

**Московская предпрофессиональная олимпиада**

**Инженерно-конструкторское направление. Инженерно-конструкторский профиль**

**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**

Государственное бюджетное общеобразовательное учреждение  
города Москвы «Школа № 1103 имени Героя Российской Федерации  
А.В. Соломатина»

**Продуктовый сектор. Профиль «Инженерия»**

**Пояснительная записка к кейсу №1**

**Принтер Брайля с голосовым управлением**

Выполнили:

ученики 11Б класса ГБОУ Школы №1103

Ларкин Григорий Максимович,

Григорьев Андрей Андреевич,

Приоров Артемий Сергеевич,

Рахманин Максим Олегович,

Донцов Илья Игоревич

Руководитель:

Сокур М.Е., учитель информатики

**Москва, 2024**

## Оглавление

1. Цель и задачи работы.....	3
2. Описание команды.....	4
3. Описание функций разработанного решения .....	5
4. Описание используемых аппаратных и программных узлов, модулей, фреймворков и других инструментов.....	6
5. Функциональное описание разработанного решение в виде UML-диаграмм.....	9
6. Описание кинематической системы разработанного устройства.....	13
7. Скриншоты разработанных 3D-моделей .....	14
8. Описание электротехнической схемы разработанного устройств .....	17
9. Алгоритм работы разработанного программного обеспечения в виде блок-схем.....	18
10. Код разработанного программного обеспечения. ....	21
11. Фотографии разработанного устройства и его составных частей. ....	21
12. Видеоролик, демонстрирующий функционирование разработанного устройства .....	21
13. Заключение .....	21
14. Список литературных источников. ....	22

## **1. Цель и задачи работы**

1.1. Цель работы – Разработать и изготовить принтер Брайля (далее программно-аппаратный комплекс или ПАК) с расширенным функционалом, в том числе с голосовым управлением, для быстрого создания печатных заметок для людей с ограничениями по зрению.

### **1.2. Задачи работы:**

- Разработать и создать корпус ПАК;
- Разработать и создать кинематическую схему печати символов;
- Разработать и создать электротехническую систему устройства;
- Выбор микроконтроллера для разработанной системы;
- Разработать систему голосового управления для распознавания текста и команд;
- Разработать алгоритмы и ПО для разработанной архитектуры ПАК.

## 2. Описание команды

Членами нашей команды являются Григорьев Андрей, Донцов Илья, Ларкин Григорий, Приоров Артемий, Рахманин Максим. Распределение ролей представлено в Таблица 1.

Таблица 1. Распределение ролей в команде

<b>Фамилия, имя ученика</b>	<b>Задача ученика в команде</b>
Григорьев Андрей	Сборка ПАК, исправление возникших проблем
Донцов Илья	Создание программного кода
Ларкин Григорий	Сборка ПАК, поиск требуемых компонентов и материалов
Приоров Артемий	Создание 3D-моделей ПАК, чертежей, кинематических схем
Рахманин Максим	Разработка документации, создание UML-схем

### **3. Описание функций разработанного решения**

3.1. Управление принтера производится с помощью голосовых команд, произносимых пользователем. Например, команда «Абзац» делает перенос на новую строку.

3.2. Когда пользователь произносит команду «Запись», встроенный микрофон начинает распознавать и записывать слова в течении 5 секунд. При необходимости максимальное время записи можно программно изменить.

3.3. Записанная на микрофон речь передаётся микрокомпьютером Raspberry Pi 3 в сервис Yandex Cloud, где происходит её распознавание. Текстовое содержимое сохраняется на микрокомпьютере.

3.4. Как только речь была записана, Raspberry Pi 3 передаёт её на микроконтроллер Arduino. Там речь переводится в текст Брайля и подаётся на печать.

3.5. Печать происходит построчно. Печатная головка двигается по горизонтали с помощью шагового мотора. Прокрутка бумаги по вертикали реализована вторым шаговым мотором и системой прижимных роликов.

#### 4. Описание используемых аппаратных и программных узлов, модулей, фреймворков и других инструментов

Список основных компонентов, используемых для создания данного проекта, представлен в Таблица 2.

Таблица 2. Используемые компоненты и ПО

Название компонента и его назначение в проекте	Внешний вид
<p>Соленоид JF-1253B</p> <p>В данном проекте он используется для печати на бумаге рельефно-точечным шрифтом Луи Брайля.</p>	
<p>Шаговый мотор Nema17</p> <p>Предназначены для перемещения печатной головки по горизонтальной плоскости, листа бумаги по вертикальной плоскости</p>	
<p>Шкив GT2-20</p> <p>С помощью ременной передачи вращение с шаговых двигателей передаётся на ремень, который приводит в движение печатающую головку принтера</p>	

<p>Концевой переключатель</p> <p>Подаёт на плату Arduino сигнал, когда печатная головка дошла до крайнего положения листа. После получения этого сигнала продвигается бумажный лист, чтобы продолжить печать на новой строке.</p>	
<p>Arduino Uno</p> <p>Микроконтроллер, отвечающий за управление шаговыми моторами и печатью. Весь программный код писался в интегрированной среде разработки Arduino IDE, использующий C-подобный язык программирования.</p>	
<p>Motor Shield</p> <p>Плата расширения, предназначенная для управления скоростью и направлению вращения шаговых моторов постоянного тока.</p>	
<p>Raspberry Pi 3</p> <p>Микрокомпьютер, на котором, за счёт облачных сервисов, реализовано распознавание речи.</p> <p>В данном проекте используются следующие модули: time, requests, RPi.GPIO, soundfile, pyaudio, wave, serial</p>	

