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 №5**

Модель автоматической парковки

Выполнили:

ученики 8И класса ГБОУ Школы №1103

Нарышкин Елисей Игоревич,

Хаиртдинов Руслан Ленарович,

БахчинянцКристиан Владиславович,

Романов Михаил Алексеевич,

Чикин Максим

Руководитель:

Сокур М.Е., учитель информатики

**Москва, 2025**

**Оглавление**

[1. Постановка задачи (условия) 3](#_Toc158470773)

[2. Описание команды 4](#_Toc158470774)

[3. Описание функций разработанного решения 5](#_Toc158470775)

[4. Описание используемых аппаратных и программных узлов, модулей, фреймворков и других инструментов 6](#_Toc158470776)

[5. Функциональное описание разработанного решение в виде UML-диаграмм 8](#_Toc158470777)

[6. Описание кинематической системы разработанного устройства 12](#_Toc158470778)

[7. Скриншоты разработанных 3D-моделей 13](#_Toc158470779)

[8. Описание электротехнической схемы разработанного устройств 16](#_Toc158470780)

[9. Алгоритм работы разработанного программного обеспечения в виде блок-схем 17](#_Toc158470781)

[10. Код разработанного программного обеспечения. 20](#_Toc158470782)

[11. Фотографии разработанного устройства и его составных частей. 20](#_Toc158470783)

[12. Видеоролик, демонстрирующий функционирование разработанного устройства 20](#_Toc158470784)

[13. Заключение 20](#_Toc158470785)

[14. Список литературных источников. 21](#_Toc158470786)

1. Цель и задачи работы
   1. Цель работы – разработка программно-аппаратного комплекса (ПАК), способного в автоматическом режиме размещать и забирать автомобиль в автоматической парковке.
   2. Задачи работы:
   * Разработать и создать ПАК;
   * Разработать и создать электротехническую систему устройства;
   * Выбор микроконтроллеров для разработанной системы;
   * Разработать алгоритмы и ПО для разработанной архитектуры ПАК.
2. Описание команды

Членами нашей команды являются Нарышкин Елисей, Хаиртдинов Руслан, Романов Михаил, Чикин Максим, Бахчинянц Кристиан. Распределение ролей представлено в таблице 1.

Таблица . Распределение ролей в команде.

|  |  |
| --- | --- |
| **Фамилия, имя ученика** | **Задача ученика в команде** |
| Нарышкин Елисей | Разработка документации, создание UML-схем |
| Хаиртдинов Руслан | Сборка ПАК, исправление возникших проблем |
| Романов Михаил | Сборка ПАК, поиск требуемых компонентов и материалов |
| Чикин Максим |  |
| Бахчинянц Кристиан | Создание программного кода |

# Описание функций разработанного решения

* 1. Разработанный ПАК работает в автоматическом режиме после подачи сигнала о начале работы (кнопочный ввод). Разработанная модель автомобиля работает в автоматическом режиме после подачи сигнала о начале работы (переключение тумблера).
  2. Видеокамера считывает QR-код с автомобиля, установленного в зоне загрузки-выгрузки разрешающий режим загрузки. Перед выгрузкой автомобиля в зону видимости видеокамеры вносится карточка с QR-кодом, разрешающая режим выдачи.
  3. Подтверждение демонстрируется записями в командной строке команд “Разрешена загрузка”, “Разрешена выгрузка”, “QR-код на автомобиле: A”, “Номер ячейки хранения автомобиля A: N”, “Карточка автомобиля: А”. Символы А - номер автомобиля 1, 2 и т.д., N - номер ячейки 1, 2 и т.д.
  4. После загрузки к зоне загрузки-выгрузки перемещается свободная ячейка по команде “Хранение”. При подаче команды “Разрешена выгрузка” автомобиль должен выехать задним ходом обратно в зону загрузки-выгрузки.
  5. В файле данных отражается время и исполняемые команды: “Заезд”, “Выезд”, “Хранение”, “Перемещение ячеек”.

