  ми | Валидация полей ввода значений сервоприводов | Для ввода допускаются числа от 600 до 2400 | 0.2 |
| Завершенность | | Область работы с файлами | Работоспособность кнопки “Save” | При нажатии на кнопку открывается окно проводника windows, в котором можно выбрать, в какой файл сохранить записанные значения | 0.2 |
|  | |  |  | Итог: | 1 |
| **За характеристику: 0,2** | | | | | |

# Требования к надежности

* Программный комплекс должен быть надежным и безопасным для использования, чтобы избежать возможных нештатных ситуаций при работе робота.
* Программный комплекс должен иметь защиту от ввода неверных значений вручную пользователем или при открытии файлов.

# Условия эксплуатации

* На объектах автоматизации должны отсутствовать такие воздействия, как: механический резонанс, механические удары, рабочие растворы и агрессивные среды. Диапазон рабочих температур от +15°С до +45°С. Относительная влажность до 80% при температуре +25°С. Запыленность до 0,4 г/м3.

# Требования к составу и параметрам технических средств

* Микроконтроллер Arduino: необходимо использовать микроконтроллер Arduino для управления роботом, так как он обеспечивает широкие возможности для программирования.
* Силовой модуль управления сервоприводами: Должен быть совместим с Arduino и иметь возможность для подключения 6 и более сервоприводов.
* Сервоприводы: для передвижения робота необходимо использовать сервоприводы, которые обеспечивают достаточную мощность и скорость движения. Количество сервоприводов должно быть от 6 до 20 включительно.
* Аккумулятор: для обеспечения автономной работы робота необходимо использовать аккумулятор, который обеспечивает достаточную емкость и напряжение для питания всех компонентов.
* Кабель: USB A-USB B с разъемами, тип “папа-папа”, длиной более 1 метра для подключения к роботу.

# Требования к информационной и программной совместимости

* Система должна быть совместима с операционной системой Windows 7, Windows 8, Windows 10, Windows 11. В системе должно быть ОЗУ не менее 4 Гб и дисковое пространство не менее 64 Гб, в том числе не менее 500 Мб свободного места для работы программы.

# Технические решения проекта ПП

# Описание информационной базы данных:

База данных должна быть представлена в виде набора файлов позиций сервоприводов в формате “.txt”. В файлах должны находиться массивы позиций сервоприводов через запятую, и если массивов несколько, то они переносятся на следующую строку. Программа должна поставляться с набором готовых стандартных файлов позиций сервоприводов для 2 основных моделей роботов – на 6 и на 8 сервоприводов. Должна быть возможность создания файлов позиций сервоприводов из программного комплекса пользователем.

Система программного обеспечения:

* Код для платформы Arduino - программа, которая загружается на платформу Arduino и управляет работой робота. Этот код должен быть написан на языке программирования C++, использовать библиотеки для работы с моторами, датчиками и коммуникационными модулями.
* Операционная система на компьютере пользователя - Windows 7, Windows 8, Windows 10, Windows 11. В системе должно быть ОЗУ не менее 4 Гб и дисковое пространство не менее 64 Гб, в том числе не менее 500 Мб свободного места для работы программы.

# Планирование тестирования программного продукта

Определение количества ожидаемых ошибок

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Объект тестирования** | **Метрика** | **Ожидаемые ошибки, вид** | **Виды и методы тестирования** | **Тестовые данные** | **Тесто**  **вый доку**  **мент** |
| **Уровень тестирования** **бизнес требований** | | | | | | |
| **Надежность: обязательность** | | | | | | |
| 1 | Старто  вое окно программ  мы | Понятность меню выбора модели робота | Опечатки в названии элемента меню | Системное тестирование, Тестирование данных,  Динамическое тестирование | Меню выбора модели робота, состоящее из 2 кнопок | Высо  коуров  невый тест-кейс |
| 2 | Изображение соответствует моделям робота | Изображения не отобража  ются на кнопках | Системное тестирование, Тестирование данных  Демонстрационное тестирование  Интеграционное тестирование  Динамическое тестирование | Меню выбора модели робота, состоящее из 2 кнопок | Высо  коуров  невый тест-кейс |
| 3 | Поле ввода количества сервоприводов подписано | Текст не отображается | Системное тестирование, Тестирование данных,  Динамическое тестирование | Меню выбора собственной модели робота | Высо  коуров  невый тест-кейс |
| 4 | Основное окно программ  мы | Все кнопки имеют надписи, говорящие об их функциях | Текст не отображается | Системное тестирование | Функцио  нальные кнопки в основном окне | Высо  коуров  невый тест-кейс |
| **Уровень тестирования** **пользовательских требований** | | | | | | |
| **Функциональность: Горизонтальная прослеживаемость** | | | | | | |
| 1 | Основное окно программ  мы | Цветовая схема соответствует техническому заданию | Цветовая гамма отличается от той, что указана в техничес  ком задании | Статическое тестирование, Демонстрационное тестирование, Тестирование данных  Системное тестирование | Техническое задание, макеты | Чек-лист |
| 2 | Надписи выполнены с соблюдением правил типографики | Текст выглядит нечитабельно | Динамическое тестирование Демонстрационное тестирование, Тестирование | Техническое задание, макеты | Чек-лист |
| **№** | **Метрика** | **Ожидаемые ошибки, вид** | **Виды и методы тестирования** | **Тестовые данные** | **Тестовый документ** |
|  |  |  | данных  Системное тестирование |  |  |
| 3 | Присутствуют элементы управления сервоприводами | Отсутству  ет один или несколько слайдеров | Динамическое тестирование,  Тестирование данных | Слайдеры для управления сервоприво  дами | Чек-лист |
| 4 | Есть кнопки “Min”, “Max”, “Def”, “Res” | При нажатии на кнопки ничего не происходит | Динамическое тестирование Демонстрационное тестирование, Тестирование данных | Кнопки для настройки диапазона работы сервоприводов | Чек-лист |
| **Уровень тестирования** **продуктных требований** | | | | | | |
| **Функциональность: Выполнимость** | | | | | | |
| 1 | Старто  вое окно программ  мы | Кнопка выбора собственной модели робота | Кнопка доступна до того, как пользователь введет количество сервоприводов в поле ввода | Динамическое тестирование Демонстрацион  ное тестирование, Тестирование данных | Поле ввода количества сервоприво  дов | Чек-лист |
| 2 | Основное окно программ  мы | Кнопка загрузки моделей движения робота | Программа не вылетает при загрузке неверного файла | Динамическое тестирование Демонстрацион  ное тестирование, Тестирование данных | Кнопка загрузки моделей движения робота ”Load” | Чек-лист |
| 3 | Основное окно программ  мы | Кнопка отправки на робота 1 позы | Программа отправляет массив положений сервоприводов, хотя пользова  тель не выбрал его из списка | Динамическое тестирование Демонстрацион  ное тестирование, Тестирование данных | Кнопка отправки на робота 1 позиции ”1” | Чек-лист |

## РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

## Разработка модуля передачи данных с ПК на Arduino

## Описание алгоритма

## Назначение и характеристика алгоритма

Назначение модуля передачи данных с ПК на Arduino с помощью протокола заключается в обеспечении надежной и эффективной передачи информации между компьютером и микроконтроллером Arduino. Это позволяет создавать различные проекты, в которых необходимо управлять платформой Arduino, используя данные, полученные с компьютера.