<p>САПР «Компас-3D»</p> <p>Средство автоматического проектирования, в котором создавались 3D модели деталей, чертежи, кинематические схемы</p>	
<p>Arduino IDE</p> <p>Интегрированная среда программирования. Предназначена для разработки и загрузки программного кода на Arduino-совместимые платы.</p> <p>В данном проекте используются следующие библиотеки: AFMotor, TMRpcm</p>	



## 5. Функциональное описание разработанного решение в виде UML-диаграмм

### 5.1. Диаграмма вариантов пользовательского взаимодействия с системой

Произнеся команду «Запись», пользователь может говорить в микрофон текст для печати, а также команды для принтера, управляя его работой. Также пользователь может класть и доставать бумагу для печати. В необходимый момент, пользователь может нажать на кнопку Reset, тем самым остановить печать.

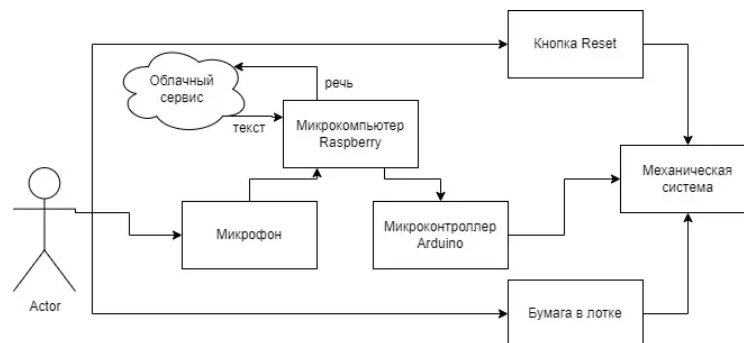


Рисунок 1. Диаграмма вариантов пользовательского взаимодействия с системой

## 5.2. Диаграмма автомата

У принтера есть 2 основных состояния: выключенное и включённое. Во втором состоянии он выполняет свои основные функции: после включения он ждёт голосовой команды «Запись» для начала работы, после чего записывает речь пользователя. Записанный аудиофайл сохраняется на микрокомпьютер Raspberry Pi 3, который отправляет его на облачный сервис Yandex Cloud, где происходит распознавание речи. Получившийся в результате работы облачного сервиса текстовый файл с распознанной речью отправляется обратно на Raspberry Pi 3. Далее текст отправляется на микроконтроллер Arduino, где преобразуется в текст Брайля и отправляется на печать.

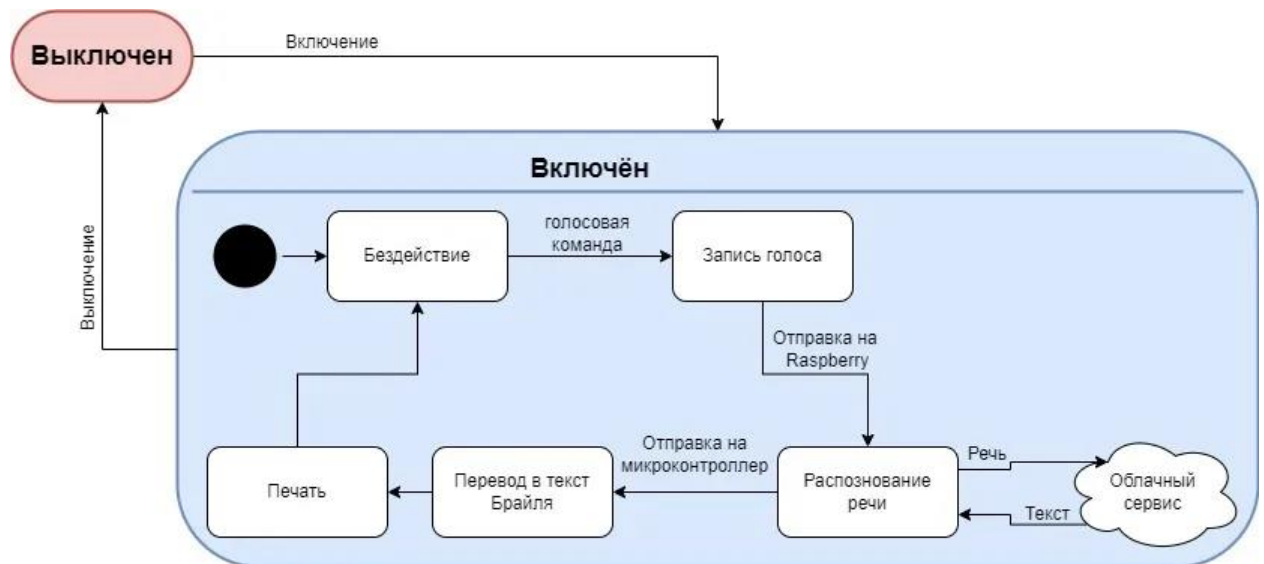


Рисунок 2. Диаграмма автомата

### 5.3. Диаграмма последовательности

После включения система ждёт, когда пользователь произнесёт команду «Запись» для начала записи. После записи полученная речь отправляется микрокомпьютером Raspberry Pi 3 для распознавания. С помощью облачного сервиса Yandex Cloud реализовано распознавание речи. После обработки микрокомпьютер передаёт полученный текст на микроконтроллер, где происходит перевод в текст Брайля и совершается отправка на печать. При обнаружении неполадки или иных непредвиденных ситуациях, пользователь может нажать на кнопку Reset, чтобы совершить аварийную остановку.

