Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Учебная практика (ознакомительная)

ОТЧЕТ по учебной практике (ознакомительной)

Выполнила

студентка: гр. 353504 Антонова Л.С.

Руководитель практики: Владымцев В.Д.

Минск 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc167283622)

[1.1 Исходная постановка задачи 3](#_Toc167283623)

[1.2 Описание средств разработки 3](#_Toc167283624)

[1.2.1 arduino ide 3](#_Toc167283625)

[1.2.2 qt creator 3](#_Toc167283626)

[1.3 Спецификация функциональных требований 4](#_Toc167283627)

[2 Проектирование и разработка программного средства 5](#_Toc167283628)

[2.1 Структура системы 5](#_Toc167283629)

[2.1.1 Механическая рука 5](#_Toc167283630)

[2.1.2 Графический редактор 5](#_Toc167283631)

[2.1.3 Взаимодействие компонентов 5](#_Toc167283632)

[2.2 Сборка робота 6](#_Toc167283633)

[2.3 Разработка программного обеспечения на ардуино для управления автоматизированной рукой с компьютера 7](#_Toc167283634)

[2.3.1 Библиотеки и глобальные переменные 7](#_Toc167283635)

[2.4 Графический редактор 9](#_Toc167283636)

[3 Тестирование и проверка работоспособности программного средства 11](#_Toc167283637)

[Список использованных источников 12](#_Toc167283638)

[Приложение А Исходный код программы 13](#_Toc167283639)

# **1 ПОСТАНОВКА ЗАДАЧИ**

## **1.1 Исходная постановка задачи**

Название данного проекта «Robot DrawMan».

Основные цели этого проекта включают разработку системы, состоящей из механической руки, которая держит маркер, и графического редактора на компьютере. После того как пользователь нарисует простые графические фигуры (такие как прямые линии, квадраты, ромбы, прямоугольники и треугольники) в графическом редакторе, автоматизированная рука должна воспроизвести рисунок на листе бумаги. Проект можно просмотреть, перейдя по ссылке [1] на GitHub.

## **1.2 Описание средств разработки**

### **1.2.1 Arduino IDE**

Для программирования Arduino используется интегрированная среда разработки Arduino (Arduino Integrated Development Environment, IDE) [2].

Arduino IDE - это интегрированная среда разработки, используемая для программирования Arduino. Это кросс-платформенное приложение на Java, которое включает в себя редактор кода, компилятор и модуль для загрузки прошивки на плату. Язык программирования, используемый для Arduino, очень похож на C++, с добавлением некоторых библиотек. Обработка программ осуществляется с помощью препроцессора, а компиляция - с помощью AVR-GCC.

Несмотря на отсутствие некоторых функций, таких как интеграция с репозиториями и автодополнение команд, Arduino IDE отмечается за простоту и удобство использования, особенно для новых пользователей.

### **1.2.2 Qt Creator**

Qt Creator [3] — это кроссплатформенная, полностью интегрированная среда разработки (IDE), которая может быть использована для создания приложений для различных платформ, включая Android и iOS. Он доступен для Linux, Windows и macOS. Проекты в Qt Creator могут быть выполнены на языках программирования C++ или Python.

Основная цель Qt Creator — упростить процесс создания графических приложений с использованием фреймворка Qt на различных платформах. Он предлагает множество функций для работы с интерфейсом, что, в сочетании с его высокой функциональностью, эффективностью и удобством настройки рабочего пространства, делает его идеальной средой разработки для данного проекта.

## **1.3 Спецификация функциональных требований**

Проект “Robot DrawMan” - это система, состоящая из механической руки, держащей маркер, и графического редактора на компьютере. Пользователь может рисовать простые геометрические фигуры в графическом редакторе, после чего автоматизированная рука воспроизводит рисунок на листе бумаги. Для обеспечения связи между графическим редактором и механической рукой была разработана система передачи данных. Механическая рука способна воспроизводить рисунки, созданные в графическом редакторе, обеспечивая высокую точность рисования. Интерфейс пользователя системы интуитивно понятен и позволяет легко рисовать геометрические фигуры и запускать процесс рисования автоматизированной рукой.