Характеристики алгоритма передачи данных с ПК на Arduino с помощью протокола:

1. Простота использования. Для передачи данных не требуется специальных знаний или навыков. Достаточно знать основы программирования и понимать, как работает протокол передачи данных.

2. Надежность. Модуль обеспечивает надежную передачу данных между компьютером и микроконтроллером Arduino. Это позволяет избежать ошибок при выполнении задач и повысить эффективность работы устройства.

3. Универсальность. Модуль передачи данных подходит для работы с различными моделями роботов. Это позволяет использовать его как с роботами на 6 сервоприводов, так и с роботами на 20 сервоприводах.

В качестве входных данных модуль передачи данных с ПК на Arduino использует значения положений сервоприводов, которые указаны в программе. Модуль конвертирует эти значения в строковый тип и, в дальнейшем, отправляет на робота.

Результатом решения данного алгоритма является строка из положений сервоприводов с символом, означающим конец массива.

# Математическое описание алгоритма

Во время своей работы модуль отправки значений на робота отправляет друг за другом значения, которые на роботе будут сформированы в строку вида:

Str = {x1, x2, … xy-1, xy, @}, где xy – значение положения сервопривода в микросекундах, y – количество сервоприводов в выбранной модели робота. Символ “@” является последним символом в строке и означает ее конец. Этот символ необходим для того, чтобы робот разделял команды компьютера на отдельные, когда те отправляются непрерывно.

# Алгоритм

1. Открыть порт последовательной передачи данных. В случае неудачи вывести сообщение пользователю “Проверьте, что робот подключен к ПК” и завершить выполнение алгоритма.
2. В цикле от 0 до servoCount:
   1. Получить значение сервопривода по текущему индексу со слайдеров или из поля сохранённых положений робота.
   2. Конвертировать значения из типа string в тип int.
   3. Отправить в последовательный порт значение слайдера.
   4. Отправить в последовательный порт символ “,” (запятая).
3. Отправить в последовательный порт символ “@” (Собачка).
4. Приостановить программу, пока команда отправляется.
5. Если в течении 5 секунд микроконтроллер “Arduino” не отправит в ответ символ “#”(Решетка), то закрыть порт последовательной передачи данных и вывести сообщение “Проверьте, что робот подключен к ПК”.

# Разработка модуля настройки поворотов сервоприводов.

# Описание алгоритма

# Назначение и характеристика алгоритма

Назначение алгоритма настройки поворотов сервоприводов заключается в установке определенных углов поворота для каждого из сервоприводов, используемых в системе. Это позволяет точно контролировать движение механизмов, которые управляются данными сервоприводами.

Характеристика алгоритма модуля настройки поворотов сервоприводов включает в себя следующие особенности:

1. Гибкость: алгоритм может быть настроен на работу с различным количеством сервоприводов, что позволяет использовать его с различными моделями роботов.

2. Простота использования: благодаря интуитивно понятному интерфейсу, алгоритм может быть легко настроен и использован даже без специальных знаний в области электроники.

3. Высокая точность: алгоритм позволяет установить точные углы поворота для каждого из сервоприводов, что обеспечивает плавное движение механизмов.

4. Надежность: благодаря тщательно протестированной работе алгоритма, он обеспечивает надежную работу системы управления сервоприводами.

# Алгоритм

Кнопки “Min”, “Max”, “Res”:

* 1. Получить значение со слайдеров.
  2. Отобразить значения в поле значения положения слайдера. Если была нажата кнопка Min или Max, то установить соответственно минимальную или максимальную границу на текущем значении слайдера. Если нажата кнопка Res, то значения границ сбрасываются.
  3. При нажатии кнопки “Установить границы”, берутся значения, указанные в полях “Мин. значение” и “Макс. значение” и устанавливаются в качестве максимального и минимального значения, в которые можно установить слайдер

Кнопка “Записать”:

1. Получить количество строк в поле сохранённых положений робота.
2. В новую нижнюю строку, в каждую колонку по количеству сервоприводов установить значение со соответствующего слайдера.
3. Изменить размеры колонок для того, чтобы все значения сервоприводов вмещались полностью.

Кнопка “Перезаписать”:

1. Если поле сохраненных положений робота не пустое и в нем выделена одна из строк, то в цикле от 0 до servoCount:
   1. Получить номер выделенной строки.
   2. Получить значения, установленные на слайдерах по текущему индексу.
   3. Записать это значение в ячейку выделенной строки.

# Описание контрольного примера

Указание минимального значения слайдера и сброс до первоначального значения.

Необходимо нажать кнопку “MIN”, чтобы установить левую границы для слайдера.

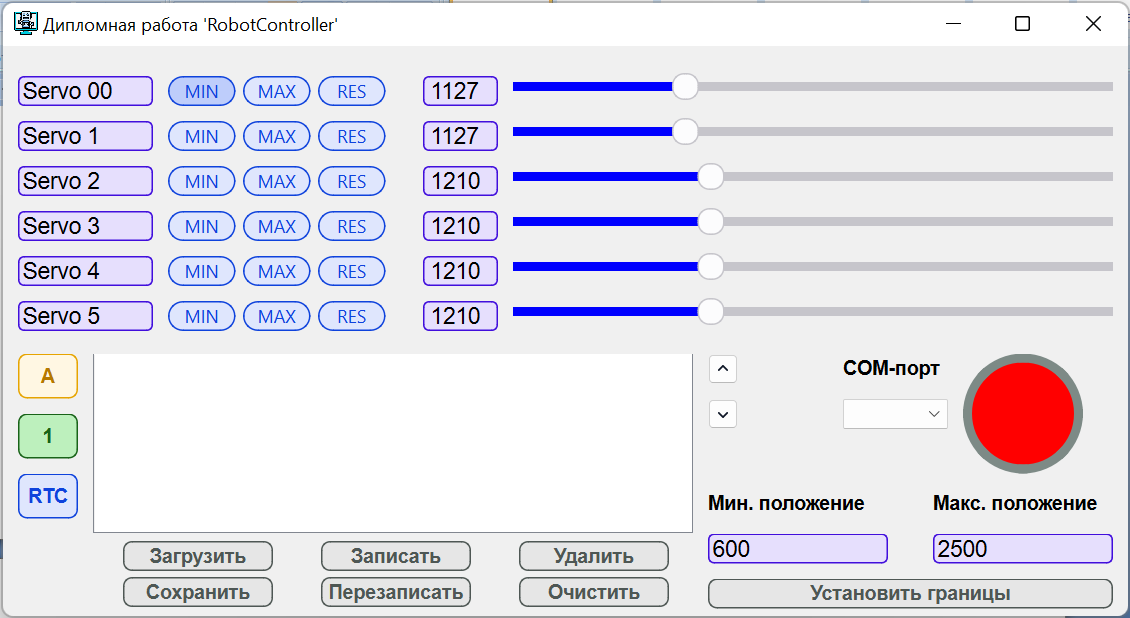


Рис.1.1 Пользователь нажимает кнопку “MIN”

