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,

Бахчинянц Кристиан Владиславович,

Романов Михаил Алексеевич,

Чикин Максим Алексеевич

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 1. 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 2. Используемые компоненты и ПО

|  |  |
| --- | --- |
| **Название компонента и его назначение в проекте** | **Внешний вид** |
| Шаговый двигатель 28YBJ-48  Используется для вращения колеса с кабинками | Picture background |
| Драйвер Mx1508  Используется для управления моторами | Picture background |
| Esp8266  Используется для управления устройствами удаленно, сбора данных с датчиков и передачи их в облако. | Picture background |
| Мини-мотор с редуктором 3B/200 Используется для вращения колес машины | Picture background |
| Аккумулятор Robiton LGC1500 30A LG 18650HB4 без защиты 15702  Обеспечивает питание для устройств, требующих значительных токов |  |
| Батарейный отсек АА / Держатель для двух батареек  Он служит для подключения двух батареек AA, обеспечивая необходимое напряжение и ток для питания | Picture background |
| использоваться для создания умных устройств и систем, которые взаимодействуют с интернетом. | Picture background |
| 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

Сначала запускается ПАК. Затем запускается ПРА, а остальные подсистемы включены и ожидают. Далее запускается подсистема ППЯ и производится включение тумблера запуска машинки. Карточка с QR-кодом вносится в область видимости камеры, а машинка ждёт сигнала. Затем она получает сигнал и едет к ячейке, а информация с QR-кода и о ячейке вносится в файл.

Рисунок 1. Диаграмма вариантов пользовательского взаимодействия с системой

* 1. Диаграмма автомата

После включения подсистема ПРА запущена, а остальные включены и ожидают. Когда завершается распознание QR-кода, запускается подсистема ППЯ. Машинка переезжает в зону выдачи и карточка с QR-кодом вносится в видимость камеры. В конце в файле отображается информация о QR-коде и о ячейке перемещения машинки.



Рисунок 2. Диаграмма автомата

* 1. Диаграмма последовательности

Изначально нажимается кнопка запуска ПАКа. Затем производится запуск ПРА и запуск ППЯ. Далее одновременно включается тумблер машинки и внос карточки с QR-кодом. Машинка ожидает сигнала на въезд, а затем, когда получила сигнал, едет в ячейку. В конце вся информация с QR-кода и о ячейке расположения машинки передаётся в файл, при этом новая информация обновляется.