Ниже на рисунке 1.1 представлена механическая рука-манипулятор.

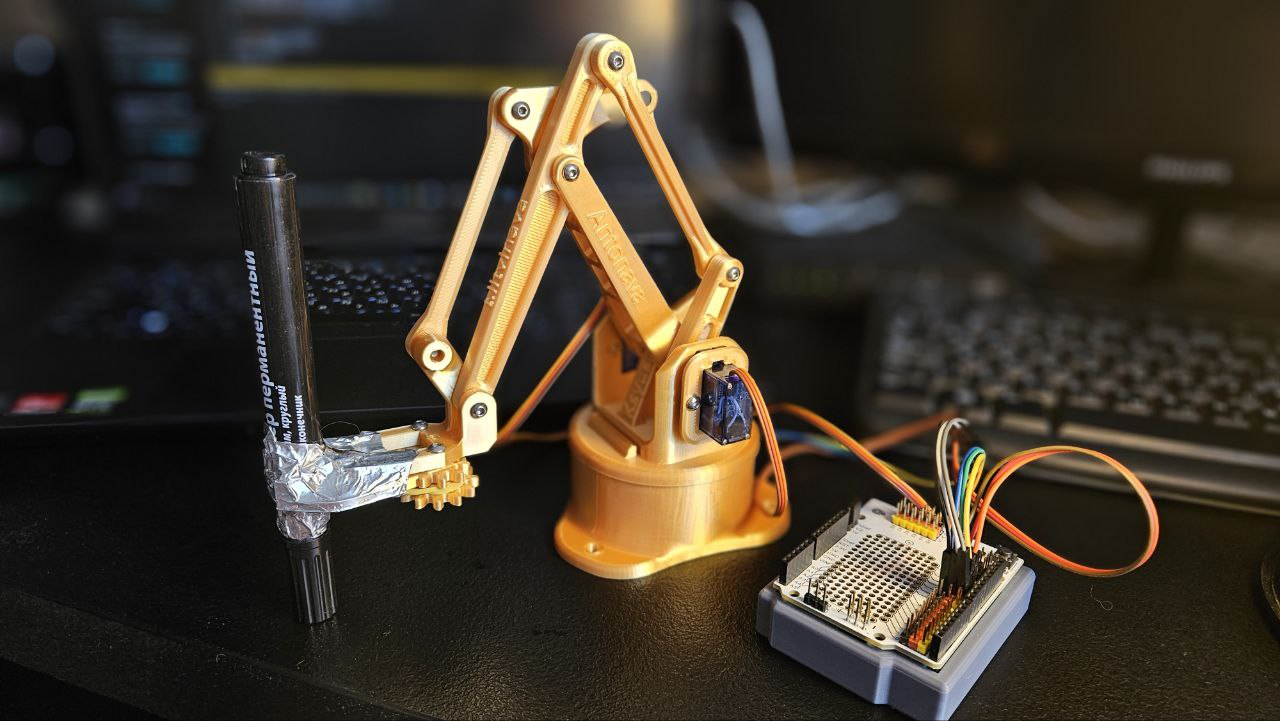


Рисунок 1.1 — Рука-манипулятор

# **2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА**

## **2.1 Структура системы**

Проект “Robot DrawMan” включает в себя два основных компонента: механическую руку и графический редактор, которые вместе образуют сложную, но эффективную систему. Эта система была разработана с учетом принципов модульности и гибкости, что позволяет легко модифицировать и адаптировать ее для различных задач и условий использования.

### **2.1.1 Механическая рука**

Механическая рука, являющаяся центральным элементом системы, представляет собой сложную конструкцию из нескольких соединенных звеньев, управляемых сервоприводами. Это позволяет руке двигаться в трех измерениях и воспроизводить различные формы. На конце руки расположен маркер, который наносит рисунок на бумагу.