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

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

Таблица . Используемые компоненты и ПО

|  |  |
| --- | --- |
| **Название компонента и его назначение в проекте** | **Внешний вид** |
| Соленоид JF-1253B  В данном проекте он используется для печати на бумаге рельефно-точечным шрифтом Луи Брайля. |  |
| Шаговый мотор Nema17  Предназначены для перемещения печатной головки по горизонтальной плоскости, листа бумаги по вертикальной плоскости |  |
| Шкив GT2-20  С помощью ременной передачи вращение с шаговых двигателей передаётся на ремень, который приводит в движение печатающую головку принтера |  |
| Концевой переключатель  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. |  |
| Arduino Uno  Микроконтроллер, отвечающий за управление шаговыми моторами и печатью. Весь программный код писался в интегрированной среде разработки Arduino IDE, использующий C-подобный язык программирования. | Arduino Uno R3 10 Pack with cables from PMD Way |
| MotorShield  Плата расширения, предназначенная для управления скоростью и направлению вращения шаговых моторов постоянного тока. |  |
| RaspberryPi 3  Микрокомпьютер, на котором, за счёт облачных сервисов, реализовано распознавание речи.  В данном проекте используются следующие модули: time, requests, RPi.GPIO, soundfile, pyaudio, wave, serial | raspberry-pi-3-model-b.1 |
| САПР «Компас-3D»  Средство автоматического проектирования, в котором создавались 3D модели деталей, чертежи, кинематические схемы |  |
| ArduinoIDE  Интегрированная среда программирования. Предназначена для разработки и загрузки программного кода на Arduino-совместимые платы.  В данном проекте используются следующие библиотеки: AFMotor, TMRpcm |  |

# Функциональное описание разработанного решение в виде UML-диаграмм

* 1. Диаграмма вариантов пользовательского взаимодействия с системой

Нажав на кнопку, 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.

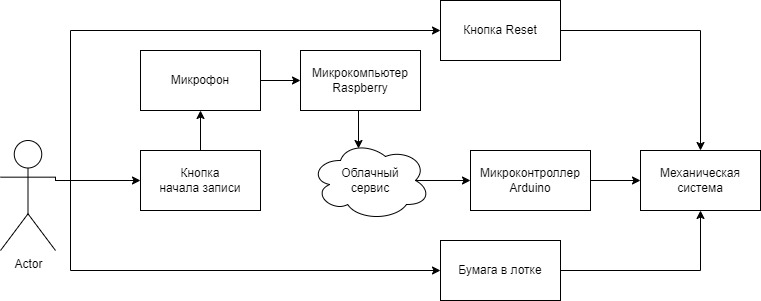


Рисунок . Диаграмма вариантов пользовательского взаимодействия с системой

* 1. Диаграмма автомата

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

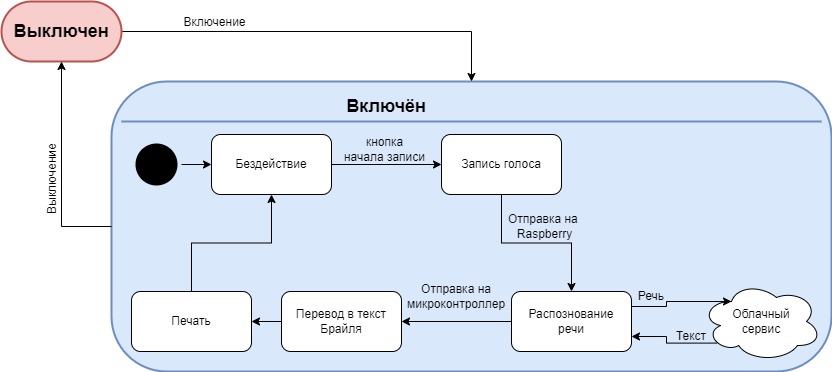


Рисунок 2. Диаграмма автомата

* 1. Диаграмма последовательности

После включения система ждёт, когда пользователь нажмёт на кнопку для начала записи. После записи полученная речь отправляется микрокомпьютером RaspberryPi 3 для распознавания. С помощью облачного сервиса YandexCloud 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.