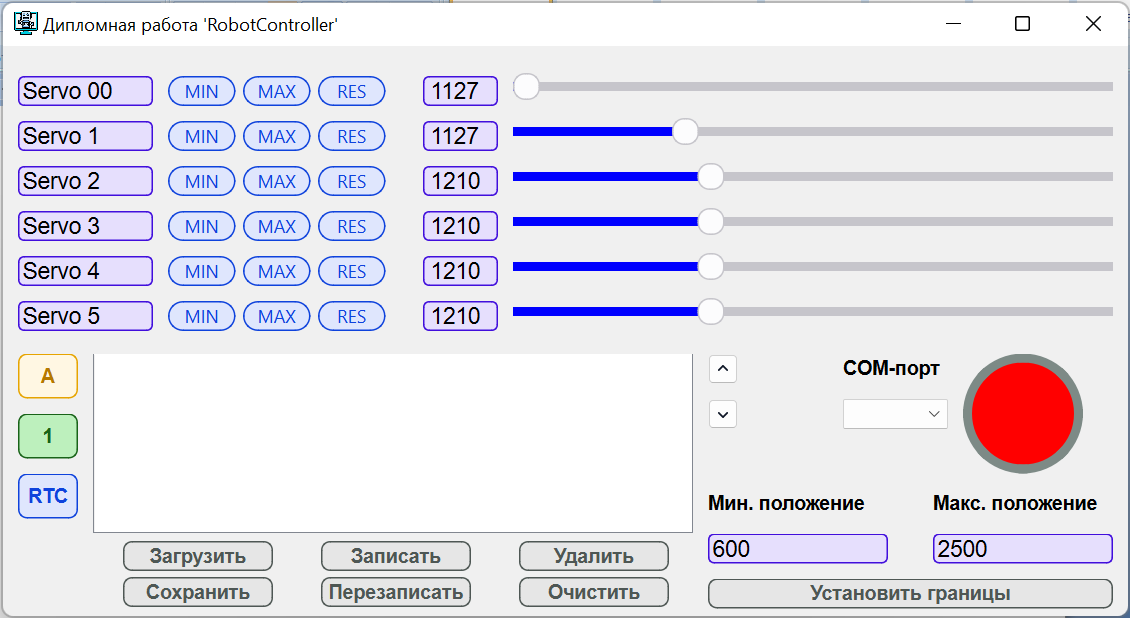


Рис.1.2 Программа установила для данного слайдера минимальное значение

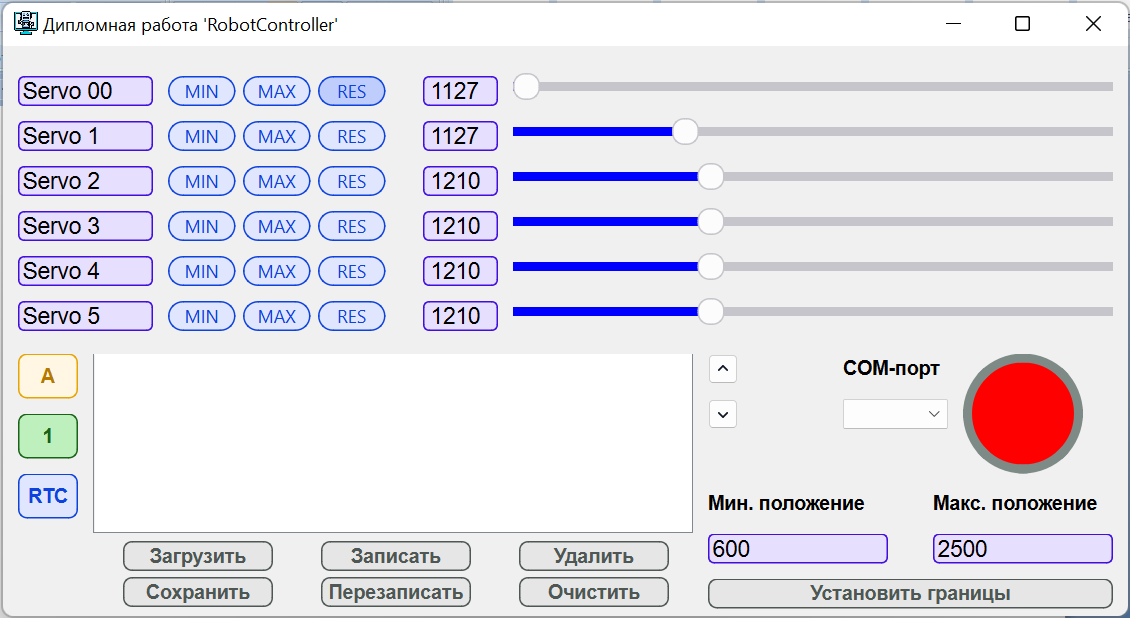
При нажатии кнопки “RES”, границы слайдера сбрасываются до первоначальных

Рис.1.3 Пользователь нажимает кнопку “RES”

При нажатии на кнопку “Установить границы”, для всех слайдеров устанавливаются максимальное и минимальное значения, указанные пользователем.