### **2.1.2 Графический редактор**

Графический редактор - это специализированное программное обеспечение, разработанное для создания рисунков. Он предоставляет пользовательский интерфейс, в котором можно выбирать различные инструменты для рисования, такие как прямые линии, квадраты, ромбы, прямоугольники и треугольники. После создания рисунка в графическом редакторе, он передается в механическую руку для воспроизведения.

### **2.1.3 Взаимодействие компонентов**

Механическая рука и графический редактор взаимодействуют друг с другом для достижения цели проекта. Компьютер взаимодействует с платой при помощи кабеля от Arduino Uno. Пользователь создает рисунок в графическом редакторе, который затем преобразуется в набор команд для кода, написанного в Arduino IDE серв механической рукой. Эти команды управляют движением руки, позволяя ей точно воспроизвести рисунок на бумаге. Этот процесс взаимодействия компонентов был тщательно продуман и оптимизирован для обеспечения высокой производительности и надежности системы.

## **2.2 Сборка робота**

Процесс сборки робота “DrawMan” начинается с печати всех необходимых деталей на 3D-принтере [4]. Используется PLA-пластик золотого цвета, который обеспечивает прочность и легкость деталей. Этот материал был выбран из-за его превосходных свойств, таких как прочность, устойчивость к воздействию окружающей среды и возможность легкой обработки после печати.

После печати каждая деталь [5] тщательно очищается от лишнего материала и проверяется на соответствие размерам. Этот этап критически важен для обеспечения точности и надежности конечного продукта.

Затем начинается сборка механической руки. Сначала устанавливаются сервоприводы, которые будут управлять движением руки. Каждый сервопривод крепится к соответствующему звену руки с помощью специальных крепежных элементов, обеспечивающих надежное и стабильное соединение.

После установки сервоприводов крепится маркер, который будет наносить рисунок на бумагу. Маркер устанавливается в специальное устройство на конце руки, которое позволяет легко заменять маркер при необходимости.

Всего используется три сервопривода. Нижний отвечает за перемещение конструкции вокруг своей оси. Один из двух боковых сервоприводов отвечает за движение вперед-назад, а вторая — за подъем механизма вверх-вниз. Эта конфигурация обеспечивает высокую гибкость и точность движений руки.

После сборки механической руки происходит подключение платы Arduino Uno. В основе данной системы лежит плата Arduino Uno. Arduino Uno — это микроконтроллерная плата на основе ATmega328P. Она имеет 14 цифровых входных/выходных пинов (из которых 6 могут использоваться как выходы ШИМ), 6 аналоговых входов, 16 МГц кварцевый резонатор, USB-разъем, разъем питания, ICSP-заголовок и кнопку сброса.

К плате Arduino Uno подключен Troyka Shield (Amperka) [6]. Этот шилд позволяет подключать до трех сервоприводов, что идеально подходит для нашего проекта, поскольку используется три сервы для управления механической рукой.

Ниже на рисунке 2.1 представлен пример подключения сервопривода к плате Arduino Uno.

Изображение выглядит как текст, электроника, Электронная техника, схема

Автоматически созданное описание

Рисунок 2.1 — Схема подключения сервопривода [7]

Подключение остальных сервоприводов происходит по тому же принципу, только к шилду. Т.е. красный провод подключается к земле (GND), желтый — 5 вольтам (5V), коричневый — к пину.

Таким образом, сборка робота “DrawMan” - это сложный процесс, который включает в себя несколько этапов: печать деталей, сборку механической руки, подключение платы Arduino Uno. Этот процесс требует тщательности и внимания к деталям, чтобы обеспечить правильную работу системы. Каждый этап был тщательно продуман и оптимизирован для обеспечения высокой производительности и надежности конечного продукта.

## **2.3 Разработка программного обеспечения на Ардуино для управления автоматизированной рукой с компьютера**

### **2.3.1 Библиотеки и глобальные переменные**

В начале кода подключаются две ключевые библиотеки: Servo.h [8] и StaticSerialCommands.h [9]. Библиотека Servo.h предоставляет функциональность для управления сервоприводами, позволяя управлять их положением с высокой точностью. Библиотека StaticSerialCommands.h используется для обработки команд, отправленных через последовательный порт, что позволяет управлять рукой в реальном времени.