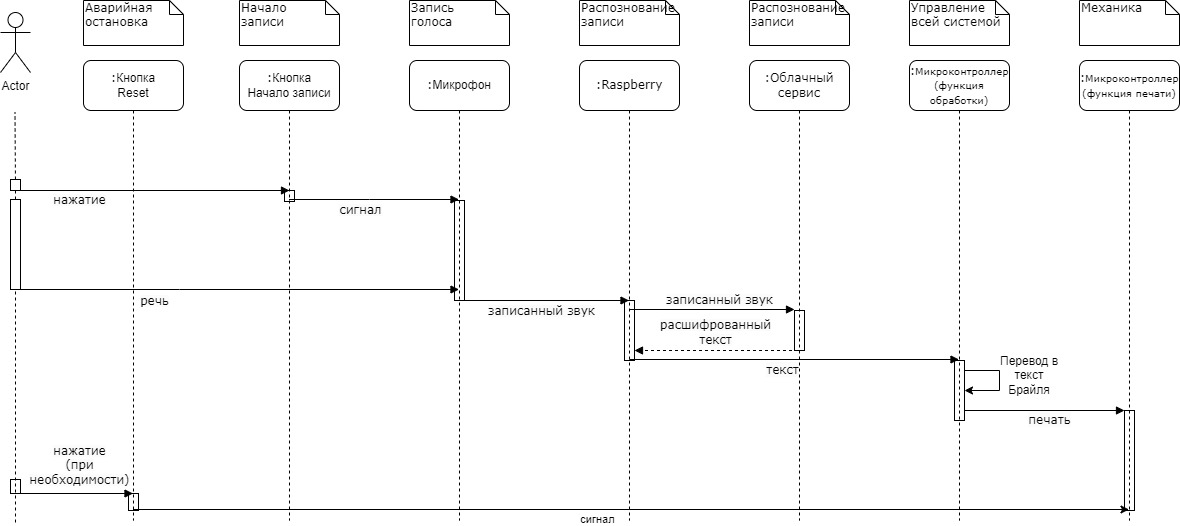


Рисунок 3. Диаграмма последовательности

* 1. Диаграмма компонентов

В нашем проекте используется один микроконтроллер, к которому подключены все составляющие механизма печати. Также мы используем микрокомпьютер RaspberryPi 3 с подключенным к нему микрофоном для распознавания речи. Кнопка 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 RaspberryPi 3, который отправляет её на облачный сервис YandexCloud, где происходит распознавание речи. Полученный таким образом текстовый файл обратно возвращается на микрокомпьютер, который передаёт его микроконтроллер ArduinoUno, где преобразуется по шрифту Брайля и отправляется на печать.