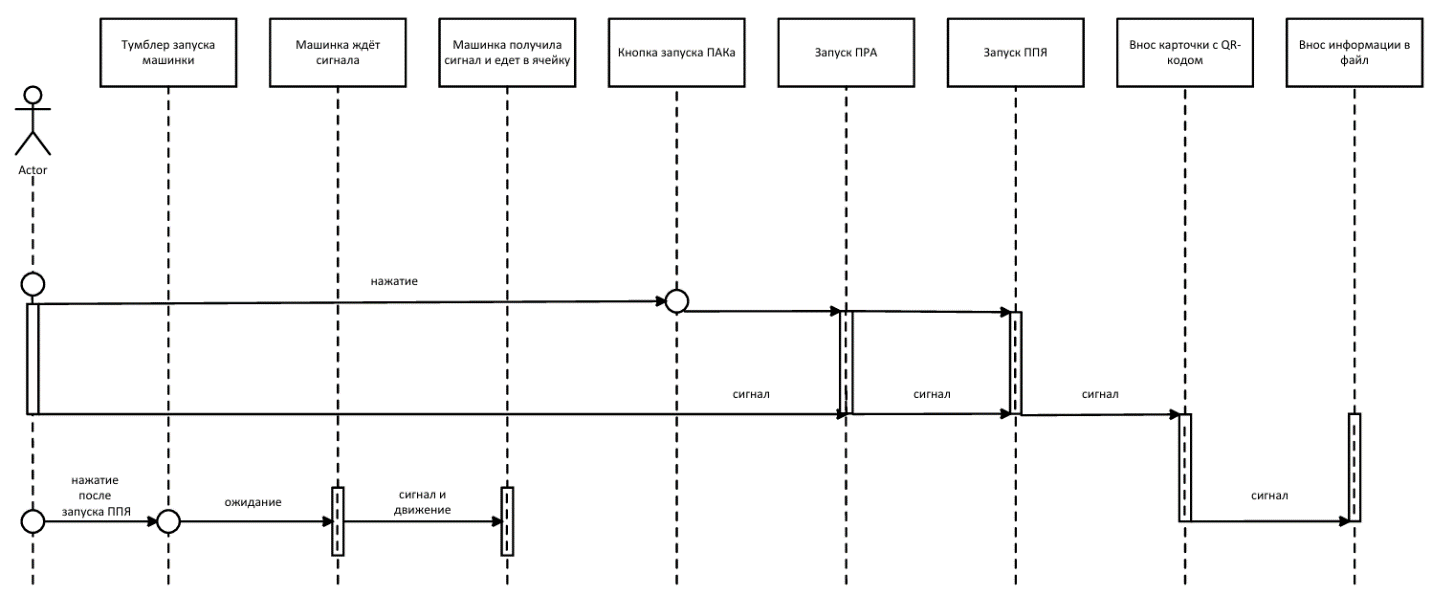


Рисунок 3. Диаграмма последовательности

* 1. Диаграмма компонентов

Две основные части, от которых идёт сигнал к работе, является ESP-модуль и Arduino Nano. Arduino Nano взаимодействует с ESP и с механизмом движения машинки. От ESP идёт подача