



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. Н.Э. БАУМАНА



ОЛИМПИАДЫ ШКОЛЬНИКОВ
«ШАГ В БУДУЩЕЕ»
МГТУ ИМ. Н.Э. БАУМАНА

$$\rightarrow N(\mu, \frac{\sigma^2}{N})$$



КОНФЕРЕНЦИЯ «ШАГ В БУДУЩЕЕ.»

XXVIII НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ МОЛОДЫХ ИССЛЕДОВАТЕЛЕЙ «ШАГ В БУДУЩЕЕ»

СБОРНИК ЛУЧШИХ РАБОТ

$$V(\bar{x}) = V\left(\frac{x_1 + \dots + x_n}{N}\right) = \frac{1}{N} \cdot N \sigma^2 = \frac{\sigma^2}{N}$$

МОСКВА, 2025



ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Андреев Д.О. СИСТЕМА ПОМОЩИ ПРИ МЫТЬЕ РУК ЛИЦАМ С ОВЗ.....	7
Белова Е.С. РАЗРАБОТКА ОБЛАЧНОГО АССИСТЕНТА ДЛЯ ПРЕПОДАВАТЕЛЯ ШКОЛЫ.....	20
Боков В.М. ТЕЛЕГРАММ-БОТ ДЛЯ ТРЕНИРОВОК ПО БЕГУ С ИИ.....	38
Бояринцев А.С. СИСТЕМА ПРЕДВАРИТЕЛЬНОГО СМЕШИВАНИЯ ФИЛАМЕНТА ДЛЯ 3D ПЕЧАТИ.....	65
Бухтиярова Д.А. МОДУЛЬНЫЙ ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ПРОВЕРКИ ЗНАНИЙ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ СКАНТЕСТ.....	94
Гончаренко А.А. ПАРСИНГ И КОДИФИКАЦИЯ КЛЮЧЕВОЙ СТАВКИ.....	114
Дорофеев В.А. ДИСТАНЦИОННО-УПРАВЛЯЕМЫЙ ПОДВОДНЫЙ РОБОТ.....	139
Жук И.В. ШИФРОВАНИЕ МЕТОДОМ ГАММИРОВАНИЯ.....	158
Иванов А.А. VISIOASSIST ИНТЕЛЛЕКТУАЛЬНЫЙ ПОМОЩНИК ДЛЯ СЛАБОВИДЯЩИХ И НЕЗРЯЧИХ.....	188
Курипта Е.К. ПРОТОТИП НАГРЕВАТЕЛЬНОГО КОНЦЕНТРАТОРА СОЛНЕЧНОГО ИЗЛУЧЕНИЯ С ТРАНСФОРМИРУЕМОЙ КОНСТРУКЦИЕЙ РЕФЛЕКТОРА.....	204
Любушин В.М. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ИЗУЧЕНИЯ ЖЕСТОВОГО ЯЗЫКА НА БЕСПЛАТНОЙ ОСНОВЕ.....	237
Меликянц Т.Г. ПРОТЕЗ НОГИ ЧЕЛОВЕКА ИЗ УГЛЕПЛАСТИКА.....	267
Мирославлев И.А. ПРЕДСКАЗАНИЕ ЗНАЧЕНИЯ УДЕЛЬНОГО ИМПУЛЬСА РАКЕТНОГО ДВИГАТЕЛЯ С ПОМОЩЬЮ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ С УЧЕТОМ ВАРЬИРОВАНИЯ ХИМИЧЕСКОГО СОСТАВА ТОПЛИВА.....	295
Михайлов В.С. ОТХОДЫ ОДНОЛЕТНИХ РАСТЕНИЙ В ПРОИЗВОДСТВЕ КОМПОЗИЦИОННЫХ МАТЕРИАЛОВ.....	310
Мухин В.Г. РОБОТ-СПАСАТЕЛЬ «ТОТОШКА» ДЛЯ ПОМОЩИ ПРИ УСТРАНЕНИИ ПОСЛЕДСТВИЙ ЧС....	330
Пономарев Н.К. СРЕДСТВА КИНЕМАТИЧЕСКОЙ ОБРАТНОЙ СВЯЗИ В ИМИТАТОРАХ ПЕРЕГРУЗОК ДЛЯ ТРЕНАЖЁРОВ ПИЛОТИРУЕМЫХ КОСМИЧЕСКИХ КОРАБЛЕЙ.....	353
Руппель М.Д. СПУТНИК ОБНАРУЖЕНИЯ ОПАСНЫХ АСТЕРОИДОВ И КОМЕТ, НАПРАВЛЯЮЩИХСЯ К ЗЕМЛЕ СО СТОРОНЫ СОЛНЦА.....	364
Рюмина М.Н. «ЗАГОР» – ЛЕГКОАТЛЕТИЧЕСКИЙ ИННОВАЦИОННЫЙ МЕТАТЕЛЬНЫЙ ДИСК.....	392
Скворцов А.С. УСТРОЙСТВО КОНТРОЛЯ ИНФУЗИИ.....	421
Хаванов И. РАЗРАБОТКА ПЛАТФОРМЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ДЛЯ ЗАДАЧ ПРОТИВОВОЗДУШНОЙ ОБОРОНЫ.....	433

ДОРОГИЕ ДРУЗЬЯ!

В этот сборник вошли лучшие статьи участников Научно-образовательного соревнования молодых исследователей «Шаг в будущее», проходившего с 17 по 28 марта 2025 года в МГТУ им. Н.Э. Баумана. Статьи подготовлены школьниками в рамках конкурса проектных работ и публикуются в авторской редакции.

Основной задачей научно-образовательного соревнования «Шаг в будущее» является выявление и вовлечение в научно-исследовательскую работу, в сферу инженерного творчества учащихся образовательных учреждений, организация сотрудничества исследователей и ученых разных поколений, создание специальных условий в стенах МГТУ им. Н.Э. Баумана для воспитания профессионально-ориентированной, склонной к научной работе молодежи, осуществление комплексного подхода к оценке интеллектуального и творческого потенциала будущего студента. «Шаг в будущее» является уникальным методом профессиональной подготовки, которая осуществляется круглогодично в научно-исследовательских лабораториях школьников, созданных на базе кафедр Университета и ведущих предприятий России, и индивидуально с преподавателями МГТУ им. Н.Э. Баумана, которые выступают в качестве научных руководителей в течение одного, двух и более лет. Результатом такой подготовки становится научно-исследовательская работа, оформленная в соответствии с требованиями, соизмеримыми с требованиями на курсовые работы и отчеты по НИР, и представляемая для публичной защиты на научных секциях конференции, которые проводятся на кафедрах МГТУ им. Н.Э. Баумана при участии ведущих преподавателей, ученых и специалистов. Научными направлениями конференции являются: Машиностроительные технологии; Информатика и системы управления; Радиооптоэлектроника; Биомедицинская техника; Специальное машиностроение, Энергетика и экология, Автоматизация, робототехника и механика; Инженерный бизнес и менеджмент; Фундаментальные науки; Образовательно-реабилитационные технологии; биологические и химические технологии. В рамках ежегодного научно-образовательного соревнования проводится Выставка - конкурс программных разработок, которая за многие годы своего существования стала традиционным смотром достижений школьников, увлеченных захватывающим процессом создания компьютерных программ, и доказала свою состоятельность оригинальностью и новизной программных продуктов, участвовавших в Выставке и использовавшихся в дальнейшем в научных работах, учебном процессе, коммерческой реализации.

Научно-образовательное соревнование «Шаг в будущее» носит не только научный, образовательный и социальный характер, но представляет собой соревнование, на котором выбираются победители в различных номинациях. Всем участникам вручаются свидетельства участника научно-образовательного соревнования.

В МГТУ им. Н.Э. Баумана в рамках работы Центра довузовской подготовки проводятся следующие мероприятия:

- Олимпиады для школьников 8-11 классов, такие как Олимпиада школьников «Шаг в будущее» по шести профилям, олимпиада по программированию для школьников.
- «ТехноКубок» совместно с VK Company и МФТИ.
- Отраслевая олимпиада школьников «Газпром» совместно с ПАО «Газпром» и другими техническими университетами России.
- Экскурсии на предприятия ОПК проводятся совместно с организациями, сотрудничающими с Университетом по целевой подготовке кадров, на ведущие в своей области предприятий страны, с целью раскрытия связи профильного образования с овладением знаний для получения конкретной профессии и будущей трудовой деятельности. Создание системы взаимодействия между школами и предприятиями через учебные заведения высшего профессионального образования состоит в том, чтобы сформировать у молодежи полное представление об их профессиональном будущем, включая выбор будущей специальности, области деятельности, места работы и учебного заведения для получения соответствующего образования. Учащимся предоставляется возможность посетить музеи предприятий, увидеть производственные помещения, побывать на опытных площадках, а также задать все интересующие вопросы, касающиеся деятельности предприятия, условий работы, особенностей поступления и обучения по целевому набору и многое другое.
- Технический лабораторный практикум – школьным инженерным классам организуется в системе университетского образования для учеников 8-11-х классов с целью интеграции методики и технического оборудования университета в учебный процесс общеобразовательных учреждений; расширения и закрепления у обучающихся базовых знаний школьного курса физики, информатики и математики, получения навыков их использования на практике; приобретения опыта в обращении с техническим оборудованием и

навыков проведения экспериментов, а также популяризации научно-технических знаний. Цикл лабораторных работ выполняется на действующих современных образцах техники и технологических машин 16-ти экспериментальных лабораторий и 5-ти Научно-образовательных центров: Центр управления полетами малых космических аппаратов; Молодежный космический центр; лабораторный комплекс Дома Физики; НУЦ «Робототехника», НИЦ «Биомедицинская техника». Лабораторные работы сгруппированы по 5-ти разделам физики (Физика твердого тела, Термодинамика и Молекулярно-кинетическая теория, Электродинамика, Оптика, Механика), информатике и математике.

- Летняя практика: Организация и проведение летней научно-образовательной практики для обучающихся 8-10 классов общеобразовательных организаций города Москвы с инженерными классами на базе научно-образовательных центров, Военного института, лабораторий и кафедр следующих факультетов: Машиностроительные технологии, Специальное машиностроение, Энергомашиностроение, Робототехника и комплексная автоматизация, Фундаментальные науки, Радиоэлектроника и лазерная техника, Информатика и системы управления, Биомедицинская техника, Инженерный бизнес и менеджмент МГТУ им. Н.Э. Баумана. Каждому школьнику предоставляется возможность прохождения практики по индивидуальной образовательной траектории, которую он выбирает из вышеназванных подразделений Университета в соответствии с собственными предпочтениями.
- «Ресурсный центр политехнического образования» начал свою работу в 2018-2019 учебном году для учащихся 8-11 классов общеобразовательных организаций города Москвы с инженерными классами, в рамках которого проходят политехнические кружки по четырем научным направлениям: Занимательная электроника; Колесные машины; Инструментальная техника и технологии; Занимательная робототехника. На основании полученных знаний учащиеся могут выполнять самостоятельную проектную работу и представлять результаты своей деятельности на научных конференциях вузовского и городского уровня с участием ведущих специалистов в области техники и технологий.
- Проект «Университетские субботы» проводится для вовлеченности в технические специальности более широкого круга школьников 8-11 классов и

учащихся колледжей, техникумов, используя потенциал системы высшего профессионального образования. Обеспечивается управляемый доступ школьников к научно-техническим ресурсам университета и работе профессиональных научных коллективов. Интеграция основного и дополнительного образования создает благоприятные условия для взаимодействия педагогов разных специальностей, формирования у подростков и старшеклассников целостной картины мира, комплексного педагогического влияния на личность школьника. В проекте проводятся следующие мероприятия: лекции, практические занятия, творческие занятия, мастер-классы по следующим направлениям: техника и технологии, экономика и управление, физико-математические науки, общественные и гуманитарные науки.

Такая система непрерывных профессиональных образовательных технологий «школа-вуз-предприятие» направлена на решение проблем, стоящих перед современной системой образования:

- устойчивая мотивация к приобретению профессиональных знаний в выбранной области, основанная на практической деятельности под руководством преподавателей вузов;
- ранняя активная и осознанная профессиональная ориентация;
- развитие сопровождающей научно-исследовательской и профессиональной подготовки в общеобразовательных учреждениях;
- разработка новых методик и новых обучающих технологий для развития современного образования.

Организаторы программы и составители данного сборника надеются, что такая форма работы заинтересует учителей и преподавателей учебных заведений Москвы и Московской области и поможет успешному развитию творческих возможностей довузовской молодежи в сфере научно-технической деятельности.

СИСТЕМА ПОМОЩИ ПРИ МЫТЬЕ РУК ЛИЦАМ С ОВЗ

Андреев Дмитрий Олегович

Москва г, ГБОУ Школа №2006, 8 класс

Секция: Интеллектуальные системы и робототехника

Научный руководитель: Поленов Дмитрий Юрьевич, к.т.н., доцент кафедры К1

МФ МГТУ им. Н.Э. Баумана

Аннотация

Цель проекта: автоматическая подача чистящего средства, воды для мытья рук и полотенца для лиц с ОВЗ. Система оборудована голосовым и визуальным помощником, смесителем для раковины, насосами подачи воды и чистящего средства для рук, робототехнической рукой для подачи полотенца. Работа системы начинается в случае обнаружения ею человека в зоне действия датчика расстояния, который также входит в систему. Отображение команд для лиц с ОВЗ происходит на дисплее, голосовой помощник реализован с помощью динамика, воспроизводящего записанные команды. Система работает от блока питания, преобразующего входное напряжение 220 В в выходное 5 В.

ОБОСНОВАНИЕ ТЕМЫ ПРОЕКТА

Я с большим удовольствием занимаюсь созданием электронных устройств, приборов и систем. Люблю собирать электронные схемы, программировать Arduino, проектировать в 3D и т.п. Надеюсь, что у меня получится поступить в технический ВУЗ, и я стану инженером. Посещая с родителями места, где скапливается большое количество людей – торговые центры, гипермаркеты и др., обратил внимание, что лицам с нарушением зрения бывает неудобно посещать туалетные комнаты, мыть руки из-за сложности в поиске раковины. Поэтому я решил создать макет системы, которая будет им помогать в этом, казалось бы легком для полноценного человека, деле звуковое сопровождение и подсказ человеку с ОВЗ, когда ему требуется найти раковину и помыть руки.

Цель проекта – создание системы помогающей людям с нарушением зрения мыть руки.

Задачи проекта:

1. Подбор необходимых электронных компонентов для работы.
2. Изучение принципа работы главных компонентов.
3. Разработка электронной схемы соединяющей все электрические компоненты.
4. Создание программного кода.
5. Проверка работы электронной схемы.
6. Создание корпуса для системы помощи.

УСТРОЙСТВО И ПРИНЦИП РАБОТЫ СИСТЕМЫ

Алгоритм работы системы

1. Если человек находится далеко от раковины, звучит команда "подойдите ближе".
2. Когда человек уже подошел, система помощи здороваётся с ним и выполняет череду команд с озвучиванием своих действий.
3. Далее, следуют звуковые команды для человека: 1. Льётся вода 2. Льётся мыло 3. Льётся вода 4. Подается салфетка.
4. Человек, помывший руки и высушивший их, уходит.

Разработка аппаратно-программной части проекта

Код программы написан на языке Си в среде программирования Arduino IDE. Arduino IDE — интегрированная среда разработки для Windows, macOS и Linux, разработанная на Си и Си++, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей.

Исходный код для среды выпущен под общедоступной лицензией GNU версии 2. Поддерживает языки Си и Си++ с использованием специальных правил структурирования кода. Arduino IDE предоставляет библиотеку программного обеспечения из проекта Wiring, которая предоставляет множество общих процедур ввода и вывода. Для написанного пользователем кода требуются только две базовые функции для запуска эскиза и основного цикла программы, которые скомпилированы и связаны с заглушкой программы `main ()` в исполняемую циклическую программу с цепочкой инструментов GNU, также включённой в дистрибутив IDE. Использует программу `avrdude` для преобразования исполняемого кода в текстовый файл в шестнадцатеричной кодировке, который загружается в плату Arduino программой-загрузчиком во встроенном программном обеспечении платы.

Чтобы написать код, я изучил, как работает каждый из электронных компонентов, написал программный код для каждого, и после этого соединил все в одно целое.

Листинг разработанной мною программы приведен в Приложении 1.

Выбор электронных компонентов

Делать свой проект я буду на базе Arduino плата `nano`.

Чтобы узнавать расстояние человека от раковины нам понадобится датчик расстояния, для воспроизведения звука понадобится `mp3-player`, а для качания воды и мыла нужны 2 насоса которые подключаются через реле. Чтобы в конце подавать сухую салфетку человеку надо сделать робо-руку работающую на сервоприводах.

Перечень электронных компонентов системы:

- 1) Плата Arduino `nano` – 1 шт.;
- 2) Сервоприводы – 3 шт.;
- 3) Датчик расстояния HC-SR04 – 1 шт.;
- 4) Динамик – 1 шт.;
- 5) Плата `mp3-модуля` – 1 шт.;
- 6) Насос – 2 шт.;
- 7) Реле – 2 шт.;
- 8) Блок питания на 5 Вольт – 1 шт.

Основные компоненты представлены на рисунках 2 – 6 (плата Arduino `nano`, Сервопривод, Датчик расстояния, Динамик, Плата `mp3-модуля`, Реле).

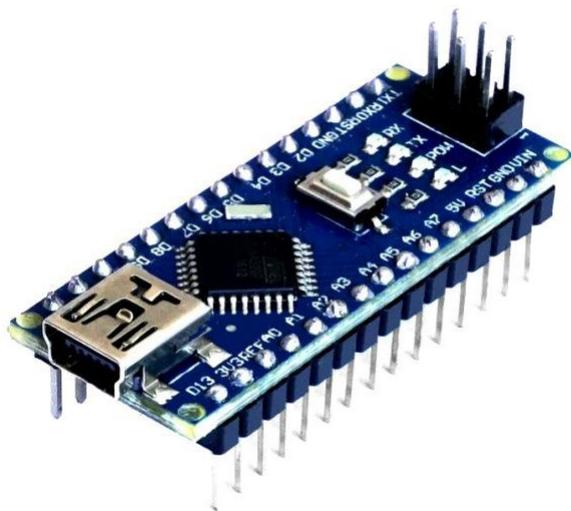


Рис. 2. Плата Arduino

Arduino – это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Устройство программируется через USB без использования программаторов.

Arduino позволяет компьютеру выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.



Рис. 3. Сервопривод



Рис. 4. Датчик расстояния



Рис. 5. Динамик



Рис. 6. Плата mp3-модуля

Структурная схема системы

Структурная схема системы представлена на рис. 7.

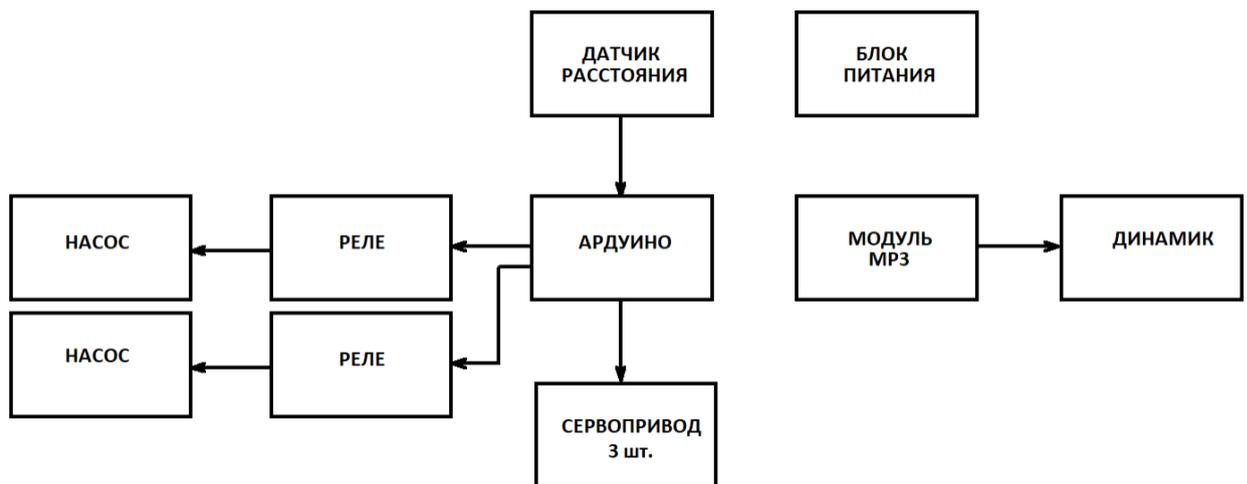


Рис. 7. Структурная схема устройства

Ядром системы является плата Arduino nano. Она опрашивает датчик расстояния. Если он показывает малое расстояние до человека (<20 см), это обозначает приближение человека. Начинает работать голосовая помощь и включение исполнительных устройств для мытья

рук. Arduino последовательно отдает команды на модуль mp3, который, в свою очередь, включает тот или иной музыкальный «трэк» для воспроизведения на динамике. Параллельно с этим поочередно включаются реле, управляющие двумя насосами – для ополаскивания рук и подачи жидкого мыла. Для вытирания рук после мытья, человеку подается сухая салфетка – эта функция реализована на базе робо-руки, управляющейся с помощью трех сервоприводов. Электрическое питание системы реализовано с помощью блока питания на 5 Вольт и 2 Ампера.

ЭКОНОМИЧЕСКАЯ ОЦЕНКА ИЗДЕЛИЯ

На изготовление макета системы ушло полгода.

Сумма затрат на электронные компоненты, составные части и расходные материалы составило 5350 руб. В таблице 1 перечислены все составляющие системы и их ориентировочная стоимость.

Таблица 1. Составляющие системы и их ориентировочная стоимость

Название элемента	Цена, руб
Ардуино нано	400
Датчик расстояния	200
Динамик	150
Плата mp3	200
SD-карта	100
Сервоприводы	500
Блок питания	500
Расходные материалы	1000
Насосы	500
Реле	300
Бачок	500
Раковина	800
Задняя панель ДСП	200
	Итого: 5350

СОЗДАНИЕ, СБОРКА И ИСПЫТАНИЯ ИЗДЕЛИЯ

На рисунках 8 – 11 показан процесс сборки системы и ее конечный вид.



Рис. 8. Сборка макета системы

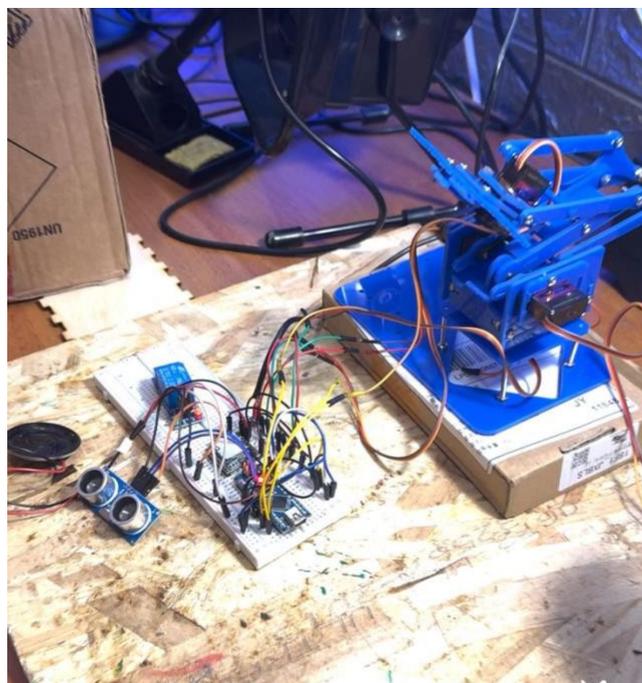


Рис. 9. Макетирование, отработка программного обеспечения управления робо-рукой



Рис. 10. Макет системы в сборе, проведение испытаний с водой



Рис. 11. Готовое изделие с раковиной, установленной системой, бачком с установленными в него емкостями для воды и мыла

После проделанных этапов макет системы получился работоспособным.

ЗАКЛЮЧЕНИЕ

При создании системы помощи проведены следующие работы:

1. Подобраны необходимые электронные компоненты: плата Arduino, mp3 плеер, реле, насосы, динамик.
2. Собрана робо-рука с сервоприводами.

3. Разработано программное обеспечение.
4. Создана электрическая схема подключения компонентов.
5. Все электрические части вложены в корпус.

В ходе работы над проектом я узнал много нового об Arduino. Научился лучше понимать принципиальные и структурные схемы. И усовершенствовал свои знания во многом другом. Созданная мной система помощи уже может использоваться людьми с ограниченными возможностями зрения.

СПИСОК ЛИТЕРАТУРЫ

1. Изучаем Arduino: инструменты и методы технического волшебства: Пер. с англ.— СПб.: БХВ-Петербург, 2017. — 336 с.: ил.
2. Проекты с использованием контроллера Arduino. — 4-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2021. — 560 с.: ил.
3. Практическая энциклопедия Arduino. Петин В. А., Биняковский А. А. – М: ДМК Пресс, 2017г – 152 с.

ПРИЛОЖЕНИЕ 1

Листинг кода программы.

```
#include <SoftwareSerial.h>

#include <DFPlayer_Mini_Mp3.h>

SoftwareSerial mySoftwareSerial(4, 5); // RX, TX для плеера DFPlayer Mini #include
<NewPing.h>

#define PIN_TRIG 7

#define PIN_ECHO 6

#define MAX_DISTANCE 200 // Константа для определения максимального расстояния,
которое мы будем считать корректным

NewPing sonar(PIN_TRIG, PIN_ECHO, MAX_DISTANCE);

#include "Servo.h"

Servo servo;//низ Servo

servo1;//прав Servo
```

```
servo2;//верх
int angle; // переменная для угла поворота микро серво

void setup () {

Serial.begin (9600);
mySoftwareSerial.begin(9600);

mp3_set_serial (mySoftwareSerial);           //set Serial for DFPlayer-mini mp3
module

delay (100);

mp3_set_volume (20);

pinMode(2, OUTPUT); // Объявляем пин реле как выход digitalWrite(2, 1);
pinMode(8, OUTPUT); pinMode(8, 0);
pinMode(8, 1);

// Выключаем реле - посылаем высокий сигнал servo.attach(11); // пин для
подключения микро серво

servo1.attach(10); // пин для подключения микро серво

servo2.attach(9); // пин для подключения микро серво

}

void loop () {

// Стартовая задержка, необходимая для корректной работы. delay(500);

// Получаем значение от датчика расстояния и сохраняем его в переменную unsigned int
distance = sonar.ping_cm();
```

```
// Печатаем расстояние в мониторе порта
Serial.print(distance);
Serial.println("см");
if (distance <=20){
delay (100);

mp3_play (1); // Здравствуйте delay
(7000);
mp3_play (7); //подставьте руки delay (5000);
mp3_play (2); // Вода delay (500);
digitalWrite(2, LOW); // Включаем реле - посылаем низкий уровень сигнала
delay(5000);

digitalWrite(2, HIGH); // Отключаем реле - посылаем высокий уровень сигнала
delay(900);

mp3_play (3); // Мыло delay (900);
digitalWrite(3, 0); // Включаем реле - посылаем низкий уровень сигнала delay(3000);
digitalWrite(3, 1); // Отключаем реле - посылаем высокий уровень сигнала mp3_play (2); //
Вода
delay (1000);

digitalWrite(2, LOW); // Включаем реле - посылаем низкий уровень сигнала
delay(7000);
digitalWrite(2, HIGH); // Отключаем реле - посылаем высокий уровень сигнала

delay (2000);

mp3_play (4); // Салфетка
for (angle = 0; angle <= 90; angle++) {
servo.write(angle); // сообщаем микро серво угол поворота delay(20); // ждем 20
мс, пока серво не сделает поворот
}
for (angle = 50; angle <= 110; angle++) {
```

```
servo1.write(angle); // сообщаем микро серво угол поворота delay(20); // ждем 20
мс, пока серво не сделает поворот
}

for (angle = 0; angle <= 180; angle++) {

servo2.write(angle); // сообщаем микро серво угол поворота delay(20); // ждем 20
мс, пока серво не сделает поворот
}

for (angle = 110; angle >= 50; angle--) {

servo1.write(angle); // сообщаем микро серво угол поворота delay(20); // ждем 20
мс, пока серво не сделает поворот
}

for (angle = 90; angle >= 0; angle--) {

servo.write(angle); // сообщаем микро серво угол поворота delay(10); // ждем 20
мс, пока серво не сделает поворот
}

for (angle = 50; angle <= 110; angle++) {

servo1.write(angle); // сообщаем микро серво угол поворота delay(20); // ждем 20
мс, пока серво не сделает поворот
}

for (angle = 180; angle >= 0; angle--) {

servo2.write(angle); // сообщаем микро серво угол поворота delay(20); // ждем 20
мс, пока серво не сделает поворот
}

for (angle = 110; angle >= 50; angle--) {

servo1.write(angle); // сообщаем микро серво угол поворота delay(20); // ждем 20
мс, пока серво не сделает поворот
}

mp3_play (5); // До свидания
```

```
}
```

```
else if(distance>20){
```

```
mp3_play (6); // Подойдите ближе delay(3000);
```

```
}
```

```
}
```

РАЗРАБОТКА ОБЛАЧНОГО АССИСТЕНТА ДЛЯ ПРЕПОДАВАТЕЛЯ ШКОЛЫ

Белова Екатерина Сергеевна

Калуга г, ГКОУКО «Калужская школа-интернат № 5 имени Ф.А. Рау», 10 класс

Секция Технические системы и средства

Научный руководитель: Гагарин Юрий Евгеньевич, к.т.н., зав. каф. ИУК4

МГТУ им. Н.Э. Баумана

Аннотация

В проекте рассматривается создание облачного ассистента для преподавателя школы. В настоящее время, преподаватели школ состоят в нескольких группах мессенджера Телеграм, таких как: Министерство образования, администрация школы, группа учителей школы, родительская группа, группа учеников класса, поэтому является актуальным создание облачного ассистента, который позволит эффективно и безопасно хранить, просматривать, редактировать получаемую из разных групп информацию. **Целью** данного проекта является создание удобного и эффективного бота для автоматизации процесса переноса файлов из мессенджера Телеграм в различные облачные хранилища. В процессе реализации проекта были решены следующие задачи: изучение основ работы с телеграм-ботами и их интеграции с внешними сервисами; изучение и анализ библиотек языка программирования; разработка интуитивно понятного интерфейса для взаимодействия с ботом; тестирование бота.

Для реализации Telegram-бота использовался язык программирования python, для работы с Telegram использовалась библиотека Telebot, в качестве среды разработки для программирования использовался PyCharm. В проекте приведен результат опроса среди учителей, направленный на выявление уровня цифровой грамотности педагогов, их предпочтений в использовании мессенджеров и отношения к внедрению искусственного интеллекта и показана возможность дальнейшего развития данного проекта.

Введение

Современные технологии развиваются со стремительной скоростью. В цифровую эпоху преподаватели получают доступ к множеству инструментов, упрощающих их работу и делающих взаимодействие с обучающимися более продуктивным. Одним из таких инструментов является облачный ассистент — бот, который обеспечивает легкий перенос файлов из Telegram в облачное хранилище.

Как правило, преподаватели школ состоят в нескольких группах мессенджера Телеграм, таких как: Министерство образования, администрация школы, группа учителей школы, родительская группа, группа учеников класса. Поэтому является актуальным создание облачного ассистента, который позволит эффективно и безопасно хранить, просматривать редактировать получаемую из разных групп информацию.

Удобство и доступность

Одним из главных достоинств ассистента является простота использования и анализа информации. Преподаватели могут без труда перемещать необходимые документы в облако, обеспечивая их организованное хранение. Это помогает избежать беспорядка в коммуникации и сосредоточиться на более важных задачах.

Повышенная безопасность данных

Безопасность данных актуальна постоянно, возможность хранить информацию в защищенном облаке является важным фактором. Облачный ассистент обеспечивает высокий уровень защиты благодаря использованию передовых технологий шифрования.

Экономия времени

Искусственный интеллект, встроенный в бота, помогает экономить время преподавателя, автоматически выбирая важные файлы и предоставляя краткое содержание чатов. Это позволяет сократить время, затрачиваемое на обработку информации, и сосредоточиться на более значимых задачах.

Целью данного проекта является создание удобного и эффективного бота для автоматизации процесса переноса файлов из мессенджера Телеграм в различные облачные хранилища.

Задачи:

1. Изучение основ работы с телеграм-ботами и их интеграции с внешними сервисами.
2. Изучение и анализ библиотек языка программирования

3. Разработка интуитивно понятного интерфейса для взаимодействия с ботом.
4. Тестирование бота.

Гипотеза.

Создание телеграмм-бота, который автоматически переносит файлы из чатов мессенджера в облачное хранилище (например, Яндекс диск, Mail облако и т.д.), что позволит значительно сократить время преподавателя на управление и организацию учебных материалов. Предполагается, что благодаря этому простому и эффективному решению, преподаватели смогут быстрее находить, сохранять и делиться необходимыми документами, что позитивно скажется на их рабочем процессе и повысит эффективность обмена информацией.

Показатели успеха будут включать:

- Простота и удобство работы с ботом.
- Повышение эффективности работы с документами и медиафайлами.

Для проверки гипотезы необходимо было провести опрос среди целевой аудитории и пилотное тестирование бота.

В ходе решения поставленных задач были использованы следующие **методы исследования**:

- Анализ
- Опрос
- Обобщение
- Описание
- Моделирование
- Программирование

Объектом исследования являются телеграм-боты.

Глава 1.

Telegram и его возможности.

Telegram (2013 г.) — кроссплатформенный мессенджер, разработанный Павлом и Николаем Дуровым. Он обеспечивает конфиденциальное общение благодаря мощному шифрованию и "секретным чатам".

- **Защита конфиденциальности:** Telegram предлагает мощное шифрование для защиты переписки. Это включает в себя функцию "секретных чатов", где сообщения могут быть удалены после определенного времени или признаны самоуничтожающимися.
- **Богатый функционал:** Telegram предлагает множество возможностей, таких как групповые чаты на большое количество участников, создание каналов для трансляции контента, функция голосовых и видеозвонков, а также интеграция с музыкальными и новостными ботами.
- **Открытая платформа для разработчиков:** Telegram предоставляет API, позволяющий разработчикам создавать собственные боты и различные интеграции, что способствует разнообразию приложений и услуг внутри платформы.
- **Отсутствие рекламы:** В отличие от многих других мессенджеров, Telegram не использует рекламу в своем интерфейсе, что создает более чистый и непрерывный пользовательский опыт.

Telegram — один из самых популярных мессенджеров (800 млн пользователей в месяц), привлекающий пользователей удобством и безопасностью.

Понимание Telegram важно для разработки ботов и их интеграции в проект.

Функциональные возможности Telegram-ботов.

Telegram-боты — это приложения для автоматизации задач в Telegram. Они обрабатывают сообщения, выполняют команды, взаимодействуют с сервисами и приложениями.

Функциональные возможности:

- ✓ Telegram-боты могут выполнять множество функций, включая отправку сообщений, ответы на запросы пользователей, выполнение команд, работу с базами данных, автоматизацию процессов и многое другое.
- ✓ Интерфейс взаимодействия: Пользователи могут взаимодействовать с ботами через обычные чаты в Telegram, отправляя им текстовые сообщения или команды.
- ✓ Боты могут отвечать на эти сообщения автоматически или с помощью специально разработанных сценариев.
- ✓ Telegram-боты применяются в образовании, бизнесе, медицине, финансах и других областях для создания сервисов, бронирования, организации конкурсов, обработки заказов и т. д.

Разработка и использование Telegram-ботов открывает перед разработчиками и пользователями много новых возможностей для автоматизации задач, улучшения пользовательского опыта и создания новых сервисов и приложений.

Язык программирования для разработки ботов.

Для создания ботов для Телеграм популярен Python

Преимущества:

- Простота и лаконичность кода, что делает его привлекательным для начинающих и опытных разработчиков.
- Большое количество библиотек, таких как python-telegram-bot, aiogram, упрощающих разработку и интеграцию с Telegram API.
- Большое сообщество разработчиков и обширная документация, что облегчает получение помощи и решение проблем.

Недостатки:

- Медленная скорость выполнения по сравнению с некоторыми другими языками.
- Ограничения в работе с многопоточностью, что может затруднить масштабирование.

Библиотека для разработки Telegram-бота.

Основные библиотеки на языке Python, которые широко используются для создания Telegram ботов. Ниже представлен краткий обзор:

Telebot - это популярная библиотека Python для создания телеграм- ботов. Она была разработана для простоты и удобства использования, и быстро завоевала популярность среди разработчиков. Эта библиотека предоставляет простой и интуитивно понятный способ создания ботов, что делает ее привлекательным выбором как для начинающих, так и для опытных разработчиков.

Преимущества:

- Простота в использовании: Telebot предоставляет простой и интуитивно понятный способ создания ботов, что упрощает начало работы.
- Обширные примеры кода: Библиотека сопровождается множеством примеров кода и хорошо структурированной документацией.
- Модульность и расширяемость: Telebot позволяет создавать модульные боты и легко расширять их функциональность.

Недостатки:

- Ограниченный функционал: Telebot может не подойти для проектов, требующих сложной функциональности.

Благодаря своей простоте и удобству использования, Telebot является отличным выбором как для начинающих, так и для опытных разработчиков, которые хотят быстро и легко создать Telegram бота.

Принципы работы

Принципы работы бота базируются на взаимодействии между пользователем и сервером Telegram через API. Основные этапы включают формирование и отправку запросов, обработку ответов, использование полученных данных и обработку возможных ошибок. Рассмотрим каждый из этих аспектов подробнее.

1. Формирование и отправка запроса

Когда пользователь отправляет команду (например, /start). Бот извлекает данные из сообщения и формирует запрос к Telegram API

2. Обработка входящих данных: Бот извлекает данные из входящего сообщения. Например, если пользователь отправил фотографию, бот определяет тип файла и сохраняет его для дальнейшей отправки на облако.

3. Формирование запроса: Для отправки ответа пользователю бот формирует запрос на основе введенной информации. Например, если пользователь хочет загрузить фото или документ, бот формирует сообщение, которое отправляется пользователю для подтверждения или дополнительной информации.

4. Обработка ответа

После отправки запроса бот ждет ответа от Telegram API, который включает разные статусы и сообщения:

Подтверждение загрузки: Бот получает подтверждение о том, что файл загружен на облако пользователю. Telegram API возвращает информацию о результатах выполнения запроса.

Обработка результатов: Если пользователь получил фотографию или документ, бот должен убедиться, что файл был успешно загружен. После загрузки бот формирует новое сообщение для отправки пользователю.

4. Использование полученных данных

Полученные данные от пользователя, такие как фотографии, видео или документ обрабатываются для дальнейшей загрузки и распределения на облаке.

5. Обработка ошибок

Ошибки могут возникать на разных этапах работы бота. Для обеспечения стабильной работы важно корректно обрабатывать ошибки:

Некорректные данные: Если пользователю прислали файл неверного формата, не отправили текст или на облачном хранилище закончилось место, бот должен сообщить ему об этом и запросить исправления.

Системные ошибки: Если происходит сбой на стороне API Telegram или сервера, бот должен уметь обрабатывать эти ситуации. Например, отменять запросы или показывать сообщения о временной недоступности сервиса.

Логирование ошибок: Важно вести журнал ошибок, чтобы отслеживать проблемы и их причины. Это поможет в дальнейшем улучшить код и выпустить новые обновления, которые улучшат работу бота.

Глава 2.

Создание Telegram-бота.

Для реализации Telegram-бота принято решение использовать язык программирования python. Python обычно требует меньше кода для достижения тех же целей, чем многие другие языки программирования. Это позволяет программистам быстрее писать код и упрощает его понимание.

Для работы с Telegram использовалась библиотека Telebot, Это простой и понятный фреймворк, который подходит для начинающих.

Среда разработки

В качестве среды разработки для программирования использовался PyCharm – бесплатный и открытый текстовый редактор с поддержкой множества языков программирования, обладающий мощными функциями, такими как подсветка синтаксиса, автодополнение и интеграция с Git. Он расширяем и кроссплатформенн, что делает его удобным инструментом для разработки на различных операционных системах. PyCharm также

имеет обширную систему расширений, позволяющую настраивать его под индивидуальные потребности разработчика.

Создание токена в Telegram

Определившись с платформой разработки, необходимо зарегистрировать бота. Для этого нужен другой бот. BotFather – официальный сервис для создания, настройки и управления созданными телеграм-ботами. Он не занимается конкретными функциями и командами – только косметическими изменениями вроде названия, описания и аватарки. Но главное – он выдает уникальный токен, который необходим для авторизации бота и работы с Bot API. Любой, кто получит токен, сможет управлять ботом. Для получения токена в BotFather необходимо:

1. Отправить в чат с BotFather команду /newbot.
2. Ввести название бота – в этой категории особых ограничений нет.
3. Ввела юзернейм бота — его техническое имя, которое будет отображаться в адресной строке. Он должен быть уникальным, написан на латинице и обязательно заканчиваться на bot. Так «Telegram» защищается от злоумышленников, которые могут выдавать ботов за реальных людей.
4. BotFather отвыдает токен бота – его можно использовать для настройки в сторонних сервисах, а главное для программирования.

Был создан бот с названием “CloudAssistant”, ID которого @CloudAssistant_Bot, определена аватарка и описания при помощи команд /setuserpic и /setabouttext соответственно, добавлены собственные команды, одна из которых: /start (Приветственную команду).

Тестирование бота.

Проведена проверка бота на работоспособность: бот готов и отлично выполняет свои функции.

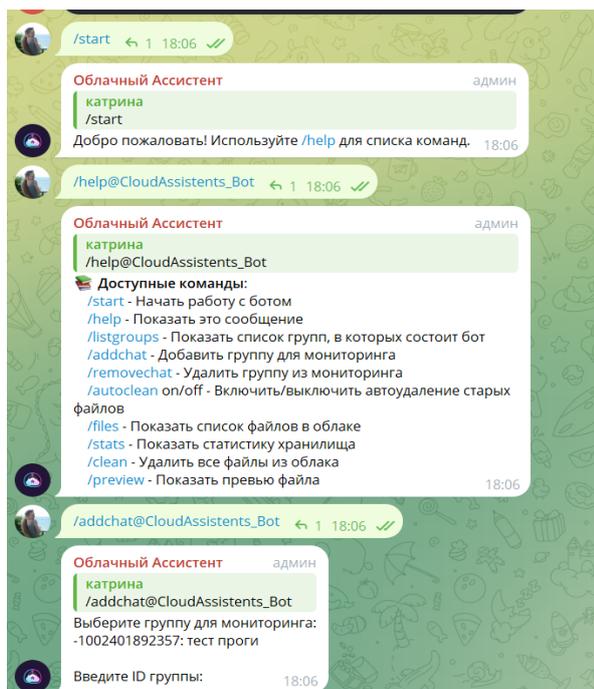


Рис. 1

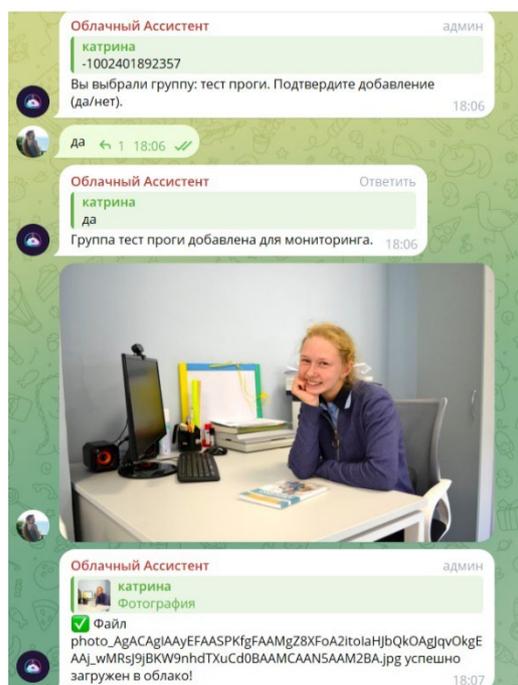
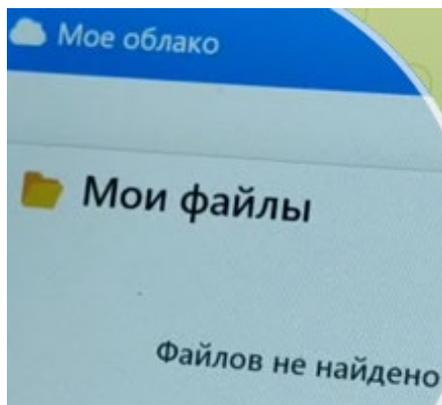


Рис. 2

Как выглядит облако на данный момент



На Рис.3 представлена часть кода работы Телеграмм бота, на Рис.4 часть кода работы олака.

```

202     await handle_private_message(update, context)
203     else:
204         await handle_group_message(update, context)
205
206
207     async def handle_private_message(update: Update, context: CallbackContext):
208         user_id = update.effective_user.id
209         message = (
210             "📌 **Главное меню**\n\n"
211             "Выберите действие:\n"
212             "📁 /myfiles - Просмотр файлов\n"
213             "⚙️ /settings - Настройки"
214         )
215         await update.message.reply_text(message, parse_mode='Markdown')
216
217
218     async def handle_group_message(update: Update, context: CallbackContext):
219         user_id = update.effective_user.id

```

Рис. 3

```

188     file_bytes = await file.download_as_bytearray()
189     files = {'file': (filename, file_bytes)}
190     response = requests.post(f'{API_URL}/upload/{user_id}', files=files)
191
192     if response.status_code == 200:
193         await context.bot.send_message(
194             chat_id=user_id,
195             text=f"✅ Файл `{filename}` успешно загружен в облако!",
196             parse_mode='Markdown'
197         )
198     else:
199         error = response.json().get('error', 'Unknown error')
200         await context.bot.send_message(
201             chat_id=user_id,
202             text=f"❌ Ошибка загрузки: {error}"
203         )
204     except Exception as e:
205         logger.error(f"File handling error: {e}")

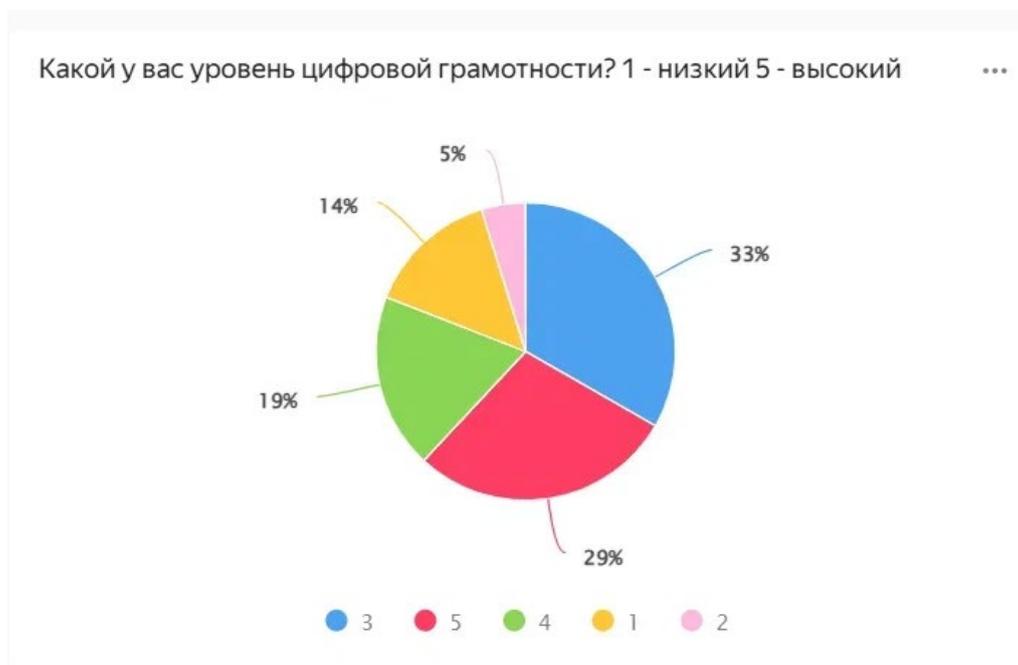
```

Рис. 4

Аналитика Опрос

В рамках подготовки проекта был проведен опрос среди учителей, в котором приняли участие 21 человек. Всего было задано 10 вопросов, направленных на выявление уровня цифровой грамотности педагогов, их предпочтений в использовании мессенджеров и отношения к внедрению искусственного интеллекта (ИИ). Рассмотрим основные результаты опроса и сделаем соответствующие выводы.

Какой у вас уровень цифровой грамотности? 1 - низкий 5 - высокий



Уровень цифровой грамотности

На вопрос о самооценке уровня цифровой грамотности от 1 до 5 баллов ответы распределились следующим образом: 29% респондентов оценили свой уровень как максимальный (5), еще 19% поставили себе оценку 4, остальные 48% оценили свои навыки ниже. Эти данные свидетельствуют о том, что хотя значительная часть опрошенных имеет хорошие навыки работы с цифровыми технологиями, остается группа педагогов, которые могли бы улучшить свою цифровую грамотность. Это подчеркивает важность проекта, который направлен на облегчение взаимодействия с различными программными продуктами, данными и снижение нагрузки на преподавателей.

Используете ли вы мессенджеры для общения?



Почти все участники опроса (95%) подтвердили использование мессенджеров для общения. Тут мне так же хочется отметить про свои личные наблюдения и общение с учителями моей школы. Почти вся информация поступает через мессенджеры.

Сколько времени вы тратите на работу с мессенджерами?



Время, затрачиваемое на работу с мессенджерами

Значительная часть участников (62%) указали, что ежедневно тратят более одного-двух часов на взаимодействие с мессенджерами. Учитывая высокую занятость учителей, сокращение этого времени станет важным шагом к повышению эффективности их работы.

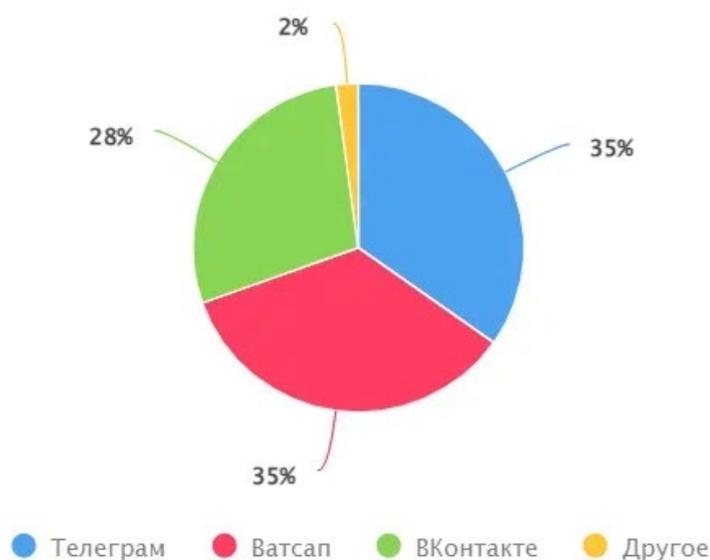
Проект предполагает внедрение решений, которые помогут снизить временные затраты на обработку и сохранение информации.

Какие мессенджеры предпочитаете для общения?

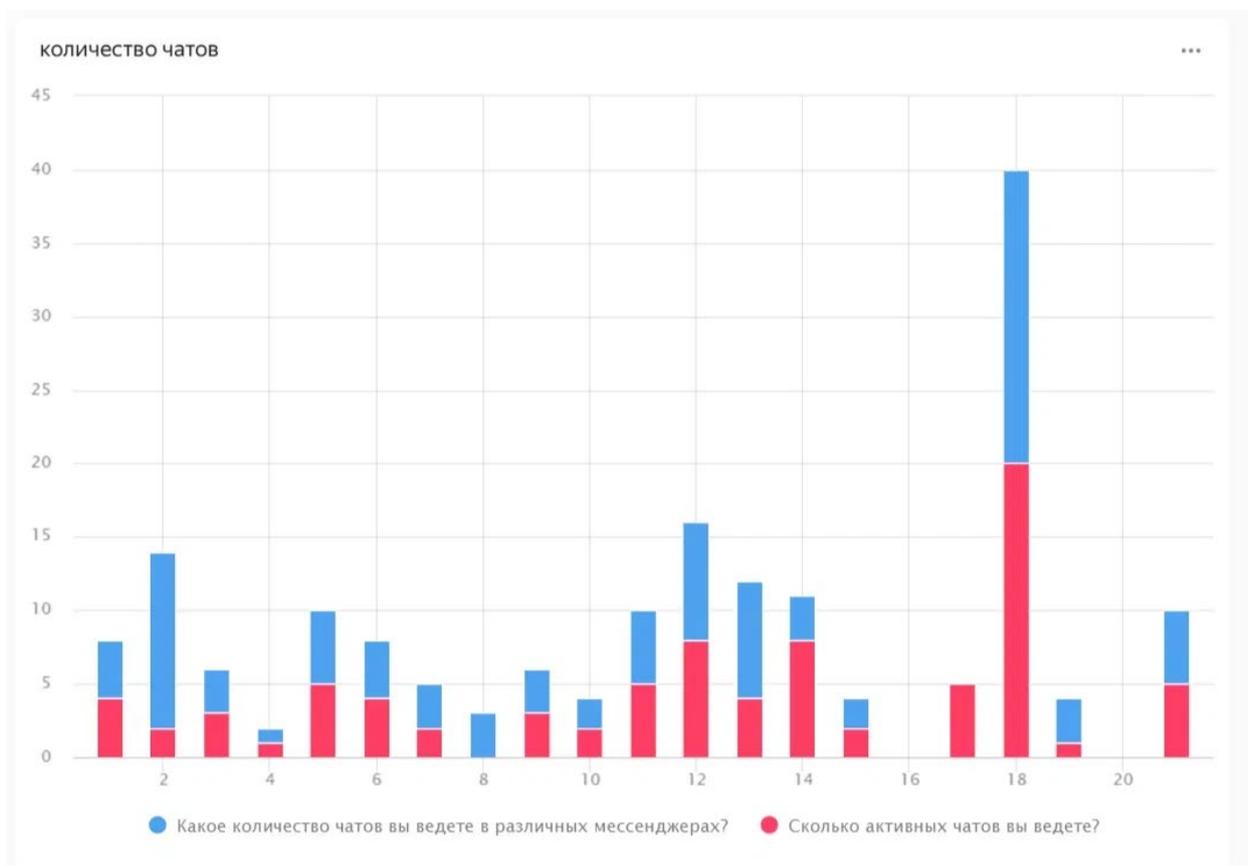
Предпочтительные мессенджеры

Согласно результатам опроса, самыми популярными мессенджерами среди учителей являются Telegram и WhatsApp, каждый из которых выбрали по 38% респондентов. На третьем месте оказался VKontakte (14%), остальные мессенджеры набрали меньший процент голосов. Это позволяет нам сосредоточиться на разработке функционала для наиболее популярных платформ.

Какие мессенджеры предпочитаете для общения? ...



Какое количество чатов вы ведете в различных мессенджерах? Сколько активных чатов вы ведете?



На этой диаграмме мы можем посмотреть количество чатов учителей и какую часть из них ведут сами преподаватели. Хотелось отметить, что в опросе принимали участие учителя с разными должностями. Поэтому можно догадаться, что большое количество чатов ведут, например, завучи школы. Максимальное количество чатов в таблице - 20. Можно предположить, что вести такое количество чатов крайне сложно.

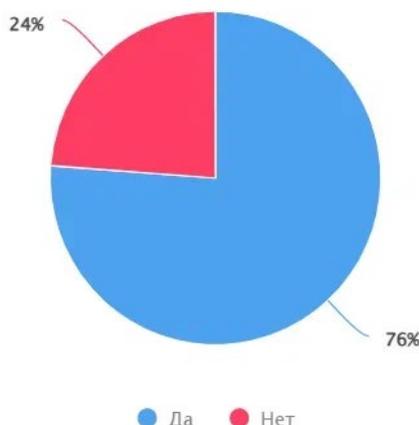
Сталкивались ли вы с проблемами при работе с файлами в мессенджерах, если да то с какими.

Проблемы при работе с файлами в мессенджерах

Многие респонденты столкнулись с трудностями при работе с файлами в мессенджерах. Наиболее часто упоминались проблемы с поиском нужных документов, сложности с сохранением больших объемов данных и неудобство навигации между различными чатами. Эти результаты подтверждают необходимость разработки удобных инструментов для управления файлами.

Используете ли вы мессенджеры для сохранения важной информации?

Используете ли вы мессенджеры для сохранения важной информации ...



Сохранение важной информации в мессенджерах

Большинство участников опроса (76%) используют мессенджеры для хранения важной информации. Это подтверждает удобство использования мессенджеров как инструмента для организации данных, однако не исключает необходимости улучшения механизмов поиска и хранения информации.

Как вы считаете, искусственный интеллект(ии) помог бы в работе с данными в мессенджерах?

Какой функционал ИИ вы бы хотели видеть в мессенджерах?

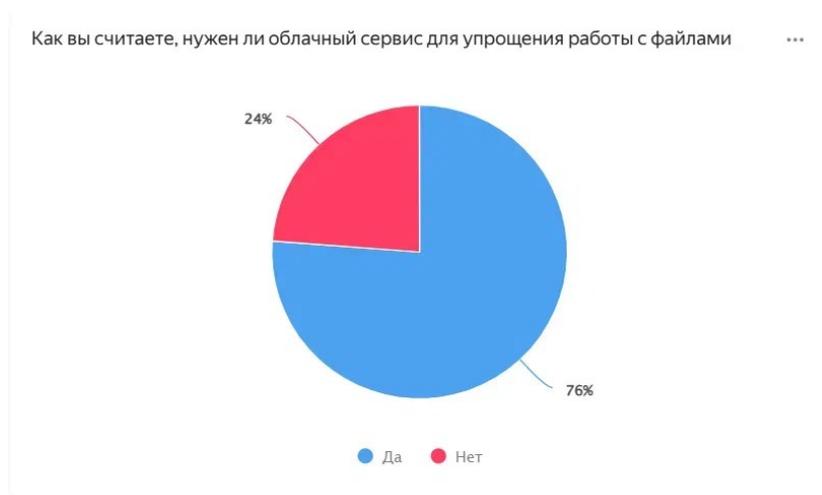
Эти 2 вопроса мне бы хотелось объединить ведь они связаны со стремительно развивающимся искусственным интеллектом. Мне стало интересно отношение преподавателей к этому инструменту.

id	Как вы считаете, искусственный интеллект(ИИ) помог бы в работе с данными в мессенджерах?	Какой функционал ИИ вы бы хотели видеть в мессенджерах?
1	Да	Никакие
2	Нет	-
3	Да	-
4	Да	-
5	Да	Генерация картинок, видео
6	Да	-
7	Да	Планирование дел
8	Да	Не знаю щ9zzzzzzzzшш
9	Да	Активности
10	Да	Чатбот
11	Да	-
12	Да	Затрудняюсь
13	Да	-
14	Да	-
15	Да	Затрудняюсь ответить
16	Да	Сервис русских субтитров в видеоконтенте
17	Да	-
18	Да	-
19	Да	Не могу конкретизировать.
20	Да	-
21	Да	-

Искусственный интеллект в мессенджерах

Вопросы об отношении к использованию ИИ в мессенджерах показали, что подавляющее большинство (90%) видят потенциал в применении технологий искусственного интеллекта для автоматизации рутинных задач. Среди предложенных функций были выделены генерация контента (картинки, видео), планирование дел, создание чат-ботов и добавление русских субтитров для видеоматериалов, что могло бы значительно облегчить работу учителей, включая тех людей, которые имеют ограниченные возможности слуха.

Как вы считаете, нужен ли облачный сервис для упрощения работы с файлами в мессенджерах?



Нужен ли облачный сервис для работы с файлами?

Ключевой вопрос касался возможности внедрения облачного сервиса для упрощения работы с файлами в мессенджерах. 76% респондентов выразили уверенность в полезности такого решения, что свидетельствует о высокой актуальности нашего проекта и его потенциальной ценности для пользователей.

Таким образом, результаты опроса демонстрируют острую потребность в оптимизации процессов работы с мессенджерами и файлами, а также подчеркивают интерес к внедрению искусственного интеллекта для повышения эффективности труда учителей.

Перспективы дальнейшего развития

Несмотря на уже достигнутые результаты, проект имеет большой потенциал для дальнейшего развития. Возможные направления включают:

Добавление новых функций:

- Подключение ИИ:

В дальнейшем планируется подключение искусственного интеллекта, который окажет существенную помощь в выводе приоритетных файлов в облаке из определенных групп, а

также в пересказе выбранных чатов и событий, произошедших в течение дня. Это позволит экономить время пользователя на анализе информации. Возможно внедрение функции распознавания изображений для категоризации фотографий, генерации кратких описаний документов и прочие полезные функции. Бот сможет автоматически создавать папки по типам файлов в облаке, а также предоставлять возможность добавления тегов для удобного поиска информации.

- Также планируется реализовать систему уведомлений. Пользователь должен получать оповещения о загрузке, а также уведомления о завершении загрузки и об ошибках, которые могут возникать во время работы. Напоминания о старых файлах также будут частью функционала, как уже упоминалось, чтобы избежать нарушения лимита по пространству в облачном хранилище.
- Важной функцией станет совместный доступ — пользователи смогут делиться ссылками на облачные папки. В такие папки могут сохраняться все данные сразу из нескольких чатов, а затем отправляться по одной ссылке людям, с которыми пользователь желает поделиться информацией.
- Мы также планируем интеграцию с другими полезными сервисами, такими как календари и почтовые сервисы, что позволит упрощать работу пользователей и напоминать о важных событиях, а также пересылать узнаваемые данные на электронную почту.

Таким образом, проект "Разработка облачного ассистента для преподавателя школы" подтверждает свою значимость и эффективность в контексте современных подходов к цифровому миру. Успех данного проекта заключается в его способности сочетать технологии с работой педагогов, предлагая пользователям новые, уникальные инструменты для работы с файлами. Бот может стать не только удобным инструментом для учителей, но и платформой для общения и обмена личными данными, что делает его поистине ценным в наше время.

Заключение

Облачный ассистент — это инновационный инструмент, который может значительно упростить жизнь преподавателя, обеспечивая удобный перенос файлов, безопасное хранение данных и помощь в управлении информацией. Использование таких технологий способствует повышению эффективности образовательного процесса и позволяет преподавателям сосредоточиться на выполнении своих основных задач. В условиях стремительного развития цифровых технологий внедрение облачного ассистента в повседневную практику становится не просто удобством, а необходимостью, отвечающей вызовам современного мира.

Список используемой литературы

1. **Официальная документация Python:**

← <https://docs.python.org/3/>

← Описание стандартных библиотек, синтаксиса и примеров использования Python.

←

2. **Официальная документация библиотеки python-telegram-bot:**

← <https://python-telegram-bot.org/>

← Полное руководство по использованию библиотеки для создания Telegram-ботов.

←

←

3. **Официальная документация Telegram Bot API:**

← <https://core.telegram.org/bots/api>

← Описание всех методов и возможностей Telegram Bot API.

←

4. **Документация Flask:**

<https://flask.palletsprojects.com/en/stable/>

← Руководство по созданию веб-приложений на Flask (необходимо для облачного хранилища).

←

5. **Документация Pillow (PIL):**

← <https://pillow.readthedocs.io/en/stable/>

← Руководство по работе с изображениями в Python.

ТЕЛЕГРАММ-БОТ ДЛЯ ТРЕНИРОВОК ПО БЕГУ С ИИ

Боков Владимир Максимович

Москва г, Школа №1251 10 класс

Секция: Инженерный бизнес и менеджмент

Научный руководитель: Соколов Михаил Александрович, к.н., доцент каф. ИБМ7

МГТУ им. Н.Э. Баумана

Аннотация

Повысить долю населения, занимающегося спортом, могли бы инструменты мотивации и профессиональной поддержки. С учетом роста интереса к фитнесу и технологий, данный проект имеет значительный коммерческий потенциал и может стать успешным инструментом на рынке фитнес-услуг.

Проект представляет из себя телеграм-бота, который генерирует план беговых тренировок, учитывает индивидуальные данные пользователей и имеет расширенный функционал. Для реализации проекта использовался асинхронный подход на языке Python с применением библиотек sqlalchemy, aiohttp и aiogram, что обеспечивает высокую производительность и отзывчивость приложения. Программа была структурирована с соблюдением принципов чистого кода. Результаты работы с фокус-группой подтверждают возможность коммерческого использования и показывают, что бот способен эффективно поддерживать пользователей в достижении их фитнес-целей, улучшая их мотивацию и вовлеченность.

Введение

В последние годы наблюдается значительный рост интереса к физической активности и здоровому образу жизни среди населения. Все больше людей выбирают бег как один из наиболее доступных и эффективных способов поддержания физической формы. Бег, как доступный и эффективный способ тренировки сердечно-сосудистой системы, привлекает множество людей. Современный образ жизни часто сопровождается высоким уровнем стресса. Физическая активность, включая бег, помогает снизить уровень стресса и улучшить общее психоэмоциональное состояние. Социальные сети и сообщества: Развитие социальных сетей и фитнесприложений способствовало созданию сообществ бегунов. Люди делятся своими достижениями, советами и мотивацией, что вдохновляет других присоединиться к этому занятию.

В последние годы наблюдается общий тренд на здоровый образ жизни, который включает правильное питание, физическую активность и заботу о ментальном здоровье. Бег становится важной частью этого образа жизни.

Эти факторы способствуют тому, что бег и фитнес в целом становятся все более популярными среди людей разных возрастов и уровней подготовки.

В то же время, с увеличением использования мобильных устройств и мессенджеров, таких как Telegram, возникает необходимость в создании удобных и функциональных инструментов для помощи в тренировках. Отличную возможность для создания продукта, позволяющего пользователям тренироваться без лишних затрат и забот, представляют чат-боты.

Чат-бот — это программа, которая выглядит как обычный чат на платформе, но представляет собой не живого собеседника, а код, выполняющийся для взаимодействия с пользователем. Они могут функционировать в различных мессенджерах (например, WhatsApp, Telegram), веб-сайтах или мобильных приложениях.

Чат-боты могут работать 24/7, предоставляя пользователям возможность получать информацию и помощь в любое время. Они могут быстро обрабатывать запросы и предоставлять ответы, что значительно сокращает время ожидания по сравнению с традиционными методами обслуживания. Современные чат-боты могут анализировать данные пользователей и предлагать персонализированные рекомендации, что повышает уровень удовлетворенности клиентов.

Таким образом, чат-боты представляют собой мощный инструмент, который может значительно улучшить взаимодействие с клиентами и повысить эффективность бизнес-процессов.

В последние годы в различные ИТ-продукты бывают интегрированы решения искусственного интеллекта(ИИ). Он позволяет создавать персонализированный опыт для пользователей, анализируя их. Чат-боты и виртуальные ассистенты, основанные на ИИ, обеспечивают быстрые и эффективные ответы на запросы пользователей, улучшая качество обслуживания и удовлетворенность клиентов. ИИ открывает новые горизонты для разработки продуктов и услуг. Например, технологии машинного обучения могут использоваться для создания новых приложений в области здравоохранения, финансов, транспорта и многих других. Автоматизация процессов с помощью ИИ может значительно снизить затраты на трудозатраты и ошибки, что делает компании более эффективными. Компании, которые интегрируют ИИ в свои продукты и услуги, могут получить значительное преимущество

перед конкурентами, предлагая более инновационные решения и улучшая свои бизнес-процессы.

В ответ на эти тенденции был разработан Telegram-бот, предназначенный для создания индивидуальных планов тренировок по бегу. Этот проект интегрирует решения на основе искусственного интеллекта, что позволяет учитывать индивидуальные данные пользователей и предоставлять персонализированные рекомендации. Бот не только помогает пользователям в планировании тренировок, но и мотивирует их на достижение поставленных целей.

Актуальность и значимость проекта

В условиях современного общества, где здоровье и физическая активность становятся приоритетами для многих людей, разработка телеграмбота для тренировок по бегу представляет собой актуальный и востребованный проект. Рост интереса к спорту и фитнесу, а также увеличение числа людей, занимающихся физической активностью самостоятельно, создают благоприятные условия для внедрения инновационных решений в этой области.

Согласно статистическим данным, доля граждан, занимающихся спортом, продолжает расти. Бег является одним из самых популярных видов физической активности, что подтверждает его выбор более трети людей, стремящихся поддерживать свою физическую форму. По опросам ВЦИОМ¹, 54% населения России занимаются спортом либо ежедневно, либо несколько раз в месяц (лишь 38% в недалеком 2006). При этом 37% из них занимаются бегом или спортивной ходьбой. Следовательно, целевая аудитория продукта очень широка и только продолжает расширяться. Данные ВЦИМО также показывают, что 27% опрошенных не занимаются спортом из-за отсутствия мотивации или тренера, а 43% занимаются спортом самостоятельно. Заниматься спортом самостоятельно может быть не всегда эффективно. Без поддержки тренера или группы сложно сохранять высокую мотивацию и дисциплину. Многие люди не имеют достаточных знаний о правильных тренировочных программах и методах восстановления, что может снизить эффективность занятий. Без помощи тренера сложно правильно отслеживать свой прогресс и корректировать программу тренировок при необходимости.

Эти факторы могут снизить эффективность тренировок и увеличить риск травм, поэтому работа с тренером или в группе может быть более предпочтительной. Однако функционал бота предупреждает подобные проблемы. Пользователь всегда может обратиться за советом, изменять план тренировок, а также найти других тренирующихся поблизости для совместных тренировок. Возможность вести дневник измерений массы тела и видеть изменения на графике поможет поднять мотивацию или обратить внимание на отсутствие положительных тенденций. Не потерять интерес к занятиям помогут уведомления о предстоящей тренировке и уведомления о пропусках тренировок. В последние годы наблюдается значительное увеличение времени, проводимого пользователями на мобильных устройствах. Мессенджеры, такие как Telegram, становятся основными платформами для общения и получения информации. Интеграция фитнес-решений в популярные мессенджеры позволяет обеспечить доступность и удобство использования, что повышает шансы на успех проекта. Telegram входит в топ 5 самых популярных мессенджеров в России². На момент 2021 года им пользовалось около 55% опрошенных. Telegram демонстрирует наибольший прирост

¹ Опрос ВЦИОМ, доступен по ссылке - <https://wciom.ru/analytical-reviews/analiticheskii-obzor/sport-dlja-vsekh>

² Опрос ВЦИОМ - <https://wciom.ru/analytical-reviews/analiticheskii-obzor/rossiiskaja-auditorija-socialnykh-seteii-messendzherov-izmeneniija-na-fone-specoperacii>

³ Опрос ВЦИОМ - <https://wciom.ru/analytical-reviews/analiticheskii-obzor/ii-vash-novyi-luchshii-drug>

аудитории среди всех социальных сетей и мессенджеров (+13 п.п. постоянных пользователей к началу 2021 г.). Изменение доли ежедневных пользователей также наиболее ощутимо: у Telegram выросла с 22% до 35% (с 2017 по 2021г). Стремительный рост пользователей в последние 3 года также не прекращается.

Большая часть россиян воспринимают ИИ позитивно³, прежде всего как помощника - 54%. То есть к технологиям ИИ относятся скорее, как к инструменту, способному облегчить повседневные задачи. Современные пользователи стремятся к персонализированным решениям. Бот, который учитывает индивидуальные данные пользователей (вес, пол, возраст, уровень подготовки, цели тренировок) и предлагает адаптированные планы беговых тренировок, отвечает на потребности аудитории и способствует повышению мотивации. Использование технологий искусственного интеллекта и анализа данных позволяет создать более эффективные и адаптивные решения для пользователей. Внедрение таких функций, как геолокация, отслеживание изменений веса и построение графиков прогресса, делает продукт более привлекательным и функциональным.

Создание качественного продукта с разнообразными функциями открывает возможности для монетизации через подписки, рекламу и платные функции. Это делает проект не только актуальным, но и коммерчески жизнеспособным. Об этом более подробно в Коммерциализация онлайн фитнес-приложений на странице. Таким образом, разработка телеграм-бота для тренировок по бегу является значимым вкладом в сферу фитнеса и здоровья, отвечая на актуальные потребности пользователей и учитывая современные тренды в использовании технологий. Проект способен не только улучшить качество жизни пользователей, но и создать успешный коммерческий продукт на растущем рынке фитнес-приложений.

Выбор средств реализации

Как уже было отмечено, Telegram был выбран в качестве платформы для реализации проекта из-за его растущей популярности. Telegram также предоставляет широкие возможности для разработки. Простой доступ к API позволяет быстро и без лишних затрат создавать чат-ботов на этой платформе.

Для самой программной реализации Telegram-бота, ориентированного на планирование и управление беговыми тренировками, был выбран современный стек технологий, обеспечивающий высокую производительность, удобство разработки и масштабируемость проекта. Каждый из используемых инструментов был тщательно подобран с учетом специфики задачи и требований к функционалу.

Язык программирования Python был выбран в качестве основного языка программирования благодаря своей универсальности, богатой экосистеме библиотек и простоте написания кода. Он широко применяется в разработке приложений, связанных с искусственным интеллектом, анализом данных и автоматизацией процессов, что делает его идеальным выбором для реализации данного проекта.

Aioogram — это современный асинхронный фреймворк для создания Telegram-ботов. Его использование позволило реализовать высокопроизводительное взаимодействие с платформой Telegram, обеспечив быстрый обмен сообщениями между пользователем и ботом. Асинхронная архитектура Aioogram позволяет эффективно обрабатывать множество запросов одновременно, что особенно важно для приложения с потенциально большой аудиторией.

SQLAlchemy был выбран как ORM (Object-Relational Mapping) для работы с базами данных. Этот инструмент предоставляет удобный интерфейс для хранения, извлечения и управления данными пользователей, такими как профили, цели тренировок, история изменений веса и другие параметры. Использование SQLAlchemy упрощает работу с базой данных, делая код более читаемым и поддерживаемым. Инструмент является одним из самых используемых среди разработчиков для взаимодействия с базами данных.

Такой подход позволяет избегать использование строк SQL запросов в файлах с кодом Python. Это делает программу безопаснее и обеспечивает защиту от, например, инъекций - один из распространённых способов взлома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода.

Для получения актуальных данных о погоде была интегрирована библиотека Aiohttp, которая обеспечивает асинхронные HTTP-запросы. Это решение позволяет быстро получать информацию о погодных условиях в зависимости от геолокации пользователя.

Для визуализации данных, таких как изменения массы тела пользователя, использовались библиотеки PIL и Matplotlib. Она(Matplotlib) предоставляет широкие возможности для создания качественных графиков, которые могут быть отправлены пользователю в виде изображений. Matplotlib была выбрана за ее гибкость, простоту использования и высокое качество рендеринга. PIL же используется для выполнения действий с файлами .jpg (картинками).

На этапе разработки были внедрены механизмы логирования и обработки ошибок, что позволяет оперативно выявлять и устранять проблемы. Для тестирования функционала использовались как автоматизированные тесты, так и ручное тестирование с участием фокус-группы. Для тестирования была неоднократно использована библиотека unittest. Полезная черта unittest - автоматизированное тестирование.

Помимо перечисленного были импортированы библиотеки requests, datetime, ast и apscheduler для запросов на сервер, работы с единицами времени, работы с конвертацией словарей и строк в другие типы данных и выполнение запланированных задач соответственно.

Перечисленные библиотеки помогли сделать и дальнейшие наработки. Например, обратную связь от пользователей. (рис.1)

Таким образом, выбранные средства реализации обеспечивают высокую производительность, гибкость и удобство разработки.

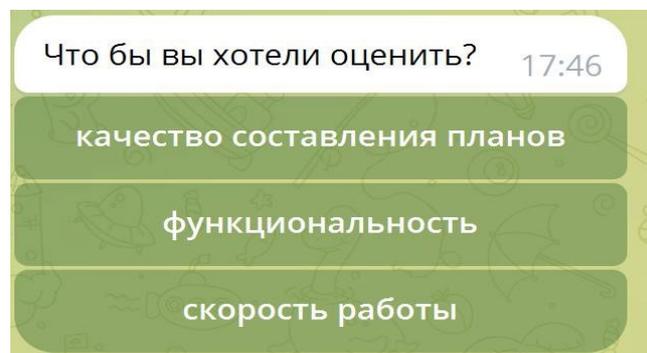


Рис. 1. Одна из форм обратной связи

Программная часть проекта. Основные методы

Весь код проекта разделен на несколько модулей (рис. 2). Каждый из модулей выполняет задачи определенной области, что способствует более организованной и структурированной разработке. Это позволяет улучшить читаемость кода, упростить тестирование и облегчить процесс отладки.

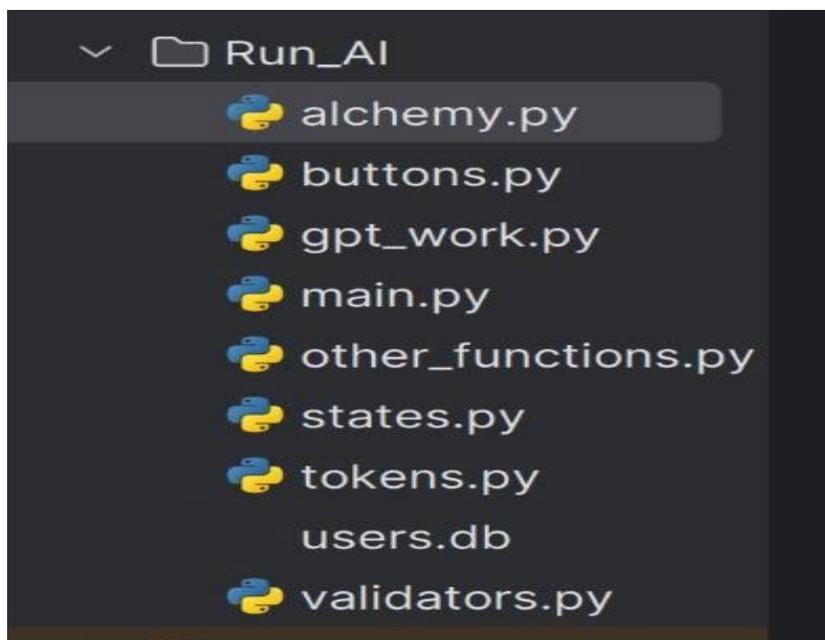


Рис. 2 Модули

Главный модуль, который выполняет роль связующего звена, является модуль main.py. Здесь происходит взаимодействие с пользователем. Выполняются строчки кода в зависимости от действий на стороне самого пользователя: обновление базы данных, запросы к модулю обеспечивающему работу с нейросетью, прием геоданных и так далее.

В main.py находятся все функции работающие на прием команд от пользователя. К примеру, на команду /start – запуск бота.(рис. 3)

```
@router.message(Command("start"))
async def cmd_start(message: types.Message, state: FSMContext):
    user_id = message.from_user.id
    first_name = message.from_user.first_name
    user_nik = message.from_user.username
    await insert_user(user_id, first_name, user_nik)
    await message.answer(
        f"Здравствуйте, {first_name}! Пожалуйста, заполните информацию о себе, чтобы приступить к тренировкам."
    )
    await message.answer("Для начала введите ваш возраст:")
    await state.set_state(Form.waiting_for_age)
```

Рис. 3. Функция запуска

После начинается процесс регистрации пользователя и добавление данных в базу. Бот переходит в состояние ожидания сообщения с числом лет(возраста). Все состояния прописаны в модуле states.py. Чтобы получить данные от пользователя созданы функции на прием текста(сообщений) и функции-кнопки. Рассмотрим первые (рис. 4)

```

@router.message(Form.waiting_for_age)
async def process_age(message: types.Message, state: FSMContext):
    age = message.text
    if check_age(age):
        await message.answer(
            "Необходимо ввести одно натуральное число, соответствующее вашему возрасту"
        )
        return
    await update_age(message.chat.id, age)
    await state.set_state(Form.waiting_for_weight)
    await message.answer("Теперь ваш вес:")

```

Рис. 4. Функция приема сообщения(возраста)

В состоянии ожидания возраста бот принимает на вход сообщение от пользователя, проверяет на корректность введенных данных (число от 0 до 150) и записывает в базу. Схожим образом реализована и функция на прием веса пользователя и количества тренировок в неделю.

Функции-кнопки представляют из себя клавиши прикрепленные к сообщению. Они позволяют удобно представлять варианты выбора. (рис. 5)

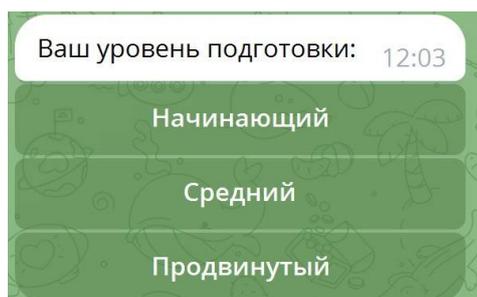


Рис. 5. Сообщение с кнопками

При нажатии функция возвращает код (callback data), обозначающий выбор.

В том же модуле main.py реализована функция отвечающая за ежедневную отправку уведомлений пользователям, у которых в этот день тренировка.

Функция handle_location ожидает на вход отправленную геолокацию. Сохраняет ее, и в случае обнаружения пользователей поблизости, выводит их контакты. Для этого используется поиск по координатам - долготе и широте.

Геолокация также необходима для определения точной погоды. (рис. 6)

```

@router.message(F.location)
async def handle_location(message: types.Message):
    latitude = message.location.latitude
    longitude = message.location.longitude
    geo = str(str(latitude)+' ', '+str(longitude))
    await save_location(message.from_user.id, geo)
    await bot.send_message(message.from_user.id, text: 'Геолокация сохранена')
    list_of_users = await get_people()
    awnser_list = ''
    for i in list_of_users:
        if 1 >= haversine(geo, i.geo) and haversine(geo, i.geo) != 0:
            if i.user_nik != None:
                awnser_list += str(i.name) + ', '+str(i.age)+' ->  @'+str(i.user_nik) + '\n'
    if awnser_list == '':
        await bot.send_message(message.from_user.id, text: f'К сожалению, список пользователей в вашем районе пуст')
    else:
        await bot.send_message(message.from_user.id, text: f'Вот список пользователей в вашем районе:\n \n{awnser_list}')

```

Рис. 6. Прием геолокации

В модуле `tokens.py` сохранены токены необходимые для API доступа к ресурсам нейросети и Telegram. В `gpt_work.py` прописаны функции для составления запросов к нейросети согласно документации сервиса. В `buttons.py` объявлены функции-кнопки, используемые для взаимодействия с пользователем. `users.db` – файл созданный для хранения таблиц. А в модуле `validators.py` функции для проверки вводимых пользователем данных.

В файле `other_functions.py` объявлены различные функции необходимые для расчетов, составления таблиц, графиков, планов тренировок, а также prompt текста для нейросети.

Рассмотрим функцию `plot_and_save_accidents` для графиков изменений массы тела (рис. 7). На вход функция берет три аргумента: `id` – идентификатор пользователя, `data` – словарь-показатели веса(кг), `filename` – название файла для отправки(по умолчанию `accidents_plot.png`). Функция использует библиотеку `Matplotlib` для построения кривой изменений. После отправки картинка, не занимая места в памяти, удаляется.

```

async def plot_and_save_accidents(id, data, filename="accidents_plot.png"):
    years = list(data.keys())
    accidents = [float(i) for i in list(data.values())]

    # Создаем график
    plt.figure(figsize=(10, 5))
    plt.plot(*args: years, accidents, marker="o", linestyle="-", color="b")

    # Добавляем заголовок и метки осей
    plt.title("График изменений массы тела")
    plt.xlabel("Число")
    plt.ylabel("Вес (кг)")

    # Добавляем сетку для удобства восприятия
    plt.grid()

    # Сохраняем график как изображение
    plt.savefig(filename)
    photo_file = FSInputFile(filename)
    await bot.send_photo(chat_id=id, photo=photo_file)
    # Закрываем график, чтобы освободить память
    plt.close()
    if os.path.exists('accidents_plot.png'):
        os.remove('accidents_plot.png')

```

Рис. 7. Построение графика, вывод картинки

Рассмотрим еще одну функцию в качестве примера. `get_weather` получает сведения о погоде используя широту и долготу, как аргументы в url. Чтобы найти необходимые данные используются регулярные выражения.

```

async def get_weather(l1, l2):
    url = f'https://yandex.ru/pogoda/ru-RU/details/auto/today?lat={l1}&lon={l2}&lang=ru&via=dnavv'

    async with aiohttp.ClientSession() as session:
        async with session.get(url) as response:
            res = await response.text()
            pattern = r'<p class="A11Y_visuallyHidden_y0sw0 visuallyHidden">0:00:s*(.*?)</p>'
            match = re.search(pattern, res)
            if match:
                return match.group(1)
            else:
                return None

```

Рис. 8. Получение сведений о погоде

Одна из важнейших частей программы составлена в `alchemy.py`. Здесь реализованы все операции работы с базами данных, таких как Create, Read, Update и Delete (называется CRUD). Для ознакомления можно взглянуть на пару таких операций.

Функция `get_geo` по `id` пользователя обращается к базе данных (`select`) и возвращает его координаты, при их наличии, так как этот параметр необязательный при регистрации (рис. 9).

```

async def get_geo(user_id):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            result = await session.execute(
                select(User.geo).where(User.user_id == user_id)
            )
            geo = result.scalars().first()
            return geo

```

Рис. 9. Запрос на получение геолокации пользователя из бд

Таким же методом реализованы и другие функции возвращающие какие-либо данные по одному пользователю (вес, кол-во тренировок и т.п.)

Вот еще один пример функции из этого модуля для работы с бд. В данном случае исходное значение по умолчанию (None) уровня подготовки спортсмена заменяется на выбранных при регистрации (рис. 10). Гибкость всей логики программы позволяет проходить регистрацию повторно при необходимости.

```

async def update_level(user_id, level):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = update(User).where(User.user_id == user_id).values(level=str(level))
            await session.execute(stmt)
            await session.commit()

```

Рис. 10. Обновление значения уровня подготовки

Составление таблицы users, хранящую записи о пользователях, можно посмотреть на рис. 11.

```

class User(Base):
    __tablename__ = "users"
    user_id = Column(Integer, primary_key=True, unique=True)
    name = Column(String)
    age = Column(Integer, default=None)
    per_week = Column(Integer, default=None)
    gender = Column(Text, default=None)
    level = Column(Integer, default=None)
    target = Column(String, default=None)
    plan = Column(String, default=None)
    weight_gr = Column(String, default=None)
    user_nik = Column(String, unique=True)
    geo = Column(String, default=None)
    lang = Column(String, default='ru')

```

Рис. 11. Описание таблицы (sqlalchemy)

Объект User, наследуемый от класса Base из библиотеки, используется при создании таблицы.

Коммерциализация онлайн фитнес-приложений

В начале уже были приведены статистические данные, доказывающие высокий потенциал коммерциализации подобных проектов. Для обеспечения прибыльности предполагается использовать несколько источников дохода.

Подписка может включать введение платных тарифных планов с расширенным функционалом, доступ к эксклюзивным материалам и персонализированным рекомендациям на основе анализа данных пользователя. Также пользователи с подпиской смогут отключать рекламу от партнеров.

Интеграция **рекламных** материалов от партнеров в области фитнеса, спортивного питания и активного образа жизни может быть предусмотрена в виде рассылки. Такой подход (**Freemium**) часто используется онлайн фитнесприложениями, где основная часть продукта или услуги предоставляется пользователям бесплатно, но за дополнительные функции, улучшения или контент может взиматься плата.

Введение дополнительных платных функций, таких как персональные консультации с тренерами, доступ к специализированным диетам, может быть также рассмотрено.

Для успешной коммерциализации продукта необходимо разработать стратегию маркетинга и продвижения, включающую социальные сети или партнерства. Например, сотрудничество с фитнес-клубами, тренерами и влиятельными личностями в области спорта для увеличения охвата

В последнее время широкое распространение получил контентмаркетинг. Все больше блогов, продвигающих платные услуги можно заметить в социальных сетях, особенно это явление часто встречается в сфере фитнеса. По данным опроса¹ ИСИЭЗ НИУ ВШЭ, уже более четверти российских интернет-пользователей - **28%** оформили платные подписки на какие-либо программы, приложения или онлайн-сервисы.

Коммерциализация телеграммбота для тренировок по бегу имеет высокий потенциал благодаря сочетанию растущего интереса к фитнесу и удобству использования мессенджеров. Разработка четкой стратегии монетизации и активное продвижение продукта помогут обеспечить его успешность и прибыльность на рынке.