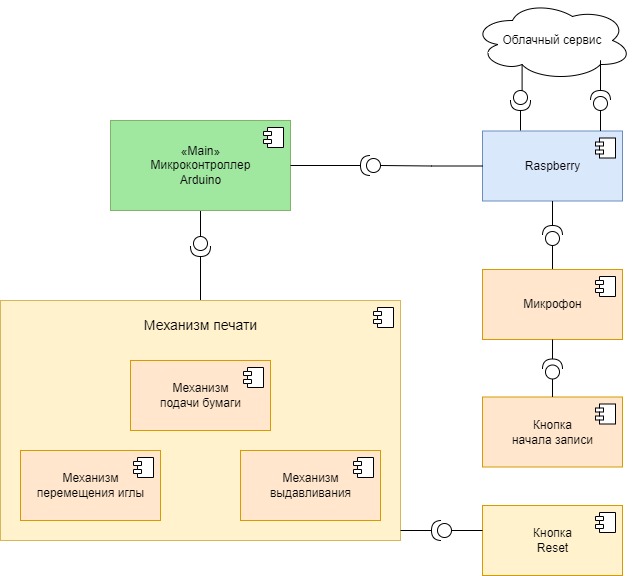
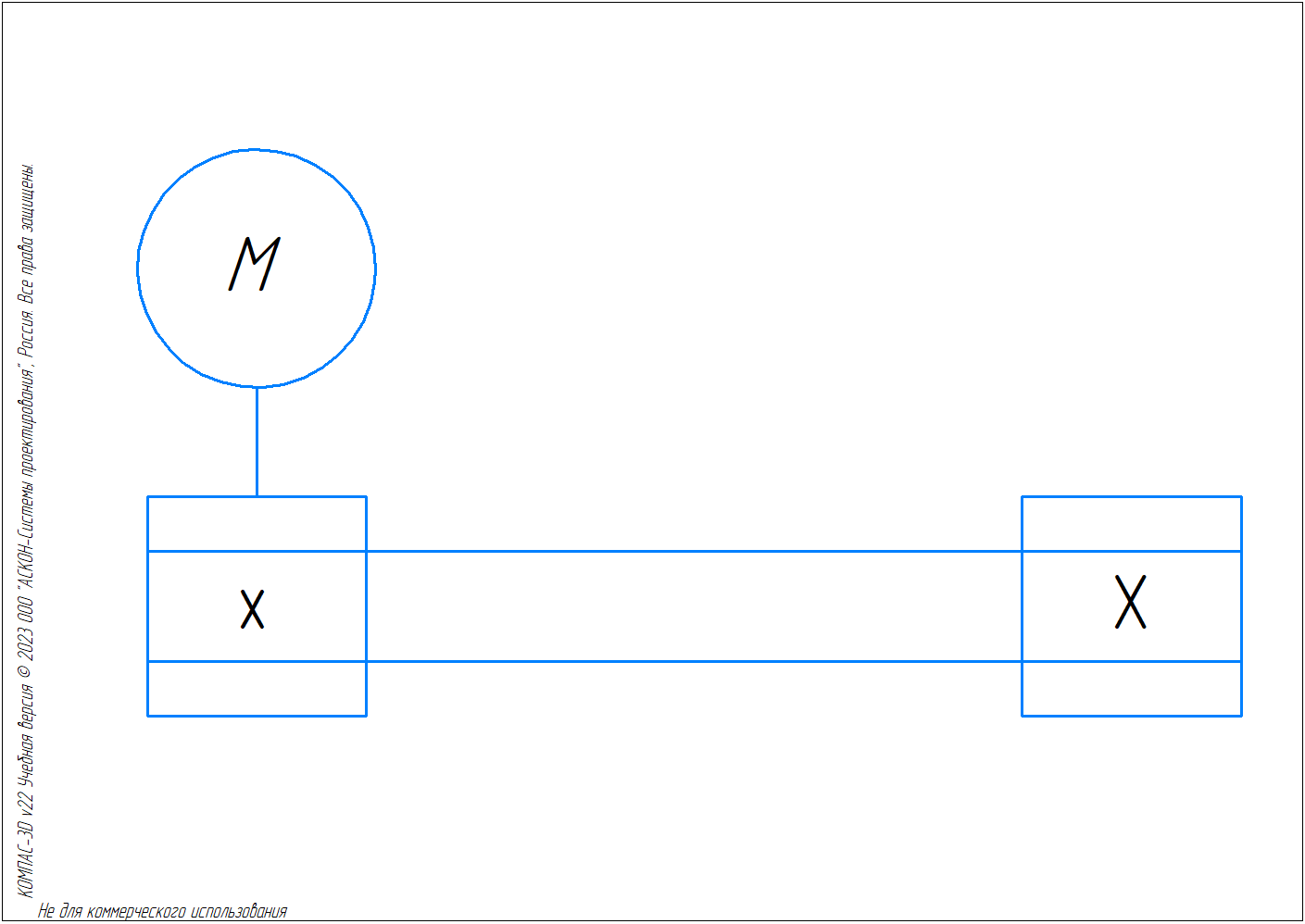