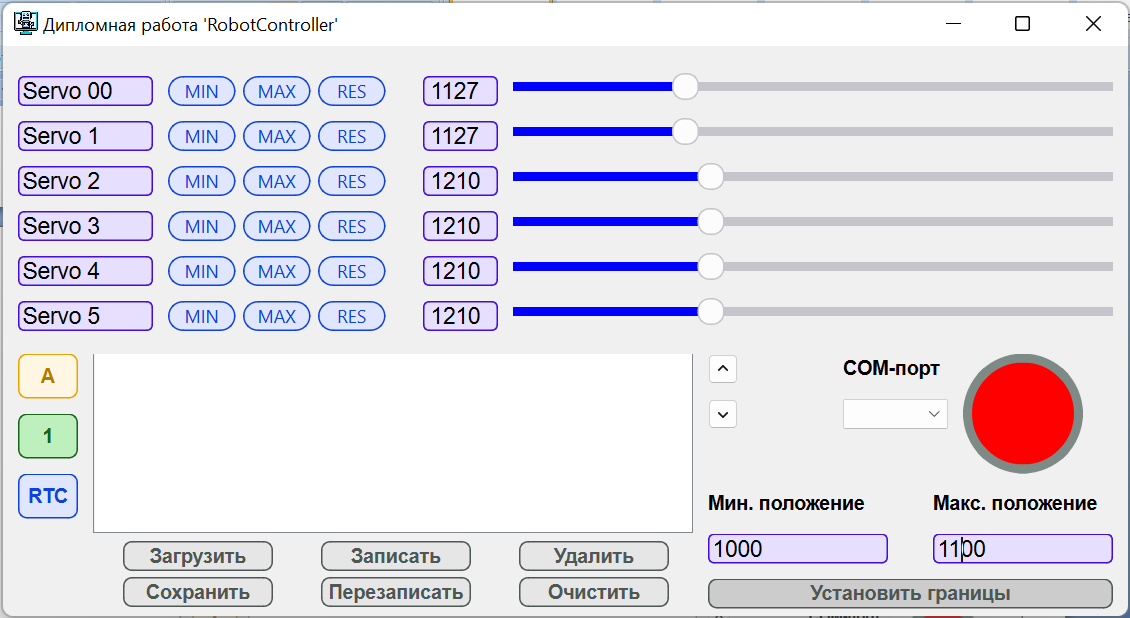


Рис.1.4 Пользователь нажимает кнопку “Установить границы”

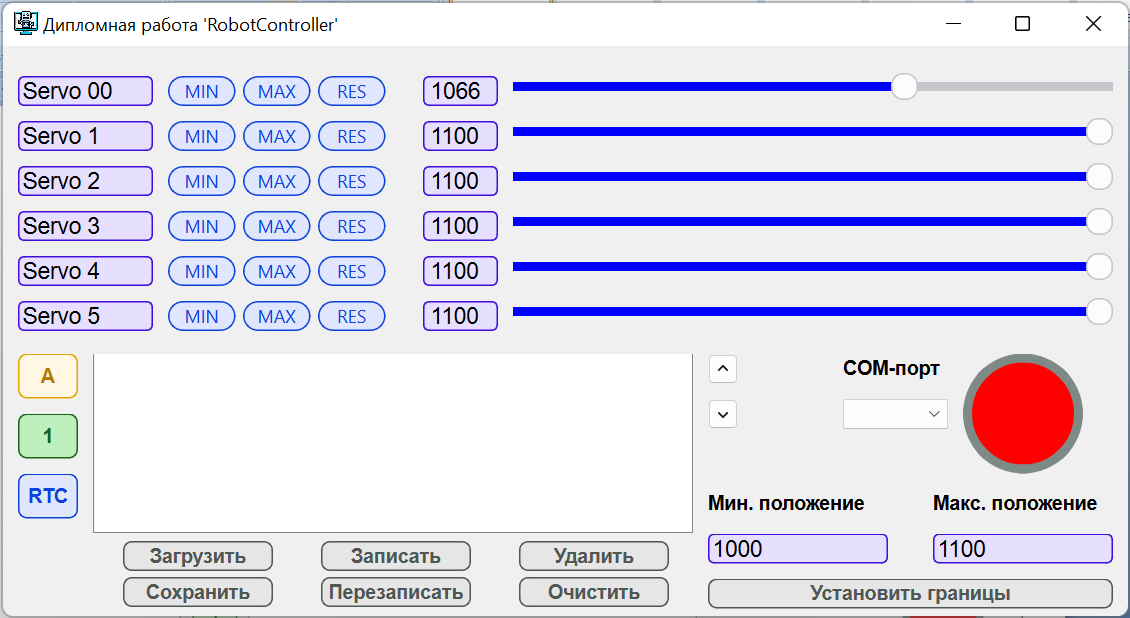


Рис.1.5 Программа установила левую и правую границы слайдеров

# Разработка модуля работы с файлами, хранящими позиции сервоприводов.

# Описание алгоритма

# Назначение и характеристика алгоритма

Назначение алгоритма модуля работы с файлами, хранящими позиции сервоприводов, заключается в сохранении и загрузке установленных углов поворота для каждого из сервоприводов в файлы на жестком диске компьютера. Это позволяет сохранять настройки сервоприводов между сеансами работы системы, а также делиться настройками с другими пользователями.

Характеристика алгоритма модуля работы с файлами, хранящими позиции сервоприводов, включает в себя следующие особенности:

1. Удобство: благодаря возможности сохранения и загрузки настроек сервоприводов в файлы, пользователь может легко переключаться между различными настройками и делиться ими с другими пользователями.

2. Надежность: алгоритм обеспечивает надежное сохранение и загрузку настроек сервоприводов, что позволяет избежать потери настроек при сбоях в работе системы.

3. Гибкость: алгоритм может работать с различным количеством сервоприводов, что позволяет использовать его в различных проектах.

# Алгоритм

Чтение из файла:

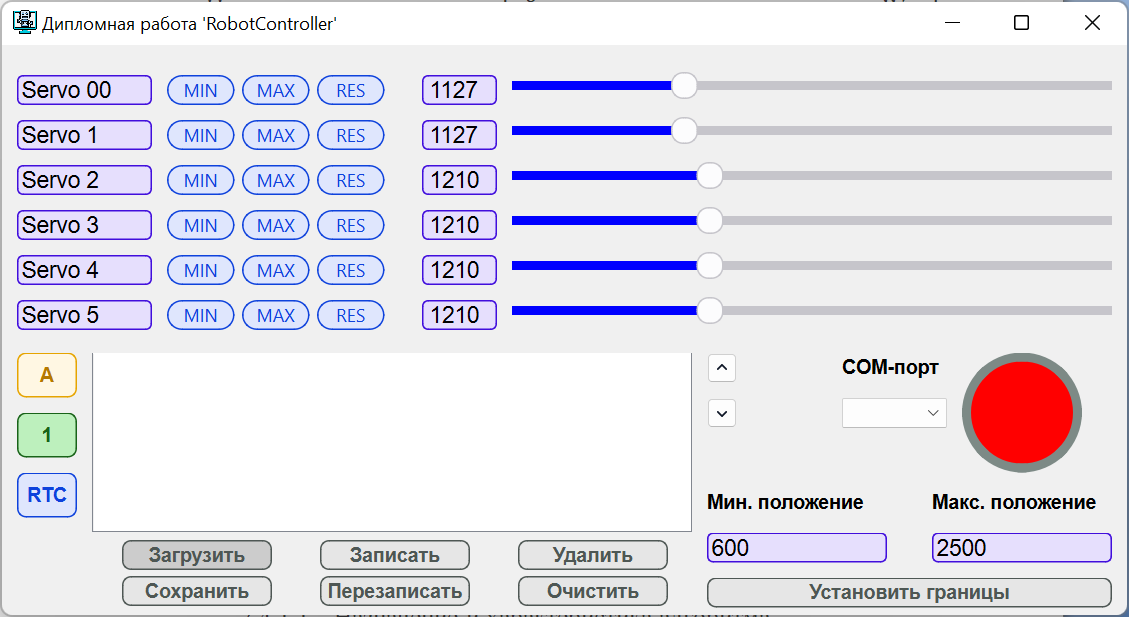
1. Выбрать файл через окно проводника Windows.
2. Открыть файл в режиме чтения.
3. Построчно прочитать содержимое файла, проверяя на соответствие шаблону строки с положением робота. Иначе вывести ошибку “Неверное содержание файла. В файле указаны неверные положения сервоприводов”.
4. Если файл имеет правильное содержание, то необходимо очистить поле сохраненных положений робота.
5. Прочитать файл построчно, записывая значения каждой строки в отдельные строки поля сохраненных положений робота.
6. Установить на слайдерах значения, соответствующие значениям, записанным в первой строке файла.
7. Закрыть файл.