Заключение

Разработка телеграмм-бота для тренировок по бегу представляет собой актуальный и перспективный проект, который отвечает современным тенденциям в области фитнеса и цифровых технологий. Учитывая растущее число людей, занимающихся спортом самостоятельно, а также увеличение времени, проводимого пользователями в мессенджерах, бот имеет потенциал стать востребованным инструментом для любителей бега. Интеграция искусственного интеллекта и расширенный функционал, включая индивидуальное планирование тренировок, геолокацию и визуализацию данных, делают продукт уникальным на рынке. Эти преимущества сильно выделяют его на фоне **конкурентов**.

Технологический стек, выбранный для реализации проекта, обеспечивает высокую производительность и надежность.

С учетом прогнозируемого роста мирового рынка фитнес-приложений, коммерческий успех нашего продукта, который будет генерировать доходы через подписку, рекламу и платные функции. Эти шаги обеспечат дальнейший рост и успех телеграмм-бота, способствуя популяризации бега и здорового образа жизни среди пользователей.

Библиография

Интернет-ресурсы

1. Документация библиотеки Matplotlib - <https://matplotlib.org/stable/index.html>
2. Документация библиотеки requests - <https://pythonru.com/biblioteki/kratkoe-rukovodstvo-po-bibliotekepython-requests>
3. Документация библиотеки sqlalchemy - <https://docs.sqlalchemy.org>
4. Документация библиотеки aiohttp - <https://docs.aiohttp.org/en/stable>
5. онлайн нейросеть <https://chatinfo.ru>

Книги

1. Книга «Чистый код» - Роберт Мартин.
2. Книга «Паттерны объектно-ориентированного проектирования» - Эрих Гамма, Ричард Хелм, Ральф Джонсон, Джон Влиссидес.
3. Книга «Искусство программирования» - Дональд Кнут.

¹. Опрос ИСИЭЗ НИУ ВШЭ - <https://issek.hse.ru/news/961648516.html>

Опросы

1. ВЦИОМ, октябрь 2023
2. опрос ВЦИОМ, апрель 2022
3. опрос ВЦИОМ, декабрь 2024
4. опрос ИСИЭЗ НИУ ВШЭ в апреле–мае 2024 г

Приложение

main.py

```
import asyncio
import datetime
from translate import Translator
from aiogram.fsm.context import FSMContext
from aiogram import Bot, Dispatcher, types, Router, F
from aiogram.filters import Command
from tokens import API_TOKEN
from alchemy import *
from states import Form
from other_functions import *
from buttons import *
from validators import *

bot = Bot(token=API_TOKEN)
dp = Dispatcher()
router = Router()

@router.message(Command("edit_plan"))
async def cmd_start(message: types.Message):
    await message.answer(
        "Что бы вы хотели изменить?",
        reply_markup=InlineKeyboardMarkup(
            inline_keyboard=[
                [InlineKeyboardButton(text="Частота тренировок",
callback_data="amo")],
                [InlineKeyboardButton(text="Сложность", callback_data="dif")],
            ]
        ),
    )

@router.message(Form.waiting_for_new_weight)
async def cmd_edit_weight(message: types.Message, state: FSMContext):
    weight = message.text
    if check_weight(weight):
        await message.answer(
            "Необходимо ввести одно действительное число, соответствующее вашему весу"
        )
    return
    weight_dict = await get_weight(message.chat.id)
    weight_dict = ast.literal_eval(weight_dict)
    weight_dict[datetime.datetime.today().strftime("%Y-%m-%d")] = weight
    await update_weight(message.chat.id, weight_dict)
    await plot_and_save_accidents(message.chat.id, weight_dict)
    await state.set_state("")
```

```

@router.message(Command("start"))
async def cmd_start(message: types.Message, state: FSMContext):
    user_id = message.from_user.id
    first_name = message.from_user.first_name
    user_nik = message.from_user.username
    await insert_user(user_id, first_name, user_nik)
    await message.answer(
        f"Здравствуйте, {first_name}! Пожалуйста, заполните информацию о себе,
        чтобы приступить к тренировкам."
    )
    await message.answer("Для начала введите ваш возраст:")
    await state.set_state(Form.waiting_for_age)

@router.message(Form.waiting_for_age)
async def process_age(message: types.Message, state: FSMContext):
    age = message.text
    if check_age(age):
        await message.answer(
            "Необходимо ввести одно натуральное число, соответствующее вашему
            возрасту"
        )
        return
    await update_age(message.chat.id, age)
    await state.set_state(Form.waiting_for_weight)
    await message.answer("Теперь ваш вес:")

@router.message(Form.waiting_for_weight)
async def process_age(message: types.Message, state: FSMContext):
    weight = message.text
    user_id = message.chat.id
    if check_weight(weight):
        await message.answer(
            "Необходимо ввести одно действительное число, соответствующее вашему
            весу"
        )
        return
    await update_weight(
        user_id,
        str(
            {
                f"Начало тренировок\n({datetime.datetime.today().strftime('%Y-
                %m-%d')}): weight
            }
        ),
    )
    await state.set_state(Form.waiting_for_times)
    await message.answer("Сколько раз в неделю вы планируете тренироваться?")

@router.message(Form.waiting_for_times)
async def process_age(message: types.Message, state: FSMContext):
    times = message.text
    if check_times(times):
        await message.answer("Необходимо ввести количество дней в неделю")
        return
    await update_times(message.chat.id, times)
    await message.answer(
        "Ваш пол:",
        reply_markup=InlineKeyboardMarkup(
            inline_keyboard=[
                [InlineKeyboardButton(text="Мужской", callback_data="m")],
            ]
        )
    )

```

```

        [InlineKeyboardButton(text="Женский", callback_data="f")],
    ],
)
await state.set_state("*")

@router.message(Form.waiting_for_times2)
async def process_age2(message: types.Message, state: FSMContext):
    user_id = message.chat.id
    times = message.text
    if check_times(times):
        await message.answer("Необходимо ввести количество дней в неделю")
        return
    plan = await get_plan_aiosql(user_id)
    text = solve(
        API_GPT,
        "text",
        f"{plan}.Сделать количество дней в этом плане {times}."
        f"Ответ также дать в форме словаря Python вида день недели : тренировка.
        Вывести в одну строку без обозначения языка Python",
    )
    try:
        await get_plan(user_id,
insert_newline_after_two_words(ast.literal_eval(text)))
        await update_plan(user_id, plan)
    except:
        await get_plan_run(user_id)
        await message.answer("Вот ваш новый план!")
        await state.set_state("*")

@router.callback_query(
    lambda c: c.data
    in [
        "m",
        "f",
        "Начинающий",
        "Средний",
        "Продвинутый",
        "low",
        "mid",
        "up",
        "dif",
        "amo",
        "p",
        "min",
        'ok',
    ]
)
async def process_callback_button(
    callback_query: types.CallbackQuery, state: FSMContext
):
    user_id = callback_query.message.chat.id
    msg_id = callback_query.message.message_id
    if callback_query.data == "m":
        await bot.answer_callback_query(callback_query.id)
        await update_gender(user_id, "Мужской")
        await bot.delete_message(chat_id=user_id, message_id=msg_id)
        await bot.send_message(
            callback_query.message.chat.id,
            "Ваш уровень подготовки:",
            reply_markup=level(),

```

```

)
if callback_query.data == "f":
    await bot.answer_callback_query(callback_query.id)
    await update_gender(user_id, "Женский")
    await bot.delete_message(
        chat_id=user_id,
        message_id=msg_id,
    )
    await bot.send_message(user_id, "Ваш уровень подготовки:",
reply_markup=level())
if callback_query.data == "Начинающий":
    await bot.answer_callback_query(callback_query.id)
    await update_level(user_id, "Начинающий")
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await bot.send_message(
        callback_query.message.chat.id, "Ваша цель:", reply_markup=target()
    )
if callback_query.data == "Средний":
    await bot.answer_callback_query(callback_query.id)
    await update_level(user_id, "Средний")
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await bot.send_message(
        callback_query.message.chat.id, "Ваша цель:", reply_markup=target()
    )
if callback_query.data == "Продвинутый":
    await bot.answer_callback_query(callback_query.id)
    await update_level(user_id, "Продвинутый")
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await bot.send_message(
        callback_query.message.chat.id, "Ваша цель:", reply_markup=target()
    )
if callback_query.data == "low":
    await bot.answer_callback_query(callback_query.id)
    await update_target(user_id, "Потеря лишнего веса")
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await get_user_data(user_id)
    await bot.send_message(user_id, "Составляем тренировочный план...")
    await state.set_state("*")
    await get_plan_run(user_id)
    user = await get_user_data(user_id)
    kkal = calculate_daily_caloric_intake(
        list(ast.literal_eval(user.weight_gr).values())[0], user.age,
user.per_week
    )
    await bot.send_message(
        user_id, f"Ваше суточное потребление калорий: {round(kkal - 500)}"
    )
if callback_query.data == "mid":
    await bot.answer_callback_query(callback_query.id)
    await update_target(user_id, "Поддержание формы")
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await get_user_data(user_id)
    await bot.send_message(user_id, "Составляем тренировочный план...")
    await state.set_state("*")
    await get_plan_run(user_id)
    user = await get_user_data(user_id)
    kkal = calculate_daily_caloric_intake(
        list(ast.literal_eval(user.weight_gr).values())[0], user.age,
user.per_week
    )
    await bot.send_message(
        user_id, f"Ваше суточное потребление калорий: {round(kkal)}"
    )
)

```

```

if callback_query.data == "up":
    await bot.answer_callback_query(callback_query.id)
    await update_target(user_id, "Улучшение результатов")
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await get_user_data(user_id)
    await bot.send_message(user_id, "Составляем тренировочный план...")
    await state.set_state("")
    await get_plan_run(user_id)
    user = await get_user_data(user_id)
    kkal = calculate_daily_caloric_intake(
        list(ast.literal_eval(user.weight_gr).values())[0], user.age,
user.per_week
    )
    await bot.send_message(
        user_id, f"Ваше суточное потребление калорий: {round(kkal)}"
    )
if callback_query.data == "amo":
    await bot.answer_callback_query(callback_query.id)
    await state.set_state(Form.waiting_for_times2)
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await bot.send_message(
        user_id, "Сколько раз в неделю вы планируете тренироваться?"
    )
if callback_query.data == "dif":
    await bot.answer_callback_query(callback_query.id)
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await bot.send_message(
        user_id, "Как бы вы хотели изменить план?", reply_markup=dif()
    )
if callback_query.data == "ok":
    await bot.answer_callback_query(callback_query.id)
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
if callback_query.data == "min":
    await bot.answer_callback_query(callback_query.id)
    plan = await get_plan_aiosql(user_id)
    text = solve(
        API_GPT,
        "text",
        f"{plan}.Сделать этот план на 10% легче."
        f"Ответ также дать в форме словаря Python вида день недели :
тренировка. Вывести в одну строку без обозначения языка Python",
    )
    await get_plan(
        user_id, insert_newline_after_two_words(ast.literal_eval(text))
    )
    await update_plan(user_id, text)
    await callback_query.message.answer("Вот ваш новый план!")
    await bot.delete_message(chat_id=user_id, message_id=msg_id)
    await state.set_state("")
if callback_query.data == "p":
    await bot.answer_callback_query(callback_query.id)
    await bot.answer_callback_query(callback_query.id)
    plan = await get_plan_aiosql(user_id)
    text = solve(
        API_GPT,
        "text",
        f"{plan}.Сделать этот план на 10% сложнее."
        f"Ответ также дать в форме словаря Python вида день недели :
тренировка. Вывести в одну строку без обозначения языка Python",
    )
    await get_plan(
        user_id, insert_newline_after_two_words(ast.literal_eval(text))
    )

```

```

        await update_plan(user_id, text)
        await callback_query.message.answer("Вот ваш новый план!")
        await bot.delete_message(chat_id=user_id, message_id=msg_id)
        await state.set_state("")

async def main():
    await create()
    dp.include_router(router)
    start_scheduler()
    await dp.start_polling(bot)

@router.message(Command("get_location"))
async def cmd_edit_weight(message: types.Message):
    await message.answer("Вы можете отправить вашу геолокацию. "
                          "Вам будут доступны сведения о погоде и пользователях"
                          "рядом",
                          reply_markup=InlineKeyboardMarkup(inline_keyboard=[[InlineKeyboardButton(text="Понятно", callback_data="ok")]]))

@router.message(Command("edit_weight"))
async def cmd_edit_weight(message: types.Message, state: FSMContext):
    await message.answer("Введите ваш вес на данный момент:")
    await state.set_state(Form.waiting_for_new_weight)

@router.message(F.location)
async def handle_location(message: types.Message):
    latitude = message.location.latitude
    longitude = message.location.longitude
    geo = str(str(latitude)+' ', '+str(longitude))
    await save_location(message.from_user.id, geo)
    await bot.send_message(message.from_user.id, 'Геолокация сохранена')
    list_of_users = await get_people()
    awnser_list = ''
    for i in list_of_users:
        if 1 >= haversine(geo, i.geo) and haversine(geo, i.geo) != 0:
            if i.user_nik != None:
                awnser_list += str(i.name) + ', '+str(i.age)+' ->
@'+str(i.user_nik) + '\n'
    if awnser_list == '':
        await bot.send_message(message.from_user.id, f'К сожалению, список пользователей в вашем районе пуст')
    else:
        await bot.send_message(message.from_user.id, f'Вот список пользователей в вашем районе:\n \n{awnser_list}')

from apscheduler.schedulers.asyncio import AsyncIOScheduler

def start_scheduler():
    scheduler = AsyncIOScheduler()
    scheduler.add_job(training_day, 'cron', hour=7, minute=0)
    scheduler.start()

async def training_day():
    async with AsyncSessionLocal() as session:
        today = datetime.datetime.now().strftime('%A').lower()
        today = Translator(from_lang='english',

```

```

to_lang='russian').translate(today)
    today = today[0].upper()+today[1:]
    print(today)
    users_with_training_today = await
session.execute(select(User.user_id).where(User.plan.contains(today)))
    users_with_training_today =
users_with_training_today.scalars().all()
    print(users_with_training_today)
    for i in users_with_training_today:
        geo = await get_geo(i)
        if geo != None:
            l1, l2 = geo.split(', ')[0], geo.split(', ')[1]
            weather = await get_weather(l1, l2)
            await bot.send_message(i, f'У вас сегодня тренировка! На
улице {weather}')
        else:
            await bot.send_message(i, f'У вас сегодня тренировка!
{today}, {datetime.datetime.today().strftime("%Y-%m-%d")}')

if __name__ == "__main__":
    asyncio.run(main())

```

alchemy.py

```

from sqlalchemy import Column, Integer, String, Text, select, update, insert,
delete
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker
from sqlalchemy.ext.asyncio import AsyncSession, create_async_engine

Base = declarative_base()
DATABASE_URL = "sqlite+aiosqlite:///users.db"
engine = create_async_engine(DATABASE_URL, echo=True)

class User(Base):
    __tablename__ = "users"
    user_id = Column(Integer, primary_key=True, unique=True)
    name = Column(String)
    age = Column(Integer, default=None)
    per_week = Column(Integer, default=None)
    gender = Column(Text, default=None)
    level = Column(Integer, default=None)
    target = Column(String, default=None)
    plan = Column(String, default=None)
    weight_gr = Column(String, default=None)
    user_nik = Column(String, unique=True)
    geo = Column(String, default=None)

async def create():
    async with engine.begin() as conn:
        await conn.run_sync(Base.metadata.create_all)

AsyncSessionLocal = sessionmaker(
    bind=engine, class_=AsyncSession, expire_on_commit=False
)

```

```

async def get_weight(user_id):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            result = await session.execute(
                select(User.weight_gr).where(User.user_id == user_id)
            )
            weight = result.scalars().first()
            return weight

async def get_geo(user_id):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            result = await session.execute(
                select(User.geo).where(User.user_id == user_id)
            )
            geo = result.scalars().first()
            return geo

async def get_user_data(user_id):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            result = await session.execute(select(User).where(User.user_id ==
user_id))
            weight = result.scalars().first()
            return weight

async def get_people():
    async with AsyncSessionLocal() as session:
        async with session.begin():
            result = await session.execute(
                select(User)
            )
            people = result.scalars().all()
            return people

async def get_plan_aiosql(user_id):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            result = await session.execute(
                select(User.plan).where(User.user_id == user_id)
            )
            plan = result.scalars().first()
            return plan

async def update_weight(user_id, weight_dict):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = (
                update(User)
                .where(User.user_id == user_id)
                .values(weight_gr=str(weight_dict))
            )
            await session.execute(stmt)
            await session.commit()

async def save_location(user_id, geo):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = (
                update(User)
                .where(User.user_id == user_id)

```

```

        .values(geo=str(geo))
    )
    await session.execute(stmt)
    await session.commit()

async def update_gender(user_id, gender):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = (
                update(User).where(User.user_id ==
user_id).values(gender=str(gender))
            )
            await session.execute(stmt)
            await session.commit()

async def update_target(user_id, target):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = (
                update(User).where(User.user_id ==
user_id).values(target=str(target))
            )
            await session.execute(stmt)
            await session.commit()

async def update_level(user_id, level):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = update(User).where(User.user_id ==
user_id).values(level=str(level))
            await session.execute(stmt)
            await session.commit()

async def update_times(user_id, times):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = (
                update(User).where(User.user_id ==
user_id).values(per_week=int(times))
            )
            await session.execute(stmt)
            await session.commit()

async def update_plan(user_id, plan):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = update(User).where(User.user_id == user_id).values(plan=plan)
            await session.execute(stmt)
            await session.commit()

async def update_age(user_id, age):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            stmt = update(User).where(User.user_id ==
user_id).values(age=int(age))
            await session.execute(stmt)
            await session.commit()

```

```

async def insert_user(user_id, first_name, user_nik):
    async with AsyncSessionLocal() as session:
        async with session.begin():
            try:
                stmt = insert(User).values(
                    user_id=user_id, name=first_name, user_nik=user_nik
                )
                await session.execute(stmt)
                await session.commit()
            except:
                await session.execute(delete(User).where(User.user_id==user_id))
                await session.commit()
                await insert_user(user_id, first_name, user_nik)

```

validators.py

```

def check_weight(weight):
    try:
        weight = float(weight)
        if weight <= 20 or weight >= 250:
            raise Exception
    except:
        return True
    return False

def check_age(age):
    try:
        age = int(age)
        if age <= 1 or age >= 150:
            raise Exception
    except:
        return True
    return False

def check_times(times):
    try:
        times = int(times)
        if times > 7 or times < 1:
            raise Exception
    except:
        return True
    return False

```

users.db – файл базы данных

tokens.py

```

API_TOKEN = "8093558481:AAEeaghG7hKwgJR6D-CpdNao80ybOZbKEHQ"
API_GPT = "92d00bef-cb99-42dc-8111-c9bale9ec782"

```

other_functions.py

```

import matplotlib.pyplot as plt
from aiogram.types import FSInputFile
from main import bot
import os
import ast
import math
from PIL import Image, ImageDraw, ImageFont
from alchemy import get_user_data, update_plan
from gpt_work import solve

```

```

from tokens import API_GPT
import aiohttp
import re

async def get_weather(l1, l2):
    url = f'https://yandex.ru/pogoda/ru-
RU/details/auto/today?lat={l1}&lon={l2}&lang=ru&via=dnav'

    async with aiohttp.ClientSession() as session:
        async with session.get(url) as response:
            res = await response.text()
            pattern = r'<p class="AllY_visuallyHidden__y0sw0
visuallyHidden">0:00:s*(.*?)</p>'
            match = re.search(pattern, res)
            if match:
                return match.group(1)
            else:
                return None

def insert_newline_after_two_words(input_dict):
    output_dict = {}

    for key, value in input_dict.items():
        # Разделяем строку на слова
        words = value.split()
        # Создаем новый список для хранения измененных слов
        modified_words = []

        # Проходим по словам и добавляем \n после каждых двух
        for i in range(len(words)):
            modified_words.append(words[i])
            # Проверяем, нужно ли добавить \n
            if (i + 1) % 2 == 0:
                modified_words.append("\n")

        # Объединяем слова обратно в строку
        output_dict[key] = " ".join(modified_words)

    return output_dict

async def plot_and_save_accidents(id, data, filename="accidents_plot.png"):
    years = list(data.keys())
    accidents = [float(i) for i in list(data.values())]

    # Создаем график
    plt.figure(figsize=(10, 5))
    plt.plot(years, accidents, marker="o", linestyle="--", color="b")

    # Добавляем заголовок и метки осей
    plt.title("График изменений массы тела")
    plt.xlabel("Число")
    plt.ylabel("Вес (кг)")

    # Добавляем сетку для удобства восприятия
    plt.grid()

    # Сохраняем график как изображение
    plt.savefig(filename)
    photo_file = FSInputFile(filename)
    await bot.send_photo(chat_id=id, photo=photo_file)
    # Закрываем график, чтобы освободить память

```

```

plt.close()
if os.path.exists('accidents_plot.png'):
    os.remove('accidents_plot.png')

def calculate_daily_caloric_intake(weight, age, training_sessions_per_week):

    bmr = 633 + (13.397 * float(weight)) - (5.677 * int(age))

    # Определяем коэффициент активности
    if training_sessions_per_week == 0:
        activity_multiplier = 1.2 # Малоподвижный образ жизни
    elif training_sessions_per_week <= 2:
        activity_multiplier = 1.375 # Легкие тренировки
    elif training_sessions_per_week <= 5:
        activity_multiplier = 1.55 # Умеренные тренировки
    else:
        activity_multiplier = 1.725 # Активные тренировки

    # Рассчитываем суточное потребление калорий
    daily_caloric_intake = bmr * activity_multiplier
    return daily_caloric_intake

async def get_plan(id, list_from_ai):
    data = [[], []]
    data[0] = list_from_ai.keys()
    data[1] = list_from_ai.values()
    # Настройки изображения
    cell_width = 270
    cell_height = 350
    table_width = cell_width * len(data[0])
    table_height = cell_height * len(data)

    # Создание нового изображения
    image = Image.new("RGB", (table_width, table_height), "white")
    draw = ImageDraw.Draw(image)

    # Загрузка шрифта (укажите путь к вашему .ttf файлу)
    font_path = "arial.ttf" # Убедитесь, что файл находится в той же папке или
укажите полный путь
    font_size = 20
    font = ImageFont.truetype(font_path, font_size)

    # Рисуем таблицу
    for i, row in enumerate(data):
        for j, value in enumerate(row):
            x0 = j * cell_width
            y0 = i * cell_height
            x1 = x0 + cell_width
            y1 = y0 + cell_height

            # Рисуем прямоугольник для ячейки
            draw.rectangle([x0, y0, x1, y1], outline="black")

            # Рисуем текст в ячейке
            draw.text((x0 + 5, y0 + 5), str(value), fill="black", font=font)
    image.save("table_pillow.jpg")
    # Сохранение изображения в файл
    photo_file = FSInputFile("table_pillow.jpg")
    await bot.send_photo(chat_id=id, photo=photo_file)
    if os.path.exists("table_pillow.jpg"):
        os.remove("table_pillow.jpg")

```

```

return

async def get_plan_run(id):
    user = await get_user_data(id)
    text = solve(
        API_GPT,
        "text",
        f"Составить план тренировок на {user.per_week} дней в неделю по бегу
включая параметры:"
        f" возраст {user.age}л, вес
{list(ast.literal_eval(user.weight_gr).values())[0]}кг, пол {user.gender}, "
        f"уровень спортсмена {user.level}, цель тренировок {user.target}. Ответ
дать в форме словаря Python вида день недели : тренировка. Вывести в одну строку
без обозначения языка Python. "
        f"Дни без тренировок не писать",
    )
    try:
        await get_plan(id,
insert_newline_after_two_words(ast.literal_eval(text)))
        await update_plan(id, text)
    except:
        await get_plan_run(id)

def haversine(coord1, coord2):
    # Радиус Земли в километрах
    R = 6371.0

    lat1, lon1 = coord1.split(' ')
    lat2, lon2 = coord2.split(' ')
    lat1, lat2, lon1, lon2 = float(lat1), float(lat2), float(lon1), float(lon2)

    # Преобразуем градусы в радианы
    lat1_rad = math.radians(lat1)
    lon1_rad = math.radians(lon1)
    lat2_rad = math.radians(lat2)
    lon2_rad = math.radians(lon2)

    # Разница координат
    dlat = lat2_rad - lat1_rad
    dlon = lon2_rad - lon1_rad

    # Формула haversine
    a = (math.sin(dlat / 2) ** 2 +
        math.cos(lat1_rad) * math.cos(lat2_rad) * math.sin(dlon / 2) ** 2)
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))

    distance = R * c
    return distance

```

gpt_work.py

```

import time
import requests
import json

def create_request(key, mode, content):
    url = "https://chatinfo.ru/api/gpt/create_request"
    parameters = {"key": key, "mode": mode, "content": content}
    payload = json.dumps(parameters)
    response = requests.post(url, data=payload)

```

```

return response.json()

def get_response(key, id):
    url = "https://chatinfo.ru/api/gpt/get_response"
    parameters = {"key": key, "id": id}
    payload = json.dumps(parameters)
    response = requests.post(url, data=payload)

    return response.json()

def solve(key, mode, content):
    # Шаг 1: создаем запрос
    request_info = create_request(key, mode, content)

    if "id" not in request_info:
        print("Failed to create request. Error: " + request_info["error"] + ".")
        return None

    request_id = request_info["id"]

    print("Created request with id " + str(request_id) + ".")
    time.sleep(2)

    # Шаг 2: ждем ответа
    max_attempts = 80
    attempts = 0

    while attempts < max_attempts:
        response_info = get_response(key, request_id)

        if "success" in response_info and response_info["success"]:
            return response_info["result"]
        elif "solving" in response_info and response_info["solving"]:
            time.sleep(2)
            attempts += 1
        else:
            print("Failed to get response. Error: " + response_info["error"] + ".")
            return None

    print("Timed out waiting for response.")
    return None

```

buttons.py

```

from aiogram.types import InlineKeyboardMarkup, InlineKeyboardButton

def level():
    inline_kb_list = [
        [InlineKeyboardButton(text="Начинающий", callback_data="Начинающий")],
        [InlineKeyboardButton(text="Средний", callback_data="Средний")],
        [InlineKeyboardButton(text="Продвинутый", callback_data="Продвинутый")],
    ]
    return InlineKeyboardMarkup(inline_keyboard=inline_kb_list)

def dif():
    inline_kb_list = [

```

```
[InlineKeyboardButton(text="Увеличить нагрузки", callback_data="p")],
[InlineKeyboardButton(text="Снизить нагрузки", callback_data="min")],
]
return InlineKeyboardMarkup(inline_keyboard=inline_kb_list)

def target():
    inline_kb_list = [
        [InlineKeyboardButton(text="Потеря лишнего веса", callback_data="low")],
        [InlineKeyboardButton(text="Поддержание формы", callback_data="mid")],
        [InlineKeyboardButton(text="Улучшение результатов",
callback_data="up")],
    ]
    return InlineKeyboardMarkup(inline_keyboard=inline_kb_list)
```

states.py

```
from aiogram.fsm.state import StatesGroup, State

class Form(StatesGroup):
    waiting_for_weight = State()
    waiting_for_age = State()
    waiting_for_new_weight = State()
    waiting_for_times = State()
    waiting_for_times2 = State()
    waiting_data = State()
```

СИСТЕМА ПРЕДВАРИТЕЛЬНОГО СМЕШИВАНИЯ ФИЛАМЕНТА ДЛЯ 3D ПЕЧАТИ

Бояринцев Аркадий Сергеевич

Балашиха г, МБОУ"Лицей", 11 класс

Секция: Робототехника и комплексная автоматизация

Работа выполнена без научного руководителя

Аннотация

Работа посвящена актуальной задаче современной 3D печати, связанной с печатью моделей на FDM 3D принтерах используя пластик разных цветов и свойств.

Целью работы является разработка Системы смешивания филамента, позволяющую изготовить нить, состоящую из нескольких пластиков, соответствующую G-code модели, для безостановочной многоцветной печати моделей на любом стандартном 3D принтере.

В результате проделанной работы:

- создана действующая установка, обеспечивающая стабильное производство нити с заданными переходами цветов, соответствующими исходной модели;
- разработано программное обеспечение управления микроконтроллером;
- спроектирована система управления установкой, на основе анализа G-code печатаемой модели.

В работе изложены этапы развития проекта с использованием метода последовательных экспериментов, подробно описана конструкция и алгоритмы работы установки, описаны технические задачи, потребовавшие решения. Продемонстрированы результаты работы – модели, фактически изготовленные с применением предложенной системы.

Разработанная система позволяет создавать множество различных сочетаний цветов и материалов, не требуя доработки конструкции 3D принтера. Данные преимущества полезны для быстрого прототипирования, печати дизайнерских моделей, а также деталей со специфическими механическими свойствами.

ВВЕДЕНИЕ

В настоящее время 3D печать очень быстро развивается. Одна из самых популярных технологий 3D печати – печать методом послойного наплавления полимерных нитей (FDM/FFF). На большинстве FDM/FFF 3D принтеров изначально установлено только одно сопло, поэтому, чтобы иметь возможность использовать несколько пластиков в одной модели, приходится применять дополнительные устройства для смены нити во время печати.

Цель моей работы – создать Систему смешивания филамента (далее – ССФ), позволяющую изготовить нить, состоящую из нескольких пластиков, соответствующую G-code модели, для безостановочной многоцветной печати моделей, на любом стандартном 3D принтере.

Задачи проекта:

1. создать прототип устройства для проверки идеи;
2. создать программное обеспечение для управления устройством;
3. создать программу управления ССФ для автоматического изготовления нити под конкретную 3D модель на основе её G-code;
4. создать действующее устройство для автоматизированного производства нити.

Для решения этих задач я использовал метод последовательных экспериментов [1]:

- производство тестовой нити;
- печать модели с использованием произведённой нити;
- оценка результатов печати;
- доработка и настройка установки;
- повторение печати.

ОСНОВНАЯ ЧАСТЬ

Обзор существующих решений

На данный момент на рынке существует несколько технологий, позволяющих печатать разными филаментами на одном принтере:

1. Печатающие головы с большим количеством сопел. – имеет несколько отдельных каналов подачи нитей на одной голове, утяжеляет и усложняет конструкцию головы.
2. Системы автоматической смены инструмента (автоматически сменяет печатающую голову) – позволяет использовать большое количество нитей при этом, относительно других вариантов меньше замедляет печать, усложняет конструкцию принтера, могут быть проблемы со смещением инструментов

3. Системы смены филамента путём выгрузки и загрузки нити (например, Bambu Lab AMS, Prusa MMU) – позволяет использовать большое количество нитей при этом, относительно других вариантов значительно замедляет печать.
4. Смешивающий хотэнд (Diamond hotend и ему подобные) – позволяет смешивать пластики, утяжеляет голову принтера.
5. IDEX системы (несколько независимых печатающих голов, размещённых на одном принтере) – усложняет конструкцию принтера, могут быть проблемы со смещением инструментов
6. Системы соединения филамента (Mosaic Palette)– автоматически отрезает и соединяет нить. - быстро сменяет цвета, могут возникать проблемы при использовании плохо сплавляемых пластиков.

На рисунке 1 представлены примеры вышеуказанных технологий, цифры под картинками соответствуют номерам в списке.

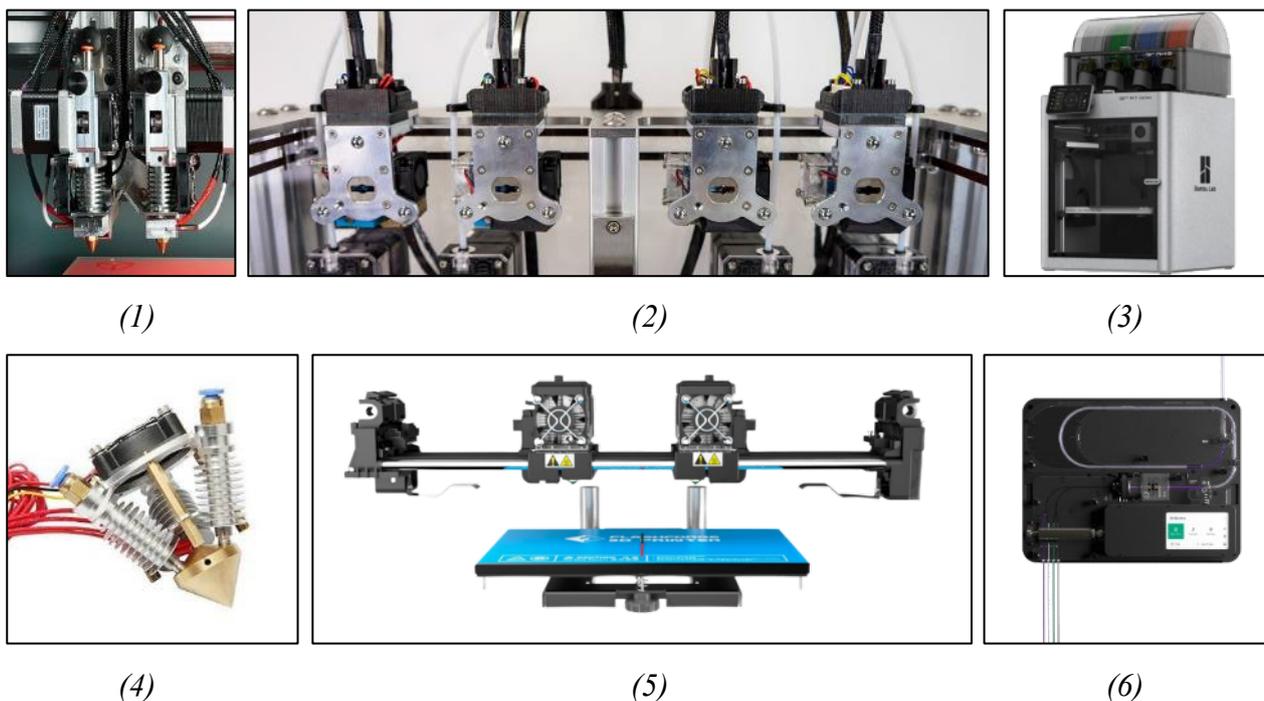


Рисунок 1 – Существующие решения систем смены филамента

Описание предлагаемой мной технологии

- Основная идея заключается в производстве нити на отдельной установке и подаче готовой нити с необходимыми переходами пластиков в принтер.
- Ключевой элемент, обеспечивающий смешивание пластиков – нагревательная камера (смешивающий хотэнд), в которую подают исходные нити в заданных пропорциях, и где происходит нагрев, смешивание и выдавливание пластика через сопло.

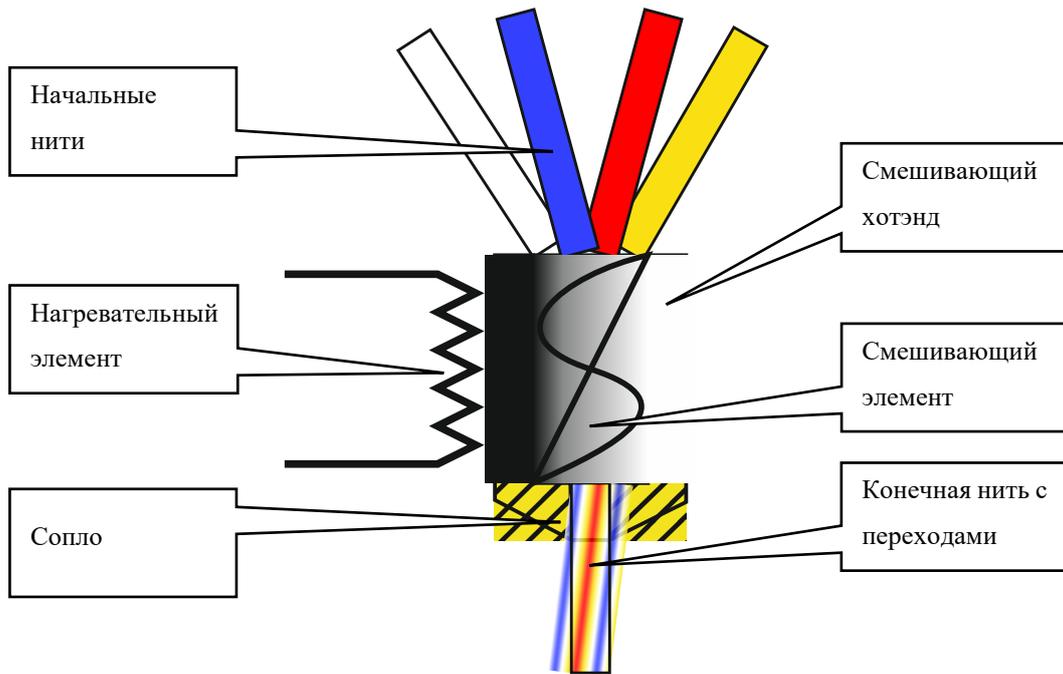


Рисунок 2 – Схема смешивающего хотэнда

- Так как смешивание пластика и формирование нити происходит до подачи в принтер, для соответствия печатаемой модели и подаваемого филамента, система автоматически анализирует G-code модели и на его основе управляет смешиванием пластика, и дает команду моторам подать требуемый пластик в запланированный момент печати.
- За счёт отсутствия необходимости прямого подключения к принтеру, данную систему можно применять почти со всеми 3D принтерами, печатающими по технологии FFF/FDM.

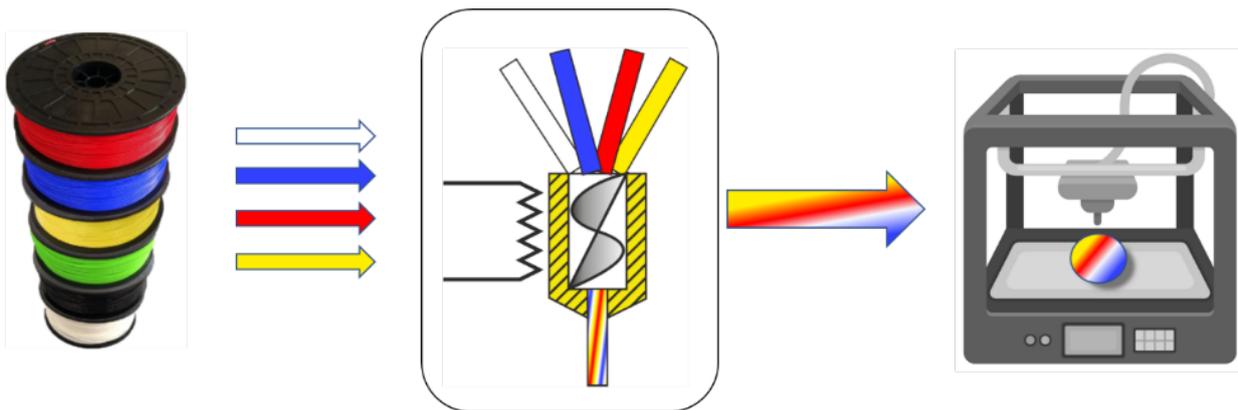


Рисунок 3 – Идея работы установки

Состав установки

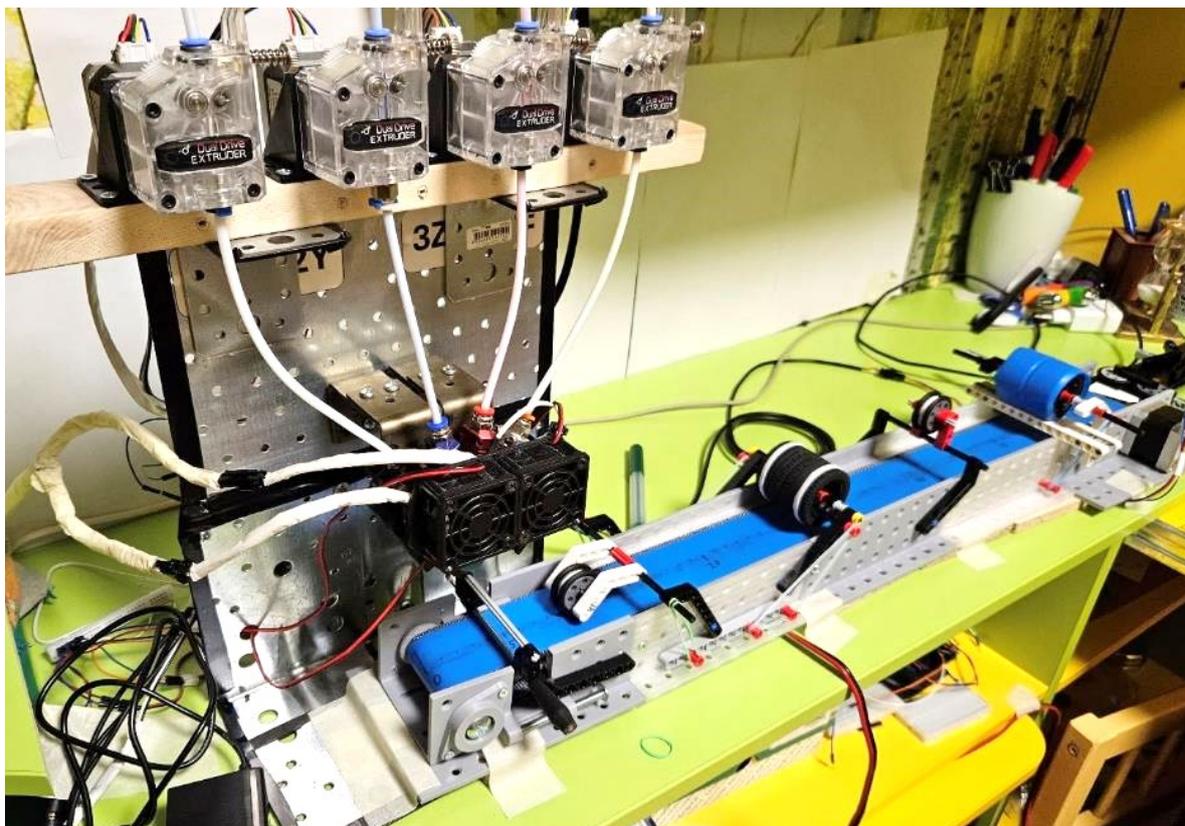


Рисунок 4 – Установка ССФ в сборе

Установка состоит из следующих блоков:

Каркас

Каркас ССФ собран на перфорированной металлической пластине и усиленных крепёжных уголках.

Для крепления экструдеров с моторами я использовал деревянный брусок.

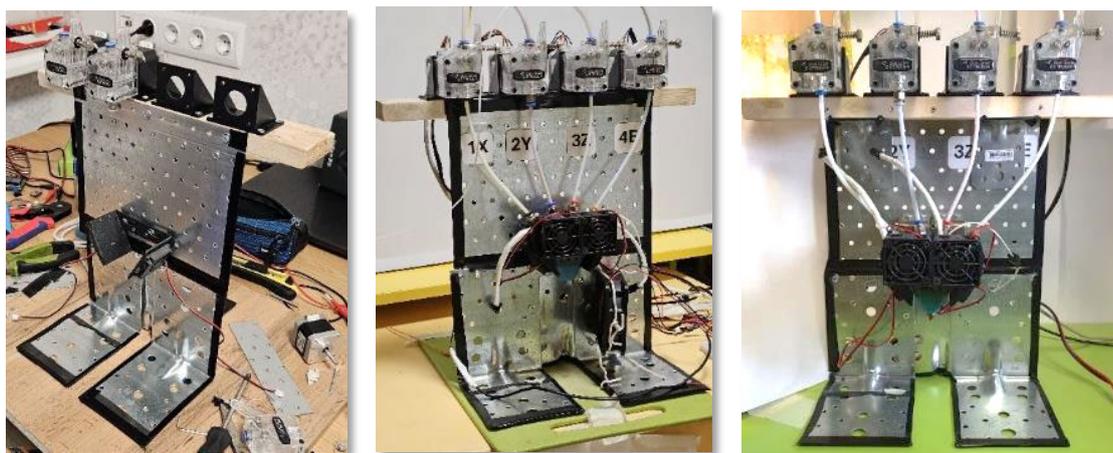


Рисунок 5 – Каркас установки

Система подачи нити

Система подачи нити собрана из 4 экструдеров (аналог BMG Dual Drive) с установленными на них шаговыми моторами HANPOSE 17HS4401. На выходе экструдеров

установлены PTFE трубки, направляющие нить на вход смешивающего хотэнда. Также мною был разработан, напечатан и собран отдельный держатель для катушек.

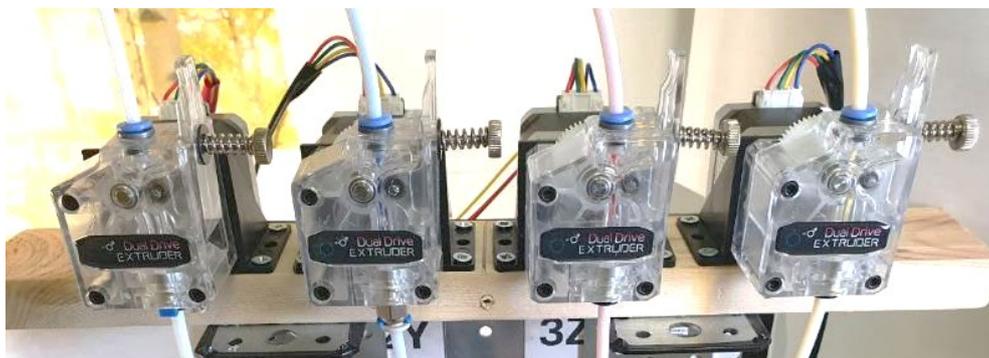


Рисунок 6 –Экструдеры с шаговыми моторами



Рисунок 7 –Держатель катушек с пластиком

Блок нагрева и охлаждения

Для нагрева смешивающего хотэнда я использую два керамических нагревательных элемента 24 Вольта x 50 ватт.

Нагреватели подключены к плате управления через твердотельное реле, на которое подаётся питающее напряжение.

Охлаждение зоны подачи нитей выполнено с использованием двух осевых вентиляторов. Охлаждение зоны выхода нити выполнено с использованием двух радиальных вентиляторов.

Блок смешивания (смешивающий хотэнд)

В качестве смешивающего хотэнда я использовал решение, предлагаемое компанией XCR3D, состоящее из базового латунного корпуса с 4-мя установленными на них радиаторами. Выходное сопло заменил на конструкцию собственной разработки (смотрите раздел 6).



Рисунок 8 – Смешивающий хотэнд

Конвейер (Система стабилизации нити)

В процессе работы выявлено, что для производства нити со стабильным диаметром, без утолщений и утонений принципиально важно обеспечить равномерное вытягивание нити до момента остывания.

Для решения этой задачи я решил использовать охлаждающий конвейер. Из сопла смешивающего хотэнда нить, в еще горячем (пластичном) состоянии, попадает на ленту конвейера и остывает при движении по конвейеру. При этом скорость движения конвейера относительно суммарной скорости подачи нитей напрямую влияет на диаметр получаемой нити.

Если диаметры входных нитей равны и скорость конвейера равна суммарной скорости подачи нитей, то диаметр получаемой на выходе из установки нити равен диаметру входной нити.

Так как я использовал исходные нити стандартного диаметра 1.75 мм, то на выходе установки получал схожий диаметр результирующей нити, в пределах 1.66–1.80 мм, что приемлемо для использования при тестировании и пригодно для печати на 3D принтере.

Конструкция конвейера состоит из следующих элементов:

- «Головы» конвейера с системой натяжения ленты;
- 3-х одинаковых модулей средней части конвейера;
- «хвоста» конвейера с шаговым двигателем;
- передний и задний валы конвейера через винты (втулки) $\text{D} 8$ мм установлены на стандартных подшипниках RS 608;
- лента конвейера сшита из текстильной стропы шириной 50 мм. Длина конвейерной ленты – 1350 мм. В модификации конвейера поверх текстильной

ленты я поместил дополнительную резиновую ленту для устранения проскальзывания нити по конвейеру;

- Для центрирования и прижима нити на конвейер были установлены дополнительные навесные элементы из пластиковых деталей с резиновыми накладками.

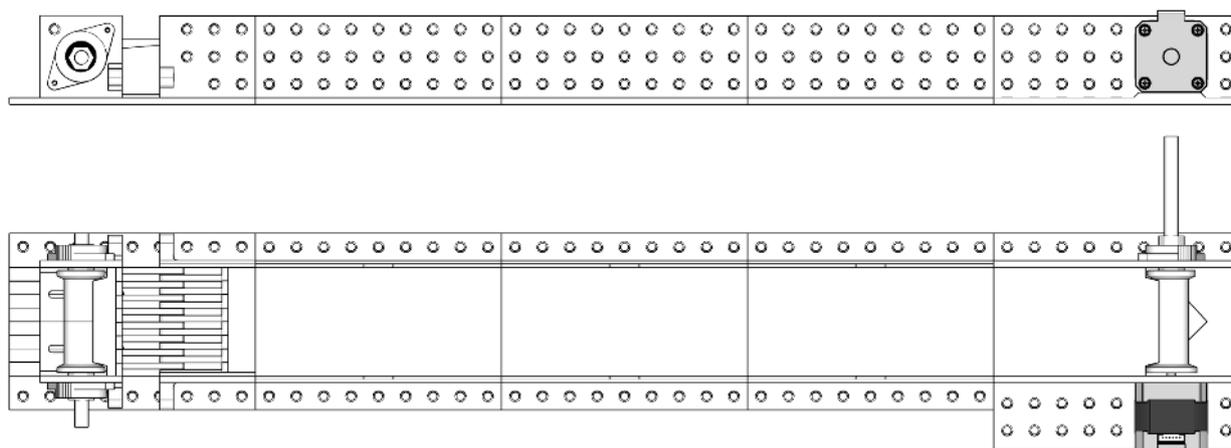


Рисунок 9 – Модель конвейера вид спереди и вид сверху

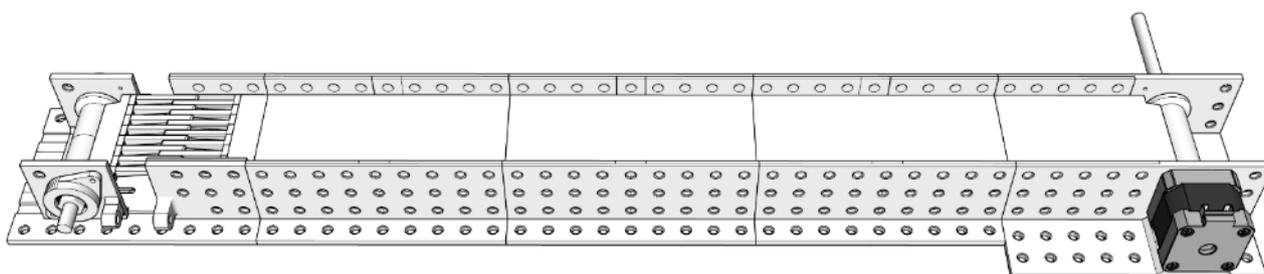


Рисунок 10 – 3Д модель конвейера

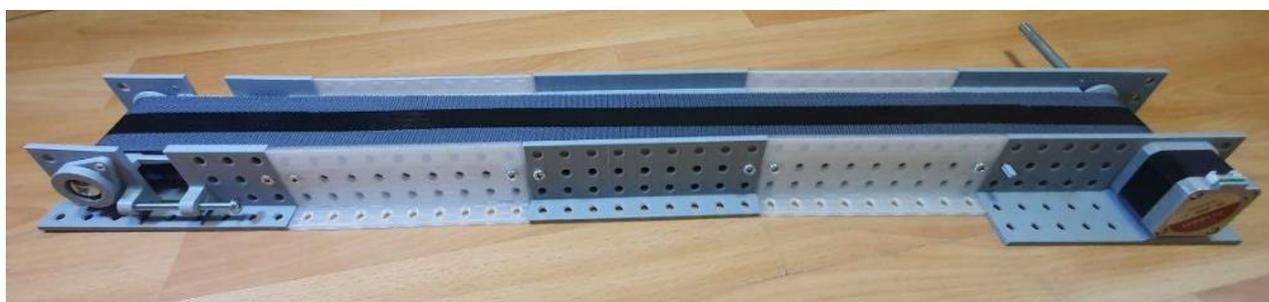


Рисунок 11 – Конвейер в сборе с текстильной лентой



Рисунок 12 – Конвейер в сборе с дополнительной резиновой лентой

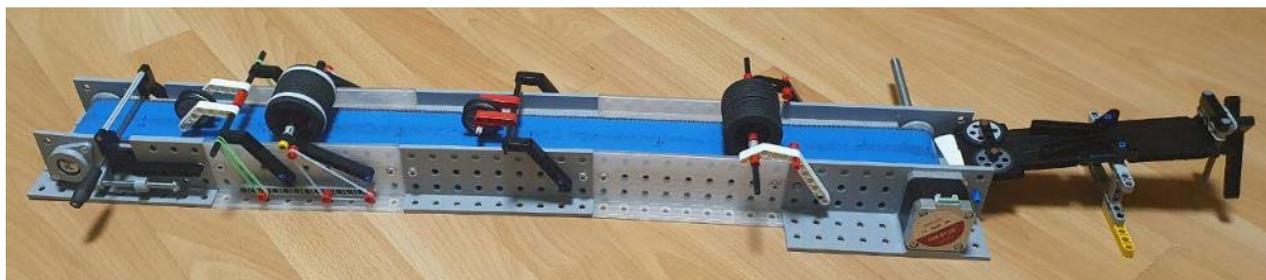


Рисунок 13 – Конвейер в сборе с навесными элементами для центрирования и прижима нити

Система электропитания

Для электропитания установки я использую 3 блока питания:

- 1) Блок питания 12В x 2А для питания платы микроконтроллера Arduino,
- 2) Блок питания 24В x 3А – для питания вентиляторов охлаждения
- 3) Лабораторный блок питания 0-30В x 10А – для питания нагревательных элементов смешивающего хотэнда;

Чтобы избежать повреждения платы микроконтроллера от больших токов (до 5 ампер при напряжении 24 вольта) питание нагревательных элементов реализовано через твердотельное реле SSR-25DD.



Рисунок 14 – Блоки питания и твердотельное реле

Управляющая часть

Управляющая часть собрана на основе аналога Arduino MEGA с установленным на ней модулем RAMPS 1.6 [2] и 5-ю драйверами шаговых двигателей A4988.

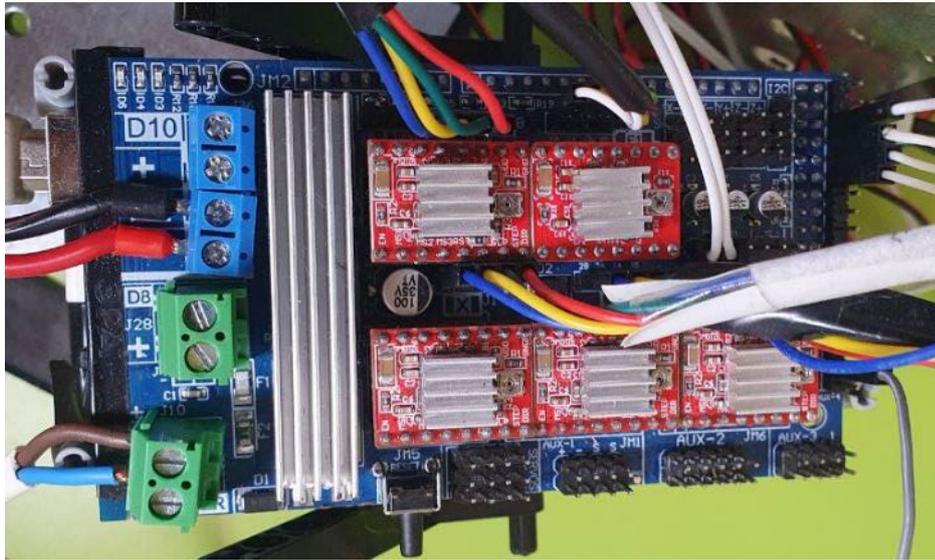


Рисунок 15 – Arduino mega с установленным модулем RAMPS 1.6, драйверами
Программа микроконтроллера

Блок – схема программы управления микроконтроллера приведена на Рисунке 5. При составлении программы я использовал библиотеки Arduino: GyverPlanner2 [4], EncButton [5], thermistor [6], GyverRelay [7]. Полный текст программы управления микроконтроллером приведён в Приложении А.

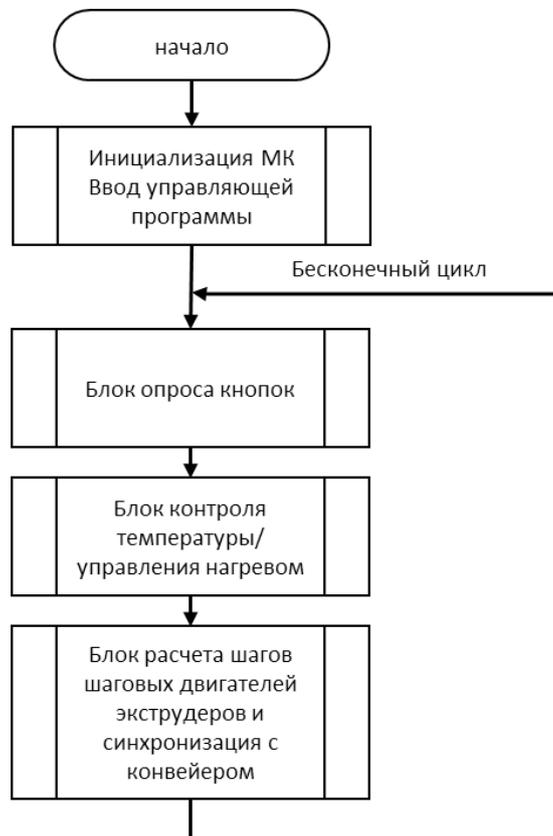


Рисунок 16 – Блок-схема программы микроконтроллера

Программа анализа G-code

Полный текст программы анализа G-code и генерации управляющей последовательности ССФ приведён в Приложении Б.

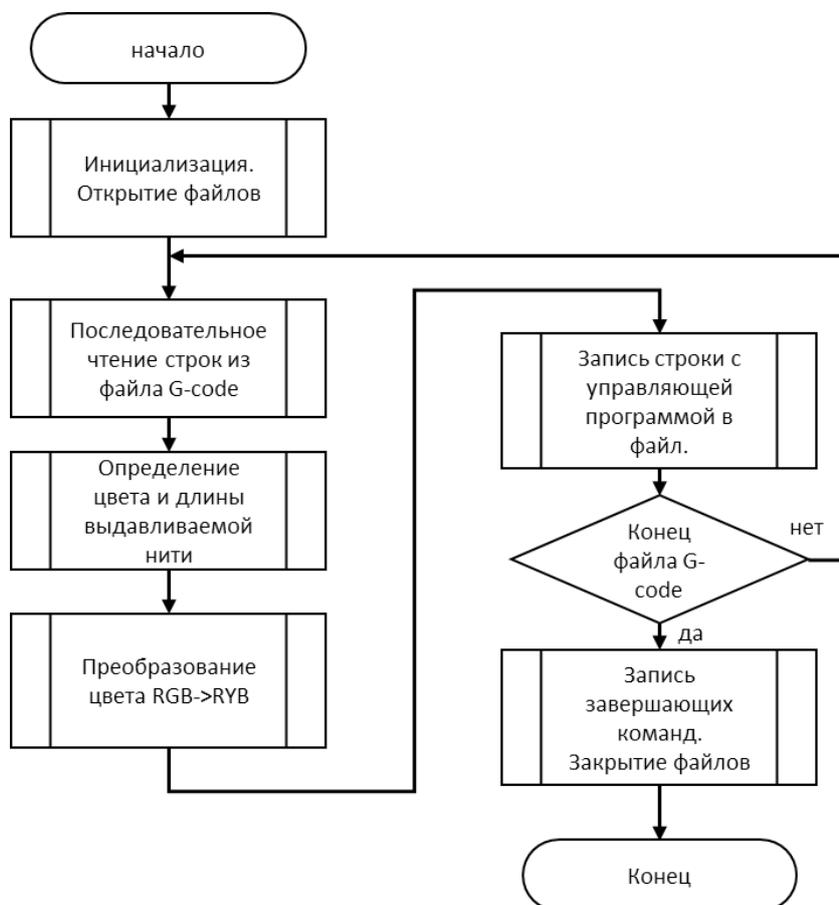


Рисунок 17 – Блок-схема программы анализа G-code

Смешиватель

Если подавать нити в зону нагрева без принудительного перемешивания, то результирующая нить будет состоять из нескольких параллельных линий пластика. В хотэнде 3D принтера смешивание также не происходит.



Рисунок 18 – Срез результирующей нити без смешивания

Для смешивания необходимо принудительно перемешивать пластик. Для этой цели я применяю статический миксер, [10] помещенный в сопло хотэнда. На качество смешивания пластика значительно влияет количество элементов смешивателя. При использовании более

длинного хотэнда можно использовать большее количество элементов смешивателя, достичь более качественного смешивания пластика.

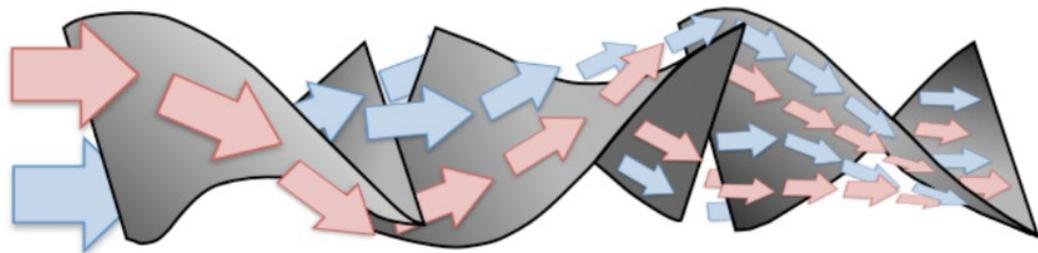


Рисунок 19 – Принцип работы статического миксера



Рисунок 20 – Степень смешивания в зависимости от числа элементов

В процессе работы я подбирал различные варианты изготовления смешивающего элемента, использовал разные варианты материалов (медь, нержавеющую сталь, оцинкованную ленту).



Рисунок 21 – Смешивающий элемент из разных материалов

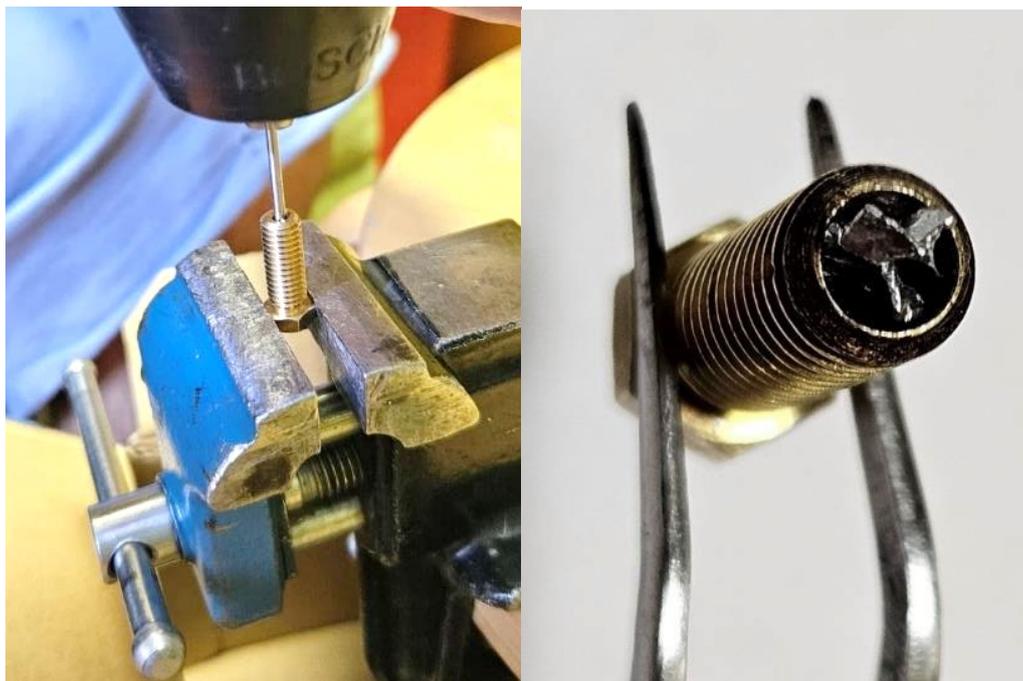


Рисунок 22 – Изготовление хотэнда

На момент написания работы я использую вариант смешивателя, приведённый на Рисунке 23.



Рисунок 23 – Используемый вариант смешивателя

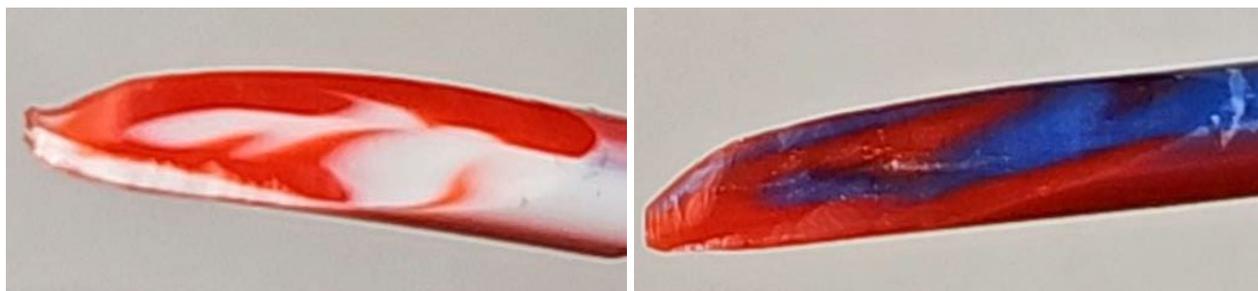


Рисунок 24 – Срез нити

На рисунке 24 показана достигнутая степень смешивания при использовании моего последнего варианта хотэнда. Хотя смешивание стало гораздо лучше, полного смешивания до состояния однородной смеси все еще не происходит. Это связано с ограничениями производства смешивателя. Для повышения однородности смеси необходимо увеличить длину смешивающей камеры и добавить большее количество элементов статического миксера.

Необходимо учитывать, если в системе стоит сопло со смешивателем, это ограничивает производство нити с резкими переходами. Резкие переходы возможно сделать путём выдавливания дополнительного объёма пластика в местах переходов при печати.

Этапы развития проекта, версии установки

Первая версия. Прототип для проверки идеи

Создавалась для проверки идеи и отладки базовых элементов. В этой версии температурой управлял вручную, комплектное сопло смешивающего хотэнда, написана первая программа управления шаговыми двигателями (задавалась скорость вращения каждого двигателя). В результате экспериментов были получены нити, содержащие несколько пластиков, нестабильного диаметра (0,5-3мм), из-за чего их практическое использование было невозможно. Также полноценного смешивания нитей не происходило. В последующих версиях описаны внесённые изменения.

Вторая версия. Первые нити.

Создан ручной стабилизатор диаметра, состоящий из катушки и ручки для вращения человеком. Добавлено больше вентиляторов обдува нити. В результате стабильность диаметра улучшилась. В связи с повышенной температурой нити в местах соединения нескольких нитей образцы неравномерно сгибались, что делало невозможным их дальнейшее использование.



Рисунок 25 – Первые нити

Третья версия. Первые напечатанные модели.

Система стабилизации была составлена из двух склеенных листов бумаги, изогнутых и расположенных под углом к горизонту, также пользователь придерживал нить рукой для уменьшения скорости её падения. Путём подбора угла, изгиба листов и силы удержания пластика пользователем, удалось достичь улучшения стабильности диаметра и производства нити, достаточного качества для подачи в принтер. Напечатаны первые модели. На этой

версии установки я также экспериментировал разными вариантами подачи пластика, в том числе двухкратной подачи пластика через ССФ для увеличения степени смешивания.



Рисунок 26 – Первые напечатанные модели

Четвёртая версия. Конвейер и управление температурой.

Разработан, напечатан на 3D принтере и собран конвейер для стабилизации нити. Добавлено автоматическое управление температурой на основе твердотельного реле. Изменена программа микроконтроллера – ШД переведены на управление по заданному перемещению (вместо задания скорости). Это позволило синхронизировать работу 4 шаговых двигателей экструдеров подачи нити с шаговым двигателем конвейера. За счёт этих изменений значительно улучшена стабильность диаметра (в среднем 1,6–1,8)

Пятая версия. Смешивающие сопла.

Реализация идеи улучшения смешивания, описанной разделе 6. Изготовил сопла с разными вариантами смешивающих элементов. В результате было значительно улучшено качество смешивания.

Шестая версия. Программа анализа G-code.

Разработана первая версия программы анализа G-code модели и генерации управляющей последовательности. Проведены тесты для проверки её работоспособности. Реализован программный перевод цветовой модели RGB (красный - зеленый - голубой), используемой для отображения на экране монитора в цветовую модель RYBW (Красный -

желтый – синий – белый). Программа стабильно и достаточно точно производит генерацию управляющей последовательности.

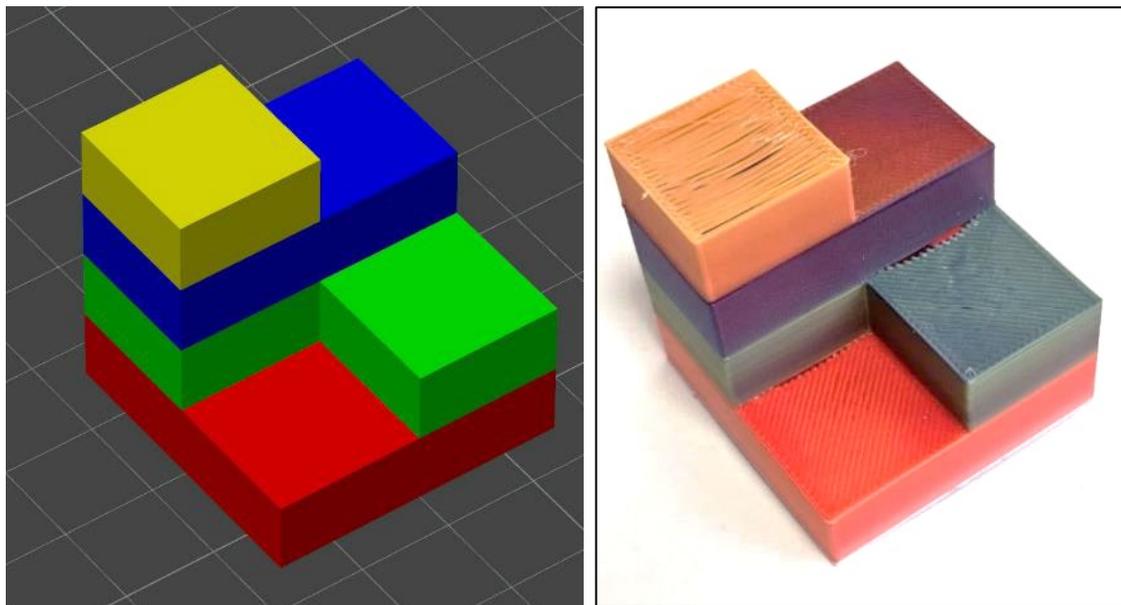


Рисунок 27 – Модель и напечатанный образец для отладки работы программы анализа G-code

Седьмая версия. Модернизация конвейера.

В процесс настройки точности работы ССФ, было выявлено проскальзывание нити по конвейеру, что приводило к накапливающейся ошибке при печати многоцветных моделей. Была произведена модернизация конвейера для устранения проскальзывания. На ленту конвейера добавлен дополнительный слой из резины. Добавлены прижимные и центрирующие ролики.

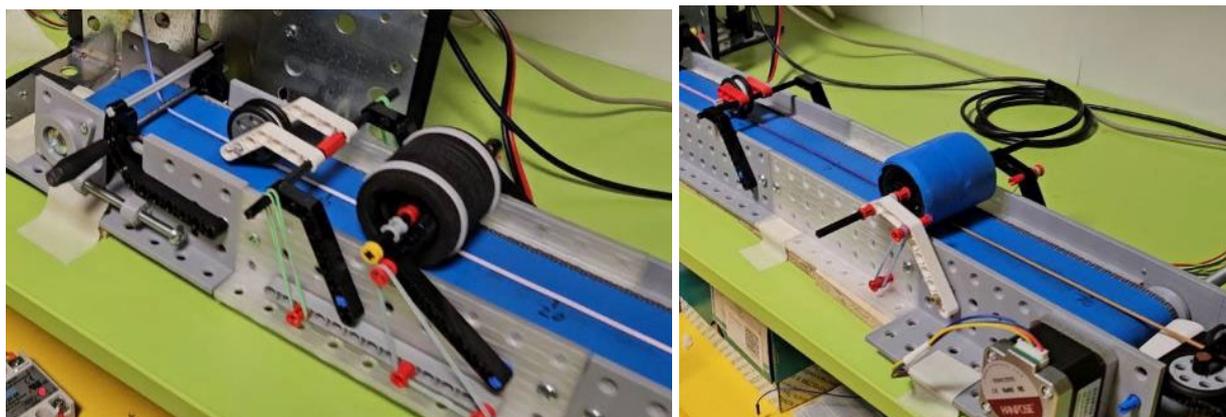


Рисунок 28 – Прижимные и центрирующие ролики конвейера

Дальнейшие планы

- Изготовление сопла с улучшенным смешиванием.
- Доработка программы анализа G-code.
- Доработка элементов управления устройством

- Доработка программы микроконтроллера.
- Доработка конструкции.
- Разработка системы намотки готовой нити на катушку

Результаты работы ССФ

Текущий вариант установки ССФ позволят стабильно автоматически производить нить, составленную из несколько пластиков в последовательности, соответствующей исходной 3D модели. Примеры моделей, распечатанных с применением филамента, изготовленного на ССФ представлены далее.

Модель с резкими и плавными переходами цветов

Для показательной печати и отладки резких и плавных переходов цветов была сформирована простая модель полукруглой разноцветной башни высотой 60 мм. Целенаправленно печать моделей проводилось на основе одного G-code, но с разными параметрами программы генерации управляющей последовательности. Башня сброса лишнего филамента в печати не использовалась

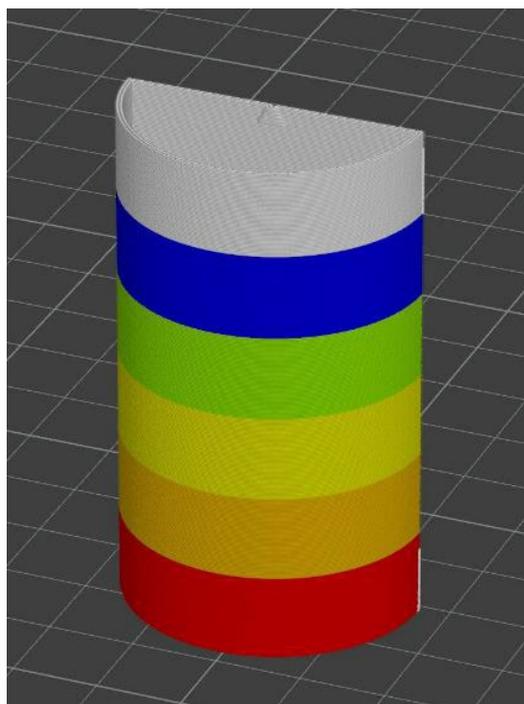


Рисунок 29 – Исходная модель для печати



Резкие переходы.



Плавные переходы

Рисунок 30 – Распечатанные модели с разными переходами.

Тест стабильности печати

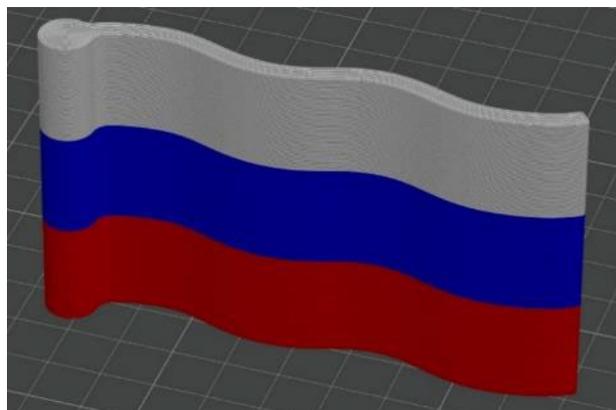


Рисунок 31 – Модель флага в Fusion 360 и Orca Slicer



Рисунок 32 – Распечатанная модель флага

Длина нити, используемая для печати флага, составила 6 метров. На всём протяжении нити диаметр был стабильным в диапазоне 1.6-1.8мм, без дефектов.

ЗАКЛЮЧЕНИЕ

Результаты работы подтверждают работоспособность предложенной идеи. Создана действующая установка, обеспечивающая стабильное производство нити с заданными переходами цветов, соответствующими исходной модели. Разработано программное обеспечение управления микроконтроллером и система анализа G-code модели для генерации необходимой управляющей последовательности установки.

Цель работы достигнута. В ходе работы я получил следующие знания и навыки:

- изучил систему программирования микроконтроллеров Arduino;
- улучшил навыки работы в системах проектирования;
- улучшил понимание принципов управления 3D принтером на основе G-code.

Я планирую продолжить работу над проектом: доработать конструкционную и программную части для улучшения качества получаемой нити и увеличения уровня автоматизации работы установки.

Выражаю благодарность моему папе Бояринцеву Сергею Юрьевичу, за закупку деталей, помощь при проведении экспериментов и моральную поддержку.

Используемые термины

Экструдер (extruder – выдавливатель)— это механизм, контролирующий подачу пластика.

Филамент (от англ. filament – нить) — в данной работе материал, применяемый в 3D печати, пластиковая нить.

Голова принтера, печатающая голова — подвижная часть принтера, состоящая из сопла, нагревателя, радиатора, системы охлаждения, в процессе печати выдавливает пластик для создания модели.

Хотэнд (Hotend) — узел, внутри которого происходит нагрев и выдавливание пластика через сопло.

FDM (Fused Deposition Modeling) / FFF (Fused Filament Fabrication) — Моделирование методом послойного наплавления полимерных (пластиковых) нитей - методы 3D печати

G-code (Геометрический код) — это язык программирования станков с числовым программным управлением.

3D принтер (принтер) — (в данной работе я говорю только о принтерах, печатающих по технологии FDM/FFF)

Статический миксер (смешиватель) — устройство для непрерывного смешивания жидких материалов без использования подвижных элементов [2].

Список использованных источников

1. ГОСТ 24026–80 Планирование эксперимента. Термины и определения
2. Информация о плате расширения RAMPS: https://reprap.org/wiki/RAMPS_1.4
3. Библиотеки и настройки Arduino: <https://alexgyver.ru/>
4. Библиотека управления шаговыми двигателями: <https://alexgyver.ru/gyverstepper/>
5. Библиотека управления кнопками: <https://github.com/GyverLibs/EncButton>
6. Библиотека работы с термисторами: <https://github.com/miguel5612/ThermistorLibrary>
7. Библиотека управления реле: <https://alexgyver.ru/gyverrelay/>
8. Эмулятор для отладки Arduino: <https://wokwi.com/>
9. Алгоритм перевода RGB в RYB: <https://github.com/bahamas10/node-rgb2ryb/blob/master/rgb2ryb.js>
10. Статический миксер: https://en.wikipedia.org/wiki/Static_mixer

Приложение А.

программа управления микроконтроллера

```
#include "GyverPlanner2.h"
#define EB_DEV_TIME 50 // таймаут гашения дребезга кнопки (кнопка)
#define EB_CLICK_TIME 500 // таймаут ожидания кликов (кнопка)
#define EB_HOLD_TIME 600 // таймаут удержания (кнопка)
#include <EncButton.h>
#include <thermistor.h>
#include <GyverRelay.h>

#define maxspeed 1000
#define maxaccel 0

//Определяем соответствия пинов
#define BUTTON_0_PIN 23
#define BUTTON_1_PIN 25
```

```

#define BUTTON_2_PIN 27
#define BUTTON_3_PIN 29

#define X_STEP_PIN 54
#define X_DIR_PIN 55
#define X_ENABLE_PIN 38
#define X_MIN_PIN 3
#define X_MAX_PIN 2

#define Y_STEP_PIN 60
#define Y_DIR_PIN 61
#define Y_ENABLE_PIN 56
#define Y_MIN_PIN 14
#define Y_MAX_PIN 15

#define Z_STEP_PIN 46
#define Z_DIR_PIN 48
#define Z_ENABLE_PIN 62
#define Z_MIN_PIN 18
#define Z_MAX_PIN 19

#define E_STEP_PIN 26
#define E_DIR_PIN 28
#define E_ENABLE_PIN 24

#define Q_STEP_PIN 36
#define Q_DIR_PIN 34
#define Q_ENABLE_PIN 30

#define SDPOWER -1
#define SDSS 53
#define LED_PIN 13

#define FAN_PIN 9

#define PS_ON_PIN 12
#define KILL_PIN -1

#define HEATER_0_PIN 9
#define HEATER_1_PIN 8
#define TEMP_0_PIN 13 // ANALOG NUMBERING
#define TEMP_1_PIN 14 // ANALOG NUMBERING

Stepper<STEPPER2WIRE> stepperX(X_STEP_PIN, X_DIR_PIN);
Stepper<STEPPER2WIRE> stepperY(Y_STEP_PIN, Y_DIR_PIN);
Stepper<STEPPER2WIRE> stepperZ(Z_STEP_PIN, Z_DIR_PIN);
Stepper<STEPPER2WIRE> stepperE(E_STEP_PIN, E_DIR_PIN);
Stepper<STEPPER2WIRE> stepperQ(Q_STEP_PIN, Q_DIR_PIN);
GPlanner2<STEPPER2WIRE, 5> planner;

// установка, гистерезис, направление регулирования
GyverRelay regulator(REVERSE);

#define BUTTONS_COUNT 4
Button btns[BUTTONS_COUNT]; //определяем кнопки

int32_t path[5] = { 0, 0, 0, 0, 0 };
//bool buttons[5];

void setup() {
  // шаг/ 0.1 мм

  Serial.begin(115200);

```

```

// добавляем шаговики на оси
planner.addStepper(0, stepperX); // ось X
planner.addStepper(1, stepperY); // ось Y
planner.addStepper(2, stepperZ); // ось Z
planner.addStepper(3, stepperE); // ось E
planner.addStepper(4, stepperQ); // ось Q

// устанавливаем ускорение и скорость
planner.setAcceleration(maxaccel);
planner.setMaxSpeed(maxspeed);

// начальная точка системы должна совпадать с первой точкой маршрута
planner.setCurrent(path);
planner.start();

// Определяем режим работы пинов

pinMode(FAN_PIN, OUTPUT);
pinMode(HEATER_0_PIN, OUTPUT);
pinMode(HEATER_1_PIN, OUTPUT);
pinMode(LED_PIN, OUTPUT);

pinMode(X_STEP_PIN, OUTPUT);
pinMode(X_DIR_PIN, OUTPUT);
pinMode(X_ENABLE_PIN, OUTPUT);

pinMode(Y_STEP_PIN, OUTPUT);
pinMode(Y_DIR_PIN, OUTPUT);
pinMode(Y_ENABLE_PIN, OUTPUT);

pinMode(Z_STEP_PIN, OUTPUT);
pinMode(Z_DIR_PIN, OUTPUT);
pinMode(Z_ENABLE_PIN, OUTPUT);

pinMode(E_STEP_PIN, OUTPUT);
pinMode(E_DIR_PIN, OUTPUT);
pinMode(E_ENABLE_PIN, OUTPUT);

pinMode(Q_STEP_PIN, OUTPUT);
pinMode(Q_DIR_PIN, OUTPUT);
pinMode(Q_ENABLE_PIN, OUTPUT);

digitalWrite(X_ENABLE_PIN, LOW);
digitalWrite(Y_ENABLE_PIN, LOW);
digitalWrite(Z_ENABLE_PIN, LOW);
digitalWrite(E_ENABLE_PIN, LOW);
digitalWrite(Q_ENABLE_PIN, LOW);

// Определим 4 кнопки
pinMode(BUTTON_0_PIN, INPUT_PULLUP);
pinMode(BUTTON_1_PIN, INPUT_PULLUP);
pinMode(BUTTON_2_PIN, INPUT_PULLUP);
pinMode(BUTTON_3_PIN, INPUT_PULLUP);

ButtonsInit();
RelayInit();

}

#define EVERY_MS(x) \
static uint32_t tmr;\
bool flag = millis() - tmr >= (x);\

```

```

if (flag) tmr += (x);\
if (flag)

//#define DEBUG_ENABLE

float a[][6] = {
    // 0 - расстояние
    // 1 - % X
    // 2 - % Y
    // 3 - % Z
    // 4 - % E
    // 5 - % Q
    { 1, 0, 0, 0, 0, 0 }, // первый элемент не удалять
    { 1, 0, 0, 0, 0, 0 },

// стартовый пакет
    {10, 80, 20, 20, 20, 140},//#FFFFFF R0 Y0 B0 W255 - продавливаем все
филаменты
    {10, 60, 00, 00, 60, 120},// заполняем сопло первым пластиком модели
    {10, 00, 00, 100, 00, 100},// заполняем сопло первым пластиком модели

// сгенерированная программа
{10, 0.0, 0.0, 0.0, 0.0, 0},//#FFFFFF R0 Y0 B0 W255
{10, 0.0, 0.0, 364.86134000000015, 0.0, 364.86134000000015},//#FF0000 R255 Y0 B0
W0
{10, 0.0, 0.0, 59.69822294117654, 217.08444705882374,
276.782670000000034},//#FFC800 R70 Y254 B0 W0
{10, 0.0, 0.0, 276.78210000000001, 276.78210000000001},//#FFF000 R0 Y255 B0 W0
{10, 0.0, 75.12668942857147, 0.0, 201.6558505714287,
276.782540000000015},//#A0FF00 R0 Y255 B95 W0
{10, 0.0, 276.782500000000037, 0.0, 0.0, 276.782500000000037},//#0000FF R0 Y0 B255
W0
{10, 276.082270000000002, 0.0, 0.0, 0.0, 276.082270000000002},//#FFFFFF R0 Y0 B0 W255
{10, 180.0, 0.0, 0.0, 0.0, 180},//#FFFFFF R0 Y0 B0 W255
{10, 0, 0, 0, 0, 0},//#FFFFFF

{10, 200, 0, 0, 0, 200},//end

    { 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0 }, // последний элемент не удалять
};

int asize = sizeof(a) / (4 * 6) - 1; //автоматически считаем количество действий

double coefEX = 1.01;
double cX = (double)100000 / 2460*coefEX;
double cY = (double)100000 / 2460*coefEX;
double cZ = (double)100000 / 2460*coefEX;
double cE = (double)100000 / 2460*coefEX;
double cQ = (double)100000 / (21410-5)*coefEX;

unsigned long time_effective = 0;
unsigned long tPrevious = 0;
bool Button1_prev_state = 0;
int t_show_counter = 0;
unsigned long tcur;
unsigned long ttarget = 0;

int StepCount = 0; // счётчик точек маршрута
//int StepMultiplicator = 100; //Умножение для пути

```

```

int LastStep = 0;
int t;

void print_steppers() {
    int32_t xst, yst, zst, est, qst;
    xst = (int32_t)stepperX.pos;
    yst = (int32_t)stepperY.pos;
    zst = (int32_t)stepperZ.pos;
    est = (int32_t)stepperE.pos;
    qst = (int32_t)stepperQ.pos;

    float sst;
    sst = (float)(xst + yst + zst + est);

    // Serial.print("StepCount=");
    // Serial.print(StepCount);
    Serial.print(", Xstep=");
    Serial.print(xst);
    Serial.print(", Ystep=");
    Serial.print(yst);
    Serial.print(", Zstep=");
    Serial.print(zst);
    Serial.print(", Estep=");
    Serial.print(est);
    Serial.print(", Qstep=");
    Serial.print(qst);
    Serial.print(", Total length (mm)=");
    Serial.println(sst);
}

void print_path() {
    Serial.print("====StepCount=");
    Serial.print(StepCount);
    Serial.print(", Xpath=");
    Serial.print(path[0]);
    Serial.print(", Ypath=");
    Serial.print(path[1]);
    Serial.print(", Zpath=");
    Serial.print(path[2]);
    Serial.print(", Epath=");
    Serial.print(path[3]);
    Serial.print(", Qpath=");
    Serial.println(path[4]);
}

void print_Current_axis() {
    Serial.print("Xcurrent=");
    Serial.print(planner.getCurrent(0));
    Serial.print(", Ycurrent=");
    Serial.print(planner.getCurrent(1));
    Serial.print(", Zcurrent=");
    Serial.print(planner.getCurrent(2));
    Serial.print(", Ecurrent=");
    Serial.print(planner.getCurrent(3));
    Serial.print(", Qcurrent=");
    Serial.println(planner.getCurrent(4));
}

void loop() {
    // здесь происходит движение моторов
    planner.tick();
    ButtonsTick();
    // если в буфере планировщика есть место

```

```

if (planner.available()) {

  if (StepCount <= asize) {
    // рассчитываем новые точки
    StepCount++;
    t = a[StepCount][0];

    Serial.println(cX);

    path[0] = cX * a[StepCount][1] * t;
    path[1] = cY * a[StepCount][2] * t;
    path[2] = cZ * a[StepCount][3] * t;
    path[3] = cE * a[StepCount][4] * t;
    path[4] = cQ * a[StepCount][5] * t;

    if (StepCount == asize) LastStep = 1;
    print_path();
    planner.addTarget(path, LastStep, RELATIVE);

  } else { //Завершение программы
    //Serial.println("Stop program");
  }
}

{
  EVERY_MS(5000) { // выводим 1 раз в N циклах
    print_steppers();
  }
}

// при удержании кнопки [3] перематываем филамент обратно
if (btns[3].hold()) {
  path[0] = 0;
  path[1] = 0;
  path[2] = 0;
  path[3] = 0;
  path[4] = 0;
  Serial.println("=====перемотка назад");

  planner.brake();
  print_steppers();
}

if (btns[0].release()) {
  // включить моторы

  Serial.print("Resume ");
  planner.resume();
}

if (btns[0].press()) {
  // выключить моторы
  Serial.print("Pause ");
  planner.brake();
}

double temp; // например читаем с датчика температуру

{ EVERY_MS(200){
temp = getHotEndTemperature();
regulator.input = temp; // сообщаем регулятору текущую температуру
}
}

```

```

digitalWrite(HEATER_0_PIN, regulator.getResultTimer()); // отправляем на реле
}
}

{
  EVERY_MS(5000) {
    Serial.print(temp);
  }
}

#define BUTTON_0_PIN 23
#define BUTTON_1_PIN 25
#define BUTTON_2_PIN 27
#define BUTTON_3_PIN 29

void ButtonsInit() {
  btns[0].init(BUTTON_0_PIN, INPUT_PULLUP, HIGH);
  btns[1].init(BUTTON_1_PIN, INPUT_PULLUP, HIGH);
  btns[2].init(BUTTON_2_PIN, INPUT_PULLUP, LOW);
  btns[3].init(BUTTON_3_PIN, INPUT_PULLUP, LOW);
}

int ButtonsTick() {
  int Flag = 0;
  Flag = Flag + btns[0].tick() * 1;
  Flag = Flag + btns[1].tick() * 2;
  Flag = Flag + btns[2].tick() * 4;
  Flag = Flag + btns[3].tick() * 8;
  return Flag;
}

void ButtonsPrintState() { // Распечатать статусы кнопок
  for (int i = 0; i < BUTTONS_COUNT; i++) {
    if (btns[i].click()) {
      Serial.print(i);
      Serial.println(" click");
    }
    if (btns[i].hold()) {
      Serial.print(i);
      Serial.println(" hold");
    }
    if (btns[i].press()) {
      Serial.print(i);
      Serial.println(" press");
    }
    if (btns[i].release()) {
      Serial.print(i);
      Serial.println(" release");
    }
  }
}

thermistor therm1(TEMP_0_PIN, 1);

double getHotEndTemperature() {
  double temp = therm1.analog2temp(); // read temperature
  return temp;
}

void RelayInit() {

// pinMode(HEATER_0_PIN, OUTPUT);

```

```

Serial.begin(115200);
pinMode(HEATER_0_PIN, OUTPUT);           // пин реле
regulator.setpoint = 200;                 //199 установка (ставим на 40 градусов)
regulator.hysteresis = 1;                 // ширина гистерезиса
regulator.k = 0.9;                        // коэффициент обратной связи
regulator.dT = 200;                       // установить время итерации для getResultTimer
}

```

Приложение Б

Программа анализа G-code и генерации управляющей последовательности ССФ

```

def RGB2RYB(r,g,b):
    k=255-max(r,g,b)
    w=min(r, g, b)
    r -= w
    g -= w
    b -= w
    mg=max(r, g, b)
    k=255-max(r,g,b)

    y = min(r, g)
    r -= y
    g -= y
    if (b and g):
        b /= 2
        g /= 2
    y += g
    b += g
    my = max(r, y, b)
    if (my):
        n = mg / my
        r *= n
        y *= n
        b *= n
    # r += w
    # y += w
    # b += w
    return r,y,b,k,w

def colorsolve(color,extrude):

    global exmultip,Rcoef,Ycoef,Bcoef,Kcoef,Wcoef,conveyorcoef

    r=int(color[1:3],16)
    g=int(color[3:5],16)
    b=int(color[5:7],16)

    r,y,b,k,w=RGB2RYB(r,g,b)
    # k - заготовка под добавление чёрного цвета

    w*=Wcoef
    r*=Rcoef
    y*=Ycoef
    b*=Bcoef

    colorsum=r+y+b+w # не используем k

    colorcode[0]=exmultip
    colorcode[1]=w*extrude/colorsum
    colorcode[2]=b*extrude/colorsum
    colorcode[3]=r*extrude/colorsum

```

```

colorcode[4]=y*extrude/colormap
#colorcode[5]=k*extrude/colormap
colorcode[5]=conveyorcoef*extrude

ssffline='{'+str(colorcode)[1:-1]+'}',+'//'+color+' R'+str(int(r))+
Y'+str(int(y))+ ' B'+str(int(b))+ ' W'+str(int(w))
    return(ssffline)

#-----настройки-----

file = open('test3.gcode','r') # название входного файла
ssf = open('ssffile','w') # название выходного файла

exmultip=10 # множитель длины 10 означает 1 единица программы равна 1мм

# коэффициенты домножения цветов
Rcoef=1
Ycoef=1
Bcoef=1
Kcoef=1
Wcoef=1
conveyorcoef=1

end_extrusion=180 # конечная нить для компенсации длины от экструдера до сопла
принтера

#-----настройки-----

extrude=0
alllength=0

color='#FFFFFF'
colorcode=[exmultip,0,0,0,0,0]

while True:
    line = file.readline()
    if not line:
        break

    if line[0]=='G': # нахождение строки с информацией о движениях принтера
        E=line.find('E') # поиск E - обозначения длины выдавленной нити
        if E!=-1:
            line=line[:-1] # удаление знаков перехода строки
            x=E+1
            Econvert=''
            while line[x]!=' ':
                Econvert+=line[x]
                x+=1
            if x==len(line):
                break
            exlen=float(Econvert)
            extrude+=exlen

    elif line==';changing_color\n': # строка-маркер смены цвета

        ssffline=coloursolve(color,extrude)
        ssf.write(ssffline+'\n')
        print(ssffline)

        color=file.readline()[:-1]
        alllength+=extrude

```

```
extrude=0

ssfline=colorsolve(color,extrude)
ssf.write(ssfline+'\n')
print(ssfline)
alllength+=extrude

# добавление конечной длины
print(alllength)

extrude=end_extrusion
ssfline=colorsolve(color,extrude)
ssf.write(ssfline+'\n')

colorcode=[exmultip,0,0,0,0,0]
ssfline='{'+str(colorcode)[1:-1]+'}', '+'//'+color
ssf.write(ssfline+'\n')

ssf.close()
```

**МОДУЛЬНЫЙ ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ПРОВЕРКИ
ЗНАНИЙ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ «СКАНТЕСТ»**

Бухтиярова Дарья Антоновна

Кемерово г, МБОУ Лицей №62, 9 класс

Секция: Цифровая инженерия

Научный руководитель: Авдеев Ярослав Всеволодович, преподаватель

Центра талантов «Кемерово» («УникУм»)

Аннотация

Область техники безопасности постоянно развивается, внося новые требования и стандарты. Модульный программно-аппаратный комплекс может быть настроен и обновлен для соответствия последним требованиям и стандартам безопасности, что дает возможность обучать и проверять знания по самым актуальным темам. Целью проекта является разработка программно-аппаратного комплекса для диагностики физиологического состояния и уровня знаний по технике безопасности. Для диагностики физиологического состояния и уровня знаний по технике безопасности был разработан модульный программно-аппаратный комплекс. На основе созданной базы данных работников реализована функция распознавания личности с помощью машинного зрения. Немаловажным фактором является оценка физиологического состояния в соответствии с требованиями к сотрудникам в зависимости от тяжести работ. На основе общих составленных вопросов по технике безопасности было реализовано тестирование для проверки знаний по охране труда. Для визуализации всех необходимых данных реализовано GUI приложение.