сигнала к механизму подачи и приёму сигналов машинки. Этот механизм также взаимодействует с механизмом движения, но и отправляет сигналы к механизму приёма машины(МПМ). А МПМ подаёт сигналы в ПАК.

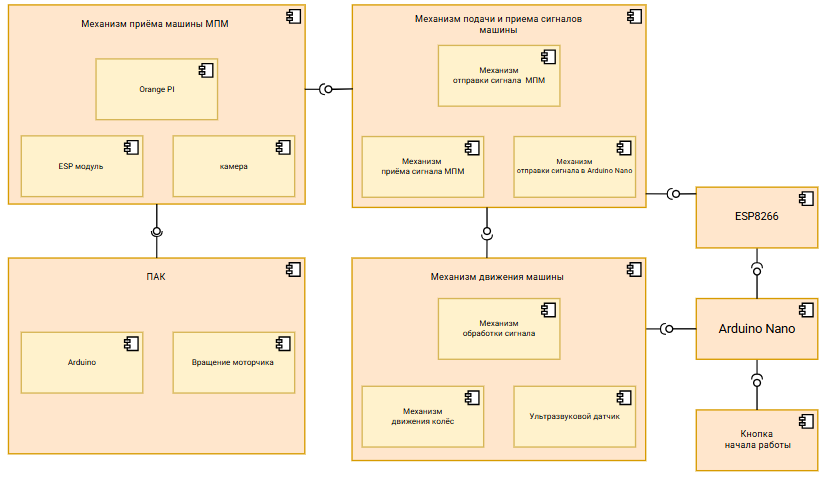
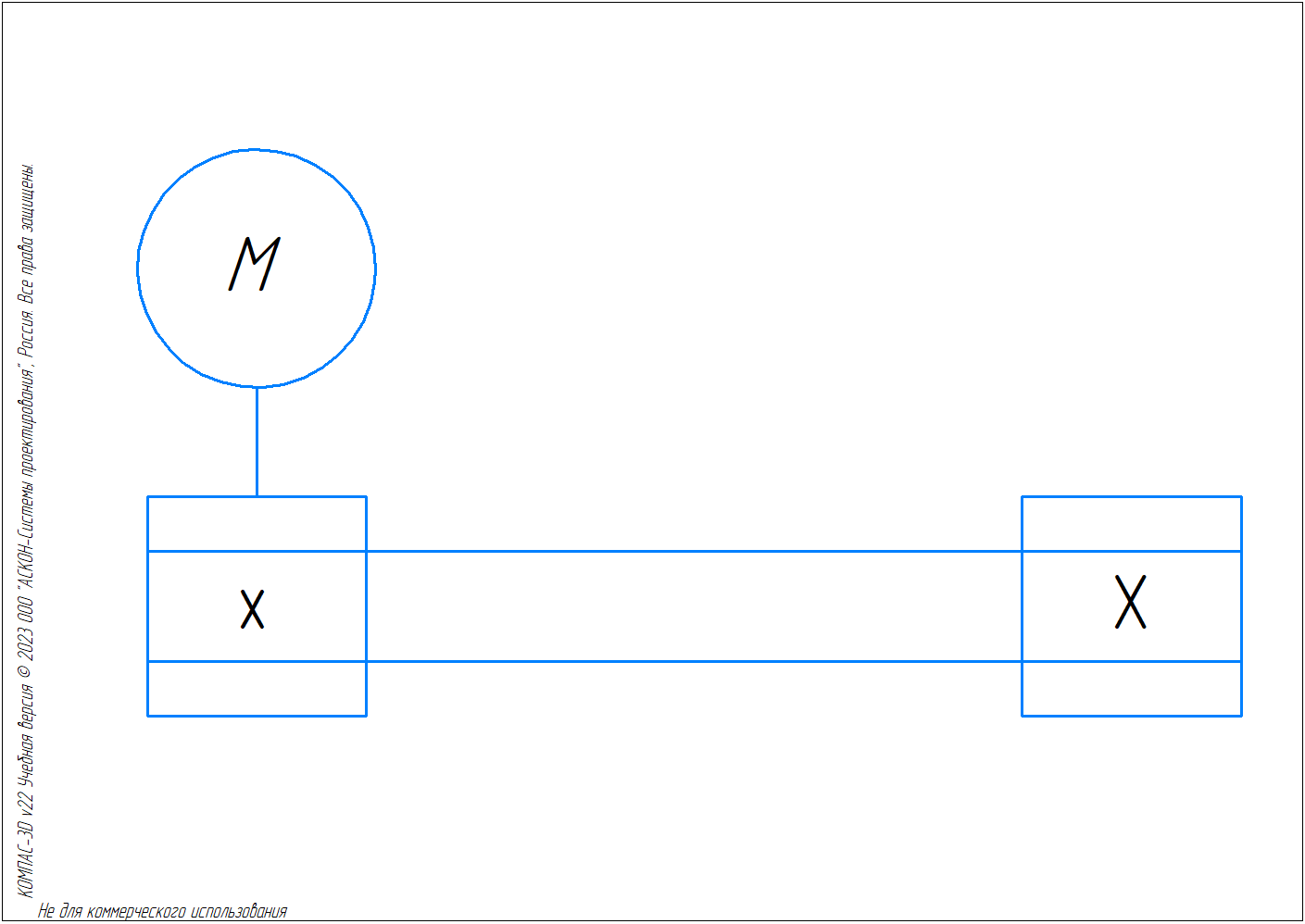