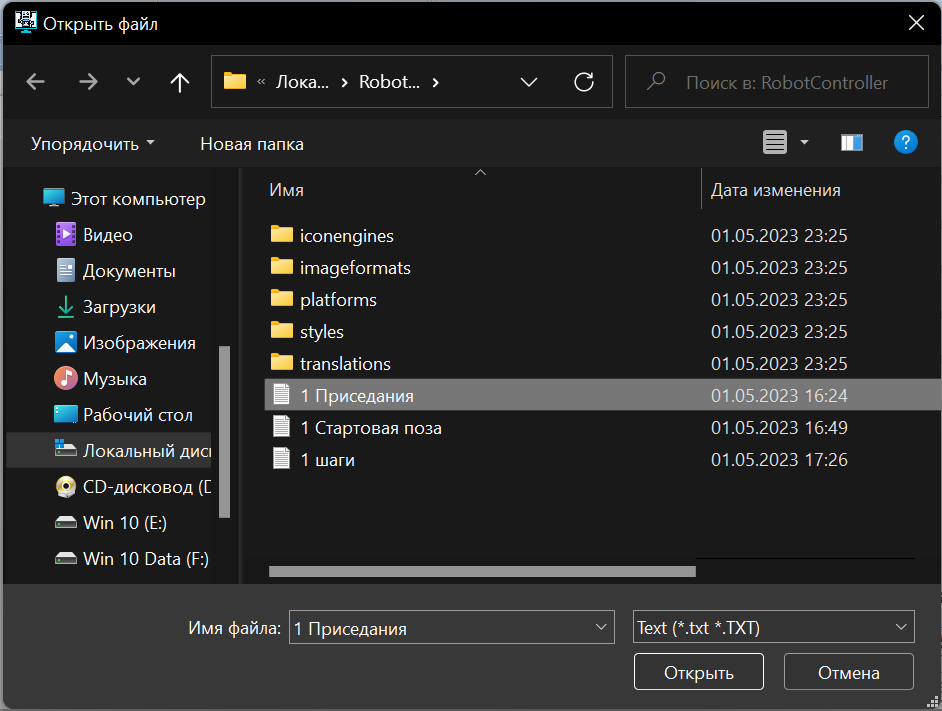
Запись в файл:

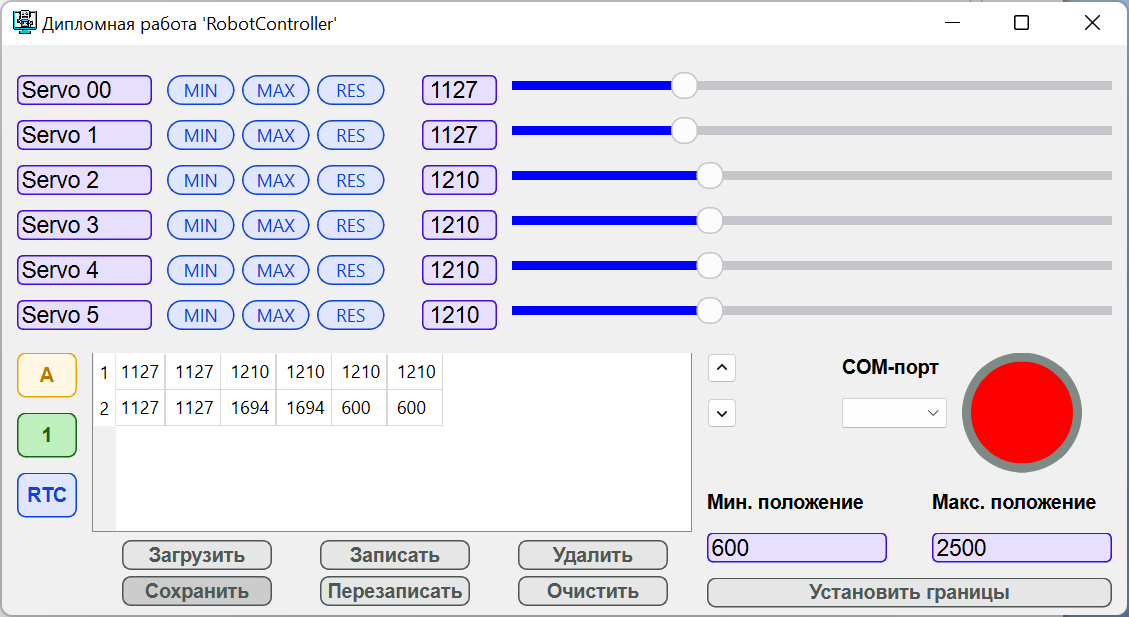
1. Выбрать файл через окно проводника Windows.
2. Открыть файл в режиме чтения, иначе вывести ошибку “Файл не удалось открыть”.
3. Построчно записать значения из поля сохранённых положений в выбранный файл.