Введение

В настоящее время сфера безопасности является одной из наиболее важных и быстроразвивающихся областей. Компании и организации стремятся обеспечить безопасность своих работников и имущества, для этого им необходимы специалисты, оснащенные техническими знаниями в области безопасности. В связи с этим увеличивается спрос на обучение и проверку знаний в области охраны труда.

Область техники безопасности постоянно развивается, внося новые требования и стандарты. Модульный программно-аппаратный комплекс может быть настроен и обновлен для соответствия последним требованиям и стандартам безопасности, что дает возможность обучать и проверять знания по самым актуальным темам. Также, немаловажным фактором является оценка физиологического состояния в соответствии с требованиями к работникам в зависимости от тяжести работ. Таким образом необходимо обеспечить комплексную проверку работника, как на знание техники безопасности, так и на пригодность для выполнения работ.

Проблема: Международная организация труда «Достойный труд – безопасный труд» отмечает, что ежегодно в результате несчастных случаев и профессиональных заболеваний, связанных с трудовой деятельностью, погибает около 2,2 миллиона человек.

По данным Росстата, в России при несчастных случаях на производстве пострадали 20,3 тысячи человек, из них 1,07 тысячи погибли за 2022 год.

Наиболее частой причиной происшествий при работе со специализированным оборудованием является человеческий фактор. Несоблюдение режима работы, а также регламентов охраны труда приводит к многочисленным травмам и приостановке производственного процесса.

Актуальность: На данный момент существует огромная база законодательных актов по защите труда, независимо от организационно-правовой формы предприятия. Для контроля за соблюдением существуют отделы, которые должны регламентировать и проверять сотрудников на знание стандартов безопасности; особенностей эксплуатации оборудования, правил безопасного соответствия технического состояния. Нормативно-правовые акты, напрямую связанные с предметной областью проекта:

1. Согласно ТК РФ Статья 76. Отстранение от работы, работодатель обязан отстранить от работы (не допускать к работе) работника:

- появившегося на работе в состоянии алкогольного, наркотического или иного токсического опьянения;
- не прошедшего в установленном порядке обучение и проверку знаний и навыков в области охраны труда;

- не прошедшего в установленном порядке обязательный медицинский осмотр, а также обязательное психиатрическое освидетельствование в случаях, предусмотренных настоящим Кодексом, другими федеральными законами и иными нормативными правовыми актами Российской Федерации; (в ред. Федеральных законов от 30.11.2011 N 353-ФЗ, от 25.11.2013 N 317-ФЗ) [1].

2. Работодатель обязан обеспечить обучение и проверку знаний требований охраны труда работников (ст. 214 ТК РФ). Порядок выполнения этой обязанности установлен в Правилах № 2464 (Постановление Правительства РФ от 24.12.2021 № 2464) [2].

3. Согласно ТК РФ Статья 225, финансирование мероприятий по улучшению условий и охраны труда работодателями (за исключением государственных унитарных предприятий и федеральных учреждений) осуществляется в размере не менее 0,2 процента суммы затрат на производство продукции (работ, услуг) [3].

Все больше компаний выделяют для себя такое направление, как цифровизация систем охраны труда. Так, к примеру, компания ОАО «РЖД» опубликовала техническое задание в перечне запросов на инновации на разработку системы оценивания сотрудников в области охраны труда с применением современных цифровых технологий для снижения фальсификации результатов опроса и постоянного мониторинга знаний. [4].

Гипотеза: Снижение уровня травматизма на производственных предприятиях путем комплексной оценки физиологического состояния проверяемого и уровня его знаний по технике безопасности.

Цель проекта: Разработка программно-аппаратного комплекса для диагностики физиологического состояния и уровня знаний по технике безопасности.

Задачи проекта:

1. Провести анализ предметной области
2. Разработать концепт устройства
3. Провести апробацию в реальных производственных условиях

Метод исследования: сбор и анализ информации с официальных источников для обоснования значимости проекта. Для анализа аналогов необходимо использовать официальные сайты конкурентов. Для разработки используются сайты с документацией кода и описанием компонентов.

Глава 1. Анализ предметной области

Центральное место в экономике занимает сфера материального производства – высокотехнологичная промышленность, которая должна отвечать требованиям глобальной

конкурентоспособности, требованиям эффективности и высокой производительности труда. Направление «Технет» посвящено развитию и применению одного из самых важных классов «сквозных технологий» – передовых производственных технологий. По итогам 2017–2019 годов в рамках конкурса «Развитие-НТИ» (II–V очереди) по лоту «Технет» было профинансировано 64 проекта из 410 заявок на общую сумму 1,13 млрд рублей [5].

Объем российского рынка цифрового проектирования и моделирования в 2021 году составил около 11,6 млрд руб. В ближайшие 5 лет ожидаемые темпы роста рынка могут составить около 6%, что может обеспечить увеличение объема рынка цифрового проектирования и моделирования до отметки 14,8 млрд руб. к 2025 году.

Целевая аудитория проекта – объекты с регламентированным режимом доступа на территорию, промышленные организации топливно-энергетического и минерально-сырьевого комплекса и логистические компании.

Ключевой основой проекта является стремление организаций к соответствию требованиям нормативных документов по обеспечению безопасности:

1. Трудовой кодекс Российской Федерации (от 30 декабря 2001 г. N 197-ФЗ)
2. "ГОСТ 12.0.004-2015. Межгосударственный стандарт. Система стандартов безопасности труда. Организация обучения безопасности труда. Общие положения" (вместе с "Программами обучения безопасности труда") (введен в действие Приказом Росстандарта от 09.06.2016 N 600-ст).
3. ГОСТ 12.0.004-2016. Организация обучения безопасности труда.
4. ГОСТ 12.0.230-2007. Системы управления охраной труда.

Глава 2. Анализ конкурентов

Проведя анализ, были выявлены основные конкуренты проекта. Главное отличие продукта от конкурентов — это подбор вопросов, основанный на анализе ошибок за все время работы сотрудника, записанных в базе данных. Также можно выделить использование машинного зрения для считывания биометрии лица во время всей проверки, что помогает исключить фальсификацию результатов тестирования. Устройство имеет индивидуальную функцию – объединение двух процессов: оценка физиологического состояния и проверка уровня знаний по технике безопасности. Система имеет возможность быстро интегрировать программное обеспечение или внутренние составляющие под разные отделы и профессии со своей спецификой.

1. Abie System – система цифрового управления процессами охраны труда и промышленной безопасностью. Система разработана на основе облачного хранения данных, что позволяет осуществлять контроль над процессами охраны труда с любого устройства, имеющего доступ в интернет. На данный момент Abie System успешно внедрена на крупных промышленных предприятиях. Разработанный сервис в режиме реального времени собирает

данные, обрабатывает их с помощью алгоритмов машинного обучения и принимает решения с использованием предиктивной аналитики [6].



Рисунок 1 - Abie System

2. ЮМС Охрана труда – видеоинформационный сенсорный терминал, предназначенный для проведения обучения, инструктажа и контроля качества знаний работников предприятия в соответствии с требованиями охраны труда и техники безопасности. Комплекс предоставляет возможность аналитической работы с данными из электронной карты работника, позволяет формировать отчеты как по всем работникам, так и по каждому работнику в отдельности. Терминал проводит ежедневные мини-тестирования работников в режиме обучения, экзамены и инструктажи.



Рисунок 2 – ЮМС Охрана труда

3. Safety Training Kiosk – автономная интерактивная система, предназначенная для проведения программ обучения технике безопасности и управления ими в различных условиях, таких как рабочие места или общественные места. Киоск для обучения технике безопасности состоит из интерфейса с сенсорным экраном, который помогает пользователям пройти ряд интерактивных модулей. Киоски для обучения технике безопасности имеют встроенные функции отслеживания и отчетности прогресса обучения работников.



Рисунок 3 – Safety Training Kiosk

Таблица 1 – Сравнительный анализ конкурентов и системы «СканТест»

Название	Тестирование знаний ТБ	Подбор вопросов на основе истории нарушений ТБ	Использование биометрии для идентификации	Проверка на наличие работника в базе данных	Проверка физиологического состояния
СканТест	+	+	+	+	+
Abie System	+	—	—	+	—
ЮМС Охрана труда	+	—	—	+	+
Safety Training Kiosk	+	—	+	+	—

Глава 3. Разработка концепции и инженерной части

В результате анализа информации о существующих решениях была разработана концепция работы системы, представляющая собой устройство для проведения комплексной оценки физиологического состояния работника и уровня его знаний по технике безопасности:

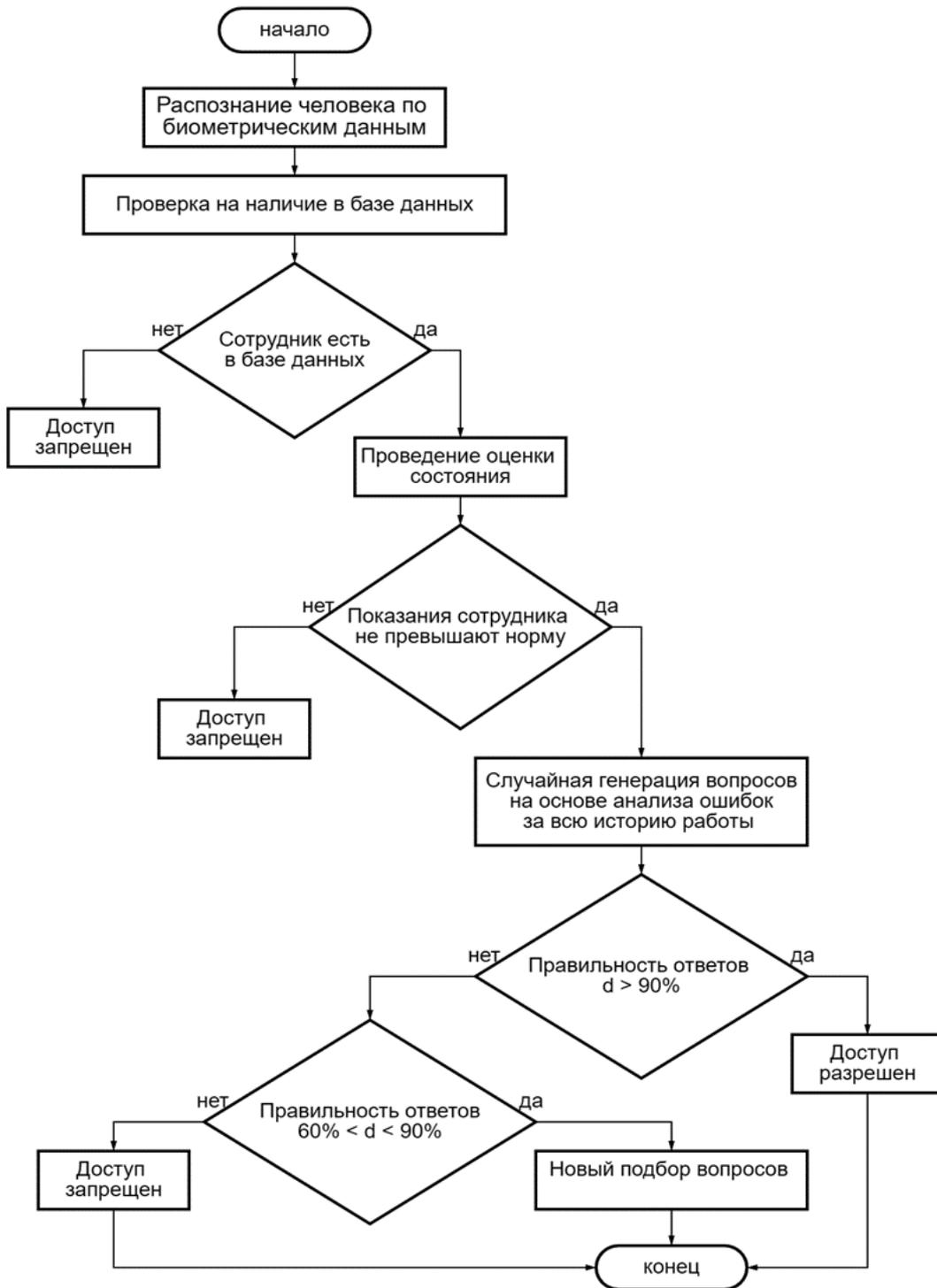


Рисунок 4 – Алгоритм работы системы

Разрабатываемое устройство разделено на два отсека:

1. Центральный отсек включает в себя одноплатный микрокомпьютер и микроконтроллер.
2. Сенсорный отсек состоит из модулей для проведения проверки физиологического состояния работника (датчик пульса и уровня SpO2, датчик паров спирта) и веб-камеры для

реализации автоматизированной системы распознавания работника с использованием машинного зрения в целях предотвращения фальсификации результатов тестирования.

Устройство проводит распознавание работника по биометрическим данным, проверяя сотрудника на наличие в базе данных. При попадании постороннего человека на территорию предприятия ограничивается доступ ко входу. После идентификации личности система генерирует случайные вопросы, основанные на прошлых ошибках работника, после каждого вопроса работник выбирает вариант ответа, который считает правильным. Далее система производит оценку физиологического состояния сотрудника. На основе ответов и оценки состояния работнику выдается доступ к работам.

В случае прохождения тестирования менее чем на 40% работник отстраняется от работы на неопределенное срок, обозначенный работодателем (ст. 76 ТК РФ). Если у работника выявлены медицинские противопоказания, работодатель обязан отстранить его от работы (не допускать к работе, исполнению трудовых обязанностей) (ст. 214 ТК РФ). Затраченное время на одного работника для прохождения тестирования и оценки физиологического состояния составляет около 1 минуты. Корпус устройства напечатан на 3D принтере с использованием филамента PETG (Приложение 2).

Разрабатываемое устройство должно соответствовать всем пунктам нормативно-правовой базы в области обеспечения внутриобъектового и пропускного режимов, обеспечивая безопасное использование и эксплуатацию оборудования в производственных помещениях, попадающих в категорию объектов с повышенным уровнем опасности, а также на предприятиях с регламентированным режимом допуска на территорию.

Выбор ресурсного обеспечения:

1. Микрокомпьютер Raspberry Pi 4 - одноплатный компьютер, построенный на ARM-архитектуре, выполненный на чипе SoC Broadcom BCM2711.



Рисунок 5 – Элементы Raspberry Pi 4



Рисунок 6 – Распиновка платы

2. **Arduino Nano** – полнофункциональная отладочная плата, адаптированная для работы с макетными платами, построенная на базе микроконтроллера ATmega328 или Atmega168.

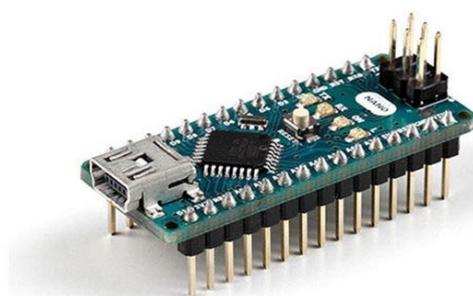


Рисунок 7 – Arduino Nano

Плата имеет 14 цифровых и 8 аналоговых пинов, из которых 6 пинов являются ШИМ. Имеет возможность общения с периферией по интерфейсу SPI.

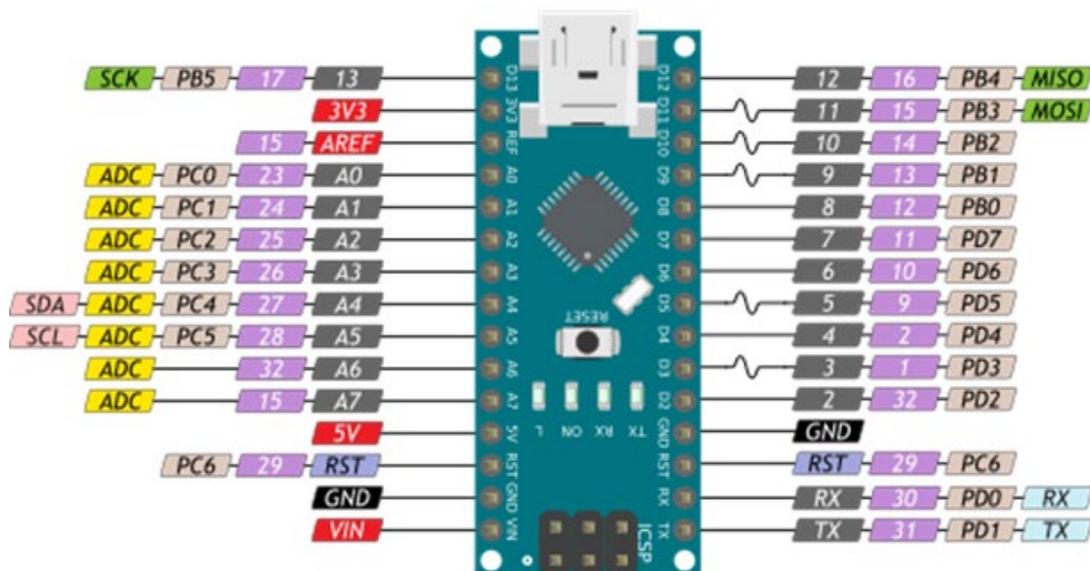


Рисунок 8 – Распиновка

3. Цифровой датчик пульса и уровня SpO₂, MAX30102



Рисунок 9 – Датчик пульса и уровня SpO₂

Принцип работы: Датчик посылает небольшие пучки света, которые проходят через кровь в пальце и частично поглощаются в насыщенном кислороде или дезоксигенированной крови. Насыщенная кровь поглощает больше инфракрасного света и пропускает больше красного света. Излучаемый свет поглощается насыщенной кровью, а остальной свет отражается от пальца и попадает на детектор, данные которого обрабатываются и считываются.

Основная микросхема - датчик MAX30102, способный измерять уровень кислорода в крови и частоту сердечного ритм (HR). Измерения осуществляется с помощью оптического датчика. Данные о частоте сердечных сокращений и содержании кислорода в крови передаются на микроконтроллеры через связь I2C. Модуль оснащен встроенным датчиком температуры. Разрешение температуры составляет 0,0625°C [11].

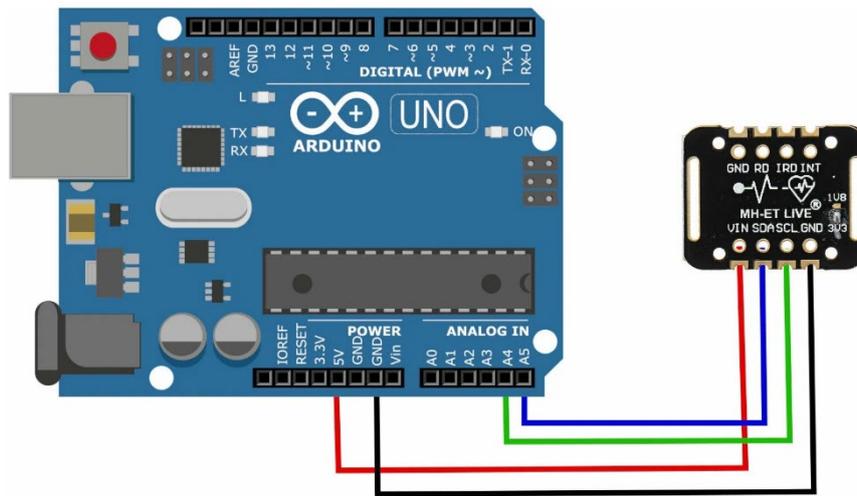


Рисунок 10 – Подключение

5. Датчик паров спирта MQ-3



Рисунок 11 – Датчик MQ-3

Принцип работы: датчик проводит изменение сопротивления тонкопленочного слоя диоксида олова при контакте с молекулами определяемого газа. Чувствительный элемент датчика состоит из керамической трубки с покрытием Al_2O_3 и нанесенного на неё чувствительного слоя диоксида олова. Внутри трубки проходит нагревательный элемент, который нагревает чувствительный слой до температуры, при которой он начинает реагировать на определяемый газ. Чувствительность к разным газам достигается варьированием состава примесей в чувствительном слое.

Модуль MQ-3 защищен от помех, таких как дым и пары углеводородов. Аналоговый датчик имеет 4 выхода: питание VCC и заземление GND, а также цифровой и аналоговый выход. Датчик MQ-3 представляет собой полупроводниковое устройство для измерения наличия паров алкоголя при их концентрации в воздухе от 0,05 мг/л до 10 мг/л [10].

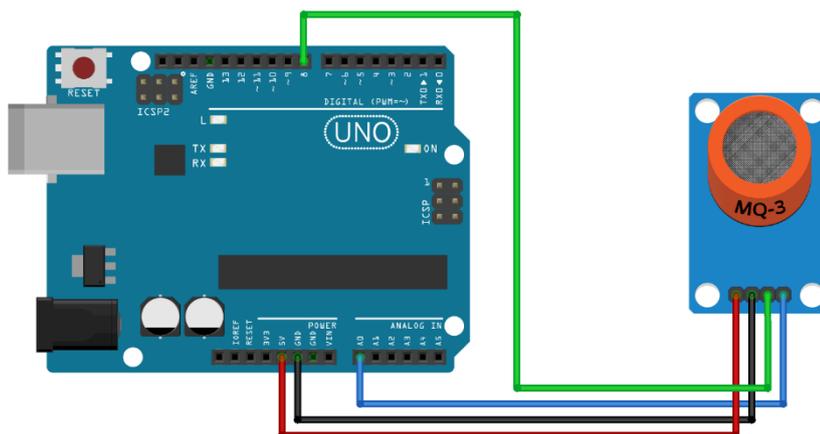


Рисунок 12 – Подключение

6. Веб-камера DEXP Chat M100R1



Рисунок 13 – Веб-камера

Интерфейс подключения камеры – USB. Кабель имеет ферритовый фильтр, обеспечивающий защиту от электромагнитных помех. Камера оснащена 2-мегапиксельной матрицей. Разрешение видео – 1920x1080. Максимальная частота кадров – 30 кадров в секунду. За запись звука отвечает встроенный микрофон. Габариты устройства – 77x60x51 мм [9].

Глава 4. Программная часть устройства

Машинное зрение реализовано с помощью библиотеки OpenCV на языке программирования Python. Применялся метод каскадов Хаара. Основные этапы данного метода:

- Изображение преобразуется в оттенки серого для анализа.
- Классификатор пропускает изображение через цепочку этапов-каскадов.
- На каждом этапе каскада применяется набор признаков Хаара - это прямоугольные области разного размера и соотношения сторон.

- Признаки используются для вычисления промежуточных вероятностей наличия объекта.
- Если изображение проходит все каскады - объект считается найденным. Выдаются координаты прямоугольной области, где находится объект.

После идентификации система проверяет человека на наличие в базе данных. Для создания базы данных была выбрана система управления базами данных SQLite3. SQLite3 отличается компактностью – она занимает один файл, благодаря чему встраивается в приложения, чтобы выполнять функции СУБД автономно, без использования сервера.

Программный код для модулей физиологического состояния был написан на языке программирования C++ в бесплатной среде Arduino IDE. Была разработана блок-схема работы программного кода для выбранных модулей (Приложение 1).

Для визуализации показателей с модулей и реализации тестирования с помощью виртуальных кнопок было разработано GUI приложение (Приложение 5). Для разработки приложения был выбран фреймворк PyQt5. Передача данных осуществляется с помощью последовательного порта для связи между Raspberry Pi и Arduino Nano.

Глава 5. Экономическая часть проекта

Себестоимость готового продукта складывается из затрат на материалы, оборудование и работу:

1. Затраты на материалы:

- Провода-перемычки = 226 рублей
- Микроконтроллер ATmega328 (в составе ОП “Arduino”) = 1090 рублей
- Цифровой датчик пульса и уровня SpO₂, MAX30102 = 510 рублей
- Датчик паров спирта MQ-3 = 410 рублей
- Филамент PETG = 995 рублей
- Веб-камера = 770 рублей
- Микрокомпьютер Raspberry Pi 4 = 17 640 рублей
- Карта памяти = 259 рублей

Итого: 21 900 рублей

2. Затраты на оборудование: 0 рублей

Итого: 0 рублей

3. Затраты на работу:

Итого: 14 820,83 рублей

Всего: 36 720,83 рублей

Цена готового продукта складывается из нескольких аспектов (Приложение 3):

- Себестоимость MVP / 52 = 421,2 рублей
 - Себестоимость готового продукта = 27 795,75 рублей
 - Наценка = 11 016, 25 рублей (30% от себестоимости готового продукта)
 - Выплата производству = 7 344,166 рублей (20% от себестоимости готового продукта)
 - Затраты на стороннего программиста = 4 600 рублей
- Всего: 51 177, 37 рублей**

Результаты проекта и дальнейшее развитие

Для диагностики физиологического состояния и уровня знаний по технике безопасности был разработан модульный программно-аппаратный комплекс. На основе созданной базы данных работников реализована функция распознавания личности с помощью машинного зрения. Немаловажным фактором является оценка физиологического состояния в соответствии с требованиями к сотрудникам в зависимости от тяжести работ. Также разработана концепция действий для внедрения в производство (Приложение 4). Планируется сотрудничество с предприятием по заказу. Реализованные функции:

1. Создана база данных сотрудников;
2. Подключено машинное зрение, написанное на языке программирования Python;
3. Реализовано тестирование для проверки знаний по охране труда;
4. Реализована возможность проверки состояния сотрудника (уровень кислорода в крови, пары спирта, ЧСС);
5. Составлены общие вопросы по технике безопасности
6. Реализовано GUI приложение для визуализации данных

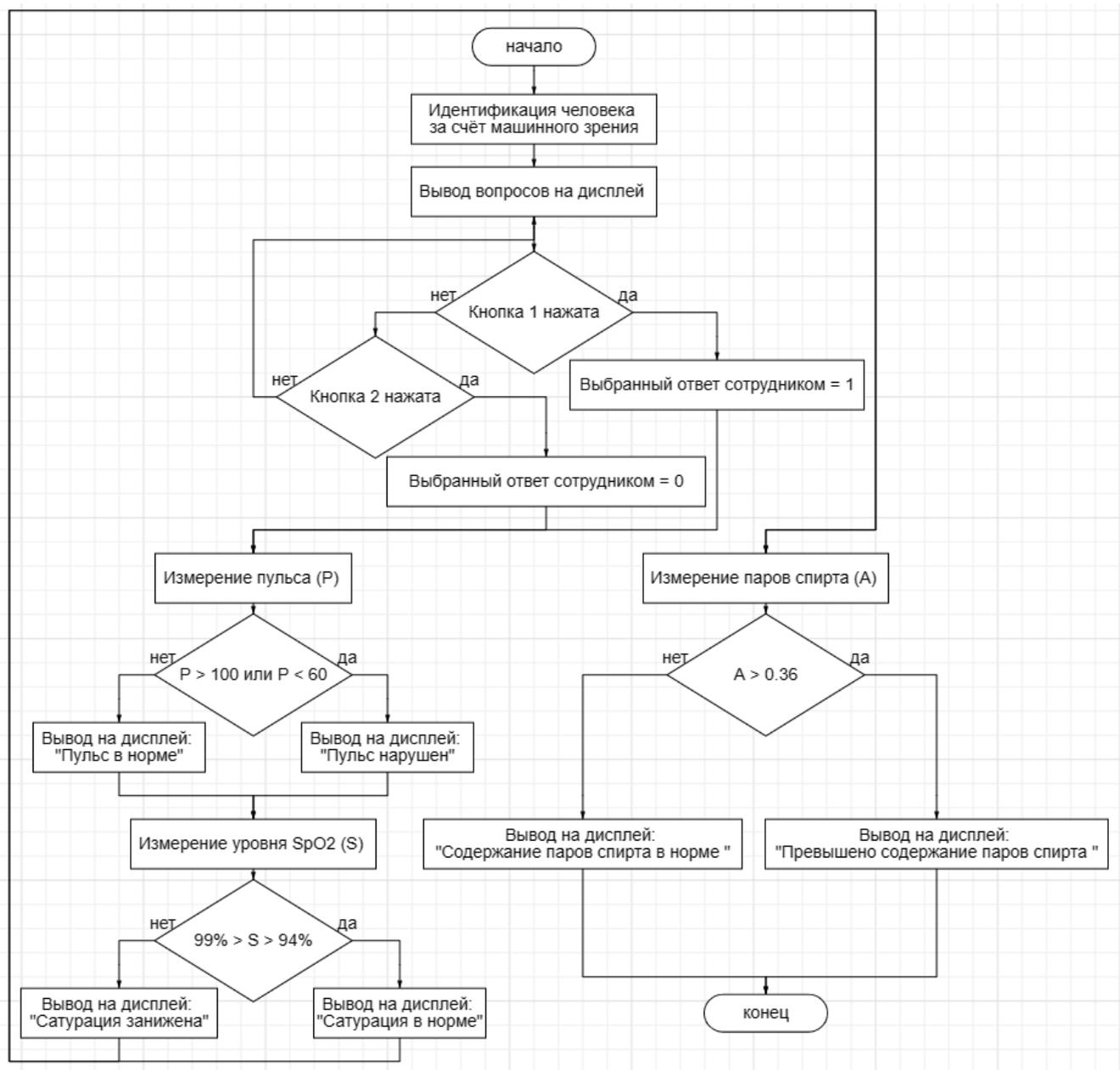
В будущем планируется заключить сотрудничество с организацией, относящейся к целевой аудитории проекта для проведения апробации в реальных производственных условиях и получения обратной связи для качественной доработки проекта. Также к дальнейшему развитию проекта можно отнести разработку второй версии устройства, содержащего улучшенный модуль камеры, более качественный экран, а также поддерживающий проверку большего количества факторов, например, уровень усталости в процессе длительного рабочего процесса.

Заключение

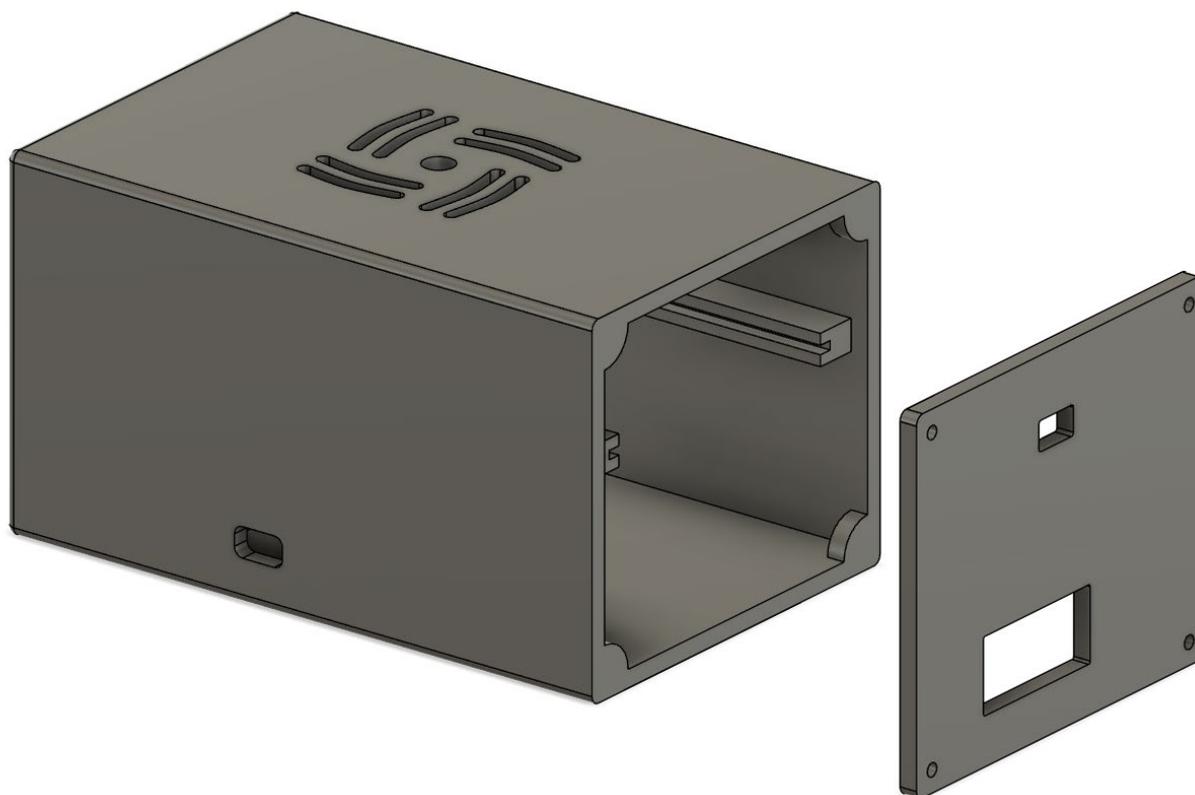
Разработка предварительного прототипа потребовала освоения множества новых знаний и стандартов. Выявлена важность соблюдения рекомендаций нормативных документов в области охраны труда и того, какой трагический опыт лежит в их основе. Получены важные знания по реализации схемотехнических решений в электронных устройствах. В результате данной работы была проведена разработка электронной и

программной части устройства, которую можно применять на объектах с регламентированным режимом доступа на территорию, промышленных организациях топливно-энергетического и минерально-сырьевого комплекса и логистических компаний.

Приложения



Приложение 1 – Блок-схема работы программного кода для аппаратной части



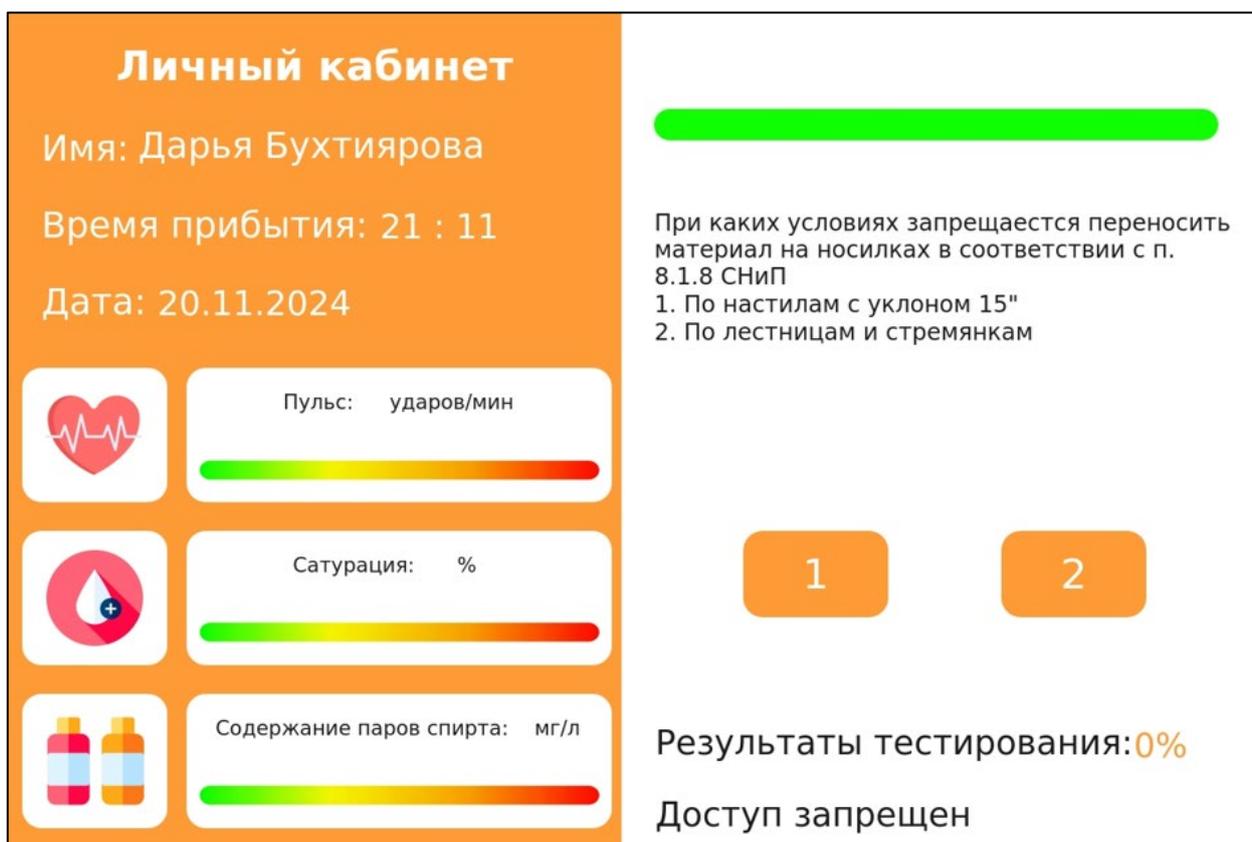
Приложение 2 – 3D-модель устройства



Приложение 3 – Аспекты цены готового продукта для расчета бюджета



Приложение 4 - Концепция действий для внедрения в производство



Приложение 5 – Внешний вид GUI приложения

Список литературы и интернет-источников

1. "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.01.2024).
URL:https://www.consultant.ru/document/cons_doc_LAW_34683/8539e92dc6f7886ba97841e38fc89ebbb7cac099/?ysclid=lsnlmna8qx359331837 (дата обращения 1.01.2024) – Текст: электронный.
2. Постановление Правительства РФ от 24.12.2021 N 2464
URL: <https://normativ.kontur.ru/document?moduleId=1&documentId=442665> (дата обращения 1.01.2024) – Текст: электронный.
3. "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.01.2024).
URL:https://www.consultant.ru/document/cons_doc_LAW_34683/ac72247ee2007f6df12028f034524dc13bc88c2e/ (дата обращения 7.01.2024) – Текст: электронный.
4. ПЕРЕЧЕНЬ ЗАПРОСОВ НА ИННОВАЦИИ ОАО «РЖД» 2023 г.
URL: [1937825 \(yandex.ru\)](https://yandex.ru/1937825) (дата обращения 13.02.2024) – Текст: электронный.
5. «Технет» НТИ
URL: <https://technet-nti.ru/index.php> (дата обращения 15.02.2024) – Текст: электронный.
6. ABIE SYSTEM
URL: <https://klp.ru/index.php?route=information/information&id=385> (дата обращения 15.02.2024) – Текст: электронный.
7. Аппараты ЮМС
URL: <https://www.umssoft.com/jums-ohrana-truda/> (дата обращения 15.10.2024) – Текст: электронный.
8. Safety Training KIOSK - STK™ Scrum System
URL: <https://scrum-system.com/safety-kiosk.html> (дата обращения 15.10.2024) – Текст: электронный.
9. Веб-камера DEXP Chat M100R1
URL: <https://irecommend.ru/content/kamera-kuplennaya-ot-bezyskhodnosti-ee-dostoinstva-i-nedostatki-test-video-i-audio> (дата обращения 15.10.2024) – Текст: электронный.
10. Анализатор утечки газа на Arduino
URL: <https://dzen.ru/a/Xtxr-IQBDByztC> (дата обращения 15.10.2024) – Текст: электронный.
11. Подключение цифрового датчика пульса MAX30102 к Arduino
<https://robotchip.ru/podklyuchenie-tsifrovogo-datchika-pulsa-max30102-k-arduino/?ysclid=lu854vi296678705878> (дата обращения 15.10.2024) – Текст: электронный.

Глоссарий

Автоматизация - применение автоматов или других технических средств, а также экономико-математических методов для частичного или полного освобождения человека от непосредственного участия в производственных (или иных) процессах.

Мониторинг - специальная форма наблюдения (слежения) за текущим изменением тех или иных процессов или объектов в пространстве и во времени, осуществляемая на постоянной основе.

Машинное зрение - область компьютерных наук, которая занимается технологиями и инструментами, позволяющими компьютерам видеть то же, что и люди, и интерпретировать окружающий мир.

Контрольно-пропускной пункт (КПП) – специально выделенное и оборудованное место для пропуска людей и автотранспорта, выноса (вноса) ценностей и документов с территории охраняемого объекта.

ARM-архитектура - семейство архитектур с набором упрощенных команд с режимами простой адресации.

Микроконтроллер - микросхема, предназначенная для управления электронными устройствами.

Широтно-импульсная модуляция (ШИМ) – способ управления подачей мощности к нагрузке. Управление заключается в изменении длительности импульса при постоянной частоте следования импульсов.

Дезоксигенированная кровь – кровь с низким содержанием кислорода в гемоглобине эритроцитов и высоким содержанием двуокиси углерода, растворенного в плазме крови.

Инфракрасный свет — электромагнитное излучение с длиной волны от 0,74 мкм до 1 мм, которое находится за пределами видимого спектра света.

Связь I2C - низкоскоростной последовательный протокол связи, подходящий для передачи данных на короткие расстояния.

Интерфейс подключения - часть аппаратного устройства, которая позволяет операционным системам и программам передавать данные через компьютерную сеть.

ПАРСИНГ И КОДИФИКАЦИЯ КЛЮЧЕВОЙ СТАВКИ

Гончаренко Антон Андреевич

Одинцово г, МБОУ ОЦ "ФЛАГМАН", 11 класс

Секция: Системы обработки информации

*Научный руководитель: Лашина Татьяна Сергеевна, методист, учитель информатики
МБОУ ОЦ «ФЛАГМАН»*

Аннотация

Целью проекта «Парсинг и кодификация ключевой ставки» является продемонстрировать возможности сбора и сортировки информации с помощью комплексной программы. Тематика проекта очень востребована в сфере сбора и анализа данных и использовалась в работе веб-сервиса АНО-заказчика.

В процессе выполнения работы использовалась группа языков программирования: Python, PHP, JavaScript. С помощью подключаемых библиотек, преобразования данных в сжатый вид и подключения удаленной базы данных были достигнуты быстрая отдача и эффективность программы.

Изучение способов написания программ проходило при помощи интернет-запросов и постепенной интеграции элементов полученной информации в код. Предварительное отсеивание ненужных решений задач позволило добиться чистоты и простоты кода.

Результатом проведенной работы стала веб-страница с инвестиционным калькулятором, а также несколько API-ссылок для свободной работы с данными не только пользователю, но и разработчику.

Выводы: Комплекс программ, написанных в рамках сферы проекта, работает исправно. На основе созданного продукта могут быть созданы аналогичные программы, работающие с совершенно другим набором данных

ВВЕДЕНИЕ

В современном мире многие коммерческие и некоммерческие компании нуждаются в простом способе получения информации с Веб-страниц ЦБ РФ.

В целях оптимизации работы инвестиционного калькулятора АНО ДО «РОБОЛАТОРИЯ» было предложено доработать существующую программу с учетом обновляющейся ключевой ставки. Актуальная информация, получаемая парсером ежедневно, исключит проблему ручного обновления данных в коде.

Актуальность. В угоду безопасности ЦБ РФ использует устаревший протокол обмена SOAP, из-за чего своевременное получение информации стало затруднительным.

Проблема. Почти полное отсутствие детальной информации о BS4 и MySQL в пределах одной программы на русском языке, имеющаяся информация не предусматривает возникновения ошибок, а также не систематизирована в один общий курс.

Цель. Создание приложения с наглядной демонстрацией собранных данных.

Задачи.

1. Расширить своё познание о методах добычи информации с помощью программного кода.
2. Показать принцип работы парсеров и БД.
3. Улучшить навык программирования и работы с библиотеками Python.
4. Создать собственный парсер на Python & BS4.
5. Создать собственную страницу с данными о ключевой ставке.

Методы. Сбор информации, отладка кода, анализ имеющейся информации, систематизация информации

1. Теоретическая часть

1.1 Систематизация данных, трудность стоящей задачи

К октябрю 2023 года была четко определена задача — создать пусть и тривиальный, но удобный в пользовании парсер на Python, а затем занести данные с парсера в базу данных (далее – БД). На просторах интернета существуют инструкции по созданию простого сборщика данных, но как оказалось, в задаче оказалось множество подводных камней. Большая часть работы зависела от наводок заказчика, так как основные коды находились у него, парсер был лишь вспомогательным инструментом — это сильно замедляло работу.

На рисунке 1 представлен упрощённый принцип работы парсера. В данном случае, разгадывать капчу не понадобится, как и создавать дополнительный файл, ведь данные сразу будут загружаться в БД. Этот парсер представляет собой библиотеку, а не отдельную программу, поэтому модулем экспорта послужит сам Python.

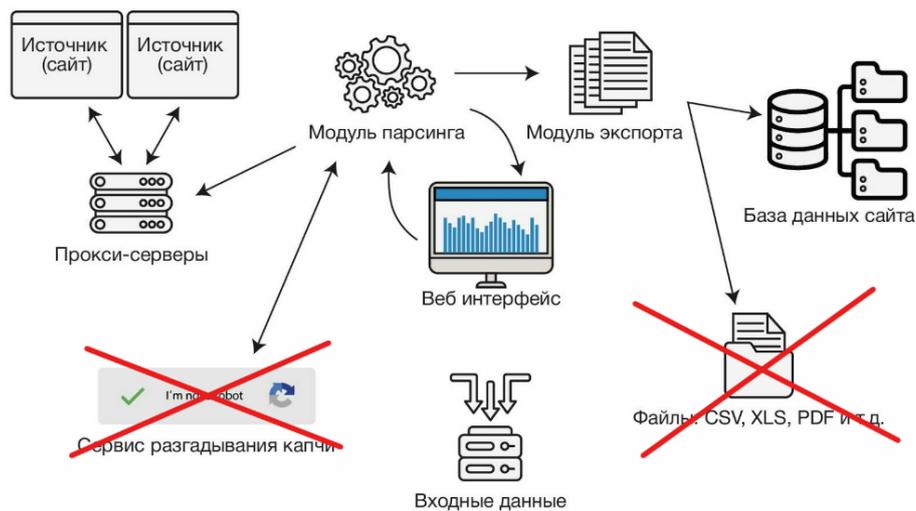


Рисунок 1 - Принцип работы парсера

Последующая работа в php (по просьбе заказчика), чтобы взаимодействовать со структурированными данными, велась с нулевым опытом в языке. Эта сложность ещё сильнее замедлила разработку.

Перед непосредственным началом работы было проведено анализ разработанности проблемы. Единственный найденный аналог (рисунок 2) был написан на языке «Go» и требовал загрузки большого количества пакетов.

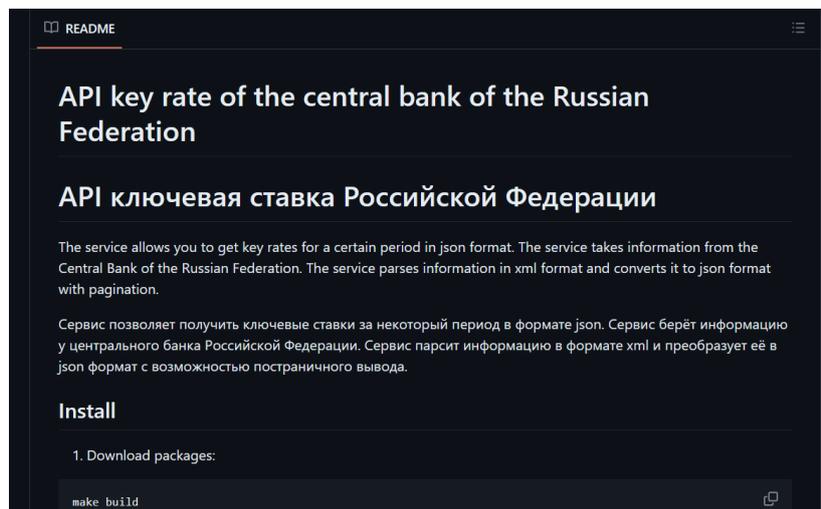


Рисунок 2 – описание программы-аналога

Программа использует иной подход к проблеме – предполагаемый пользователь должен загрузить весь код программы, часть которого приведена в рисунке 3, а затем получать ставку непосредственно в консоли. Так же информация парсится в формате xml, а не кодом HTML. Другие найденные аналоги не предоставляли открытого API и являлись копиями

страницы центробанка. Таким образом, создаваемый калькулятор ставки будет быстрее в загрузке, а также позволит другим программам подключаться к API.

```

make install-swag

make swag

4. Run:

make run

API

GET | v1/keyrate | returns the key rate for today | возвращает ключевую ставку на сегодня

Response:

{
  "current_page": 1,
  "data": [
    {
      "date": "2022-02-16T00:00:00+03:00",
      "rate": "8.50"
    }
  ],
  "from_date": "2022-02-16",
  "per_page": 3,
  "total": 0,
  "total_pages": 0
}

```

Рисунок 3 – принцип работы программы-аналога

1.2 Beautiful Soup 4 и вспомогательные библиотеки

Beautiful Soup 4 — Простая в установке библиотека для Python, предназначенная для работы с файлами HTML и XML. Ввиду незначительного количества требуемых данных для парсинга, именно она была выбрана вместо Scrapy, аналога BS4 с более сложным процессом установки, но большей эффективностью. BS4 работает в тандеме со второй библиотекой.

«Requests» — вторая Python-библиотека, дающая возможность отдавать http-запросы. Она отправляет GET-запрос к ссылке, после BS4 «пользуется» этим запросом и переносит всю страницу к нам в код. Со стороны производительности решение не лучшее, с другой стороны, это происходит лишь раз — дальше только найти нужную информацию, подключаться заново не нужно.



BeautifulSoup

```

# Making an HTTP Request
import requests

url = 'https://crawler-test.com/'
response = requests.get(url)

print('Status code: ', response.status_code)
print('Text: ', response.text[:50])

```

```

from bs4 import BeautifulSoup

# Parse the HTML
soup = BeautifulSoup(response.text, 'html.parser')

# Extract any HTML tag
soup.find('title')

```

Рисунок 4 - Части кода Requests и BeautifulSoup под соответствующими логотипами

Пример кода, который парсит заголовок сайта с доменом «crawler-test.com/», а также выводит данные, чтобы убедиться в правильности работы программы, изображён на рисунке 4.

Помимо этого, используются библиотеки «Pymysql» и «Datetime», но они будут разобраны позже.

1.3 Протокол SOAP у сайтов ЦБ РФ и связанные с ним проблемы

SOAP — протокол обмена данными между компьютерными системами. На нём и работают все страницы с теми или иными данными центробанка. В рисунке 5 представлена часть страницы, вышущенной ЦБ РФ:

```

BankPapers
Информация о ценных бумагах, принимаемых в обеспечение по кредитам Банка России (mod1) (как DataSet)

Тест
Форма для тестирования доступна только для запросов от локальных компьютеров.

SOAP 1.1
В следующем примере показаны запрос и ответ SOAP 1.1. Вместо элементов-заполнителей следует подставить фактические значения.

POST /secinfo/secinfo.asmx HTTP/1.1
Host: www.cbr.ru
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://web.cbr.ru/BankPapers"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/">
  <soap:Body>
    <BankPapers xmlns="http://web.cbr.ru/">
      <OnDate>dateTime</OnDate>
    </BankPapers>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/">
  <soap:Body>
    <BankPapersResponse xmlns="http://web.cbr.ru/">
      <BankPapersResult>
        <xsd:schema>schema</xsd:schema>xml</BankPapersResult>
      </BankPapersResponse>
    </soap:Body>
  </soap:Envelope>

```

Рисунок 5 - Метод запросов SOAP, предлагаемый ЦБ РФ

Центробанк предоставляет удобные черновики для SOAP-запроса (В этом примере запрашиваются данные о ценных бумагах), казалось бы, то, что нужно в условиях задачи. Тем не менее, запрос такого вида проводится исключительно через XML, как и работает только с документами XML, что влечет за собой ряд минусов:

- Отсутствие должной оптимизации файлов XML, файлы такого типа зачастую имеют значительный, nepозволительный для частой работы размер
- Вышеописанная поддержка лишь XML файлов
- Техническая сложность обращения с SOAP-протоколами из-за строгой стандартизации
- Низкая эффективность запроса в угоду большей безопасности – именно поэтому SOAP часто используется государством.

В случае этой работы требовалось быстро получать относительно малое количество информации, а после заносить её в базу данных. XML не предоставляет такой гибкости в обработке данных, поэтому от использования таких запросов и работы с API через SOAP пришлось отказаться.

1.4 БД MySQL

MySQL как система управления базами данных (далее СУБД) была выбрана неслучайно — она удобна в использовании, а так же имеет коннекторы для множества языков программирования, включая задействованные в работе php и Python. Таким коннектором выступала библиотека «PyMySQL», работающая связующим звеном между кодом Python и непосредственно БД. С её помощью можно в любой момент запустить файл с кодом Python,

```
import pymysql

def mysqlconnect():
    # To connect MySQL database
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "pass",
        db='College',
    )

    cur = conn.cursor()

    # Select query
    cur.execute("select * from STUDENT")
    output = cur.fetchall()

    for i in output:
        print(i)

    # To close the connection
    conn.close()
```

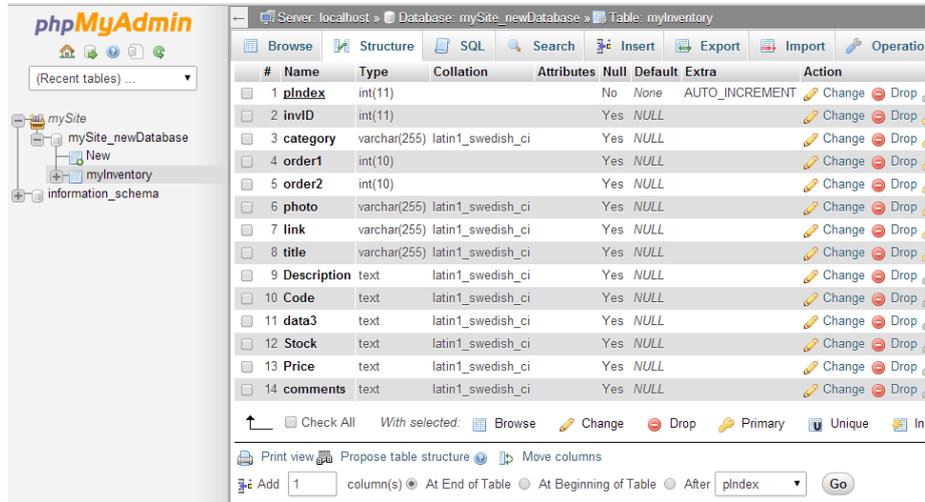
Рисунок 6 - Работа с данными в Python pymysql

где выполнялись надлежащие команды самой СУБД.

В приведенном в рисунке 6 примере программа подключается к БД «College», а затем выводит все данные из таблицы «Student», выполняя команду с синтаксисом MySQL напрямую («select * from STUDENT»). В распоряжении заказчика было множество файлов на разных языках программирования и с совершенно разными функциями, тем не менее, СУБД использовалась одна. Именно поэтому и был выбран способ с Python, ведь с помощью только одной встроенной библиотеки можно гибко взаимодействовать с данными таблицы.

Все БД и включённые в них таблицы размещаются в веб-приложении phpMyAdmin. Его основное преимущество — непосредственное администрирование MySQL без дополнительного ввода команд, а также непосредственная связь с языком php, на котором строятся многие программы и сайты со стороны заказчика. Тем не менее, чистый MySQL использован не был, ведь phpMyAdmin поддерживает и СУБД MariaDB.

MariaDB – СУБД, созданная на основе MySQL теми же разработчиками с целью дать более открытый доступ к модификации и модернизации исходного кода, в отличие от политики выкупившей MySQL компании Oracle. С технической точки зрения MySQL и



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	plindex	int(11)			No	None	AUTO_INCREMENT	Change Drop
2	invID	int(11)			Yes	NULL		Change Drop
3	category	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop
4	order1	int(10)			Yes	NULL		Change Drop
5	order2	int(10)			Yes	NULL		Change Drop
6	photo	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop
7	link	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop
8	title	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop
9	Description	text	latin1_swedish_ci		Yes	NULL		Change Drop
10	Code	text	latin1_swedish_ci		Yes	NULL		Change Drop
11	data3	text	latin1_swedish_ci		Yes	NULL		Change Drop
12	Stock	text	latin1_swedish_ci		Yes	NULL		Change Drop
13	Price	text	latin1_swedish_ci		Yes	NULL		Change Drop
14	comments	text	latin1_swedish_ci		Yes	NULL		Change Drop

Рисунок 7 - Интерфейс phpMyAdmin

MariaDB практически идентичны и взаимозаменяемы. В любом случае, именно последняя использовалась в БД заказчика.

Выше, на рисунке 7, приведен пример отображаемой в phpMyAdmin таблицы. В связи с закрытым доступом к таблице и невозможностью взаимодействовать с ней с помощью персонального компьютера, правильность работы кода в репозитории проверялась именно там. Рисунок 8 — логотипы MySQL и MariaDB соответственно.

1.5 PHP как инструмент запроса к данным

Php - один из немногих вариантов работы с веб-сайтами, язык, подобный C, основная



Рисунок 8 - Логотипы MySQL и MariaDB

структура наравне с HTML кодом в страницах заказчика. Поскольку целью проекта требуется создать доступный метод взаимодействия с данными, был выбран именно php. Код отправляет GET-запрос к таблице, забирая оттуда нужные данные, а после отображает в веб-странице под доменом, который при правильной настройке сможет открыть каждый желающий.

Принцип работы GET-запроса через браузер, отображенный в схеме на рисунке 9 (аналогичные действия и у запроса кода php):

Код совершает исходный запрос к веб-странице =>

Сервер даёт указания о заголовках HTTP по новому расположению =>

Код совершает запрос по новым указаниям с целью получить данные =>

Сервер отправляет запрашиваемые по индексу данные.

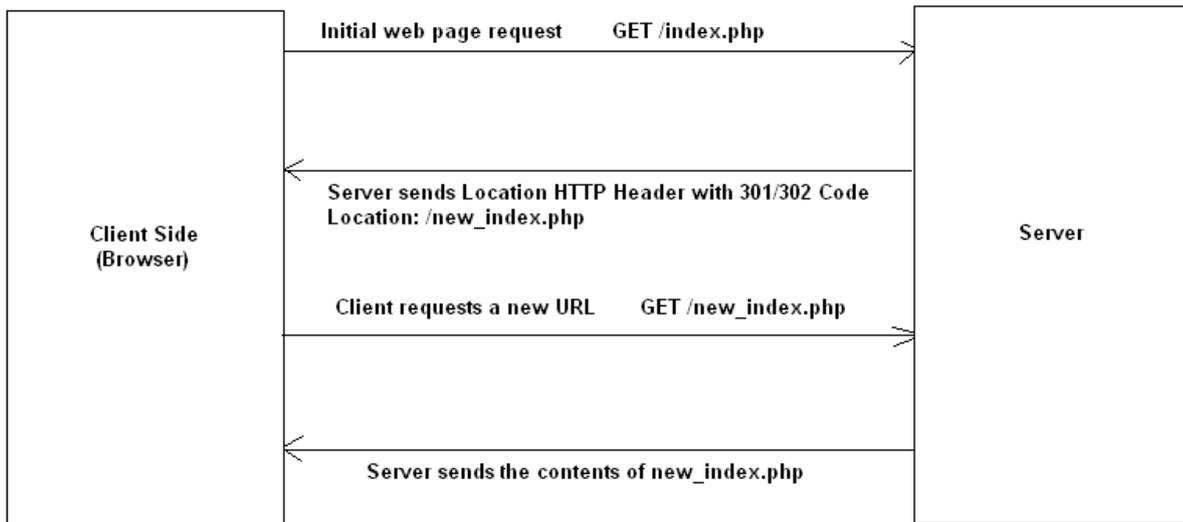


Рисунок 9 - Схема взаимодействия стороннего клиента с сервером

1.6 HTML с JS и библиотеки форматирования

Чтобы наглядно визуализировать продукт, в процессе работы было решено создать веб-страницу. В ней любой пользователь прямо через браузер сможет достать всю нужную информацию, содержащуюся в наших БД. В отличие от имеющего у ЦБ РФ сайта, данная веб-страница позволяет фильтровать данные по простейшим параметрам.

Основой страницы является «сырой» код HTML (HyperText Markup Language), который писался вручную, без использования программ-конструкторов (Webflow, Wix и др.). С одной стороны, использовать программы, не требующие работы с кодом, было бы эффективнее, но с другой, одной из целей проекта было расширить познание языков, а также уметь объяснить, как и за что отвечает та или иная часть кода.

Единственным значительным фреймворком стал W3.CSS, включающий множество базовых элементов форматирования CSS. Тем не менее, использовался он редко, так как изменение параметров под себя W3 не предусматривает. На рисунке 10 отображён способ

```

<!DOCTYPE html>
<html>
<title>My Web</title>
<meta name="viewport" content="width=device-width, initial-
scale=1">

<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
    
```

подключения W3.CSS.

Поскольку элементов в странице было не так много, все стили форматирования записывались прямоком в

документ с кодом HTML, без сторонних файлов стилей. Помимо этого, в файл включался футер (англ. «footer», подвал сайта) отдельным php

Рисунок 10 - Принцип подключения фреймворка с помощью атрибута href

документом, но писался он заказчиком, поэтому подробно освещаться он не будет.

В заключении, создание собственной веб-страницы позволит наглядно показать замысел проекта в удобном и простом виде, нежели в специализированных средах по типу phpMyAdmin.

1.7 Дополнительные инструменты

Продолжать работу в IDE на этапе с MySQL стало крайне затруднительно, ведь таблица блокировала любые попытки подключиться с локальных устройств. Решением проблемы стала FileZilla — FTP-клиент (File Transfer Protocol). С помощью данных, предоставленных заказчиком, удалось установить подключение к основному серверу и воспользоваться Log-файлами, показывающими основные ошибки кода. На рисунке 11 изображён логотип FileZilla.

Git репозиторий являлся точкой выгрузки данных с IDE, файл с которыми мог напрямую взаимодействовать с MySQL. Там находились все смежные с программой файлы,



Рисунок 11 - Логотип FileZilla



Рисунок 12 - Логотип

позволяющие запускать парсер раз в сутки для получения новой информации (С помощью Cron).

Логотип Git изображён на рисунке 12.

Модуль «datetime» - ключевая библиотека для преобразования данных формата «string» в объект типа «DateTime». Изображенное на рисунке 13 — способ подачи даты на сайте ЦБ РФ, его формат делает невозможным простой перенос этих данных в тип «Date» БД

MySQL, имеющих вид «ГГГГ-ММ-ДД». Datetime не только позволяет преобразовать строчные данные, но и заменить точки на нужный для формата знак. Рисунок 14 — типичный вариант преобразования строчного формата в Python-объект с помощью библиотеки «datetime».

Дата	
06.12.2023	
05.12.2023	


```

from datetime import datetime
date_in_string = '2021-11-19'
convert_date_to_object = datetime.strptime(date_in_string, '%Y-%m-%d')
print(convert_date_to_object)

```

Рисунок 13, 14 – Дата на сайте ЦБ и ее преобразование

2. Практическая часть

Практическая часть делится на три разных направления, сильно отличающиеся друг от друга содержанием. Большинство приведенных далее изображений будут взяты из вышеупомянутого Git репозитория (ссылку на файлы предоставить нельзя из-за ограниченного доступа. Вместо этого будет предоставлена общая ссылка на портал.), если не будет сказано обратного.

2.1 Часть Python и MySQL

2.1.1 Установка библиотек и подключение к прокси-серверу

Чтобы начать работу, требовалось установить нужные библиотеки. Изначально работа велась в Visual Studio Code, интерфейс которого изображён на рисунке 15, для распаковки библиотек требовалось создать виртуальную среду, в которой содержался бы и Python интерпретатор, и сами библиотеки. Команда «python -m venv [Название]» создавала такую, но в связи с проблемами с дисками на персональном ноутбуке, от виртуальной среды, как и от библиотеки, пришлось отказаться. Новым IDE стал PyCharm.

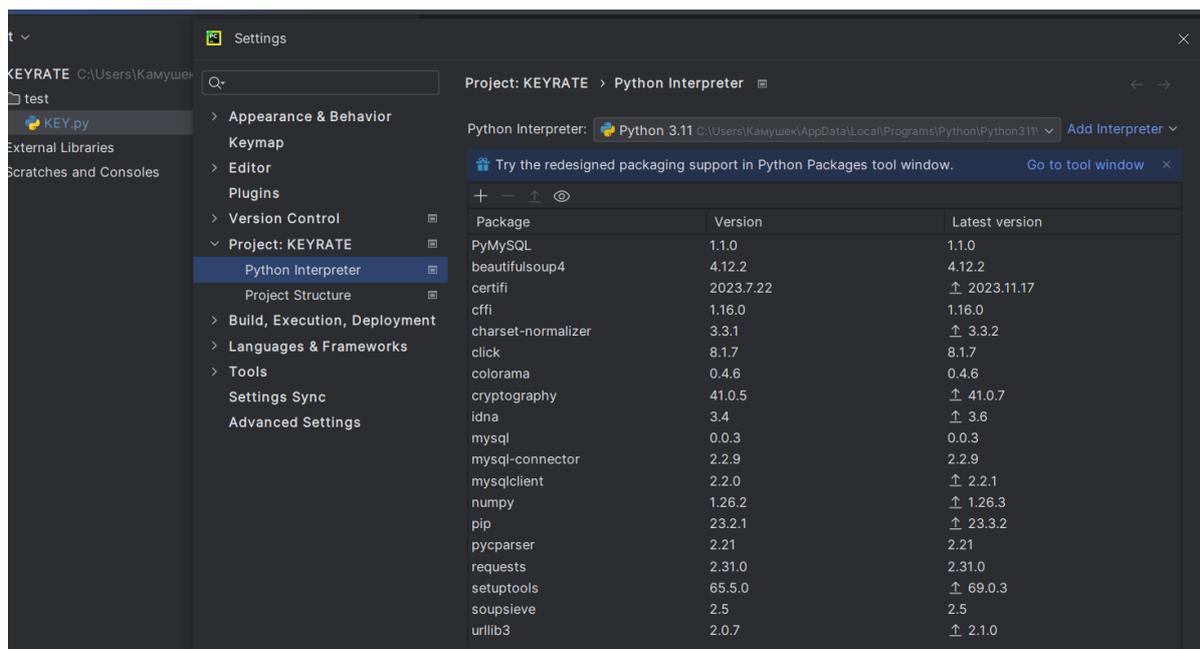


Рисунок 15 – Библиотеки в PyCharm

Так выглядит окно добавления библиотек в PyCharm (рисунок 16). Вместо добавления библиотек с помощью «pip install [Ваша Библиотека]» в консоли, достаточно было нажать значок плюса, ввести название библиотеки и установить по кнопке, без создания лишних папок.

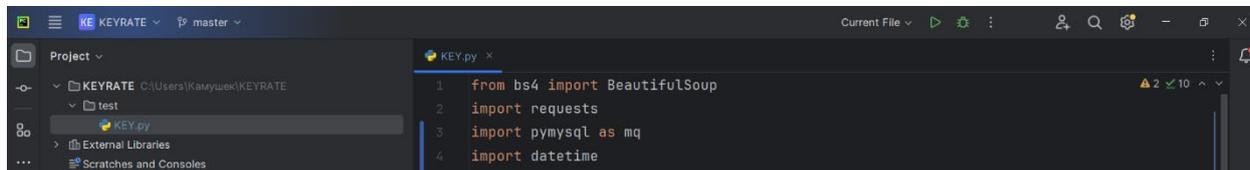


Рисунок 16 – Подключение библиотек

Вот нужные библиотеки: «bs4» и её префикс «BeautifulSoup» (Чтобы не писать его в самом коде), «requests», «pymysql» как mq, для упрощения кода, и «datetime». Для удобства, дальнейшие изображения кода будут из Git репозитория.

Поскольку основной сервер заказчика находится за границей, без прокси-сервера код выдавал ошибку: вероятнее всего, проблемой являлись санкции по отношению к нашим адресам. Тут указана и ссылка на страницу ЦБ РФ, и адрес прокси-сервера, который был скрыт, так как является платным и был куплен заказчиком. «Requests» подаёт запрос, включая и сервер, и саму ссылку запроса. Код приведён в рисунке 17. Далее начинается код с BS4 и Datetime.

```

http_proxy = "http://[redacted]"

proxies = {
    "https" : http_proxy,
}

url = "https://cbr.ru/hd_base/KeyRate/"
request = requests.get(url, proxies=proxies)

```

Рисунок 17 – Прокси-сервер

2.1.2 Парсинг и преобразование данных

```

13 request = requests.get(url, proxies=proxies)
14
15 soup = BeautifulSoup(request.text, "html.parser")
16 # Начинаем парсить
17 data = soup.find(string="Ставка").findNext('td')
18 for finalD in data:
19     date = finalD
20     print(date)
21     date_time_obj = datetime.datetime.strptime(date, '%d.%m.%Y')
22     print('Дата:', date_time_obj.date())
23     date=date_time_obj.date()
24
25 keydata = soup.find(string="Ставка").findNext('td').findNext('td')
26 for finalK in keydata:
27     key = finalK
28     print(key)
29     key=key.replace(',','.')

```

В рисунке 18, строка 15, BS4 получает весь HTML текст страницы.

Строка 17, парсер ищет дату последней ставки. Чтобы понять, что происходит, взглянем на HTML код, имеющийся на рисунке 19.

Таблица начинается с заголовков «th», строковые

The screenshot shows a web browser displaying a table of interest rates. The table has two columns: the date and the rate. The first row is highlighted in blue. The developer tools on the right show the HTML structure of the table, with the selected row expanded to show the date and rate cells.

Ставка	% годовых
16,00	804,36 × 32,68
16,00	03.01.2024
16,00	29.12.2023
16,00	28.12.2023
16,00	27.12.2023

Рисунок 18 & 19 – Парсер и код считываемой им страницы

названия которых (Ставка) находятся через «soup.find», а затем находятся данные на строку ниже (.findNext('td')), первый элемент «td», который и является датой. После, строка 21 отвечает за преобразование даты в удобный для таблицы формат, объектом «date_time_obj». Эта деталь критична для дальнейшего преобразования в формат «Date» уже в MySQL.

Схожим с датой образом находится и ставка, поиск проматывается ещё на одну строку вниз, строка 29 заменяет запятую на точку, всё для той же MySQL. Единственный минус – данные на сайте не всегда обновляются ежедневно, иногда задержка может составлять несколько дней, несколько дней могут быть полностью пропущены. Это зависит лишь от администраторов центробанка, и с этой проблемой придется считаться в будущем.

2.1.3 Румysql и начало работы через Git

Существует несколько библиотек MySQL для Python, но все они однотипны. Так или иначе, они позволяют давать SQL команды прямо через Python, присоединяясь к имеющимся БД. На данном этапе работа в PyCharm стала невозможна, так как БД была приватной, и доступ с персонального ноутбука блокировался.

В поиске ошибок помог FTP-клиент FileZilla, при вводе в который данных с 32 по 36 строки, приложение даёт доступ к устройству заказчика, позволяет смотреть логи ошибок. Опять же, содержание файлов конфиденциально, поэтому соответствующее изображение предоставлено не будет.

Изначально таблица была создана через Python, но кусок кода был удалён из-за ненужности, а также для ускорения выполнения. В рисунке 20, создаётся таблица с помощью

```

31  try:
32      connection = mq.connect(
33          host = "localhost",
34          user = "di-re_robolatoriyatools",
35          password = " ",
36          database = "di-re_robolatoriyatoolsdb"
37      )
38      print("successfully connected...")
39      print("#" * 20)
40
41
42      cursor = connection.cursor()
43      insert_query = "INSERT INTO cbrfkeyrate(DateStamp, Rate) VALUES ('{}', {});".format(date, key)
44      print(insert_query)
45      cursor.execute(insert_query)
46      print(f"{cursor.rowcount} details inserted")
47      connection.commit()
48      connection.close()
49
50  except Exception as ex:
51      print("Connection refused...")
52      print(ex)

```

Рисунок 20 – Подключение к базе данных и занос информации

команды «CREATE TABLE IF NOT EXISTS», создать её можно и в phpMyAdmin. Строки с 32 по 39 отвечают за подключение к БД, всё это обернуто в команду «try».

Если подключение успешно, создаётся курсор, нужный для исполнения команд (строка 42), а затем запрос, вводящий нужные данные в нужные строки с помощью встроенной в Python команды «.format()». После команда выполняется, а курсор отключается (строки 42-48). Если появляется ошибка, то она выводится в консоль. Многочисленные команды «print» нужны для отладки, оставлены в коде для устранения возможных ошибок в будущем.

2.1.4 Демонстрация отображаемой информации в phpMyAdmin

PhpMyAdmin – инструмент не для простого пользователя, поэтому некоторые элементы там не совсем интуитивны. Тем не менее, с помощью программы заказчика, парсер запускается каждую минуту — изменения ставки улавливаются с минутной точностью. Всё это заносится в БД:

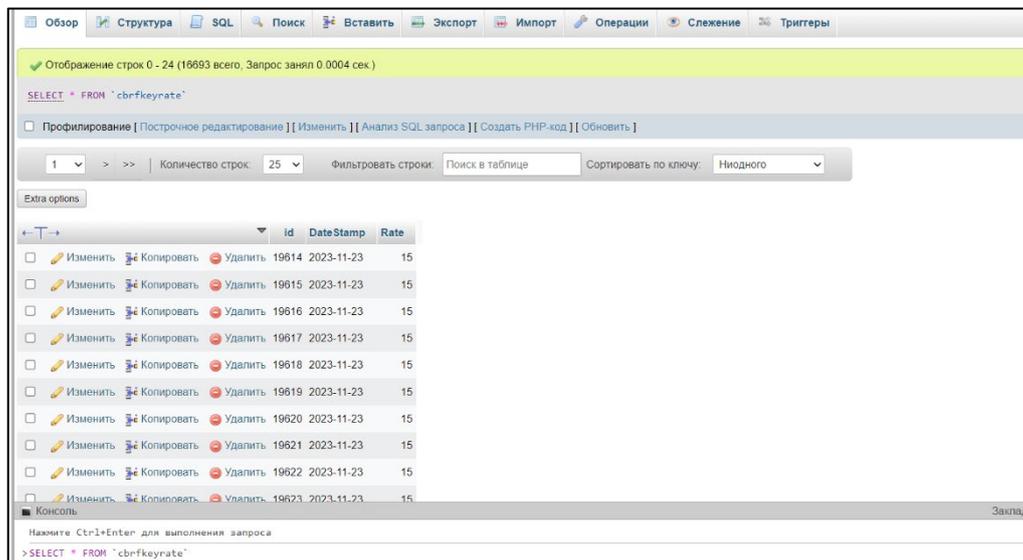


Рисунок 21 – Данные в phpMyAdmin

На рисунке 21 у каждой строки есть номерное обозначение, Datestamp имеет тип данных «Date», для которого и подключалась библиотека «datetime», Rate тоже имеет компактный типаж данных. С парсером было покончено, теперь работа перешла на язык Php, чтобы эти данные обратно вытащить.

2.2 Часть Php

Данные по значению ключевой ставки, дате, ставке инвестиций Роболатории должны выводиться при вводе в браузер ссылки домена /cbrfkeyrate/invest, но, чтобы сделать это, нужно:

- Создать Get запрос
- С помощью запроса достать данные
- Рассчитать ставку
- Вывести все данные

2.2.1 Получение данных через GET запрос

Пункт первый. Метод Get запроса уже встроен в общую библиотеку файлов, осталось его применить. Строка 37 в рисунке 22 отвечает за Get запрос и ссылку нашего запроса. «\$urldata» отвечает за индексы в ссылке, и этот конкретный запрос вступает лишь тогда, когда 0-й индекс будет под «invest», «\$isuser» отвечает за авторизацию пользователя, которая в данном случае не нужна.

Пункт второй, с помощью команды SQL и команды подсоединения, программа подключается к таблице и выбирает последние её значения, затем присваивает их

```

35 // Получение ключевой ставки
36 // GET /cbrfkeyrate/invest
37 if ($method === 'GET' && count($urldata) === 1 && $urldata[0] === 'invest' && $isuser===false) {
38
39     // Вытаскиваем ставку и дату
40     $sql="SELECT Rate, DateStamp FROM `cbrfkeyrate` ORDER BY id DESC LIMIT 1";
41     $result= $connect->query($sql);
42     $ratedata=$result->fetch_assoc();
43
44
45     $keypercent=$ratedata['Rate'];
46     $amount = $formData['amount'];
47     $month = $formData['month'];
48     if ($amount >= 50000 && $amount < 100000) {
49         $percent=$keypercent+2.5+($month-5)*0.25;
50     }elseif ($amount >= 100000 && $amount < 500000) {
51         $percent=$keypercent+3+($month-5)*0.25;
52     }elseif ($amount >= 500000 && $amount < 1000000) {
53         $percent=$keypercent+3.5+($month-5)*0.25;
54     } else {
55         $percent=0;
56     }
57     $totalamount=$amount+((($percent*$amount)/100)*$month/12;
58     $profit=((($percent*$amount)/100)*$month/12;
59     $permonthprofit=$profit/$month;

```

Рисунок 22 – GET запрос в
php

соответствующим переменным (строки 40-47).

Пункт третий, данные проходят через формулы с вычислениями. Можно заметить переменные «\$amount» и «\$month», которые задаются самим пользователем в командной строке. С точки зрения потребителя, довольно неудобно, но весь этот код помещается в калькулятор с удобным интерфейсом. Его разработка велась другим человеком, а создать php запрос нужно было для другой веб-страницы. Таким образом, были убиты оба зайца: оптимизирован калькулятор инвестиций, и создана основа для будущего приложения.

Стоит отметить, что некоторая часть кода находилась в отдельных файлах/была сделана другим человеком, соответственно и не была показана.

2.2.2 Подстановка данных в переменные

```

61
62
63 //Выводим ставку и дату
64 echo json_encode(array(
65     'rate' => $ratedata['Rate'],
66     'keypercent' => $percent,
67     'amount' => $amount,
68     'totalamount' => $totalamount,
69     'profit' => $profit,
70     'permonthprofit' => $permonthprofit,
71     'months' => $month,
72 ),JSON_UNESCAPED_UNICODE);
73
74 return;
75 }
76
77 // Возвращаем ошибку
78 header('HTTP/1.0 400 Bad Request');
79 echo json_encode(array(
80     'error' => 'Bad Request'
81 ));

```

Рисунок 23 – Вывод данных

И наконец пункт 4, вывод данных. Всё выводится в формате json, после строчной записи выводится переменная. Итого получаем ключевую ставку, количество внесенных денег, количество денег, которые инвестор получит, выгода, месячная выгода и количество месяцев, на которые производится взнос. Строки 78-81 в рисунке 23 сообщают о возникшей ошибке.

2.3 Часть веб-страницы

Этап работы, связанный с созданием и последующим оформлением HTML документа, требовал увеличить количество GET запросов, чтобы удовлетворить целям задачи. Весь код, связанный с сайтом, делился на четыре файла: два файла с HTML телом (основная и справочная страницы), вышеописанный файл php, содержащий GET запросы, а также файл JavaScript, который отвечал за отображение графика и другие, побочные функции. Созданный график должен отображать данные по ставке за неделю, месяц, год.

2.3.1 Структура основного файла

Файлы HTML всегда делятся на две части — head и body, содержащие виртуальную информацию (название вкладки, стили, то, что невидимо пользователю в сыром виде) и тело страницы со всеми элементами, отображаемыми там.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Ключевая ставка ЦБ РФ</title>
5
6 <meta charset="UTF-8">
7 <meta name="description" content="Демоверсия Базы данных ЦБ РФ"/>
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
10 <meta name="yandex-verification" content="6cc493c019945920" />
11 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
12 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Montserrat">
13 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
14 <link rel="stylesheet" href="https://my.robolatoriya.com/static/loader.css">
15 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
16 <link rel="icon" type="image/png" href="favicon.png" />
17 <style>
18 body,h1,h2,h3,h4,h5,h6 {font-family: "Montserrat", sans-serif}
19 .w3-bar,.footer,.content,h1,button {font-family: "Montserrat", sans-serif}
20 .fa-anchor,.fa-list {font-size:200px}
21 @media (max-width: 1023px) {
22   .ord-form {margin:0!important;width:100!important}
23 }
24 .top-text {text-align: left; font-size: 30px; color:#FEF0FC; margin:0; padding-bottom:0px; padding-top:38px; @media (max-width:1023px) {
25   text-align: left; font-size: 20px; color:#FEF0FC; margin:0; padding-bottom:0px; padding-top:28px;}
26 }
27 html {
28   body { min-height: 100vh;
29     width:100%;}
30   .top-rectangle { width:100%;height:15.0vh; background: #11708A; @media (max-width: 1023px) {
31     width:100%;height:10.0vh; background: #11708A;}

```

Рисунок 24 – часть элемента «head»

Первая часть документа в рисунке 24 содержит все нужные фреймворки, а также включает в себя элемент «style» (строка 17), содержащий внутри CSS характеристики каждого элемента. Название даётся любое, главное включить его в нужный элемент в body. Каждый класс в стилях начинается с точки, в отличие от id (которое начинается с «#»). Id в файле использовались исключительно для работы с JavaScript, поэтому в оформлении их не найти.

«@media» — медиа-запрос, позволяющий отформатировать документ, опираясь на технические характеристики устройства пользователя. Говоря проще, для разной величины экрана нужно разное оформление сайта. Для этого было установлено ограничение в ≤ 1023 px, при соответствии условиям которого подставлялись другие параметры отступов, величины шрифта и масштаба некоторых элементов.