После подключения библиотек объявляются глобальные переменные и объекты, которые будут использоваться в коде. Это включает в себя объекты Servo для каждого из сервоприводов, а также переменные для хранения текущих углов сервоприводов и состояния пера. Эти переменные играют важную роль в управлении поведением руки и отслеживании ее текущего состояния.

**2.3.2 Функции управления сервоприводами**

Функции setServo, setAlpha, setBeta и setPen используются для управления сервоприводами. Они принимают углы или состояния в качестве параметров и преобразуют их в сигналы, которые могут быть поняты сервоприводами. Эти функции обеспечивают гибкость и точность управления движением руки.

**2.3.3 Функции обработки команд**

Функции cmd\_help, cmd\_pen, cmd\_alpha и cmd\_beta служат для обработки команд, отправленных через последовательный порт. Они принимают объект SerialCommands и аргументы команды в качестве параметров и выполняют соответствующие действия. Эти функции обеспечивают возможность интерактивного управления рукой через последовательный порт, что делает систему удобной и гибкой в использовании.

Код Arduino содержит команды, которые позволяют компьютеру взаимодействовать с платой:

1Команда “help” выводит список всех доступных команд.

2 Команда “p” устанавливает положение пера, принимая аргумент isDown, который может быть равен 0 или 1. Если isDown равно 1, перо опускается, если isDown равно 0, перо поднимается.

3 Команда “a” устанавливает угол alpha для сервопривода sAlpha, принимая аргумент angle, который должен быть в диапазоне от 60 до 120 градусов.

4 Команда “b” устанавливает угол beta для сервоприводов sBeta и sBetaUp, принимая аргумент angle, который должен быть в диапазоне от 20 до 70 градусов. Эти команды можно отправить на плату Arduino через последовательный порт с помощью программы на компьютере. Это обеспечивает гибкость и точность управления, что является важным аспектом для обеспечения высокого качества работы системы.

**2.3.4 Настройка и основной цикл**

В функции setup происходит инициализация сервоприводов и последовательного порта. Затем устанавливаются начальные углы сервоприводов и состояние пера. Это обеспечивает правильную инициализацию системы перед началом работы.

В функции loop происходит чтение и обработка команд, отправленных через последовательный порт. Это обеспечивает непрерывное взаимодействие с рукой в реальном времени.

Код данного программы можно увидеть на GitHub. На представленном ниже рисунке демонстрируется окно взаимодействия компьютера и платы.

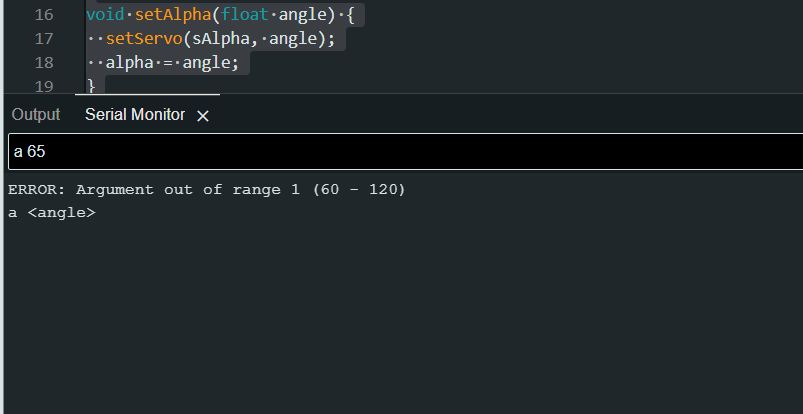


Рисунок 2.2 — Окно взаимодействия компьютера и платы через SerialCommands

## **2.4 Графический редактор**

Графический редактор [10], разработанный для проекта “Robot DrawMan”, представляет собой интуитивно понятный интерфейс, который позволяет пользователям легко создавать рисунки. Он включает в себя набор инструментов для рисования различных геометрических фигур, таких как прямые линии, квадраты, ромбы, прямоугольники и треугольники.