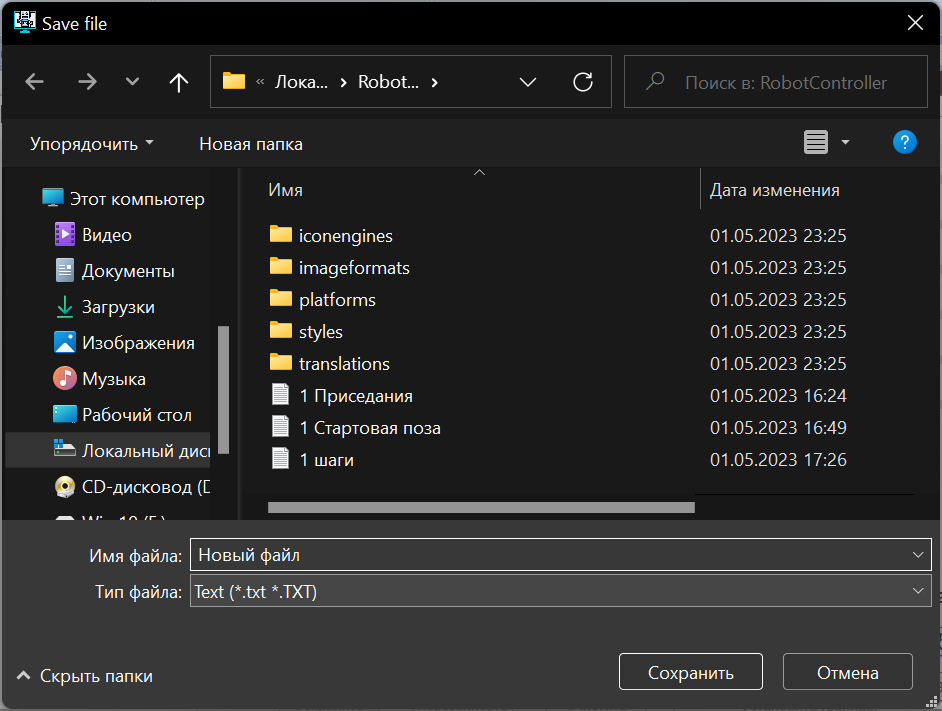
# Описание контрольного примера

Нажав на кнопку загрузки файла, откроется окно проводника Windows, в котором можно выбрать один текстовый файл. Когда программа отобразит содержимое файла, его можно записать в другой файл.









# Разработка программного кода для управления роботом на платформе Arduino.

# Описание алгоритма

# Назначение и характеристика алгоритма

Программа робота на Arduino предназначена для управления действиями робота, который использует платформу Arduino. Она позволяет управлять сервоприводами, обрабатывать данные, полученные через COM-порт и принимать решения на основе полученных данных.

Характеристики программы робота на Arduino включают:

1. Простоту использования: программа может быть написана на языке программирования C++, который легко понять и использовать даже для начинающих разработчиков.

2. Гибкость: программа может быть адаптирована для различных типов роботов.

3. Надежность: программа должна быть стабильной и надежной, чтобы робот мог выполнять свои задачи без сбоев.

4. Совместимость: программа должна быть совместима с платформой Arduino и другими компонентами, которые используются в роботе.

5. Эффективность: программа должна быть эффективной и быстрой, чтобы робот мог быстро изменять свое положение.

# Математическое описание алгоритма

При установке сервопривода в необходимое положение используется функция map(), которая выглядит так:

(x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min

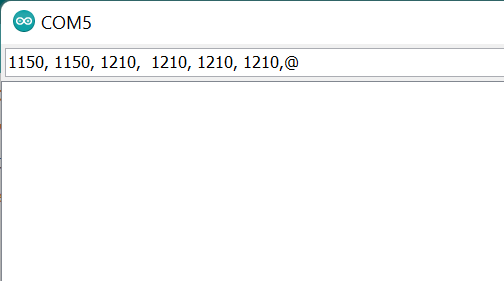
in\_min – минимальное значение текущего диапазона, in\_max - максимальное значение текущего диапазона, out\_min - минимальное значение нового диапазона, out\_max - максимальное значение нового диапазона, i – текущее значение.

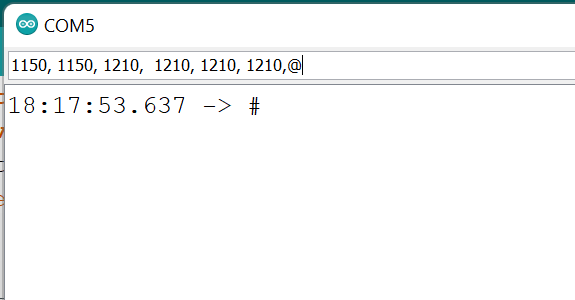
# Алгоритм

1. Один раз, при запуске робота, выставить все сервоприводы в стартовую позицию.
2. В бесконечном цикле проверять, пришли ли какие-либо значения в com-порт.
3. Записать во временную строку полученные значения.
4. Разбить временную строку на массив чисел.
5. Проверить все значения из массива, что каждое больше 540 и меньше 2400.
6. Записать значения, которые прошли проверку в другой массив, соответствующий следующему положению робота.
7. Отправить в COM-порт символ “#”(решётка).
8. Очистить временную строку.
9. Плавно поменять положение робота с текущей на необходимую:
   1. В цикле от 0 до servoCount найти максимальную разницу положений сервоприводов между текущим положением и необходимым положением, записать в переменную maxDelta.
   2. Разделить значение переменной maxDelta на 10 чтобы определить количество шагов, которые должен сделать робот, чтобы добраться до новой позиции.
   3. В цикле от 0 до maxDelta:
      1. В цикле от 0 до servoCount:
         1. Установить сервопривод по индексу servoCount в положение, которое равняется текущему плюс maxDelta. Функция map() используется для преобразования текущего шага в соответствующую позицию сервопривода.
         2. Приостановить цикл, чтобы сервопривод успел перейти в необходимое положение.
10. Записать в массив с текущим положением сервоприводов новые значения.

# Описание контрольного примера

В качестве контрольного примера отправляется строка с набором позиций сервоприводов. Микроконтроллер обрабатывает полученную строку и возвращает ответ в виде символа “решетка”(#).