```

59 <a href="about.php" class="w3-button">API</a>
60 </div>
61 </div>
62 <?
63 $date = strtotime("today");
64 $date7 = strtotime("-7 day");
65 $date30 = strtotime("-1 month");
66 $date365 = strtotime("-1 year");
67 $date7=date('Y-m-d', $date7);
68 $date30=date('Y-m-d', $date30);
69 $date365=date('Y-m-d', $date365);
70 $date=date('Y-m-d', $date);
71 ?>
72 <div id="app">
73 <div class="w3-light-gray w3-display-container" id="home">
74 <div class="w3-row-padding w3-padding-64 w3-center">
75 <h1 class="w3-margin w3-jumbo">{{ rate }}%/h1>
76 <h3>ключевая ставка ЦБ РФ сегодня</h3>
77 <a class="w3-button w3-amber w3-round-xxlarge w3-padding-large w3-xlarge w3-margin-top" onclick="showhide(mydiv)"><b>Смотреть график!</b>
78 </div>
79 </div>
80
81 <!-- Second Grid -->
82 <div class="w3-row-padding w3-white w3-padding-32 w3-container" id="about">
83 <div class="w3-content">
84 <div class="w3-twothird">
85 <h1>Открытое API</h1>
86 <h4 class="w3-padding-32">Наш сервис предоставляет открытое API, благодаря котрому вы можете получать в реальном времени данные о текуще
87 <a href="/about.php" class="w3-white w3-button w3-round-xxlarge w3-xlarge w3-border w3-hover-gray w3-left">Попробовать бесплатно</a>
88 </div>
89 <div class="w3-third w3-center">
90 <i class="fa fa-rocket w3-padding-16 w3-text-amber w3-margin-right" style="font-size:250px;"></i>
91 </div>
92 </div>

```

Рисунок 25 – преобразования с датой, отображение сегодняшней ставки

В начале (строки 63-70, рисунок 25) записана php-вставка, в будущем применяемая в графике. С помощью описанного в п. 2.2.1 запроса страница получает последние данные, а затем выводит их пользователю. Строки классов объектов содержат описанные в стилях страницы классы, иногда в оформлении элементов применяется W3.CSS, функции которого значительно экономят время на централизации.

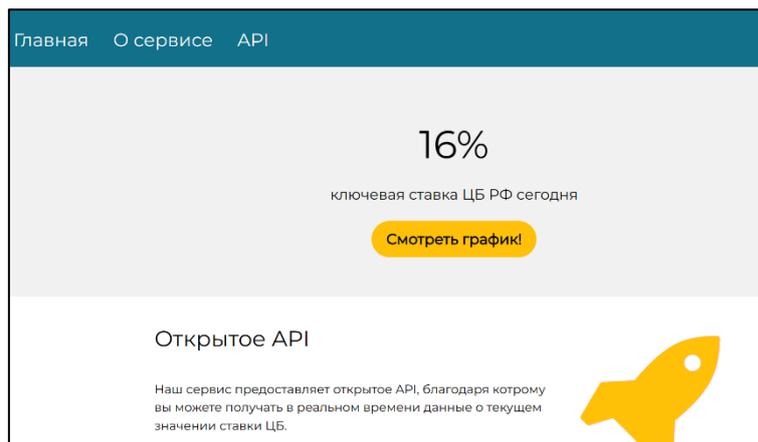


Рисунок 26 – отображаемая информация

Примечание: код, отвечающий за хедер, не вошёл в изображение, полные коды файлов (а также полные скриншоты сайта) будут предоставлены в приложениях.

Запрос «.../cbrfkeyrate/all» в рисунке 27 при отсутствии дополнительных настроек выдает всю имеющуюся в базе данных информацию, опираясь на «стартовое» и «конечное»

значения. Для этого (см рисунок 25) в HTML файле и были записаны значения date, как заготовки под информацию за неделю, месяц и год.

```

34
35 // Получение ключевой ставки
36 // GET /cbrfkeyrate/all
37 if ($method === 'GET' && count($urlData) === 1 && $urlData[0] === 'all' && $isuser===false) {
38     if(empty($formData['start']) && empty($formData['end'])){
39         $sql="SELECT Rate, DateStamp FROM `cbrfkeyrate`";
40     } else {
41         $sql="SELECT Rate, DateStamp FROM `cbrfkeyrate` WHERE DateStamp BETWEEN '". $formData['start']."' AND '". $formData['end']."'";
42     }
43

```

Рисунок 27 – новый GET запрос

В строках 97-99 вызывается JS функция `getData`, соответствующие временные параметры подставляются внутрь, а затем вся информация выводится в графике. Заметим, что и кнопки, и график, обернуты в класс «`graph`», который используется в функции JS, отвечающей за скрытие/отображение графика.

2.3.2 Код JavaScript

```

81 <!-- Second Grid -->
82 <div class="w3-row-padding w3-white w3-padding-32 w3-container" id="about">
83 <div class="w3-content">
84 <div class="w3-twothird">
85 <h1>Открытое API</h1>
86 <h4 class="w3-padding-32">Наш сервис предоставляет открытое API, благодаря которому вы можете получать в реальном времени данные о текуще
87 <a href="/about.php" class="w3-white w3-button w3-round-xxlarge w3-xlarge w3-border w3-hover-gray w3-left">Попробовать бесплатно</a>
88 </div>
89 <div class="w3-third w3-center">
90 <i class="fa fa-rocket w3-padding-16 w3-text-amber w3-margin-right" style="font-size:250px;"></i>
91 </div>
92 </div>
93 </div>
94
95 <div class="w3-center" id="graph">
96 <div class="container">
97 <button class="w3-button rowbutton" @click="getData(<? echo "{$date7}"; ?>,<? echo "{$date}"; ?>)">Неделя</button>
98 <button class="w3-button rowbutton" @click="getData(<? echo "{$date30}"; ?>,<? echo "{$date}"; ?>)">Месяц</button>
99 <button class="w3-button rowbutton" @click="getData(<? echo "{$date365}"; ?>,<? echo "{$date}"; ?>)">Год</button>
100 </div>
101
102 <div class="w3-center">
103 <div id="chart_div" class="chart" style="height: 300px;"></div>
104 </div>
105 </div>
106 </div>
107
108 </div>
109 <?
110 include 'footer.php';
111 ?>

```

Рисунок 28 – кнопки информации за неделю, месяц и год

```

getData: function(startdate, enddate){
    axios.get('https://tools-api.robolatoriya.com/cbrfkeyrate/all?start='+startdate+'&end='+enddate)
    .then(function (response) {
        console.log(response.data);
        app.alldata=response.data['Dataset'];
        chart(app.alldata);
    })
    .catch(function (error) {
        console.log(error);
    });
},

```

Рисунок 29 – функция `getData`

Функция `getData` достаёт информацию по заданным временным маркировкам (Общему массиву в `php` присвоено название «`Dataset`», теперь он имеет значение `data`).

Одновременно с инициализацией, функция выводит все данные в консоль, подтверждая собственную работоспособность.

```

55 function chart(items) {
56   var data = new google.visualization.DataTable();
57   data.addColumn('string', 'Дата');
58   data.addColumn('number', 'Значение');
59
60   data.addRows(items.map(function(item){return [item.DateStamp,parseFloat(item.Rate)]}));
61
62   var options = {
63     legend: {position: 'none'},
64     chartArea: { "left": "auto", "right": "auto", "top": 50, "bottom": "10", "width": "80%", "height": "60%",
65     width: "100%",
66     hAxis: {
67       textPosition: 'none',
68     },
69   };
70
71   var chart = new google.visualization.LineChart(document.getElementById('chart_div'));
72
73   chart.draw(data, options)
74
75 }

```

Рисунок 30 – Функция графика, основанная на образце Google Charts

За параметры графика отвечает функция «chart», изображенная на рисунке 30. Из data вынимаются две колонки, которым присваивается строчное и номерное значение соответственно. Затем (строка 60) добавляются строки с данными даты и ставки, формат последней меняется на число с плавающей точкой. Строка 73 отображает график, а также указанные опции форматирования.

```

76
77 mydiv = document.getElementById("graph");
78
79 function showhide(panels) {
80   panels.style.display = (panels.style.display !== "none") ? "none" : "block";
81 }
82
83
84 function redraw() {
85   chart.draw(data, options);
86 }

```

Рисунок 31 – функций сокрытия графика и его обновления

Функция на строке 79, рисунок 31, работает с форматированием CSS, придавая/убирая значение «none» классу graph. Кнопка (см рисунок 26), в свою очередь, привязана к функции событием onclick.

Функция обновления графика стоит обособленно, и вызывается при определенных условиях, связанными с перезагрузкой страницы. Выполняет она то же самое, что и последняя строка основной функции.

2.3.3 Финальная демонстрация

Проверим отображение графика за год (с начала создания таблицы), рисунок 32:

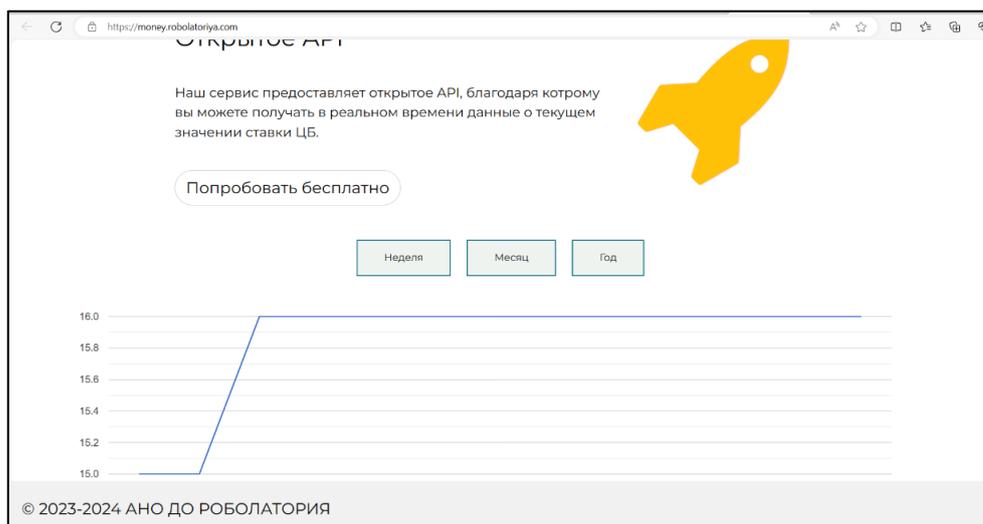


Рисунок 32

При наведении на точку в графике, он отображает ставку и дату её публикации во всплывающем окне, показано на рисунке 33:

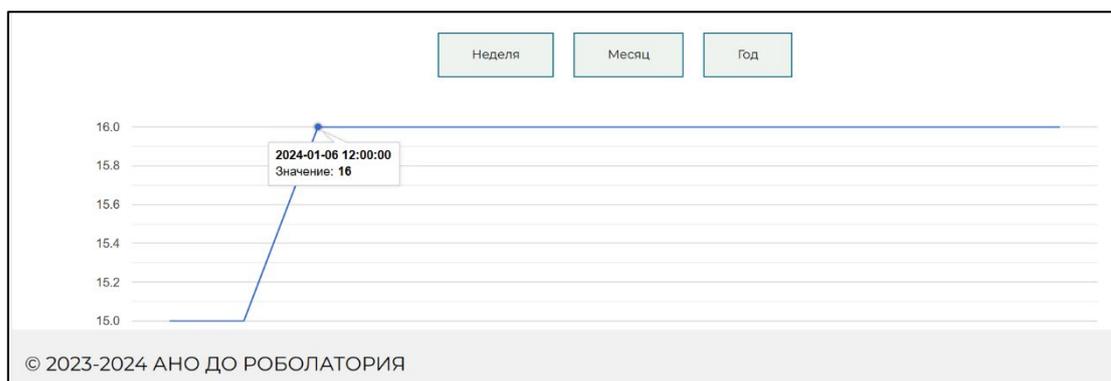


Рисунок 33

3. Заключение

Все поставленные задачи были выполнены к сроку. За многомесячный период работы были сделаны:

1. Парсер ключевой ставки ЦБ РФ
2. Таблица в MySQL с ежедневным обновлением данных
3. Открытые API
4. Приложение с наглядной демонстрацией собранных данных

Созданный продукт не только поможет продемонстрировать мои навыки программирования, но и будет использоваться на практике в инвестиционном калькуляторе. В будущем возможна разработка парсеров иной информации центробанка, расширение до полноценной библиотеки с API, с помощью которой любой разработчик сможет достать нужную информацию, запустив лишь один блок кода.

Python без языков-помощников, бесспорно, не имеет весомой силы. Таким образом, удалось расширить познание других языков, окунуться так и в frontend программирование, так и в backend.

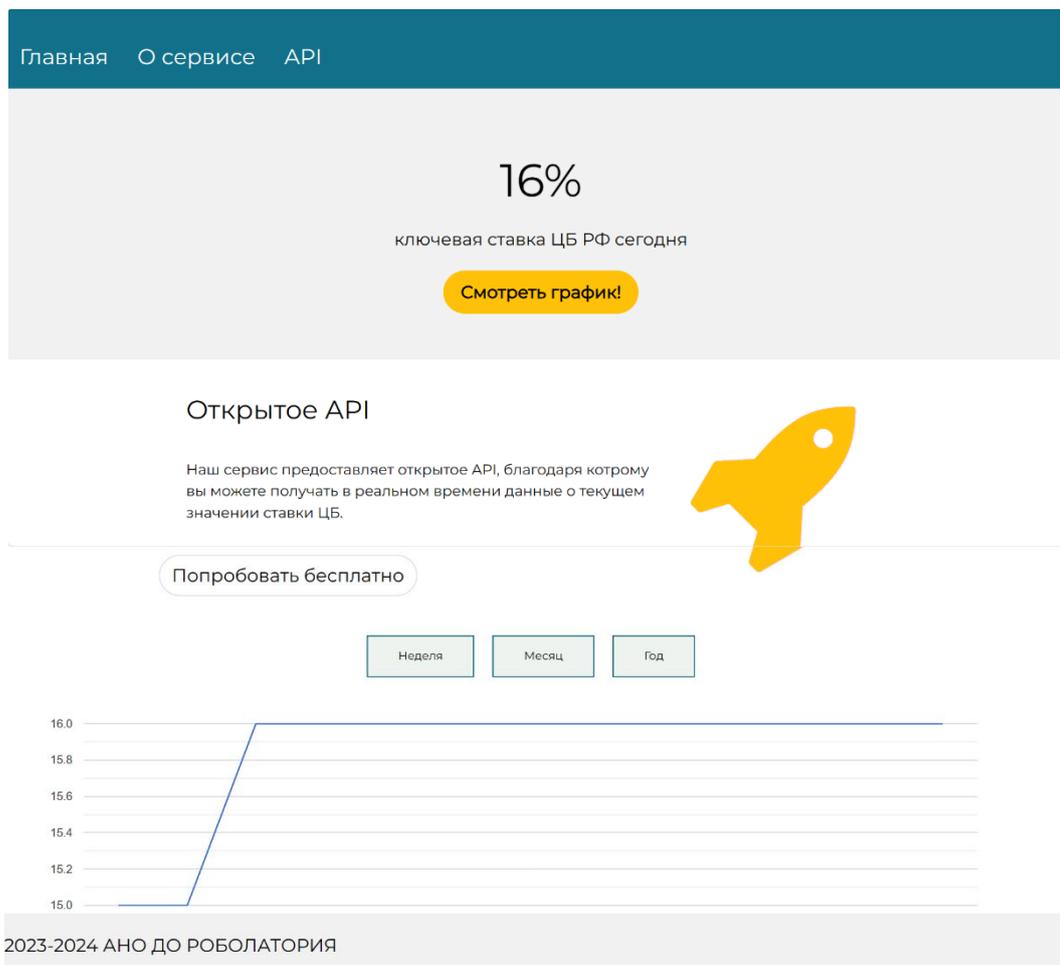
Благодарности: Огромное спасибо Лашиной Татьяне Сергеевне — за помощь в оформлении и публикации проекта, а также в предоставлении гибкого графика.

Список использованных информационных ресурсов

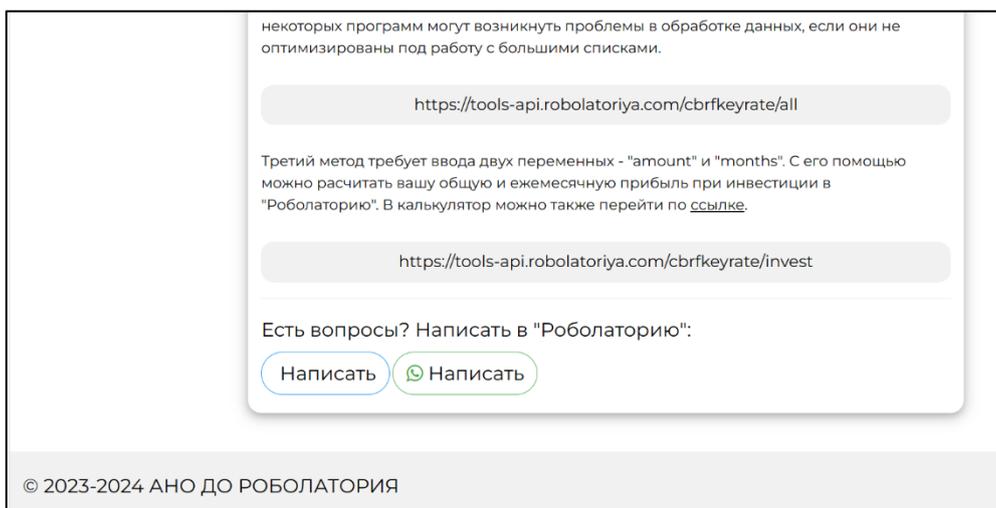
1. Гайд по подключению W3.CSS (рисунок 10), URL: https://www.w3schools.com/w3css/w3css_downloads.asp
2. Логотип FileZilla (рисунок 11), URL: <https://www.sanomnews.net/wp-content/uploads/2019/01/FileZilla.jpg>
3. Логотип Git (рисунок 12), URL: <https://fuzeservers.ru/wp-content/uploads/3/0/c/30c29ce4cc08523ecc6e1f205bc207d0.jpeg>
4. Сайт ключевой ставки ЦБ РФ (рисунки 13, 19), URL: https://cbr.ru/hd_base/KeyRate/
5. Информация о Datetime (рисунок 14), URL: <https://blog.logrocket.com/python-datetime-module-handling-dates-time/>
6. Git репозиторий, URL: <https://git.robolatoriya.org/user/login>
7. Большая часть ошибок исправлялась с помощью информации, полученной на сайте Stack Overflow, URL: <https://stackoverflow.com/>
8. Инвестиционный калькулятор, URL: <https://money.robolatoriya.com/>

Приложение

Скриншоты веб-страницы:



The screenshot shows a web page with a dark blue header containing navigation links: Главная, О сервисе, API. The main content area features a white box with a title 'Инвестиционный Калькулятор'. Below the title, there is a paragraph of text: 'База данных создана в рамках проектной работы при поддержке АНО ДО "Роболатория", Гончаренко А.А. Обратившись к странице с данными, любой человек/организация сможет получить данные о ставке с суточной точностью.' Below this text, there is a section titled 'API' with a paragraph: 'Нужная информация получается через использование API (Application Programming Interface) программыми разного рода.' Below this text, there is a yellow button with the URL 'https://tools-api.robolatoriya.com/cbrfkeyrate'. Below the button, there is a paragraph: 'Первый метод представлен выше. Он отдаёт актуальное значение ключевой ставки на данный момент.' Below this text, there is a paragraph: 'Второй метод даёт доступ ко всему массиву данных о ключевой ставке ЦБ РФ по датам.' Below this text, there is a paragraph: 'Примечание: по прошествии времени, данных в таблице становится больше. У'



Полный код основной страницы HTML, кроме JS ссылок:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Ключевая ставка ЦБ РФ</title>
5
6 <meta charset="UTF-8">
7 <meta name="description" content="Демоверсия Базы данных ЦБ РФ"/>
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
10 <meta name="yandex-verification" content="6cc493c019945920" />
11 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
12 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Montserrat">
13 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
14 <link rel="stylesheet" href="https://my.robolatoriya.com/static/loader.css">
15 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
16 <link rel="icon" type="image/png" href="favicon.png" />
17 <style>
18 body,h1,h2,h3,h4,h5,h6 {font-family: "Montserrat", sans-serif}
19 .w3-bar,.footer,.content,h1,button {font-family: "Montserrat", sans-serif}
20 .fa-anchor,.fa-list {font-size:200px}
21 @media (max-width: 1023px) {
22   .ord-form {margin:0!important;width:100%!important}
23 }
24 .top-text {text-align: left; font-size: 30px; color:#FEFCFC; margin:0; padding-bottom:0px; padding-top:38px; @media (max-width:1023px) {
25   text-align: left; font-size: 20px; color:#FEFCFC; margin:0; padding-bottom:0px; padding-top:28px;}
26 }
27 html { }
28 body { min-height: 100vh;
29   width:100%;}
30 .top-rectangle { width:100%;height:15.0vh; background: #11708A; @media (max-width: 1023px) {
31   width:100%;height:10.0vh; background: #11708A;}
32 }
33
34 .datapanel {height: 30vh; background: #EEF3EF; width: 60%; margin-left: 400px; margin-top:100px; @media (max-width:1023px) {
35   height: 20vh; background: #EEF3EF; width: 80%; margin-left: 40px; margin-top:50px;}}
36
37 .Datapanel {height: 50vh; background: #CED6DC; width: 60%; margin:auto; @media (max-width:1023px) {
38   height: 20vh; background: #B7C3CE; width: 80%; margin-top:50px; margin:auto}}
39
40
41 .container {margin-top:10px;}
42 .rowbutton {
43   display: inline-block;
44   padding: 15px 40px;
45   text-align: center
46   font-size: 25px;
47   cursor:pointer;
48   border: 2px solid #11708A;
49   background-color: #EEF3EF; margin: 10px;}
50
51
52 </style>
53 </head>
54 <body>
55 <div class="top-rectangle w3-middle">
56   <div class="top-text w3-bottomleft">
57     <a href="index.php" class="w3-button">Главная</a>
58     <a href="about.php" class="w3-button">О сервисе</a>
59     <a href="about.php" class="w3-button">API</a>
60   </div>
61 </div>
62 </>

```

```

63  $date = strtotime("today");
64  $date7 = strtotime("-7 day");
65  $date30 = strtotime("-1 month");
66  $date365 = strtotime("-1 year");
67  $date7=date('Y-m-d', $date7);
68  $date30=date('Y-m-d', $date30);
69  $date365=date('Y-m-d', $date365);
70  $date=date('Y-m-d', $date);
71  ?>
72  <div id="app">
73  <div class="w3-light-gray w3-display-container" id="home">
74  <div class="w3-row-padding w3-padding-64 w3-center">
75  <h1 class="w3-margin w3-jumbo">{{ rate }}%</h1>
76  <h3>ключевая ставка ЦБ РФ сегодня</h3>
77  <a class="w3-button w3-amber w3-round-xxlarge w3-padding-large w3-xlarge w3-margin-top" onclick="showhide(mydiv)"><b>Смотреть график!</b></a>
78  </div>
79  </div>
80
81  <!-- Second Grid -->
82  <div class="w3-row-padding w3-white w3-padding-32 w3-container" id="about">
83  <div class="w3-content">
84  <div class="w3-twothird">
85  <h1>Открытое API</h1>
86  <h4 class="w3-padding-32">Наш сервис предоставляет открытое API, благодаря котрому вы можете получать в реальном времени данные о текущей ставке ЦБ РФ.
87  <a href="/about.php" class="w3-white w3-button w3-round-xxlarge w3-xlarge w3-border w3-hover-gray w3-left">Попробовать бесплатно</a>
88  </div>
89  <div class="w3-third w3-center">
90  <i class="fa fa-rocket w3-padding-16 w3-text-amber w3-margin-right" style="font-size:250px;"></i>
91  </div>
92  </div>
93  </div>
94
95  <div class="w3-center" id="graph">
96  <div class="container">
97  <button class="w3-button rowbutton" @click="getData(<? echo '{{ $date7 }}'; ?>,<? echo '{{ $date }}'; ?>)">Неделя</button>
98  <button class="w3-button rowbutton" @click="getData(<? echo '{{ $date30 }}'; ?>,<? echo '{{ $date }}'; ?>)">Месяц</button>
99  <button class="w3-button rowbutton" @click="getData(<? echo '{{ $date365 }}'; ?>,<? echo '{{ $date }}'; ?>)">Год</button>
100 </div>
101
102 <div class="w3-center">
103 <div id="chart_div" class="chart" style="height: 300px"></div>
104 </div>
105 </div>
106 </div>
107
108 </div>
109 <?
110 include 'footer.php';
111 ?>
112

```

ДИСТАНЦИОННО-УПРАВЛЯЕМЫЙ ПОДВОДНЫЙ РОБОТ

Дорофеев Владимир Алексеевич

Москва г, ГБОУ школа 1265, 10 класс

Секция: Многоцелевые роботы и летательные аппараты

Научный руководитель: Сухоцкий Владимир Андреевич, педагог ГБОУ «Воробьевы горы»,
Центр технического образования

Аннотация

Цель работы – разработать бюджетный вариант подводного робота, который можно использовать для видео и фотосъёмки различных подводных объектов.

Проектирование осуществлялось с использованием САПР «Компас 3D». Для изготовления конструктивных элементов использовались станок лазерной резки, 3D-принтер, настольный токарный станок и слесарный инструмент. Особое внимание в работе уделено использованию доступных на российском рынке недорогих комплектующих с целью возможности тиражирования конструкции в школьных лабораториях. В качестве корпуса робота используется прозрачная колба от фильтра для питьевой воды, выдерживающая давление до 8 бар. Для ориентации в пространстве используются акселерометр и электронный компас, для избегания столкновений планируется применить ультразвуковой датчик расстояния. С целью передачи оператору высококачественного изображения робот оснащен Full HD видеокамерой.

Наземная станция управления роботом на базе ноутбука использует российскую операционную систему ALT Линукс. Для управления оператор использует джойстик. Для взаимодействия наземной станции с роботом используется оптоволоконный кабель. В надводном положении предусмотрена возможность использования сети WiFi, что позволяет сохранить управляемость робота при обрыве оптоволоконного кабеля.

Цели проекта достигнуты, создан прототип дистанционно-управляемого подводного робота из доступных комплектующих.

Введение

Цель работы

Разработать бюджетный вариант подводного робота, который можно использовать для видео и фотосъёмки различных подводных объектов.

Задачи проекта

- Проанализировать области применения дистанционно управляемых подводных роботов в различных областях науки и техники;
- Произвести подбор и тестирование основных комплектующих из числа производимых серийно, в том числе, не используемых ранее в подводной технике;
- Подобрать, настроить и доработать управляющее ПО с открытым исходным кодом;
- Разработать систему проводного управления роботом по оптоволокну;
- Разработать систему беспроводного управления роботом в надводном положении для использования в аварийных ситуациях, а также для тестирования;
- Провести подбор и тестирование, а также разработку и изготовление необходимых блоков управляющей электроники и датчиков;
- Провести проектирование и изготовление конструктивных элементов;
- Провести компоновку элементов и регулировку плавучести.

Актуальность проекта

Подводная робототехника – это область науки и техники, посвященная разработке, производству и использованию роботов в водной среде. Подводные роботы, оснащенные камерами, стали незаменимым инструментом в таких областях, как океанография, экологический мониторинг, инспекция подводной инфраструктуры, поисково-спасательные операции и во многих других. Их способность работать в условиях, недоступных или опасных для человека, делает их ключевым элементом современных подводных исследований.

На рынке подводных роботов разработанный вариант может заинтересовать потребителей, которые решают задачи, где необходимо использование видеокамеры для мониторинга каких-либо объектов, процессов.

Конструкция робота

Общий вид робота представлен на рисунке 1.



Рис. 1 Общий вид робота

Конструктивно робот представляет собой герметичный корпус с установленными четырьмя винтомоторными агрегатами мощностью до 350 ватт каждый, два из которых используются для вертикального перемещения, а два – для горизонтального. Расположение двигателей отражено на схеме (рис.2).

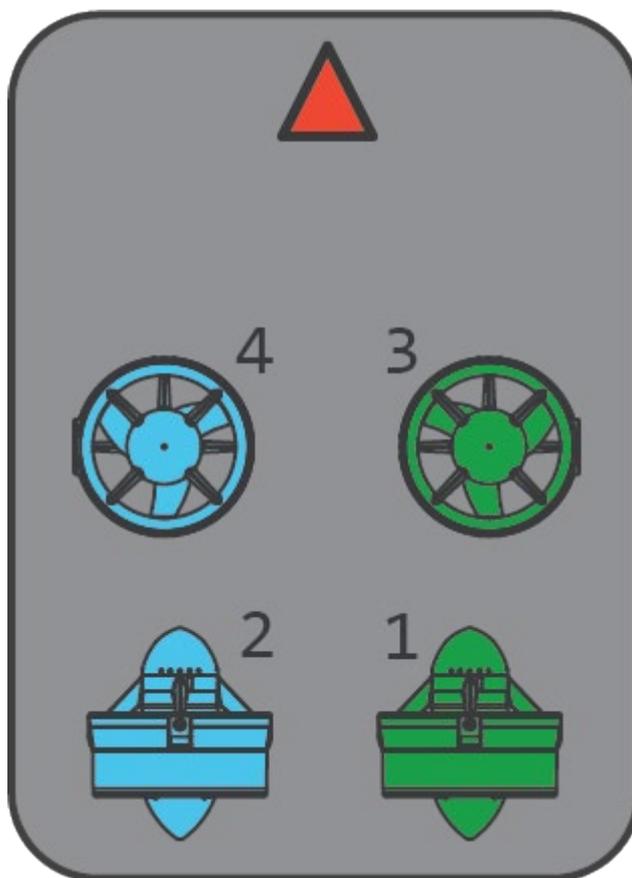


Рис. 2 Схема расположения винтомоторных агрегатов

Расчётная максимальная глубина погружения робота – 15 метров. Габаритные размеры – 440x420x260мм. В надводном положении предусмотрено управление через Wi-F, при движении под водой для управления роботом используется оптоволоконный кабель.

С учетом максимального разрядного тока аккумулятора, мощность винтомоторных агрегатов ограничена программно на уровне 25%, запас хода при таком режиме составляет 20 минут.

Корпус

В качестве корпуса использована колба магистрального фильтра для питьевой воды (рис. 3). Она прозрачная, недорогая, герметичная, имеет вводы под резьбу 1”, что позволяет использовать стандартные сантехнические элементы для ввода проводов внутрь колбы.



Рис. 1 Колба магистрального фильтра

В соответствии с данными производителя, использованная колба выдерживает внутреннее давление до 8 бар. С учетом формы колбы можно предположить, что максимальное внешнее давление, выдерживаемое колбой, составит также не менее 8 бар, что теоретически позволяет использовать робота на глубине до 80 метров. Однако, при конструировании подводных аппаратов существует проблема герметизации вводов коммуникаций (сигнальных и силовых кабелей) в герметичный корпус робота. В представленной конструкции робота для решения этой проблемы используется создание избыточного давления внутри колбы. Для этого непосредственно перед погружением с помощью автомобильного компрессора производится закачка воздуха в колбу до достижения избыточного давления около 1,5 бара. Предположительно, дальнейшее повышение давления внутри колбы может негативно отразиться на работоспособности литий-ионного аккумулятора. В связи с изложенным принято решение ограничить максимальную глубину погружения робота 15 метрами.

Винтомоторные агрегаты

В качестве винтомоторных агрегатов были выбраны бесколлекторные электродвигатели с гребными винтами, предназначенными для использования под водой. Для

управления электродвигателями используются контроллеры ESC (Electronic Speed Control) на максимальный ток 40А. Указанные контроллеры имеют возможность водяного охлаждения, которое в конструкции робота пока не используется, так как контроллеры не работают в режиме максимальной мощности.

Мощность винтомоторных агрегатов ограничена программным образом на уровне 25%. Максимальный ток, потребляемый одним агрегатом при питании от аккумулятора напряжением 24 вольта, составляет, таким образом, составляет $0,25 \cdot 350 / 24 = 3,65\text{А}$. При движении по горизонтали используются два винтомоторных агрегата. Емкость аккумулятора составляет $2,5\text{А} \cdot \text{ч}$. Таким образом, расчетный запас хода по горизонтали до полного разряда аккумулятора составит $2,5\text{А} \cdot \text{ч} / (3,65\text{А} \cdot 2) \approx 0,34$ часа (около 20 минут), что при измеренной экспериментально скорости робота $0,2$ м/с составит $0,34 \cdot 60 \cdot 60 \cdot 0,2 \approx 245$ метров в режиме максимальной скорости.



Рис. 2 Винтомоторный агрегат



Рис. 3 ESC контроллер

Электронные модули

Шасси

Все электронные модули расположены внутри корпуса на шасси из оргстекла:

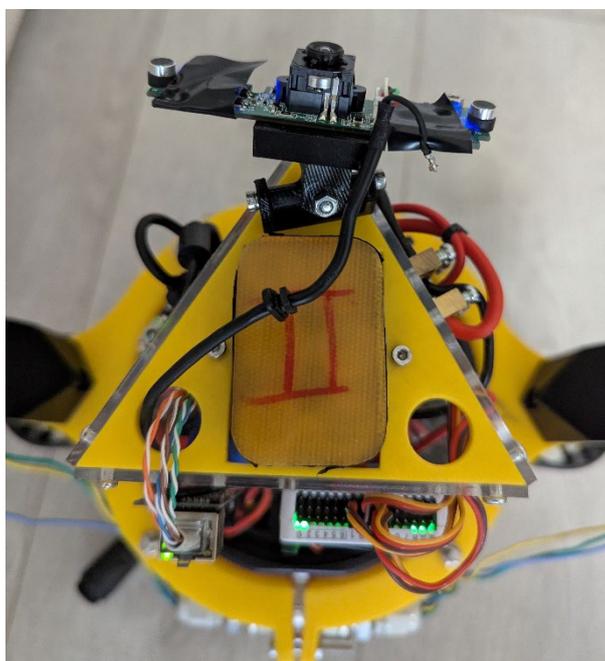


Рис. 4 Модули на шасси, вид сверху



Рис. 5 модули на шасси, вид сбоку

Модули управления

В качестве основного контроллера робота была выбрана Raspberry Pi 3B (рис.8), а в качестве контроллера низкого уровня – Pixhawk 2.4.8 (рис.9). Данные контроллеры достаточно широко используются в подводной робототехнике. В частности, именно они используются в открытом (Open source) проекте ArduSub, о котором будет сказано далее.



Рис. 8 Raspberry Pi 3B



Рис. 9 Pixhawk 2.4.8

В качестве видео камеры используется изделие Logitech HD Pro C920. Камера имеет небольшие размеры, разрешение - 1920x1080 пикселей, 30 кадров в секунду, подключение производится через интерфейс USB. Для уменьшения размеров камеры она используется в бескорпусном варианте. Камера установлена на специальном креплении с возможностью настройки угла зрения (рис.10).

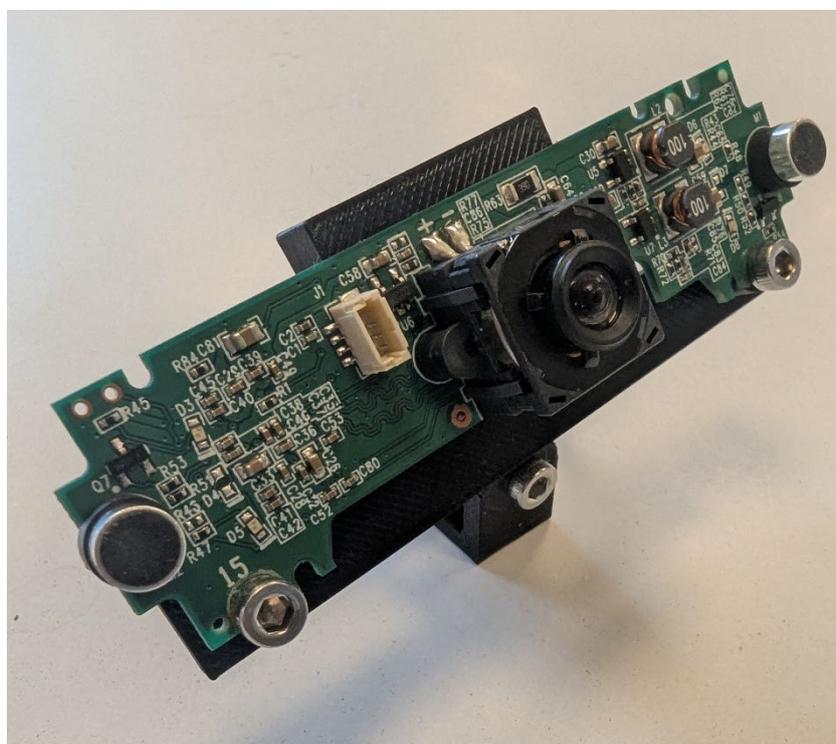


Рис. 10 Веб-камера Logitech HD Pro C920 без корпуса

Датчик глубины.

В качестве датчика глубины был выбран MS5837-30BA (рис. 11), он простой в использовании и недорогой, подходит к выбранной бортовой системе управления. Подключение к микроконтроллеру производится по интерфейсу I2C. Датчик был протестирован с контроллером ESP32, а также с контроллером Pi4hawk 2.4.8, который и используется в проекте.



Рис. 11 Датчик глубины MS5837-30BA

Датчик расстояния.

В качестве датчика расстояния был выбран ультразвуковой датчик, используемый для роботов для чистки бассейнов (рис.12). Подключение к микроконтроллеру производится по протоколу UART. Датчик был протестирован с контроллером ESP32. Его планируется подключить непосредственно к Raspberry PI. Указанный датчик пока не установлен на робота.



Рис. 12 Ультразвуковой датчик расстояния

Бортовое электропитание.

В качестве аккумулятора используется конструкция из 6 литиевых элементов 18650 и платы BMS (battery management system), суммарное напряжение - 24В. От аккумулятора производится питание всей электроники и винтомоторных агрегатов. Для получения напряжений 5В для питания Raspberry Pi и оптического медиаконвертера используются отдельные понижающие конвертеры.



Рис. 13 Аккумулятор

Система управления

В качестве системы управления был выбран открытый проект ArduSub. Проект ArduSub — это полнофункциональное решение с открытым исходным кодом для дистанционно управляемых и автономных подводных аппаратов. ArduSub имеет обширные возможности «из коробки», включая контроль устойчивости с обратной связью, удержание глубины и курса, а также автономную навигацию. Он также использует преимущества других частей платформы ArduPilot, включая симуляторы, инструменты анализа журналов и API-интерфейсы более высокого уровня для управления и контроля транспортных средств.

Несмотря на то, что проект больше не развивается, он отлично подходит под цели проекта. В будущем планируется вносить изменения в исходный код программных модулей для добавления необходимых функций.

Архитектура ArduSub

На рис.14 приведена схема работы программной части, которая сильно помогла в изучении проекта ArduSub. На ней подробно отражены все процессы, используемые протоколы, это позволило организовать совместную одновременную работу канала управления через Wi-Fi и оптоволоконно.

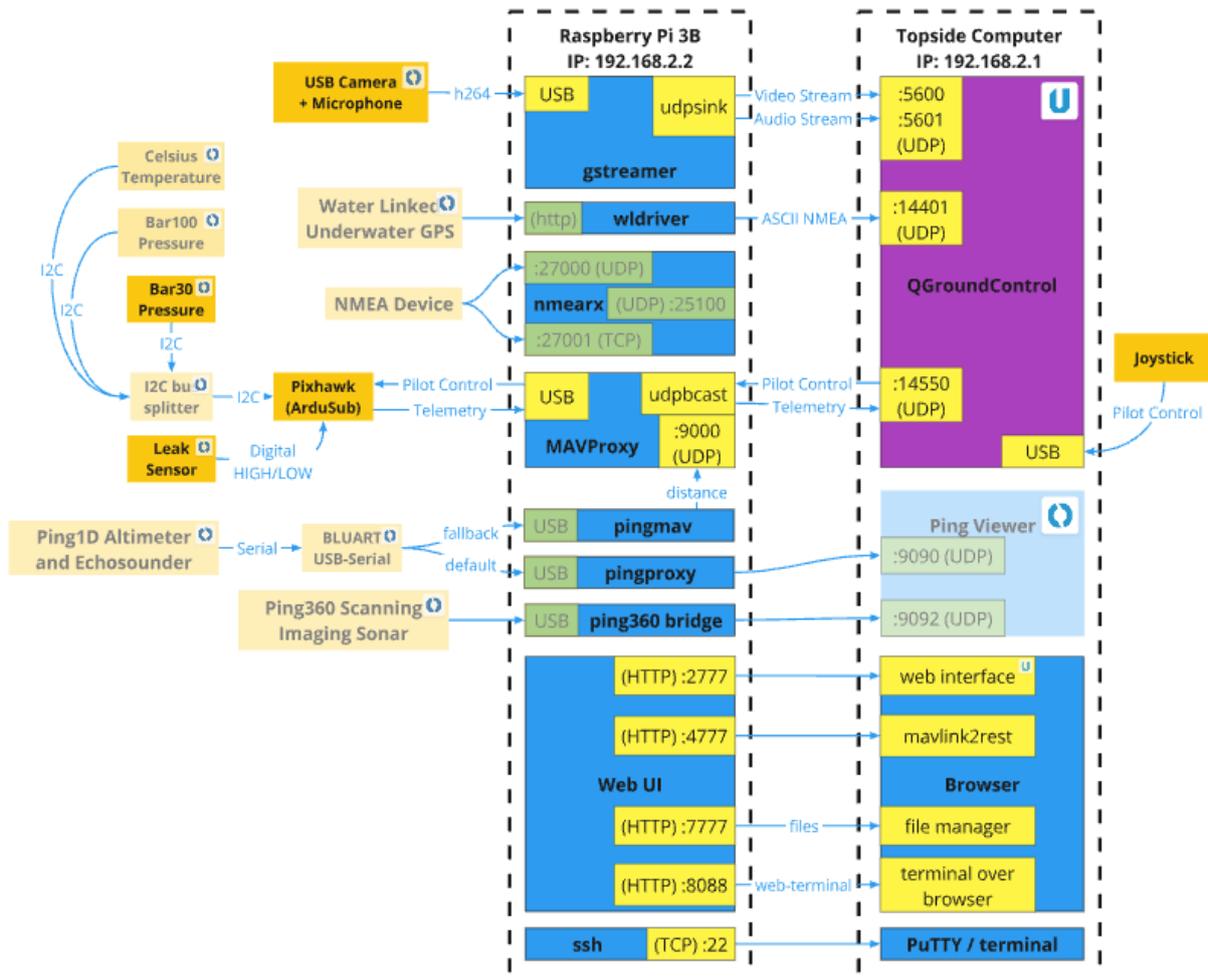


Рис. 14 Схема работы ArduSub

На рис.15 показана упрощенная схема соединений управляющих модулей.

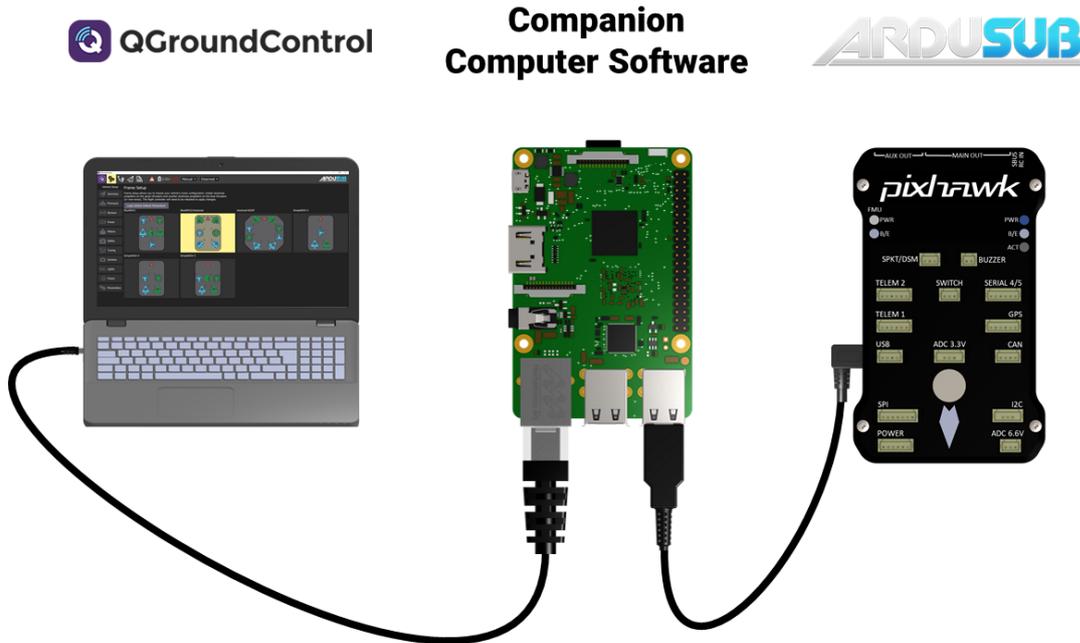


Рис. 15 Упрощенная схема соединений управляющих модулей

В качестве наземной станции управления роботом используется ноутбук с программой QGroundControl. Наземная станция может работать под управлением российской операционной системой Альт Линукс. К наземной станции подключен джойстик, который используется для управления движением робота. В работе установлены Raspberry Pi 3 (далее Rpi3) и контроллер Pixhawk. На Rpi3 установлено программное обеспечение Companion Computer Software. Rpi3 выполняет роль Companion computer – его задачами являются обмен данными между наземной станцией и контроллером Pixhawk, а также передача изображения с камеры на наземную станцию. Задача контроллера Pixhawk – работать с устройствами низкого уровня: датчики, контроллеры двигателей и т.д.

Система дистанционного управления

Для дистанционного управления роботом используется оптоволоконная линия связи или Wi-Fi. Для тестирования робота в надводном состоянии можно использовать Wi-Fi. Для полноценного управления роботом под водой используется оптоволокно, в данный момент используется стандартный патч-корд single-mode длиной 20 метров. Этот тип связи был выбран в силу того, что оптический кабель не подвержен коррозии, обеспечивает высокую скорость передачи данных, легкий и тонкий. Для преобразования и передачи сигнала по оптоволокну используются медиаконвертеры (рис. 16), один расположен в роботе, другой подключен к наземной станции. Используемые медиаконвертеры могут обеспечить передачу данных на скорости 1 гбит/с по оптоволоконному кабелю длиной до 10 км.



Рис. 16 Медаконвертер

Балансировка робота

Балансировка подводного робота является критически важным аспектом его конструкции, так как она напрямую влияет на устойчивость, управляемость и энергоэффективность аппарата.

При балансировке робота необходимо было учитывать два свойства подводного аппарата:

Плаву́честь – это свойство погружённого в жидкость тела оставаться в равновесии, не выходя из жидкости и не погружаясь дальше, то есть плавать.

Остойчивость — способность плавательного средства противостоять внешним силам, вызывающим его крен или дифферент, и возвращаться в состояние равновесия по окончании возмущающего воздействия.

Для того, чтобы робот мог “парить” в воде, он должен иметь нейтральную плаву́честь, то есть находиться в состоянии, при котором вес робота полностью компенсируется выталкивающей силой воды. На практике, однако, целесообразно придать роботу небольшую положительную плаву́честь, при которой робот всплывает с небольшой скоростью. Для удержания на глубине при этом используется слабая тяга вертикально расположенных винтомоторных агрегатов. Достоинством такой настройки плаву́честь является то, что в случае потери канала управления робот самостоятельно поднимется на поверхность.

Для достижения требуемых настроек плаву́честь используется основной балласт (поясной груз для дайвинга весом 2 кг), а также две емкости для свинцовой дроби, которая

используется для точной настройки плавучести. Емкости для дроби закрываются легко снимаемыми крышками, в крышках и емкостях предусмотрены отверстия для выхода воздуха.

Остойчивость робота достигается размещением основного балласта в нижней части корпуса робота, а емкостей для свинцовой дроби – в нижней полусфере (рис.17). Такое размещение балласта эффективно минимизирует крен и дифферент робота при внешних воздействиях. Для достижения нулевого крена и дифферента при отсутствии воздействий балласт расположен на съёмном хомуте, устанавливаемом на колбу робота непосредственно перед погружением, и имеющем возможность поворота и продольного перемещения по оси робота. Для дополнительного выравнивания по крену используется функция автоматической стабилизации крена, реализованная в ПО ArduSub.



Рис. 18 Дробь, используемая в качестве балласта

Тестирование на воде

Ходовые испытания робота были проведены в судомodelьном бассейне Московского Дворца пионеров. В ходе испытаний были протестированы: движение робота по вертикали (погружение/всплытие), горизонтальное движение, система автоматической компенсации

крена. В ходе испытаний робот управлялся как через Wi-Fi, так и через оптоволоконный канал связи.



Рис. 16 Робот в воде

Изготовление отдельных элементов в условиях лаборатории

Крепления для большинства элементов были изготовлены на 3D принтере из материала PET-G. Этот материал был выбран в силу своей доступности, прочности и умеренной устойчивости к ультрафиолетовому излучению, воде и окружающей среде. При изготовлении компонентов на 3D-принтере требовалось использовать заполнение 100%, так как при меньшем заполнении во время погружения печатные конструкции набирали воду, из-за чего плавучесть аппарата становилась отрицательной.

Также для изготовления деталей был использован станок лазерной резки и оргстекло толщиной 6мм.

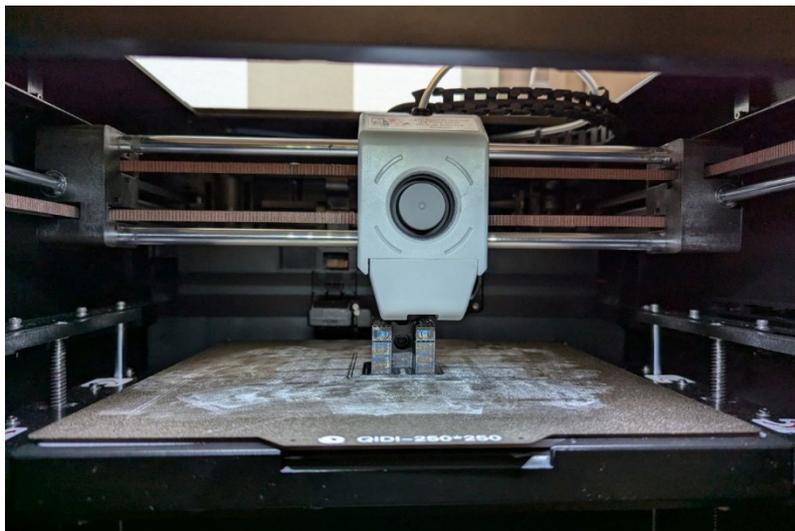


Рис.20 Изготовление элементов на 3D-принтере



Рис. 21 Изготовление элементов на станке лазерной резки

Анализ рынка

Рынок дистанционно управляемых подводных роботов в последние годы демонстрирует активный рост, чему способствуют несколько ключевых факторов, включая развитие технологий, увеличение потребности в подводных исследованиях и улучшение функциональности самих роботов. Одной из главных тенденций является **увеличение спроса на такие роботы в разных отраслях.**

Существуют компании, которые занимаются подводой робототехникой, вот некоторые из них:

- **Blue Robotics** — американская компания, которая предлагает доступные и компактные дистанционно управляемые подводные роботы для любителей подводных исследований и небольших коммерческих проектов. Эти роботы идеально подходят для использования в образовательных целях, исследовательских и экологических проектах.

Например, модель **BlueROV2** имеет небольшие размеры, но позволяет проводить качественные съемки и мониторинг на глубинах до 100 метров.

- **Saab Seaeye** — шведская компания, являющаяся одним из ведущих производителей дистанционно управляемых подводных роботов для широкого спектра применения, включая энергетический сектор, морские исследования и экологический мониторинг. Она известна своими компактными и мощными моделями, такими как **Seaeye Falcon**, которые могут работать в ограниченных пространствах, а также оснащены современными камерами и сенсорами для визуального контроля и сбора данных на больших глубинах. Эти роботы активно используются для обслуживания подводных установок и мониторинга состояния морских экосистем.

- **Гидроскан** — российская компания занимающиеся подводной робототехникой, предоставляют услуги по фотограмметрии.

- **ArduSub** – проект с открытым исходным кодом, используемый в подводной робототехнике для управления подводными аппаратами

Перспективы развития проекта

В будущем на корпус робота под углом 45° будут установлены два дополнительных винтомоторных агрегата для движения по горизонтали, два существующих винтомоторных агрегата также будут повёрнуты под углом 45° , что позволит роботу перемещаться в любом направлении по горизонтали значительно повысив его маневренность. Эта конструкция отражена на схеме (рис.22)

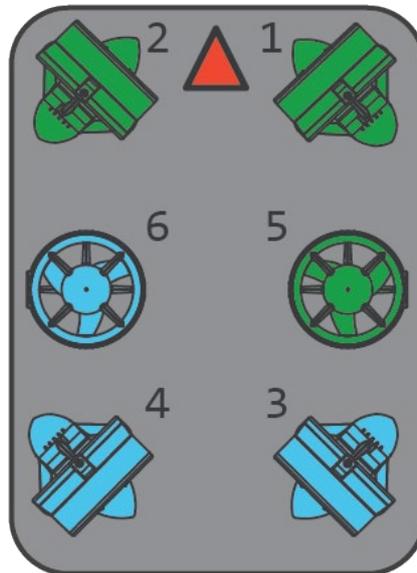


Рис.22 Схема расположение винтомоторных агрегатов

Также, планируется установить датчики глубины и расстояния, описанные выше, доработать программное обеспечение с добавлением функций автоматического удержания расстояния до дна и предотвращения столкновений с препятствиями.

Планируется установка сервомоторов для дистанционного управления поворотом видеокамеры в двух плоскостях, что позволит оператору не терять обследуемый объект из виду при маневрировании.

С целью улучшения качества видеосъемки в условиях недостаточной освещенности планируется установка светодиодных прожекторов подсветки.

Для удобства оператора планируется разместить элементы наземной станции в ударопрочном кейсе.

Заключение

В ходе работы создан бюджетный телеуправляемый необитаемый подводный аппарат. Датчики глубины и расстояния пока не установлены. Робота можно использовать для фото и видеосъёмки различных объектов на глубине до 15 метров. Управление роботом осуществляется с использованием оптоволоконного кабеля длиной 20 метров. Видео с ходовыми испытаниями загружено на Яндекс Диск: <https://disk.yandex.ru/d/3sBJ8XNKbCLCIw>

Список использованной литературы

1. Агеев М. Д., Киселев Л. В., Матвиенко Ю. В. Автономные подводные роботы: системы и технологии / отв. ред. Киселев Л. В. — М.: Наука, 2005. — 398 с. — ISBN 5-02-033526-6.
2. Войтов Д. В. Телеуправляемые необитаемые подводные аппараты. — М.: Моркнига, 2012. — 506 с. — ISBN 978-5-903081-60-8.
3. Ляхов Д. Г. Современные задачи подводной робототехники // Подводные исследования и робототехника. — 2012. — № 1. — С. 15-23.
4. Половко С. А., Шубин П. К., Юдин В. И. Концептуальные вопросы роботизации морской техники // Научно-технические ведомости СПбГПУ : журнал. — 2013. — № 1. — С. 104—110.
5. Сайт компании “Гидроскан”: <https://hydroscan.ru/>
6. Официальная страница команды “Гидронавтика”: <https://vk.com/hydronautics>
7. Сайт проекта “ArduSub”: <https://www.ardusub.com/>
8. Сайт компании “Bluerobotics”: <https://bluerobotics.com/>
9. Видео-материалы центра развития робототехники https://marine.robocenter.org/gallery/video_list

ШИФРОВАНИЕ МЕТОДОМ ГАММИРОВАНИЯ

Жук Илья Владиславович

Звездный городок п, МБОУ СОШ имени В.М. Комарова с углубленным изучением английского языка, 10 класс

Секция: Информационная безопасность и цифровая криминалистика

Научный руководитель: Троицкий Игорь Иванович, к.т.н., доцент каф. ИУ8

МГТУ им. Н.Э. Баумана

Аннотация

В современном обществе информация — это один из самых ценных ресурсов, который требует защиты от несанкционированного проникновения лиц, не имеющих к ней доступа. Поэтому предметом исследования проекта определены следующие аспекты криптографии: шифрование методом гаммирования, относящееся к потоковому симметричному методу шифрования, а также генераторы случайных чисел.

Цель проекта: разработка и реализация универсальной модели гарантированно стойкого шифрования методом гаммирования на языке программирования Python для различных операционных систем (ОС), обеспечивающей защиту конфиденциальной текстовой информации при отправке сообщений по незащищённым каналам.

В ходе работы использовались следующие основные методы исследования и приёмы: для анализа теоретических основ шифрования методом гаммирования и разработки критериев криптостойкости использован анализ источниковой базы. Для оценки генераторов псевдослучайных чисел (ГПСЧ) на предмет соответствия криптостойкости - методы сравнительно-сопоставительного и табличного анализа. Для разработки программного обеспечения ПО - встроенные модули языка программирования Python.

На основе анализа основ шифрования методом гаммирования и функционирования генераторов случайных чисел (ГСЧ) разработана методика, на основании которой разработан и реализован прототип модели гарантированно стойкого шифрования. В ходе исследований был проведён анализ системных ГПСЧ различных ОС, разработана методика по передаче ключа для симметричного шифрования методом гаммирования, которая успешно реализована в модели гарантированно стойкого шифрования. По результатам исследований разработаны предложения по созданию универсального криптостойкого системного ГПСЧ для существующих и перспективных ОС.

ВВЕДЕНИЕ

Актуальность шифрования информации определяется тем, что в настоящее время информация как одна из самых основных составляющих безопасности общества, требует защиты от несанкционированного проникновения лиц, не имеющих к ней доступа.

Причины актуальности шифрования заключаются в следующем:

- в части защиты **конфиденциальных** данных шифрование скрывает информацию от неавторизованных пользователей при передаче или при её хранении;

- в части обеспечения **целостности** данных шифрование предотвращает изменение информации при передаче или хранении;

- в части **аутентификации** источника информации шифрование используется для проверки подлинности источника данных и предотвращения отказа отправителя информации от того факта, что данные были отправлены именно им.

Способов шифрования информации существует огромное множество. Одним из самых распространённых является шифрование методом гаммирования. Однако далеко не всегда при применении данного способа используются надёжные генераторы случайных чисел, что в свою очередь часто приводит к снижению стойкости шифрованной информации и её раскрытия в дальнейшем. Именно поэтому **целью проекта** является разработка и реализация универсальной модели гарантированно стойкого шифрования методом гаммирования на языке программирования Python для различных операционных систем, где наибольшее внимание уделяется разработке и применению универсального и надёжного генератора случайных чисел.

Для достижения указанной цели в работе поставлены следующие **задачи**:

1 Анализ шифрования методом гаммирования и функционирования генераторов случайных чисел.

2 Разработка методики шифрования/расшифрования методом гаммирования с использованием языка программирования Python.

3 Разработка и реализация прототипа модели гарантированно стойкого шифрования.

4 Исследование генераторов псевдослучайных чисел различных операционных систем.

5 Разработка методики по передаче ключа для симметричного шифрования методом гаммирования.

6 Разработка и реализация модели гарантированно стойкого шифрования методом гаммирования.

7 Разработка предложений и методики по созданию универсального криптостойкого системного ГПСЧ для существующих и перспективных ОС.

Основная часть работы состоит из 5-и разделов. Дополнительные материалы (тексты разработанных методик, листинги программ) представлены в приложениях к работе.

В ходе работы над проектом были применены следующие основные методы исследований и приёмы решения задач:

1 Для исследования теоретических основ шифрования методом гаммирования и составления критериев криптостойкости проведён анализ источниковой базы, состоящей из 26 источников, включая 20 иностранных источников.

2 Для оценки ГПСЧ на предмет соответствия криптостойкости использовались методы сравнительно-сопоставительного и табличного анализа.

3 Для разработки ПО использовались встроенный модуль **Python os** ввиду возможности взаимодействия с системными ГПСЧ и модуль **hashlib** для хэширования ключей при реализации методики по передаче ключа для симметричного шифрования методом гаммирования.

Теоретические основы шифрования методом гаммирования и функционирования генераторов случайных чисел, постановка задач

Шифрование методом гаммирования – это метод поточного симметричного шифрования (шифрование и расшифрование происходит с использованием одного ключа), подразумевающий последовательное сложение символов открытого текста с символами некоторой специальной последовательности, которая называется гаммой [1].

Основные преимущества шифрования информации методом гаммирования заключаются в следующем [1, 2]:

- 1) гарантированная стойкость шифрованного текста;
- 2) большая скорость получения шифрованных текстов, так как операция сложения по модулю 2 очень быстро выполняется на вычислительных устройствах и поэтому наложение гаммы даже на очень большой текст выполняется практически мгновенно.

Однако основным недостатком шифрования методом гаммирования является: невозможность отправки одного шифрованного сообщения нескольким адресатам.

Анализ шифрования информации методом гаммирования

Типичный потоковый шифр шифрует открытый текст по одному байту, хотя потоковый шифр может быть разработан для работы с одним битом или с единицами, превышающими байт [7].

Существуют два способа сложения гаммы с открытым текстом [1]:

1) При первом способе символы открытого текста и гаммы заменяют цифровыми эквивалентами, которые затем складываются по модулю k :

$$R_i = (S_i + G) \bmod (k-1), \quad (1)$$

где R_i , S_i , G – символы соответственно шифрованного, открытого текста, гаммы.

2) При втором способе символы исходного текста и гаммы представляют в виде двоичного кода, затем соответствующие разряды складывают по модулю 2, или методом **XOR**:

$$R_{i2} = S_{i2} \oplus G_2, \quad (2)$$

где R_{i2} , S_{i2} , G_2 – двоичная запись соответственно шифрованного, открытого текста, гаммы (последовательности бит/байт).

Ввиду быстроты выполнения операции сложения по модулю 2 в настоящее время при программно-аппаратной реализации шифрования методом гаммирования применяют второй способ (сложение двоичных записей гаммы и открытого текста по модулю 2) [2]. Пример шифрования методом гаммирования представлен в таблице 1.

Таблица 1 - Шифрование методом гаммирования

Открытый текст	Д	У	Б
	110010	100000	010010
Знаки гаммы	8	1	7
	001000	000001	000111
Шифрованный текст	111010	100001	010101

Расшифрование шифрованного текста реализуют посредством наложения гаммы на шифрованный текст, в результате которого получается открытый текст (S_{i2}):

$$R_{i2} \oplus G_2 = (S_{i2} \oplus G_2) \oplus G_2 = S_{i2} \oplus (G_2 \oplus G_2) = S_{i2} \oplus 0 = S_{i2} \quad (3)$$

Таким образом, качество гаммы в свою очередь будет определять стойкость шифрования методом гаммирования [3].

Поэтому для обеспечения гарантированной стойкости необходимо, чтобы последовательность символов в пределах периода гаммы обладала следующими свойствами [2]: была случайной (должна отсутствовать закономерность в появлении символов гаммы); распределение символов алфавита гаммы должно быть близко к равномерному (равновероятному) распределению; совпадала по размеру или была больше шифруемого открытого текста; применялась только один раз.

Обязательность одноразового использования гаммы доказывается следующим образом:

$$\oplus \begin{cases} X1 \oplus Y = Z1 \\ X2 \oplus Y = Z2 \end{cases} \quad (3)$$

с учётом, что

$$(X1 \oplus Y) \oplus (X2 \oplus Y) = Z1 \oplus Z2, \quad (4)$$

$$X1 \oplus Y \oplus X2 \oplus Y = Z1 \oplus Z2, \quad (5)$$

$$(X1 \oplus X2) \oplus (Y \oplus Y) = Z1 \oplus Z2, \quad (6)$$

$$(X1 \oplus X2) \oplus 0 = Z1 \oplus Z2, \quad (7)$$

$$X1 \oplus X2 = Z1 \oplus Z2. \quad (8)$$

В формулах (3) ÷ (8) приняты следующие обозначения: X1- открытый текст 1, X2 - открытый текст 2, Y - ГАММА, Z1 – шифротекст 1, Z2 - шифротекст 2. Сложенные методом XOR два зашифрованных текста равны сложенным методом XOR открытому тексту 1 и открытому тексту 2.

Анализ функционирования генераторов случайных чисел и постановка задачи

Несмотря на все достоинства шифрования методом гаммирования, одной из ключевых проблем их применения на практике является получение качественных гамм. Для процедур получения гамм можно использовать генераторы истинно случайных чисел (ГИСЧ) или криптостойкие генераторы псевдослучайных чисел (ГПСЧ).

ГПСЧ генерирует псевдослучайную последовательность чисел, которая была вычислена по определённой процедуре, но имеет все свойства случайной последовательности чисел в рамках решаемой задачи. Отличие истинно случайных последовательностей от псевдослучайных в этом случае заключается в невозможности предсказания (расчёта, определения) символов в ней [2].

ГИСЧ генерирует свои последовательности путём оцифровки физических или антропогенных процессов [2]. Среди физических процессов, используемых для получения гамм, можно выделить следующие: шумы разного рода; движение жидкостей в лавовых лампах; движение ленточек в потоке воздуха вентилятора; радиоактивный распад; электромагнитное излучение; лавинный пробой в полупроводниках; нестабильная частота свободно работающего осциллятора.

К антропогенным процессам, используемых в качестве источника случайных последовательностей, можно отнести следующие: время между нажатиями клавиш; движения компьютерной мыши; изменение скорости вращения жесткого диска, вызванного турбулентностью воздуха; время между принятыми пакетами в сети.

Для большинства рассмотренных выше генераторов на базе физических и антропогенных процессов характерны определённые недостатки [2]: сложность решения вопроса передачи ключа по сравнению с ГПСЧ; медленная скорость генерации гамм; антропогенные процессы могут иметь скрытые зависимости – каждый пользователь обладает своим «подчерком» работы с компьютером.

Таким образом, в работе принято, что ГПСЧ генерирует свои последовательности путём применения рекуррентных формул или полноценных алгоритмов [2]. При этом криптостойкий ГПСЧ должен обладать следующими свойствами [3, 4]:

1) невозможностью определения следующих и предыдущих членов последовательности по известным k членам последовательности, которую сгенерировал ГПСЧ;

2) период, в течение которого генератор не повторяет свою последовательность, должен быть как можно больше, чтобы предотвратить возможность атак перебора (минимальный период, которым должен обладать ГПСЧ для криптографических приложений – 2^{256});

3) распределение символов последовательности, которую генерирует ГПСЧ, должно быть близко равномерному (равновероятному).

Также одним из важнейших требований криптостойкого ГПСЧ является эффективность алгоритма, а именно большая скорость его работы и малые затраты памяти.

При шифровании методом гаммирования ГПСЧ также должны быть синхронными или самосинхронизирующимися [2].

Основные преимущества синхронных ГПСЧ, в которых поток ключей генерируется независимо от потока сообщения, заключаются в следующем [2, 5]:

1) отсутствие распространения ошибок, то есть если во время передачи бит шифротекста изменит своё значение, что намного вероятнее его потери, то только испорченный бит расшифруется неправильно, а все предшествующие и последующие биты не изменятся;

2) предохранение от любых вставок и удалений шифротекста, так как из-за этого нарушится синхронизация, что приведёт к неправильному расшифрованию всех последующих элементов и соответственно раскрытию действий злоумышленника.

Основные недостатки синхронных ГПСЧ [2, 3]:

1) если один из ГПСЧ пропустит цикл или если бит шифротекста теряется при передаче, то каждый символ шифротекста будет расшифрован неправильно;

2) отсутствие полной защиты от битовых сбоев, то есть если злоумышленнику известны отдельные биты открытого текста, он может изменить эти биты так, как ему надо;

3) сложность реализации повторной синхронизации по сравнению с самосинхронизирующимися ГПСЧ.

Для использования самосинхронизирующихся ГПСЧ необходимо учитывать то обстоятельство, что для реализации данного режима каждое сообщение начинается случайным заголовком длиной n бит. Этот заголовок шифруется, передаётся, а затем расшифровывается. В качестве входных параметров обоих ГПСЧ передаётся шифротекст этих n бит, которые в свою очередь не являются ключом. После того, как второй ГПСЧ примет шифротекст этих n бит, их расшифровка как правило будет неправильной. Однако после принятия этих n бит оба ГПСЧ будут синхронизированы.

Основные преимущества самосинхронизирующихся ГПСЧ [2, 6]:

1) при потере бита шифротекста повторная синхронизация в отличие от синхронных генераторов намного проще;

2) размешивание статистики открытого текста, так как каждый знак открытого текста влияет на следующий шифротекст и статистические свойства открытого текста распространяются на весь шифротекст. Следовательно, гамма, генерируемая самосинхронизирующимися ГПСЧ, может быть более устойчива к атакам на основе избыточности открытого текста, чем гамма, генерируемая синхронными ГПСЧ.

Однако среди недостатков самосинхронизирующихся ГПСЧ необходимо отметить [3]:

1) распространение ошибки: для каждого бита заголовка, искажённого при передаче, расшифрующий генератор выдаёт n некорректных битов гаммы и как следствие, изменённый бит влияет на внутреннее состояние, то есть каждой ошибке шифротекста будет соответствовать k ошибок открытого текста до тех пор, пока испорченный бит не перестанет влиять на внутреннее состояние генератора;

2) чувствительность к вскрытию повторной передачей (к вставкам и удалениям шифротекста).

Для криптографических приложений инициализирующее значение, которое служит входными данными для ГПСЧ, должно быть защищённым. Поскольку ГПСЧ является детерминированным алгоритмом, то злоумышленник может вычислить исходное значение и тогда выходные данные также могут быть определены. Следовательно, исходное значение должно быть непредсказуемым. Фактически, само начальное значение должно быть случайным или псевдослучайным числом. Как правило, начальное значение может генерироваться с помощью ГИСЧ, как это представлено на рисунке 1 [7]. При этом минимальным требованием по длине инициализирующего значения является 128 бит.



Рисунок 1 - Генерация начального значения для ГПСЧ

Важной проблемой при шифровании методом гаммирования является вопрос передачи ключа. Для этого должны использоваться асимметричные системы шифрования (с открытыми ключами) или физические способы передачи ключа [2, 6].

Вывод: исходя из результатов анализа теоретических основ шифрования методом гаммирования были определены основные требования (осуществлена **постановка задачи**) к последовательности, которую генерируют ГИСЧ или ГПСЧ, а также требования к ГПСЧ, который должен быть криптостойким и способен воспроизводить одинаковые последовательности у отправителя и получателя.

Разработка и реализация прототипа модели гарантированно стойкого шифрования методом гаммирования на языке программирования Python

Для создания прототипа модели гарантированно стойкого шифрования, а также генератора случайных чисел (ГСЧ), необходимого для работы программы, выполняющей функцию генерирования гаммы, был выбран **модуль random** языка программирования Python. Кроме того, для решения поставленной задачи были использованы встроенные функции Python, такие как: **ord()** - для представления символов в виде целых чисел; **bin()** - для перевода целых чисел в двоичный вид; **chr()** - для представления целых чисел в виде символов; **int()** - для перевода чисел из двоичной системы счисления в десятичную.

Как было отмечено выше, в современных технических системах для большей стойкости текста гамма генерируется с помощью ГСЧ. Для эффективности наложения гаммы на открытый текст осуществляется сложение символов открытого текста в двоичном виде с гаммой, которая является последовательностью бит.

С целью обеспечения гарантированно стойкого шифрования текста в создаваемом прототипе учитывалось, что гамма должна обладать указанными выше свойствами.

Для реализации прототипа модели гарантированно стойкого шифрования методом гаммирования на языке программирования Python разработана соответствующая обобщённая схема, представленная на рисунке 2.

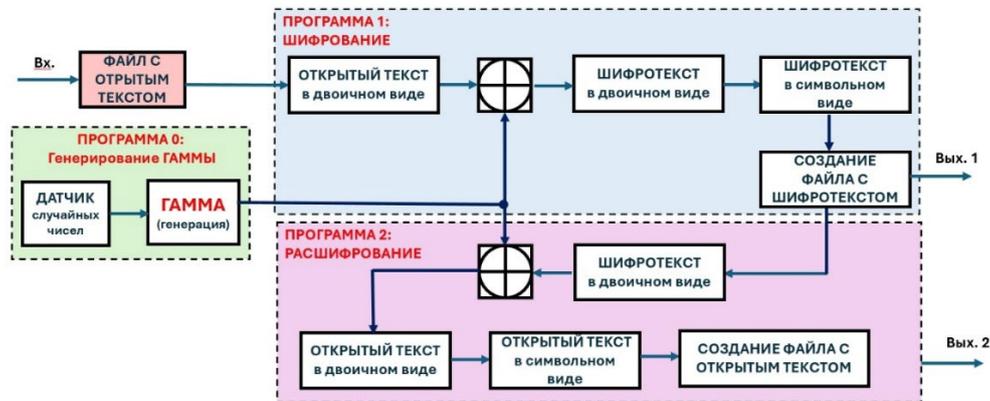


Рисунок 2 - Обобщённая схема шифрования/расшифрования методом гаммирования

В соответствии с представленной схемой была разработана методика шифрования/расшифрования методом гаммирования с использованием языка программирования Python, которая представлена в приложении А. Листинги ПО, реализуемое в соответствии с методикой представлены в приложении Б.

Несмотря на то, что в прототипе модели гарантированного стойкого шифрования была успешно реализована сущность шифрования методом гаммирования (успешно было произведено наложение случайной гаммы на открытый текст и его расшифровка), ГСЧ, основанный на базовых функциях модуля **random** языка программирования Python и который использовался при генерации гаммы, являлся не криптостойким.

Объекты класса **random.Random()** (базовые функции модуля **random**) в качестве ГСЧ используют алгоритм Mersenne Twister (Вихрь Мерсенна), который генерирует числа с плавающей точкой 53-битной точности и имеет период $2^{19937}-1$. Однако будучи полностью детерминированным, он подходит не для всех целей и совершенно не подходит для криптографических [8].

Вывод: Разработанный прототип модели гарантированно стойкого шифрования методом гаммирования на базе объектов класса **random.Random()** на языке программирования Python **не обеспечивает** гарантированно стойкого шифрования и не решает вопрос передачи ключа в условиях, когда абоненты находятся на больших расстояниях друг от друга. Поэтому были проведены дальнейшие исследования по разрешению указанных проблем.

Исследование криптостойких системных ГПСЧ в различных ОС

Для решения указанных выше проблем были проведены исследования других вариантов (проектных решений и способов) по реализации криптостойкого ГПСЧ с помощью возможностей языка программирования Python на различных ОС. При этом **было принято решение**: рассмотреть и исследовать возможность использования в модели гарантированно стойкого шифрования **функции `os.urandom()`** языка программирования Python, которая использует системные ГПСЧ для генерации случайных бит.

Анализ реализации **функции `os.urandom()`** на семействе Unix-подобных систем на базе ядра Linux

Основная особенность Linux заключается в том, что если доступен системный вызов **`getrandom()`**, то он блокируется до тех пор, пока не будет инициализирован системный пул энтропии (ядро собирает 256 бит энтропии) для повышения безопасности [7].

`Getrandom()` в режиме блокировки основан на работе специального символического файла **`/dev/urandom`** с запретом генерировать псевдослучайные биты до тех пор, пока не наберётся достаточно начальной энтропии, и никогда не блокирует после этого момента [9]. Обобщённая схема Linux-ГСЧ (Linux-RNG) представлена на рисунке 3, в соответствии с которой работу Linux-ГСЧ можно охарактеризовать следующим образом [9, 10, 25]:

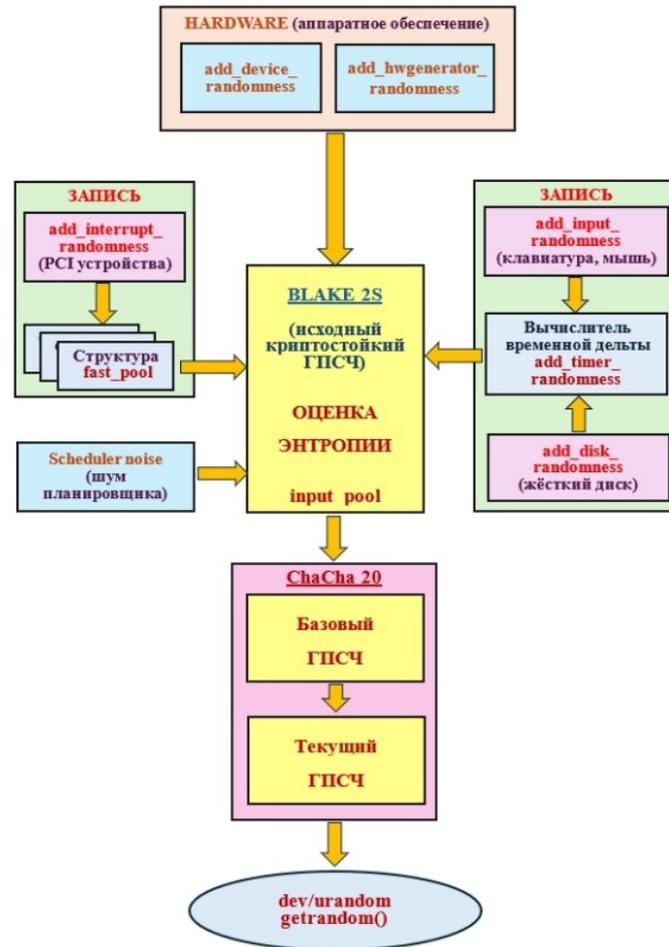


Рисунок 3 - Обобщённая схема Linux-ГСЧ

1. После возникновения аппаратного события, такого как прерывание, время события и его значение хэшируются в пул энтропии, который называется пулом входных данных для Blake2s.

2. Linux-ГСЧ (Linux-PRNG) присваивает этому событию оценку энтропии.

3. При генерации данных из входного пула выполняется заключительная операция хэширования Blake2s.

Это означает, что все данные с момента инициализации пула математически объединяются и затем хэшируются. Полученные результирующие данные обрабатываются аналогично механизму извлечения и расширения с использованием Blake2s.

На первом этапе (этапе извлечения) новый ключ генерируется из объединённых выходных данных пула входных данных и выходных данных в 256 бит из дополнительного ГСЧ на базе центрального процессора. Полученный дайджест сообщения (полученный результат хэширования с помощью Blake2s) используется для повторной инициализации хэш-контекста Blake2s с целью обработки последующих данных из источников шума.

Результат первого этапа используется для выполнения фазы расширения для генерации случайных битов размером 32 байта, которые используются для заполнения ключа базового ChaCha20 ГПСЧ.

4) ChaCha20 ГПСЧ работает путём многократного вызова операции блока текущего ChaCha20 ГПСЧ до тех пор, пока не будет сгенерировано запрошенное количество байт. Следовательно, функция вывода этого ГПСЧ основана на операции блока текущего ChaCha20.

Применение операции блока ChaCha20 изменяет состояние текущего ГПСЧ, то есть значение счётчика состояния увеличивается на единицу после каждой операции блока ChaCha20. Чтобы обеспечить устойчивость к обратному отслеживанию, применяют метод быстрого стирания клавиш, то есть информация о предыдущих состояниях текущего ChaCha20 стирается.

Для проверки на соответствие Linux-ГПСЧ требованиям криптостойкого ГПСЧ необходимо было проанализировать на предмет криптостойкости его выходной ГПСЧ, который генерирует сами случайные биты. Полученные результаты исследований представлены в таблице 2.

При составлении таблицы необходимо было узнать период алгоритма ChaCha20 с использованием счётчика, где под периодом подразумевают максимальное количество выходных значений до повторения последовательности. При этом каждый блок ChaCha20 генерирует 512 случайных бит [9].

Таблица 2 - Оценка криптостойкости алгоритма генерации случайных чисел Linux-ГПСЧ

ChaCha20 с использованием счётчика	Минимальные критерии криптостойкости ГПСЧ	Соответствие критериям криптостойкости
1 ГПСЧ ChaCha20 соответствует требованиям DRG.3.3 [9]: «ГСЧ обеспечивает обратную секретность, даже если известно текущее внутреннее состояние». 2 ГПСЧ ChaCha20 соответствует требованиям DRG.3.2 [9]: «ГСЧ обеспечивает секретность вправо, даже если известно текущее внутреннее состояние».	Невозможность определения следующих и предыдущих членов последовательности по известным k , сгенерированными ГПСЧ [4].	Да
Длина ключа – 256 бит [9].	Минимальная длина ключа – 128 бит [7].	Да
Период – 2^{544} .	Минимальный период для ГПСЧ – 2^{256} [3].	Да
Блоки ChaCha20 являются близким приближением к равномерному	Распределение символов последовательности для ГПСЧ, должно быть	Да

распределению, поддерживаемому лавинным эффектом [9].	близко равномерному (равновероятному).	
---	--	--

Счётчик в структуре блоков ChaCha20 в Linux-ГПСЧ представляет из себя 32-битное значение [10]. При использовании одинакового ключа и постоянной инкрементации счётчика может быть сгенерировано всего $2^{512} \cdot 2^{32}$ значений. Тогда период текущего ChaCha20 будет равен 2^{544} .

Для достижения криптостойкости при поточном шифровании также необходимо, чтобы ключ ГПСЧ был либо истинно случайным или псевдослучайным [7].

Ключ для текущего ChaCha20 определяется младшими выходными битами базового ChaCha20. При этом ГПСЧ, который генерирует ключ для базового ChaCha20, должен быть также криптостойким соответственно. За генерацию ключа для базового ChaCha20 в Linux-ГПСЧ отвечает хэш-функция Blake2S. И хотя её нельзя назвать ГПСЧ, хэш-функция всё же имеет все свойства ГПСЧ. Роль хэш-функции огромна, так как она обеспечивает непредсказуемость поступивших данных в пул энтропии после их обработки [11, 13].

Результаты оценки криптостойкости алгоритма Blake2s при формировании ключа для ChaCha20 в структуре Linux-ГПСЧ представлены в таблице 3.

Таблица 3 - Оценка криптостойкости алгоритма Blake2s

Blake2s	Минимальные критерии криптостойкости ГПСЧ	Соответствие критериям криптостойкости
Безопасность Blake2 эквивалентна безопасности HMAC (при использовании секретного ключа), то есть хэш-функция проходит тест на следующий бит [12]. А функция, которая проходит тест на следующий бит, обладает свойством справа [4].	Невозможность определения следующих и предыдущих членов последовательности по известным k членам последовательности, которую сгенерировал ГПСЧ [4].	Да
Длина ключа – 256 бит [9].	Минимальная длина ключа – 128 бит [8].	Да
Один блок Blake2s выводит 256-битный хэш, значит его период – 2^{256} .	Минимальный период для ГПСЧ – 2^{256} [3].	Да
Хэш-функция Blake2 – пример современной хэш-функции с превосходными статистическими свойствами, что подразумевает равномерность распределения выходных данных [11].	Распределение символов последовательности для ГПСЧ, должно быть близко равномерному (равновероятному).	Да

Таким образом, хэш-функция Blake2s, которая генерирует ключ для базового ГПСЧ на основе ChaCha20 также является криптографически стойкой. Использование хэш-функции при запуске системы без ключа компенсируется объёмом и непредсказуемостью большинства входных данных, которые также проходят через алгоритм ChaCha20.

Необходимо также отметить, что каждые 60 секунд хэш-функция повторно генерирует ключ для базового ChaCha20 из новой энтропии, что обеспечивает дополнительный уровень защиты [9]. При этом в течение следующих 60 секунд вторичные блоки ChaCha20 берут ключ как старшие 256 бит вывода первого вторичного блока в прошлый период, что минимизирует риск использования данных входного пула с неприемлемым количеством энтропии [9].

Исходя из проведённого выше анализа на криптостойкость Linux-ГПСЧ можно сделать **вывод**, что реализация функции **os.urandom()** на семействе Unix-подобных систем на базе ядра Linux **полностью подходит для модели гарантированно стойкого шифрования методом гаммирования**.

Также необходимо отметить, что хэширование с помощью Blake2s с использованием ключа даёт баланс между криптостойкостью и эффективностью. Данное утверждение касается и ChaCha20 [11].

Анализ реализации функции **os.urandom()** на Unix-подобных системах OpenBSD, MacOS и Free BSD

В OpenBSD чтение из устройства **dev/urandom** имеет примерно такую же схему, как и в Linux, но с небольшими отличиями: хэширование пула не происходит. Однако гораздо большее внимание уделяется сбору энтропии из аппаратных ГПСЧ, благодаря чему в пуле постоянно находится достаточно энтропии после его инициализации [14]. Таким образом чтение из **dev/urandom** при реализации функции **os.urandom** на OpenBSD подходит для реализации криптографически стойкого ГПСЧ.

Чтение случайных чисел из **/urandom** на MacOS зависит от устройства: если есть возможность использовать Secure Enclave, то случайные числа будут генерироваться с помощью шифрования методом AES-256 данных аппаратного ГСЧ, который использует физические источники шума. В противном случае используется криптостойкий ГПСЧ, детали работы которого засекречены [15]. Однако известно, что этот ГПСЧ соответствует стандартам NIST SP 800-90A [16]. Из чего можно сделать вывод, что чтение из **dev/urandom** при реализации функция **os.urandom** на MacOS подходит для реализации криптографически стойкого ГПСЧ.

В FreeBSD чтение из устройства **dev/urandom** использует алгоритм **Fortuna**, имеющий со схемой Linux следующие различия [17, 26]:

- вместо одного входного пула используется 32 подпула для сбора энтропии, что защищает от атак, связанных с компрометацией одного источника;
- преобразование данных в пуле происходит с помощью блочного AES шифрования;
- генерация выходных данных происходит с помощью блочного шифрования AES со счетчиком, при этом ключ для шифрования обновляется каждые сгенерированный Мбайт данных.

Можно сделать вывод, что реализация функции **os.urandom()** на других Unix-подобных системах полностью подходит для модели гарантированно стойкого шифрования методом гаммирования.

Анализ реализации функции **os.urandom()** на ОС Windows

В настоящее время корпорация Microsoft, которая создала ОС Windows, не раскрывает подробностей алгоритма криптостойкого ГПСЧ (CSPRNG). Известно, что ГПСЧ Windows удовлетворяет NIST SP800-90 стандартам [18]. Исходя из чего можно сделать вывод (предположить), что реализация функции **os.urandom** на ОС Windows подходит для реализации криптографически стойкого ГПСЧ в модели гарантированно стойкого шифрования методом гаммирования.

Таким образом, генерация случайных чисел с помощью функции языка программирования Python **os.urandom()** на всех рассмотренных выше ОС является криптостойкой.

Вывод: функция **os.urandom()** удовлетворяет критериям (условиям) для реализации ГПСЧ в универсальной модели гарантированно стойкого шифрования на языке программирования Python для различных ОС.

Разработка и реализация модели гарантированно стойкого шифрования методом гаммирования

В процессе исследований, представленных выше, был затронут вопрос о передаче ключа при шифровании методом гаммирования. Как было указано, для передачи ключа необходимо использовать асимметричные системы шифрования или передачу ключа с помощью доверенного третьего лица.

В современных условиях, когда вычислительные ресурсы компьютеров очень велики, целесообразно использовать способ шифрования RSA-4096 с обменом отпечатками ключей и аутентификацией [19, 20].