Рисунок 4. Диаграмма компонентов

# 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 . Кинематическая схема печатной головки

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

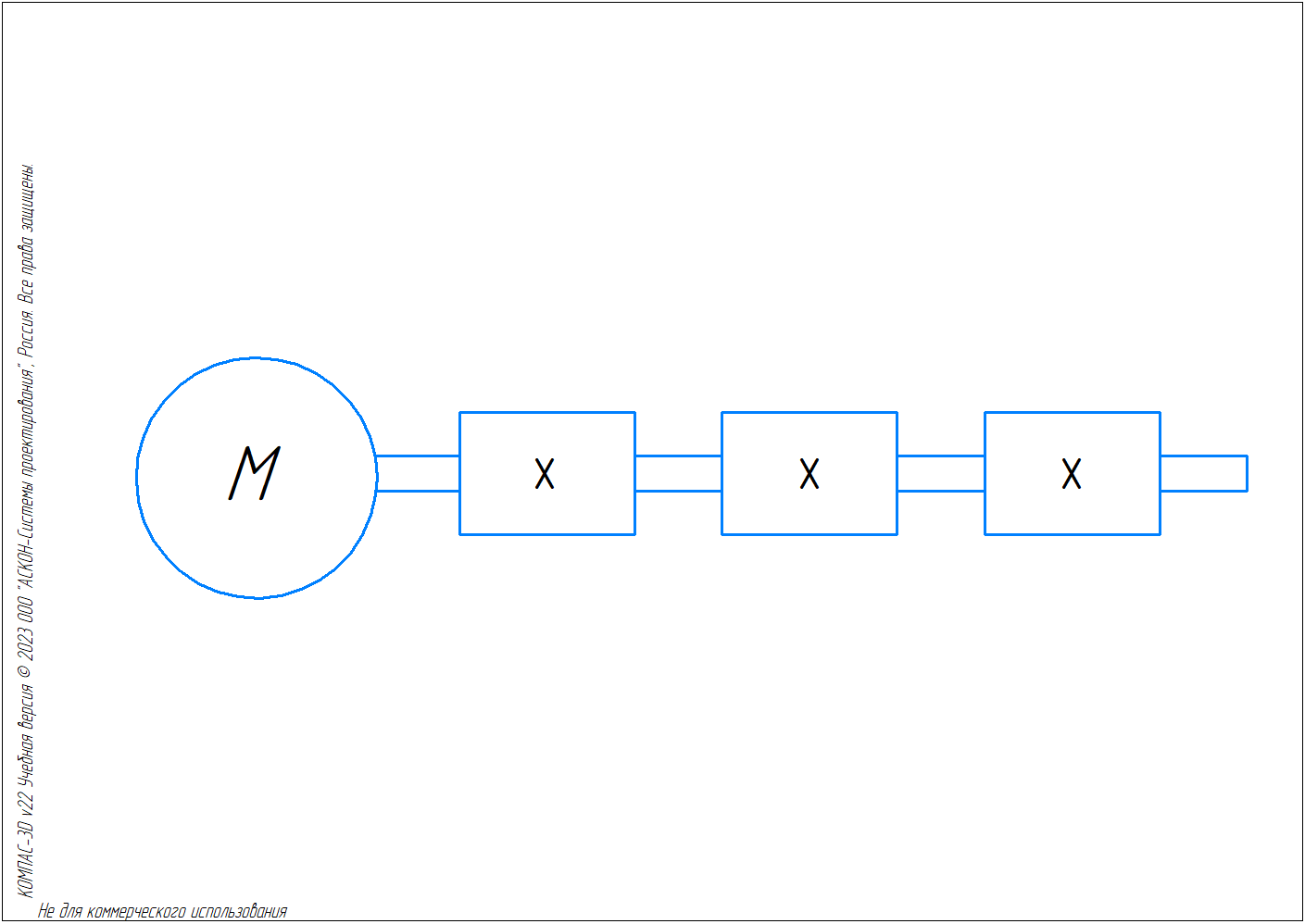


Рисунок 6. Кинематическая схема вала с прижимными роликами

# Скриншоты разработанных 3D-моделей

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

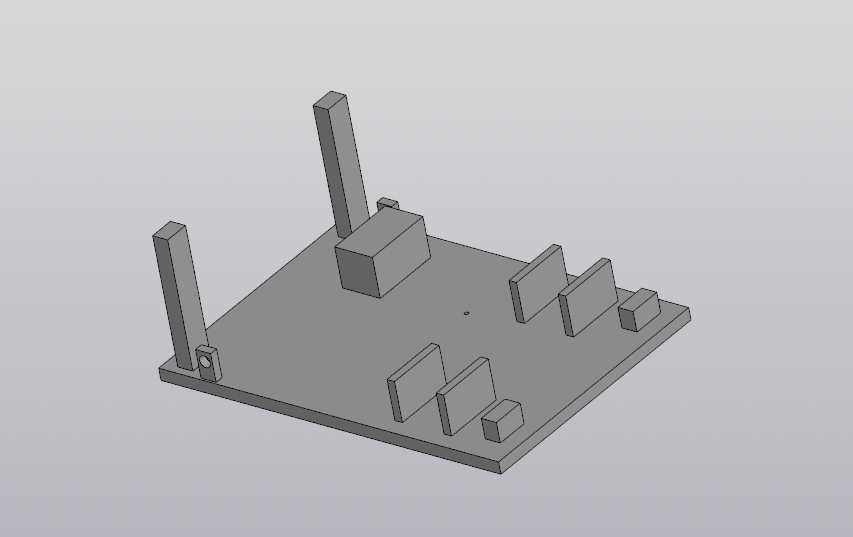


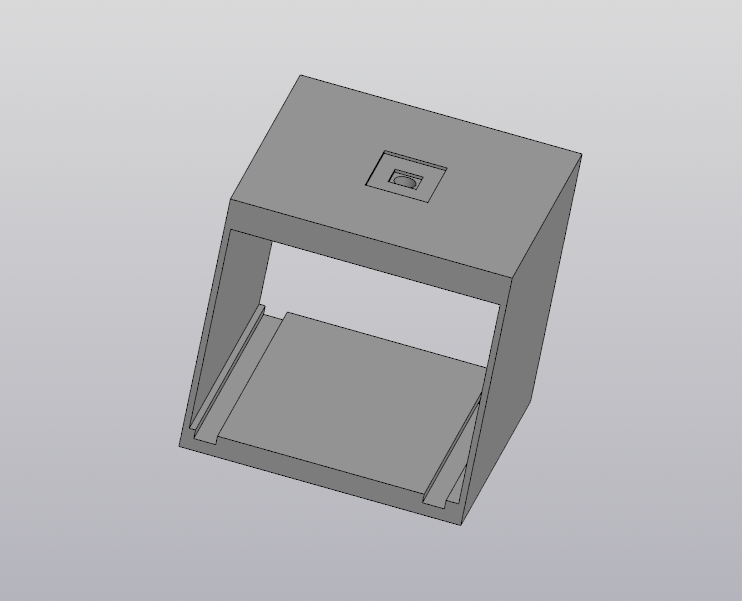
Рисунок 7. Основа машины

Рисунок 8. КПП



Рисунок 9. Колесо 1



Рисунок 10. Колесо 2