Каждый из этих инструментов представлен кнопкой на интерфейсе редактора. Пользователи могут выбрать нужный инструмент, нажав на соответствующую кнопку. Затем, чтобы нарисовать фигуру, пользователю нужно выбрать точку на холсте, которая станет центром фигуры, и потянуть в сторону для рисования.

В процессе рисования координаты центра фигуры отображаются в нижней части интерфейса. Эти координаты вычисляются относительно левого верхнего угла холста.

После того, как рисунок создан, пользователь может нажать на кнопку [11] “Click me”. Это активирует процесс передачи информации о рисунке на компьютер, который затем преобразует эту информацию в набор команд для механической руки. В результате рука начинает воспроизводить рисунок на бумаге, повторяя каждую линию и форму.

На рисунке 2.3 представлен пользовательский интерфейс графического редактора.

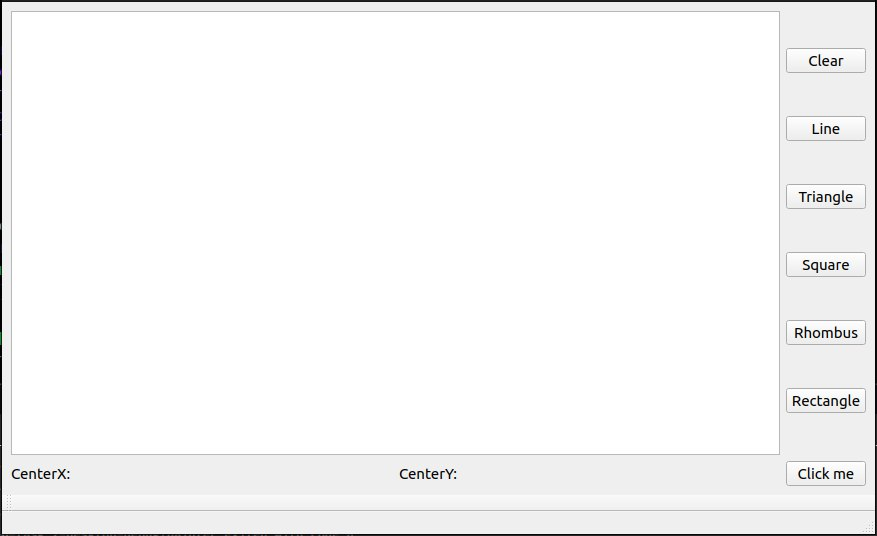


Рисунок 2.3 — Пользовательский интерфейс

Этот интерфейс был разработан с учетом удобства использования, обеспечивая простоту и интуитивность взаимодействия с пользователем. Он демонстрирует, как пользователи могут легко и эффективно создавать рисунки, используя доступные инструменты и функции.

# 3 **ТЕСТИРОВАНИЕ И ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММНОГО СРЕДСТВА**

В ходе тестирования программного обеспечения проекта “Robot DrawMan” [12] были выявлены и устранены определенные ошибки, что способствовало улучшению его функциональности и надежности.

Во время одного из запусков манипулятора было обнаружено, что он не работает должным образом, а сервоприводы не издавали характерного звука работы. После диагностики двух сервоприводов было выявлено, что внутри них сломано несколько шестерен. Дальнейший анализ показал, что на сервоприводы оказывалась слишком большая нагрузка из-за сложности конструкции и большого веса маркера. В результате, сервоприводы, рассчитанные на вес 1,4 кг, пришлось заменить на более мощные, с грузоподъемностью 1,8 кг. Это позволило успешно решить проблему и улучшить работоспособность манипулятора.

Другая ошибка была визуального характера и связана с высотой крепления маркера. При первом запуске маркер был закреплен слишком низко, что приводило к тому, что он не скользил по бумаге, а упирался в поверхность, мешая движению манипулятора. После нескольких итераций тестирования и корректировки была найдена оптимальная высота крепления маркера, что позволило устранить эту проблему и обеспечить более плавное и точное движение манипулятора.