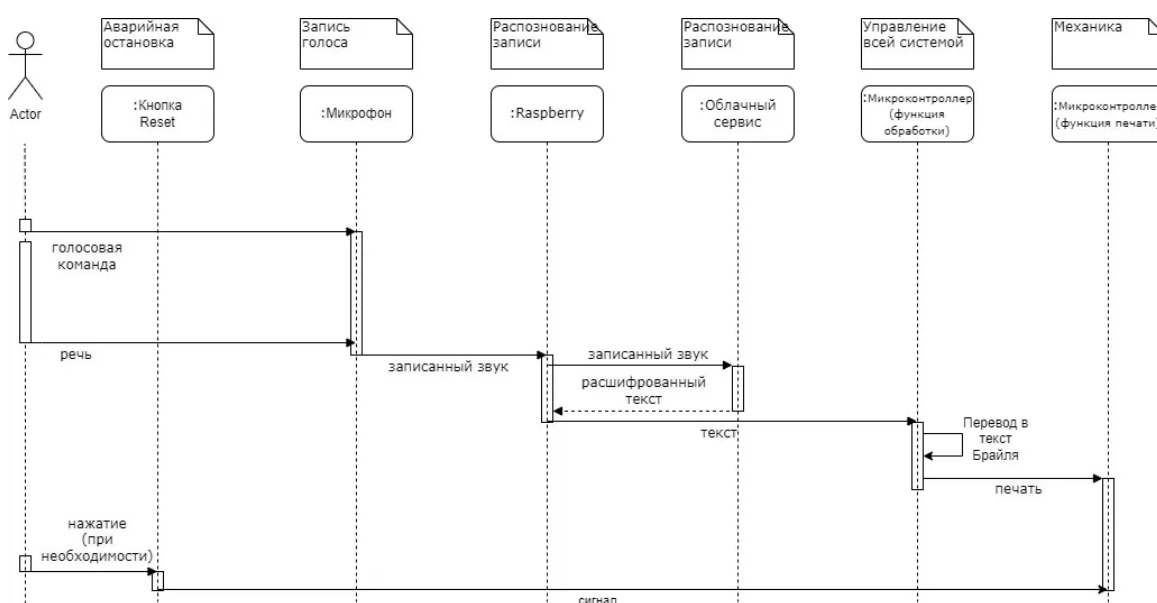


Рисунок 3. Диаграмма последовательности

#### 5.4. Диаграмма компонентов

В нашем проекте используется один микроконтроллер, к которому подключены все составляющие механизма печати. Также мы используем микрокомпьютер Raspberry Pi 3 с подключенным к нему микрофоном для распознавания речи. Специальная команда начала записи активирует микрофон, а кнопка Reset напрямую взаимодействует с механизмом печати и может остановить процесс при необходимости. Полученная при записи речь сохраняется на микрокомпьютер Raspberry Pi 3, который отправляет её на облачный сервис Yandex Cloud, где происходит распознавание речи. Полученный таким образом текстовый файл обратно возвращается на микрокомпьютер, который передаёт его микроконтроллер Arduino Uno, где преобразуется по шрифту Брайля и отправляется на печать.