Допустимая пара ключей RSA в своей базовой форме должна состоять из **открытого ключа RSA (n, e)** и **закрытого ключа RSA (n, d)**, где:

1. **n** - общедоступный модуль, который является произведением двух различных нечётных положительных простых чисел, **p** и **q**, которые хранятся в секрете. Длина **n** при RSA-4096 равна 4096 бит.

Для проверки множителей на простоту должно проводиться специальное тестирование. Обязательной частью тестирования является тест Миллера-Рабина. Рекомендуемое количество раундов (итераций) в тесте для значений **p** и **q**, где каждое имеет длину 2048 бит каждое, – 4 раунда. При 4-х раундах вероятность ошибки в тесте Миллера-Рабина равна 2^{-144} . Для проведения теста необходимо использовать криптостойкий ГПСЧ.

2. Значение **e** должно быть нечётным целым числом, которое выбирается до генерации **p** и **q** таким образом, чтобы:

$$65,537 \leq e < 2^{256} \quad (9)$$

Значение **e** должно быть взаимно простым как для **p – 1**, так и для **q – 1**:

$$\text{НОД}(e, \text{НОК}(p-1, q-1)) = 1. \quad (10)$$

3. Множители **p** и **q** должны быть длиной 2048 бит каждый при RSA-4096 шифровании. При этом

$$|p - q| > 2^{((n\text{Bits} / 2) - 100)}. \quad (11)$$

Так как **nBits** = 4096, то множители **p** и **q** должны удовлетворять следующему неравенству:

$$|p - q| > 2^{1948}. \quad (12)$$

При этом числа **p** и **q** также должны удовлетворять следующим условиям:

$$\begin{aligned} 2^{((n\text{Bits} - 1)/2)} < p < 2^{(n\text{Bits}/2)}, \quad 2^{((4096 - 1)/2)} < p < 2^{(4096/2)}, \\ 2^{((n\text{Bits} - 1)/2)} < q < 2^{(n\text{Bits}/2)}, \quad 2^{((4096 - 1)/2)} < q < 2^{(4096/2)}, \quad (13) \\ 2^{((4096 - 1)/2)} < p < 2^{(4096/2)}, \quad 2^{((4096 - 1)/2)} < q < 2^{(4096/2)}. \end{aligned}$$

4. Простые числа **p** и **q**, а также значение **d** должны быть выбраны таким образом, чтобы:

$$2^{(n\text{Bits}/2)} < d, \quad (14)$$

$$2^{2048} < (e^{(-1)}) \bmod (\text{НОК}(p-1, q-1)). \quad (15)$$

Аутентификация и сохранение целостности информации при асимметричном шифровании RSA-4096 имеет огромное значение, так как атака «Человек посередине», связанная с постоянной подменой публичного ключа может долгое время оставаться незамеченной. Для аутентификации и сохранения целостности информации могут применяться следующие методы [3, 7, 19]: передача публичного ключа с помощью доверенного третьего лица или доверенного центра; обмен отпечатками ключей; применение цифровой подписи.

В условиях, когда нет доверенного источника, для сохранения целостности информации и аутентификации применяются обмен отпечатками ключей и цифровая подпись.

С учётом выше сказанного и в соответствии с [7, 19, 20, 21] была разработана методика по передаче ключа для симметричного шифрования методом гаммирования, представленная в приложении В.

Представленная методика была успешно реализована в модели гарантированно стойкого шифрования с помощью разработанного ПО, листинги которого представленном в приложении Г.

Вывод: При разработке модели гарантированно стойкого шифрования методом гаммирования учтено, что несмотря на то, что функция `os.urandom()` использует системный ГСЧ, синхронизация системных ГПСЧ на различных устройствах **невозможна** ввиду постоянного обновления энтропии, уникальной для каждого устройства.

Максимальный размер сообщения, который можно передать при RSA-4096 шифровании, равен 447 байт. Поэтому 256 бит функции `os.urandom()` могут выступать в качестве ключа для реализованного с помощью Python ChaCha20 ГПСЧ, применяемого в представленной модели гарантированно стойкого шифрования.

5 Разработка предложений по созданию универсального криптостойкого системного ГПСЧ для существующих и перспективных ОС

В настоящее время ввиду накладываемых Западом санкций как никогда актуален вопрос совершенствования существующих и разработка перспективных системных ГПСЧ.

С учётом проведённых исследований и полученных в работе научных результатов целесообразно за основу принять структуру Linux-ГСЧ (Linux-RNG). При этом необходимо учитывать следующие характерные особенности современных Linux-ГСЧ:

1. Использование хэш-функций семейства BLAKE (Blake2, Blake2s и Blake3) как самых быстрых и доступных в настоящее время криптостойких хэш-функций [11].

2. Использование ChaCha20 как основного алгоритма для генерации случайных чисел обеспечивает не только генерацию больших объёмов криптографических псевдослучайных данных, но и универсальность ГПСЧ. Алгоритм ChaCha20 является независимым от специального оборудования и показывает значительные превосходства в скорости в сравнении с генерацией случайных бит с помощью AES шифрования на устройствах, где нет AES-оборудования, а также имеет сопоставимую скорость с генерацией случайных бит с помощью AES шифрования на тех устройствах, где есть AES-оборудование [22].

Основные требования сбора и оценки энтропии для предлагаемого системного ГПСЧ:

1. Генератор случайных чисел должен экспортировать служебные функции, которые размещаются в строго определённых местах в коде ядра для получения событий, связанных с оборудованием. Эти события и временная метка, когда они происходят, будут использоваться для перемешивания в нескольких входных пулах, для обеспечения защиты от атак, связанных с компрометацией одного источника.

2. Для различных классов аппаратных событий должны быть определены различные функции сбора энтропии и различные пулы энтропии соответственно. Количественная оценка энтропии может быть выполнена только с использованием эвристического подхода, который связывает оценку энтропии с данными, полученными от источников шума. Тогда системный ГПСЧ (для различных операционных систем) будет считаться консервативным и, следовательно, наверняка будет обладать криптографической стойкостью, определённой с помощью оценки энтропии.

Для разделения сбора и оцифровки шума предложено выделить следующие классы энтропии и входные пулы соответственно (за основы разделения взята структура Linux-ГСЧ):

1. **Пул 1** с энтропией на базе временных меток завершённых операций планировщика до первой инициализации базового ChaCha20.

2. **Пул 2** с энтропией на базе данных драйверов устройств, включающий временные метки сбора информации. (Эти данные не будут инкрементировать счётчик энтропии, однако будут обеспечивать большее различие в генерации случайных последовательностей на разных устройствах, использующих одну и ту же ОС).

3. **Пул 3** с энтропией на базе аппаратных ГПСЧ (при наличии таковых).

4. **Пул 4** с энтропией на базе значений и временных меток нажатия клавиш, перемещения мыши и аналогичных событий.

5. **Пул 5** с энтропией на базе временных меток произошедших событий на диске, типа событий.

6. **Пул 6** с энтропией на базе временных меток аппаратных прерываний процессора 1.

N. **Пул N** с энтропией на базе временных меток аппаратных прерываний процессора N.

Для управления пулами предложено использовать счётчик энтропии (аналогично как в Linux-ГСЧ), суть работы которого заключается в следующем:

1) после оцифровки шума и его оценки с помощью специальных функций ядра, результат этих функций инкрементирует счётчик на k -е количество бит;

2) когда количество бит энтропии достигает 256, происходит инициализация пулов и извлечение энтропии с помощью хэш-функции Blake3 для генерации ключа для основного ГПСЧ. При этом в дальнейшем инициализация пулов будет происходить при

128 битах в счётчике энтропии, так как минимальная длина ключа для достижения безопасности на сегодняшний день – 128 бит [7].

Методика функционирования предлагаемого универсального ГПСЧ с момента запуска операционной системы представлена в приложении Д.

Вывод: Универсальные ГПСЧ для существующих и перспективных ГПСЧ возможно разработать на базе комбинированных технологий Linux, MacOS, Windows, OpenBSD и FreeBSD. При этом целесообразно за основу принять технологии, реализующие ГПСЧ на Linux системах.

ЗАКЛЮЧЕНИЕ

Актуальность гарантированно стойкого шифрования в современных условиях обусловлена тем, что информатизация и автоматизация всех сфер человеческой деятельности приводят к росту угроз несанкционированного доступа к информации. Поэтому необходимо постоянно поддерживать, обновлять и совершенствовать методы её защиты. Исходя из выше сказанного была поставлена следующая **цель**: разработка и реализация универсальной модели гарантированно стойкого шифрования методом гаммирования на языке программирования Python для различных ОС.

С учётом анализа основ шифрования методом гаммирования и функционирования генераторов случайных чисел разработана соответствующая методика шифрования/расшифрования методом гаммирования с использованием языка программирования Python. На основе указанной методики разработан и реализован прототип модели гарантированно стойкого шифрования, программное обеспечение которого включает: 1) прототип ПО, способного генерировать криптостойкую гамму для шифрования текстовой информации; 2) прототип ПО, способного шифровать текст и 3) прототип ПО, способного расшифровывать текст.

В ходе работы был проведён сравнительный анализ системных ГПСЧ различных ОС с соответствующей оценкой на предмет криптостойкости, а также разработана методика по передаче ключа для симметричного шифрования методом гаммирования. Полученные результаты были успешно применены для разработки модели гарантированно стойкого шифрования методом гаммирования.

В реализованной модели гарантированно стойкого шифрования использован вариант универсального криптостойкого ГПСЧ, основанного на функциях языка программирования Python. Модель успешно протестирована в условиях, когда абоненты используют разные устройства, что значительно расширяет её практическое применение по сравнению с прототипом. При этом выбор языка программирования Python позволит в дальнейшем

модифицировать программную часть реализованной модели с минимальными трудозатратами.

Поставленная цель работы была достигнута: универсальная модель гарантированно стойкого шифрования методом гаммирования была успешно разработана и реализована. Указанная модель может быть реализована в различных системах для шифрования текстовой информации при отправке конфиденциальной информации по незащищённым каналам на базе существующих и перспективных ОС. Полученные в работе результаты исследований предполагают проведение дальнейших исследований по развитию и совершенствованию универсальных моделей гарантированно стойкого шифрования на различных языках программирования. Поэтому в работе разработаны соответствующие предложения по созданию универсального криптостойкого системного ГПСЧ для существующих и перспективных ОС

Выражаю благодарность своему научному руководителю – Троицкому Игорю Ивановичу, доценту, кандидату технических наук, доценту кафедры ИУ-8 МГТУ им. Баумана.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мельников В.П. Информационная безопасность и защита информации / В.П. Мельников, С.А. Клейменов, А.М. Петраков ; под. ред. С.А.Клейменова. — М.: Издательский центр «Академия», 2012. — 336 с.
2. Анисимов В.А. Шифры гаммирования. Лекции // Дальневосточный государственный университет путей сообщения - URL: <https://sites.google.com/site/anisimovkhv/learning/kripto/lecture/tema6> (дата обращения 09.03.2025).
3. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си – М.: Диалектика, 2022. – 1040 с.
4. Слеповичев И.И. Генераторы псевдослучайных чисел – Саратов: СГУ, 2017. – 118 с.
5. Зензин О.С., Иванов М.А. Стандарт криптографической защиты - AES. Конечные поля / Под ред. М.А.Иванова – М.: КУДИЦ-ОБРАЗ, 2002. – 176 с.
6. Кучерик А.О. Защита информации. Курс лекций – Владимир: Изд-во ВлГУ, 2017. – 104 с.
7. Stallings W. Cryptography and network security – Pearson Education, publishing as Prentice Hall, 2016. – 754 p.
8. Python: Generate pseudo-random numbers - URL: <https://docs.python.org/3/library/random.html> (обращения 09.03.2025).
9. Stephan M. Documentation and Analysis of the Linux Random Number Generator Version: 5.11 – Federal Office for Information Security, 2024. – 116 p.
10. Linux kernel source tree // Актуальный код 10. Linux-ГСЧ - URL: <https://web.git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/drivers/char/random.c?id=4df86c6ea5c37fe0452638f39a1e4b189da75c54> (дата обращения 09.03.2025).

11. Aumasson Jean-Philippe. Serious cryptography. A practical introduction to modern encryption / Foreword by Mathew D.Green – San Francisco: No Strach Press, 2018. – 434 p.
12. RFC 7693 - The Blake2 Cryptographic Hash and Message Authentication Code (Mac) - November 2015. – 30 p. - URL: <https://2rfc.net/7693> (дата обращения 09.03.2025).
13. Bernstein J. ChaCha, a variant of Salsa20 // Department of Mathematics, Statistics, and Computer Science (M/C 249). The University of Illinois at Chicago - Chicago, IL 60607–7045. – 6 p.
14. Random number generator - URL: <https://man.openbsd.org/arc4random.3> (дата обращения 09.03.2025).
15. Secure Enclave // Безопасность платформы Apple - URL: <https://support.apple.com/guide/security/secure-enclave-sec59b0b31ff/web> (дата обращения 09.03.2025).
16. Barker Elaine, Kelsey John. Recommendation for Random Number Generation Using Deterministic Random Bit Generators / Nist special Publication 800-90A, Revision 1 // National Institute of Standards and Technology – U.S. Department of Commerce, 2015. – 128 p.
17. Random: Avoid magic numbers // Free BSD - URL: <https://github.com/freebsd/freebsd-src/blob/main/sys/dev/random/fortuna.c> (дата обращения 09.03.2025).
18. BCryptGenRandom function (bcrypt.h) - URL: <https://learn.microsoft.com/en-us/windows/win32/api/bcrypt/nf-bcrypt-bcryptgenrandom> (дата обращения 09.03.2025).
19. Barker Elaine, Chen Lily, Roginsky Allen, Vassilev Apostol, Davis Richard, Simon Scott. Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography / NIST Special Publication 800-56B Revision 2 // National Institute of Standards and Technology – U.S. Department of Commerce, 2019. – 121 p.
20. Barker Elaine, Kelsey John. Digital Signature Standard (DSS) / Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8900 FIPS 186-5 // National Institute of Standards and Technology – U.S. Department of Commerce; 2023. – 77 p.
21. RFC 8017 - PKCS #1: RSA Cryptography Specifications Version 2.2 – November, 2016. - URL: <https://datatracker.ietf.org/doc/html/rfc8017> (дата обращения 09.03.2025).
22. RFC 8439 – ChaCha20 and Poly1305 for IETF Protocols - URL: <https://datatracker.ietf.org/doc/rfc8439/> (дата обращения 09.03.2025).
23. Heninger Nadia, Durumeric Zakir, Wustrow Eric, Halderman J. Alex. Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices // 21-st USENIX Security Symposium - URL: <https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/heninger> (дата обращения 09.03.2025).
24. BLAKE3 - URL: <https://github.com/BLAKE3-team/BLAKE3#faq> (дата обращения 09.03.2025).
25. Chapter 16. Security // Free BSD - URL: <https://docs.freebsd.org/en/books/handbook/security/index.html> (дата обращения 09.03.2025).
26. Актуальный код реализации алгоритма ChaCha20 - URL: <https://web.git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/include/crypto/chacha.h?id=4df86c6ea5c37fe0452638f39a1e4b189da75c54> (дата обращения 09.03.2025).

Приложение А. Методика шифрования/расшифрования методом гаммирования с использованием языка программирования Python

1 Представление открытого текста в виде файла в формате **.txt**

2 Реализация функции «ГЕНЕРАЦИЯ ГАММЫ» (Работа программы 0: генерация гаммы)

2.1 Вызывается **функция «generating»** в скрипте, подав ей на вход путь файла формата **.txt**.

2.2 Получив на входе путь текстового файла, **функция «generating»** открывает файл в режиме чтения и считывает в одну строку весь текст, написанный в файле.

2.3 С помощью встроенной **функции «len»** определяется длина открытого текста в файле, которая записывается как переменная **K**, после чего файл закрывается.

2.4 Далее идёт сама **генерация ГАММЫ**, которая представляет из себя случайную последовательность единиц и нулей от генератора случайных чисел.

Для создания датчика случайных чисел на языке программирования Python используется **модуль random**, который предоставляет функции для генерации случайных чисел, букв, случайного выбора элементов последовательности, и **метод randint**, который случайно выбирает целое число из диапазона **[1; 2²⁴]**. Для большой стойкости гаммы осуществляется выбор числа **N** тоже рандомным. Число **N** является случайным числом из диапазона **[K + 5*K; 100*K]**. Обращение к датчику случайных чисел происходит с помощью **цикла for i in range(N)**. Каждое сгенерированное целое число переводится в двоичную систему счисления с помощью встроенной **функции «bin»** и добавляется в **строку gamma**.

2.5 Далее **список gamma** записывается в **программу 1** и **программу 2** для дальнейшего сложения **методом XOR** с открытым и зашифрованным текстом.

3 Реализация функции «ШИФРОВАНИЕ» (Работа программы 1: шифрование)

3.1 Вызывается **функция «coding»** в скрипте, подав ей на вход путь файла формата **.txt** и **гамму**.

3.2 Получив на вход путь текстового файла и гамму, **функция «coding»** открывает его в режиме чтения и считывает в одну строку весь текст, написанный в файле.

3.3 С помощью **цикла for** каждый символ строки перебирается, после чего он переводится в целое число по кодировочной таблице UTF-8 с помощью встроенной **функции «ord»**. Затем это целое число представляется в двоичном виде с помощью встроенной **функции «bin»**.

3.4 Число, представленное в двоичном виде, добавляется в **список lst**. Таким образом в списке получим двоичную запись каждого символа текста.

3.5 Записанная в файл **программы 1** в виде списка гамма, состоящая из блоков записей целых чисел в двоичном виде теперь, проверяется на то, чтобы каждый **n**-ый блок был равен длине двоичной записи **n**-ного символа открытого текста.

Для этого каждый n -ный блок гаммы до числа K , где K – число символов открытого текста, перебирается **циклом** «`for i in range(len(lst))`» и сравнивается с длиной двоичной записи n -ного символа открытого текста. В случае, если n -ный блок гаммы длиннее двоичной записи n -ного символа открытого текста, n -ный символ текста удлиняется с начала на «0» $\cdot N$, где N - разница между длиной n -ного блока гаммы и n -ного символа открытого текста. В случае, если n -ный блок гаммы короче двоичной записи n -ного символа открытого текста, n -ный блок гаммы удлиняется с начала на «0» $\cdot N$, где N - разница между длиной двоичной записи n -ного символа открытого текста и длины n -ного блока гаммы.

3.6 Для корректного сложения **методом XOR** список `lst` удлиняется на строку, состоящей из (начальная двоичная запись текста – длина гаммы) нулей, после чего эта строка разбивается на блоки, длина n -ого из которых будет равна длине $(len(lst)+n)$ -ого блока гаммы.

3.7 Сложение **методом XOR** (сложение по модулю два) гаммы и открытого текста с помощью **цикла** «`for i in range(len(lst))`».

Каждый n -ый блок **списка** `lst` складывается с n -ым блоком гаммы **методом XOR**, после чего полученная двоичная запись n -ого символа зашифрованного текста добавляется в **список** `lst2`, для его дальнейшего перевода в символьный вид.

3.8 Каждый элемент **списка** `lst2`, который является двоичной записью n -ного символа зашифрованного текста, с помощью **цикла** «`for i in range(len(lst2))`» переводится в целое число с помощью встроенной **функции** «`int`». После чего данное целое число представляется в виде символа зашифрованного текста с помощью встроенной **функции** «`chr`» и добавляется в **строку** `str_`.

Вывод: Получившаяся **строка** `str_` является **зашифрованного текста**. Для записи в файл полученного зашифрованного текста в **программе 1** создается новый текстовый файл, в который записывается получившаяся **строка** `str_`.

4 Реализация функции «РАСШИФРОВАНИЕ» (Работа программы 2: расшифрование)

4.1 Для расшифрования файла в скрипте **программы 2** необходимо вызвать **функцию** «`decoding`», подав ей на вход название файла в формате `.txt` с зашифрованным текстом и гамму.

4.2 Получив на вход название файла и гамму, **функция** «`decoding`» открывает файл в режиме чтения и считывает весь зашифрованный текст, написанный в файле, в одну строку.

4.3 С помощью **цикла** «`for`» каждый символ строки переводится в целое число по кодировочной таблице UTF-8 с помощью встроенной **функции** «`ord`», после чего это целое

число представляется в двоичном виде с помощью встроенной **функции «bin»** и добавляется в **список lst**.

4.4 Записанная в файл **программы 2** в виде списка гамма, состоящая из блоков записей целых чисел в двоичном виде теперь проверяется на то, чтобы каждый **n**-ный блок был равен длине двоичной записи **n**-ного символа шифрованного текста.

Для этого каждый **n**-ный блок гаммы, где $n \leq$ длинны **списка lst** перебирается **циклом «for i in range(len(lst))»** и сравнивается с длиной двоичной записи **n**-ного символа шифрованного текста. В случае, если **n**-ный блок гаммы длиннее двоичной записи **n**-ного символа открытого текста, **n**-ный символ шифрованного текста удлиняется с начала на «0»***N**, где **N** - разница между длиной двоичной записи **n**-ного блока гаммы текста и длины **n**-ного символа шифрованного текста. В случае, если **n**-ный блок гаммы короче двоичной записи **n**-ного символа шифрованного текста, **n**-ный блок гаммы удлиняется с начала на «0»***N**, где **N** - разница между длиной двоичной записи **n**-ного символа открытого текста и длины **n**-ного блока гаммы.

4.5 Сложение **методом XOR** (сложение по модулю два) гаммы и шифрованного текста с помощью **цикла «for i in range(len(lst))»**.

Каждый **n**-ый блок **списка lst** складывается с **n**-ым блоком гаммы **методом XOR**, после чего полученная двоичная запись **n**-ого символа расшифрованного текста добавляется в список **lst2**, для его дальнейшего перевода в символьный вид

4.7 Каждый элемент **списка lst2**, который является двоичной записью **n**-ного символа расшифрованного текста, с помощью **цикла «for i in range(len(lst2))»** переводится в целое число с помощью **функции «int»**. После чего это целое число представляется в виде символа по кодировочной таблице UTF-8 расшифрованного текста с помощью встроенной **функции «chr»** и добавляется в **строку str2_**.

Вывод: Получившаяся **строка str2_** является **расшифрованным текстом**. Для записи полученного расшифрованного текста в файл **программа 2** создаёт конечный текстовый файл и записывает в него получившийся расшифрованный текст (**строка str2_**).

Приложение Б. Программная реализация прототипа модели гарантированно стойкого шифрования методом гаммирования на языке программирования Python (листинги программ)

```

Генерация гаммы.py - C:\Users\Evgeny Zhuk\Downloads\Генерация гаммы.py (3.10.7)
def generating(key):
    from random import randint
    f = open(rf'{way}', encoding='utf-8')
    text = f.read()
    K = len(text)
    f.close()
    #далее идет генерация гаммы с помощью генерации целых положительных чисел
    gamma = []
    N = randint(K//5, K*10)
    for i in range(N):
        n = randint(1, 9999)
        #используем значения суррогатных пар вместо запятой
        #на их основе в конкретном файле
        gamma_symbol = bin(n)[2:]
        gamma.append(gamma_symbol)
    #теперь записываем гамму в обе программы
    a = gamma
    with open('Шифрование.py') as f:
        content = f.readlines()
        content[0] = f'gamma = {a}\n'
    with open('Расшифрование.py', 'w') as f:
        for i in range(len(content)):
            f.write(content[i])

    with open('Дашифрование.py') as f:
        content = f.readlines()
        content[0] = f'gamma = {a}\n'
    with open('Расшифрование.py', 'w') as f:
        for i in range(len(content)):
            f.write(content[i])
    generating("C:\\Users\\viliiaz\\Documents\\Шаг в будущее\\Практическая Демонстрация\\wyfile.txt")
    
```

```

Шифрование.py - C:\Users\Evgeny Zhuk\Downloads\Шифрование.py (3.10.7)
gamma = ['1011111100001', '110000100100', '1111100111100', '1110111011100', '100
def coding_file(file_name, gamma):
    f = open(rf'{file_name}', encoding='utf-8')
    text = f.read()
    f.close()
    #далее идет перевод текста в двоичный вид
    lst = []
    for i in range(len(text)):
        n = bin(ord(f'{text[i]}'))[2:]
        #каждый символ текста сначала представляется целым числом(по таблице Uni
        #а затем это целое число переводится в двоичный вид и идет в список
        lst.append(n)
        #print(ord(f'{text[i]}'))
    #сравним длину блоков гаммы с длиной символами открытого текста
    for i in range(len(lst)):
        if len(lst[i]) > len(gamma[i]):
            gamma[i] = '0'*(len(lst[i])-len(gamma[i])) + gamma[i]
        if len(lst[i]) < len(gamma[i]):
            lst[i] = '0'*(len(gamma[i])-len(lst[i])) + lst[i]
    #дополним исходный список символов текста нулями
    for i in range(len(lst), len(gamma)):
        str_ = '0'*len(gamma[i])
        lst.append(str_)
    #теперь получим зашифрованный текст в двоичном виде
    lst2 = [] #!!!
    for i in range(len(lst)):
        str_ = ''
        for j in range(len(lst[i])):
            if lst[i][j] != gamma[i][j]: str_ += '1'
            else: str_ += '0'
        lst2.append(str_)
    
```

Рисунок Б1 – Генерация ГАММЫ

```

#переведем его из двоичного вида в символьный
str_ = ''
for i in range(len(lst2)):
    number = int(lst2[i], 2)
    str_ += chr(number)
#запишем зашифрованный текст в новый файл
#при желании можно вывести зашифрованный текст
print(str_)
#теперь создадим новый текстовый файл и запишем туда зашифрованный текст
#второй файл txt с зашифрованным текстом будет выходом блока "Шифрование"
file_name2 = 'new_file.txt'
new_file = open(file_name2, 'w', encoding='utf-8')
new_file.write(str_)
new_file.close()
coding_file("C:\\Users\\viliiaz\\Documents\\Шаг в будущее\\Практическая Демонстрац
    
```

Рисунок Б2 – Шифрование текста

```

Расшифрование.py - C:\Users\Evgeny Zhuk\Downloads\Расшифрование.py (3.10.7)
gamma = ['1011111100001', '110000100100', '1111100111100', '1110111011100', '100
def decoding(file_name2, gamma):
    f = open(rf'{file_name2}', encoding='utf-8')
    content = f.read()
    f.close()
    lst = []
    #переведем зашифрованный текст из символьного вида в двоичный
    for i in range(len(content)):
        n = bin(ord(f'{content[i]}'))[2:]
        lst.append(n)
    #сравним длину блоков гаммы с блоками двоичной записи зашифрованного текста
    for i in range(len(lst)):
        if len(lst[i]) > len(gamma[i]):
            gamma[i] = '0'*(len(lst[i])-len(gamma[i])) + gamma[i]
        if len(lst[i]) < len(gamma[i]):
            lst[i] = '0'*(len(gamma[i])-len(lst[i])) + lst[i]
    #сложим гамму с зашифрованным текстом
    lst2 = []
    for i in range(len(lst)):
        str_ = ''
        for j in range(len(lst[i])):
            if lst[i][j] != gamma[i][j]: str_ += '1'
            else: str_ += '0'
        lst2.append(str_)
    #переведем получившийся текст в символьный вид из двоичного:
    #он должен совпасть с начальным
    str2 = ''
    for i in range(len(lst2)):
        number = int(lst2[i], 2)
        str2 += chr(number)
    #при желании можно вывести ршифрованный текст
    print(str2_)
    #теперь запишем расшифрованный текст в третий текстовый файл,
    #который является выходом блока "Расшифрование"
    DName = 'finalfile.txt'
    final_file = open(DName, mode='w+', encoding='utf-8')
    final_file.write(str2_)
    final_file.close()
    decoding("new_file.txt", gamma)
    
```

Рисунок Б3 – Расшифрование текста

Приложение В. Методика передачи ключа для симметричного шифрования методом гаммирования

- 1) Генерация пары частных ($n1, d1$), ($n2, d2$) и публичных ключей ($n1, e1$), ($n2, e2$).
- 2) Вычисление хэша публичных пар с помощью одобренных хэш-функций (SHA-256, SHA-512 и др.) [2].
- 3) Передача дайджеста хэш-функций по нескольким каналам.
В случае, если получатели публичного ключа обнаруживают несовпадение хэшей, передача хэшей по каналам прерывается и производится повторно по новым каналам до тех пор, пока хэши во всех каналах не совпадут для публичного ключа ($n1, e1$) и публичного ключа ($n2, e2$) соответственно.
- 4) Получатель ключа для симметричного шифрования пересылает свой публичный ключ ($n1, e1$) отправителю, меняя каналы до тех пор, пока ключ не будет соответствовать хэшу отправителя ключа для симметричного шифрования.
- 5) Отправитель пересылает свой публичный ключ ($n2, e2$) получателю, меняя каналы до тех пор, пока ключ не будет соответствовать хэшу получателя ключа для симметричного шифрования.
- 6) Отправитель шифрует ключ для симметричного шифрования публичным ключом ($n1, e1$).
- 7) Отправитель подписывает своё сообщение с помощью своего частного ключа ($n2, d2$).
- 8) Отправитель отправляет зашифрованное сообщение.
- 9) Получатель проверяет подпись отправителя ключа для симметричного шифрования с помощью публичного ключа ($n2, e2$) и расшифровывает сообщение с помощью своего частного ключа ($n1, d1$).

Приложение Г. Реализация модели гарантированно стойкого шифрования методом гаммирования (листинги программ)

```

RSA-4096-n1_e1_d1.py - C:\Users\Evgeny Zhuk\Downloads\RSA-4096-n1_e1_d1.py (3.10.7)
File Edit Format Run Options Window Help
from random import SystemRandom
from os import urandom
from math import gcd, lcm
sr = SystemRandom()

e = 65537 #можно использовать в RSA-4096 практически всегда

#тест Миллера-Рабина
def precheck(n): #эта функция для оптимизации
    """Проверка на деление по первым простым числам"""
    small_primes = [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,7
        179,181,191,193,197,199,211,223,227,229,233,239,241,251,257,
    for p in small_primes:
        if n % p == 0:
            return True
    return False

def is_prime(w, iterations) -> bool:
    """Реализация теста Миллера-Рабина по FIPS 186-5"""
    # Шаг 1: Находим a и m (w-1 = 2^a * m)
    m = w - 1
    a = (m & -m).bit_length() - 1 # Находим a
    m >>= a

    wlen = w.bit_length() # Шаг 3
    # Шаг 4: Основной цикл итераций
    for i in range(iterations):
        # Шаг 4.1-4.2: Генерация b в c длиной в битах wlen с принятием ведущих и
        while True:
            b = sr.randint(0, 2**(wlen - 1))
            if b > 1 and b < w - 1:
                break

        # Шаг 4.3: Вычисление z = b^m mod w
        z = pow(b, m, w)
        # Шаг 4.4: Первая проверка
        if z == 1 or z == w - 1:
            continue

        # Шаг 4.5: Цикл по j
        composite = True
        for j in range(1, a):
            z = pow(z, 2, w)
            if z == w - 1:
                composite = False
            break

        elif z == 1:
            return False
        if composite: return False
        else: continue

# Шаг 5: Все проверки пройдены
return True

while True:
    p = int.from_bytes(urandom(256), byteorder="big") | (1 << 2047)
    q = int.from_bytes(urandom(256), byteorder="big") | (1 << 2047)
    #вычисляем d и n
    if abs(p-q) <= 2**1948 or p == q:
        continue
    if precheck(p) or precheck(q): continue
    if is_prime(p, 4) == False or is_prime(q, 4) == False:
        continue

    n = p * q
    if n.bit_length() != 4096:
        continue
    #НОК(p-1, q-1)
    t = lcm(p-1, q-1)
    try:
        d = pow(e, -1, t) #если e и t не взаимно просты, то будет ошибка
    except ValueError:
        continue
    if d <= 2**2048:
        continue
    break

private_key = (n, d)
public_key = (n, e)
#полученная пара приватного и публичного ключа записывается в файл того,
#кто хочет получить гамму
with open('Получатель ключа для симметричного шифрования.py') as f:
    content = f.readlines()
    content[0] = f'private_key = "{private_key}"\n'
    content[1] = f'public_key = "{public_key}"\n'
del p, q, t
    
```

Рисунок Г1 - Генерация пары частных и публичных ключей для получателя открытого текста

```

RSA-4096-n2_e2_d2.py - C:\Users\Evgeny Zhuk\Downloads\RSA-4096-n2_e2_d2.py (3.10.7)
File Edit Format Run Options Window Help
from random import SystemRandom
from os import urandom
from math import gcd, lcm
sr = SystemRandom()

e = 65537 #можно использовать в RSA-4096 практически всегда

#тест Миллера-Рабина
def precheck(n): #эта функция для оптимизации
    """Проверка на деление по первым простым числам"""
    small_primes = [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,7
        179,181,191,193,197,199,211,223,227,229,233,239,241,251,257,
    for p in small_primes:
        if n % p == 0:
            return True
    return False

def is_prime(w, iterations) -> bool:
    """Реализация теста Миллера-Рабина по FIPS 186-5"""
    # Шаг 1: Находим a и m (w-1 = 2^a * m)
    m = w - 1
    a = (m & -m).bit_length() - 1 # Находим a
    m >>= a

    wlen = w.bit_length() # Шаг 3
    # Шаг 4: Основной цикл итераций
    for i in range(iterations):
        # Шаг 4.1-4.2: Генерация b в c длиной в битах wlen с принятием ведущих и
        while True:
            b = sr.randint(0, 2**(wlen - 1))
            if b > 1 and b < w - 1:
                break

        # Шаг 4.3: Вычисление z = b^m mod w
        z = pow(b, m, w)
        # Шаг 4.4: Первая проверка
        if z == 1 or z == w - 1:
            continue

        # Шаг 4.5: Цикл по j
        composite = True
        for j in range(1, a):
            z = pow(z, 2, w)
            if z == w - 1:
                composite = False
            break

        elif z == 1:
            return False
        if composite: return False
        else: continue

# Шаг 5: Все проверки пройдены
return True

while True:
    p = int.from_bytes(urandom(256), byteorder="big") | (1 << 2047)
    q = int.from_bytes(urandom(256), byteorder="big") | (1 << 2047)
    #вычисляем d и n
    if abs(p-q) <= 2**1948 or p == q:
        continue
    if precheck(p) or precheck(q): continue
    if is_prime(p, 4) == False or is_prime(q, 4) == False:
        continue

    n = p * q
    if n.bit_length() != 4096:
        continue
    #НОК(p-1, q-1)
    t = lcm(p-1, q-1)
    try:
        d = pow(e, -1, t) #если e и t не взаимно просты, то будет ошибка
    except ValueError:
        continue
    if d <= 2**2048:
        continue
    break

private_key = (n, d)
public_key = (n, e)
#полученная пара приватного и публичного ключа записывается в файл того,
#кто хочет получить гамму
with open('Отправитель ключа для симметричного шифрования.py') as f:
    content = f.readlines()
    content[0] = f'private_key = "{private_key}"\n'
    content[1] = f'public_key = "{public_key}"\n'
del p, q, t
    
```

Рисунок Г2 - Генерация пары частных и публичных ключей для отправителя открытого текста

```

Отправитель_ключа_для_симметричного_шифрования.py - C:\Users\Evgeny Zhuk\Downl...
File Edit Format Run Options Window Help
private_key_2 = (920181859837550803829209739683467278110151289396572721825117019
public_key_2 = (9201818598375508038292097396834672781101512893965727218251170197
#сверку (n2, e2)
"""После получения открытого и закрытого ключа теперь нужно получить его хэш
для обмена отпечатками ключей"""
import hashlib
combined_str = f"{hex(public_key_2[0])}:{hex(public_key_2[1])}"
hash2 = hashlib.sha512(combined_str.encode("utf-8")).hexdigest()
with open("hash_n2_e2[1].txt", "w") as f:
    f.write(hash2)
with open("hash_n2_e2[2].txt", "w") as f:
    f.write(hash2)
with open("hash_n2_e2[3].txt", "w") as f:
    f.write(hash2)

"""получим отпечаток от отправителя ключа
для симметричного шифрования из всех каналов"""
import sys
with open("hash_n1_e1[1].txt") as f:
    hash1_a = f.read()
with open("hash_n1_e1[2].txt") as f:
    hash1_b = f.read()
with open("hash_n1_e1[3].txt") as f:
    hash1_c = f.read()
if hash1_a == hash1_b == hash1_c: print('Обмен отпечатками публичных ключей прой
else:
    print('Необходимо сменить каналы для обмена отпечатками ключей')
    sys.exit()
del hash1_b, hash1_c
"""
обмен отпечатками должен продолжаться до тех пор,
пока отпечатки публичного ключа не совпадут во всех трех каналах
"""
#снизу (n1, e1)
"""теперь необходимо получить открытый ключ отправителя ключа для симметричного
а также отправить ему свой"""
utf8_bytes_n2_e2 = combined_str.encode("utf-8")
with open("public_key_n2_e2.txt", "wb+") as f:
    f.write(utf8_bytes_n2_e2)

with open("public_key_n1_e1.txt", "rb") as f:
    utf8_bytes_n1_e1 = f.read()
# Декодируем в строку
combined_str_2 = utf8_bytes_n1_e1.decode("utf-8")
# Разделяем на n_hex и e_hex
n_hex, e_hex = combined_str_2.split(":")
public_key_1 = (int(n_hex, 16), int(e_hex, 16))
"""проверим ключ на соответствие отпечатку"""
hash1_2 = hashlib.sha512(combined_str_2.encode("utf-8")).hexdigest()
if hash1_2 == hash1_a: print("Публичный ключ (n1, e1) принят")
else:
    print(hash1_2, hash1_a)
    print("Публичный ключ был видоизменен, его необходимо передать повторно")
    sys.exit()
#следующий этап: генерация гаммы, шифрование главного открытого текста,

```

Рисунок Г3 - ПО для отправителя открытого текста

```

Получатель_ключа_для_симметричного_шифрования.py - C:\Users\Evgeny Zhuk\Downlo...
File Edit Format Run Options Window Help
private_key_1 = (703913345007834965018703122629243434228394934899764812965209503
public_key_1 = (7039133450078349650187031226292434342283949348997648129652095038
#сверку (n1, e1)
"""После получения открытого и закрытого ключа теперь нужно получить его хэш
для обмена отпечатками ключей"""
import hashlib
combined_str = f"{hex(public_key_1[0])}:{hex(public_key_1[1])}" #важно
hash1 = hashlib.sha512(combined_str.encode("utf-8")).hexdigest()
with open("hash_n1_e1[1].txt", "w") as f:
    f.write(hash1)
with open("hash_n1_e1[2].txt", "w") as f:
    f.write(hash1)
with open("hash_n1_e1[3].txt", "w") as f:
    f.write(hash1)

"""получим отпечаток от отправителя ключа
для симметричного шифрования из всех каналов"""
import sys
with open("hash_n2_e2[1].txt") as f:
    hash2_a = f.read()
with open("hash_n2_e2[2].txt") as f:
    hash2_b = f.read()
with open("hash_n2_e2[3].txt") as f:
    hash2_c = f.read()
if hash2_a == hash2_b == hash2_c: print('Обмен отпечатками публичных ключей прой
else:
    print('Необходимо сменить каналы для обмена отпечатками ключей')
    sys.exit()
del hash2_b, hash2_c
"""
обмен отпечатками должен продолжаться до тех пор,
пока отпечатки публичного ключа не совпадут во всех трех каналах
"""
#снизу (n2, e2)
"""теперь необходимо получить открытый ключ отправителя ключа для симметричного
а также отправить ему свой"""
utf8_bytes_n1_e1 = combined_str.encode("utf-8")
with open("public_key_n1_e1.txt", "wb+") as f:
    f.write(utf8_bytes_n1_e1)

with open("public_key_n2_e2.txt", "rb") as f:
    utf8_bytes_n2_e2 = f.read()
# Декодируем в строку
combined_str_2 = utf8_bytes_n2_e2.decode("utf-8")
# Разделяем на n_hex и e_hex
n_hex, e_hex = combined_str_2.split(":")
public_key_2 = (int(n_hex, 16), int(e_hex, 16))
"""проверим ключ на соответствие отпечатку"""
hash2_2 = hashlib.sha512(combined_str_2.encode("utf-8")).hexdigest()
if hash2_2 == hash2_a: print("Публичный ключ (n2, e2) принят")
else:
    print("Публичный ключ был видоизменен, его необходимо передать повторно")
    sys.exit()

```

Рисунок Г4 - ПО для получателя открытого текста

Приложение Д. Методика функционирования предлагаемого универсального ГПСЧ

Этап 1: специальные функции ядра оцифровывают шумы.

Этап 2: каждое оцифрованное событие, которое определяет величину энтропии, оценивается. Полученный результат оценки инкрементирует счётчик на k -е количество бит.

Этап 3: после оценивания энтропии, каждое оцифрованное событие добавляется в определённый пул в соответствии со своим классом.

Параллельно шагам 2–3 выполняется генерация ключа для функции хэширования, дайджест сообщения которой будет являться ключом для базового ГПСЧ, который генерирует ключ для текущего ГПСЧ. Так как безопасность ключа для базового ГПСЧ имеет огромное значение, ключ для функции хэширования, формирующей другой ключ, должен при запуске системы формироваться из самых надёжных источников энтропии. Таковыми являются временные метки высокого разрешения аппаратных прерываний или данные аппаратных ГПСЧ [23].

При отсутствии аппаратных ГПСЧ на устройствах предлагается использовать хэшированные временные метки высокого разрешения аппаратных прерываний в качестве ключа для другой хэш-функции, формирующей ключ для базового ГПСЧ. При запуске системы данные класса пулов $6 \div N$ после оценки энтропии будут инкрементировать другой счётчик энтропии (предлагается ввести второй счётчик энтропии для аппаратных прерываний).

За основу оценки аппаратных прерываний предлагается взять способ оценки энтропии от аппаратных прерываний в Linux-ГСЧ: при накоплении 1024 аппаратных прерываний в подпуле или по прошествии секунды с момента последнего добавления в подпул временной метки высокого разрешения данные будут извлекаться из подпула и добавляться в один из пулов $6 \div N$ (в зависимости от того, к какому процессору относятся эти аппаратные прерывания), инкрементируя первый и второй счётчики энтропии на 1 бит. При достижении вторым счётчиком значения в 256 данные из пулов $6 \div N$ математически объединяются в одну строку и равномерно распределяются с помощью быстрой не криптостойкой хэш-функции. Определить подходящую хэш-функцию для сжатия и равномерного распределения энтропии от аппаратных прерываний является одной из задач на будущее. Дайджест этой функции, равный 32 байтам, используется в следующем этапе.

Такой способ генерации ключа для хэш-функции, генерирующей ключ для основного ГПСЧ, целесообразно использовать только лишь при первом запросе на случайные байты, то есть в следующих запросах второй счётчик энтропии прекращает работать.

Этап 4: как только значение главного счётчика с момента запуска системы становится 256 бит, пул инициализируется. После чего в следующих запросах на случайные биты они будут выдаваться при достижении значения счётчика в 128 бит.

Этап 5: полученный 32 байтный дайджест сообщения не криптостойкой хэш-функции используется в качестве ключа для хэширования (извлечения энтропии) с помощью самой быстрой криптостойкой функции на сегодня – Blake3 [24]. Полученный дайджест сообщения криптостойкой хэш-функции используется в качестве ключа для основного ГПСЧ. Также результат этапа 5 записывается как ключ хэш-функции Blake3 в следующем запросе на случайные биты.

Этап 6: основной ГПСЧ на основе блока ChaCha20 генерирует 512 выходных бит. Старшие 256 бит становятся перезаписанным ключом, который был получен в этапе 5. При этом результат прошлого этапа стирается для соответствия стандартам FIPS [20, 24]. Младшие 256 бит вывода основного ГПСЧ становятся ключом во вторичном ГПСЧ.

Этап 7: вторичный (текущий) ГПСЧ, основанный на блоках ChaCha20 с использованием 32-битного счётчика, генерирует поток выходных бит, которые предоставляются пользователю.

**VISIOASSIST: ИНТЕЛЛЕКТУАЛЬНЫЙ ПОМОЩНИК ДЛЯ
СЛАБОВИДЯЩИХ И НЕЗРЯЧИХ**

Иванов Андрей Александрович

Москва г, Инженерная школа № 1581, 11 класс

Секция: Информационные технологии

Научный руководитель: Смирнова Светлана Юрьевна, преподаватель ГБОУ Инженерная школа №1581

Аннотация

В современном мире незрячие и слабовидящие люди сталкиваются с многочисленными трудностями в повседневной жизни. Чтение текстов, ориентация в пространстве и восприятие окружающей среды требуют значительных усилий. Развитие технологий компьютерного зрения позволяет создавать устройства, способные компенсировать эти ограничения.

Цель: разработать компактное устройство на базе Raspberry Pi 5 и создать программное обеспечение для помощи слабовидящим и незрячим людям, способные распознавать текст и описывать изображения.

По результатам работы можно сделать следующий вывод: создано сопровождающее программное обеспечение и устройство, оснащённое камерой, динамиками и двумя тактовыми кнопками:

- Одна кнопка активирует оптическое распознавание текста (OCR). Если сервер недоступен, эта кнопка выполняет функцию первой.
- Вся обработанная информация озвучивается через специальный звуковой модуль.

Устройство успешно выполняет поставленные задачи: чтение текстов и получение описания окружающей среды, что значительно повышает качество жизни слабовидящих и незрячих людей.

ВВЕДЕНИЕ

Современные технологии играют всё более существенную роль в повышении качества жизни людей с ограниченными возможностями. Одной из наиболее востребованных областей является создание устройств и систем, направленных на поддержку и упрощение повседневной жизни людей с нарушениями зрения. Несмотря на значительные успехи в развитии аппаратных и программных средств для помощи слабовидящим и незрячим, всё ещё остаются области для дальнейших усовершенствований – от распознавания текста до контекстного описания окружающей среды.

Цель: разработать устройство и программное обеспечение для свободного доступа к нейросетям, которые обеспечат удобный, легкий в освоении, универсальный и относительно недорогой инструмент для людей с нарушением зрения, который позволит им проще ориентироваться в окружающем мире, получать информацию из текста, а также получать описания объектов или сцен на изображениях. Таким образом, устройство может стать важным шагом к большей доступности и самостоятельности слабовидящих и слепых пользователей.

Актуальность: разрабатываемый проект «VisioAssist» актуален, поскольку он имеет возможность помочь многим людям, например, с временной потерей зрения, объединяет в себе несколько передовых технологий – оптическое распознавание текста (OCR), разработка сложного программного обеспечения для свободного доступа к нейросетям, модель компьютерного зрения для описания изображений – делая их доступными через удобный, тактильно ориентированный интерфейс. С одной стороны, это устройство может работать автономно, обеспечивая возможность чтение текста, с другой – при подключении к сети расширяет функционал, позволяя получать более богатую и контекстную информацию о визуальной среде.

Задачи:

1. Настроить Raspberry Pi 5 и подключить его к двум тактовым кнопкам.
2. Организовать процесс автоматического захвата и обработки изображения по нажатию кнопки.
3. Подключить и настроить OCR на Raspberry Pi 5.
4. Разработать и интегрировать персонально обученную модель описания изображений на локальный сервер.
5. Создать собственный ПО с удобным для незрячих людей интерфейсом и встроить туда функции OCR и модель для описания изображений.
6. Настроить обращение Raspberry Pi 5 к общедоступному серверу с последующим получением описания.

1. Аналоги

В качестве аналогов были взяты самые распространенные устройства и приложения с аналогичными функциями. Характеристики и стоимость были взяты с официальных сайтов. Для созданного устройства характеристики были получены при тестировании. Стоимость устройства указана приблизительно на основе цен на детали из маркетплейсов и от поставщиков с сайтов по размещению объявлений.

	OrCam MyEye Pro 3	Seeing AI	VisioAssist
Стоимость	500000p (с учетом доставки и таможи)		15000p

Возможности	-Распознавание текста -Описание изображений -Распознавание купюр	-Распознавание текста -Описание изображений -Распознавание купюр	- Распознавание текста - Описание изображений
Платформа	Физическое устройство	Android/IOS	Физическое устройство / ПО
Формат доступа	Офлайн	Онлайн	Офлайн/Онлайн

Orcam MyEye Pro 3

Плюсы:

- Удобство использования, за счет крепления на очки и ненужности подключать устройство к интернету.
- Четкие и понятные ответы.

Минусы:

- Высокая стоимость.
- Недоступность в Российской Федерации.

Seeing AI

Плюсы:

- Бесплатный доступ к приложению.
- Доступно на телефонах.

Минусы:

- Не возможность использования без доступа к интернету.
- Не удобный интерфейс для слабовидящих и незрячих.

VisioAssist:

Плюсы:

- Доступность в виде физического устройства с понятным функционалом и в виде бесплатного/общедоступного ПО.
- Дешевизна устройства.
- Высокая автономность.

Минусы:

- Генерация описания изображений осуществляется только при подключенном интернете.

2. Выбор средств для разработки проекта:

2.1 Выбор одноплатного компьютера:

Для основы проекта решено было взять одноплатный компьютер, его преимущества:

- Невысокая стоимость.
- Удобность работы.
- Большое количество обучающего контента.
- Большое количество доступных плат расширений.
- Постоянная поддержка новыми обновлениями.

Рассмотренные к использованию платы:

Стоимость была взята от поставщиков с сайта avito.ru.

Предполагаемый объём, занятый оперативной памяти равен 2 гигабайтам.

	Raspberry Pi 4 Model B 4 GB	Raspberry Pi 5 4GB	Raspberry Pi 5 8GB
Стоимость	7000р	8000р	9000р
Процессор	Quad-core Cortex-A72	Quad-core Cortex-A76	Quad-core Cortex-A76
Объём оперативной памяти	4 гигабайта	4 гигабайта	8 гигабайта
Наличие встроенного цифро-аналогового преобразователя	Есть, но плохого качества.	Нет	Нет

Учитывая все выше предоставленные факторы, самым подходящим под наши задачи устройством, является Raspberry PI 5 GB, из-за:

- Относительно низкой стоимостью по сравнению со старшей моделью, но достаточным объёмом оперативной памяти.
- Более мощного процессора по сравнению с Raspberry Pi 4.

2.2 Выбор языка программирования для платы:

В качестве языка программирования был выбран Python, его основные достоинства:

- Высокая популярность (рис. 1)

Feb 2025	Feb 2024	Change	Programming Language	Ratings	Change
1	1		 Python	23.88%	+8.72%
2	3	▲	 C++	11.37%	+0.84%
3	4	▲	 Java	10.66%	+1.79%
4	2	▼	 C	9.84%	-1.14%
5	5		 C#	4.12%	-3.41%
6	6		 JavaScript	3.78%	+0.61%
7	7		 SQL	2.87%	+1.04%
8	8		 Go	2.26%	+0.53%
9	12	▲	 Delphi/Object Pascal	2.18%	+0.78%
10	9	▼	 Visual Basic	2.04%	+0.52%
11	11		 Fortran	1.75%	+0.35%
12	15	▲	 Scratch	1.54%	+0.36%
13	18	▲	 Rust	1.47%	+0.42%
14	10	▼	 PHP	1.14%	-0.37%

Рисунок 1 – Статистика популярности языков программирования.

Взято с сайта tobe.com.

- Множество библиотек для обучения и работы с нейросетями.
- Простой и понятный синтаксис.
- Высокая гибкость настройки.

2.3 Выбор модуля камеры.

Стоимость модулей была взята от поставщиков с сайта avito.ru.

Рассматриваемые варианты модулей камер:

	Raspberry Pi camera module 1	Raspberry Pi camera module 3 NOIR	Raspberry Pi camera module 3	Raspberry Pi camera module 3 wide
Стоимость	1000р	3400р	3700р	5400р
Автофокус	Нет	Есть	Есть	Есть
Разрешение	5 МП	12МП	12МП	12МП
Поле зрения	54° по горизонтали	75° по горизонтали	75° по горизонтали	120° по горизонтали
Инфракрасный фильтр	Есть	Нет	Есть	Есть
Максимальный формат видео:	1920 × 1080p 30fps	4608 × 2592 30fps	4608 × 2592 30fps	4608 × 2592 30fps

Для разрабатываемого устройства критически важно иметь автофокус и хорошее максимальное разрешение видео. Под эти критерии идеально подходит модуль Raspberry Pi camera module 3 NOIR, а отсутствие инфракрасного фильтра дает возможность камере лучше снимать в темное время суток.

2.4 Выбор алгоритма OCR:

Для использования рассматривалось несколько алгоритмов:

- 1) EasyOCR - библиотека для распознавания текста на изображениях, написанная на Python. Она позволяет легко и быстро извлекать текстовую информацию из изображений:

Плюсы:

- Использование новых нейросетей на базе PyTorch.
- Легкость настройки.

Минусы:

- Требуется большого объема оперативной памяти и высокопроизводительного процессора/видеокарты.

- 2) Tesseract – свободная компьютерная программа для распознавания текстов. Разрабатывалась Hewlett-Packard с середины 1980-х по середину 1990-х:

Плюсы:

- Требуется мало оперативной памяти и маломощного процессора.
- Большое количество расширений.
- Высокая надежность.
- Легкая настройка

Минусы:

- Устаревшие алгоритмы.

Для разрабатываемого устройства больше всего подходит Tesseract, из-за его высокой надежности и малых системных требований.

3. Разработка программного обеспечения для сервера.

3.1 Обучение собственной нейросети для генерации подписей к изображениям.

Основным файлом для обучения является train.py.

После его запуска он делает следующие действия:

- 1) Инициализация датасета.
- 2) Первичная обработка изображений.
- 3) Загрузка предобученного энкодера – “google/vit-base” и декодера – “gpt2”.
- 4) Инициализация оптимизатора – “AdamW”.
- 5) Загрузка обучаемой нейросети (если не первая эпоха)
- 6) Запуск обучения.
- 7) Сохранение модели после каждой эпохи.

В качестве датасета был выбран большой датасет COCOdataset он включает в себя 17 тысяч изображений с описаниями на английском языке.

Нейросеть обучалась 5 эпох, используя cuda ядра видеокарты nvidia rtx 3070 ti, время обучения составило свыше 35 часов (рисунок 2).

```

You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
Use cuda
Training Epoch 1/5: 0% | 0/73970 [00:00<?, ?it/s]E
:\Project\venv\lib\site-packages\torch\nn\modules\activation_functions\gelu.py:277: UserWarning: Torch was not compiled with flash attention. (Triggered internally at C:\actions-run
er\work\pytorch\pytorch\builder\windows\pytorch\aten\src\ATen\native\torch\cuda\sdpa_utils.cpp:555.)
context_layer = torch.nn.functional.scaled_dot_product_attention(
Training Epoch 1/5: 100% | 73970/73970 [7:04:39<00:00, 2.90it/s, loss=0.4017]
Validation Epoch 1/5: 100% | 3127/3127 [05:54<00:00, 8.82it/s, val_loss=0.6330]
Epoch 1/5, Train Loss: 0.4454, Val Loss: 0.4249
Model saved to models\model_epoch_1.pt
Training Epoch 2/5: 100% | 73970/73970 [6:59:23<00:00, 2.94it/s, loss=0.1626]
Validation Epoch 2/5: 100% | 3127/3127 [06:08<00:00, 8.49it/s, val_loss=0.5867]
Epoch 2/5, Train Loss: 0.3967, Val Loss: 0.4189
Model saved to models\model_epoch_2.pt
Training Epoch 3/5: 100% | 73970/73970 [7:05:18<00:00, 2.90it/s, loss=0.4102]
Validation Epoch 3/5: 100% | 3127/3127 [10:00<00:00, 5.21it/s, val_loss=0.6752]
Epoch 3/5, Train Loss: 0.4185, Val Loss: 0.4474
Model saved to models\model_epoch_3.pt
Training Epoch 4/5: 100% | 73970/73970 [7:01:28<00:00, 2.93it/s, loss=0.3676]
Validation Epoch 4/5: 100% | 3127/3127 [09:59<00:00, 5.22it/s, val_loss=0.5927]
Epoch 4/5, Train Loss: 0.3651, Val Loss: 0.4195
Model saved to models\model_epoch_4.pt
Training Epoch 5/5: 100% | 73970/73970 [7:01:57<00:00, 2.92it/s, loss=0.2022]
Validation Epoch 5/5: 100% | 3127/3127 [10:00<00:00, 5.21it/s, val_loss=0.6877]
Epoch 5/5, Train Loss: 0.3598, Val Loss: 0.4369
Model saved to models\model_epoch_5.pt

```

Рисунок 2 – процесс обучения собственной нейронной сети.

3.2 Создание программы для обращения к нейросети.

Основным файлом для перевода является `neural_network.py`.

После его запуска он делает следующие действия:

- 1) Выбор версии нейросети и устройства, на котором она будет запущена.
- 2) Функция “initialize_model” для инициализации модели.
- 3) Функция “process_image” для открытия изображения.
- 4) Функция “generate_caption”, которая по открытому изображению из пункта 3 дает описание.

Данная программа запускается на cuda ядрах видеокарты nvidia rtx 3070 ti, что обеспечивает высокую скорость выполнения запросов.

3.3 Создание программы для перевода ответов.

Основным файлом для перевода является `translate.py`.

После его запуска он делает следующие действия:

- 1) Создание класса “EnglishToRussianTranslator”.
- 2) Функция для инициализации нейросети для перевода “Helsinki-NLP/opus-mt-en-ru”
- 3) Функция для обращения к инициализированной нейронной сети, для получения результатов.

Для перевода полученных ответов от созданной нейросети п. 3.1 будем использовать готовую нейросеть для перевода с английского на русский с сайта huggingface.co.

3.4 Создание программы для остр.

Файл называется `tesseract.py`.

После его запуска он делает следующие действия:

- 1) Инициализация класса `tesseract`.
- 2) Бинаризация изображения для четкости букв.
- 3) Вызов модуля `pytesseract` для получение текста.

4. Разработка программного обеспечения для свободного доступа к созданным алгоритмам.

В качестве ПО для доступа к нейросетям будет выступать web-страница с удобным интерфейсом.

4.1 Backend сайта.

Основным и единственным файлом является main.py.

После его запуска он делает следующие действия:

- 1) Инициализация сервера flask, функций “generate_caption”, ”initialize_model” из файла “neural_network.py”, классов “get_translator”, “TesseractOCRProcessor” из соответствующих файлов “translate.py”, “tesseract.py”.
- 2) Функция “fix_image_orientation” по исправлению ориентации изображения по exif данным.
- 3) Функция “test” для тестирования соединения по методу “GET”.
- 4) Функция “process_image” для генерации результатов по методу “POST”.
 - a. Загрузка изображения и метода из http-запроса.
 - b. Исправление ориентации п.2.
 - c. По выбранному методу вызываются функции либо ocr, если метод равен 1, либо генерация описаний, если метод равен 2.
 - d. Высылает обратно получившийся результат.
- 5) Запуск flask сервера по порту 5000.

Для создания общедоступного сервера:

- 1) Добавление правила переадресации на маршрутизаторе (рисунок 3).
- 2) покупка статического ip адреса у оператора.

Добавить правило

Включено

Имя:	<input type="text" value="backend"/>
Интерфейс:	<input type="text" value="nas1 (IP_HOST)"/>
Частный IP-адрес (хост):	<input type="text" value="Вручную"/>
Внутренний IP:	<input type="text" value="192.168.1.6"/>
Удаленный хост:	<input type="text"/>
Внутренний диапазон портов:	<input type="text" value="5000"/> - <input type="text" value="5000"/>
Внешний диапазон портов:	<input type="text" value="5000"/> - <input type="text" value="5000"/>
Тип протокола:	<input type="text" value="TCP"/>

Добавить/Применить

Удалить

Рисунок 3 – переадресация портов на маршрутизаторе.

Теперь сайт является общедоступным со статическим ip - <http://91.77.167.51:5000>.

4.2 Frontend сайта.

Структура сайта:

- index.html – структура сайта (рисунок 4).
- styles.css – визуальное оформление сайта.
- scripts.js – скрипты для сайта:
 - 1) получение элементов DOM.
 - 2) Обработка функции выбора файла.
 - 3) Обработка перетаскивания файла.
 - 4) Функция http-запроса с методом и с изображением на адрес <http://91.77.167.51:5000>.
 - 5) Функция для озвучивания текста.
 - 6) Функция для переключения вкладок.

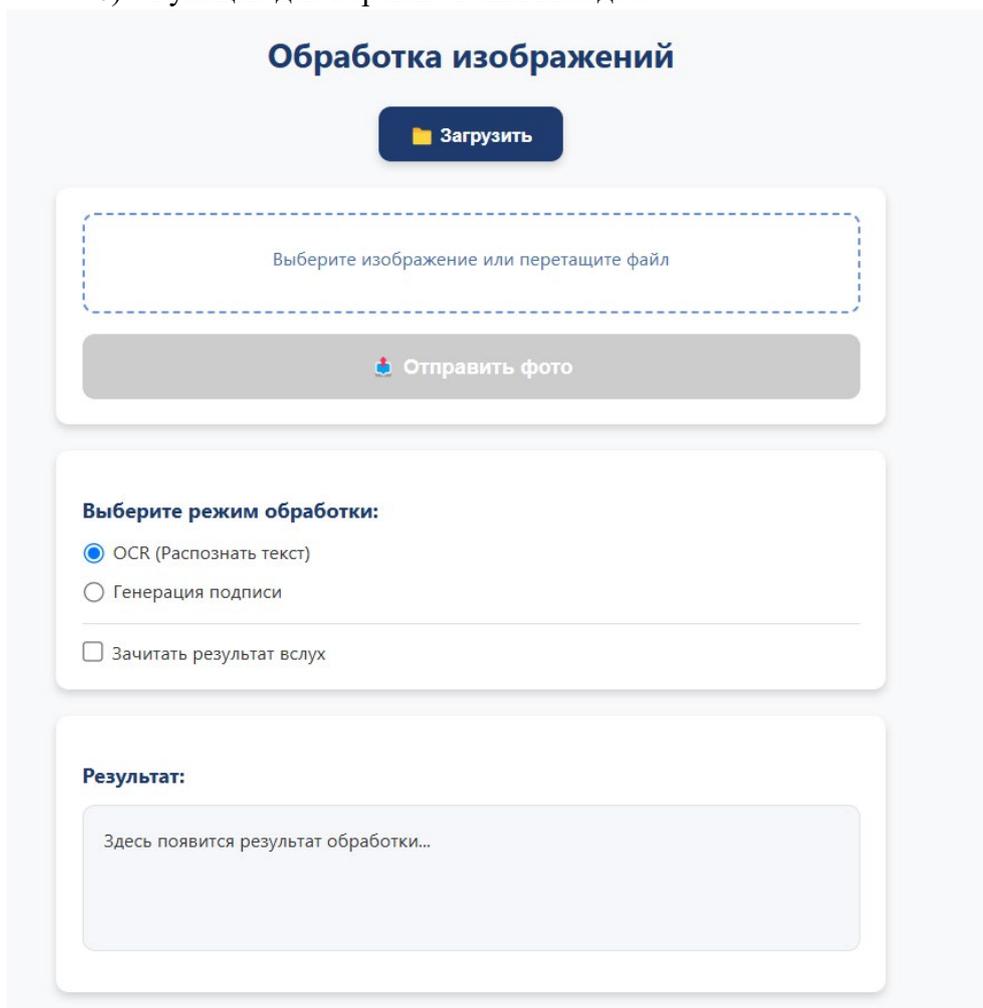


Рисунок 4 – структура сайта

Для создания общедоступного сайта по домену <http://visioassist.ru/>:

- 1) Добавление еще одного правила переадресации на маршрутизаторе (рисунок 5).
- 2) Покупка личного домена на сайте <https://www.reg.ru> (рисунок 6) и добавление правила для переадресации на ip адрес маршрутизатора (рисунок 7)

Добавить правило

Включено

Имя:

Интерфейс:

Частный IP-адрес (хост):

Внутренний IP:

Удаленный хост:

Внутренний диапазон портов: -

Внешний диапазон портов: -

Тип протокола:

Добавить/Применить

Удалить

Рисунок 5 – второе правило переадресации

Домены Всего 2

Поиск Фильтры

Услуга	Действует до	Автопродление	Подробности	Действия
<input type="checkbox"/>  visioassist.online	07.03.2026	<input checked="" type="checkbox"/>	Не подключён к сайту Подключить	
<input type="checkbox"/>  visioassist.ru	07.03.2026	<input checked="" type="checkbox"/>	ns1.reg.ru...	

Рисунок 6 – регистрация домена на сайте <https://www.reg.ru>

Ресурсные записи  Очистить зону

[+ Добавить запись](#)

A	@ → 91.77.167.51	 
A	www → 91.77.167.51	 

Рисунок 7 – добавление правил переадресации на ip адрес маршрутизатора

После всех проделанных операций сайт располагается по адресу <http://visioassist.ru/>.

4.3 Получившаяся структура файлов сервера

(Рисунок 7.1)

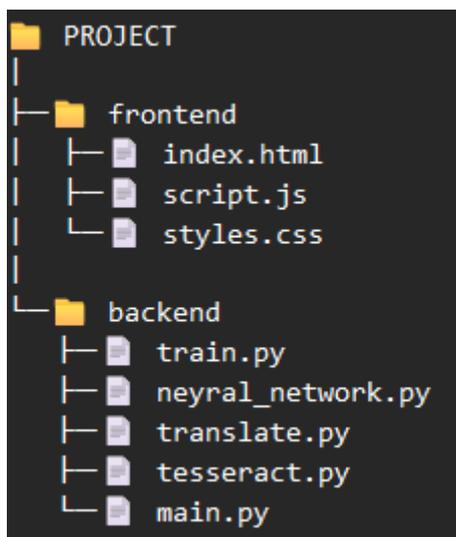


Рисунок 7.1 – структура файлов сервера

5. Создание устройства для слабовидящих и незрячих на базе Raspberry Pi 5.

5.1 Воспроизведение звуков.

Воспроизведение звуков происходит за счет внешней звуковой карты для Raspberry Pi 5 (Рисунок 8). Она крепится снизу основной платы, подключается автоматически через пины GPIO и USB.



Рисунок 8 – внешняя звуковая карта

5.2 Тактовые кнопки.

Для разрабатываемого устройства в качестве цифрового ввода были выбрана две 4-pin тактовые кнопки, они были припаяны к проводам мама – мама и подключены к GPIO 17, GPIO 27 и GND.

5.3 Модуль камеры.

Модуль камеры был установлен в 1 разъем mini 22-pin, формат снимков был выбран максимальный 4608×2592 .

5.4 Питание.

Питание платы были сделано за счет разобранного PowerBank со встроенным разъемом type-c, провода были удлинены за счет пайки (Рисунок 9)



Рисунок 9 – усовершенствованный PowerBank

5.5 Улучшение системы охлаждения.

Система охлаждения была усовершенствована за счет двух медных радиаторов. (Рисунок 10)



Рисунок 10 – радиаторы охлаждения

5.6 Корпус

Был спроектирован корпус для удобства пользования (Рисунок 11).

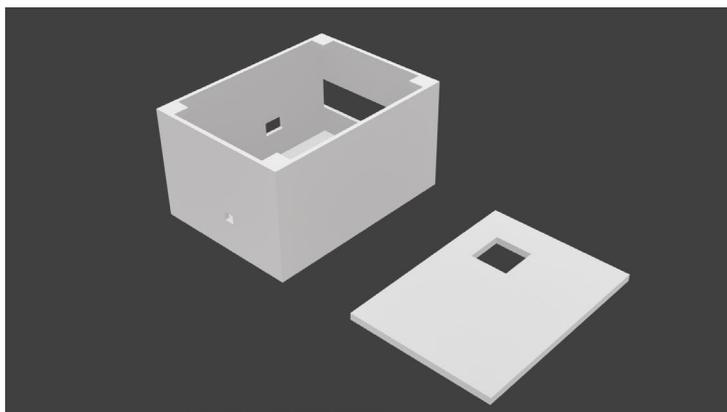


Рисунок 11 – 3-д модель корпуса

5.7 Полностью собранная внутренняя часть

(Рисунок 12).

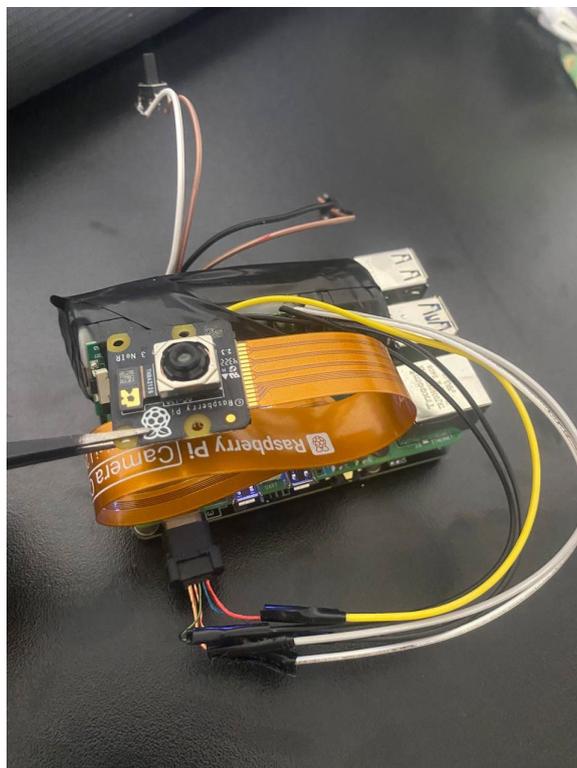


Рисунок 12 – собранная внутренняя часть

6. Разработка программного обеспечения для платы Raspberry Pi 5

В качестве операционной системы была выбрана и установлена Raspberry Pi OS (64-bit) для Raspberry Pi 5.

6.1 Создание программы для осг.

Копия программы п. 3.4.

6.2 Разработка программы для создания снимков.

Основной файл программы – `neural_network.py`. Оно делает снимки с помощью модуля `picamera2` и сохраняет в разделе с подписью `photo_max.jpg`.

6.3 Функция для озвучивания текстов.

Для озвучивания текстов в офлайн была использована нейросеть “`silero tts`”, с голосом “`xenia`”.

6.4 Функция для обращения к серверу.

В файле `api.py` две функции:

- `ping` – проверяет доступность сервера.
- `generation_text` – отправляет `http` запрос по адресу http://91.77.167.51:5000/process_image, после чего ожидает ответа.

6.5 Главный файл Raspberry pi.

Файл с названием `main.py` является связующим всех предыдущих программ п. 5. Она работает по данному алгоритму:

- 1) Инициализация всех классов.
- 2) Проверка соединения с сервером.
- 3) Постоянное считывание сигналов с кнопок, если кнопка нажата, то начинается выполнение функции, если есть подключение:
 - 1 кнопка – происходит запрос на сервер для распознавания текстов.
 - 2 кнопка - происходит запрос на сервер для генерации описания изображений

Если нет подключения к серверу обе кнопки выполняют одну и ту же функцию – распознавание текста на самом устройстве.

- 4) Озвучивание полученных результатов.

7. Перспективы проекта.

Я собираюсь разработать еще одно устройство на базе одноплатного компьютера `Raspberry Pi zero 2w`. Эта плата позволит сделать ношение моего устройства максимально незаметным для самого человека, ведь она будет крепиться к дужке очков.

Также будут производиться постоянные обновления всего ПО, в частности нейросетей. А при дальнейшем возможном сотрудничестве с волонтерскими компаниями можно будет обеспечить этими устройствами многих людей с серьезными проблемами в зрении.

При достаточном финансировании проекта может быть рассмотрен вариант создания своей платы, с нейронным ускорителем, для полного офлайн режима устройства.

ЗАКЛЮЧЕНИЕ

В результате выполненной работы был создано универсальное ПО (сайт) для помощи незрячим и слабовидящим. Разработано собственное устройство на базе одноплатного компьютера `Raspberry Pi 5`, которое является более удобным и быстрым аналогом смартфона с умной камерой, а из-за возможности улавливать инфракрасный спектр, еще и способно работать в слабоосвещенных местах.

Разработанная для устройства нейросеть в ходе испытания проявила себя как достаточно точную модель, что поможет слепым людям четче воспринимать мир на слух используя данное устройство или вебсайт.

В ходе тестирования устройства оно показало себя с лучшей стороны, обладает такими качествами как: высокая эргономичность, хорошая портативность, отличная автономность, громкий и четкий звук.

Из-за большого количества встраиваемых модулей как на программном, так и на аппаратном уровне устройство обладает огромным простором для модификаций.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Raspberry Pi. Официальное руководство для начинающих / пер. с англ. Черников С. В. – М.: ДМК Пресс, 2021. – 262 с.
2. Компьютерное зрение на Python. Первые шаги / Э. Д. Шакирьянов. – М.: Лаборатория знаний. 2021. – 160с.
3. Гринберг, М. Разработка веб-приложений с использованием Flask на языке Python / М. Гринберг. – М.: ДМК Пресс, 2014. – 272 с.
4. Удаленный сервер своими руками. От азов создания до практической работы / Левицкий Н. Д. – М.: Наука и техника, 2021. – 400с.

Приложение А

Листинги программ

Все коды программ занимают свыше 25 страниц, в связи с этим все программы загружены на Github.com, по ссылке <https://github.com/Zamoroz228/Visioassist>, по qr коду (Рисунок А.1)



Рисунок А.1 – qr код для доступа к программам проекта

Приложение Б

QR код для доступа к web сайту visioassist.ru.



Рисунок Б.1 – qr код для доступа к web сайту visioassist.ru

ПРОТОТИП НАГРЕВАТЕЛЬНОГО КОНЦЕНТРАТОРА СОЛНЕЧНОГО ИЗЛУЧЕНИЯ С ТРАНСФОРМИРУЕМОЙ КОНСТРУКЦИЕЙ РЕФЛЕКТОРА

Курипта Егор Константинович

Комсомольск-на-Амуре г, МОУ Гимназия №45, 11 класс

Секция: Космические аппараты и ракеты-носители

*Научный руководитель: Кириевский Дмитрий Алексеевич, ведущий конструктор сектора
АО «ГКНПЦ им. М.В. Хруничева»*

Аннотация

Целью работы является проектирование функционального макета НКСИ с трансформируемой конструкцией рефлектора с целью испытаний системы раскрытия и ключевых технических решений.

В ходе работы была рассчитана предварительная оптическая схема концентратора с помощью аналитических методов, а также с использованием сервиса desmos. В программе Компас-3D спроектирована зонтичная двухступенчатая конструкция рефлектора с тросовым приводом раскрытия, ступица и корпус НКСИ. Разработан и протестирован механизм юстировки силовых спиц рефлектора.

Произведена сборка и доводка системы раскрытия рефлектора. Проведены успешные испытания и по их результатам сформулированы задачи на дальнейшую доработку конструкции.

Перспективой развития данного проекта является применение в качестве нагревательной установки в области космической металлургии, использование в качестве силовой установки для строительства на поверхности Луны, эксплуатация для освещения кратеров на Южном полюсе Луны

СПИСОК ПРИНЯТЫХ СОКРАЩЕНИЙ

Сокращение	Расшифровка
БСР	Быстросъёмный Разъём
НК	Нагревательная Камера
НКСИ	Нагревательный Концентратор Солнечного Излучения
ОП	Отражающее Полотно
ПН	Полезная Нагрузка
СЭС	Система Энергоснабжения
ФКП	Факторы Космического Полёта

ВВЕДЕНИЕ

В перспективе развития космонавтики лежит множество задач и процессов, требующих создания и долговременного поддержания высокой температуры. Например, задачи космической металлургии на поверхности планет: создание сплавов, получения конструкционных материалов; строительные задачи: спекание грунта в строительные блоки; задачи систем жизнеобеспечения: поддержание тепла, генерация электроэнергии, проведение высокотемпературного электролиза. Вопрос получения энергии для поддержания высоких температур в вышеописанных процессах за пределами околоземной орбиты остаётся актуальным. Одним из наименее затратных, относительно массы установки к развиваемой мощности и количеству затрачиваемой энергии на поддержание работоспособности системы, вариантов получения и поддержания высоких температур является концентрация солнечного излучения.

Мощность концентраторов солнечного излучения напрямую зависит от фокальной площади главного зеркала (рефлектора). В таком случае, при использовании солнечных концентраторов в космосе становится актуальным вопрос доставки крупногабаритной конструкции на орбиту или поверхность небесного тела. Используя развёртываемую конструкцию рефлектора под головным обтекателем ракеты-носителя можно установить рефлектор большей площади.

В то же время, масса конструкции пропорциональна фокальной площади рефлектора, в таком случае появляется необходимость минимизации массы отражающей поверхности, чего можно добиться за счёт использования мягкого отражающего полотна.

В данной работе рассмотрен прототип конструкции нагревательного концентратора солнечного излучения (НКСИ) оптической схемы Кассегрена с трансформируемым рефлектором зонтичной конструкции.

Цель работы: Проектирование функционального макета НКСИ с трансформируемой конструкцией рефлектора с целью испытаний системы раскрытия и ключевых технических решений.

Задачи:

1. Провести анализ аналогичных проектов;
2. Произвести расчёт оптической системы и разработать конструктивно-компоновочную схему НКСИ;
3. Спроектировать и создать функциональный макет трансформируемой конструкции рефлектора;
4. Установить и отъюстировать отражающее полотно;
5. Разработать и создать систему автоматического раскрытия рефлектора и индикации раскрытия силовых лучей;
6. Разработать и создать силовую конструкцию НКСИ с корпусом и такелажными элементами.

ОСНОВНАЯ ЧАСТЬ

1. Анализ существующих решений

Для анализа были взяты некоторые существующие работы по теме нагревательных оптических концентраторов в космонавтике.

1. Cassegrain Solar Concentrator System for ISRU Material Processing

В работе[1] рассмотрен нагревательный солнечный концентратор для плавления лунного реголита и получения ресурсов in-situ(на месте).

Концентратор собран по оптической схеме Кассегрена, рефлектор изготовлен из алюминия, в качестве отражающей поверхности используется серебряное покрытие. Апертура рефлектора равна 0.5 м, коэффициент концентрации(отношение фокальной площади рефлектора к площади фокального пятна в точке фокуса) равен 3000 к 1.

Достоинства:

- При помощи короткофокусной оптической схемы Кассегрена было достигнуто фокальное пятно диаметром 2.2 см;
- При ведении Солнца в течении 45 минут и плотности потока 800 Вт/м^2 удалось достичь температуры заготовки около 980°C и размеров: $9.5 \times 9.5 \times 2.4 \text{ мм}$;
- Расширенная оптическая схема, позволяющая увеличить коэффициент концентрации до 6000 к 1, а также стабильно удерживать фокальное пятно.

Недостатки:

- В процессе эксплуатации конструкции рабочая группа систематически наблюдала значительные тепловые потери капсулы с реголитом, приходящиеся на излучение в инфракрасном диапазоне спектра;
- Недолговечность серебряного покрытия, деградация его отражательной способности, выявленная в процессе испытаний рефлектора;
- Аберрации, наблюдаемые у фокального пятна, что в конечном счёте снижает коэффициент концентрации и равномерность проплава заготовки.

2. Solar Thermal Propulsion System for Small Spacecraft

В работе [2] представлена газодинамическая силовая установка, принцип работы которой заключается в испарении рабочего тела при помощи концентрированного солнечного излучения. Солнечное излучение собирается несколькими рефлекторами, собранными по схеме Кассегрена и передаётся по оптоволоконному каналу в приёмник излучения, откуда дальше идёт в топливный бак.

Solar Thermal Propulsion System for Small Spacecraft

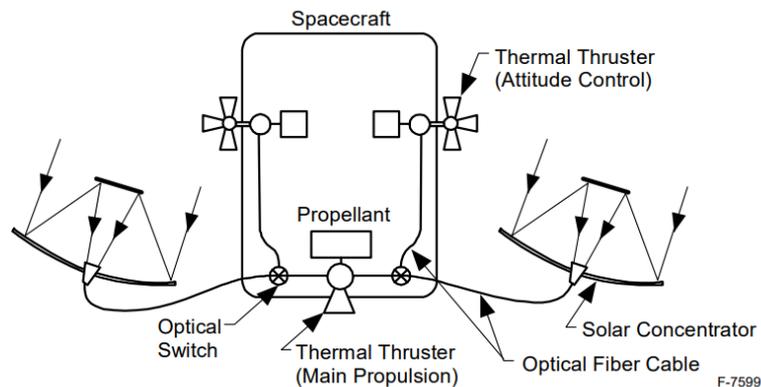


Рисунок 1 - Эскиз солнечной терморактивной системы для малых космических аппаратов

Достоинства:

- Использование оптоволоконной передачи солнечного излучения, что упрощает оптическую схему при передаче сфокусированного излучения на расстояние;
- фасетная конструкция рефлектора (собирающая поверхность образуется несколькими отражателями) упрощает её изготовление, повышает отказоустойчивость системы и упрощает вывод конструкции на орбиту.

Недостатки:

- Трудоёмкое синхронное наведение нескольких отражателей на Солнце, проблема с сохранением юстировки и стабилизации в течении длительного времени.

3. Solar Dynamic Power for Space Station Freedom

В работе[3] описана конструкция расширенной системы энергоснабжения(СЭС) ранних концепт-проектов ОС Freedom. Принцип действия СЭС заключается в том, что испарённый и нагретый на освещённом участке полёта теплоноситель циркулирует по замкнутому контуру и вращает турбину индукционного электрического генератора переменного тока. Испарение и нагрев теплового аккумулятора с теплоносителем производится при помощи факетного параболического концентратора.

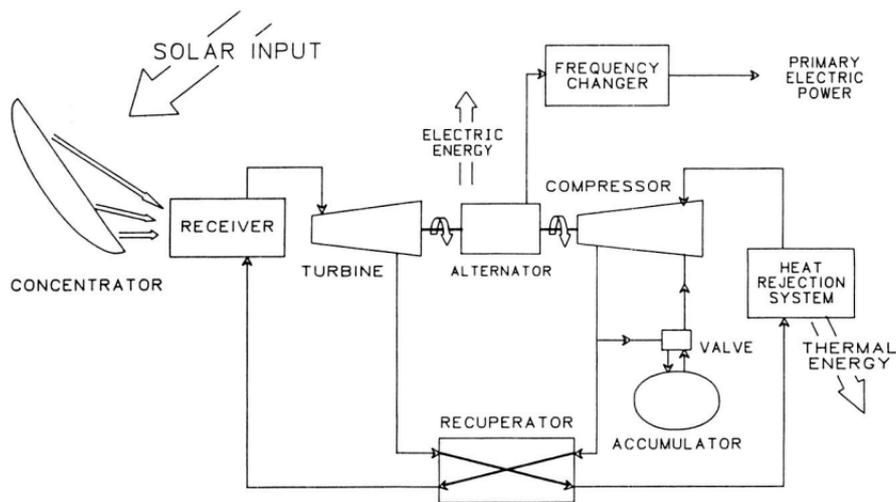


FIGURE 2. - SOLAR DYNAMIC CLOSED BRAYTON CYCLE.

Рисунок 2 - Принципиальная схема солнечной силовой установки закрытого цикла Брайтона

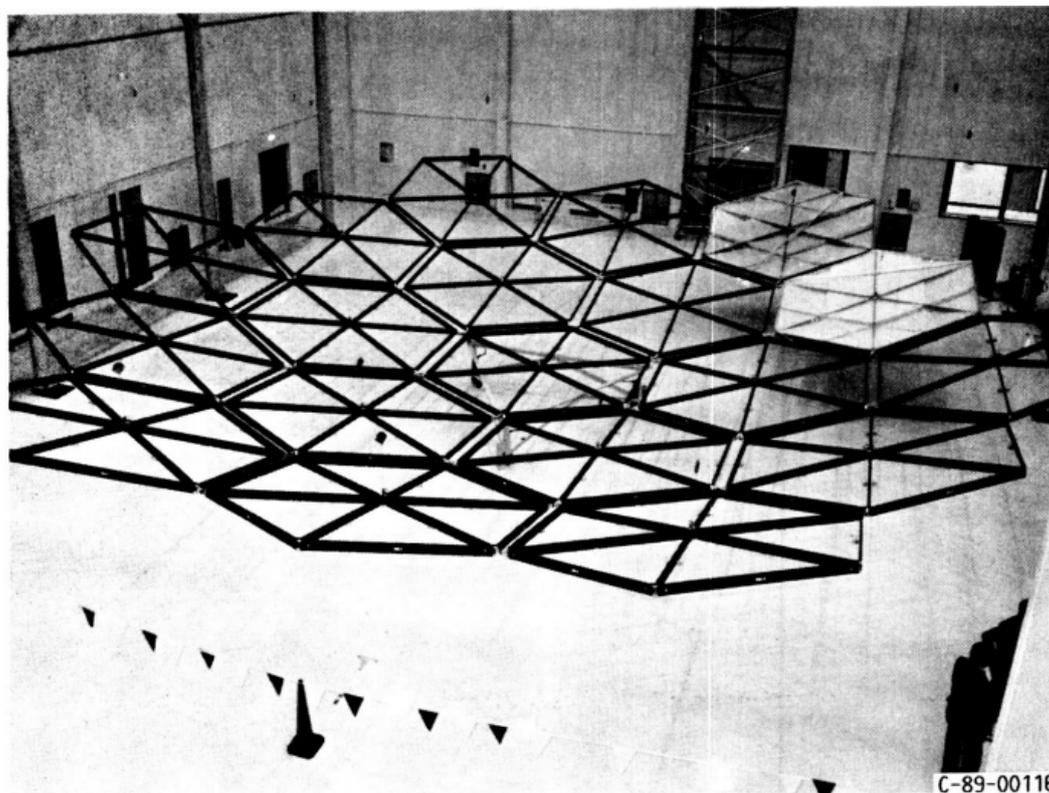


FIGURE 7. - CONCENTRATOR PANEL STRUCTURE.

Рисунок 3 - Конструкция рефлектора

Достоинства:

- КПД машины, работающей по циклу Брайтона значительно выше КПД фотоэлектрических преобразователей;
- Система представлялась долговечнее на тот момент ввиду отсутствия устойчивых к деградации ФЭП.

Недостатки:

- Сборку параболического отражателя необходимо было осуществлять вручную, что значительно усложняет и продлевает процесс введения системы в эксплуатацию;
- В современных реалиях столь крупный отражатель (апертура 18 м) подвержен стремительной деградации от частиц космического мусора и микрометеороидов.

1.1 Результаты анализа аналогов

По результатам проведённого анализа были сделаны следующие выводы, которые впоследствии легли в основу идеи разрабатываемого проекта:

1. Во всех работах рефлектор в эксплуатационном состоянии представляет из себя монолитную конструкцию. В таком случае масштабируемость возможно лишь при увеличении количества отражателей, а соответственно узлов системы и, как следствие

уменьшение её надёжности. Также растёт масса конструкции и сложность транспортировки.

2. Также, с ростом площади собирающей поверхности растёт и масса конструкции, увеличивается момент инерции, усложняется система наведения и стабилизации. Помимо этого, доставка жёсткой отражающей поверхностью сопряжено с риском поломки конструкции.

1.2 Подходы, принятые в разрабатываемой конструкции

- Для сохранения возможности масштабирования (увеличение апертуры) конструкция рефлектора разрабатывается трансформируемой. Также, трансформируемость конструкции позволит обеспечить доставку НКСИ на орбиту/поверхность небесного тела в полностью работоспособном состоянии с минимум подготовительных операций перед применением. К тому же, без трансформируемой конструкции рефлектора отражающая поверхность подвергается значительной деградации под влиянием факторов космического полёта (ФКП), соответственно, сокращается срок службы устройства;
- Для снижения масса устройства, отражающая поверхность представляет из себя мягкую оболочку.
- Для увеличения отказоустойчивости конструкции при проектировании силового каркаса рефлектора предусматривается возможность замены фрагмента отражающего полотна(ОП).

2. Оптическая схема

Оптическая схема является основополагающей частью проекта. В зависимости от неё будет производиться дальнейшее проектирование конструкции НКСИ.

В данном разделе приведены результаты расчета фокальной площади рефлектора, а также геометрической формы элементов оптической схемы.

2.1 Расчёт параметров рефлектора

Исходными данными для расчета были приняты:

- Материал сплава - сплав Розе;
- объем заготовки для переплавки;
- время плавления;
- мощность солнечного излучения на уровне поверхности Земли.

Допущениями, принятыми при расчёте является:

- Не учитываются погрешности при производстве и сборке конструкции рефлектора;
- отсутствие конвективного теплообмена между нагревательной камерой (НК) и окружающим воздухом;
- равномерный и постоянный поток солнечного излучения;

- удельная теплоёмкость сплава Розе и нагревательной камеры приняты постоянными величинами;
- рассмотрен процесс плавления уже нагретой до температуры плавления заготовки.