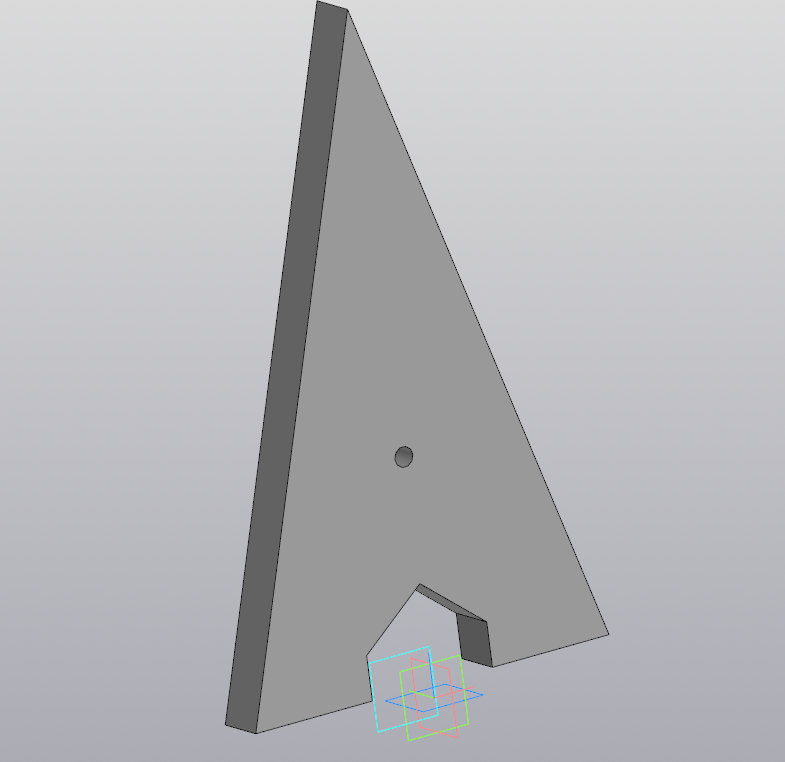


Рисунок 11. Подставка

Рисунок 12. Ячейка для машины

# Описание электротехнической схемы разработанного устройств

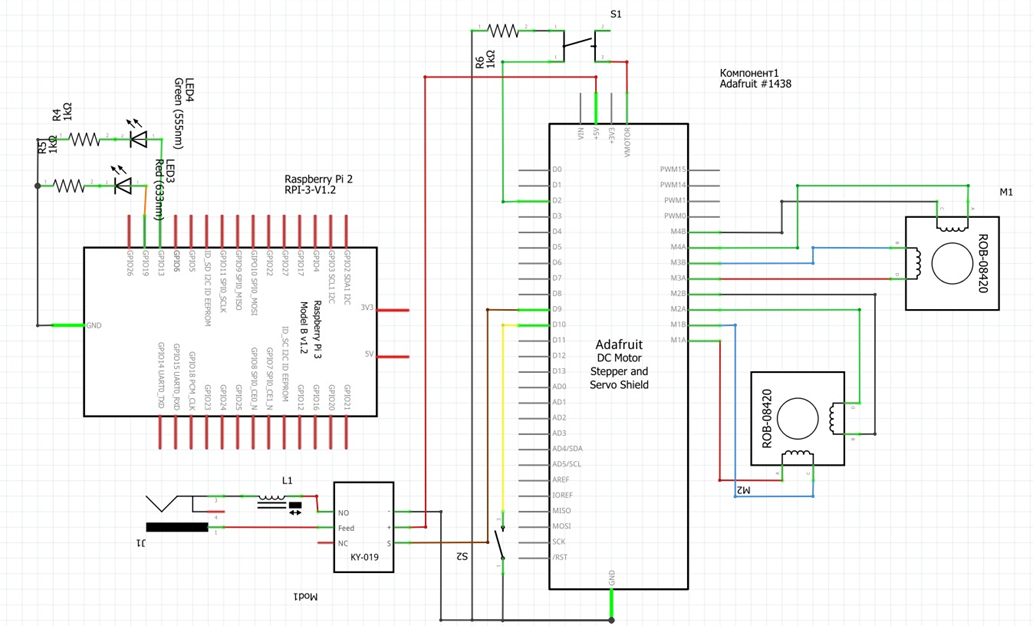


Рисунок 19. Принципиальная электрическая схема

# Алгоритм работы разработанного программного обеспечения в виде блок-схем

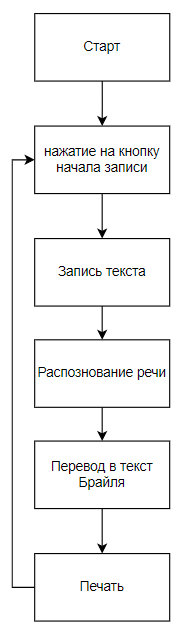


Рисунок 20. Общий алгоритм ПО

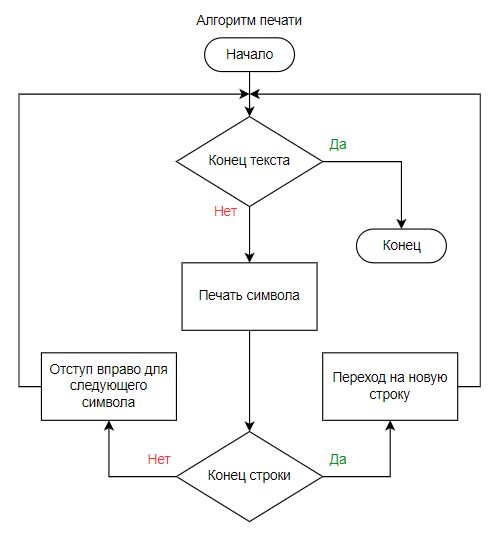


Рисунок 21. Алгоритм печати

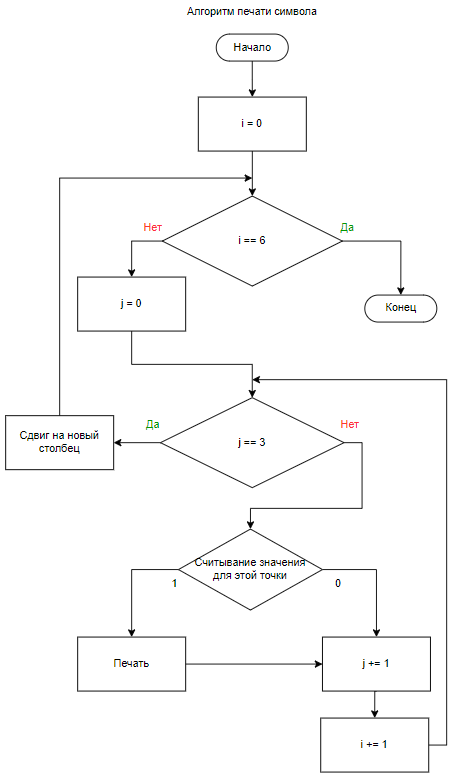


Рисунок 22. Алгоритм печати символа

# Код разработанного программного обеспечения.

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

* [https://github.com/Deimos48/Braille/tree/main/arduino](https://github.com/Deimos48/Braille/tree/main/arduino%20)
* [https://github.com/Deimos48/Braille/tree/main/python](https://github.com/Deimos48/Braille/tree/main/python%20)

# 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/%D0%A4%D0%BE%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%B8_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0)

# 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/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D1%80%D0%BE%D0%BB%D0%B8%D0%BA)

# Заключение

В результате работы был создан принтер Брайля с голосовым управлением, который способен по нажатию кнопки воспринимать речь и переводить ее на бумагу рельефно-точечным шрифтом Луи Брайля, реализована специальная голосовая команда, указывающая на перенос печати на новую строку.

# 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, RaspberryPi 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