После устранения этих ошибок было проведено несколько дополнительных тестов программного обеспечения. Эти тесты включали в себя проверку работоспособности системы в различных условиях, тестирование различных функций и возможностей программного обеспечения, а также проверку стабильности и надежности системы при длительной работе. Теперь при каждом включении система работает корректно и без сбоев, что подтверждает ее высокую работоспособность и надежность.

В целом, процесс тестирования и валидации был важным этапом разработки программного обеспечения проекта “Robot DrawMan”. Он позволил обнаружить и устранить различные ошибки, улучшить функциональность и надежность системы, а также обеспечить ее безопасность и удобство использования.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Github [Электронный ресурс]. — Режим доступа: <https://github.com/Vediz-antonova/RobotArm>. – Дата доступа: 21.05.2024.

[2] Ардуино на русском [Электронный ресурс]. — Режим доступа: <https://arduino-site.ru/spravochnik-arduino/>. – Дата доступа: 21.05.2024.

[3] Qt Documentation [Электронный ресурс]. – Режим доступа: <https://doc.qt.io/qt-5/resources.html>. – Дата доступа: 15.05.2024.

[4] Alex Gyver Technologies [Электронный ресурс]. — Режим доступа: <https://alexgyver.ru/3d-printers/>. – Дата доступа: 21.05.2024.

[5] Thingiverse [Электронный ресурс]. — Режим доступа: https://www.thingiverse.com/thing:6025043. – Дата доступа: 21.05.2024.

[6] Amperka [Электронный ресурс]. — Режим доступа: <https://amperka.ru/product/arduino-troyka-shield>. – Дата доступа: 21.05.2024.

[7] Alex Gyver Technologies [Электронный ресурс]. — Режим доступа: <https://github.com/fritzing/fritzing-app/releases/download/CD-548/fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.windows.64.zip>. – Дата доступа: 21.05.2024.

[8] Arduino [Электронный ресурс]. — Режим доступа: <https://www.arduino.cc/reference/en/libraries/servo/>. – Дата доступа: 21.05.2024.

[9] Arduino [Электронный ресурс]. — Режим доступа: [[SerialCommand - Arduino Reference](https://www.arduino.cc/reference/en/libraries/serialcommand/)](https://www.arduino.cc/reference/en/libraries/servo/). – Дата доступа: 21.05.2024.

[10] ItProger [Электронный ресурс]. — Режим доступа: <https://itproger.com/course/qt-creator>. – Дата доступа: 22.05.2024.

[11] Qt Reference [Электронный ресурс]. — Режим доступа: https://doc.qt.io/qt-6/qpushbutton.html. – Дата доступа: 21.05.2024.

[12] YouTube [Электронный ресурс]. — Режим доступа: <https://youtube.com/@vediz9556?si=TEmadEVwwHeNgkIL>. – Дата доступа: 21.05.2024.

# ПРИЛОЖЕНИЕ А Исходный код программы

#include <Servo.h>

#include <StaticSerialCommands.h>

Servo sAlpha;

Servo sBeta;

Servo sBetaUp;

bool penDown;

float alpha;

float beta;

void setServo(Servo &servo, float angle) {

float microSeconds = 1856.0 \* angle / 180 + 544;

servo.writeMicroseconds(microSeconds);

}

void setAlpha(float angle) {

setServo(sAlpha, angle);

alpha = angle;

}

float getPenUpBetaModifier(float angle) {

if (penDown) { return 0; }

if (angle <= 40) { return 12.0; }

if (angle <= 50) { return 16.0; }

if (angle <= 60) { return 20.0; }

return 28.0;

}

void setBeta(float angle) {

float revAngle = 180.0 - angle;

float betaModifier = getPenUpBetaModifier(angle);

setServo(sBeta, revAngle - betaModifier - 10);

setServo(sBetaUp, revAngle + betaModifier + 10);

beta = angle;

}

void setPen(bool isDown) {

penDown = isDown;

setBeta(beta);

}

void cmd\_help(SerialCommands& sender, Args& args) {

sender.listCommands();

}

void cmd\_pen(SerialCommands& sender, Args& args) {

int isDown = args[0].getInt();

setPen(isDown);

}

// … Остальное на GitHub