Таблица 1 - Расчёт радиуса рефлектора

Исходные данные	Расчёт
<p>$T = 367 \text{ K}$ - температура плавления сплава Розе;</p> <p>$S_{\text{НК}} = 3,73 \cdot 10^{-3} \text{ м}^2$ - площадь излучающей поверхности НК;</p> <p>$\varepsilon = 0,9$ - коэффициент поглощения поверхности НК;</p> <p>σ - постоянная Стефана-Больцмана;</p> <p>$\alpha = 0,85$ - коэффициент отражения отражающего полотна рефлектора;</p> <p>$P_{\text{СИ}} = 200 \text{ Вт/м}^2$ - мощность солнечного излучения на уровне моря;</p>	<p>1) $P_{\text{пот}} = P_{\text{НКСИ}}$ - условие осуществления нагрева материала.</p> <p>Где $P_{\text{пот}} = \sigma \cdot \varepsilon \cdot S_{\text{НК}} \cdot T^4$ - мощность тепловых потерь излучением,</p> <p>$P_{\text{НКСИ}} = P_{\text{СИ}} \cdot S_{\text{р}} \cdot \alpha$ - мощность НКСИ.</p> <p>Где $S_{\text{р}}$ - площадь собирающей поверхности рефлектора</p> <p>2) $S_{\text{р}} = k \cdot \frac{\sigma \cdot \varepsilon \cdot S_{\text{НК}} \cdot T^4}{P_{\text{СИ}} \cdot \alpha} =$ - количество теплоты для нагрева сплава Розе.</p> <p>Где k - коэффициент запаса равный 20.</p>
Предмет расчёта	Результат
$r_{\text{НКСИ}}$ - радиус рефлектора	$r_{\text{НКСИ}} = 0,341 \text{ м}$

В результате проделанного расчёта было получено значение необходимого радиуса рефлектора и взято ближайшее большее значение апертуры: 700 мм

2.2 Проектирование оптической схемы

Оптической схемой была выбрана схема Кассегрена, где образующей рефлектора и контррефлектора является парабола.

Для предварительного построения и расчёта геометрических характеристик был использован сервис desmos. Исходя из рассчитанной апертуры было выбрано фокусное расстояние, так, чтобы кривизна рефлектора позволяла сложить силовой луч в два колена. Проверка этого производилась моделированием в среде Компас-3D и выполнялась итеративно.

Предварительное проектирование (Рисунок 4) дало представление о следующих характеристиках НКСИ:

- Диаметр фокального пятна в точке сбора лучей на нагревательной камере - 5 мм;
- Апертура контррефлектора;
- Фокусное расстояние контррефлектора;
- Расположение контррефлектора - 554.5 мм (от верхней грани диска крепления кронштейнов первого колена до вершины образующей);
- Диаметр входного отверстия ступицы - 42 мм.

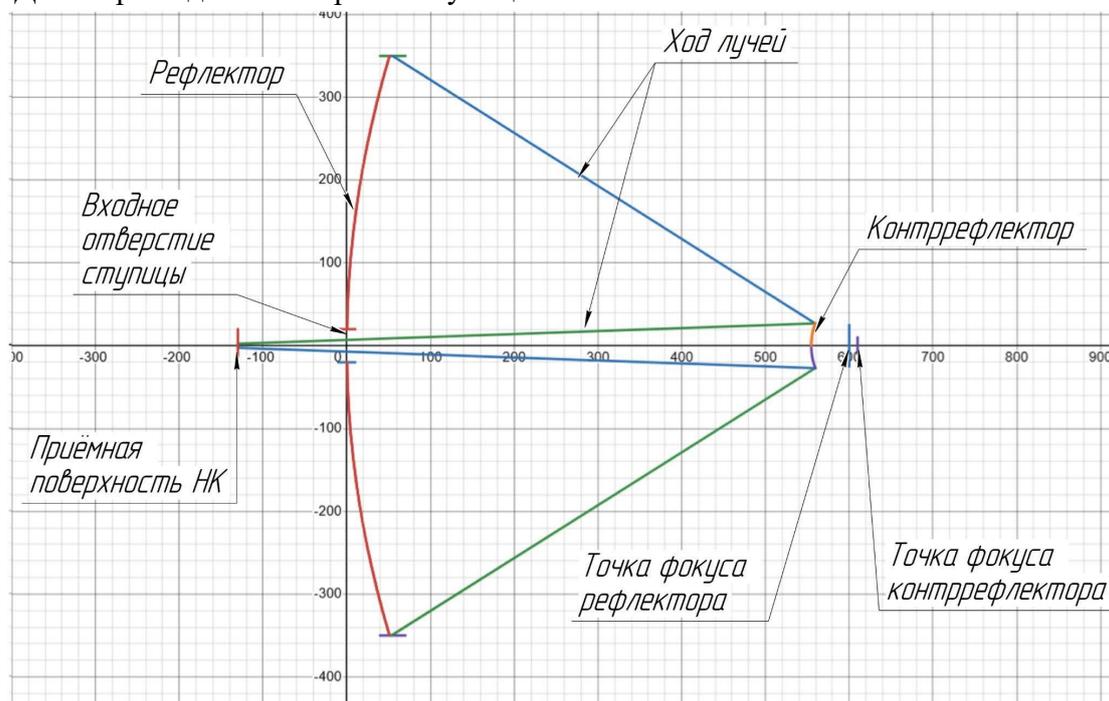


Рисунок 4 - Предварительная геометрия оптической схемы НКСИ, построенная в сервисе *Desmos*

Параметры рефлектора:

Апертура - 700 мм

Радиус кривизны - 1225 мм

Фокусное расстояние - 600 мм

2.2.1 Расчёт параметров контррефлектора

Исходными данными для расчета были приняты:

- Параметры рефлектора;
- фокусное расстояние рефлектора;
- расположение точки сбора лучей;
- допустимый диаметр фокального пятна в точке сбора лучей в пределах 10 мм.

Таблица 2 - Расчёт фокусного расстояния контррефлектора

Исходные данные	Расчёт
$a_1 = 684,5$ мм - расстояние от точки сбора лучей до вершины контррефлектора; $a_2 = 45,5$ мм - расстояние от точки сбора лучей до вершины контррефлектора	$f = \frac{a_1 \cdot a_2}{a_1 + a_2} = 42,67$ мм
Предмет расчёта	Результат
f - фокусное расстояние контррефлектора	$f = 42,67$ мм

Параметры контррефлектора:

Апертура - 54 мм

Радиус кривизны - 85,3 мм

Фокусное расстояние - 42,67 мм

Расстояние от вершины параболы до точки сбора лучей - 684,5 мм,
 где точкой сбора лучей является крышка НК.

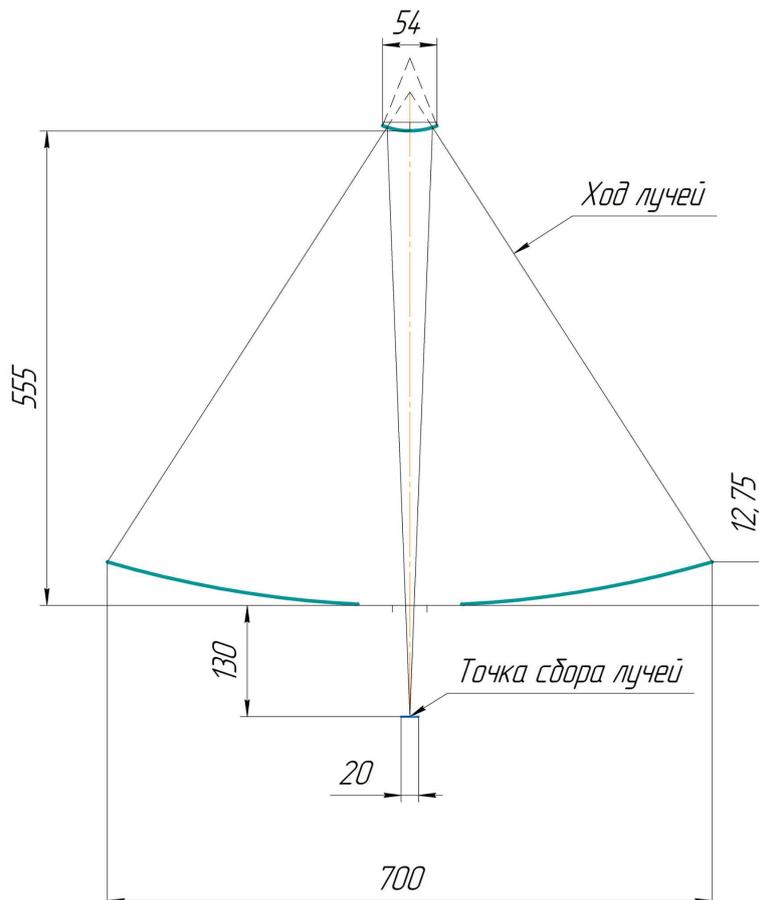


Рисунок 5 - Оптическая схема НКСИ

3. Конструктивно-компоновочная схема прототипа НКСИ

Основным принципом проектирования является модульность разрабатываемой конструкции, по причине потенциальной необходимости в доработке отдельных деталей на основе опыта сборки и тестирования. Также, в процессе разработки конструкции во многие технические решения была заложена необходимость в будущей масштабируемости.

Прототип состоит из следующих частей:

1. Силовой каркас рефлектора;
2. Штанга контррефлектора;
3. Контррефлектор
4. Ступица;
5. Двигатель системы раскрытия;
6. Корпус;
7. Контейнер нагревательной камеры.

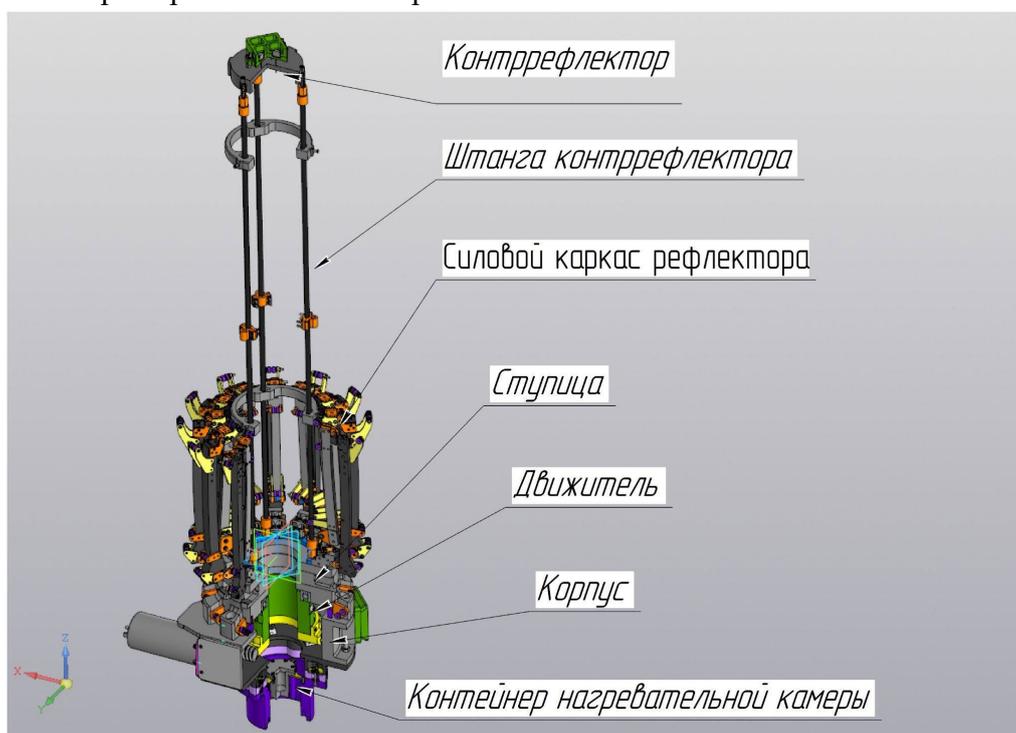


Рисунок 6 - Состав прототипа НКСИ в разрезе

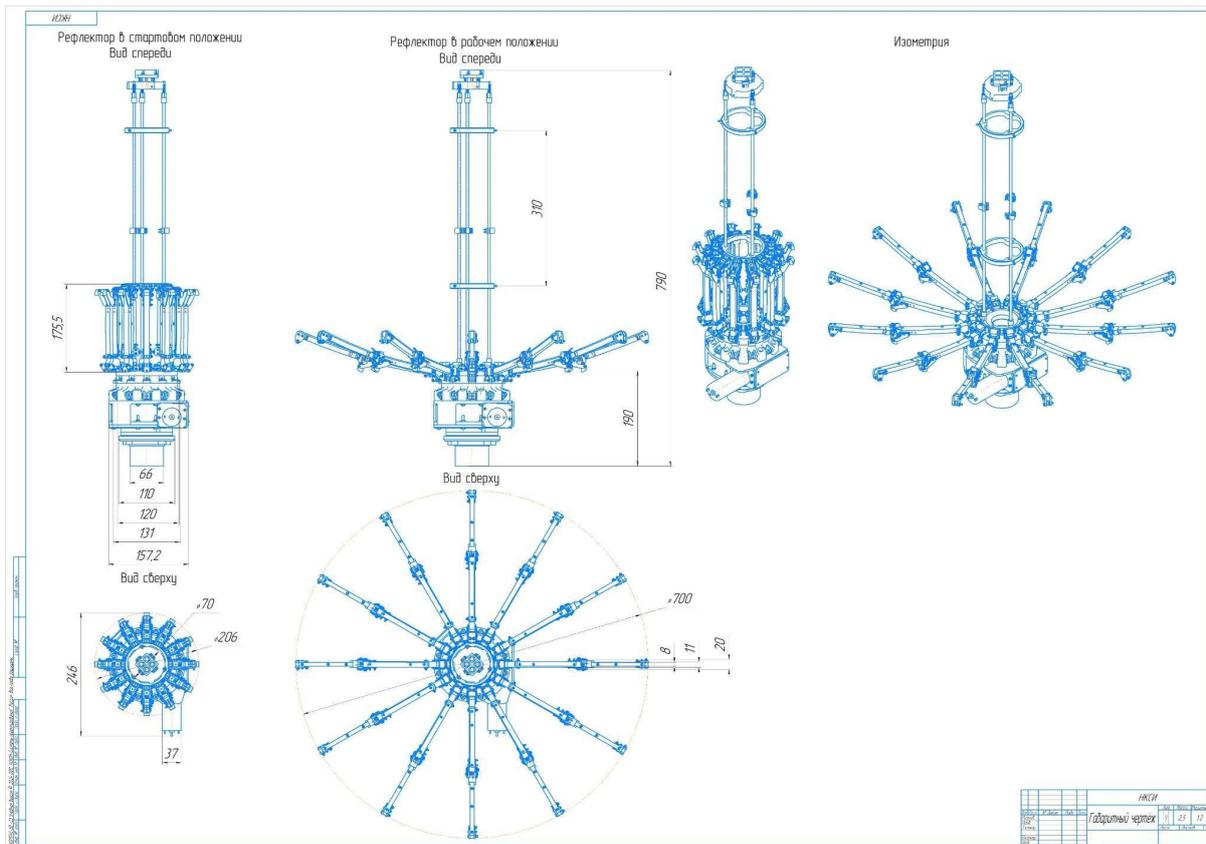


Рисунок 7 - Габаритный чертёж НКСИ в транспортировочном и рабочем положении

Масса: 2,5 кг

Габаритные размеры в транспортном положении: 790x243x206 мм

Габаритные размеры в рабочем положении: 790x700x700 мм

3.1 Система раскрытия рефлектора

В качестве системы раскрытия была выбрана зонтичная система с жёсткими рёбрами, раскрытие которой осуществляется путём натяжения тросов за счёт намотки на барабан.

Критериями выбора системы раскрытия являлись:

- технологическая реализуемость в рамках проектной работы;
- надёжность при большом количестве циклов “раскрытие-складывание”;
- отсутствие ударной нагрузки при раскрытии.

Последние два критерия являются проверкой масштабируемости конструкции, так как в будущем апертура рефлектора будет составлять несколько метров. Ударные нагрузки и возникающие вследствие них вибрации будут негативно сказываться на качестве отражающей поверхности. К тому же, рефлектор будет необходимо сделать складываемым, чтобы защитить его от эрозии со стороны мелкодисперсной пыли в случае использования на поверхности Луны

или микрометеороидов в случае использования на орбите. Также, в случае деградации оптической поверхности необходимо предусмотреть возможность её замены.

Тросовая системы была введена на основе анализа патента[4] конструкции трансформируемой параболической антенны аппарата форм-фактора CubeSat 6U. Была отмечена высокая ударная нагрузка при раскрытии второго колена силовых лучей, а также невозможность складывания рефлектора, что недопустимо при проектировании НКСИ.

3.2 Силовой каркас рефлектора

Силовой каркас рефлектора формирует геометрию отражающего полотна. В состав силового каркаса входят 12 силовых лучей.

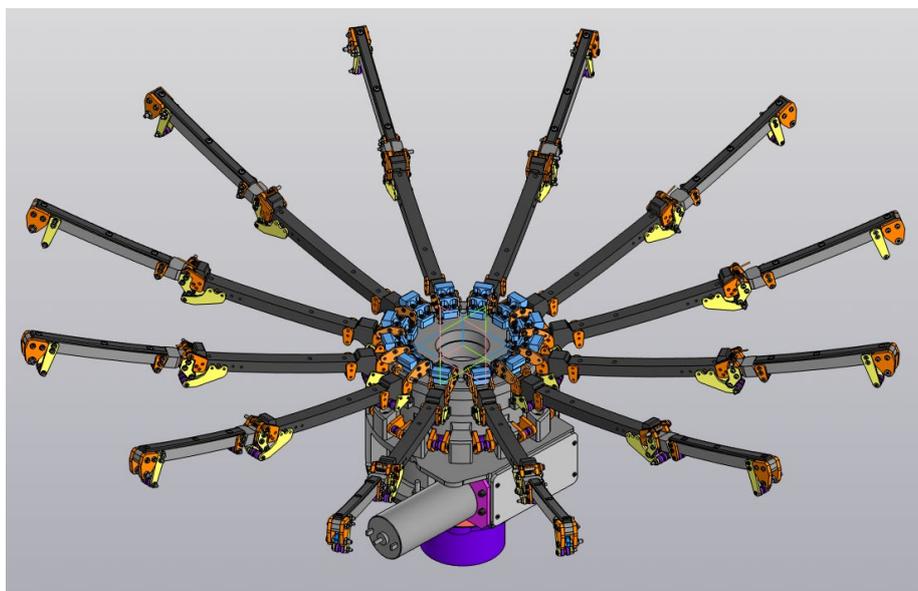


Рисунок 8 - Силовой каркас НКСИ в раскрытом положении

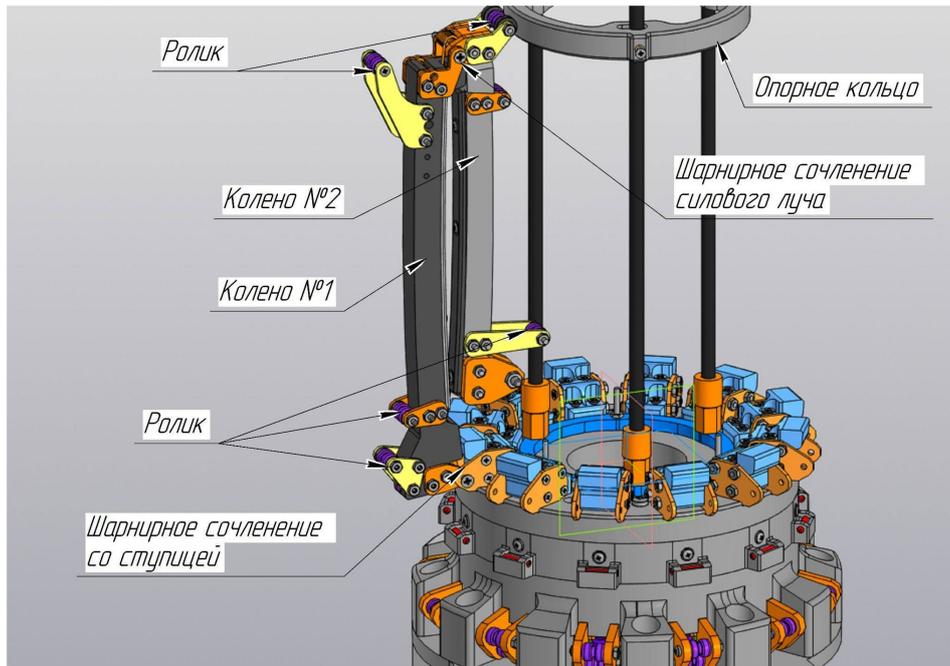


Рисунок 9 - Устройство силового каркаса рефлектора

Силовой луч является отдельным узлом, это было сделано для обеспечения ремонтпригодности конструкции, недостаток которой был выявлен в процессе тестирования механизма раскрытия.

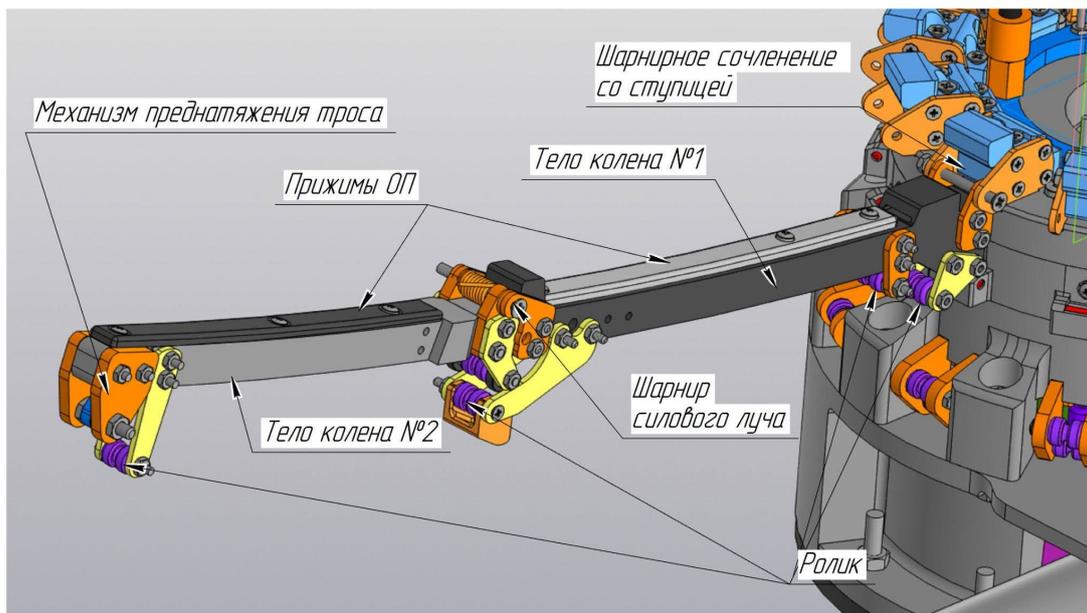
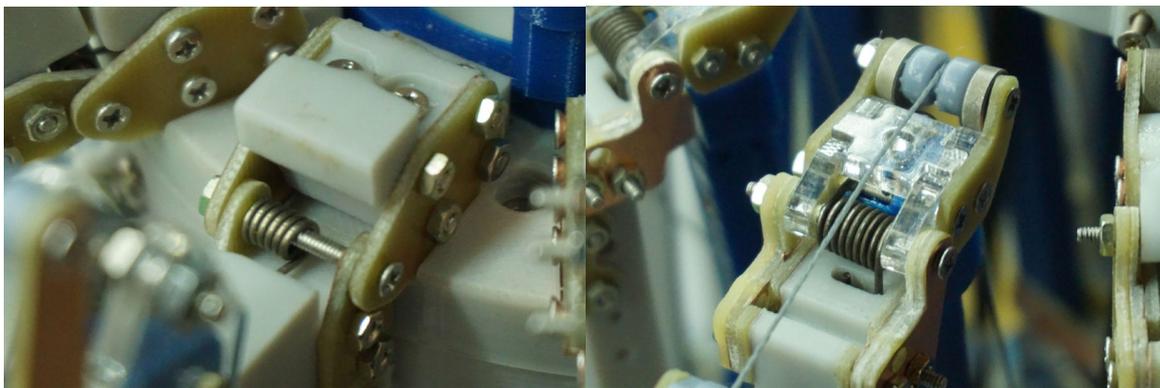


Рисунок 10 - Устройство силового луча

В шарнирных сочленениях первого колена и ступицы, а также между первым и вторым коленом установлены пружины кручения, работающие на стягивание к центру ступицы, что является прототипом системы складывания рефлектора



*Рисунок 11 - Пружина кручения
между первым коленом и ступицей*

*Рисунок 12 - Пружина кручения
в сочленении первого и второго колен*

Для индикации установки силового луча в рабочем положении в нижнем торце каждого колена установлен магнит, а на ступице и верхнем торце первого колена установлены герконы. Система магнит-геркон необходима для индикации установки силового луча в рабочем положении. Сигналом электроприводу к остановке процесса раскрытия является отсечка по пределу силы тока мотора.

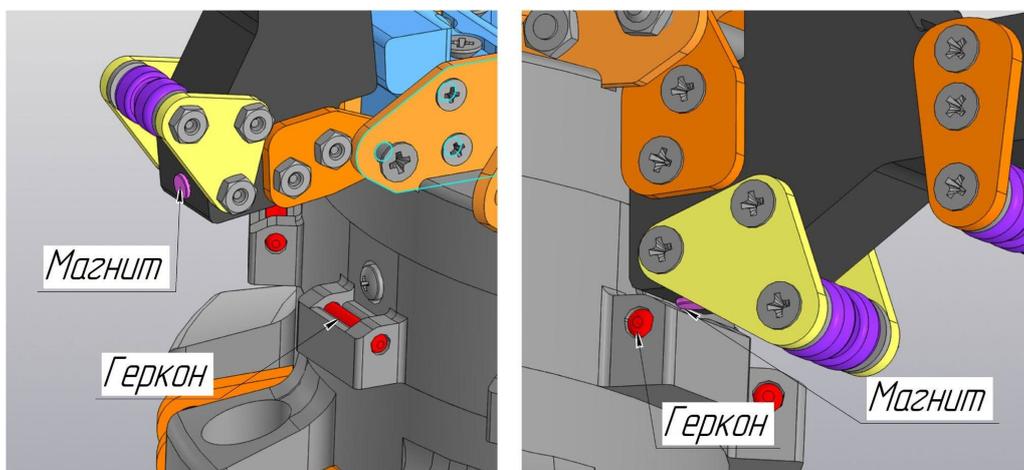


Рисунок 13 - Система индикации положения силового луча

Между ступицей и первым коленом силовых лучей, раскрытых в рабочее положение установлены регулировочные винты, необходимые для юстировки угла раскрытия первого колена для компенсации дефектов изготовления отдельных деталей НКСИ.

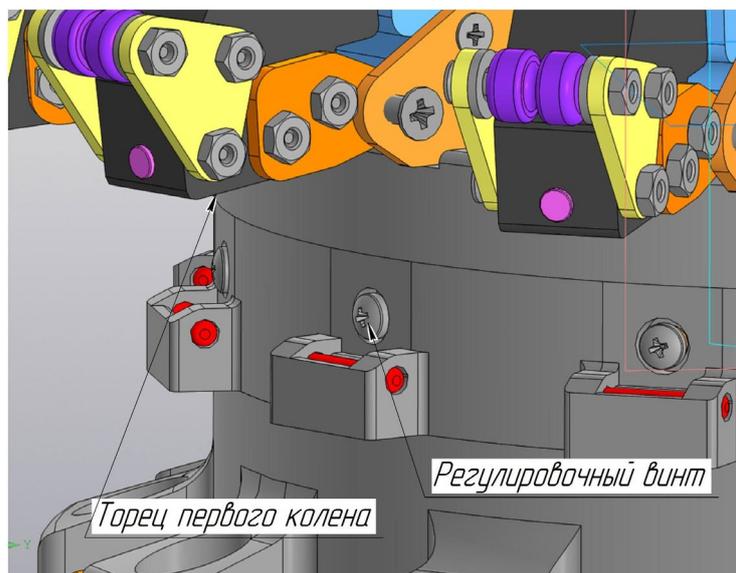


Рисунок 14 - Регулировочный винт

3.2.1 Силовой луч

Силовой луч представляет из себя двухсекционную конструкцию. Колено №1 и колено №2 соединены шарнирно. Отношение длин колен по образующей составляет 1 к 0,9 , соответственно.

Методом для изготовления тела колен выбран метод FDM-печати PETG и PLA пластиком.

Проушины крепления роликов изготовлены посредством фрезеровки из стеклотекстолита, омеднённого с одной стороны. Некоторые шарнирные петли изготовлены из органического стекла посредством резки на лазерном станке.

Прижимы и ролики изготовлены методом фотополимерной печати в силу высоких требований к качеству поверхности.

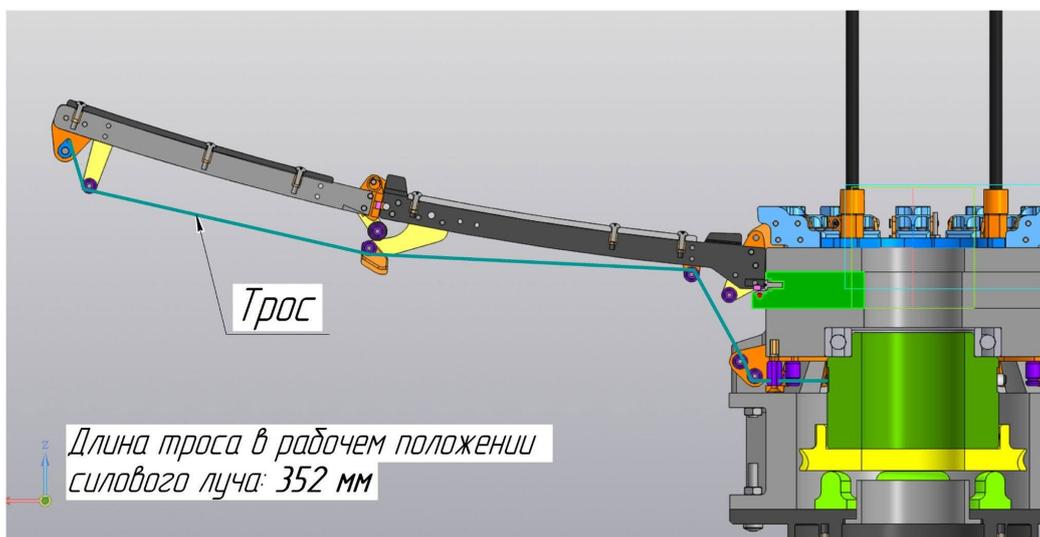


Рисунок 15 - Силовой луч в продольном сечении. Схема прокладки троса(зелёная кривая) на силовом луче в рабочем положении

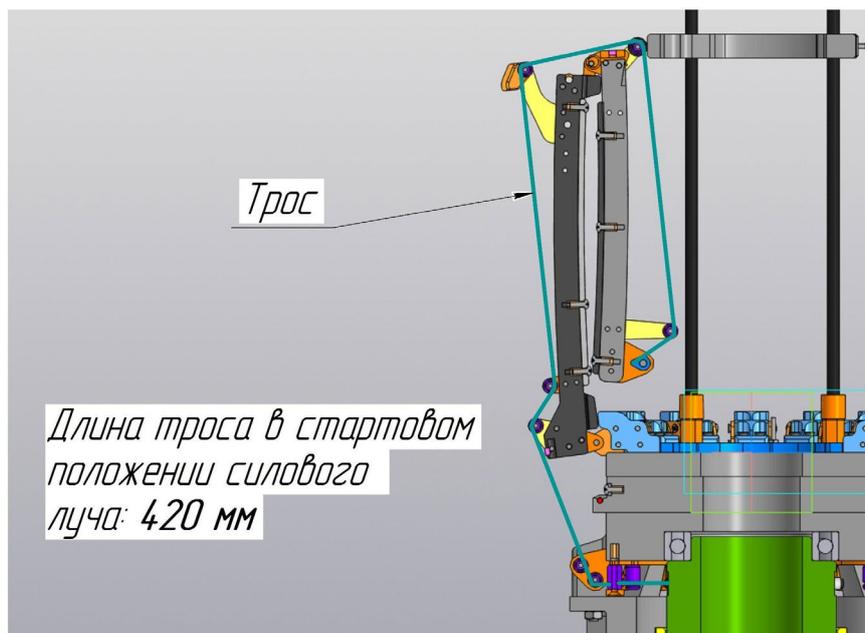


Рисунок 16 - Силовой луч в продольном сечении. Схема прокладки троса(зелёная кривая) на силовом луче в стартовом положении.

3.3 Ступица

Ступица является силовой конструкцией, к которой осуществляется крепление силовых лучей, элементов системы раскрытия, деталей корпуса НКСИ и штанги контррефлектора.

Ступица состоит из трёх деталей:

- диск крепления силовых лучей (Рисунок 17);
- промежуточный диск (Рисунок 18);
- диск системы раскрытия (Рисунок 19).

Одним из требований, предъявляемых к конструкции является унификация и ремонтпригодность. Детали проектировались, предусматривая возможность изменения конструкции, а также для удобства работы по сборке-разборке НКСИ.

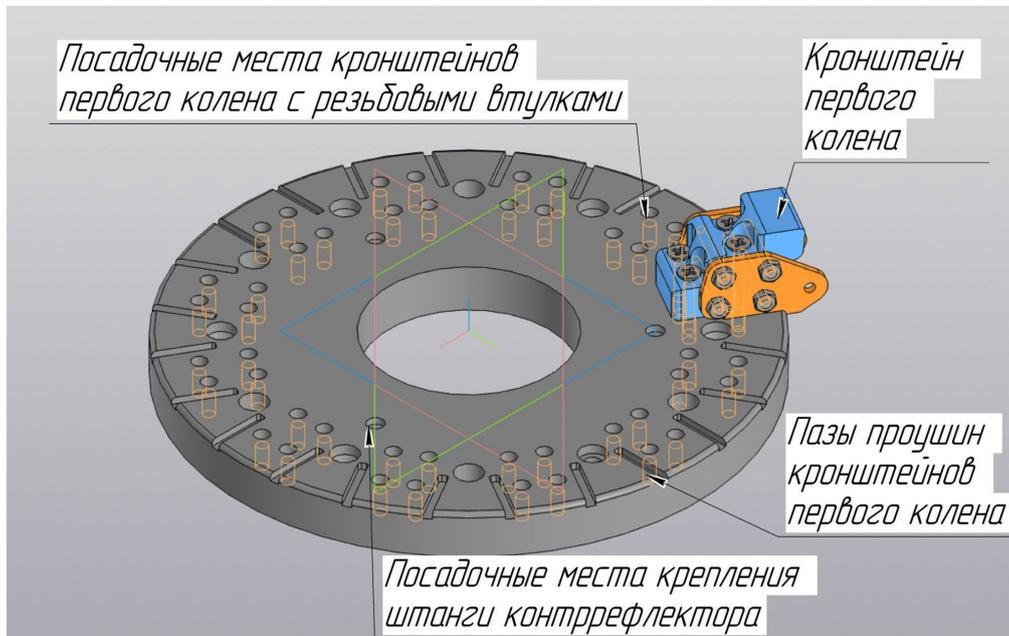


Рисунок 17 - Диск крепления силовых лучей с одним прикрепленным кронштейном первого колена

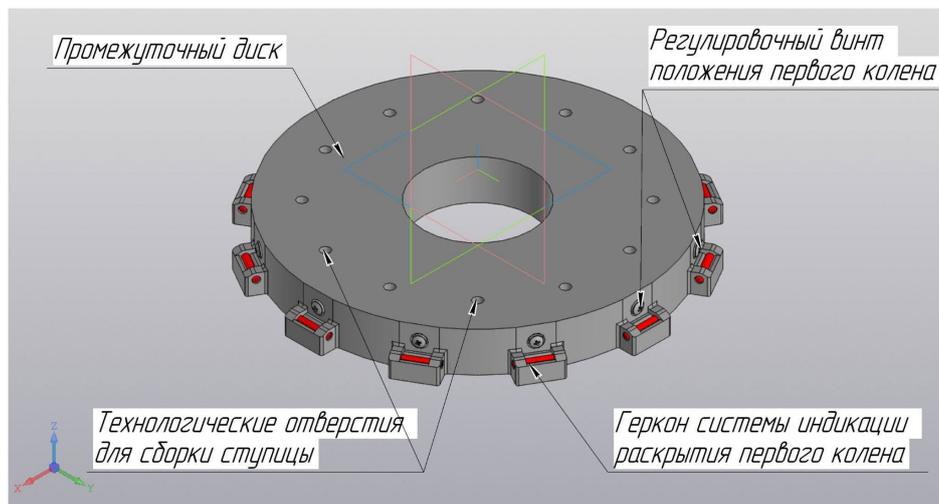


Рисунок 18 - Промежуточный диск ступицы с установленными герконами системы индикации раскрытия первого колена

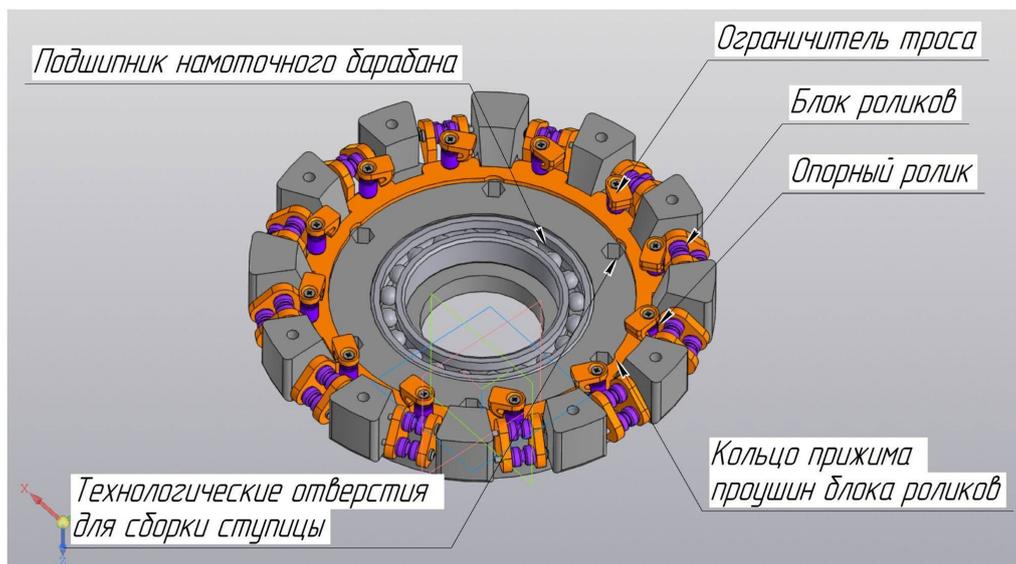


Рисунок 19 - Диск крепления системы раскрытия, вид снизу

Ступица собрана каскадом. Диски фиксируются на крепление болт-гайка.

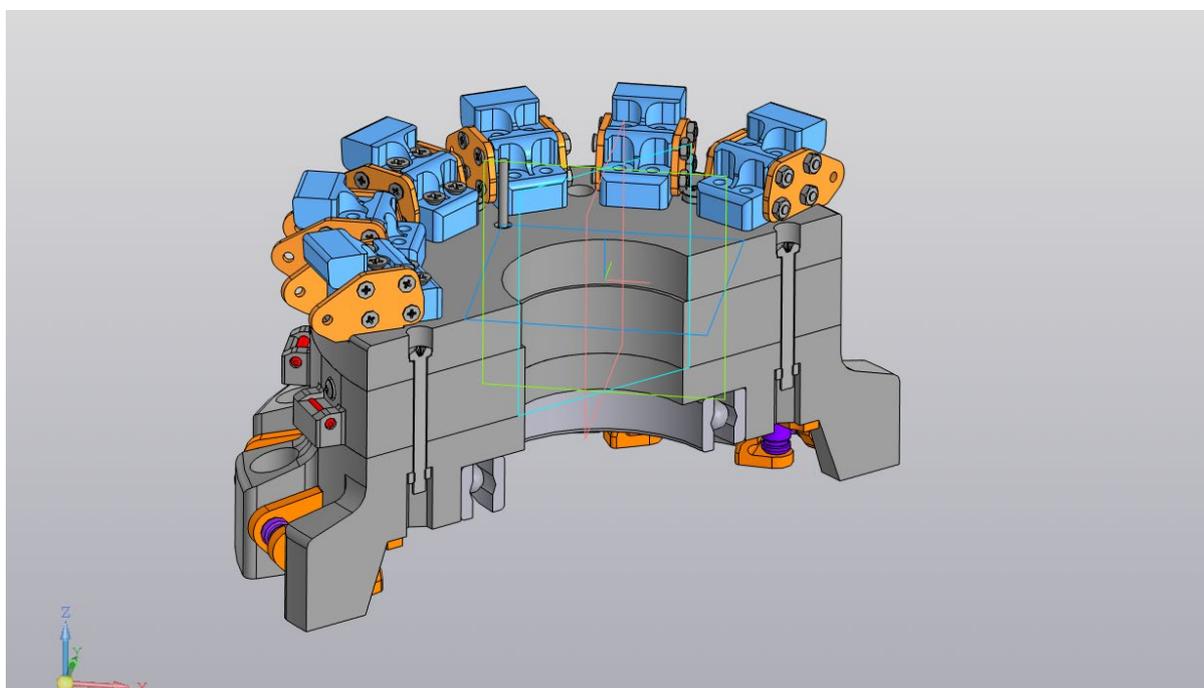


Рисунок 20 - Ступица в собранном виде в поперечном разрезе

3.4 Штанга контррефлектора

Штанга контррефлектора необходима для:

- фиксации стакана контррефлектора;
- фиксации блока фоторезисторов наведения на Солнце;
- установки опорного кольца, которое выполняет роль ограничителя при установке силовых лучей в транспортировочном положении.



Рисунок 21 - Состав штанги контррефлектора

Направляющие штанги выполнены из углеволоконных трубок наружным диаметром 5 мм. Два опорных кольца центруют направляющие, вся система в сборе установлена соосно главной оптической оси.

На данный момент на месте контррефлектора установлена мишень для оценки геометрических параметров собираемого рефлектором фокального пятна для будущих испытаний.

3.5 Кинематика системы раскрытия

Конструкция системы раскрытия должна удовлетворять следующим требованиям:

- плавность раскрытия. Движение без рывков;
- геометрия рефлектора в рабочем положении близкая к расчётной;
- ремонтпригодность;
- модульность.

Исходя из этих требований была разработана система, включающая следующие элементы:

- мотор-редуктор;
- червячный редуктор;
- намоточный барабан;
- система роликов;
- узел преднатяжения троса.

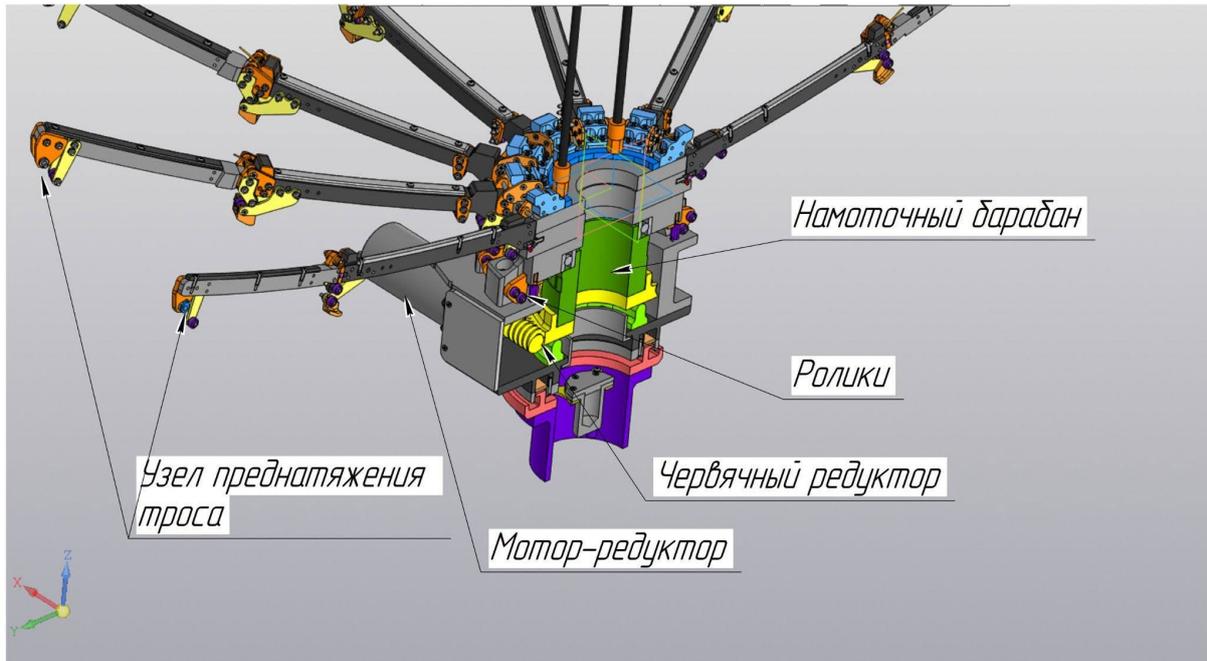


Рисунок 22 - Состав системы раскрытия

Трос прокладывается следующим образом: трос зафиксирован на винт, который через шайбу закручивается в намоточный барабан. Материал троса - плетёный полиэтиленовый шнур с допустимой нагрузкой на разрыв - 26 кгс. Затем трос через опорный ролик и систему роликов в ступице прокладывается на силовом луче с фиксацией на узел преднатяжения.

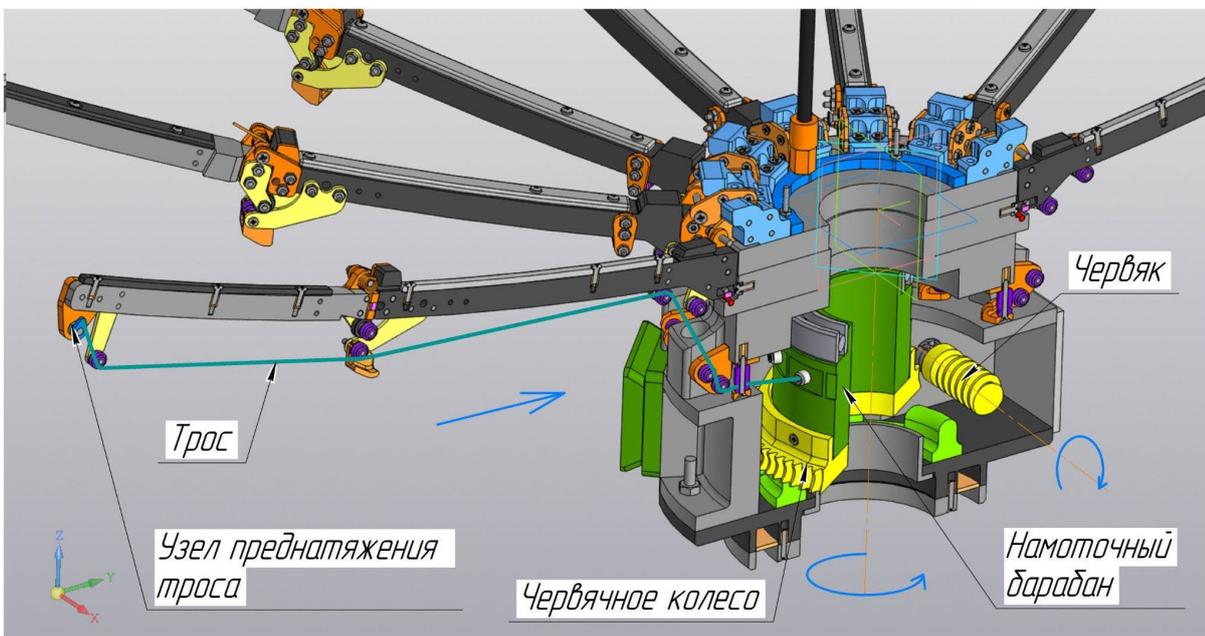


Рисунок 23 - Общая схема кинематики системы раскрытия

Благодаря наличию пружин кручения, установленных в шарнирных сочленениях обеспечивается раскрытие без рывков, а также складывание силовых лучей при движении вала мотор-редуктора в обратном направлении.

3.5.1 Выбор электропривода

Для приведения намоточного барабана в движение применяется электродвигатель постоянного тока. Для определения требуемого крутящего момента был произведён следующий расчёт.

Таблица 3 - Расчёт суммарного крутящего моменты, требуемого на раскрытие силовых лучей

Исходные данные	Расчёт
$F = 22$ Н - сила раскрытия силового луча. Данные измерений; $L = 35 \cdot 10^{-3}$ м - длина плеча, равная радиусу намоточного барабана; $K = 70$ - передаточное отношение червячного редуктора ⁴ ; $n = 12$ - количество силовых лучей; $M_m = 15$ кгс · см - крутящий момент мотора	1) $M_{\text{раскр}} = F \cdot L = 0,77$ Н · м - крутящий момент раскрытия силового луча; 2) $M_{\Sigma} = n \cdot M_{\text{раскр}} = 9,24$ Н · м - крутящий момент на раскрытие всех силовых лучей; 3) $M' = M_m \cdot K = 102,9$ Н · м - крутящий момент на червячном колесе; 4) $\frac{M_{\text{мотор}}}{M_{\Sigma}} = 11,15$ - отношение крутящего момента мотора, преобразованного редуктором к требуемому суммарному крутящему моменту на раскрытие силовых лучей.
Предмет расчёта	Результат
M_{Σ} - суммарный момент на раскрытие силовых лучей	$M_{\Sigma} = 9,24$ Н · м

⁴ Червячный редуктор проектировался исходя из геометрических ограничений корпуса НКСИ

Измерение усилия на раскрытие производилось при помощи динамометра. Трос натягивался по касательной к намоточному барабану в соответствии с рисунком 22, в плоскости узлов фиксации троса к намоточному барабану.

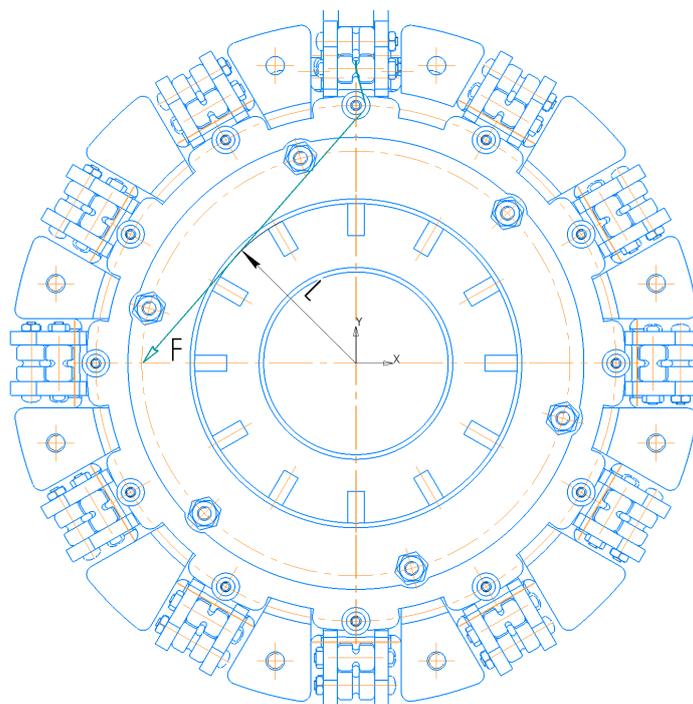


Рисунок 24 - Схема измерения усилия на раскрытие (трос обозначен зелёной утолщённой линией)

3.5.2 Движитель системы раскрытия

Намоточный барабан приводится в движение мотором-редуктором JGB37-550-S330[5], который был выбран, как удовлетворяющий требованию по создаваемому крутящему моменту. Для увеличения момента силы и снижения люфта между намоточным барабаном и мотором установлен червячный редуктор.

Намоточный барабан закреплён на нижнем диске ступицы через подшипник качения. В намоточном барабане находятся 12 отверстий для вплавляемых гаек, в которые устанавливаются винты с закреплёнными на них тросами.

На намоточный барабан посредством винтового крепления устанавливается червячное колесо. Совместно с червяком, закреплённым на валу мотор-редуктора оно образует червячный редуктор. Мотор редуктор крепится к корпусу НКСИ.

Параметры червячного редуктора:

Число витков червяка: 1

Число зубьев червячного колеса: 70

Модуль зубьев: 1,25 мм

Передаточное отношение редуктора: 70

Материал: PETG-пластик, PLA-пластик

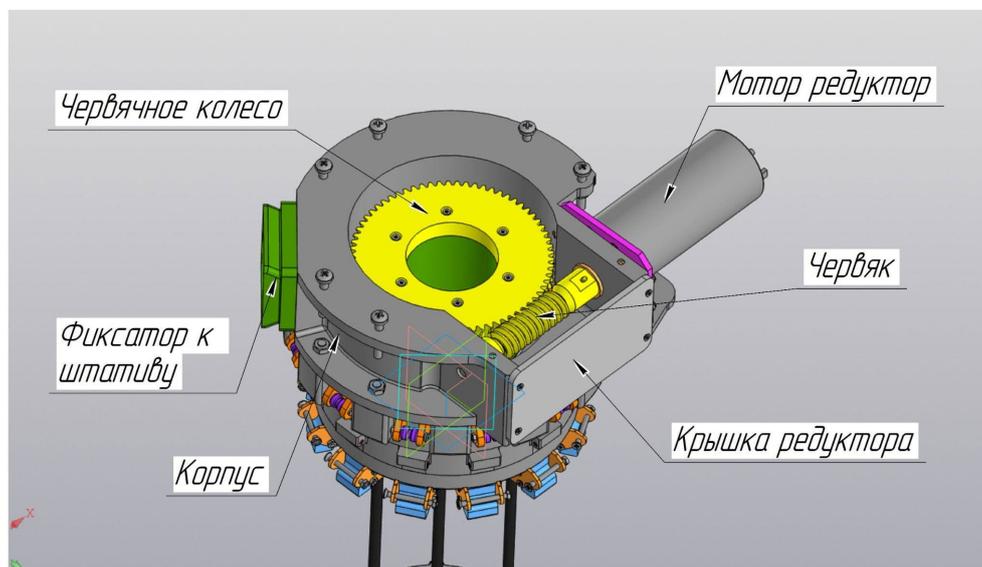


Рисунок 25 - Червячный редуктор привода раскрытия

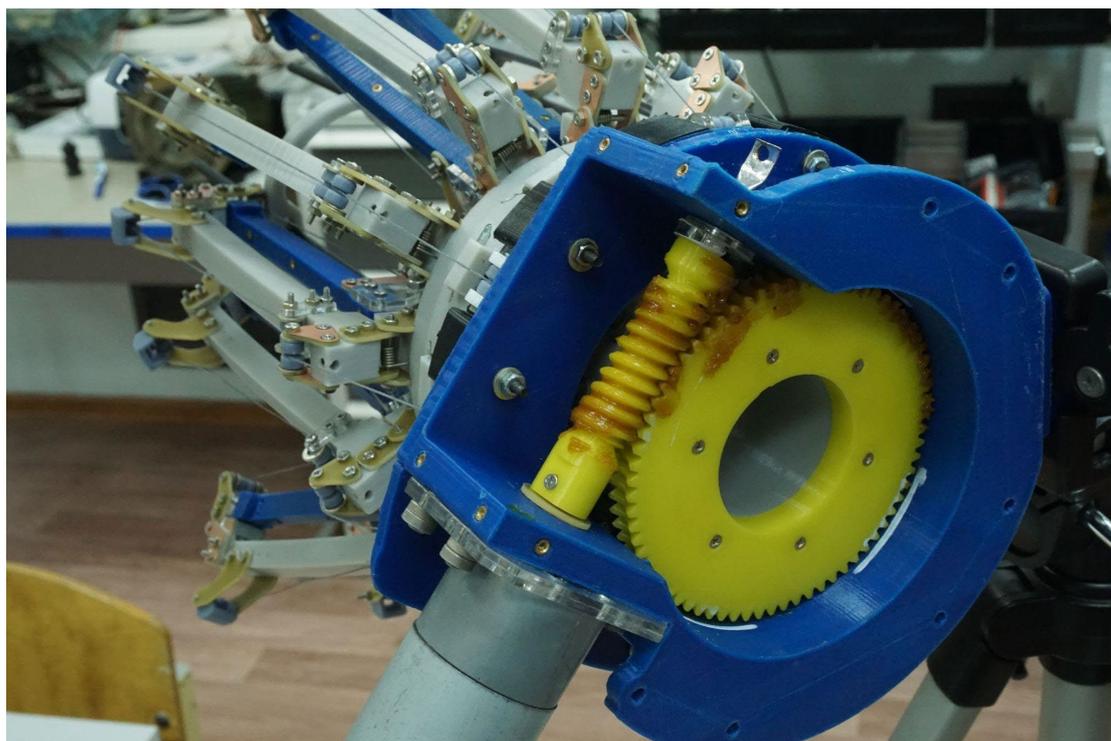


Рисунок 26 - Червячный редуктор привода раскрытия

4 Отражающее полотно

Мягкое отражающее полотно(ОП) в качестве отражающей поверхности было выбрано благодаря меньшему удельному весу и большей ремонтпригодности.

В данном проекте в качестве ОП используется алюминизированный с одной стороны майлар.

4.1 Юстировка рефлектора

Для обеспечения правильной геометрии отражающей поверхности необходимо провести процедуру юстировки силовых лучей рефлектора.

Юстировка заключается в натяжении троса в узле преднатяжения, установленном в верхней части второго колена. Узел преднатяжения представляет из себя две проушины, между которыми устанавливается винт. На винт закреплён эксцентриковый фиксатор, на котором закреплён трос. Намоткой троса регулируется длина части, проходящей через силовой луч, тем самым, меняя кривизну рефлектора.

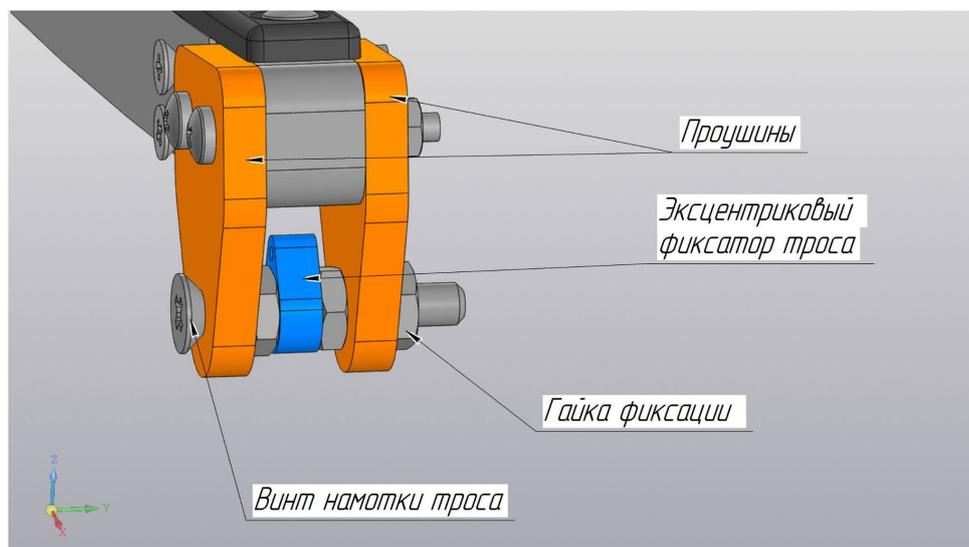


Рисунок 27 - Конструкция узла преднатяжения



Рисунок 28 - Конструкция узла преднатяжения на прототипе

Юстировка осуществляется с использованием юстировочного трафарета, который изготовлен из фанеры на лазерном станке.

С помощью трафарета регулируется степень закручивания регулировочного винта, соответственно, степень раскрытия первого колена. Также, с помощью трафарета регулируется натяжка троса, соответственно, степень раскрытия второго колена.

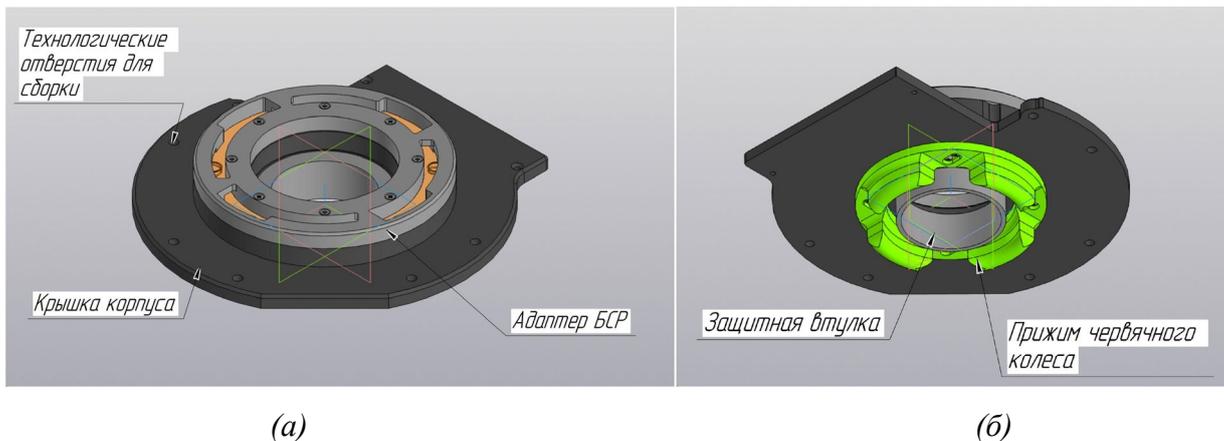
Проверка точности такой методики юстировки должна быть проверена формой фокального пятна, фокусируемого на мишени, которая устанавливается на месте контррефлектора.



Рисунок 29 - Юстировочный трафарет в процессе юстировки

5 Крышка корпуса и быстросъёмный разъём полезной нагрузки

Со внутренней стороны крышки корпуса закреплён прижим червячного колеса. С внешней стороны установлен адаптер быстросъёмного разъёма (БСР) полезной нагрузки (ПН).



а - Крышка корпуса, вид сверху-сбоку;

б - крышка корпуса, вид снизу-сбоку

Рисунок 30 - Крышка корпуса

5.1 Быстросъёмный разъём полезной нагрузки

Быстросъёмный разъём необходим для оперативной замены контейнера нагревательной камеры(НК) или смены типа ПН.

БСР представляет из себя байонетный разъём и состоит из трёх частей:

- подпружиненная проставка;
- байонет пассивный (адаптер БСР);
- байонет активный, установленная на контейнере нагревательной камеры.

Пассивный байонет посредством винтового крепление установлен на крышке корпуса. В пазы до упора вставляется активный байонет, после чего поворачивается на 90° и устанавливается в пазы под действием силы упругости пружин.

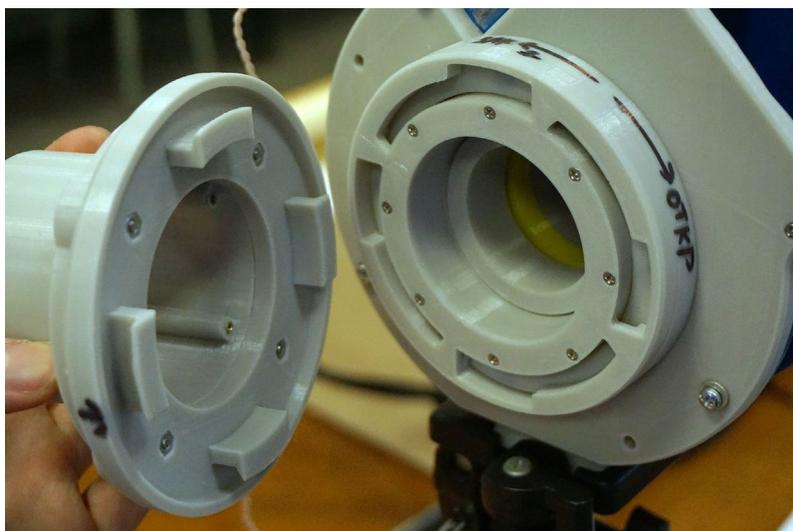


Рисунок 31 - Быстросъёмный разъём ПН. Активный байонет слева, адаптер БСР справа

6 Демонстратор полезной нагрузки. Нагревательная камера

НК представляет из себя алюминиевый полый цилиндр с фланцевым креплением крышки. НК была изготовлена на токарном станке.

Характеристики НК:

Материал: Д16Т

Масса: 19.5 гр

Объём ёмкости: 2.8 см³

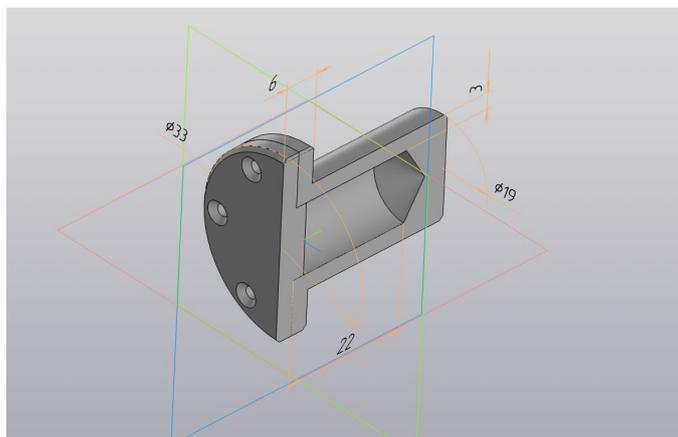


Рисунок 32 - НК в продольном сечении с указанием некоторых размеров



Рисунок 33 - Сборка НК с текстолитовой проставкой

НК устанавливается в контейнер, который представляет из себя полу цилиндрическую деталь, внутренние стенки которой покрыты отражающим материалом для минимизации тепловых потерь излучением в условиях космического полёта. Для минимизации тепловых потерь теплопроводностью НК устанавливается через проставку.

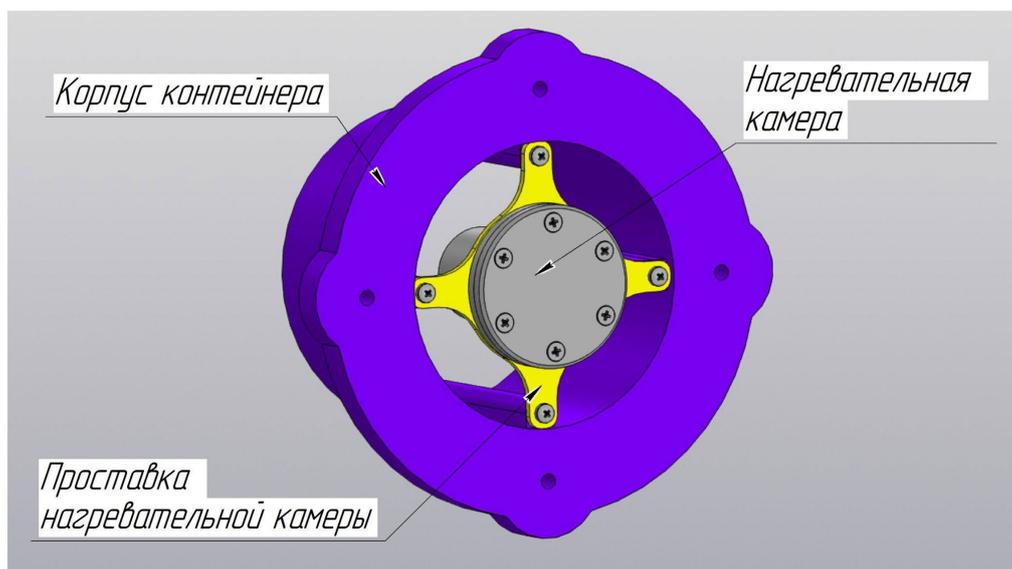


Рисунок 34 - Контейнер установки НК

К контейнеру нагревательной камеры крепится активный байонет для фиксации БСР.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы была достигнута цель по разработке и испытанию силового каркаса и системы раскрытия рефлектора. Выполнена теоретическая часть работы по проектированию оптической схемы и проведению расчётов характеристик НКСИ. Разработан, испытан и итерационно доработан прототип НКСИ с зонтичной системой раскрытия рефлектора.

Отработаны ключевые технические решения в конструкции НКСИ и в то же время, для дальнейшего улучшения проекта необходим качественно иной технологический подход, иначе не получится обеспечить точность деталей и узловых сборок, а значит и добиться требуемых оптических характеристик.

Задачами дальнейшей разработки является:

- разработка методики установки ОП, изготовление юстировочного оборудования ОП. Установка полотна и проверка юстировки рефлектора по оценке собирающей способности;
- разработка контррефлектора и нанесение отражающего покрытия;
- проведение испытаний по плавлению сплава Розе;
- проведение испытаний конструкции системы раскрытия и силового каркаса рефлектора, моделирующих влияние факторов космического полёта. В приоритете стоят испытания на вибростенде для проверки надёжности узла преднатяжения тросов; тепловакуумные испытания для оценки влияния градиента температур на юстировку конструкции рефлектора; испытания на инерционность для оценки параметров необходимой к разработке системы наведения и стабилизации.

Список используемой литературы

- [1] Colozza, Anthony & Macosko, Robert & Castle, Charles & Sacksteder, Kurt & Suzuki, Nantel & Mulherin, James. (2012). Cassegrain Solar Concentrator for ISRU Material Processing. 10.2514/6.2012-637.
- [2] Nakamura, Takashi & Sullivan, D. & McClanahan, J. & Shoji, J. & Partch, R. & Quinn, S.. (2004). Solar Thermal Propulsion for Small Spacecraft. 10.2514/6.2004-4138.
- [3] Harper, WB, Jr., Boyle, RV, & Kudija, CT. "Solar Dynamic CBC Power for Space Station Freedom." Proceedings of the ASME 1990 International Gas Turbine and Aeroengine Congress and Exposition. Volume 3: Coal, Biomass and Alternative Fuels; Combustion and Fuels; Oil and Gas Applications; Cycle Innovations. Brussels, Belgium. June 11–14, 1990. V003T08A002. ASME.

- [4] Thomson, M. W., Hodges, R. E., Chanat, N. E., Sauder, J., & Rahmat-Samii, Y. (2016, December 1). PARABOLIC DEPLOYABLE ANTENNA.
- [5] JGB37-550 High Torque DC Gear Motor // Hands on Tech URL: https://handsontec.com/dataspecs/motor_fan/JGB37-550-Geared%20Motor.pdf (дата обращения: 15.02.2025).
6. Гряник М.В., Ломан В.И. Развертываемые зеркальные антенны зонтичного типа. - М.: Радио и связь, 1987. - 72 с.

Приложение А

Маршрутная карта проекта

Таблица А.1 - Маршрутная карта проекта

		Первая итерация конструкции рефлектора			
Август 2024 г		Сентябрь-декабрь 2024 г		Январь 2025 г	
Выполняемые задачи	Результат	Выполняемые задачи	Результат	Выполняемые задачи	Результаты
<p>1.Сбор и анализ информации об аналогичных проектах и исследованиях по теме;</p> <p>2.составление технического задания на проект с формулированием требований к конструкции;</p> <p>3.проведение расчётов</p> <p>4.разработка предварительной оптической схемы в среде desmos.</p>	<p>1.Сформулированы требования к конструкции;</p> <p>2.создана ориентировочная конструктивно-компоновочная схема НКСИ;</p> <p>3.выбрана система раскрытия рефлектора;</p> <p>4.определены предварительные сроки реализации проекта и отмечены контрольные точки.</p>	<p>1. Проектирование ступицы;</p> <p>2. проектирование силовых лучей;</p> <p>3. проектирование узлов системы раскрытия рефлектора;</p> <p>4. закупка расходных материалов;</p> <p>5. отработка технических решений и методов работы(выбор материала троса, пробная 3D-печать деталей)</p>	<p>1. Разработана 3D-модель ступицы и изготовлены прототипы деталей;</p> <p>2. разработана 3D-модель силового каркаса рефлектора; изготовлены детали и собран прототип силового луча;</p> <p>3. разработана 3D-модель системы раскрытия, изготовлен первый прототип;</p> <p>4. проведены испытания кинематической схемы раскрытия силового луча (9 ноября)</p> <p>6. проведены комплексные испытания системы раскрытия рефлектора (29 декабря)</p>	<p>1. Разработка и изготовление крышки корпуса редуктора с быстросъёмным разъёмом(БСР) блока ПН;</p> <p>2. разработка и изготовление нагревательной камеры, штанги контррефлектора, стакана контррефлектора, блока фоторезисторов наведения на Солнце;</p> <p>3. Проведение испытаний системы раскрытия со снятыми пружинами кручения в сочленении первого и второго колена силового луча;</p> <p>4. Первые попытки установки отражающей поверхности на рефлектор</p>	<p>1. Из сплава Д16Т выточена нагревательная камера;</p> <p>2. Изготовлены и собраны штанга контррефлектора с мишенями для последующей юстировки рефлектора;</p> <p>3. Изготовлен быстросъёмный пружинный разъём контейнера НК, по результатам отработки перенесён на доработку;</p> <p>4. По результатам испытаний системы раскрытия выявлена необходимость доработки.</p>
Вторая итерация конструкции рефлектора					

Февраль 2025	Результаты	Март 2025	Результаты		
<p>1. Проектирование изменённой конструкции крепления силовых лучей к ступице;</p> <p>2. перепроектирование деталей червячного редуктора;</p> <p>3. изменение трассы троса системы раскрытия;</p> <p>4. изменение конструкции первого колена;</p> <p>5. разработка транспортировочного короба</p>	<p>1. Спроектированы, изготовлены и протестированы кронштейны силовых лучей;</p> <p>2. перепроектирован и изготовлен новый червячный редуктор, что показало улучшение в механике раскрытия;</p> <p>3. перепроектирована кинематическая схема силового луча;</p> <p>4. перемоделировано и изготовлено первое колено новой конструкции;</p> <p>5. разработан транспортировочный короб.</p>	<p>1. Разработка автоматической системы раскрытия рефлектора;</p> <p>2. проведение комплексных испытаний с новой конструкцией силового каркаса рефлектора и изменённой системой раскрытия;</p> <p>3. сборка транспортировочного короба;</p>	<p>1. Собран транспортировочный короб;</p> <p>2. Подготовлена расчётно-пояснительная записка;</p> <p>3. Проведена юстировка рефлектора и оценена изменённая конструкция рефлектора.</p>		

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ИЗУЧЕНИЯ ЖЕСТОВОГО ЯЗЫКА НА БЕСПЛАТНОЙ ОСНОВЕ

Любушин Владимир Михайлович

Москва г, ГБОУ Школа №1498, 11 класс

Секция: Компьютерные системы

Научный руководитель: Минитаева Алина Мажитовна, к.н., доцент каф. ИУБ
МГТУ им. Н.Э. Баумана

Аннотация

В рамках работы был разработан сайт, с множеством уроков для изучения РЖЯ, с функцией проверки пройденных тестов для закрепления информации, а также с удобным интерфейсом для всех возрастных групп.

Для написания сайта был использован язык PHP [1] и фреймворк Laravel [2], а также bootstrap для стилизации, система управления базами данных MySQL [3]. Используются языки программирования и разметки: JavaScript [4], HTML и CSS. Веб-приложение реализовано в среде разработки Visual Studio Code.

Выполнен анализ предметной области и анализ существующих аналогов разрабатываемого приложения. Выявлены преимущества и недостатки рассмотренных приложений.

Создана технология тестирования веб-приложения, которая охватывает оценочное и функциональное тестирование. Проведено тестирование в соответствии с данной технологией. Процесс разработки системы сопровождается графическими материалами и основные принципы функционирования приложения

Введение

В наше время существует относительно небольшое количество ресурсов, где можно изучить русский жестовый язык бесплатно, хотя многие люди, не только слабослышащие и глухие, хотят его выучить, для того чтобы понимать язык, в случае необходимости.

Цель: Разработка веб-приложения «Sign Language» направлена на создание платформы для пользователей, желающих освоить жестовый язык. Данное приложение будет способствовать изучению жестового языка и предоставит возможность пользователям проверять свои знания через специальные тесты.

Задачи: В данном проекте, для достижения цели, были поставлены следующие задачи:

- анализ жестов русского жестового языка;
- анализ приложений для изучения жестового языка, определение их недостатков и преимуществ;
- выбор средств разработки;
- разработка и реализация интерфейса веб-приложения;
- проектирование базы данных;
- реализация веб-приложения с использованием выбранных средств;
- разработка технологии тестирования;
- тестирование веб-приложения в соответствии с разработанной технологией.

Объект исследования - русский жестовый язык.

Предмет исследования - приложения для изучения русского жестового языка.

Актуальность: Данное приложение актуально, так как, по данным на июнь 2024 года, в России более 13 миллионов людей с нарушением слуха, в том числе более 1 миллиона из них - дети. Соответственно в каждой такой семье родным и близким важно знать РЖЯ, для того чтобы поддерживать контакт со всеми членами семьи.

Практическая значимость: В нынешних условиях изучение любых теоретических основ требует современных подходов в виде веб-приложений, которые можно использовать как на компьютерах, так и на телефонах или планшетах. Разрабатываемым веб-приложением будет обеспечена практическая значимость для изучения русского жестового языка.

Научная новизна проекта заключается в проведенном анализе предметной области и анализе существующих аналогов разрабатываемого приложения.

1. Анализ предметной области и сравнительный обзор аналогов

1.1 Анализ предметной области

Язык жестов – это самостоятельная лингвистическая система, которая состоит из жестов, со своей собственной грамматикой и лексикой, которую используют слабослышащие и глухие люди по всему миру, без надобности передавать информацию голосом.

Русский Жестовый Язык (РЖЯ) – это язык, используемый для глухих и слабослышащих людей. Впервые язык был открыт в 1806 году, при открытии первого училища для глухих в Санкт-Петербурге. Грамматически РЖЯ отличается от обычного разговорного русского языка очень сильно, так как в русском языке присутствуют сложные конструкции, которые трудно перенести на язык жестов.

1.2 Национальные жестовые языки других стран

Международный Язык жестов (Жестуно) - всемирная федерация глухих, появившаяся в 1951 году, создала официальный язык для глухих - International Sign Language (ISL). Чаще всего данный язык используют на каких-либо международных мероприятиях для того, чтобы глухие люди тоже понимали, о чем идет речь, т.к. много неслышащих специалистов присутствуют на множестве конгрессов и конференций.

Американский язык жестов (Амслен) - данный язык, American Sign Language (ASL), используется не только в США, но и в других странах, например в англоязычной Канаде, а также во множестве стран на западе Африки и на юге Азии. Амслен, как и РЖЯ произошёл от французского жестового языка.

Французский язык жестов - считается родственным для многих жестовых языков, таких как: РЖЯ, амслен, нидерландский, ирландский и многих других. История данного языка очень интересна, т.к. основоположник языка основал первые школы для изучения языка жестов, а в дальнейшем популяризировал идея о том, что глухие люди способны обучаться.

1.3 Образовательные программные системы

В интернете есть множество платных сервисов и школ для изучения жестового языка и процесс обучения займет долгие месяцы. Существует небольшое количество ресурсов, где можно выучить жестовый язык бесплатно.

1.4 Сравнительный обзор веб-приложений для изучения жестового языка

Для начала рассмотрим отечественный сайт <https://puzzlesignlanguage.ru> и зарубежный сайт <https://lifeprint.com>.

Оба сайта обучают жестовому языку, один РЖЯ, другой ASL "American sign language", что в переводе означает Американский жестовый язык. Изучив дизайны сайт, можно заметить, что интерфейс зарубежного сайта сильно отличается от отечественного. Он сложный для понимания, неинтуитивный, а также неприятен глазу. В отечественном источнике приятный интерфейс, будет понятен абсолютно всем, присутствуют анимации и не трудная навигация по сайту.

Что касается информационной части, отечественный сайт имеет структурированную форму обучения, состоящую из множества лекций, материал можно закреплять играми, а также есть отдельный раздел со словарём, где можно выбрать и изучить конкретное слово, но их не так много. В зарубежном сайте тоже присутствуют уроки, но реализовано все не лучшим образом, нет никакого дизайна страницы, все написано сплошным текстом практически без каких-либо картинок. Из достоинств стоит отметить, что присутствует словарь с огромным количеством слов, для каждого есть пояснения и в редких случаях есть видео с демонстрацией конкретного слова носителем.

В итоге моего исследования, касательно других сайтов для изучения жестовых языков, я выявил что, оба ресурса предлагают разные преимущества. Один обладает понятным и приятным интерфейсом, с хорошо оформленным подходом к обучению. Другой сайт, несмотря на имеющийся огромный словарь и наличие некоторых видеоматериалов, не обладает должным дизайном, что намного усложняет обучение.

	Puzzlesignlanguage	Lifeprint	Sign Language (разрабатываемое веб-приложение)
Изучение жестов	+	+	+
Отслеживание пройденных уроков	+	-	+
Есть видеоматериалы для обучения	+	+	+
Личный профиль с возможностью смены пароля	-	-	+
Тестирования	+	-	+

1.5 Выводы

В рамках данного проекта была поставлена цель разработки веб-приложения для изучения русского жестового языка (РЖЯ). Это приложение направлено на создание платформы, которая будет доступна для всех желающих освоить жестовый язык, предоставляя возможность не только изучать его, но и проверять свои знания с помощью тестов. В процессе работы над проектом были поставлены задачи, такие как анализ жестов РЖЯ, исследование существующих образовательных приложений для изучения жестового языка, выбор инструментов разработки, создание интерфейса веб-приложения, проектирование базы данных и реализация самого приложения. Также была разработана технология тестирования, с помощью которой проводились проверки функциональности веб-приложения.

Особое внимание было уделено анализу существующих ресурсов и образовательных систем для изучения жестового языка, выявлению их достоинств и недостатков. Это исследование позволило понять, какие функции и особенности важны для пользователей, а какие могут затруднять процесс обучения.

Это приложение может стать важным инструментом для людей, желающих овладеть жестовым языком, а также для семей, в которых есть люди с нарушениями слуха. Практическая значимость приложения заключается в его доступности, так как оно будет работать на различных устройствах, включая компьютеры, телефоны и планшеты, что делает обучение гибким и удобным.

2. Разработка веб-приложения для изучения жестового языка

2.1 Разработка диаграммы состояний интерфейса

Пользователю нужно предоставить интуитивно понятный и адаптивный интерфейс, чтобы при использовании веб-приложения не возникало сложностей на любых компьютерах. При взаимодействии с веб-приложением пользователь может переключать интерфейс между несколькими состояниями [5], которые показаны на рисунке 1.

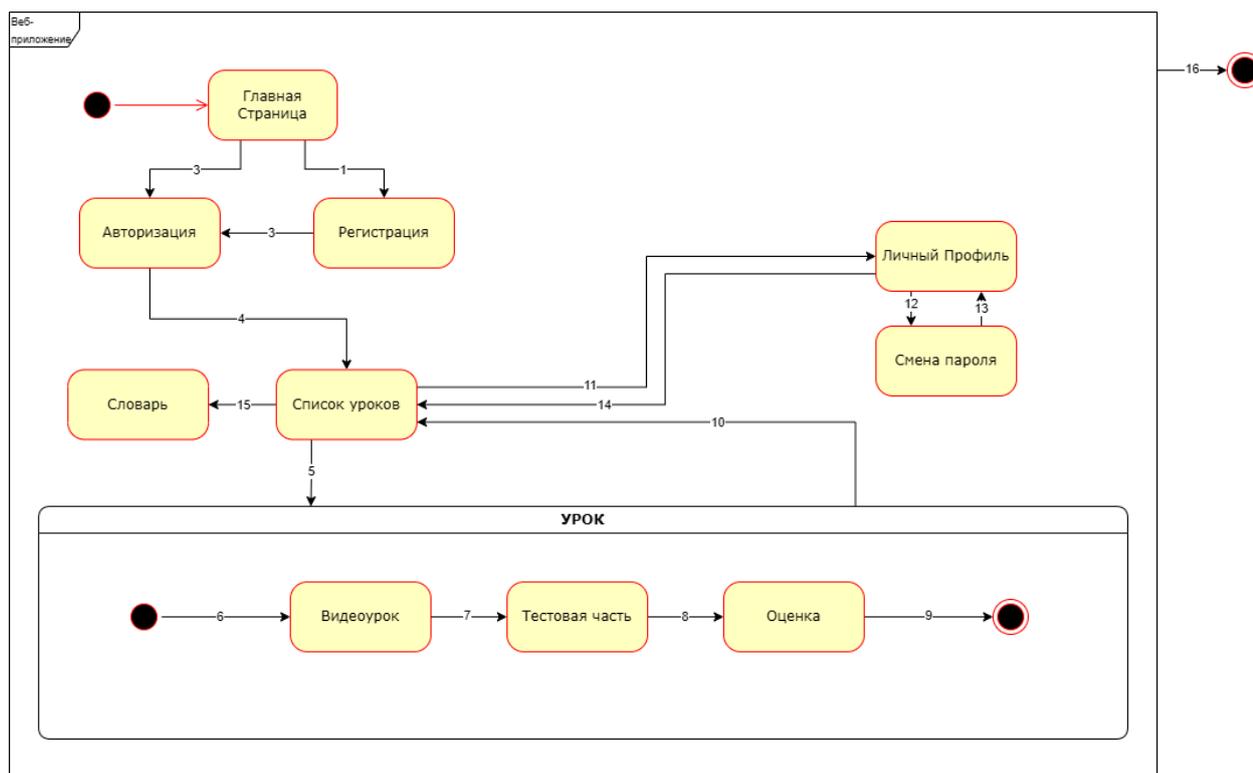


Рисунок 1 - Граф состояний интерфейса

После входа на сайт интерфейс находится в состоянии «Главная страница», где отображается приветственная страница сайта. После нажатия на кнопку «Регистрация» можно начать регистрацию, тогда открывается форма регистрации. Если пользователь зарегистрирован, он нажимает кнопку «Авторизация» и вводит свои данные в форму. После прохождения авторизации откроется основная страница с выбором уроков.

Действия, обозначенные на диаграмме состояний интерфейсов цифрами:

- 1) нажатие на кнопку «Регистрация»;
- 2) нажатие на кнопку «Авторизация»;
- 3) вход на страницу регистрации;
- 4) нажатие на кнопку «Вход»;
- 5) нажатие на кнопку «Перейти к уроку»;
- 6) вход в урок;
- 7) прохождение тестирования;
- 8) нажатие на кнопку «Проверить»;
- 9) выход из урока;
- 10) нажатие на кнопку «К списку уроков»;
- 11) нажатие на кнопку «Личный профиль»;

- 12) нажатие на кнопку «Изменить пароль»;
- 13) нажатие на кнопку «Сменить пароль»;
- 14) нажатие на кнопку «Список уроков»;
- 15) нажатие на кнопку «Словарь»
- 16) выход из веб-приложения (закрытие браузера);

Выполнение действий пользователя и отклик системы представлены в Таблице 1.

Таблица 1 - Выполнение действий

Действия исполнителя	Отклик системы
1. Пользователь нажимает на кнопку регистрации	2. Система регистрирует по переход по кнопке и перенаправляет на страницу
3. Пользователь после регистрации попадает на страницу авторизации	4. Система отправляет данные пользователя в БД и перенаправляет на страницу
5. Пользователь выбирает один из уроков	6. Система регистрирует по переход по кнопке и отрисовывает шаблон урока
7. Пользователь инициирует процесс проверки теста	8. Система проверяет данный ответ по заранее заданным данным и выводит ответы
9. Пользователь нажимает на кнопку сменить пароль в профиле	10. Система меняет данные из БД на новые, которые ввёл пользователь

2.2 Разработка форм пользовательского интерфейса

Для каждого состояния интерфейса необходимо разработать формы ввода и вывода информации. Следовательно, следующим этапом является проектирование интерфейсных форм, необходимо разработать адаптивный интерфейс, который будет удобно использовать как на десктопных, так и на мобильных устройствах. Разработанные формы интерфейса для десктопной и мобильной версий представлены на рисунках 2-10.