# СПИСОК ЛИТЕРАТУРЫ

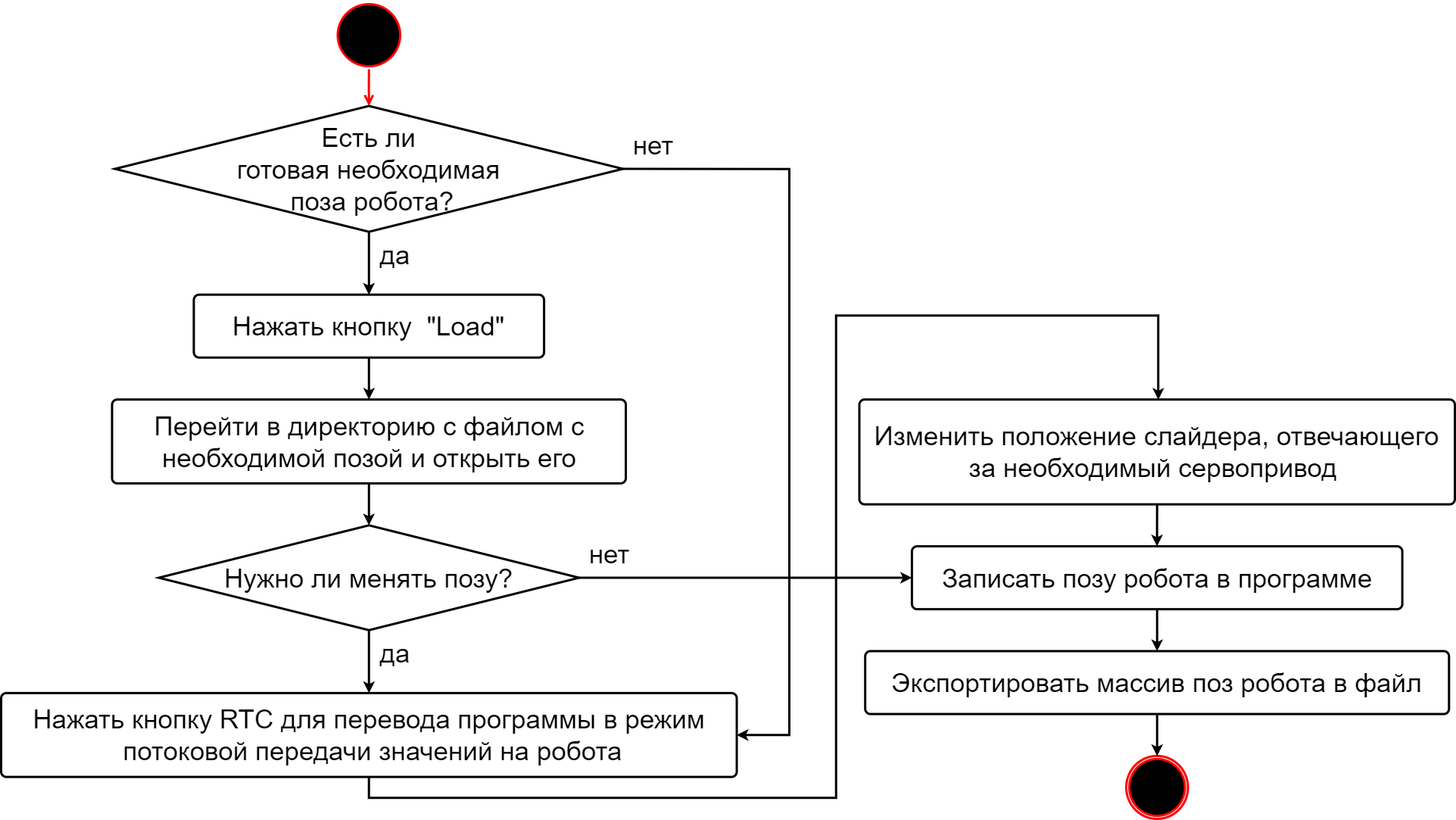
1. Robo+ Motion user manual. : сайт. – URL: https://emanual.robotis.com/docs/en/software/rplus1/motion/ (дата обращения: 13.06.2023)
2. Robo+ Motion 2.0 user manual. : сайт. – URL: https://robotis.co.uk/software/roboplus-2-0.html (дата обращения: 13.06.2023)
3. RoboSoul H5S : сайт. – URL: https://www.hiwonder.com.cn/store/learn/35.html (дата обращения: 13.06.2023)
4. Справочник языка Arduino на русском языке. : сайт. – URL: https://arduino.ru/Reference (дата обращения: 13.06.2023)
5. Монитор порта, отладка – Alex Gyver. : сайт. – URL: https://alexgyver.ru/lessons/serial/ (дата обращения: 13.06.2023)