Рисунок 4. Диаграмма компонентов

# 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 5. Кинематическая схема печатной головки

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

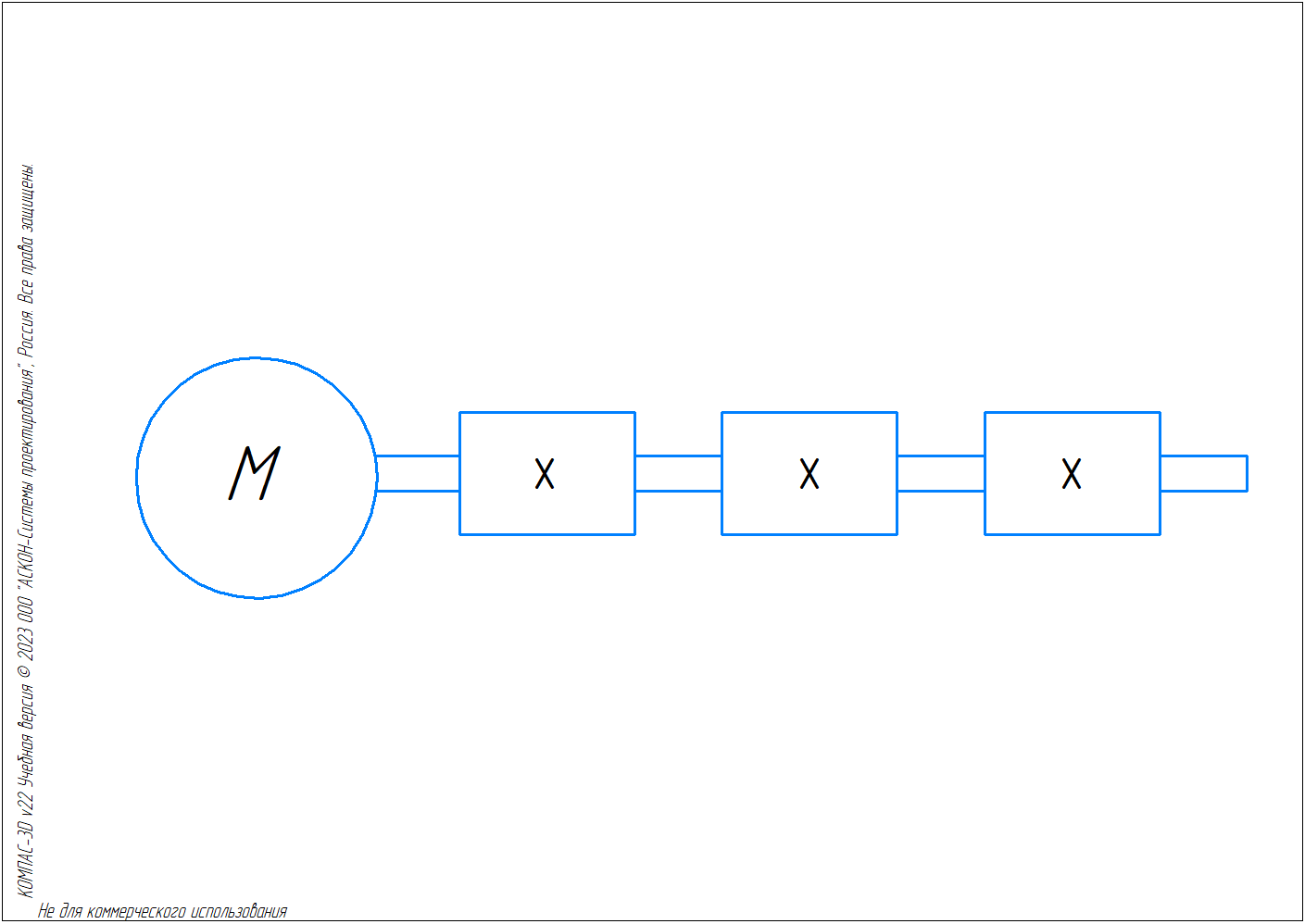


Рисунок 6. Кинематическая схема вала с прижимными роликами

# Скриншоты разработанных 3D-моделей

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