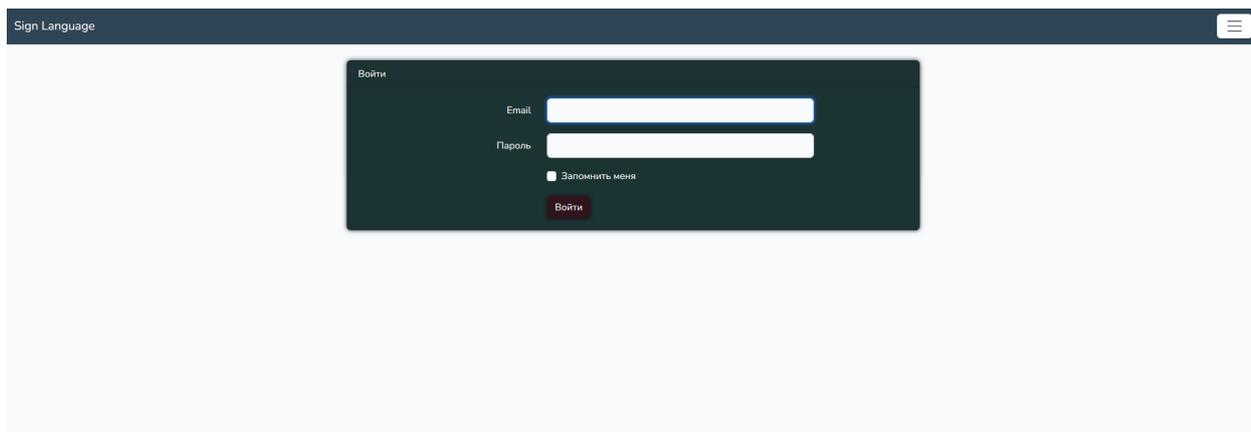


Рисунок 2 - Страница авторизации (desktopная версия)

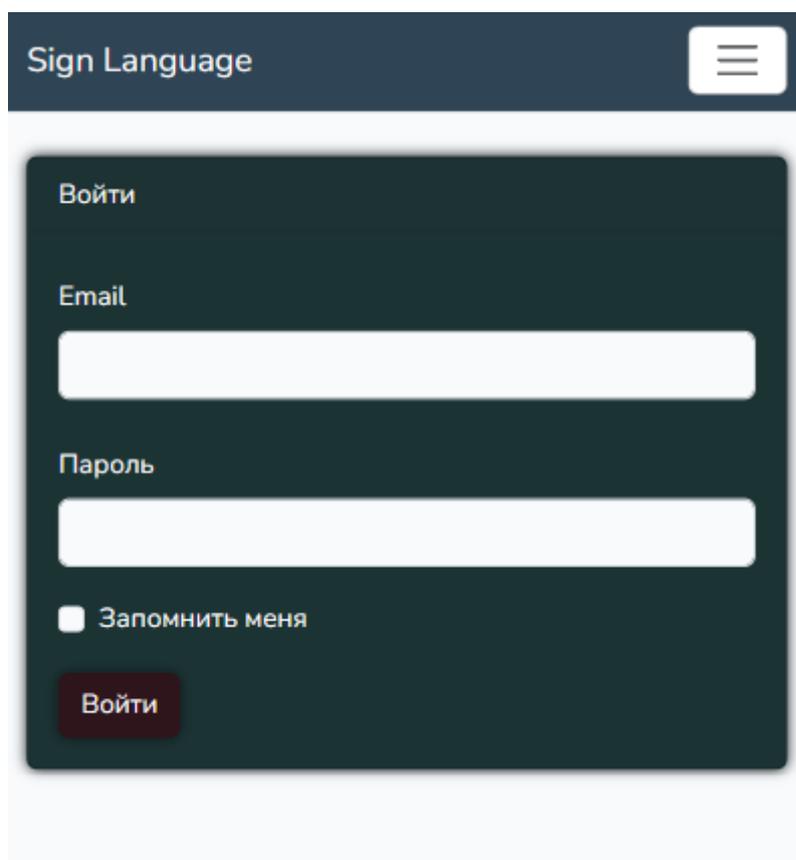


Рисунок 3 - Страница авторизации (мобильная версия)

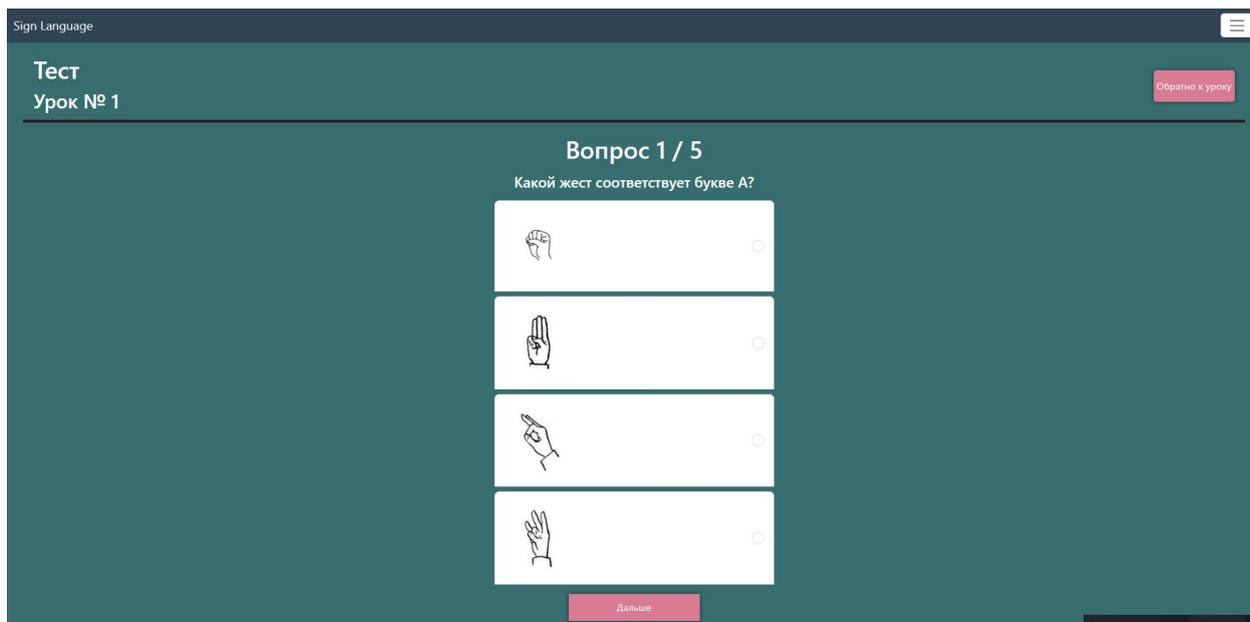


Рисунок 4 - Страница тестирования (desktopная версия)

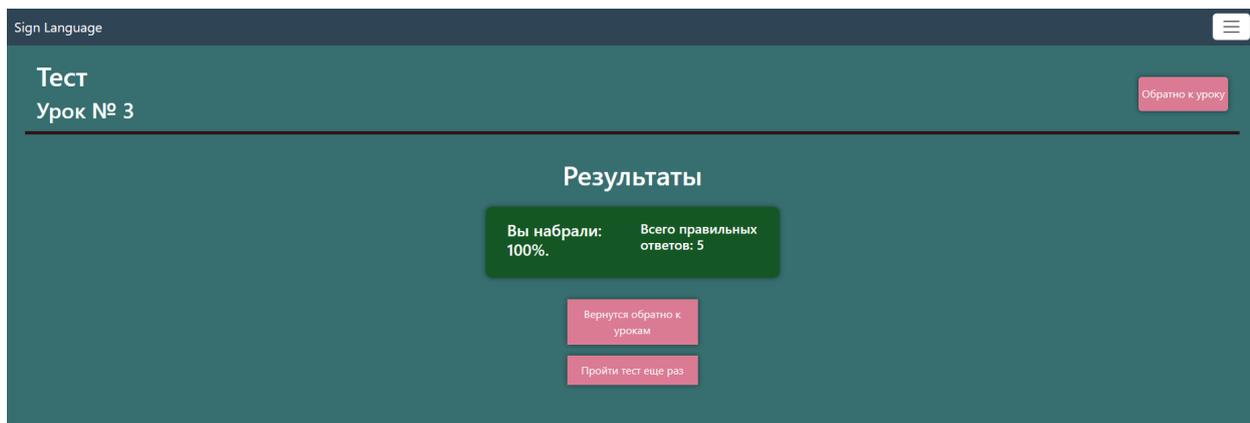


Рисунок 5 - Страница результатов тестирования (desktopная версия)

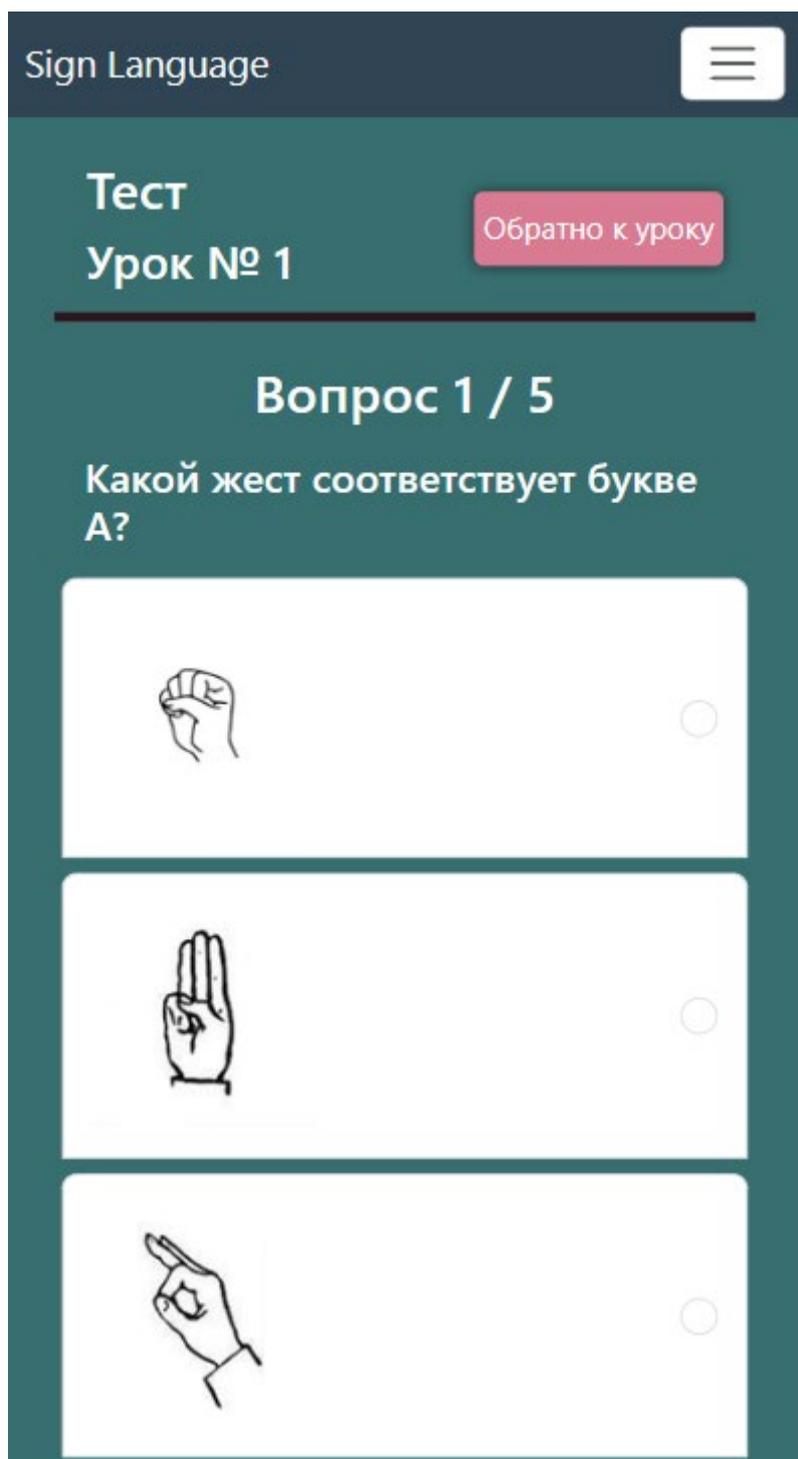


Рисунок 5 - Страница тестирования (мобильная версия)



Рисунок 6 - Страница результатов тестирования (мобильная версия)

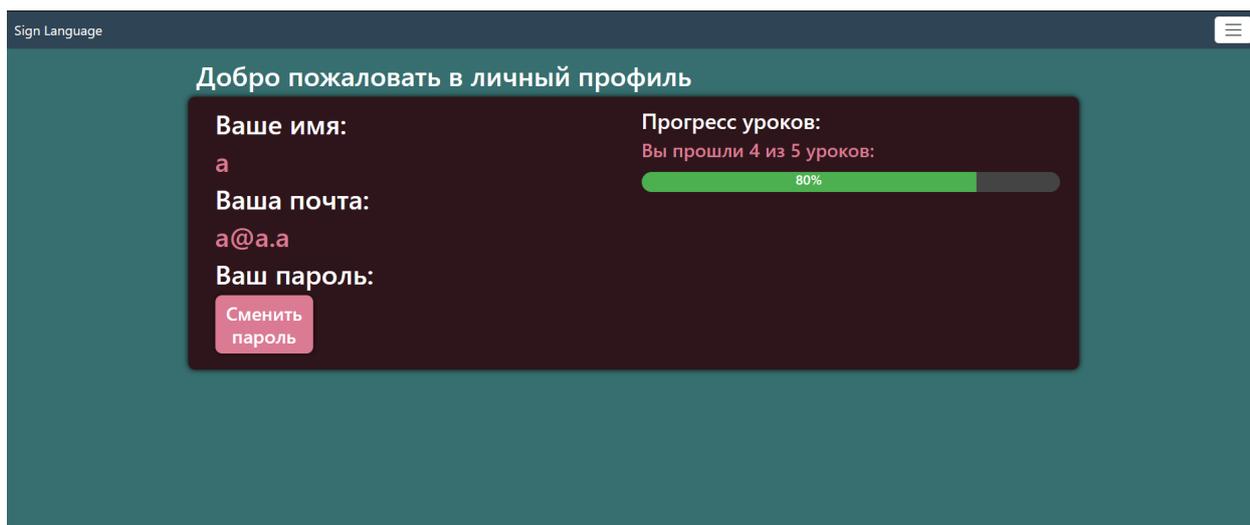


Рисунок 7 - Страница личного профиля (desktopная версия)

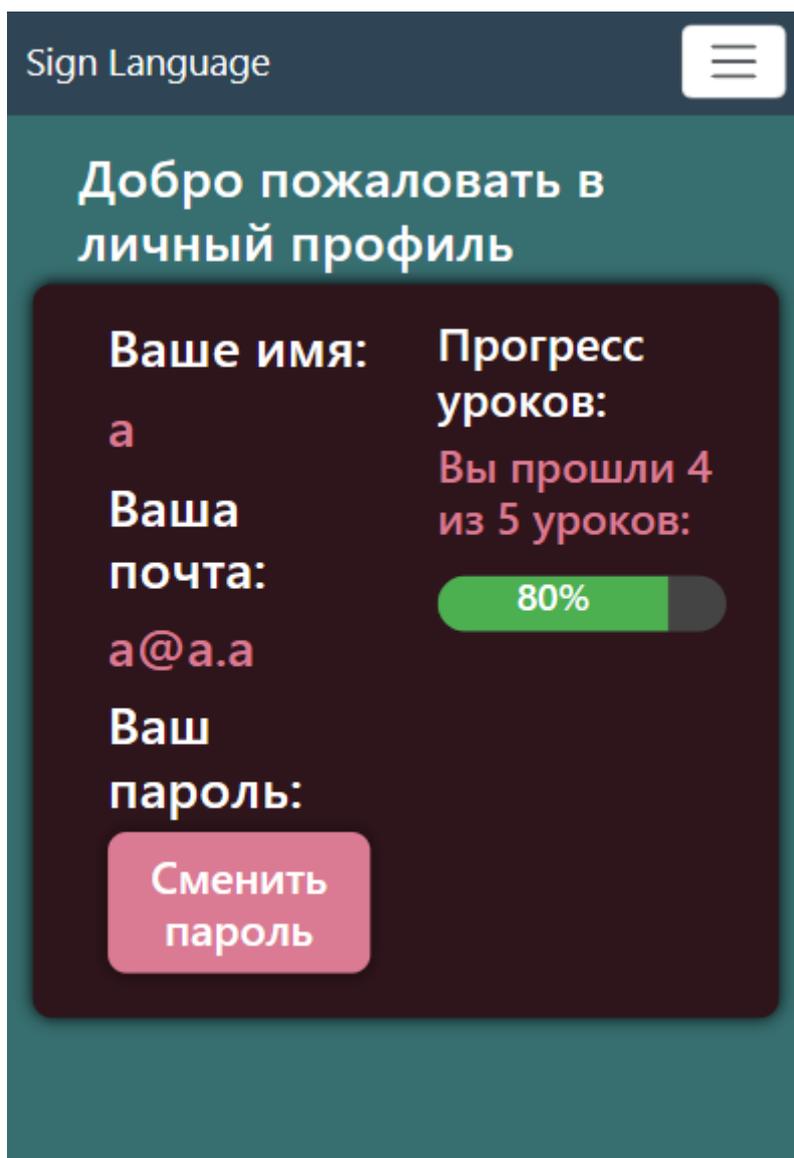


Рисунок 8 - Страница личного профиля (мобильная версия)

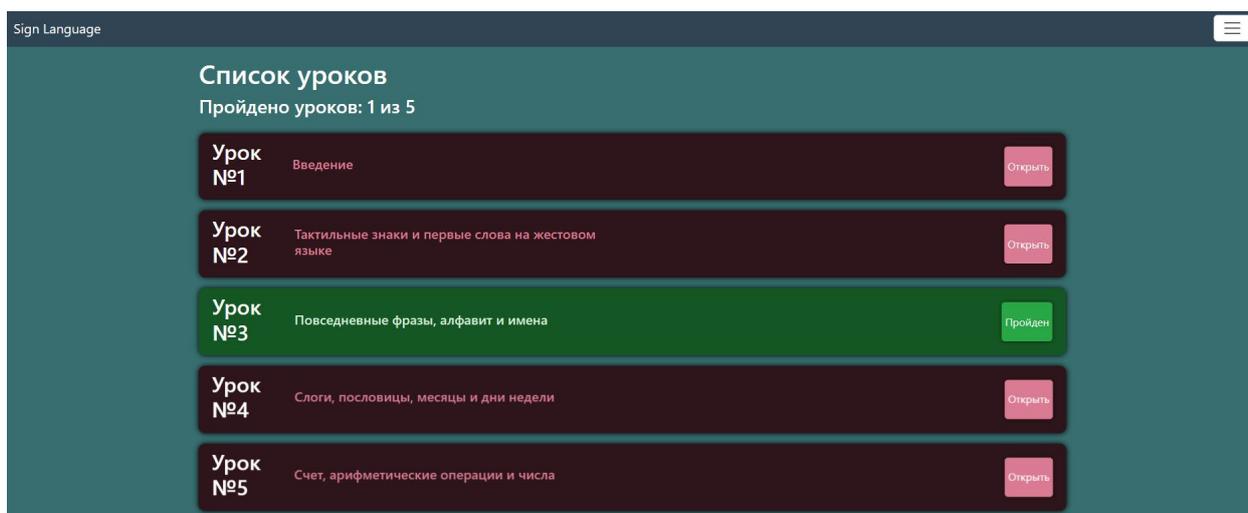


Рисунок 9 - Страница с уроками (десктопная версия)



Рисунок 10 - Страница с уроками (мобильная версия)

2.3 Диаграмма вариантов использования

Действующим лицом системы является Пользователь сайта. Пользователю доступны просмотр уроков и профиля, для незарегистрированных пользователей доступен только просмотр главной страницы[5].

Полученная диаграмма вариантов использования отражает типичное взаимодействие пользователя с разрабатываемым программным обеспечением. Анализ вариантов использования показывает, что вариант смены пароля расширяет вариант просмотра личного профиля, в то время как авторизация расширяет вариант регистрации.

На основе составленных требований были составлены варианты использования (таблицы 2 – 8).

Таблица 2 - Вариант использования "Регистрация"

Название варианта	Регистрация
Цель	Регистрация в системе для дальнейшей авторизации
Действующие лица	Незарегистрированный пользователь
Краткое описание	Пользователь осуществляет регистрацию путем ввода имени, электронной почты и пароля. Форма отправляется на сервер и сохраняется в базе данных
Альтернативный вариант	Если система после проверки выявляет некорректность данных, то она выведет ошибку ввода данных

Таблица 3 - Вариант использования "Авторизация"

Название варианта	Авторизация
Цель	Авторизация в системе необходима для начала обучения
Действующие лица	Неавторизированный пользователь
Краткое описание	Пользователь авторизируется путем ввода электронной почты и пароля. Система проверяет данные на идентичность с базой данных и допускает пользователя к приложению
Альтернативный вариант	Если система определяет, что данные различаются, выводится ошибка о неверности ввода пароля или почты

Таблица 4 - Вариант использования "Выбор урока"

Название варианта	Выбор урока
Цель	Прохождение уроков осуществляется путем прохождения уроков для изучения жестового языка
Действующие лица	Авторизированный пользователь
Краткое описание	Пользователь выбирает урок и начинает просмотр представленного видеоматериала на основе темы урока. Далее пользователь проходит тестирование и видит свою оценку. После пройденного урока, он отмечается пройденным

Таблица 5 - Вариант использования "Прохождение тестовой части"

Название варианта	Прохождение тестовой части
Цель	Прохождение тестирования для закрепления полученных знаний
Действующие лица	Авторизированный пользователь
Краткое описание	Пользователь приступает к тестированию, система выводит тестовое задание состоящие из нескольких вопросов, у каждого вопроса есть 4 варианта ответа и только один из них правильный. После ответов на вопрос пользователь нажимает кнопку, и система начинает проверку отмеченных ответов на соответствие с записями в базе данных. Далее система отображает количество правильных ответов.

Таблица 6 - Вариант использования "Просмотр личного профиля"

Название варианта	Просмотр личного профиля
Цель	Просмотр личных данных таких как – электронная почта, имя и пароля
Действующие лица	Авторизированный пользователь
Краткое описание	Приложение отображает пользователю его электронную почту, имя, пароль и количество пройденных уроков.

Таблица 7 - Вариант использования "Просмотр личного профиля"

Название варианта	Просмотр словаря
Цель	Просмотр слов, с целью изучения или повторения
Действующие лица	Авторизированный пользователь
Краткое описание	Приложение отображает все слова, которые использовались

Таблица 8 - Вариант использования "Изменение пароля"

Название варианта	Изменение пароля
Цель	Изменение пароля пользователя в случае его желания
Действующие лица	Авторизированный пользователь
Краткое описание	Система изменяет запись пароля в базе данных в соответствии с введенными данными. Далее система обновляет страницу и выводит результат изменения пароля пользователя.

На основе вариантов использования была составлена диаграмма прецедентов. Диаграмма прецедентов представлена на рисунке 11.

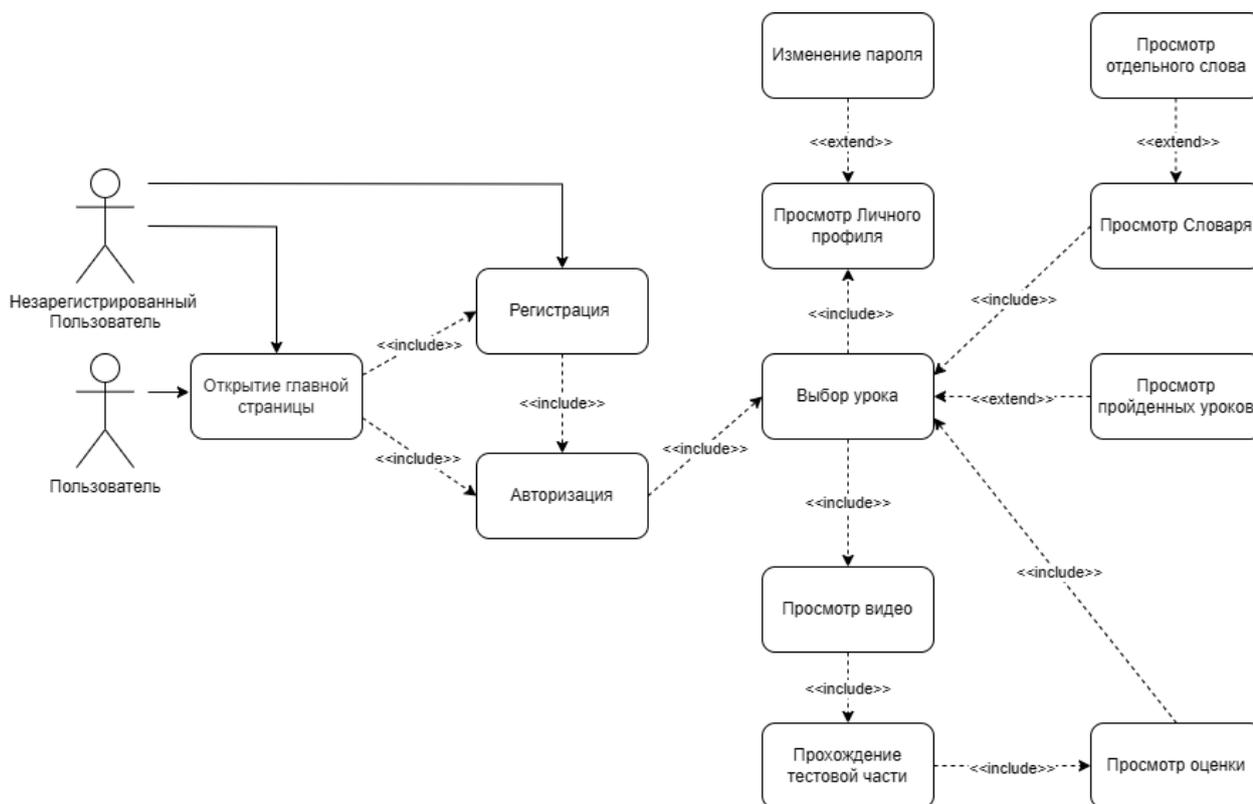


Рисунок 11 - Диаграмма вариантов использования

2.4 Проектирование базы данных

Для работы с базами данных я буду использовать Eloquent в Laravel, который реализует шаблон ActiveRecord. При использовании Eloquent каждая таблица БД имеет соответствующую Модель, которая используется для взаимодействия с этой таблицей. Помимо получения записей из таблицы БД, модели Eloquent также позволяют вставлять, обновлять и удалять записи из таблицы.

AR (ActiveRecord) работает следующим образом: каждой таблице соответствует свой класс, экземпляр этого класса представляет собой строку в таблице, а его атрибуты соответствуют колонкам. При создании нового экземпляра класса в таблицу добавляется новая запись. При чтении полей объекта считываются соответствующие значения записи таблицы баз данных. При изменении или удалении какого-либо объекта изменяется соответствующая ему запись.

Были определены 5 моделей. «Lessons» для информации об уроках, «questions» для хранения вопросов к каждому уроку, «lessonsquestions» для связи id из таблиц «lessons» и

«questions», «answers» для хранения ответов на определенных вопросов и модель «users» для хранения информации о пользователях.

Первичные ключи создаются автоматически с именем «id» и инкрементируются при добавлении новых для того, чтобы они были уникальными.

Рассмотрим связи, которые используются между таблицами в имеющейся базе данных. Между таблицами «answers» и «questions» имеется связь один ко многим, то есть у теста есть множество вариантов ответов, как правильных, так и неправильных, поэтому для этого в таблице «answers» создан столбец «questions_id», который берет id из таблицы «questions».

Тип связи «многие ко многим» используется между таблицами «lessons» и «questions» для связи столбцов id этих двух таблиц была создана дополнительная таблица «lessonsquestions», в которой есть три столбца – уникальный «id» и два «lessons_id», «questions_id» который берет id из двух исходных таблиц.

Также, один пользователь может завершить множество уроков, что фиксируется в таблице «completed_lessons», поэтому между таблицами «users» и «completed_lessons» используется тип связи «один ко многим».

Итоговая схема базы данных представлена на рисунке 12.

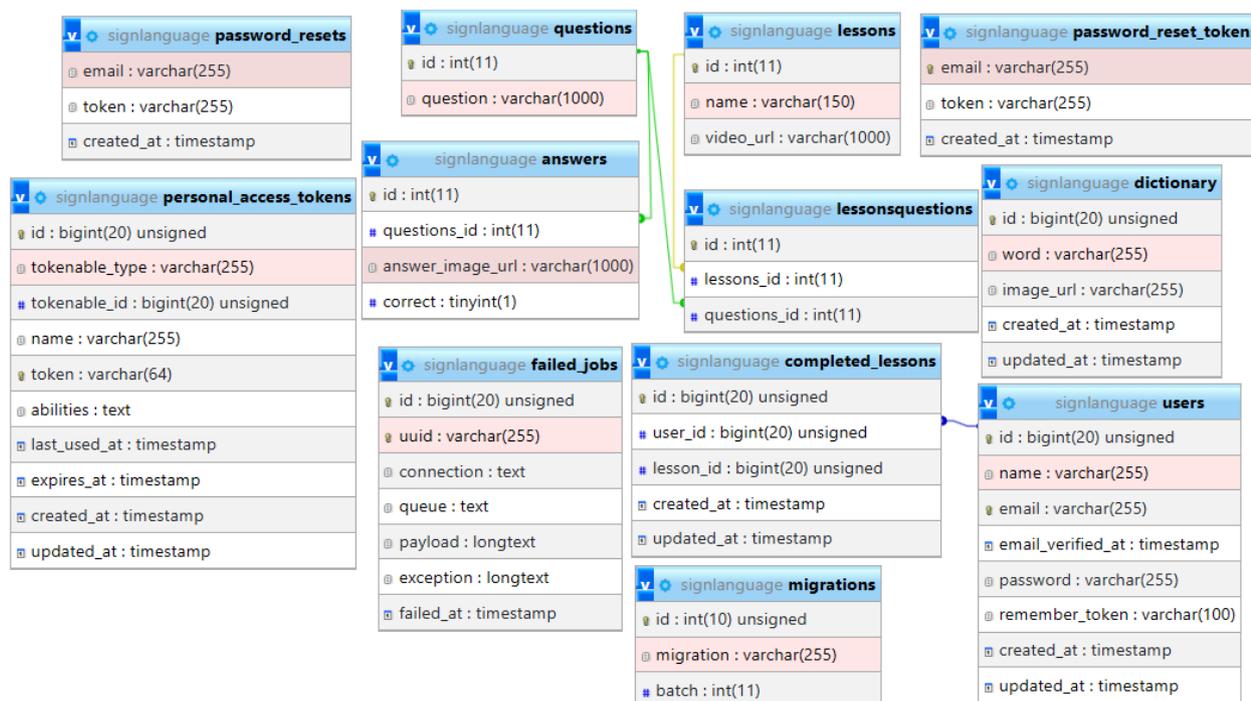


Рисунок 12 – Схема базы данных

Ниже в таблицах 9-15 приведены значения для атрибутов таблиц в спроектированной базе данных.

Таблица 9 - Таблица "lessons"

Название поля	Тип	Назначение
id (Primary key)	int(11)	Первичный ключ
name	varchar(150)	Название урока
video_url	varchar(1000)	Ссылка на видеоурок

Таблица 10 - Таблица "questions"

Название поля	Тип	Назначение
id (Primary key)	int(11)	Первичный ключ
question	varchar(1000)	Текст вопроса

Таблица 11 - Таблица "answers"

Название поля	Тип	Назначение
id (Primary key)	int(11)	Первичный ключ
questions_id	int(11)	Внешний ключ - вопрос

answer_image_url	varchar(1000)	Ссылка на изображение варианта ответа
correct	tinyint(1)	Правильность варианта ответа

Таблица 12 - Таблица "lessonsquestions"

Название поля	Тип	Назначение
id (Primary key)	int(11)	Первичный ключ
questions_id	int(11)	Внешний ключ - вопрос
lessons_id	int(11)	Внешний ключ - урок

Таблица 13 - Таблица "users"

Название поля	Тип	Назначение
id (Primary key)	bigint(20)	Первичный ключ
name	varchar(250)	Имя пользователя
email	varchar(250)	Электронная почта
password	varchar(1000)	Хэш Пароля

email_verified_at	timestamp	Дата подтверждения почты
remember_token	varchar(100)	Защита для авторизации
created_at	timestamp	Дата создания записи

Таблица 14 - Таблица "completed_lessons"

Название поля	Тип	Назначение
id (Primary key)	bigint(20)	Первичный ключ
name	varchar(250)	Имя пользователя
user_id	bigint(20)	Внешний ключ - пользователь
lesson_id	bigint(20)	Внешний ключ - урок
created_at	timestamp	Дата создания записи
updated_at	timestamp	Дата обновления записи

Таблица 15 - Таблица "dictionary"

Название поля	Тип	Назначение
id (Primary key)	int(11)	Первичный ключ

word	varchar(250)	Текст вопроса
image_url	varchar(250)	Ссылка на изображение варианта ответа
created_at	timestamp	Дата создания записи
updated_at	timestamp	Дата обновления записи

2.5 Диаграмма классов

Для классов проекта была составлена UML диаграмма, были показаны классы моделей, контроллеров, их атрибуты и связи между ними.

На диаграмме представлены такие связи как `hasMany` и `belongsTo`. Рассмотрим пример связи между классом `answer` и `question`. Между ними связь `belongsTo`, то есть для каждого `answer`, есть свой `question`, такой тип связи называется агрегация.

Также можно рассмотреть `User` и `Lessons`, мы видим что между ними связь `hasMany`, то есть у каждого пользователя есть множество уроков, такая связь называется ассоциацией, уроки сразу могут принадлежать нескольким экземплярам пользователей.

Для того, чтобы лучше разобраться в типах связи приведу в пример таблицу 16. Диаграмма представлена на рисунке 13.

Таблица 16 – Различия между типами связей

Свойства	Агрегация	Ассоциация
Отношения	Части-целое	Объекты не связаны между собой
Члены могут принадлежать сразу нескольким классам	Да	Да
Существование членов управляется классами	Нет	Нет
Тип отношений	Имеет	Использует

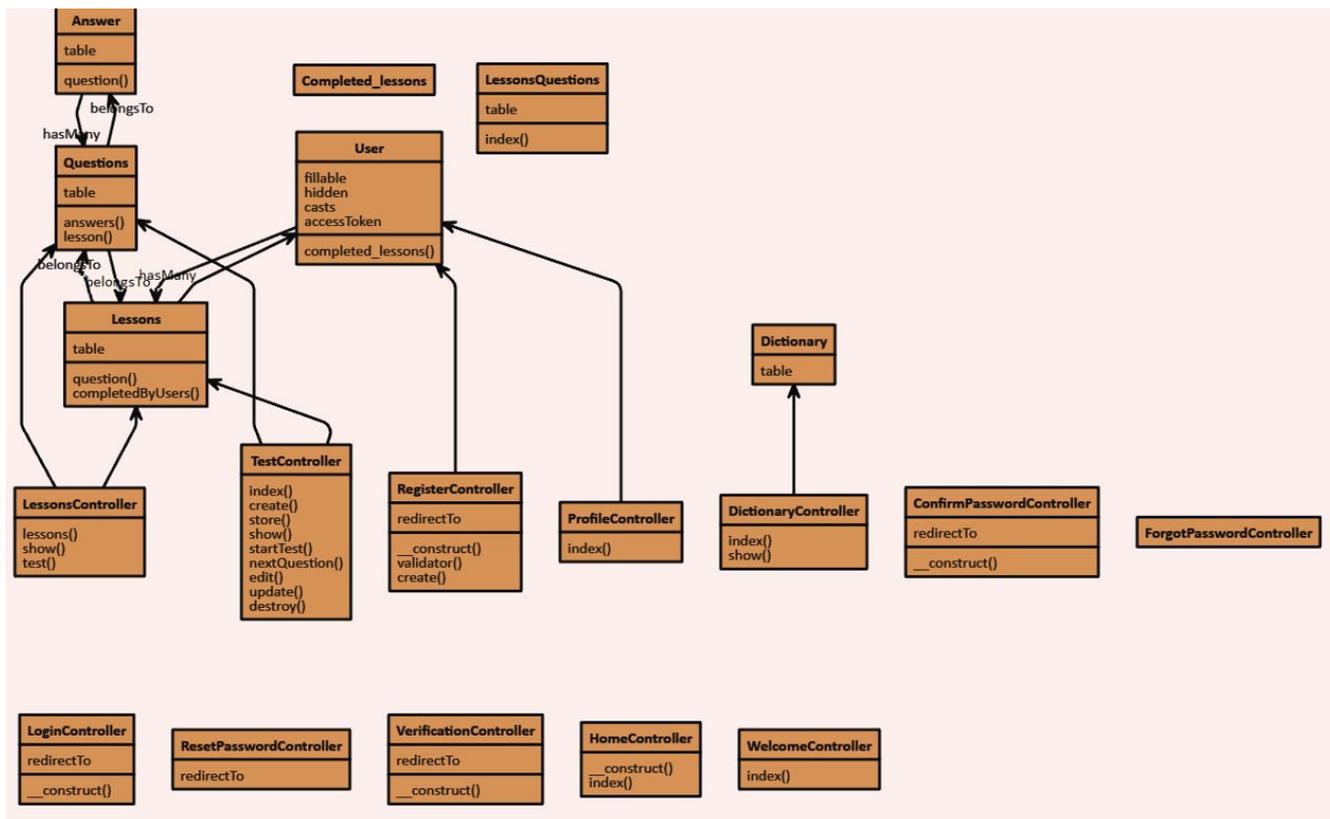


Рисунок 13 – UML диаграмма классов

В верхней части диаграммы показаны классы моделей, такие как: Answer, Questions, Lessons, User, Completed_lessons, LessonsQuestions, Dictionary, им соответствуют одноименные таблицы в базе данных: Answer, questions, lessons, user, completed_lessons, lessonsquestions и dictionary.

Далее приведу таблицу 17, где показывается соответствие между контроллерами и их функционалом.

Таблица 17 – Модели и соответствующие таблицы в БД

Контроллер	Функционал
TestController	Отвечает за логику прохождения тестов, вывод результатов и запись пройденных уроков в БД
LessonsController	Отвечает за вывод всех имеющихся уроков, за вывод и отображение на странице пройденных уроков
ProfileController	Отвечает за логику передачи информации о пользователе: его имени, почте, количестве пройденных уроков

DictionaryController	Отвечает за вывод всех слов из БД
----------------------	-----------------------------------

2.6 Выводы

В разработке веб-приложения для обучения важно создать интуитивно понятный и адаптивный интерфейс, который будет удобен для пользователей на любых устройствах. При этом пользователь должен иметь возможность переходить между различными состояниями интерфейса, такими как главная страница, форма регистрации и авторизации, а также основная страница с уроками после успешного входа. Важно, чтобы каждый этап взаимодействия был понятен и не создавал трудностей.

Для реализации работы с базами данных используется технология Eloquent в Laravel, которая применяет шаблон ActiveRecord. Это означает, что для каждой таблицы базы данных существует соответствующая модель, которая отвечает за взаимодействие с таблицей. Модели Eloquent позволяют не только получать данные, но и добавлять, обновлять и удалять записи.

Для проекта определены пять моделей: «lessons», «questions» (вопросы), «lessonsquestions» (связь между уроками и вопросами), «answers» (ответы на вопросы) и «users». Связи между таблицами, такие как «один ко многим» и «многие ко многим», организованы для эффективного управления данными. Это позволяет хранить и обрабатывать информацию о пользователях, уроках и тестах, обеспечивая функциональность, необходимую для успешного взаимодействия с приложением. Также была спроектирована UML диаграмма классов с их атрибутами и с подробным объяснением типов связей.

3. Разработка технологии тестирования

3.1 Виды тестирований

Существует несколько видов тестирования веб-приложений, но можно их разделить на две подгруппы: функциональное и нефункциональное.

Функциональное тестирование сосредоточено на оценке правильности работы функциональных возможностей приложения. Основная задача этой категории тестирования заключается в проверке соответствия разработанного программного обеспечения первоначальным функциональным требованиям. В рамках функционального тестирования, как правило, применяется метод тестирования черного ящика (Black Box Testing), который не затрагивает исходный код приложения. Этот вид тестирования охватывает такие аспекты, как

пользовательский интерфейс, база данных, безопасность, API и прочие функциональные характеристики.

Нефункциональное тестирование представляет собой процесс оценки характеристик системы, которые не связаны непосредственно с её функциональными возможностями. К таким характеристикам относятся, например, измерение показателей производительности, анализ стабильности и потребления ресурсов при различных уровнях нагрузки, а также тестирование работоспособности системы при различных условиях и удобства использования.

Для тестирования данного веб-приложения было проведено два вида тестирования - функциональное тестирования форм интерфейса и оценочное тестирование, где пользователи оценивали удобство пользования интерфейсом и совместимость с разными системами и браузерами.

3.2 Функциональное тестирование

Проведем функциональное тестирование с целью оценки веб-приложения на соответствие его функциональным требованиям. Для реализации данного тестирования необходимо изменять входные данные, что даст возможность оценить работоспособность разработанной системы без необходимости знания ее внутренней структуры.

Начнем с функционального тестирования форм окна авторизации. Будем поочередно вводить в поля ввода как корректные, так и некорректные данные, после чего сравним полученные результаты с ожидаемыми. Результаты тестов показаны в таблице 18.

Таблица 18– Функциональное тестирование формы входа в систему

Электронная почта	Пароль	Ожидаемый результат	Полученный результат
Поле оставлено пустым	Поле оставлено пустым	Выведено сообщение: «Введите данные»	Выведено сообщение: «Введите данные»

Введена почта зарегистрированная в базе данных	Поле оставлено пустым	Выведено сообщение: «Введите данные»	Выведено сообщение: «Введите данные»
Введена почта незарегистрированная в базе данных	Поле оставлено пустым	Выведено сообщение: «Пароль или почта введены некорректно»	Выведено сообщение: «Пароль или почта введены некорректно»
Введена почта зарегистрированного пользователя.	Введен неверный пароль	Выведено сообщение: «Пароль или почта введены некорректно»	Выведено сообщение: «Пароль или почта введены некорректно»
Введена почта зарегистрированного пользователя.	Введен верный пароль	Выполняется вход в систему	Выполняется вход в систему

Результаты тестирования продемонстрировали, что пользователь может успешно войти в систему при условии, что логин и пароль соответствуют установленным требованиям. В случае неправильного ввода не будет осуществлен доступ к системе и выводится сообщение с ошибкой введенных данных.

Были также протестированы и остальные элементы пользовательского интерфейса. Результаты тестирования представлены в таблице 19.

Таблица 19- Функциональное тестирование остальных форм интерфейса

Тестируемая форма интерфейса	Вид проверки	Ожидаемый результат	Полученный результат
------------------------------	--------------	---------------------	----------------------

Регистрация нового пользователя	Ввод некорректных данных	Вывод ошибки на экран	Вывод ошибки на экран
	Ввод корректных данных	Регистрация пользователя	Регистрация пользователя
Изменение пароля	Ввод некорректного текущего пароля	Вывод ошибки на экран	Вывод ошибки на экран
	Ввод корректного текущего пароля	Изменение пароля пользователя	Изменение пароля пользователя
	Поле нового пароля оставлено пустым	Вывод ошибки на экран	Вывод ошибки на экран
Режим тестирования	Нажатие на кнопку завершения тестирования	Вывод результатов тестирования	Вывод результатов тестирования

3.3 Оценочное тестирование

По завершении успешного функционального тестирования было проведено оценочное тестирование. В ходе него были выбраны два типа тестирования: тестирование удобства эксплуатации и тестирование совместимости с разными устройствами и браузерами. Для оценки удобства эксплуатации веб-приложения было привлечено 5 человек, которые оценивали его по различным критериям. Удобство использования каждого критерия было отмечено знаками "+" и "-". Результаты данного тестирования представлены в таблице 20.

Таблица 20- Оценочное тестирование удобства эксплуатации веб-приложения

Критерий оценки	1	2	3	4	5
Навигация по сайту	+	+	+	-	+
Понятны действия выполняемые кнопками	+	+	+	+	+
Удобство режима тестирования	-	+	+	+	+
Шрифт и размер шрифта	+	+	+	+	+
Используемая цветовая гамма	+	+	+	+	-
Понятна ли система обучения	+	-	+	+	-
Удобство восприятия информации на видеороликах	-	+	+	-	+
Расположение кнопок и полей ввода/вывода	+	+	+	+	+

Результаты тестирования показали, что удобство пользования веб-приложением составило примерно 83%. Основные проблемы, на которые указывали пользователи, касаются системы обучения и просмотра видеоматериала, поэтому система обучения требует доработки.

Для тестирования совместимости и корректности отображения данных веб-приложения с разными браузерами было выполнено тестирование поочередно на трех разных системах. Результат тестирования представлен на таблице 21.

Таблица 21- Оценочное тестирование совместимости

Тестируемый браузер	Операционная система	Результат тестирования
Google Chrome версии 131.0.6778.86	Windows 10	Успешно
Google Chrome версии 131.0.6778.81	Mac OS	Успешно
Google Chrome версии 131.0.6778.81	Android 14	Успешно
Опера версии 114.0.5282.243	Windows 10	Успешно
Опера версии 114.0.5282.243	Mac OS	Успешно
Опера версии 85.7.4500.82229	Android 14	Успешно
Yandex Browser версии 24.10.6.36	Windows 10	Успешно

Yandex Browser версии 24.10.6.36	Mac OS	Успешно
Yandex Browser версии 24.10.6.36	Android 14	Успешно
Safari версии 18.1	Mac OS	Успешно

Оценочное тестирование прошло успешно на всех системах и вышеперечисленных браузерах.

3.4 Выводы

В ходе функционального тестирования веб-приложения было проверено его соответствие заявленным функциональным требованиям. Тестирование форм окна авторизации показало, что система правильно обрабатывает как корректные, так и некорректные данные, обеспечивая доступ при правильном вводе логина и пароля и выводя сообщения об ошибках при неверных данных. Тестирование остальных элементов пользовательского интерфейса также продемонстрировало соответствие функциональности требованиям.

После функционального тестирования было проведено оценочное тестирование, включающее тестирование удобства эксплуатации и совместимости с различными устройствами и браузерами. В тестировании удобства эксплуатации участвовали 5 человек, и результаты показали, что уровень удобства использования составил 83%. Основными проблемами, отмеченными пользователями, стали система обучения и просмотр видеоматериалов, что указывает на необходимость доработки этих функций. Тестирование совместимости с разными браузерами подтвердило корректное отображение данных и функциональность приложения на различных системах и браузерах.

Заключение

В рамках проекта было успешно разработано веб-приложение для изучения русского жестового языка (РЖЯ), которое решает важную задачу доступности образования для широкой аудитории. Приложение предоставляет пользователям возможность не только

изучать РЖЯ, но и проверять свои знания с помощью тестов, что делает процесс обучения более интерактивным и эффективным. В ходе разработки была проведена тщательная работа по анализу существующих решений, что позволило учесть их сильные и слабые стороны и создать функционал, наиболее удобный для пользователей.

Важной частью работы стала разработка интуитивно понятного интерфейса, подходящего для различных устройств, а также проектирование базы данных с использованием технологии Eloquent в Laravel, что обеспечило эффективную работу с данными. Были созданы модели для управления уроками, вопросами, ответами и пользователями, что позволило организовать удобную структуру данных.

Тестирование приложения подтвердило его работоспособность и соответствие функциональным требованиям. Функциональное тестирование показало правильность обработки данных, а тестирование удобства использования и совместимости с различными устройствами выявило незначительные проблемы, которые будут доработаны.

В итоге, проект достиг поставленных целей, обеспечив создание полезного и удобного инструмента для обучения жестовому языку, доступного на различных платформах.

Литература

1. Зандстра М., PHP 8. Объекты, шаблоны и методики программирования, М.: Диалектика, 2021.
2. Дронов В. А., Laravel 9. Быстрая разработка веб-сайтов на PHP, СПб.: БХВ-Петербург, 2023.
3. Томсон Л. Веллинг Л., Разработка веб-приложений с помощью PHP и MySQL, М.: Вильямс, 2017.
4. Трахтенберг А. Склад Д., PHP. Рецепты программирования, М.: Питер, 2015.
5. Иванова Г. С., Технология программирования, М.: Кнорус, 2021.

ПРОТЕЗ НОГИ ЧЕЛОВЕКА ИЗ УГЛЕПЛАСТИКА

Меликянц Тигран Георгиевич

Москва г, ОАНО "Школа ЛЕТОВО", 10 класс

Секция: Ракетно-космические технологии и средства выведения

Научный руководитель: Новиков Андрей Дмитриевич, к.н., доцент каф. СМ13

МГТУ им. Н.Э. Баумана

Аннотация

Целью настоящего проекта является создание функционального протеза ноги человека из композиционных материалов для людей, утративших конечность, или страдающих заболеваниями опорно-двигательного аппарата.

Для достижения поставленной цели был поставлен ряд критериев, которым должен отвечать, разрабатываемый протез: функциональность, долговечность, упругость, влагостойкость, легкость.

Технологию изготовления протеза была разделена на пять этапов:

- Подготовка 3D-модели, включая моделирование кинематики движения;
- Изготовление оснасток для композитных деталей;
- Изготовление композитных частей протеза;
- 3D-печать деталей коленного сустава;
- Окончательная сборка протеза.

В результате работы была спроектирована трехмерная модель протеза ноги человека, на основе которой изготовлен функциональный прототип. Конструкция протеза позволяет людям, потерявшим конечность, пройти реабилитацию в кратчайшие сроки, а простота конструкции и невысокая себестоимость позволяет масштабировать производство подобных изделий до серийных масштабов.

За счет применения композитных материалов такой протез оказывается легче, прочнее и долговечнее аналогов из классических металлов и сплавов, может использоваться в любых погодных условиях. При этом протез будет выгоднее аналогов за счет высокой устойчивости к внешним факторам, из чего следует, что такое изделие будет реже нуждаться в ремонте. Относительная простота технологии изготовления показывает возможность расширить набор типоразмеров и быстро нарастить необходимый объем производства.

Введение

Согласно статистике, около 12% людей на планете имеют нарушения структур организма, снижающие его функциональность. Более 50 миллионов человек каждый год приобретают инвалидность по разным причинам, из них около 300 000 теряют ноги. Утрата конечности не только приводит к физическим ограничениям, но и несет в себе социальные проблемы для пострадавшего. Для решения медицинских проблем такого типа существуют различные виды протезирования [2].

История протезирования начинается в древнем Египте (около 900 лет до нашей эры), тогда изготавливали протезы из кожи и дерева. Речь не могла идти о полноценной замене конечностей таких как ноги или руки, однако более мелкие части такие как пальцы и фаланги могли заменяться и изготавливаться. Далее уже в Древнем Риме описывалось как полководец Марк Сергий использовал железный протез руки для удержания щита. Технологий и знаний для создания чего-то большего не хватало.



Рисунок 1 – Железный протез руки в Древнем Риме

Много лет спустя уже в средние века, в эпоху рыцарей и пиратов, железный протез как у Марка Сергия стал более массовым. Многочисленные сражения были причиной нередкой потери рук у воинов. Протезы были неудобные, не функциональные, однако позволяли держать щит и частично возвращали в строй бойца. Пираты тоже внесли не мало инноваций в протезирование, крюки и деревянные протезы ног, были просты, но выполняли функционал конечностей, также были легче и удобнее в изготовлении, чем железные протезы для удержания щита. Например, крюком можно было взять кружку, завязать канат, отразить атаку и держаться за перила лестницы [3,7].

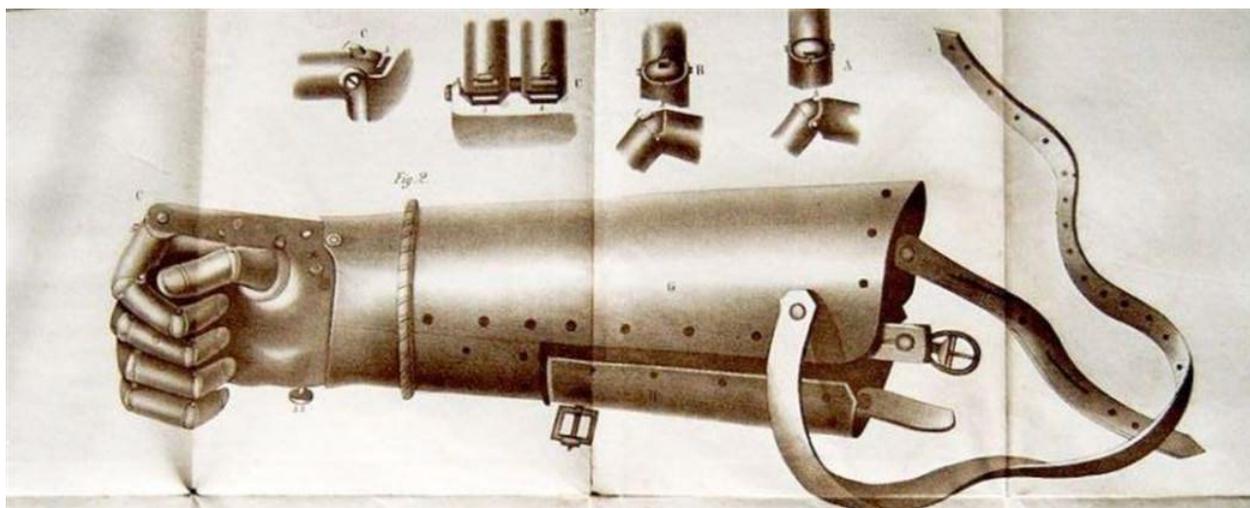


Рисунок 2 – Эскизы первых образцов протезов

Больше всего инноваций в протезировании произошло в эпоху ренессанса. Был изготовлен первый протез с подвижными пальцами, уже думали об эргономике, визуальной составляющей, а также заботились о функциональности. Проектировались и собирались рабочие суставы и механизмы, заменяющие работу отсутствующих конечностей. Вместо коленных суставов использовались шарниры, а уже известные железные руки имели возможность сгибания кисти и пальцев.



Рисунок 3 – Протез ноги эпохи ренессанса

В 19 веке множество гражданских и межгосударственных войн стали причиной необходимости огромного количества протезов, тогда началось их массовое производство протезов. Джеймс Хенгер, сам являвшийся калекой (у него не было ноги), создал компанию, которая стала самым крупным производителем протезов в мире в период конца 19 века [7].

В настоящее время благодаря инновационным прорывам в компьютерных технологиях и нейронауке мы имеем возможность создавать современные, улучшенные, роботизированные

версии протезов, которые раньше можно было увидеть только в фильмах, книгах научной фантастики. Например, в 1993 году была создана технология целевой реиннервации мышц, которая позволила улучшить управления протезами. Благодаря этому существуют протезы, которыми можно управлять почти также как живой рукой, что значительно увеличивает пользу и удобство протезов. В ходе многочисленных попыток и исследований удалось выяснить, что является необходимым для создания бионического протеза, и так в 2010 году появилась первая бионическая рука, которая управлялась сигналами от мозга пациента.

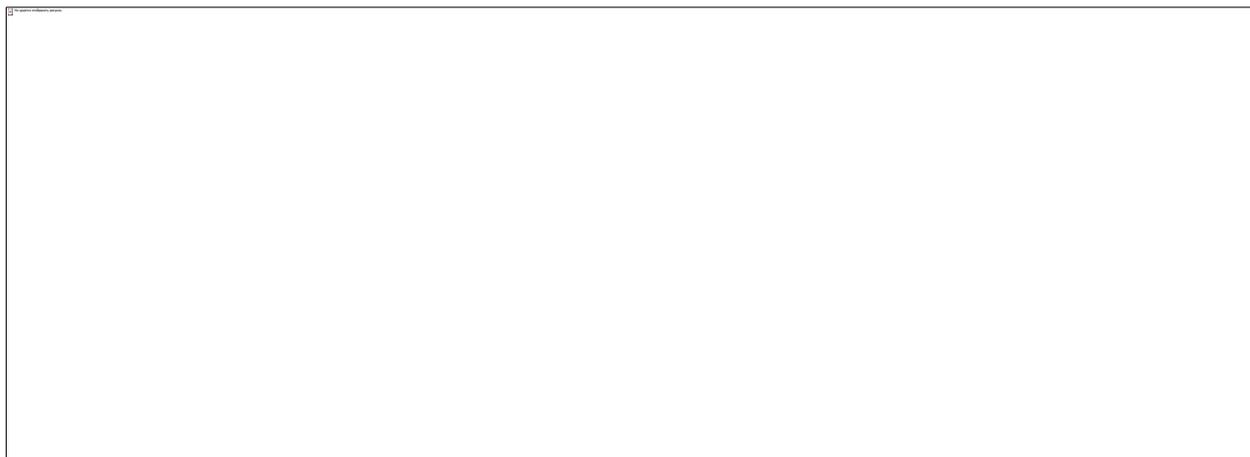


Рисунок 4 – Современные протезы

Улучшения протезов заключались не только в технологиях и механизмах, но также и в материалах, используемых в изготовлении. Начиналось все с железа, дерева и кожи, не очень долговечных, эргономичных и удобных в обработке. Сейчас же используются такие материалы как силикон, титан и композиты [4]. Титан считается очень прочным и легким, за счет чего идеален для создания каркасов, силикон также легок и приятен на ощупь, делает изделие комфортным, а композитные материалы составляют основу протеза, так как удобны для изготовления, обладают лучшими свойствами по сравнению с аналогами, легки и долговечны.



Рисунок 5 – Механизированный протез руки

На протяжении столетий протезирование было доступно только состоятельным людям, со временем деревянные палки вместо ног у пиратов стали полноценными протезами, которые имитировали вид и функции конечностей человека. Переломным моментом для доступности протезов стала промышленная революция, когда стало возможно массовое производство [3, 7]. Благодаря этому в настоящее время рынок протезов полон альтернатив, услуг и возможностей, хотя высокотехнологичные протезы остаются дорогими, государство выделяет миллиарды на льготы и развитие индустрии.

Протезы также стали производиться не только для повседневной жизни, но и для спорта. В 1976 году состоялись первые параолимпийские игры, что подтолкнуло производителей протезов к разработке технологичных протезов для занятий спортом. В 1988 году на таких играх спортсменами использовались протезы из углеволокна, что значительно улучшило их результаты и по их словам повысило комфорт. А в 2012 году человек с ампутированной ногой впервые учувствовал в олимпийских играх наравне с людьми без инвалидности. Конструкция спортивных протезов отличается от конструкции повседневных, она спроектирована исходя из удобства занятий определенным видом спорта, улучшения эргономики, амортизации, а также тракции [7].

На протяжении столетий протезирование было доступно только состоятельным людям, со временем деревянные палки вместо ног у пиратов стали полноценными протезами, которые имитировали вид и функции конечностей человека. Переломным моментом для доступности протезов стала промышленная революция, когда стало возможно массовое производство. Благодаря этому в настоящее время рынок протезов полон альтернатив, услуг и возможностей, хотя высокотехнологичные протезы остаются дорогими, государство выделяет миллиарды на льготы и развитие индустрии.



Рисунок 6 – Беговой протез

Хотелось бы отметить важный этап в развитии протезирования такой как 3D печать. Благодаря этой технологии, весь процесс изготовления протеза ускорился, начиная с подбора размеров и типа конструкции, заканчивая проектированием прототипов с нововведениями. В настоящее время существуют компании изготавливающие протезы на более чем 80% состоящие из деталей, напечатанных на 3D принтере [5].

Применение композиционных материалов, в частности, углепластика, позволяет снизить массу готового изделия, сохраняя жесткость и упругость. Наиболее доступной технологией производства качественного углепластика на сегодняшний день является вакуумная инфузия, позволяющая пропитывать углеродную ткань связующим в безвоздушном пространстве, минимизируя пористость материала [1].

Целью работы является создание прототипа бегового протеза из углепластика с подвижным коленным суставом.

Для достижения поставленной цели необходимо решить ряд задач:

1. Составить план-график выполнения проекта (диаграмму Ганта);
2. Создать 3D модель протеза и коленного сустава;
3. Создать 3D модель оснасток для изготовления деталей протеза;
4. Изготовить оснастки для формования изделия;
5. Изготовить композитные части протеза, а также пластиковые детали;
6. Собрать протез;
7. Провести макетные испытания.

Создаваемый протез должен решать ряд проблем пользователя, чем выгодно будет отличаться от аналогов, сохраняя актуальность долгие годы:

1. Долговечность (протез обладает высокой износостойкостью),

2. Упругость (используемые материалы позволяют использовать протез для занятий бегом),
3. Влагостойкость (протез выдерживает любые погодные условия),
4. Легкость и эргономичность (используемые материалы легче имеющихся аналогов для изготовления протезов),
5. Ремонтпригодность и модульность (конструкция протеза позволит подетально изменять его под конкретного пользователя, сохраняя функциональность).

Методика выполнения работы

Планирование работ

Ключевые задачи / сроки выполнения	Август	Сентябрь	Октябрь	Ноябрь	Декабрь	Январь	Февраль
Анализ данных, поиск информации	Сбор информации и	Анализ имеющихся аналогов					
3D моделирование			Моделирование моделей деталей и оснасток	Моделирование коленного сустава			
Создание оснастки			Создание оснастки культеприемной гильзы	Создание оснастки голени			
Изготовление деталей изделия					Изготовление культеприемной гильзы и голени из композитов	3D-печать деталей коленного сустава	
Оформление работы							Сборка протеза

Подготовка 3D моделей протеза

Моделирование самого протеза. Создание объемной модели в программе Autodesk Inventor 2020 с учетом анатомической формы ноги реального человека [5,6].

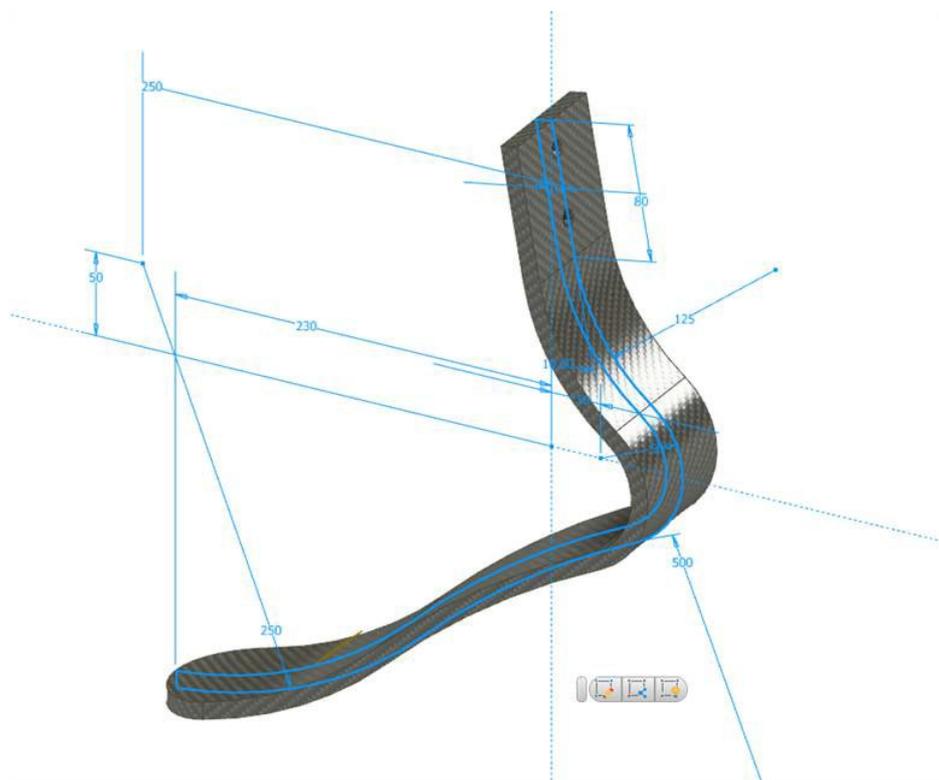


Рисунок 7 – Моделирование голени протеза

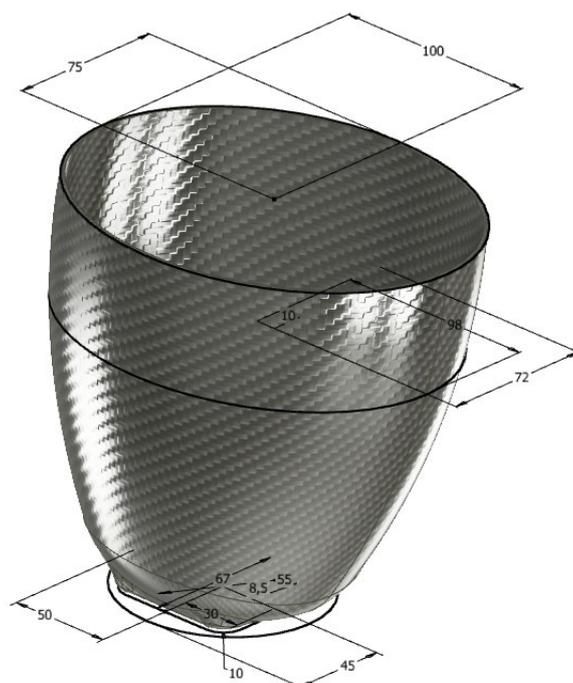


Рисунок 8 – Моделирование культеприёмной гильзы, с учетом анатомического строения бедра

Создание 3D-модели коленного сустава

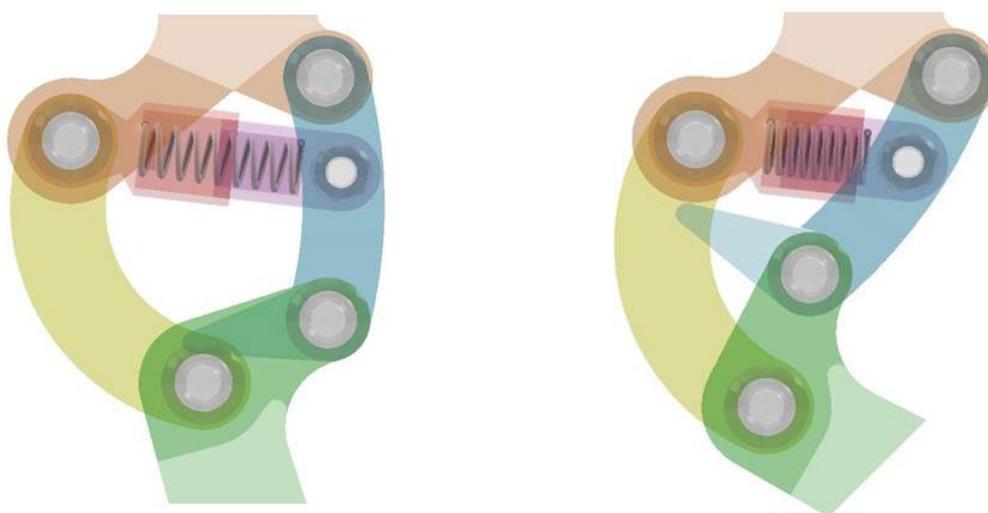


Рисунок 9 – Моделирование коленного сустава с использованием реактивного элемента в разжатом (а) и сжатом (б) состоянии

Коленный сустав спроектирован с возможностью изменения угла сгибания от 0 до 45 градусов [5].



Рисунок 10 – Итоговая 3D-модель протеза

В итоговой 3D-модели протеза добавлены опоры крепления коленного сустава, кронштейн фиксации голени, а также износостойкая накладка на голень.

Создание 3D моделей оснасток

Опираясь на 3D-модели необходимых изделий, в Autodesk спроектированы модели на оснастку, с учетом примерных размеров заготовки из МДФ.

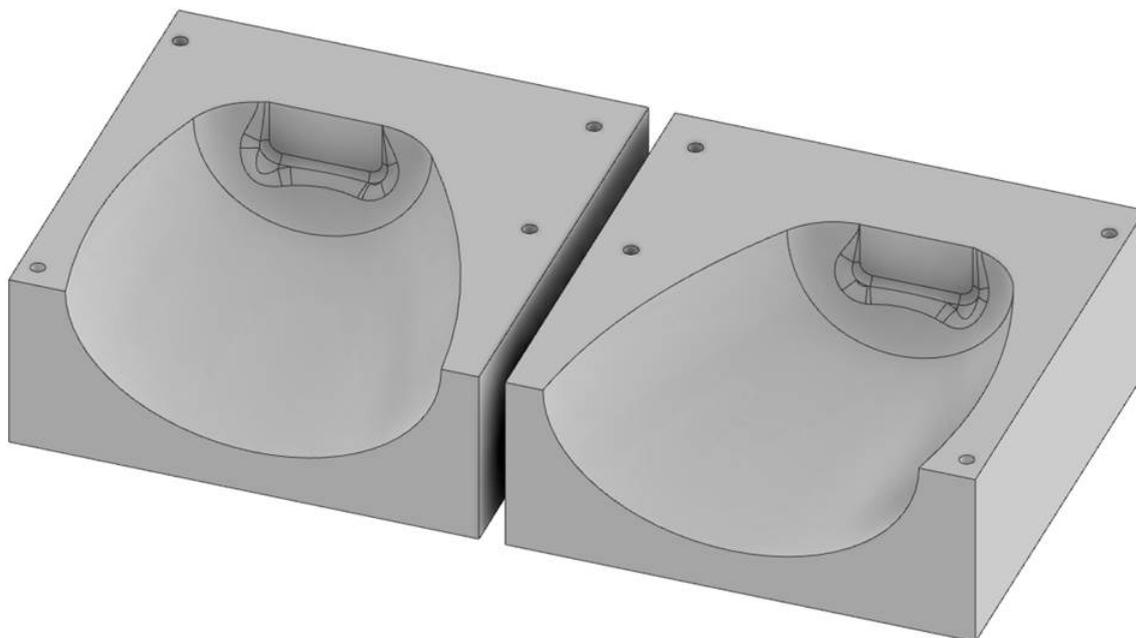


Рисунок 11 – Итоговая 3D-модель оснастки культеприемной гильзы

Изготовление оснасток

Склеивание заготовок

Склеивание заготовок МДФ для работы с фрезерным станком. Склеивание листов с помощью смолы, закрепление конструкции зажимами.



Рисунок 12 – Склеивание заготовки из МДФ

Создание управляющих программ для ЧПУ

На основе 3D-моделей были созданы программы для ЧПУ-фрезеровки в программном обеспечении Autodesk PowerMill.

Черновая фрезеровка оснасток

Фрезеровка оснасток. Фиксация с помощью зажимов и болтов склеенных заготовок МДФ на рабочей поверхности станка. Передача программы станку с помощью флешки. Калибровка станка по координатам. Очистка станка по необходимости от опилок, задерживающих и перегревающих фрезу.



Рисунок 13 – Фрезеровка оснасток на станке с ЧПУ

Черновая фрезеровка выполняется с припуском 0,3...0,5 мм по рабочей поверхности модели для последующего поразуполнения связующим и чистовой фрезеровки.

Пропитка оснасток смолой

Замешивание смолы и покрытие ей поверхности оснастки. Сушка промышленным феном при высокой температуре. Повторение операции слой за слоем, до тех пор, пока поверхность не перестанет впитывать смолу.



Рисунок 14 – Пропитка оснасток

Чистовая фрезеровка оснасток

Была создана программы для чистовой фрезеровки оснасток. Необходимо снять поверхностный слой смолы, чтобы получить оснастку с полным соответствием 3D-модели.



Рисунок 15 – Оснастки культеприемной гильзы после чистовой фрезеровки

Изготовление композитных частей протеза

Нанесение разделителя

Произведена очистка рабочих поверхностей оснасток бензином. Снятие мелких частиц от шлифовки с помощью ткани, смоченной бензином.



Рисунок 16 – Очистка рабочих поверхностей оснастки

Оснастки были покрыты разделительным воском, для удобства снятия готового изделия. Нанесение воска производилось в 3 слоя с интервалом 30 минут, финальный слой отполирован ветошью.

Раскрой армирующих материалов

Раскрой углеткани различного плетения для получения армирующих слоев протеза. Для соблюдения направления укладки углеткани на поверхность оснастки и обеспечения экономного расхода материала.



Рисунок 17 – Раскрой углеткани

Формование изделия

Укладка армирующих слоев, используя мелкодисперсный распылительный клей для фиксации, податливости и крепления кусков углеволоконных полотен [4,6].

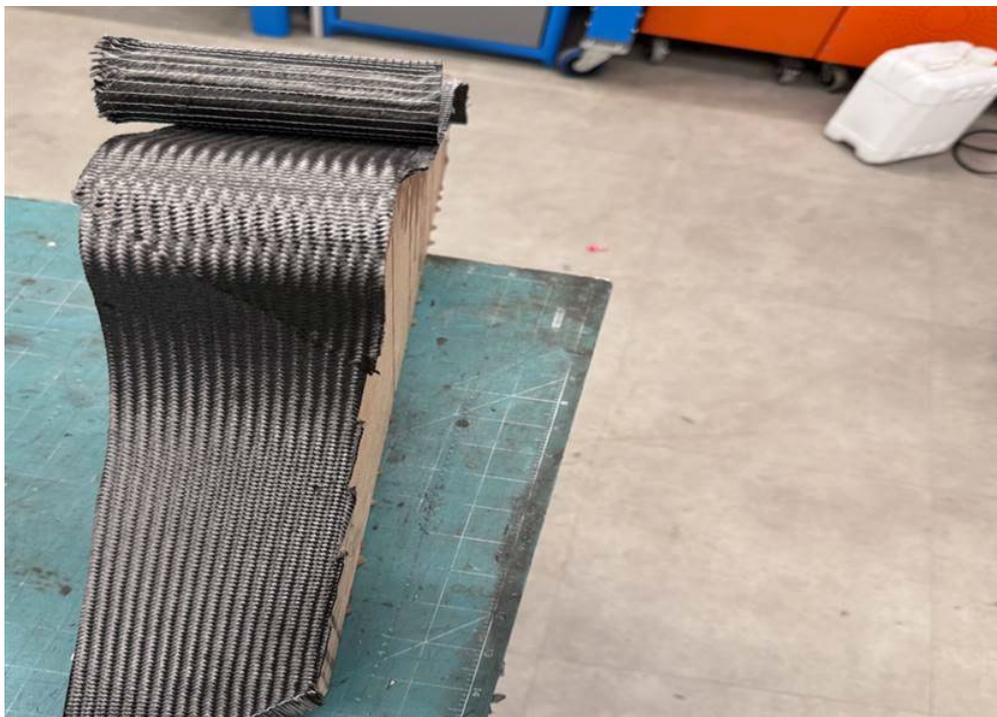


Рисунок 18 – Формование изделия



Рисунок 19 – Формование культеприемной гильзы

Укладка вспомогательных материалов

Укладка жертвенной ткани, которая служит для облегчения удаления смолопроводящей сетки, а также, через которую связующее будет постепенно и равномерно распределяться по армирующему материалу представлена на рис.20.

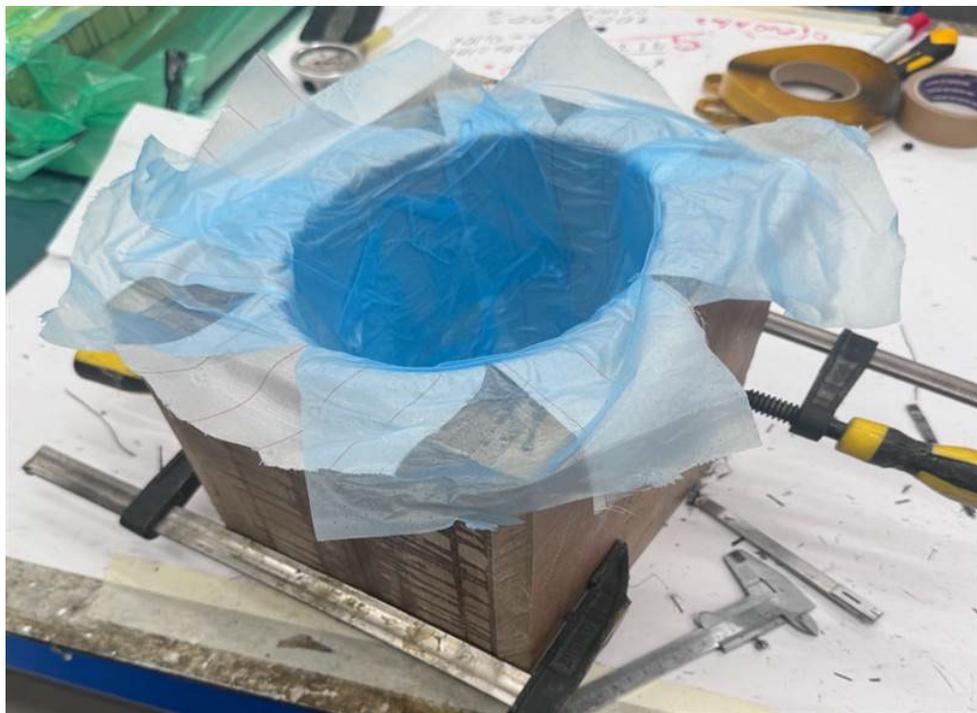


Рисунок 20 – Укладка жертвенной ткани

Сборка вакуумного пакета

Произведена сборка герметичного пакета для вакуумной инфузии, используется герметизирующий жгут по периметру для предотвращения натекания воздуха. Во время сборки пакета вставлены трубки для подачи связующего и откачки воздуха с помощью вакуумного насоса.

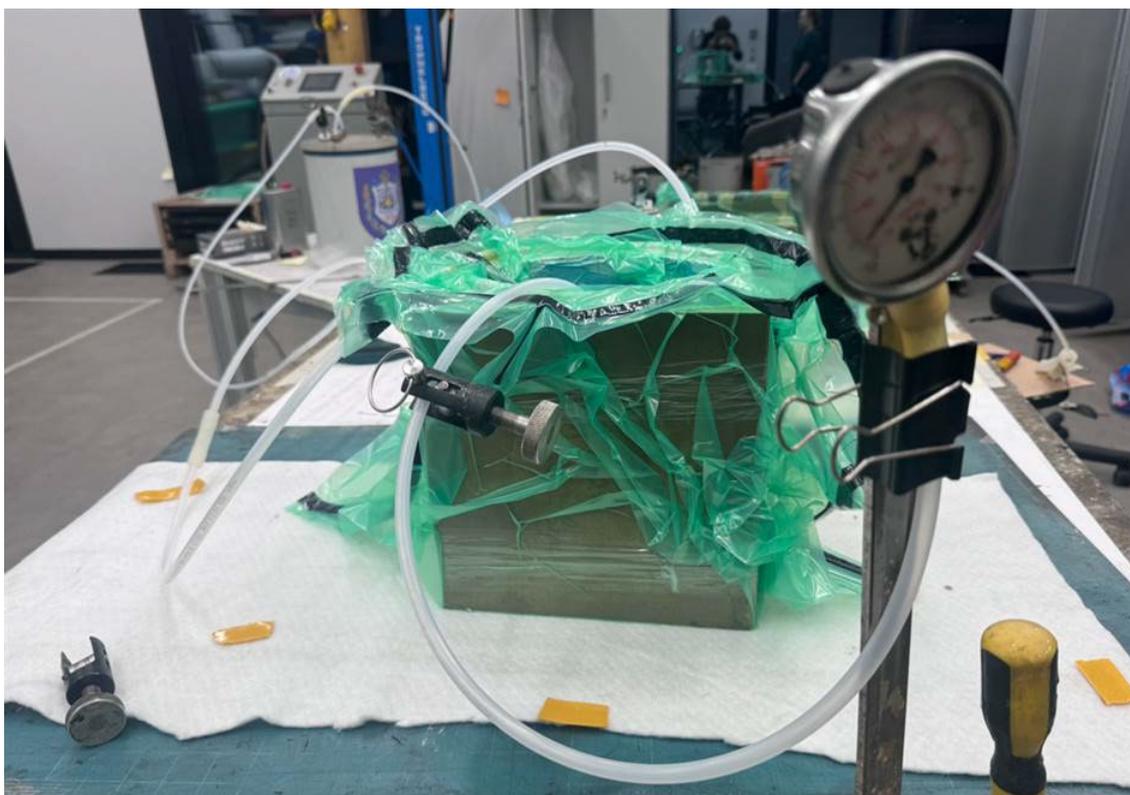


Рисунок 21 – Сборка вакуумного пакета, откачка воздуха

Пропитка изделия

Пропитка изделия осуществлялась путем подачи связующего через трубку подачи. Выбрано низковязкое инфузионное связующее отечественного производителя Т20-60.

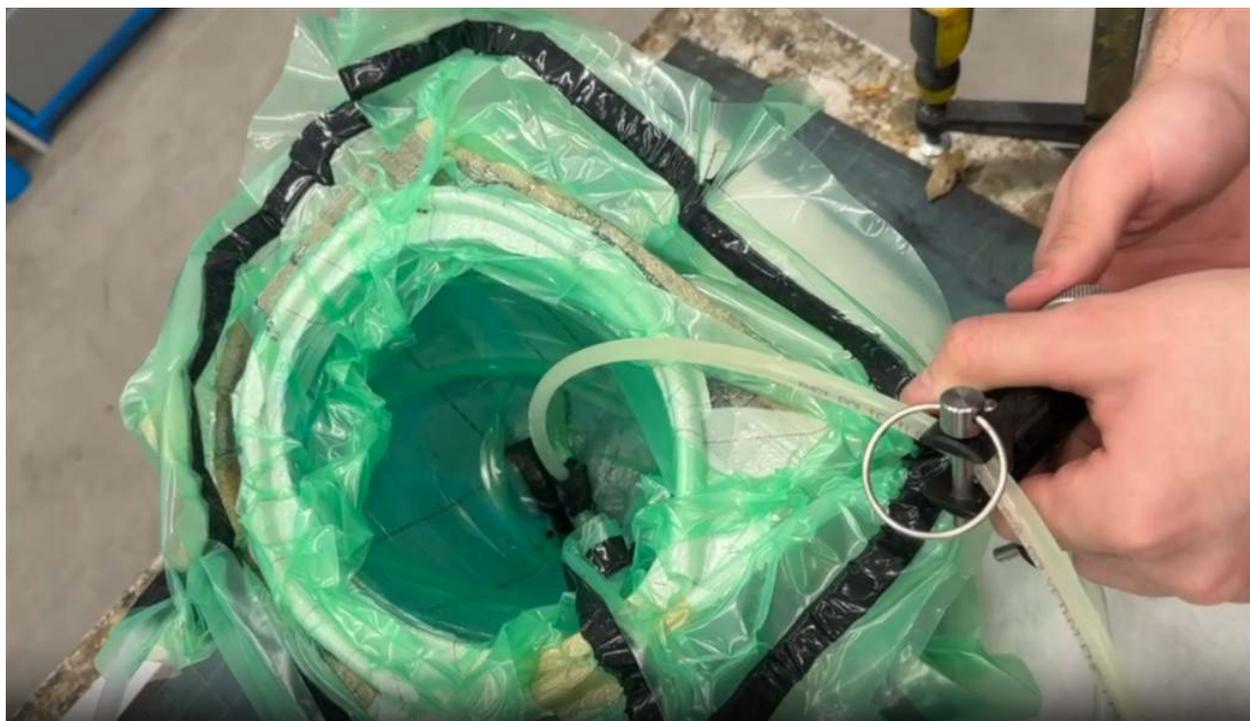


Рисунок 22 – Пропитка изделия связующим

Съем изделия

После осуществления пропитки связующее отверждалось при комнатной температуре в течение 48 часов. Вакуумный пакет был разобран, удалена жертвенная ткань, изделие снято с оснастки.

Снятое изделие требует обрезки по контуру и сверления отверстий для фиксации коленного сустава.



Рисунок 23 – Культеприемная гильза после съема с оснастки

Изготовление деталей коленного сустава

Детали коленного сустава были изготовлены посредством 3D-печати из PETG-пластика. Для обеспечения качества была выбрана толщина слоя 0,15 мм [5]. Детали стопы напечатаны из упругого пластика TPU-soft.



Рисунок 24 – 3D-печать деталей коленного сустава

Механическая обработка углепластиковых деталей

Обработка поверхностей

После снятия деталей с оснасток, предстояло обрезать изделия по контуру, отшлифовать поверхности, избавив их от избытков связующего, неровностей и острых углов, а также покрыть детали лаком, исходя из соображений о визуальной привлекательности протеза в качестве готового продукта.

На первом этапе был обрезан крупный облой с помощью болгарки. Для финишной обработки кромок деталей, использовалась наждачная бумага с зернистостью P80 и P120.

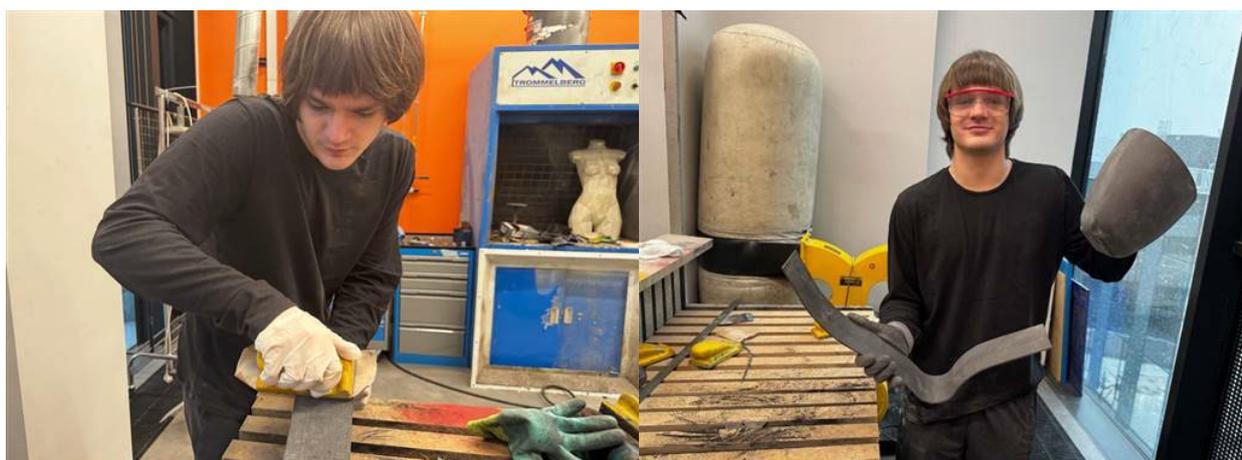


Рисунок 25, 26 – Механическая обработка деталей протеза

Для обработки лицевой поверхности деталей последовательно использовалась наждачная бумага P240 и P360. После механической обработки изделие было очищено от пыли с помощью ветоши и бензина «Галоша».

Нанесение рисунка и покрытие лаком

Для повышения привлекательности внешнего вида изделия, было принято решение о нанесении рисунка и покрытии лицевой стороны изделий лаком. Кроме того, рисунок на голени протеза позволяет различить левое и правое исполнение детали.

Предварительно на изделие был наклеен трафарет из малярного скотча. После этого поверхность изделия была обезжирена и нанесена краска распылением на немаскированные поверхности.

После высыхания краски малярный скотч был удален с изделий. Далее изделия были покрыты матовым лаком для устойчивости к повреждению поверхности и привлекательности внешнего вида.



Рисунок 27 – Композитные части протеза с рисунком, покрытые лаком

Сборка протеза

Сборка сустава

Сборка сустава включала в себя установку всех подвижных частей. Для облегчения сустава и обеспечения компактности в конструкции применены гайки Эриксона (цилиндрические с внутренней резьбой). В центральной части сустава установлена пружина со следующими параметрами: длина L-40 мм, внешний диаметр D-13,7 мм, диаметр проволоки d-1,0 мм. Пружина устанавливается внутри цилиндра в преднагруженном (сжатом) состоянии и сжимается сильнее по мере увеличения угла сгибания сустава [5,6].



Рисунок 28 – Коленный сустав протеза в сборе

Сборка голени

Для сборки голени были просверлены необходимые монтажные отверстия в композитной детали. Далее были присоединены кронштейн крепления сустава к голени, а также накладка стопы, которая отвечает за надежное сцепление протеза с землей, амортизацию, комфорт и износостойкость.



Рисунок 29 – Голень с кронштейном и накладкой стопы, напечатанными на 3D-принтере

Конструкция обеспечивает возможность оперативной замены накладки стопы при ее износе, для этого требуется открутить лишь два болта.

Сборка культеприёмной гильзы

В культеприёмной гильзе были просверлены отверстия для установки фиксатора коленного сустава. Фиксатор сустава был закреплен на 4 болта у культеприёмной гильзе, обеспечивая их надежное соединение.