# ПРИЛОЖЕНИЕ А

Диаграмма использования



Диаграмма деятельности программы для компьютера



Структурная схема

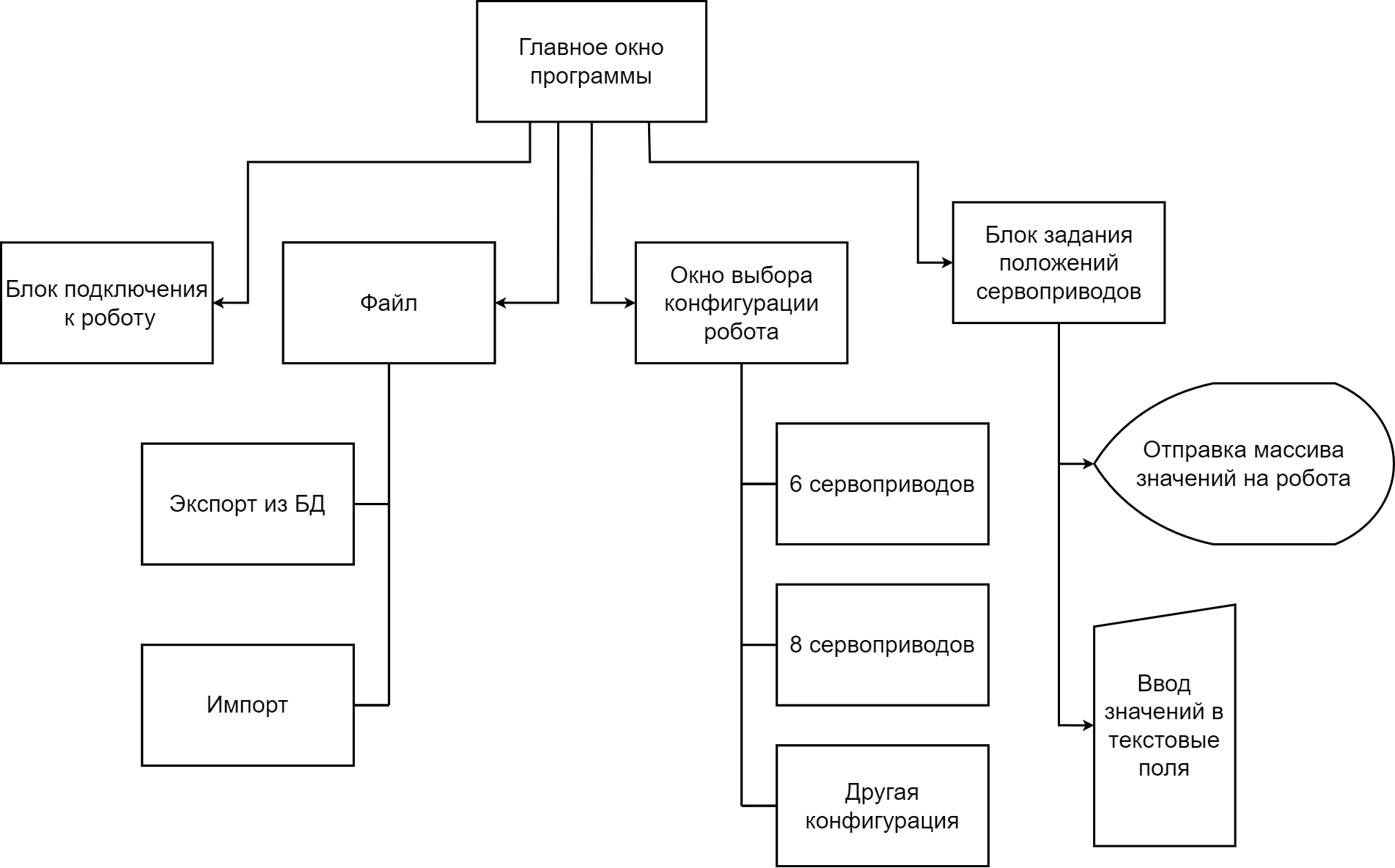


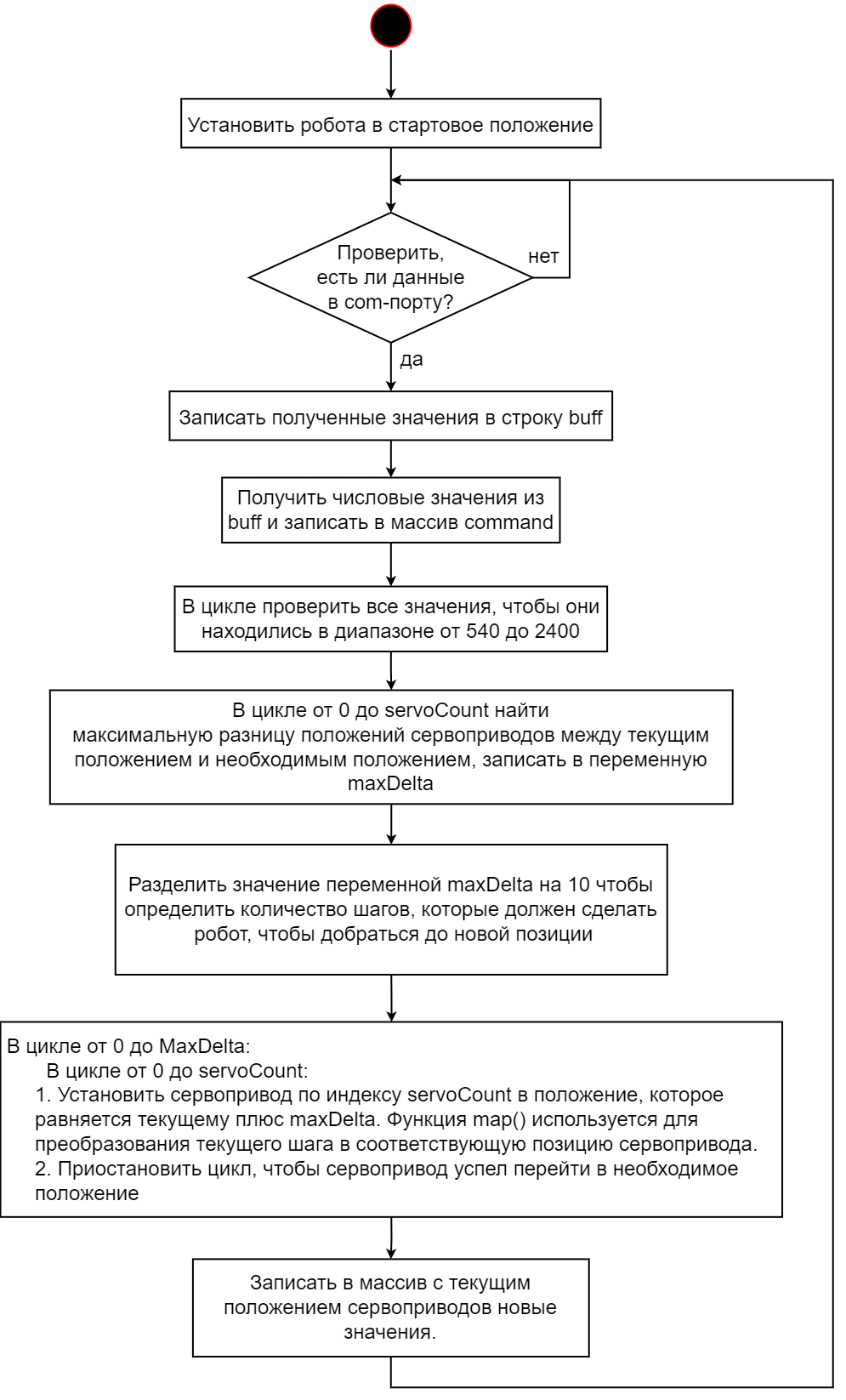
Диаграмма последовательности



Диаграмма компонентов



Диаграмма деятельности программы микроконтроллера Arduino



# ПРИЛОЖЕНИЕ Б

Текст программы для микроконтроллера Arduino:

#include <SoftwareSerial.h>

#include <Servo.h>

#include <Multiservo.h> //Подключение библиотеки multiservo

#include <Wire.h> //Подключение библиотеки Wire

String str;//="1150, 1150, 1210, 1210, 1210, 1210,@ ";

int ServCount = 6;//Константа хранит количество сервоприводов

Multiservo Servos[6];//Массив сервоприводов

int NumControl[] = {1, 2, 3, 5, 6, 7};

int MinTiming = 540;//Минимальная длительность импульса

int MaxTiming = 2400;//Максимальная длительность импульса

int APos[6];//Массив хранит предыдущие координаты сервоприводов для

//расчета дискретных шагов приращения для перехода на заданную позицию

int BPos[6];//Массив для передачи параметров новой позиции

int StartPos[] = {1150, 1150, 1210, 1210, 1210, 1210};

int NextPos[] = {1150, 1150, 1210, 1210, 1210, 1210};

int command[ServCount+1];

void MovePos(int xPos[]){

int Delta=0;

int MaxDelta=0;

for (int i=0; i<ServCount; i++){

Delta=abs(APos[i]-xPos[i]);

if (MaxDelta < Delta) MaxDelta = Delta;

}

MaxDelta/=10;

for (int i=0; i<MaxDelta; i++)

for (int j=0; j<ServCount; j++){

Servos[j].writeMicroseconds(map(i,0,MaxDelta,APos[j],xPos[j]));

delayMicroseconds(5000);//влияет на скорость выполнения

}

for (int i=0; i<ServCount; i++)

APos[i]=xPos[i];

}

void setup() {

Serial.begin(9600);

Serial.setTimeout(30);

for (int i=0; i<ServCount; i++){

Servos[i].attach(NumControl[i],MinTiming,MaxTiming);

}

MovePos(StartPos);

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = {0, -1};

int maxIndex = data.length()-1;

for(int i=0; i<=maxIndex && found<=index; i++){

if(data.charAt(i)==separator || i==maxIndex){

found++;

strIndex[0] = strIndex[1]+1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found>index ? data.substring(strIndex[0], strIndex[1]) : "";

}

void loop() {

while (Serial.available()){

str=Serial.readString();

String buff;

for(int i=0; i<ServCount+1; i++){

buff = getValue(str, ',', i);

command[i] = buff.toInt();

}

for(int i=0; i<ServCount; i++){

if(command[i]>540 && command[i]<2400)

NextPos[i] = command[i];

//Servos[i].writeMicroseconds(command[i]);

}

Serial.write("#");

str="";

buff ="";

MovePos(NextPos);

}

}