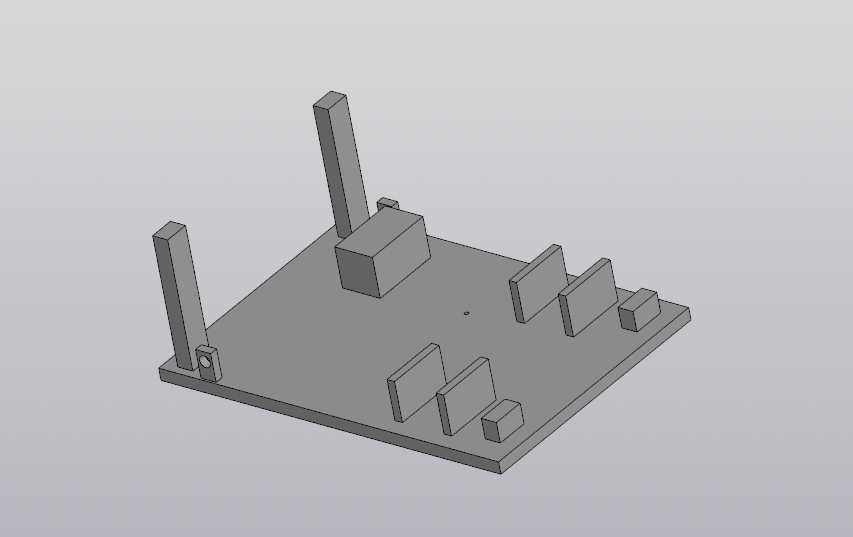


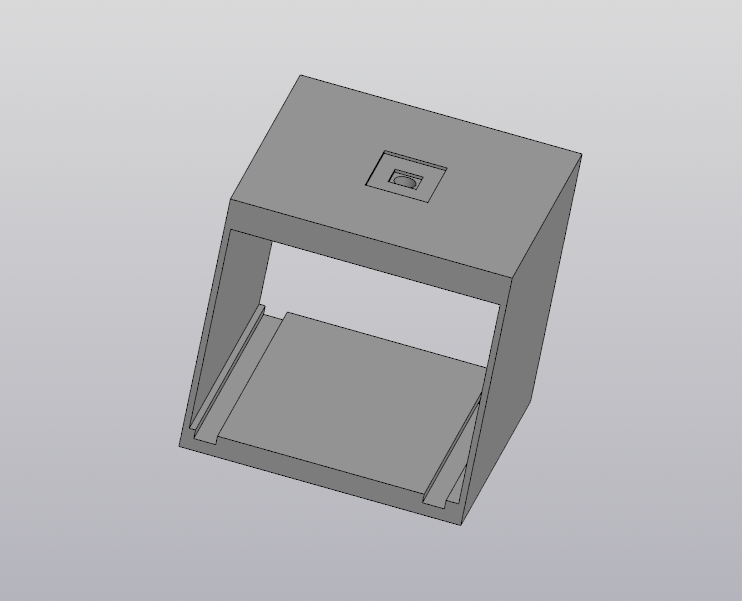
Рисунок 7. Основа машины

Рисунок 8. КПП



Рисунок 9. Колесо 1



Рисунок 10. Колесо 2

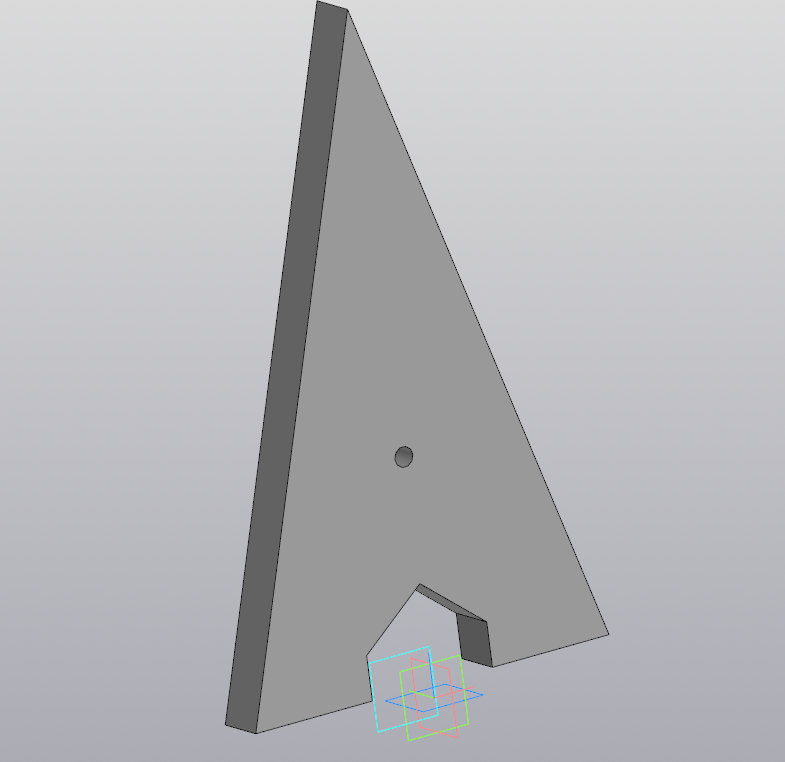


Рисунок 11. Подставка

Рисунок 12. Ячейка для машины

# Описание электротехнической схемы разработанного устройств

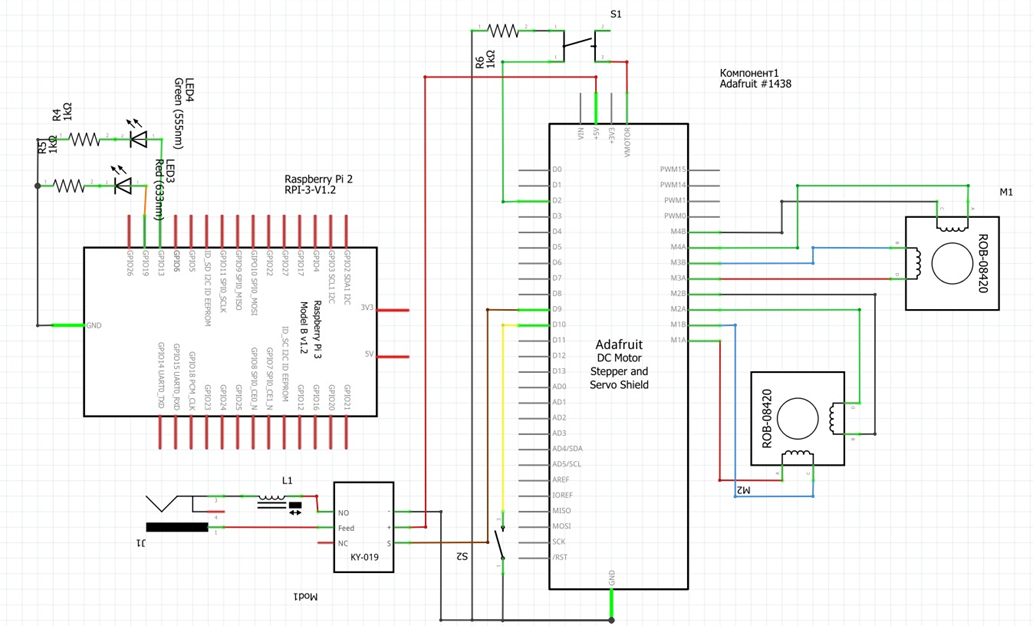