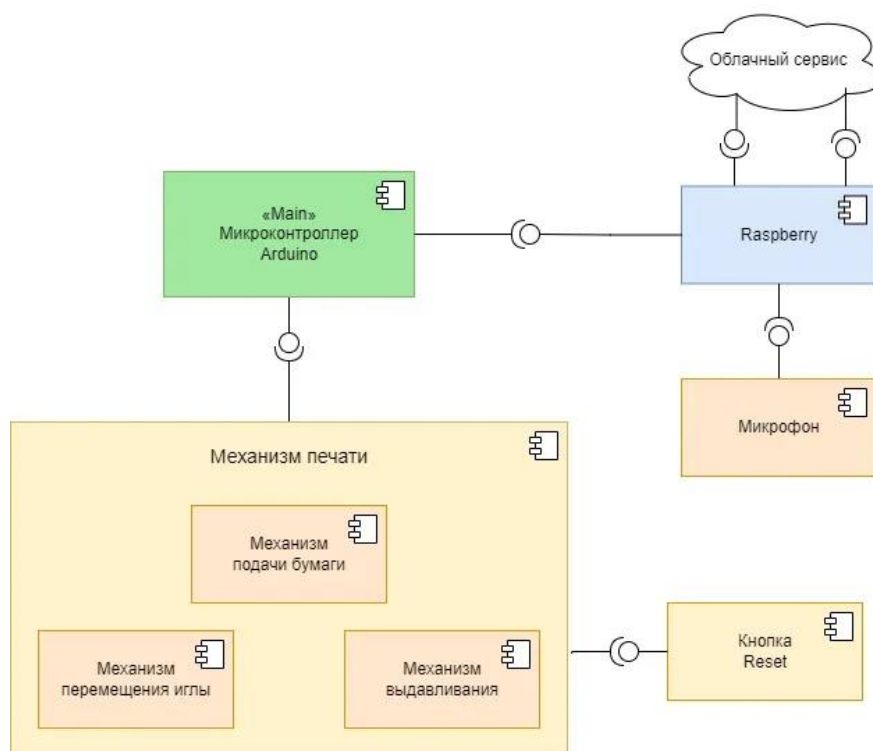


Рисунок 4. Диаграмма компонентов

## 6. Описание кинематической системы разработанного устройства

Моторчик, обозначенный на схеме буквой М, вращает шкив GT2-20, приводя в движение ремень, который перемещает печатную головку по горизонтальной плоскости.

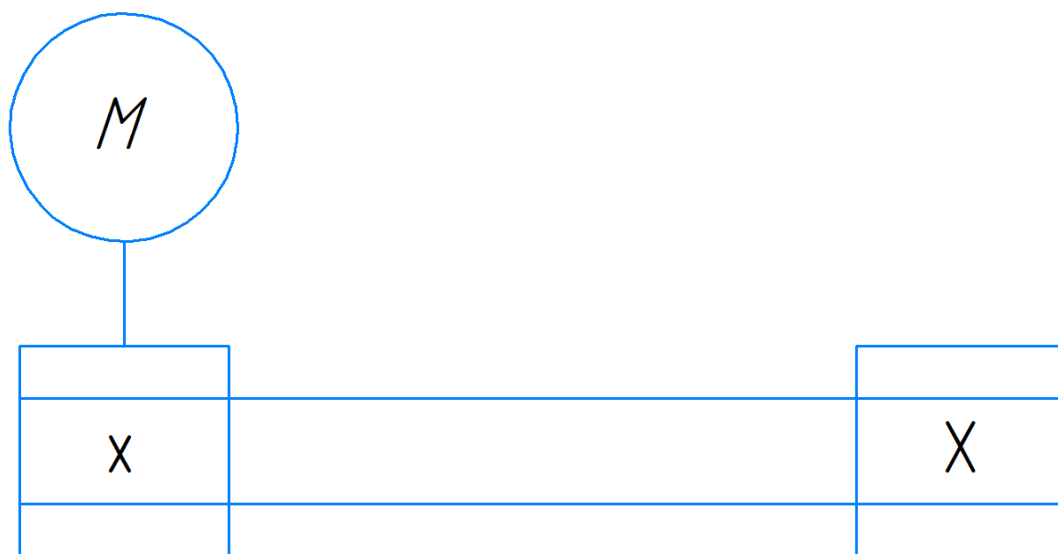


Рисунок 5. Кинематическая схема печатной головки

Ниже представлена кинематическая схема вала, на котором закреплены ролики, прижимающие лист бумаги идвигающие его в вертикальной плоскости.

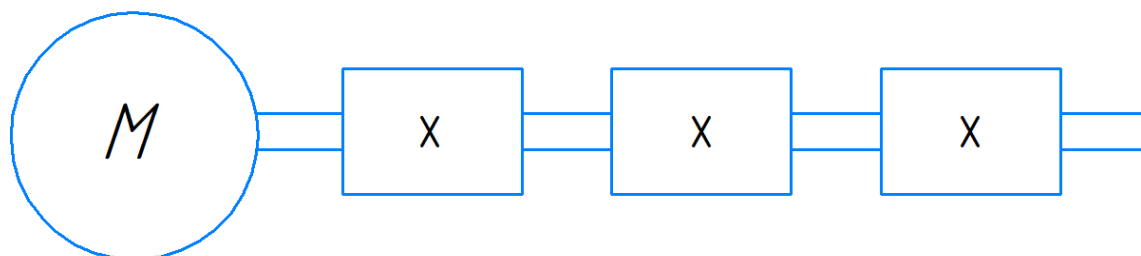


Рисунок 6. Кинематическая схема вала с прижимными роликами

## 7. Скриншоты разработанных 3D-моделей

Далее мы представим некоторые скриншоты разработанных 3D-моделей. Все скриншоты Вы сможете найти в папке «3D-модели» репозитория.