Рисунок 19. Принципиальная электрическая схема

# Алгоритм работы разработанного программного обеспечения в виде блок-схем

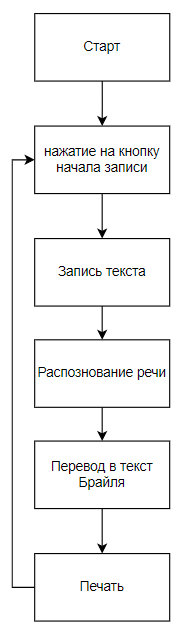


Рисунок 20. Общий алгоритм ПО

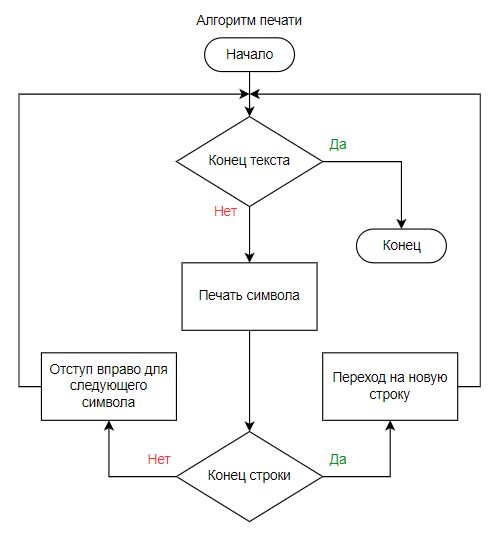


Рисунок 21. Алгоритм печати

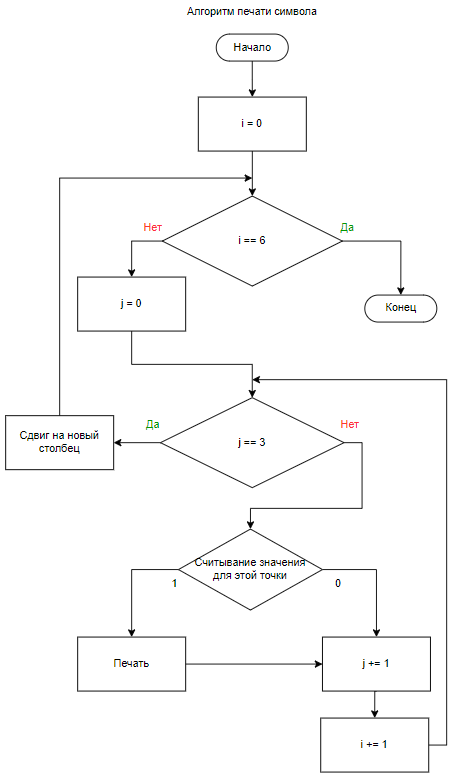


Рисунок 22. Алгоритм печати символа

# 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