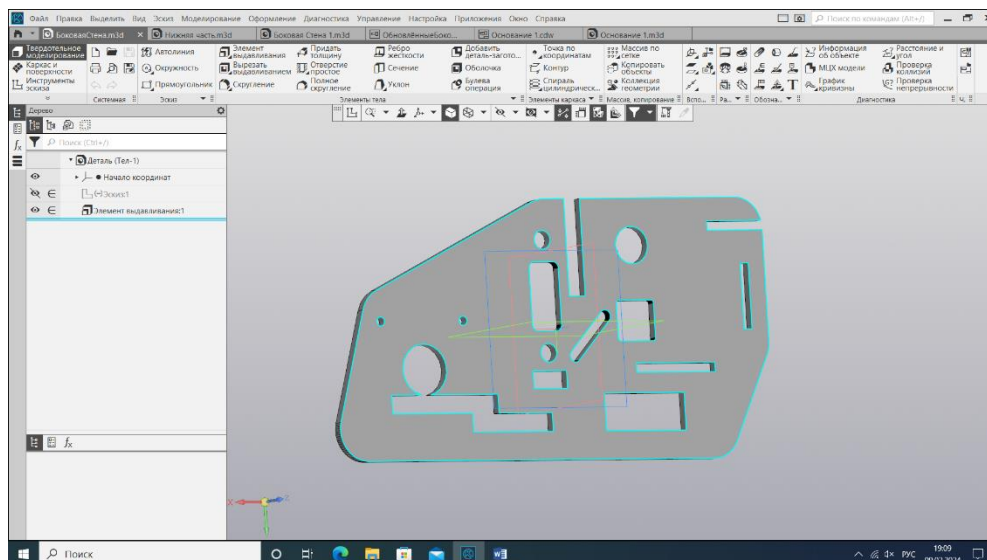


Рисунок 7. Боковая стенка

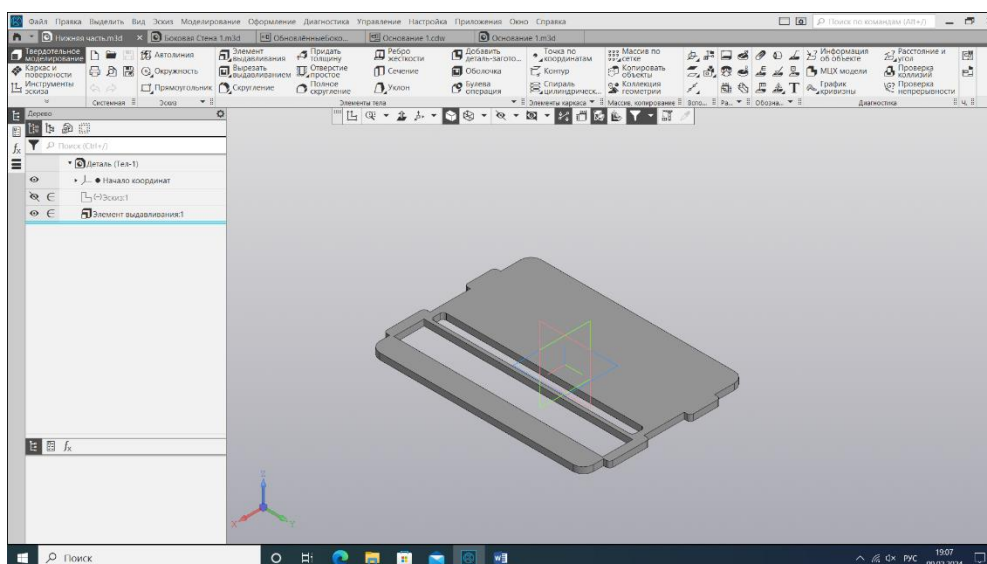


Рисунок 8. Нижняя деталь

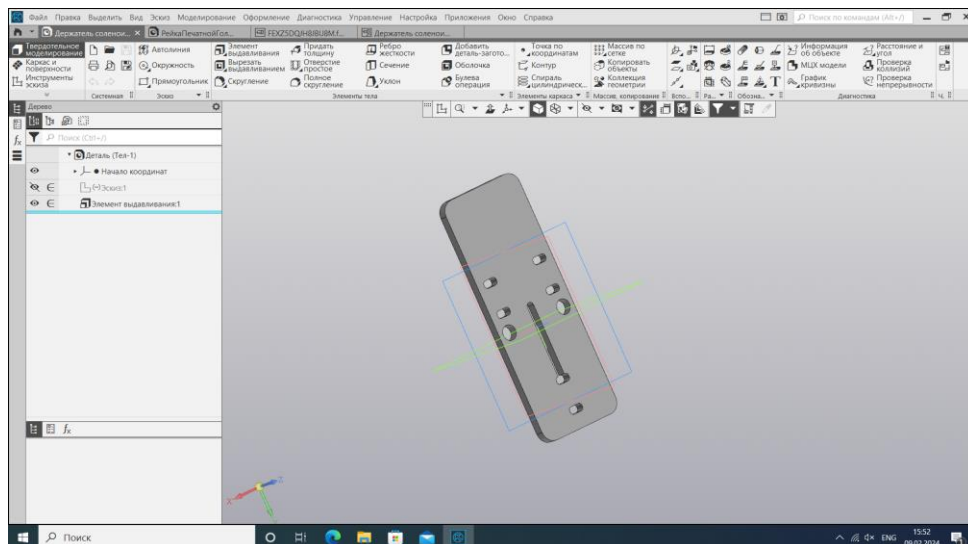


Рисунок 9. Держатель соленоида

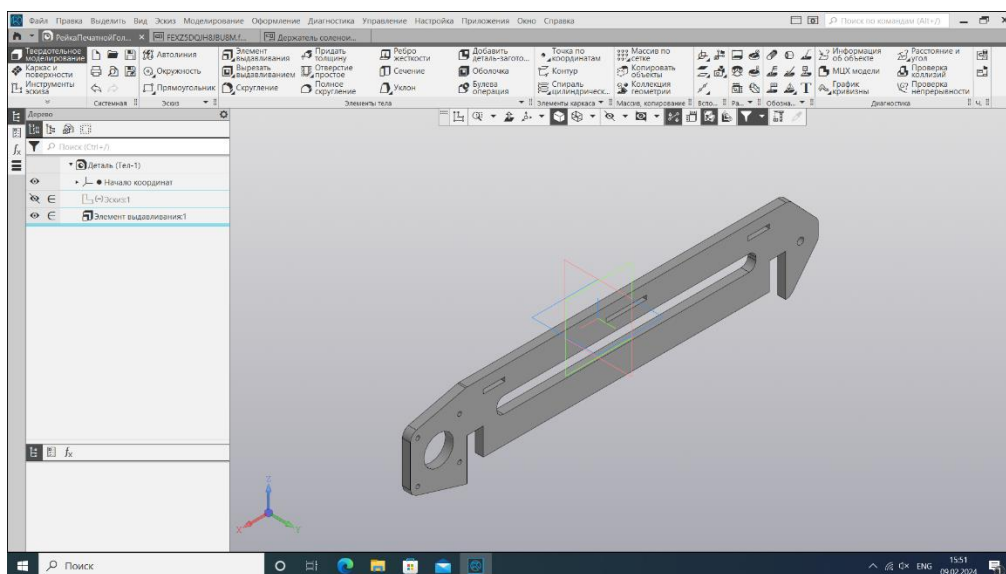


Рисунок 10. Рейка печатной головки

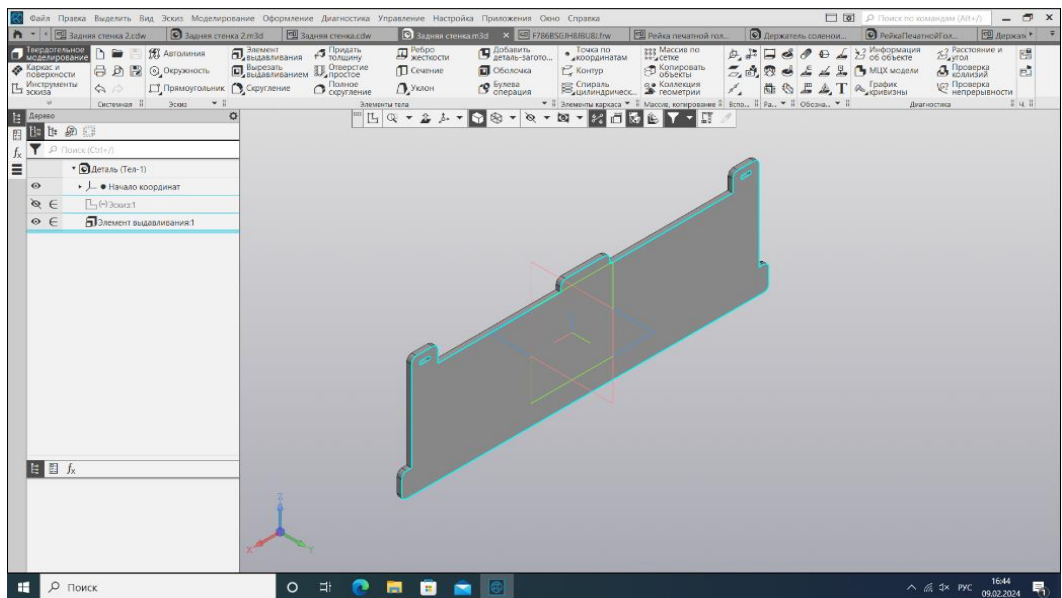


Рисунок 11. Задняя стенка

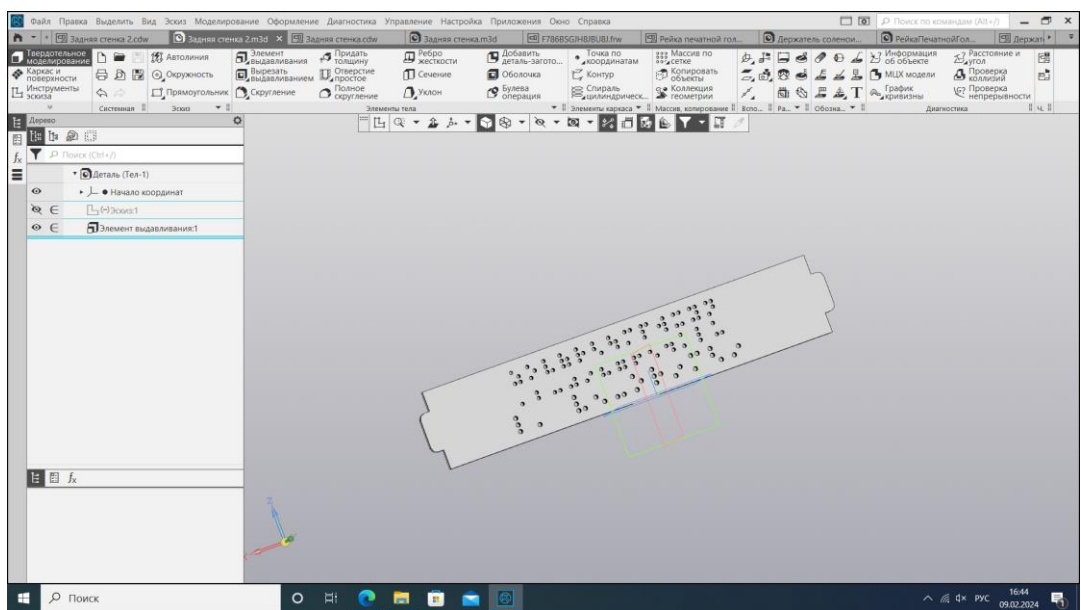


Рисунок 12. Задняя стенка 2



17

**9. Алгоритм работы разработанного программного обеспечения  
в виде блок-схем**

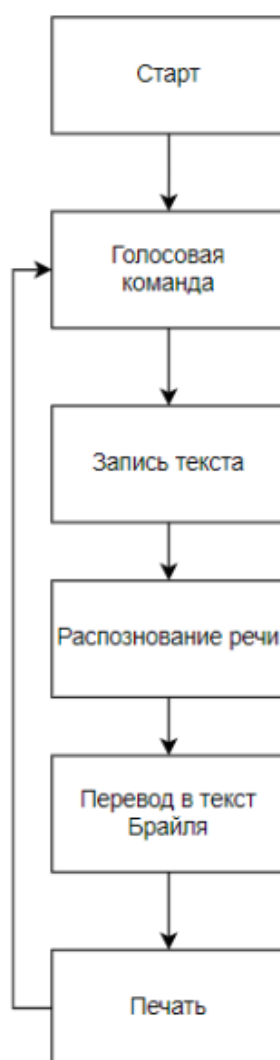


Рисунок 14. Общий алгоритм ПО

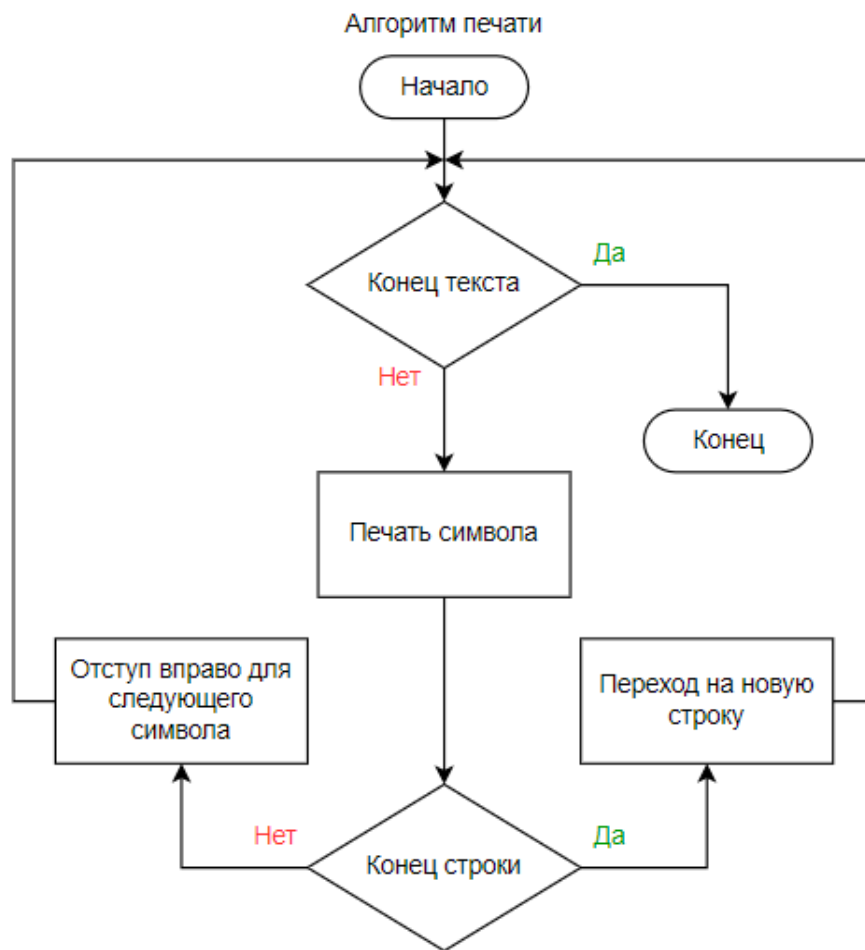


Рисунок 15. Алгоритм печати

Алгоритм печати символа

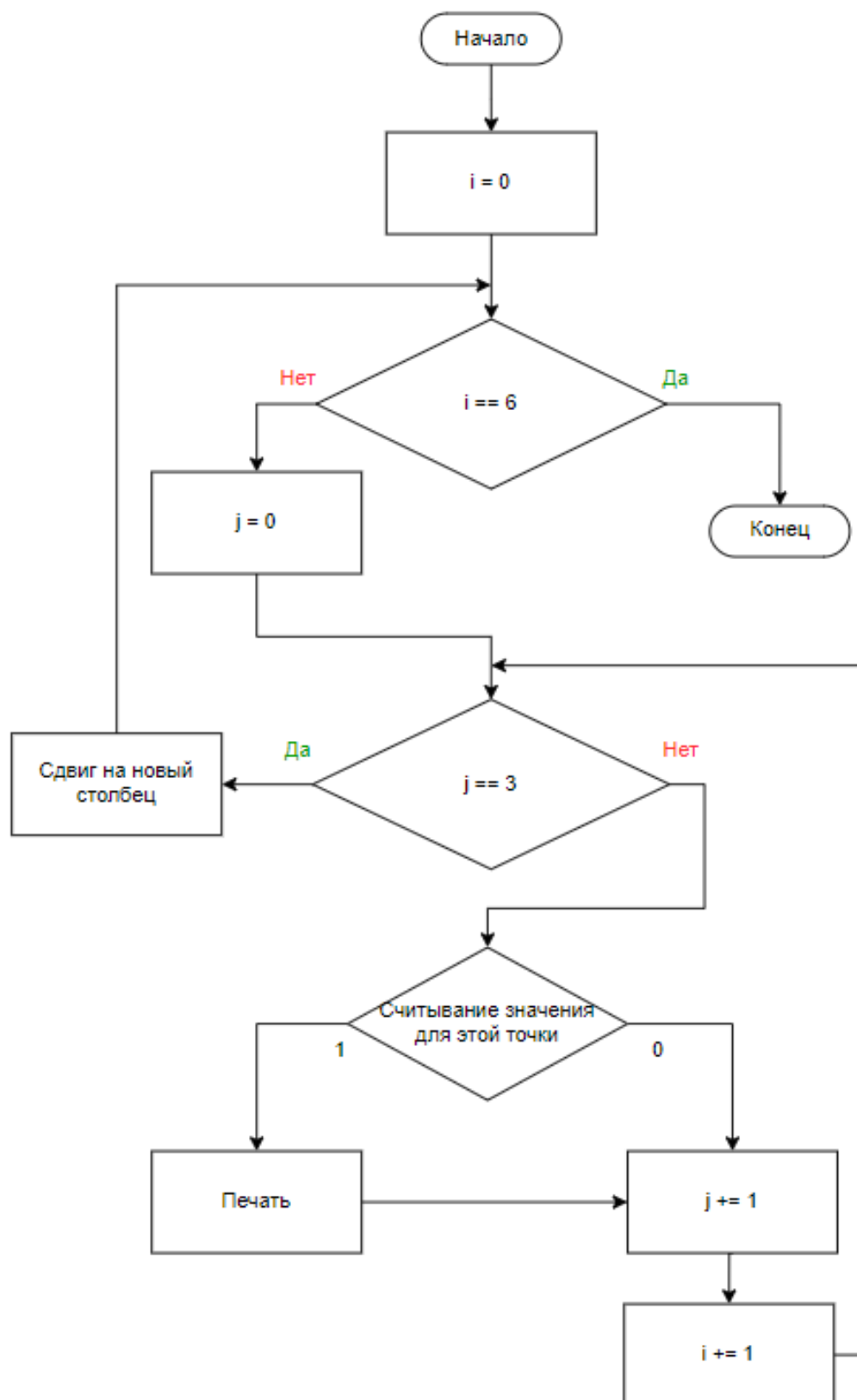


Рисунок 16. Алгоритм печати символа

## **10. Код разработанного программного обеспечения.**

Код для работы нашего проекта Вы можете найти по этой ссылке:

- [https://github.com/Deimos48/Braille\\_Public/tree/main/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9\\_%D0%BA%D0%BE%D0%B4/Python](https://github.com/Deimos48/Braille_Public/tree/main/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4/Python)
- [https://github.com/Deimos48/Braille\\_Public/tree/main/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9\\_%D0%BA%D0%BE%D0%B4/Arduino](https://github.com/Deimos48/Braille_Public/tree/main/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4/Arduino)

## **11. Фотографии разработанного устройства и его составных частей.**

Фотографии разработанного устройства Вы можете найти по этой ссылке:

- [https://github.com/Deimos48/Braille\\_Public/tree/main/Фотографии\\_разработанного\\_устройства](https://github.com/Deimos48/Braille_Public/tree/main/Фотографии_разработанного_устройства)

## **12. Видеоролик, демонстрирующий функционирование разработанного устройства**

Видеоролик Вы можете найти, по ссылке:

- [https://github.com/Deimos48/Braille\\_Public/tree/main/Видеоролик](https://github.com/Deimos48/Braille_Public/tree/main/Видеоролик)

## **13. Заключение**

В результате работы был создан принтер Брайля с голосовым управлением, который способен:

1. По специальной голосовой команде считывает речь пользователя;
2. Производить печать произнесённой речи на бумагу рельефно-точечным шрифтом Брайля;
3. Переносить печать на новую строчку с помощью специальной голосовой команды.

#### **14. Список литературных источников.**

1. Рельефно-точечная система обозначений Л. Брайля: учебное пособие / сост. Н.П.Шведова, В.З. Денискина. – Москва: МПГУ, 2019. – 76с.
2. ГОСТ Р 56832-2020 Шрифт Брайля. Требования и размеры
3. Обучающие уроки и проекты для Arduino, ESP, Raspberry Pi 3 Pi. URL: <https://lesson.iarduino.ru>
4. КОМПАС-3D - российская система трехмерного проектирования. URL: <https://kompas.ru/solutions/education/>
5. Документация по GitHub. URL: <https://docs.github.com/ru/get-started/quickstart/hello-world>