Рисунок 30, 31 – Сверление отверстий и сборка культеприемной гильзы

Окончательная сборка протеза

Окончательная сборка протеза заключается в фиксации коленного сустава, культеприемной гильзы и голени в едины механизм. Поскольку все соединения болтовые, разъемные, то это обеспечивает высокую ремонтпригодность протеза, а также демонстрирует модульность представленной конструкции. Такой подход обеспечивает легкую замену любой части протеза при износе или поломке, а также расширяет возможности по изменению и модернизации конструкции под конкретного пользователя.



Рисунок 32 – Протез в сборе

Макетные испытания протеза

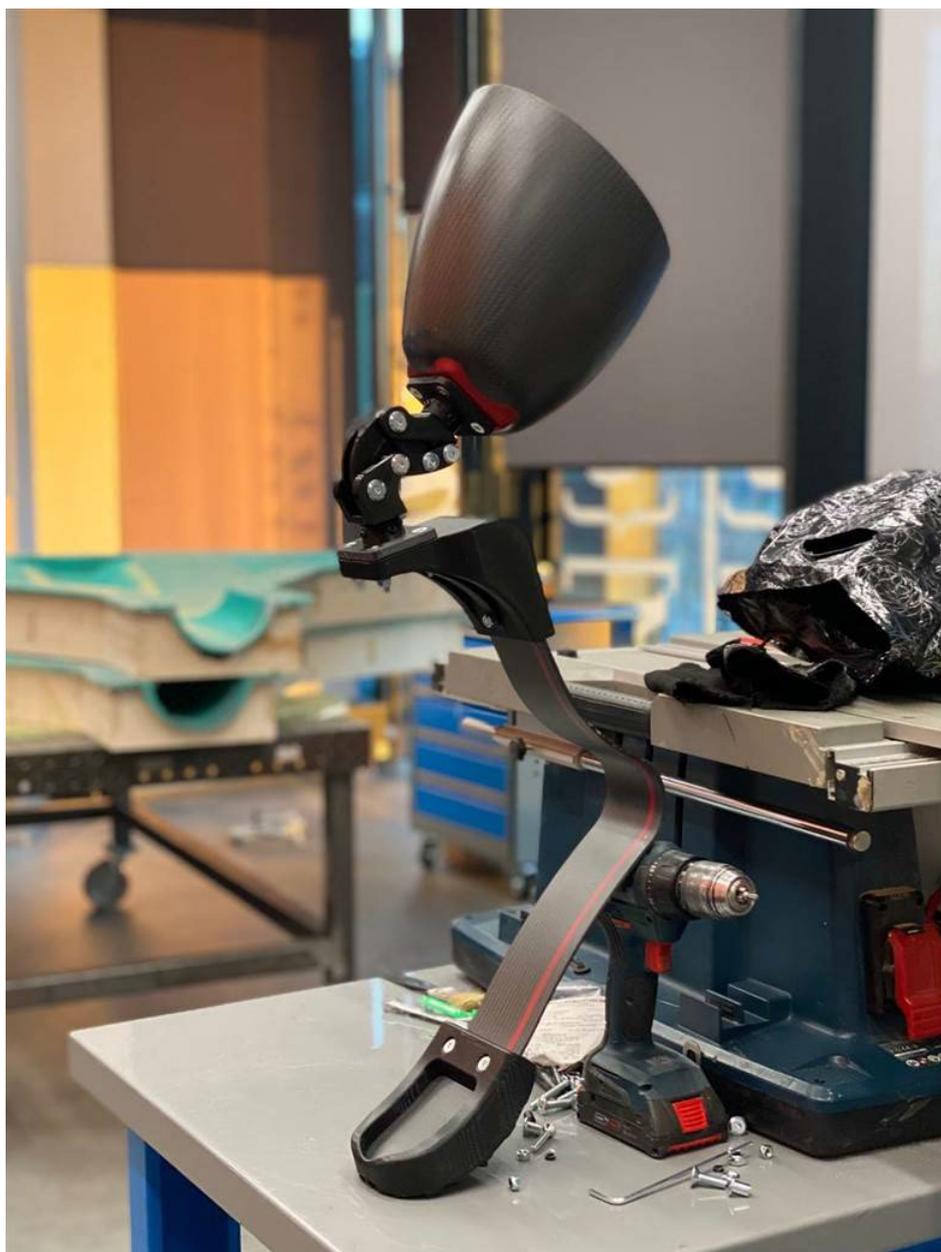


Рисунок 33 – Протез в сборе, с согнутым коленным суставом

Макетные испытания показали, что собранный протез ожидаемо упруго ведет себя при вертикальной нагрузке, способен сгибаться на 45° и автоматически возвращаться в исходное положение. Протез представляет собой модульное устройство и каждая деталь может быть оперативно заменена в полевых условиях с использованием карманного набора шестигранников.

Заключение

В результате работы над проектом спроектирован и собран прототип бегового протеза из углепластика с подвижным коленным суставом.

Для достижения поставленной был решен ряд задач:

- Создана 3D-модель протеза, включающая культеприемную гильзу, голень, подвижный коленный сустав, кронштейны их крепления между собой, накладку стопы;
- Для изготовления композитных деталей протеза создана 3D-модель оснасток;
- Изготовлены оснастки для формования изделий с применением фрезерного станка с ЧПУ;
- Изготовлены композитные части протеза методом вакуумной инфузии, а также пластиковые детали методом 3D-печати (FDM);
- Протез собран и проведены макетные испытания работоспособности коленного сустава.

Разработанный беговой протез из композиционных материалов демонстрирует способность решать прямую потребность человека в утраченной конечности, при этом обеспечивая параметры:

- Амортизации – конструкция обеспечивает демпфирование при беге,
- Легкости – низкий вес всего протеза за счет применения композиционных материалов и оптимизации конструкции вспомогательных деталей,
- Износостойкости и надежности – модульная конструкция обеспечивает легкую замену любой части протеза при износе или поломке,
- Влагоустойчивости – в протезе отсутствуют электронные компоненты или детали, подверженные коррозии во влажном климате.

Высокая степень проработанности протеза говорит о доступности технологий и материалов для его изготовления в массовом производстве и масштабирования размерной сетки для людей разного телосложения.

Список используемых источников

1. Иванов Д.А., Ситников И.А., Шляпин С.Д. Композиционные материалы. – 2024
2. 54% всех инвалидов СВО прошли ампутацию — большая часть лишилась ног // Дзен URL: <https://dzen.ru/a/ZS5ix3YIUbaam2tI> (дата обращения: 10.09.2024).
3. История протезирования: от древности до наших дней // Архангельское протезно-ортопедическое предприятие URL: <https://prop29.ru/articles/istoriya-protezirovaniya/> (дата обращения: 15.09.2024).
4. Кузовкин А.В. Технологичность конструкций. Лабораторный практикум. Уч. пособие. - М.: ЭБС Лань, 2019
5. Юрковский О.И. Модульный принцип протезирования нижних конечностей. - СПб.: ЭЛБИ-СПб, 2004

6. Протезирование нижних конечностей // ОртоРеал URL: <https://ortoreal.ru> (дата обращения: 21.10.2024).
7. Протезирование. "История протезирования." Prop29. Доступно по адресу: <https://prop29.ru/articles/istoriya-protezirovaniya> (дата обращения: 26.02.2025).

**ПРЕДСКАЗАНИЕ ЗНАЧЕНИЯ УДЕЛЬНОГО ИМПУЛЬСА РАКЕТНОГО
ДВИГАТЕЛЯ С ПОМОЩЬЮ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ С УЧЕТОМ
ВАРЬИРОВАНИЯ ХИМИЧЕСКОГО СОСТАВА ТОПЛИВА**

Мирославлев Илья Антонович

Иваново г, МБОУ "СШ №4", 10 класс

Секция: Энергетическое машиностроение

Научный руководитель: Луканов Михаил Михайлович, аспирант

АНОО ВО «Сколковский институт науки и технологий»

Аннотация

Бурные темпы развития науки и технологий промодутируют человечество на совершенствование уже имеющихся и создание новых технологий в различных областях промышленности и индустрии. К одной из таких областей, безусловно, можно отнести аэрокосмическую индустрию, где научная составляющая является главной движущей силой всего развития. Исследование космоса, развитие системы спутниковой связи, обеспечение государственной безопасности – всё это невозможно без создания эффективных ракетных двигателей и видов их топлива. Таким образом, выбор и совершенствование ракетных двигателей и видов топлива является весьма актуальной задачей на сегодняшний день.

К настоящему времени накоплено большое количество опытных данных, которые можно и нужно использовать для создания новых технологий. Метод «проб и ошибок» является далеко не эффективным в современных реалиях. На его смену приходит другой – метод моделирования, посредством которого можно прогнозировать и предсказывать новые характеристики и свойства различных систем. В качестве таких методов широкое применение нашли алгоритмы машинного обучения, эффективность и предсказательная способность которых позволяет быстро и в некоторых случаях «дешево» получать точные математические модели.

В данной работе использованы методы машинного обучения для прогнозирования удельного импульса ракетного двигателя на основе характеристик ракетного топлива. Результаты данной работы будут реализованы в виде программного кода

ВВЕДЕНИЕ

Бурный темп роста информационных и компьютерных технологий за последние десятилетия сильно способствуют развитию различных областей науки и техники, которые, на первый взгляд, кажутся совершенно далёкими от этих технологий. Конечно, такая наукоёмкая область, как космонавтика, не является исключением из общего правила. Можно, а порой даже нужно и необходимо использовать современные вычислительные технологии для решения практически важных задач. В противном случае может возникнуть стагнация в этой области (что, к сожалению, просматривается уже сейчас в сравнении с советской эпохой космонавтики). В этой работе представлено применение современных методов математического моделирования, в качестве которых выступили методы машинного обучения (или, как это модно стало произносить, методы искусственного интеллекта) для решения задачи определения удельного импульса ракетного двигателя.

Безусловно, искусственный интеллект никогда не сможет заменить интеллект человеческий - ведь он был создан не для этого! Да, современные технологии способствуют в каком-то смысле «обломовщине» среди людей, но это вина далеко не только искусственного интеллекта. Современные методы машинного обучения в своей основе требуют от инженера фундаментальной подготовки по математике, физике и другим дисциплинам. Например, современные диффузионные модели, которые изначально создавались для генерации картинок, сегодня используются для задач химии и физики, а в их основе лежит уравнение Фокера-Планка. Поэтому к методам машинного обучения нужно относиться с такой же серьёзностью, как и к другим методам математического моделирования, которые помогают проливать свет на некоторые природные явления, помогающие лучше понять законы природы.

1. Теоретическая часть

1.1. Общие сведения о химических ракетных двигателях и видах их топлива

Хочется начать эту часть с базового определения в рамках данной работы - с определения ракетного двигателя. Ёмкое и, на мой взгляд, исчерпывающее определение дано в учебном пособии Егорычева В.С. [1], где ракетный двигатель - автономный реактивный двигатель, работающий на бортовых ресурсах массы и энергии. В данном определении отражены два фундаментальных аспекта:

1) движение, осуществляемое ракетным двигателем, по своей природе является реактивным видом движения;

2) для функционирования ракетного двигателя необходимы ресурсы массы и энергии (что эквивалентно в силу соотношения Эйнштейна $E = mc^2$), т.е. необходимо топливо.

Связь между пунктами 1 и 2 есть не что иное, как закон преобразования одного вида энергии (тепловой и потенциальной энергии продуктов горения топлива) в другой (кинетической энергии струи газов). Схематично связь между этими пунктами можно представить следующей схемой в соответствии с рисунком 1.

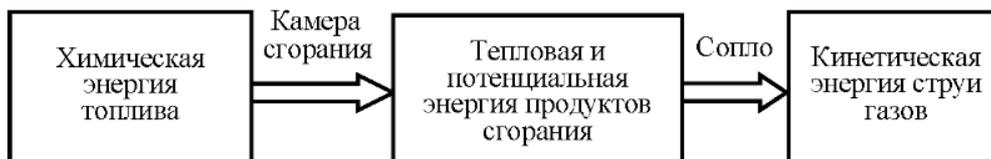


Рисунок 1. Преобразование энергии, реализуемое в ракетном двигателе (под химической энергией понимается потенциальная энергия, аккумулированная в химических связях молекул топлива).

Как можно заметить из схемы, основное внимание в данной работе будет уделено именно химическим видам топлива, хотя существуют и альтернативы (ядерное, электрическое и другие виды).

По агрегатному состоянию используемого топлива ХРД подразделяют на несколько видов:

- 1) жидкостные ракетные двигатели,
- 2) ракетные двигатели твёрдого топлива,
- 3) гибридные ракетные двигатели.

Конструкция ХРД будет определяться в том числе агрегатным состоянием топлива, и именно поэтому выбор химического состава ХРТ является решающим этапом при конструировании и модернизации ракетных двигателей.

Для большей краткости, здесь будут приведены общие свойства ХРТ, частные характеристики и требования могут варьироваться в зависимости от агрегатного состояния.

Общее определение ХРТ можно выразить следующим образом: химическое топливо представляет собой вещество или совокупность веществ, способных в результате экзотермических химических превращений образовывать высокотемпературные продукты, создающие реактивную тягу.

Очень часто используют двухкомпонентные ХРТ, в которых осуществляется реакция горения. В качестве компонентов топлива выступают окислитель и горючее. Простым примером горючего и окислителя могут служить водород (H_2) и кислород (O_2) соответственно.

Существует ряд требований, предъявляемых к ХРТ. В частности, можно выделить энергетические характеристики топлива, его кинетические свойства, эксплуатационные характеристики, а также экономические и экологические свойства. В данной работе особое внимание было уделено такой энергетической характеристике топлива, как удельный импульс или массовая теплота сгорания.

1.2. Удельный импульс

Для введения понятия удельного импульса определим, что называют тягой двигателя. Примем, что скорость истечения продуктов горения из сопла двигателя равна \vec{u} , а μ - их масса, проходящая через сопло в единицу времени. Тогда опыт и теория показывают, что на тело будет действовать сила \vec{T} , модуль которой определяется как:

$$T = u \quad (1)$$

Сила эта направлена противоположно направлению истечения продуктов горения и называется реактивной силой. Заметим, что масса ракеты будет уменьшаться, причем, если скорость истечения продуктов горения постоянна, то:

$$m = m_0 - \mu t, \quad (2)$$

где m_0 - массы ракеты в начальный момент времени, кг;

m - массы ракеты в текущий момент времени, кг.

Применяя второй закон Ньютона и учитывая, что на ракету действуют сила тяжести и сила сопротивления, можно получить следующее уравнение движения:

$$(m_0 - \mu t)(a + g) = T - R, \quad (3)$$

где a - ускорение ракеты, м/с²;

g - ускорение свободного падения, м/с²;

R – сила сопротивления, Н.

Если говорить честно, то здесь не совсем корректно применять второй закон Ньютона, так как ускорение в нём введено для тела постоянной массы, что не выполняется в случае реактивного движения ракеты. Однако оказывается, что три закона Ньютона позволяют получить аналогичное выражение совершенно строго. Это впервые доказал И. В. Мещерский в 1897 году в работе «Динамика точки переменной массы». Уравнения подобного типа носят название уравнений Мещерского.

В качестве основного показателя эффективности ракетного двигателя используют величину силы тяги, отнесённой к весу рабочего тела (продуктов горения), израсходованного ракетным двигателем за единицу времени:

$$P = \frac{T}{\mu g}, \quad (4)$$

где P – сила тяги, Н.

Такой показатель называют удельным импульсом (удельной тягой) двигателя, измеряемого в единицах времени. Стоит оговориться, что в литературе встречаются разные характеристики (удельный импульс (тяга) ракетного двигателя, удельный импульс тяги ракетного двигателя, объёмный удельный импульс ракетного двигателя), которые по существу являются эквивалентными и отличаются на постоянный размерный множитель. Чаще оказывается удобнее пользоваться понятием удельного импульса тяги:

$$I = \frac{T_0}{\mu} = u, \quad (5)$$

где I – удельный импульс тяги, м/с;

T_0 – расчётная тяга ракетного двигателя, Н;

u – эффективная скорость истечения рабочего тела, м/с.

Расчетная тяга ракетного двигателя характеризует тягу в пустоте, то есть при отсутствии взаимодействия с атмосферой посредством разницы давлений с выходом сопла. Фактическая тяга ракетного двигателя зависит от влияния показателей давления окружающей среды.

$$T = T_0 + S(P_A - P_H), \quad (6)$$

где T – фактическая тяга ракетного двигателя, Н;

S – площадь выходного сечения камеры сгорания, м²;

P_A – давление в камере сгорания, Па;

P_H – давление невозмущенной среды, Па.

Тогда выражение для удельного импульса тяги переписется в виде:

$$I = \frac{T_0}{\mu} + S(P_A - P_H) = u + S(P_A - P_H) \quad (7)$$

По аналогии вводится понятие объёмного удельного импульса (отношение величины тяги к объёмному расходу рабочего тела).

Говоря о ракетах, нельзя не упомянуть отечественного учёного, Константина Эдуардовича Циолковского, который получил уравнение, впоследствии названное его именем (формула Циолковского).



Рисунок 2. Белорусский почтовый блок 2002 года. Формула Циолковского (внизу) приведена в близком к записанному Циолковским виде.

Классическое ракетное уравнение, или уравнение идеальной ракеты, - это математическое уравнение, которое описывает движение транспортных средств, следующих основному принципу ракеты. Этот принцип гласит: устройство, которое может придавать ускорение самому себе, используя тягу, выбрасывая часть своей массы с высокой скоростью, может двигаться за счет сохранения импульса. В 1903 году в работе «Исследование мировых пространств реактивными приборами» К.Э. Циолковским было опубликовано основное уравнение движения ракеты, определяющее её характеристическую скорость. Оно было выведено ученым в рукописи «Ракета» в 1897 году. По формуле Циолковского определяется максимальная скорость, которую может получить одноступенчатая ракета в идеальном случае, когда её полёт происходит не только вне пределов атмосферы, но и вне пределов поля тяготения Земли. Формулу Циолковского можно получить из уравнения Мещерского, описанного выше. Перепишем это уравнение в дифференциальной форме:

$$m(t) \frac{d\vec{v}}{dt} = \frac{dm(t)}{dt} \vec{u} + \vec{F}, \quad (8)$$

где масса тела m - функция от времени, кг;

\vec{v} - скорость тела, м/с;

\vec{u} - скорость продуктов горения, м/с;

\vec{F} - сила, действующая со стороны других тел или поля, Н.

В случае отсутствия внешнего поля и постоянной скорости истечения \vec{u} путём интегрирования можно прийти к формуле Циолковского:

$$\int_{m_0}^m \frac{dm(t)}{m(t)} = \int_{\vec{v}_0}^{\vec{v}} \frac{d\vec{v}}{\vec{u}} \quad (9)$$

$$\vec{v} = \vec{v}_0 - \vec{u} \ln\left(\frac{m_0}{m}\right) \quad (10)$$

Заметим, что \vec{u} есть не что иное, как введённый ранее удельный импульс тяги. Перепишем уравнение в скалярной форме, приняв начальную скорость равную нулю:

$$v = I \ln\left(\frac{m_0}{m}\right) \quad (11)$$

Таким образом, эффективность ракеты тем выше, чем больше удельный импульс тяги при прочих равных условиях. Чем выше удельный импульс, тем меньше требуется вещества и больше энергии для создания единицы тяги. Стоит, однако, заметить: отличие в формуле реальной скорости от характеристической заключается в том, что в условиях реального полёта скорость, развиваемая ракетами, как правило, ниже характеристической из-за потерь, вызываемых силами гравитации, сопротивления среды и другими факторами. Формула Циолковского даёт только верхнюю границу скорости ракеты. Действительная конечная скорость всегда будет меньше вследствие неизбежных потерь (преодоление силы тяготения при подъёме ракеты, сил аэродинамического сопротивления и др.). Выведенная в конце XIX века формула Циолковского и сегодня составляет важную часть математического аппарата, используемого при проектировании ракет, в частности при определении их основных массовых характеристик.

1.3. Задачи моделирования ракетных характеристик. Методы машинного обучения

Для оптимального выбора ХРТ, а также расчёта характеристик ракетных двигателей незаменимыми помощниками становятся различные подходы математического моделирования. В последнее время особую популярность получили так называемые методы машинного обучения, которые используются для решения различных регрессионных (предсказание численного значения) и классификационных (предсказания класса объекта) задач. В этой главе будет рассмотрен алгоритм работы одного из широко используемых методов машинного обучения, который был использован в данной работе. Выкладки, представленные ниже, будут основаны на материале онлайн-учебника от «Школы анализа данных», с которым можно ознакомиться по следующей ссылке [2].

Для решения различного рода нелинейных задач иногда полезным оказывается использование методов на основе случайного леса. Для красочного и лаконичного объяснения

работы данного алгоритма приведем пример из жизни древних греков: если много человек проголосуют независимо друг от друга, то вместе они придут к разумному решению, несмотря на то что опыт каждого из них субъективен. Аналогом голосования в мире машинного обучения является бэггинг, на основе которого работает случайный лес. Идея бэггинг может быть выражена следующей формулой:

$$a(x) = \frac{1}{k} (b_1(x) + b_2(x) + \dots + b_k(x)), \quad (12)$$

где a - ансамбль моделей;

x - входное значение;

k – количество базовых (простых) моделей;

b_i - какая-либо базовая (простая) модель.

Каждая простая модель обучается (т.е. параметризуется, или настраивается) для решения конкретной задачи не на всех имеющихся данных X , а на подвыборке из этих данных $X' \subset X$. После того как отдельные модели обучены, их можно использовать для моделирования значения (проведения голосования в примере с древними греками). Если решается задача предсказания численного значения, то в качестве выхода ансамбля принимается среднее значение от всех базовых моделей, как представлено в формуле (12). На этом принципе основана работа случайного леса, который был использован в данной работе для предсказания удельного импульса ракетного двигателя.

2. Практическая часть

2.1. Анализ литературных источников и сбор базы данных

Одной из самых значимых частей любой работы, связанной с использованием и дальнейшим применением методов машинного обучения, является предварительная работа с исходными данными. Поиск, грамотная обработка, а также анализ данных будет являться залогом успеха любого алгоритма, который использует процедуры обучения. Так как не имеется готовых баз данных по ХРТ, находящихся в свободном доступе, в данной работе были сделаны попытки ручного поиска данных по химическим топливам, а также показателям удельного импульса ракетных двигателей. Одними из критериев отбора источников данных являлись:

- авторитетность автора/места публикации,
- наличие внешней рецензии опубликованного материала.

В основном под эти критерии подходили справочные и учебные материалы [1, 3, 4, 5, 6, 7, 8], которые имели внешнюю рецензию квалифицированных специалистов (людей, имеющих научные степени в соответствующих областях).

По итогам работы была собрана база данных, состоящая из 132 индивидуальных неповторяющихся данных, которые включают следующую информацию:

- химический состав топлива,
- компьютерное представление химического состава в виде нотации *SMARTS* (описание представлено в подразделе 2.2),
- коэффициент объемного расширения продуктов сгорания топлива,
- удельный импульс топлива (в с).

Вся собранная база данных, как и код одностраничного web-сервиса (<http://momentum-predictor.ru/>) с имплементированной моделью по предсказанию удельного импульса двигателя доступны по следующей ссылке (https://github.com/Miros003004/impulse_predictor).

2.2. Компьютерное представление химического состава топлива

Для использования методов компьютерного моделирования необходимо представлять данные в виде машиночитаемого кода. Как правило, вся задача сводится к векторизации данных, т.е. представлению их в виде множеств числовых значений. Химические данные в этом плане не являются исключением, так как структурная формула или молекулярный граф непосредственно не являются машиночитаемыми представлениями. В данной работе мы использовали два подхода.

- На первом этапе был применён набор простых (с точки зрения вычисления) численных значений, или дескрипторов, которые так или иначе отражают химический состав и строение молекул топлива. В качестве таких дескрипторов были использованы значения молекулярной массы, количества атомов углерода в соответствующей гибридизации, количества замкнутых циклов и др. Для расчёта молекулярных дескрипторов использовался модуль Rdkit, названия всех используемых в данной работе дескрипторов могут быть найдены по ссылке [9].

- На втором этапе мы использовали молекулярные отпечатки пальцев (*molecular fingerprints*), работа которых основана на использовании алгоритмов хеширования данных. В этом случае молекулярная структура представляется в виде вектора определённой размерности. Важно, что конкретной молекуле алгоритм хеширования даёт всегда один и тот же вектор. Схематичное изображение работы алгоритма хеширования представлен на рисунке 3. В данной работе для этой задачи использовался модуль Rdkit, упомянутый выше.

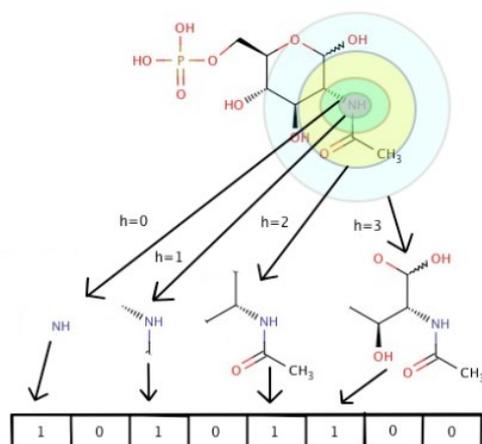


Рисунок 3. Принцип работы алгоритма хеширования для создания молекулярного отпечатка пальца.

Для более удобной и компактной записи химического состава использовалась нотация *SMARTS*, которая представляет собой строковую запись молекулярного состава топлива, где каждая индивидуальная молекула обособляется точками. Рассмотрим следующий пример:



Здесь представлены две молекулы (молекула водорода - $[\text{H}][\text{H}]$ и молекула воды - O), причём друг от друга они отделены точкой. Набор правил, по которым составляются *SMARTS* молекул, очень широк, однако существуют готовые инструменты, которые позволяют делать это автоматически. Стоит подчеркнуть, что представление *SMARTS* было использовано в настоящей работе только для удобства.

2.3. Результаты обучения моделей. Имплементация готового решения в виде web-ресурса

В качестве моделей для предсказания удельного импульса ракетного двигателя были выбраны следующие:

1. Алгоритм случайного леса,
2. Модель CatBoost, разработанная в рамках компании Яндекс,
3. Модель К-ближайших соседей.

Полное математическое описание соответствующих моделей выходит за рамки этой работы, однако было бы не лишним приведение основных принципов их работы. Модель случайного леса была подробно описана в подразделе 1.3, поэтому в этой части работы сразу перейдём к двум другим.

Алгоритм CatBoost является ансамблевым методом (по аналогии со случайным лесом), однако не берётся определённое заранее число простых моделей, а происходит их

постепенное добавление в процессе обучения. Данный алгоритм нашел широкое применение для решения различных регрессионных задач, к каковым относится и задача данной работы.

Принцип работы метода К-ближайших соседей можно уловить из названия и описать пословицей: «Скажи мне, кто твой друг, и я скажу, кто ты». Здесь ищутся группы данных, которые наиболее близко расположены друг к другу. Затем предсказание делается на основе того, к какой группе ближе находится данный объект.

Выбор соответствующих моделей был сделан, в первую очередь, в связи с их широким распространением в области классического машинного обучения, а также с наличием готовой реализации в библиотеке scikit-learn [10].

Для обучения использовалась собранная база данных, которая была разделена на обучающую, валидационную и тестовую подвыборки в соотношении 8:1:1 соответственно. Для оценки корректности выдаваемых моделью значений в данной работе использовался случайный регрессор. Принцип его действия основан на том, что он всегда выдаёт одно и то же число (например, среднее значение удельного импульса в обучающем наборе). Если какая-либо модель показывает худшую эффективность работы, чем случайный регрессор, она сразу «бракуется».

Стоит отметить, что были предприняты попытки использования дескрипторов не только на основе молекулярных отпечатков пальцев, но и на основе обычных молекулярных характеристик (молекулярная масса, количество атомов того или иного типа и ряда других, представленных в библиотеке rdkit). Данные обучения всех моделей представлены на рисунках 4 и 5.

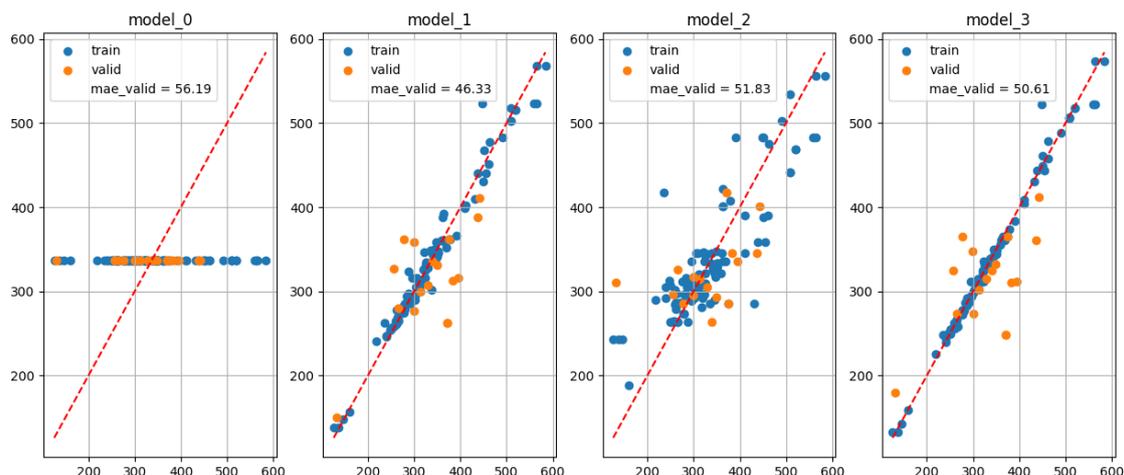


Рисунок 4. Результаты обучения моделей на молекулярных отпечатках пальцев (0, 1, 2 и 3 - случайный регрессор, случайный лес, алгоритм К-средних и CatBoost соответственно, синим цветом отмечены данные обучающего набора, оранжевым - валидационного; в легенде графика указана

средняя абсолютная ошибка в с).

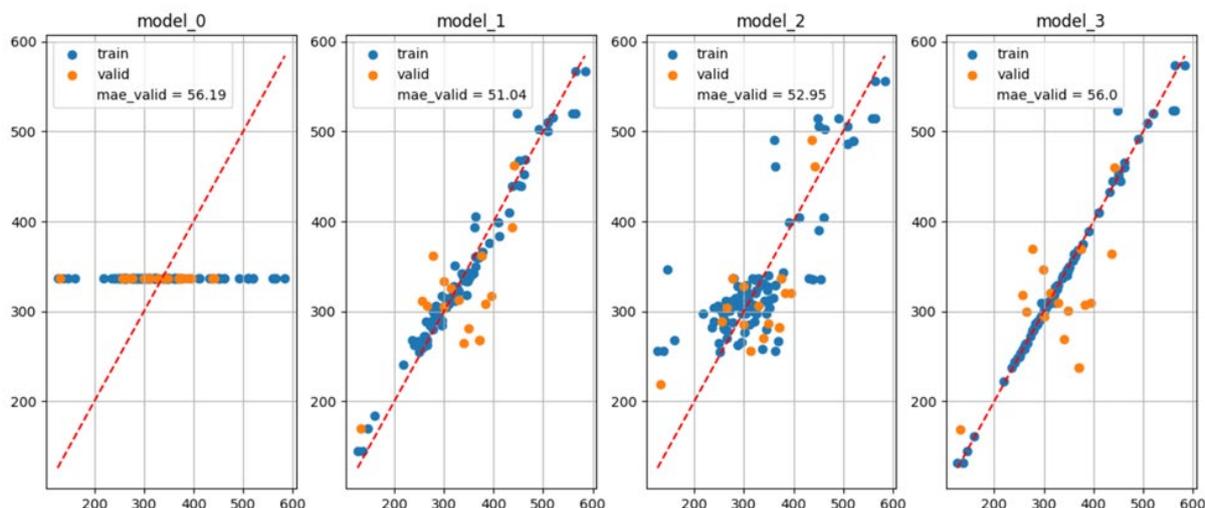


Рисунок 5. Результаты обучения моделей на простых молекулярных дескрипторах Rdkit (0, 1, 2 и 3 - случайный регрессор, случайный лес, алгоритм K-средних и CatBoost соответственно, синим цветом отмечены данные обучающего набора, оранжевым - валидационного; в легенде графика указана средняя абсолютная ошибка в с).

Для оценки качества прогноза использовалась такая метрика, как средняя абсолютная ошибка (MAE). Так как значение MAE на валидационном наборе было меньшим для модели случайного леса на молекулярных отпечатках пальцев (46.33 с), в дальнейшем использовалась именно эта модель.

Для создания удобного интерфейса был создан одностраничный web-сайт <https://momentum-predictor.ru/>, фрагмент которого представлен на рисунке 6. Web-ресурс работает с любой кроссплатформенной современной версией браузера, включая браузеры для операционных систем Android, Windows, Linux и Mac OS

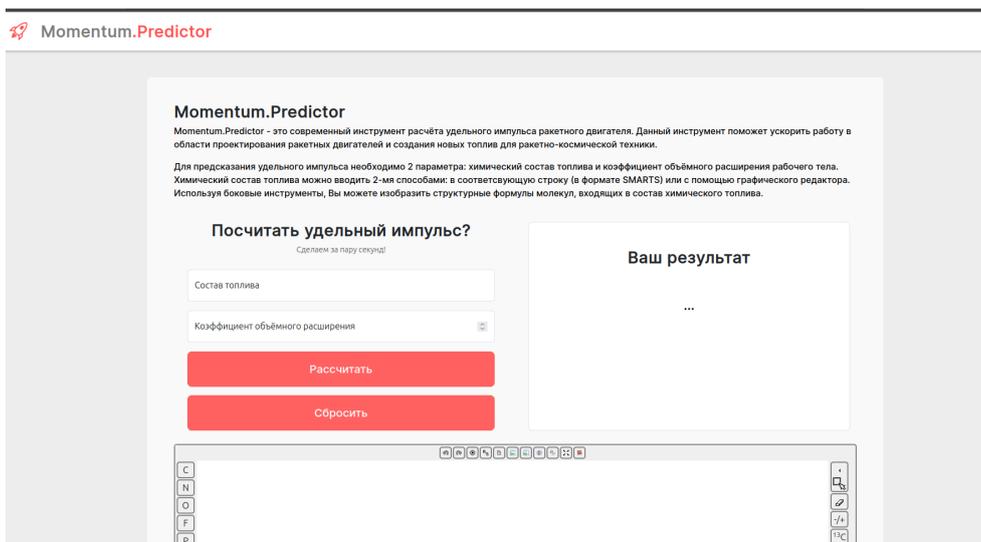


Рисунок 6. Фрагмент одностраничного сайта, где реализована модель случайного леса, полученная в ходе данного проекта

Для предсказания удельного импульса с помощью данного web-ресурса необходимо два параметра: химический состав топлива и коэффициент объёмного расширения рабочего тела. Химический состав топлива можно вводить двумя способами: в соответствующую строку (в формате SMARTS) или с помощью графического редактора. Используя боковые инструменты, можно изобразить структурные формулы молекул, входящих в состав химического топлива.

Выводы

В процессе выполнения работы была составлена база данных ракетных топлив, обучены модели на основе алгоритмов машинного обучения и создан web-сервис. Материалы работы загружены на платформу GitHub.

В ходе работы были опробованы три вида моделей:

1. Алгоритм случайного леса,
2. Модель CatBoost, разработанная в рамках компании Яндекс,
3. Модель K-ближайших соседей.

В качестве дескрипторов применялись коэффициент объёмного расширения рабочего тела и такие представления молекул, как молекулярные отпечатки пальцев и простые молекулярные характеристики. Качество предсказаний оценивалось по значению абсолютной ошибки. Результаты работы всех моделей сравнивались с результатами работы случайной модели.

Наилучший результат среди всех моделей показал алгоритм случайного леса, который работал на комбинации дескрипторов «fingerprint + коэффициент объёмного расширения».

Далее эта модель была интегрирована в web-сервис «Momentum.Predictor», с которым можно ознакомиться по ссылке <https://momentum-predictor.ru/>.

В ходе работы:

- 1) опробованы 3 модели на основе алгоритмов машинного обучения;
- 2) опробованы 2 комбинации дескрипторов;
- 3) составлена база данных видов ракетного топлива;
- 4) создан web-сервис «Momentum.Predictor»

ЗАКЛЮЧЕНИЕ

В ходе работы над данным проектом была создана модель на основе случайного леса, предсказывающая значение удельного импульса ракетного двигателя. Данная модель была реализована в виде одностраничного web-сервиса <https://momentum-predictor.ru/>, который заметно упрощает работу с обученным алгоритмом.

В дальнейшем планируется продолжение работы над данным проектом, в частности, будут использованы более продвинутые методы машинного обучения (например, глубокое обучение), позволяющие добиваться высоких предсказательных способностей соответствующих алгоритмов.

Отдельно хочется ещё раз подчеркнуть большой потенциал использования методов машинного обучения в различных областях науки и техники. В данной работе представлены классические алгоритмы машинного обучения, которые являются относительно простыми и, возможно, малоинтерпретируемыми. Однако для решения проблемы, поставленной в данном проекте, были использованы специальные представления молекул, которые в себе содержат информацию об атомах и их соседях (молекулярные отпечатки пальцев), поэтому результаты алгоритмов, представленных выше, можно считать вполне физически обусловленными.

СПИСОК СОКРАЩЕНИЙ

м ²	квадратный метр
м/с	метр в секунду
м/с ²	метр на секунду в квадрате
Н	ньютон
Па	паскаль
ХРД	химический ракетный двигатель
ХРТ	химическое ракетное топливо

СПИСОК ЛИТЕРАТУРЫ

1. Егорычев В.С. Топлива химических ракетных двигателей: учебное пособие [Текст] / В. С. Егорычев — Самара: СГАУ, 2007 — 71 с.
2. Артемьев М., Ашуха. А Учебник по машинному обучению / Артемьев М., Ашуха. А [Электронный ресурс] // Яндекс.Образование : [сайт]. — URL: <https://education.yandex.ru/handbook/ml> (дата обращения: 14.12.2024).
3. Алемасов В.Е., Дрегалин А.Ф., Тишин А.П. Теория ракетных двигателей [Текст] / В.Е.Алемасов, А.Ф.Дрегалин, А.П.Тишин — 4-е изд. — Москва: Машиностроение, 1989 — 462 с.\
4. Добровольский М. Ф. Жидкостные ракетные двигатели : основы проектирования [Текст] / М. Ф. Добровольский — 4-е изд. — Москва: Машиностроение, 1989 — 462 с.
5. Штехер М. С. Топлива и рабочие тела ракетных двигателей [Текст] / М. С. Штехер — Москва: Машиностроение, 1976 — 302 с.
6. Паушкин Я. М. Ракетные топлива [Текст] / Паушкин Я. М. — Москва: Мир, 1975 — 188 с.
7. Мелькумов, Т. М. Ракетные двигатели [Текст] / Т. М. Мелькумов — Москва: Машиностроение, 1968 — 511 с.
8. Васильев, А. П. Основы теории и расчёта жидкостных ракетных двигателей [Текст] / А. П. Васильев — 3-е изд. — Москва: Высш. шк., 1983 — 703 с.
9. RDKit: Open-Source Cheminformatics Software / [Электронный ресурс] // <https://rdkit.org/> : [сайт]. — URL: [www.rdkit.org/docs/source/rdkit.Chem.rdMolDescriptors.html](https://rdkit.org/docs/source/rdkit.Chem.rdMolDescriptors.html) (дата обращения: 20.12.2024).
10. scikit-learn Machine Learning in Python / [Электронный ресурс] // scikit-learn : [сайт]. — URL: <https://scikit-learn.org/stable/> (дата обращения: 20.12.2024).

ОТХОДЫ ОДНОЛЕТНИХ РАСТЕНИЙ В ПРОИЗВОДСТВЕ КОМПОЗИЦИОННЫХ МАТЕРИАЛОВ

Михайлов Владимир Сергеевич

Пушкино г, МБОУ "Софринский образовательный комплекс", 9 класс

Секция: Химические технологии

Научный руководитель: Пасько Юлия Вячеславовна, к.т.н., доцент каф. ЛТ9

МФ МГТУ им. Н.Э. Баумана

Аннотация

Цель работы – исследование возможности применения однолетних растений в производстве композиционных материалов с применением минеральных вяжущих.

Ежегодно в современном сельском хозяйстве образуется большое количество отходов однолетних растений (и, в частности, лузги подсолнечника), в связи с чем, возникает острая необходимость в их утилизации или переработке. Переработка с целью получения экологически безопасных композиционных материалов была и остается приоритетной. Известно, что данные отходы могут успешно конкурировать с древесным наполнителем при получении теплозвукоизоляционных материалов. Такие материалы могут быть получены на основе применения технологии изготовления арболита. В качестве связующих предложено минеральное вяжущее – цементное тесто, а в качестве наполнителя – лузга подсолнечника.

Работ по данному направлению практически нет, поэтому, с научной точки зрения, получение теплозвукоизоляционного материала из лузги подсолнечника и цементного вяжущего является весьма интересным и перспективным.

ВВЕДЕНИЕ

Ежегодно в современном сельском хозяйстве образуется большое количество отходов однолетних растений, например, *лузги семян подсолнуха*, в связи с чем, возникает острая необходимость в их утилизации или переработке.

Переработка с целью получения экологически безопасных композиционных материалов была и остается приоритетной.

Известно, что данные отходы могут успешно конкурировать с древесным наполнителем при получении теплозвукоизоляционных материалов. В результате применения однолетних растений возможна экономия древесины при получении композиционных материалов.

В качестве связующих при получении различных древесных материалов применяют синтетические смолы, которые, как известно, обладают токсичностью. Поэтому в данной работе предложено минеральное вяжущее – цементное тесто, которое, по сравнению с синтетическими смолами, токсичностью не обладает.

Таким образом, было решено изготовить теплозвукоизоляционный арболит, наполнителем у которого будет не измельченная древесина, а лузга подсолнечника в сочетании с цементом.

Цель данной работы – исследование возможности применения однолетних растений (лузги подсолнечника) в производстве композиционных материалов с применением минеральных вяжущих.

Для решения поставленной цели необходимо решение следующих **задач**:

1. Определить вид и количество недревесного наполнителя.
2. Разработать рецептуру композита на основе недревесного наполнителя и минерального вяжущего.
3. Изготовить композит на основе разработанной рецептуры.
4. Определить физико-механические свойства полученного композита.

1. Роль подсолнечника в современном мире

Подсолнухи ценятся не только за ядра, которые идут на изготовление растительного масла, широкое применение находит также побочная продукция, образующаяся при переработке семян: корзинки, шрот, жмых, лузга подсолнечника [1].

Лузга отделяется при подготовке семян подсолнечника к извлечению масла в процессе обрушивания, шрот – во время извлечения масла путем экстрагирования, а жмых образуется, если масло получают методом прессования. Эти отходы подсолнечника благодаря высокому

содержанию белка и витаминов считаются одними из самых ценных и относительно дешевых кормов для сельскохозяйственных животных [1].

При выделении ядра подсолнечника на маслопрессовых заводах образуется большое количество лузги – около 14% от общего количества семян. Утилизировать ее в таких объемах затратно и сложно. К счастью, сейчас существует несколько вариантов выгодной утилизации подсолнечной шелухи [1]: ее используют в качестве сырья при производстве фурфурола, кормовых дрожжей и этилового спирта; добавляют в измельченном виде к грубым кормам в животноводстве; применяют в садоводстве в качестве органического удобрения для разрыхления и улучшения почвы; шелуха подсолнечника приносит большую пользу при выращивании грибов; используют для получения биогаза [1].

Строительство является одной из наиболее важных отраслей отечественной экономики. Ее состояние во многом определяет уровень развития общества и его производственных сил. Связи с этим, можно считать возможность использования лузги подсолнечника в строительстве одним из самых выгодных вариантов утилизации подсолнечной шелухи [1]. Например, турецкие исследователи установили, что применение шелухи уменьшает плотность бетона и способствует увеличению сопротивления материала при резких перепадах температуры от минусовой к плюсовой и в обратную сторону [1].

2. Подсолнечная лузга как альтернативное сырье для получения арболита

В настоящее время, с увеличением потребления древесных ресурсов, возникает необходимость в разработке альтернативных источников сырья для производства древесных композиционных материалов с применением различных синтетических связующих и минеральных вяжущих, например, арболита и его ближайших аналогов (велокс, дюризол), цементно-стружечных плит (ЦСП), древесностружечных плит (ДСП) и других композитов. Одним из возможных альтернативных растительных сырьевых материалов является лузга подсолнечника, которая образуется в больших количествах при производстве масла в результате очистки семян подсолнечника.

В ходе изучения различных источников литературы анализировались химический состав древесины, лузги подсолнечника, а также особенности ее применения при получении древесных композиционных материалов, например, арболита.

Арболит представляет собой легкий бетон, полученный с применением цементного вяжущего, органических заполнителей (до 80-90% объема) и химических добавок. Его нередко называют древобетоном.

Обычно в состав арболита входят:

органический наполнитель в виде дробленки хвойных (или смеси хвойных и лиственных) пород древесины плоской, игольчатой структуры;

основной вяжущий компонент – минеральное вяжущее – портландцемент марок М400 или М500;

чистая техническая или водопроводная вода;

химические добавки, обеспечивающие сцепляемость (адгезионный контакт) органических наполнителей с цементным тестом и прочность готового материала.

Арболит, являющийся экологически безопасным строительным материалом, активно применяется для жилого, хозяйственного и производственного строительства. Обычно его используют в виде готовых строительных блоков или плит для возведения самонесущих стен или внутренних перегородок зданий, а также в качестве теплоизоляционного и звукоизоляционного материала.

Одним из основных сырьевых материалов, используемых в этом процессе, является древесная дробленка, получаемая из различных источников, включая рубки ухода за лесом и древесные отходы.

Леса являются не только источником сырья для производства строительных материалов, но и важным элементом поддержания здоровой экосистемы и устойчивой окружающей среды. Поэтому крайне важно осуществлять разумное потребление древесного сырья и по возможности сохранять лесные ресурсы. В связи с этим, проводятся различные исследования с целью нахождения заменителей (альтернативных источников) древесного сырья для производства композитных плитных материалов [2].

Недревесные лигноцеллюлозные материалы и сельскохозяйственные отходы привлекают все больше внимания в качестве альтернативного сырья. Они образуются из возобновляемых растительных источников и являются невостребованным отходом, что позволяет использовать их в качестве дополнительного источника дохода и снизить экологическую нагрузку на окружающую среду.

Одним из видов не древесного растительного сырья являются отходы переработки подсолнечника, в частности – плодовые оболочки (лузга).

Подсолнечник (однолетний) имеет относительно короткий цикл роста и легко адаптируется к различным почвенным условиям, поэтому выращивается во всем Мире и является одной из наиболее важных масличных культур.

Лузга используется для изготовления топливных пеллет, применяется в качестве основы для выращивания грибов, а также как добавка в корм для животных [2-5]. Использование подсолнечной лузги в качестве сырья для производства композиционных плитных материалов предоставляет возможность сократить использование древесного сырья

и достигнуть значительного экологического эффекта путем утилизации огромных объемов лузги, которые в настоящее время скапливаются в отвалах.

Химический состав и свойства подсолнечной лузги

Лузга составляет 21-30% от массы семян и имеет средние размеры частиц по длине 10,59 мм, ширине 3,12 мм, толщине 0,26 мм. Внешний вид подсолнечной лузги представлен на рисунке 1.



Рисунок 1 – Размер подсолнечной лузги

Плотность лузги очень низкая и составляет 212 кг/м³ при 10% влажности [6]. В основном состав древесины включает целлюлозу, гемицеллюлозу, лигнин, экстрактивные вещества и некоторые минеральные вещества [2, 7].

В таблице 1 представлено процентное содержание компонентов в зависимости от породы дерева [2, 8].

Таблица 1 – Содержание основных компонентов древесины в зависимости от породы

Компонент	Порода древесины	
	Хвойная	Лиственная
	Содержание, масс. %	Содержание, масс. %
Целлюлоза	42-51	42-49
Гемицеллюлоза	27-40	24-30
Лигнин	18-24	25-30
Экстрактивные вещества	1-10	2-9
Зола	0,2-0,8	0,2-0,8

В плодовых оболочках подсолнечника находятся целлюлоза, гемицеллюлоза и лигнин, а также клетчаточный каркас, обладающий сорбционными свойствами. Химический состав лузги может быть нестабильным в зависимости от сортовых особенностей и почвенно-климатических условий выращивания. В таблице 2 представлены сводные литературные данные по химическому составу лузги.

Таблица 2 – Химический состав плодовых оболочек подсолнечника

Состав	% масс.	Источник
<i>Экстрактивные вещества</i>		

Экстрагируемые бензолом	1,1	[2, 9]
Экстрагируемые водой	13,8	[2, 10]
<i>Органические вещества</i>		
Целлюлоза	20,0-67,0	[2, 10-13]
Гемицеллюлоза	18,0-35,0	[2, 11, 12]
Холоцеллюлоза	62,5	[2, 10]
Гексозаны	33,8	[2, 9]
Пентозаны	26,0-32,0	[2, 13]
Липиды	0,5-5,2	[2, 14-16]
Лигнин		
Метоксилы	23,0-35,0	[2, 10]
Гидроксилы		
<i>Функциональные группы</i>		
Метоксильные группы	4,6-4,9	[2, 9]
Карбоксильные группы	1,8	[2, 17]

Целлюлоза и гемицеллюлоза являются скелетом. Соответственно значительная прочность достигается при высоком содержании целлюлозы [2, 18]. Гемицеллюлоза же приводит к водопоглощению благодаря своим гидрофильным свойствам.

Авторы [2, 19] отмечают, что содержание липидов в лузге обратно пропорционально доле целлюлозы. Согласно [2, 20], содержание липидов в группе высокомасличных сортов колеблется в диапазоне от 2,7 до 2,8%. В состав липидов плодовой оболочки входят воскоподобные вещества, количество которых составляет от 0,28% до 2,34% от общей массы лузги [2, 21-23].

Эти воскообразные вещества, покрывающие поверхность сельскохозяйственных отходов, таких как подсолнечная лузга, могут оказывать негативное влияние на процесс осмоления связующими и вяжущими при изготовлении композиционных материалов.

Черный слой, покрывающий внешнюю поверхность плодовой оболочки, является – *фитомеланином*, выступающем в роли защитного барьера семян. Он не имеет устойчивой структуры, а его состав точно не установлен [2, 24, 25].

Особенности подсолнечной лузги, как сырья для изготовления ДСП

Специфическая форма подсолнечной лузги, а именно форма *лодочки*, оказывает негативное воздействие на прочность готовых изделий. Именно из-за этой формы могут образовываться участки микродефектов – *пустот*. Пустоты крупным планом показаны на рисунке 2.

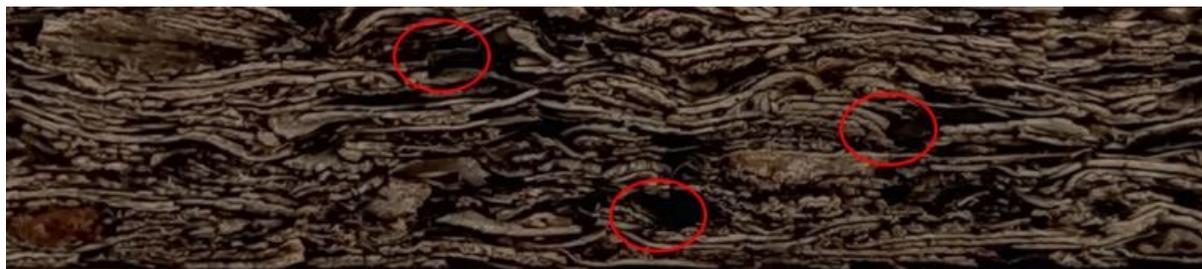


Рисунок 2 – Пустоты в плитном материале или арбалитовом блоке

Небольшие линейные размеры подсолнечной лузги, также не способствуют хорошим прочностным показателям.

Одним из ключевых факторов, определяющих физико-механические свойства древесных материалов, является геометрия используемых частиц. По результатам поисковых исследований литературы было установлено, что эталонной длиной частиц считается диапазон 20-40 мм [2, 26]. Именно в этом диапазоне можно достичь хороших показателей прочности готовой продукции и исключить проблемы более длинных частиц, а именно от 40 мм при их осмолении.

Кроме того, толщина частичек также оказывает значительное влияние на физико-механические свойства древесных плит. С уменьшением толщины частиц увеличивается прочность плит, что обусловлено более эластичной структурой частиц и увеличением площади контакта между древесными частицами. А это, в свою очередь, способствует повышению прочности плит и уменьшению расхода связующего на единицу поверхности [2, 27]. Таким образом, геометрические параметры сырья оказывают существенное влияние на физико-механические свойства древесных материалов и должны учитываться при их производстве.

По итогам изучения литературных данных и анализируя подведенные авторами итоги, можно выделить основные параметры, по которым возможно использование такого альтернативного вида сырья – лузги подсолнечника, но с учетом выявленных на стадии изучения факторов:

1. Рассмотреть возможность химической ликвидации поверхностного (*воскового*) слоя для улучшения покрытия частиц цементным тестом.
2. Необходимо изменить форму лодочки, придать ей более плоский вид, для устранения областей микродефектов и повышения прочностных показателей готовых плит.
3. Использование подсолнечной лузги в качестве альтернативного сырья вместо древесины для производства композиционных плитных материалов, может привести значительный экологический эффект и экономическую выгоду для предприятий [2, 13].

3. Методические положения экспериментальных исследований

Данная глава посвящена методикам проведения экспериментов в настоящей работе. Здесь представлены методики изготовления арболита из лузги подсолнечника и цементного вяжущего и определения основных физико-механических свойств полученного теплозвукоизоляционного композита.

Арболит из лузги подсолнечника и цементного вяжущего изготавливался в лаборатории кафедры ЛТ9-МФ МГТУ им. Н.Э. Баумана.

Размеры полученного образца 100×100×100 мм.

3.1 Методика получения арболита из лузги подсолнечника и цементного вяжущего

Сначала была разработана рецептура арболитовой смеси, которая включала в себя: лузгу, цемент, воду и химические добавки для улучшения твердения цемента.

Затем был произведен замес арболитовой смеси: сначала взвесили лузгу, которую обработали раствором жидкого стекла, ввели цемент и добавили воду для затворения цемента. Тщательно перемешали все компоненты до получения однородной смеси.

Далее смесь утрамбовали в стандартной пресс-форме, объемом 1 кубический дециметр. Форму закрыли и оставили на 28 суток в сухом отапливаемом помещении для полного схватывания и твердения цементного теста.

3.2 Определение физико-механических свойств арболита

Все испытания полученного арболита проводились в машинном зале лаборатории кафедры ЛТ9-МФ МГТУ им. Н.Э. Баумана.

По прошествии 4-х недель после изготовления лабораторного образца арболита определили плотность, влажность и прочность на сжатие полученного материала.

Необходимый набор инструментов и оборудования: линейка, штангенциркуль, электронные весы и испытательная машина для определения предела прочности на сжатие.

Определение плотности арболита

Штангенциркулем измеряют размеры полученного образца арболита с точностью до 0,1 мм, вычисляют его объем, а затем взвешивают эти образцы на электронных весах [28].

За окончательный результат измерений образцов кубической формы принимают среднее арифметическое трех измерений каждой грани [28].

Объем образца, см^3 , имеющего форму куба или параллелепипеда считают по формуле (1) [28]:

$$V = a \times b \times h, \text{ см}^3, \quad (1)$$

где a , b и h – средние значения размеров граней образца, см.

Зная объем и массу образца, среднюю плотность арболита рассчитывают по формуле (2) [28]:

$$\rho = m / V, \text{ г/см}^3, \quad (2)$$

где m – масса образца арболита, г [28].

Определение предела прочности при сжатии арболита

Образцы для испытания могут иметь форму цилиндра, параллелепипеда с квадратным или прямоугольным основанием. В данном случае образец полученного арболита обладает квадратным основанием, поскольку имеет форму куба. Опорные плоскости образцов должны быть параллельными в пределах допуска 0,1% от высоты образца в направлении, перпендикулярном направлению приложения нагрузки. Для испытания были взяты образцы размером $100 \times 100 \times 100$ мм. Перед испытанием измеряют высоту, ширину, толщину образца с погрешностью не более 0,01 мм. Образец располагают на опорных площадках (рисунок 3) таким образом, чтобы продольная ось образца совпадала с направлением действия нагрузки [28].

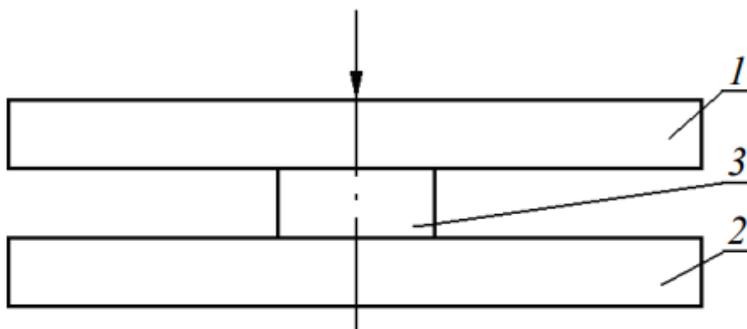


Рисунок 3 – Схема испытания образца арболита на сжатие:

1, 2 – площадки испытательной машины, 3 – образец арболита [28]

Устанавливают необходимую величину скорости сближения площадок. После выполнения подготовительных операций испытуемый образец подвергают действию возрастающей нагрузки до его разрушения. По шкале машины отсчитывают величину нагрузки в единицах килограмм-силы (кгс), при которой произошло разрушение образца. Предел прочности при сжатии $\sigma_{сж}$, МПа, рассчитывают по формуле (3) [28]:

$$\sigma_{сж} = P / F, \text{ МПа}, \quad (3)$$

где P – разрушающее усилие, Н;

F – площадь поперечного сечения образца, к которой приложено разрушающее усилие, м² [28].

Определение влажности арболита

После испытаний на сжатие приступают к определению влажности арболита. Для этих целей отбирается навеска весом примерно 100 г, помещается в термощкаф для высушивания до постоянной массы при температуре 105±5 °С. Влажность считается по формуле (4):

$$W = \frac{(g_0 - g)}{g} \times 100, \% \quad (4)$$

где g_0 – первоначальная масса навески, г;

g – масса навески после сушки, г.

4. Экспериментальная часть научных исследований

Сначала была разработана рецептура арболитовой смеси, которая включала в себя: лузгу подсолнечника, цемент, воду и химические добавки для улучшения твердения цемента. В качестве химической добавки использовалось натриевое жидкое стекло. Рецептура арболитовой смеси представлена в таблице 3.

Таблица 3 – Состав арболитовой смеси из лузги подсолнечника, жидкого стекла, цемента и воды

№ п/п	Наименование компонента	Количество компонента
1.	Лузга подсолнечника, г	320
2.	Цемент марки М-500, г	410
3.	Жидкое стекло натриевое, г	90
4.	Вода, г	193

Затем был произведен замес арболитовой смеси: сначала взвесили лузгу, которую обработали раствором жидкого стекла, ввели цемент и добавили воду для затворения цемента. Тщательно перемешали все компоненты до получения однородной смеси.

Лузга обрабатывается раствором жидкого стекла с целью улучшения адгезионного контакта между лузгой и цементом. Если этого не сделать, то возможно получение такой ситуации, при которой целостная структура материала не сможет образоваться, и при распалубке такое изделие просто рассыплется на отдельные компоненты. Кроме того, натриевое жидкое стекло обладает обволакивающим действием, увеличивает прочность

арболита, ускоряет твердение цемента, увеличивает огнестойкость арболита и является хорошим антисептиком.

Далее смесь утрамбовали в прессформе, объемом 1 кубический дециметр. Форму закрыли и оставили на 28 суток для полного схватывания и твердения цементного теста.

По прошествии 4-х недель определили плотность, влажность и прочность на сжатие полученного арболита. Расчеты показателей качества полученного арболита определялись по методикам, представленным в предыдущей главе работы.

Физико-механические свойства арболита из лузги подсолнечника и цементного теста представлены в таблице 4.

Таблица 4 – Физико-механические свойства арболита из лузги подсолнечника и цемента

№ п/п	Наименование параметра	Значение
1.	Плотность образца, кг/м ³	780
2.	Предел прочности при сжатии, МПа	1,03
3.	Влажность образца, %	11
4.	Класс арболита, в зависимости от прочности на сжатие	B0,75
5.	Марка арболита по прочности на сжатие	M10
Класс арболита B0,75 и марка M10 относят его к теплоизоляционным материалам		

Проведенные испытания показали, что полученный арболит имеет достаточно высокую прочность и может применяться не только как изоляционный, но и как конструкционный материал при строительстве малоэтажных сооружений.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данного научного эксперимента была разработана технология получения арболита, состоящего из лузги подсолнечника и минерального (цементного) вяжущего. Плотность этого материала соответствует необходимым требованиям.

Проведенные испытания показали, что арболит, полученный с применением в качестве наполнителя лузгу подсолнечника, имеет достаточно высокую прочность и может применяться не только как изоляционный, но и как конструкционный материал при строительстве малоэтажных сооружений различного назначения. Кроме того, полученный материал относится к экологически чистым материалам.

Следовательно, на данном этапе цель и задачи настоящей научно-исследовательской работы выполнены, однако, в дальнейшем существует необходимость для углубления и расширения полученных знаний в этом направлении.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шелуха семечек как материал для производства ДСП [Электронный ресурс]. Режим доступа: <https://school-science.ru/2/1/29805?ysclid=m7uvvk21bu137276884>. (Дата обращения: 05.03.2025 г.).
2. Подсолнечная лузга – как альтернативное сырье для производства древесностружечных плит [Электронный ресурс]. Режим доступа: <https://na-journal.ru/4-2023-tehnologii-proizvodstva/4975-podsolnechnaya-luzga-kak-alternativnoe-syrye-dlya-proizvodstva-drevesnostrujehnyh-plit?ysclid=m7uvlb2xld822929343>. (Дата обращения: 04.03.2025 г.).
3. Sunflower hulls as a component of feeds / A. Robertiello, L. Angelini, L. Conte, F. Sciaraffia, P.L. Genevini, S. Pialorsi // *Agricultural Wastes*. 1984. Vol. 10. P. 257-266.
4. Получение кормовых экструдатов на основе подсолнечной лузги / В.Г. Коротков, С.В. Кишкилев, С.Ю. Соловых, С.В. Антимонов // *Технические науки – от теории к практике*. 2013. №18. С. 124-131.
5. González Matute, R. Sunflower seed hull based compost for *Agaricus blazei* Murrill cultivation / R. González Matute, D. Figlas, N. Curvetto // *International Biodeterioration & Biodegradation*. 2010. Vol. 64. С. 742-747.
6. Gamea G.R. Physical properties of sunflower seeds components related to kernel pneumatic separation // *International Journal of Engineering & Technology*. 2013. Т.13. №1. С. 103-114.
7. Терентьева Э.П., Удовенко Н.К., Павлова Е.А. Химия древесины, целлюлозы и синтетических полимеров: учебное пособие / СПбГТУРП. СПб., 2015 Ч.2 83 с.
8. Lohmann, U. *Holzlexikon*, 4th ed.; Nikol Verlagsgesellschaft mbH & Co. KG: Hamburg, Germany, 2015.
9. Справочник по производству древесностружечных плит / И.А. Отлев [и др.]. 2-е изд., перераб. и доп. М.: Лесная промышленность, 1990. 384 с.
10. Wood handbook – Wood as an engineering material / D. Bender [et al.]. Madison, WI: U.S. Department of Agriculture, Forest Service, Forest Products Laboratory. 463 p.
11. Волынский, В.Н. Технология стружечных и волокнистых древесных плит / В.Н. Волынский. Таллинн: Дезидерата, 2004. 192 с.
12. Курдюмова, В.М. Исследование и разработка технологии изготовления плит из стеблей хлопчатника: дис. ...канд. техн. наук: 05.21.05 / Курдюмова Валентина Мифодьевна. Л., 1981. 220 с.
13. Turzyński Tomasz, et al. "Comparative analysis of pelletized and unpelletized sunflower husks combustion process in a batch-type reactor." *Materials*10 (2021): 2484.
14. Pandey, A.K. Phytomelanin in Compositae / A.K. Pandey, M.R. Dhakal // *Current Science*. 2001. Vol. 80, N8. P. 933-940.
15. О структуре лузги и панцирности подсолнечника: отчет о НИР ВНИИЭМК / Ильина А.И. Краснодар: Советская Кубань, 1958. С. 129-132.
16. Arromdee, P. A comparative study on combustion of sunflower shells in bubbling and swirling fluidized-bed combustors with a cone-shaped bed / P. Arromdee, V.L. Kuprianov // *Chemical Engineering and Processing*. 2012. Vol. 62. P. 26-38.
17. Красильников, В.Н. О содержании воскообразных веществ в липидах лузги высокомасличного подсолнечника / В.Н. Красильников, В.П. Ржехин, Т.В. Карасев // *Масложировая промышленность*. 1972. №3. С. 17-19.

18. Baharoğlu M. et al. Effects of anatomical and chemical properties of wood on the quality of particleboard // *Composites Part B: Engineering*. 2013. Т. 52. С. 282-285.
19. Щербаков, В.Г. Сорбционные свойства плодовой оболочки современного подсолнечника к растительному маслу и другим неполярным органическим жидкостям / В.Г. Щербаков, С.Ю. Ксандопуло, А.В. Александрова // *Известия вузов. Пищевая технология*. 2003. №5-6. С. 27-30.
20. Дублянская, Н.Ф. Состав и биохимические свойства околоплодника высокомасличных сортов подсолнечника / Н.Ф. Дублянская // *Доклады Всесоюзной ордена Ленина Академии сельскохозяйственных наук им. В. И. Ленина*. 1967. №5. С. 5-8.
21. Щербаков, В.Г. Липиды лузги семян подсолнечника и их влияние на качество масла / В.Г. Щербаков, И.П. Кудинов // *Масложировая промышленность*. 1967. №7. С. 11-14.
22. Красильников, В.Н. О содержании воскообразных веществ в липидах лузги высокомасличного подсолнечника / В.Н. Красильников, В.П. Ржехин, Т.В. Карасев // *Масложировая промышленность*. 1972. №3. С. 17-19.
23. Изучение восков подсолнечных семян / В.П. Ржехин, В.Н. Красильников, Т.В. Карасева, Н.А. Недачина, Э.И. Горшкова // *Масложировая промышленность*. 1968. №7. С. 13-16.
24. Pandey A.K., Dhakal M.R. Phytomelanin in compositae // *Current Science*. 2001. С. 933-940.
25. Kumar J.B., Kumar M.S. A general idea about phytomelanin layer in some Compositae // *Indian Journal of Plant Sciences*. 2013. Т.2. №. С. 44-48.
26. Модлин, Б.Д. Изготовление стружки для древесно-стружечных плит / Б.Д. Модлин, А.А. Хатилович. М.: Лесная промышленность, 1988. 152 с.
27. Грибенчикова, А.В. Материаловедение в производстве древесных плит и пластиков / А.В. Грибенчикова. М.: Лесная пром-ть, 1988. 120 с.
28. Соловьева, Т.В. Технология древесноволокнистых плит, технология древесностружечных плит, технология композиционных материалов и пластиков. Лабораторный практикум: учеб.-метод. пособие для студентов специальности 1-48 01 05 «Химическая технология переработки древесины» специализации 1-48 01 05 02 «Технология древесных плит и пластиков» / Т.В. Соловьева, А.А. Пенкин. Минск: БГТУ, 2009. 144 с.

ПРИЛОЖЕНИЕ

1 слайд

Олимпиада школьников «Шаг будущее»
по профилю «Химические технологии»

Научный проект на тему

**«Отходы однолетних растений в производстве
композиционных материалов»**

Руководитель: Пасько Юлия Вячеславовна,
кандидат технических наук, доцент кафедры ЛТ9
«Химия и химические технологии в лесном комплексе» МФ МГТУ им. Н. Э. Баумана

Докладчик: Михайлов Владимир,
ученик 9Г класса, МБОУ «Софринский образовательный комплекс» МО

Мытищи-2025 г.

2 слайд

Цель работы – исследование возможности применения однолетних растений в производстве композиционных материалов с применением минеральных вяжущих.

Для решения поставленной цели необходимо было решить следующие задачи:

1. Определить вид и количество недревесного наполнителя.
2. Разработать рецептуру композита на основе недревесного наполнителя и минерального вяжущего.
3. Изготовить композит на основе разработанной рецептуры.
4. Определить физико-механические свойства полученного композита.

2

3 слайд

Химический состав лузги подсолнечника:

- жиры — 3%;
- белки — 3,4%;
- безазотистые экстрактивные вещества — 29,7%;
- клетчатка — 61,1%;
- зола — 2,83%.

Основным компонентом лузги является **клетчатка** — сплетение растительных волокон. Она состоит из лигнина, целлюлозы и гемицеллюлозы.

Целлюлоза — 30–45%.

Гемицеллюлозы — 25–35%.

Лигнин — 25–30%



4 слайд

Знакомство с арболитовой продукцией



5 слайд

Разработка рецептуры смеси для изготовления арболита из лузги подсолнечника и цементного теста



№ п/п	Наименование компонента	Количество
1.	Лузга подсолнечника, г	320
2.	Цемент марки М-500, г	410
3.	Жидкое стекло натриевое, г	90
4.	Вода, г	193

Порядок загрузки компонентов:

1. Лузга подсолнечника.
2. Жидкое стекло натриевое.
3. Цемент.
4. Вода.

6 слайд



Введение жидкого стекла в лузгу подсолнечника

Подготовка цемента марки М-500



7 слайд



Совмещение и перемешивание всех компонентов арболитовой смеси

Укладка и трамбовка арболитовой смеси в пресс-форму, объемом 1 куб. дециметр



8 слайд



Замыкание пресс-формы с уплотненной арболитовой смесью

Арболитовая смесь в пресс-форме «зрела» до готового изделия 28 суток



9 слайд

Арболит из лузги подсолнечника и цемента по прошествии 28 суток и после распалубки, полностью готовый к определению физико-механических свойств



The image shows a young man in a green lab coat holding a rectangular sample of arbolite. To his right is a close-up photograph of a large, textured, cylindrical block of arbolite, which is a composite material made of sunflower husks and cement.

10 слайд

Определение плотности полученного арболита: обмер линейных размеров образца и его массы.



The image contains two photographs. The left photograph shows a young man in a green lab coat measuring the dimensions of a rectangular arbolite sample on a workbench. A woman in a white lab coat is standing nearby. The right photograph shows the same young man weighing the arbolite sample on a scale.

11 слайд

Определение предела прочности при сжатии арболита



12 слайд

Определение влажности арболита и подсчет результатов эксперимента



Физико-механические свойства арболита из лозги подсолнечника и цемента

№ п/п	Наименование параметра	Значение
1.	Плотность образца, кг/м ³	780
2.	Предел прочности при сжатии, МПа	1,03
3.	Влажность образца, %	11
4.	Класс арболита, в зависимости от прочности на сжатие	B0,75
5.	Марка арболита по прочности на сжатие	M10

Класс арболита B0,75 и марка M10 относят его к теплоизоляционным материалам

13 слайд

ВЫВОД

Проведенные испытания показали, что полученный арболит имеет достаточно высокую прочность и может применяться не только как изоляционный, но и как конструкционный материал при строительстве малоэтажных сооружений.

Цель и задачи выполнены.

Благодарю за внимание!

10

РОБОТ-СПАСАТЕЛЬ «ТОТОШКА» ДЛЯ ПОМОЩИ ПРИ УСТРАНЕНИИ ПОСЛЕДСТВИЙ ЧС

Мухин Владимир Геннадьевич

Звенигород г, МБОУ "Первая школа имени М.А. Пронина" г. Звенигород, 11 класс

Секция: Робототехнические системы

Работа выполнена без научного руководителя

Аннотация

Целью работы является разработка и создание прототипа робота-спасателя, пульта управления и системы передачи видеоизображения с робота на рабочее место оператора. При работе над проектом были использованы следующие методы и приёмы: поиск информации в интернете и её анализ, определение основных узлов робота, подбор необходимых компонентов, 3D-моделирование в САПР Компас-3D и печать на 3D-принтере, программирование в средах разработки Arduino IDE и Visual Studio Code. Разработанный робот имеет 4-х колёсную базу, 2 дополнительных откидных колеса, манипулятор, а также систему видеосвязи. Несмотря на небольшие габариты (ДхШхВ) 39х30х34 см и массу 5.4 кг, робот может преодолевать препятствия высотой до 22 см, подниматься по наклонной плоскости (угол наклона до 40 градусов) и стандартной человеческой лестнице (высота ступени 150 мм, ширина – 340 мм), имеет грузоподъёмность около 5 кг. Удобное управление позволяет выполнять точные действия колёсной тележкой и манипулятором. Испытания робота на специальном полигоне подтвердили удачность принятых технических решений и надёжность конструкции. Разработанный робот может быть полезен для помощи спасателям при устранении последствий ЧС и использоваться при выполнении таких задач, как разведка местности и оценка повреждений/ущерба, поиск людей и животных в зоне ЧС, доставка полезной нагрузки в зону обрушений, перемещение подозрительных предметов, установка ретрансляторов для увеличения дальности связи, перекрытие вентилей

ВВЕДЕНИЕ

Актуальность и практическая значимость проекта

В последнее время роботы стали довольно часто использоваться для помощи спасателям при устранении последствий ЧС. В арсенале МЧС есть довольно много различных робототехнических комплексов (РТК), но большинство из них имеют слишком большие габариты и массу. Поэтому зачастую их транспортировка одним человеком невозможна и для этого необходимы специальные прицепы и фургоны. Также РТК и запасные части для них имеют очень большую стоимость, из чего следуют большие расходы на обслуживание.

Чтобы решить эту проблему, я решил разработать робота небольшого размера и массы, простого и дешёвого в использовании и обслуживании.

Целью работы является разработка и создание прототипа робота-спасателя небольших размеров, пульта управления и системы передачи видеоизображения с робота на рабочее место оператора. Робот может использоваться при выполнении небольших, но важных задач, таких как:

- разведка местности и оценка повреждений/ущерба;
- поиск людей и животных в зоне ЧС;
- доставка полезной нагрузки в зону обрушений;
- перемещение подозрительных предметов;
- установка ретрансляторов для увеличения дальности связи;
- перекрытие вентиляей.

Новизна работы заключается в создании малогабаритного бюджетного робота с повышенной проходимостью, способного работать в экстремальных условиях окружающей среды.

Для достижения цели проекта необходимо решить *ряд задач*: провести расчёты узлов робота и спроектировать их, подобрать комплектующие, сделать 3D-модели необходимых деталей, собрать робота, сделать пульт управления, написать программу, протестировать готовую систему, разработать и установить систему для передачи и приёма видеоизображения с камер робота, провести сравнение созданного робота с уже существующими роботами.

При работе над проектом были использованы следующие *методы*:

- поиск информации в интернете и её анализ;
- определение основных узлов робота;
- подбор необходимых компонентов;

3D-моделирование в САПР Компас-3D и печать на 3D-принтере;
программирование в средах разработки Arduino IDE и Visual Studio Code.

В разделе 1 описывается процесс разработки робота. Выделены основные узлы: ходовая часть, манипулятор, откидные колёса (хвост), датчики и система видеосвязи с роботом. Приведена таблица с расчётами параметров моторов в зависимости от диаметра колёс. Описан механизм манипулятора и хвоста. Приведены возможные автономные действия, выполняемые роботом по показания датчиков. Описывается система передачи видеоизображения на рабочее место оператора.

В разделе 2 описывается пульт управления роботом. Выделены основные электронные составляющие пульта. Приведена схема питания и взаимодействия электронных компонентов, а также краткая инструкция по управлению роботом.

В разделе 3 приводится таблица сравнения характеристик созданного робота с другими роботами. Также рассказывается о планах на будущее.

Разработка робота

Ходовая часть

Основа любого сухопутного робота – ходовая часть. Она может быть колесной, гусеничной и шагающей. Рассмотрим плюсы и минусы каждой из них в отдельности.

Преимущества колесной базы: скорость, плавность хода, хорошее сцепление, простота изготовления. Главным недостатком является маленькая площадь соприкосновения с поверхностью, а следовательно, и плохое распределение давления. Преимущества гусеничной базы, наоборот, в хорошем распределении давления на единицу площади поверхности, а главный недостаток – сложность изготовления качественных гусениц и большие габариты робота вследствие необходимости подъема по лестнице (при подъёме робот должен цепляться минимум за 3 ступени). Шагающий робот наиболее сложен в изготовлении из перечисленных баз.

Я выбрал колёсный вариант. Колесная база по типу подвески делится на несколько видов: зависимая, полунезависимая и независимая.

После анализа разных типов подвески мой выбор остановился на 4-х колёсной базе с жесткой подвеской, так как этот вариант наиболее прост в реализации, имеет большую жёсткость конструкции, сравнительно небольшой вес, а также хорошую ремонтпригодность.

Таблица 1 – Расчёт параметров моторов в зависимости от диаметра колёс

Диаметр колеса, мм	75	100	125	130
Длина окружности колеса, м	0,236	0,314	0,393	0,408
Скорость рабочая				
Скорость, м/с	0,40	0,40	0,40	0,40
Скорость, м/мин	24	24	24	24
Обороты вала, об/мин	102	76	61	59
Скорость минимальная				
Скорость, м/с	0,01	0,01	0,01	0,01
Скорость, м/мин	0,60	0,60	0,60	0,60
Обороты вала, об/мин	2,5	1,9	1,5	1,5
Скорость максимальная				
Скорость, м/с	0,75	0,75	0,75	0,75
Скорость, м/мин	45	45	45	45
Обороты вала, об/мин	191	143	115	110
Масса робота, кг	5,4	5,4	5,4	5,4
Запас по мощности	20%	20%	20%	20%
Крутящий момент (3 двигателя), кг·см	8,1	10,8	13,5	14,0
Крутящий момент (4 двигателя), кг·см	6,1	8,1	10,1	10,5

Исходя из расчётов, приведённых в таблице выше, был выбран диаметр колёс 125 мм и коллекторные моторы со следующими характеристиками:

- скорость вращения вала 110 оборотов в минуту;
- крутящий момент 14 кг·см;
- рабочий ток до 1А;

- ток при блокировке колеса 3.8 А.

Также выбранные моторы имеют смещенный вал, что позволяет увеличить дорожный просвет робота.

Драйверы моторов выбрал VNH2SP30 [1]. Один такой драйвер может выдавать ток до 30 А. Было решено поставить два драйвера – по одному на каждую сторону робота. Ходовая часть при этом управляется по танковой схеме.

Геометрия колёсной базы (расстояние между колёс, дорожный просвет, центр тяжести) была подобрана исходя из необходимости подъема по лестнице с высотой ступени 150 мм и шириной 340 мм (стандартные размеры человеческой лестницы), а также эффективной езды по пересечённой местности.

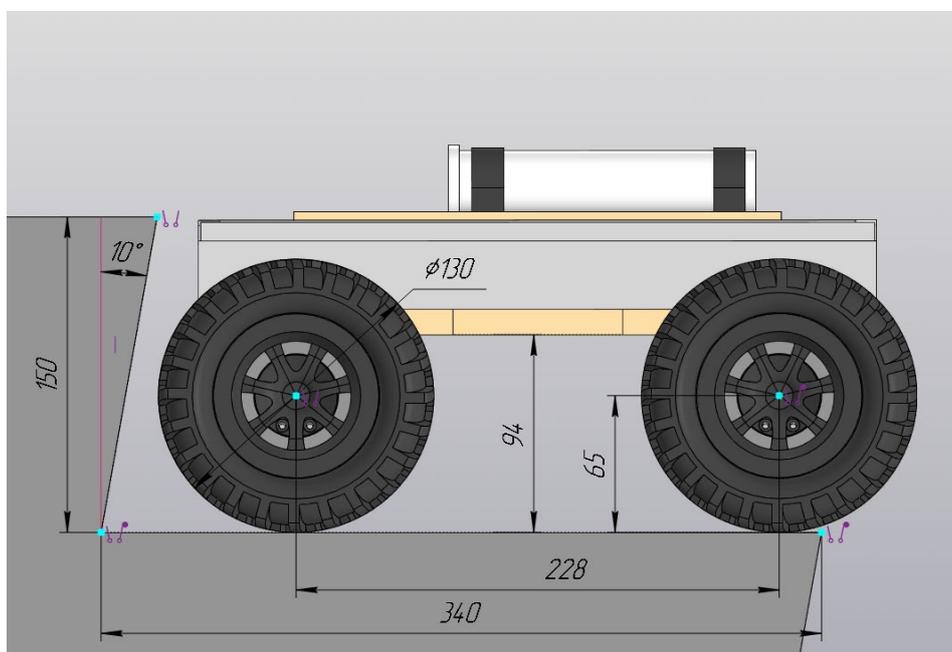


Рисунок 1 – параметры колёсной тележки робота

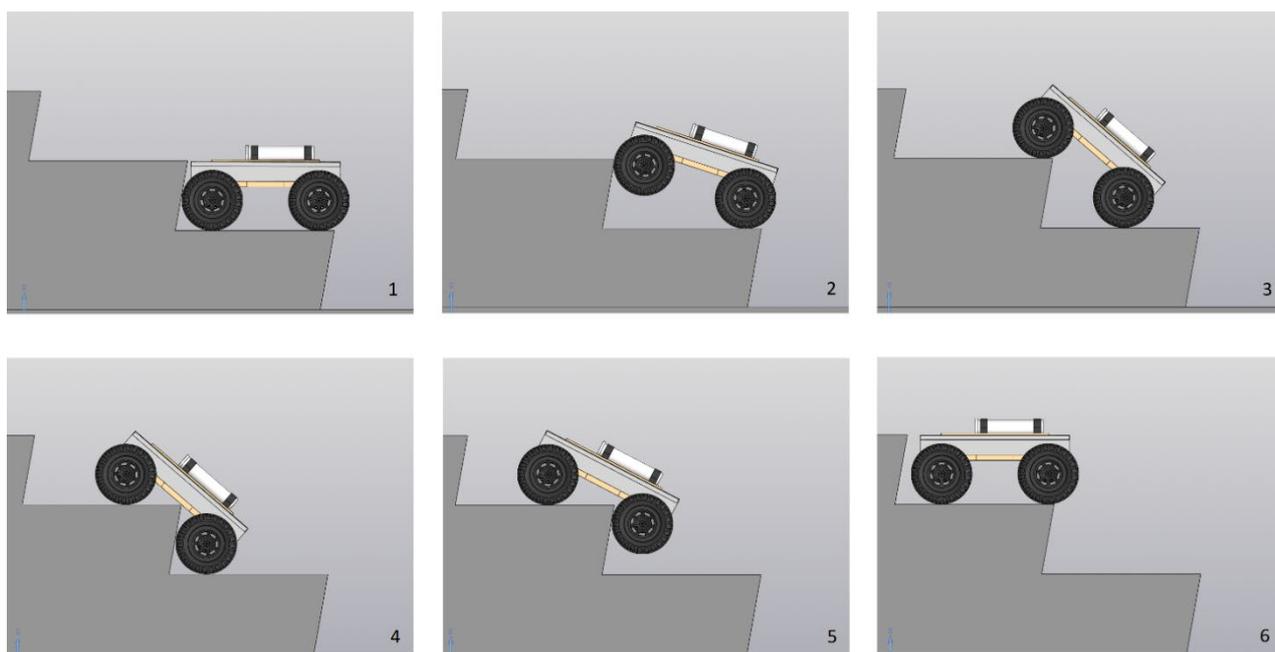


Рисунок 2 – Процесс подъёма робота по лестнице

В приложении А приведена 3D-модель колёсной тележки робота в различных проекциях.

Манипулятор

Чтобы робот мог не только ездить, но ещё и выполнять полезные действия, необходима рука-манипулятор. Для начала я выделил возможные сферы применения манипулятора: захват объектов с последующей транспортировкой, перекрытие вентилей, разбор завалов и др. А также сформулировал основные требования к манипулятору: простота и надёжность конструкции, функциональность, точность манипуляций, возможность сложить руку «на спину».

На рисунке 3 приведены фотографии колёсной тележки с манипулятором в разных проекциях.

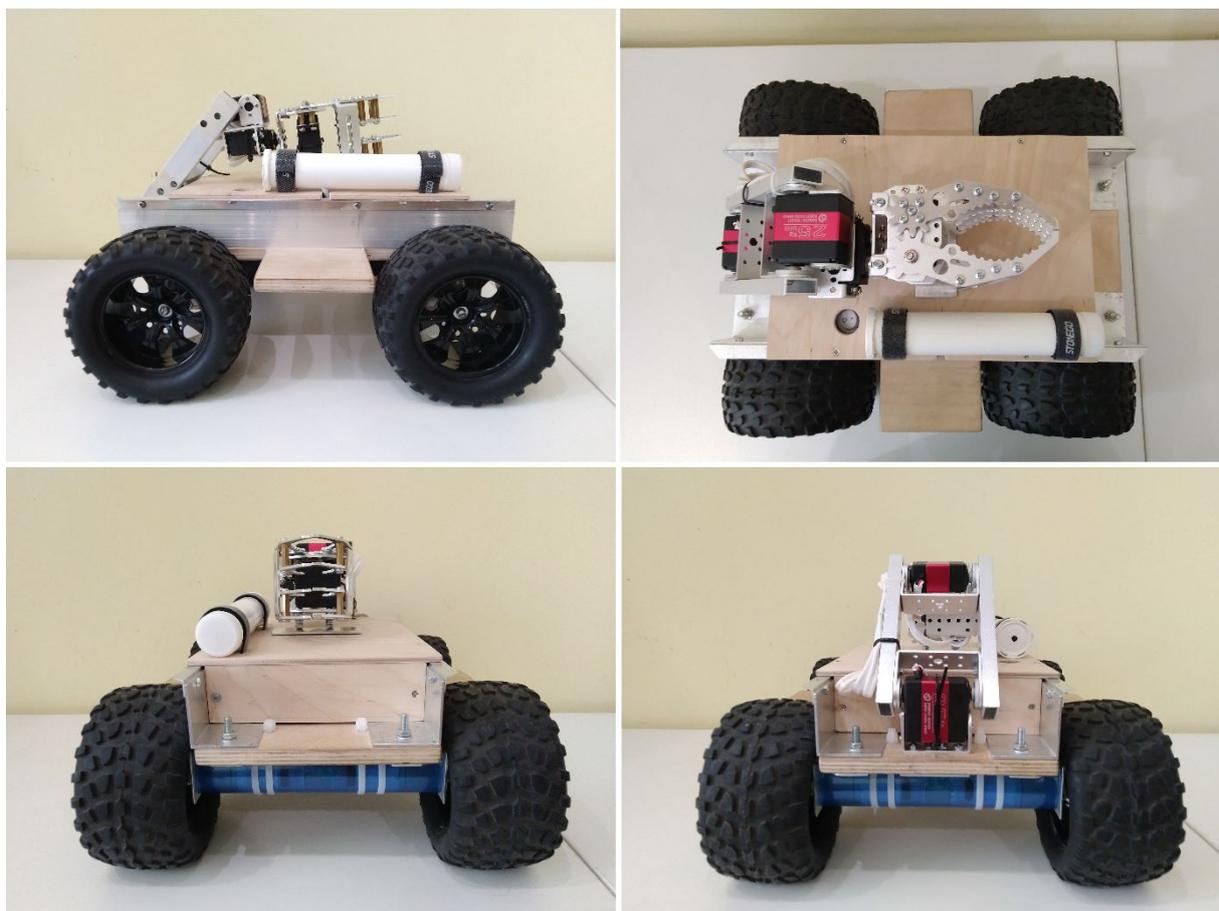


Рисунок 3 – Колесная тележка с манипулятором

Я решил сделать манипулятор с четырьмя степенями свободы: плечо, локтевой сустав, механизм поворота клешни и сам захват. Далее я рассчитал крутящие моменты сервоприводов. Манипулятор робота должен уметь в вытянутом положении поднимать и удерживать 1 кг нагрузки. Оптимальная длина руки для данных габаритов робота была выбрана 25 см. Из этого следует, что сервопривод на плечевом суставе должен иметь крутящий момент не менее 25 кг·см. Я выбрал RDS3225. На локтевом суставе установлен сервопривод, аналогичный сервоприводу на плече. На механизме поворота клешни и самом захвате рычаги имеют меньшую длину, поэтому на этих суставах установлены менее мощные сервоприводы MG995 с крутящим моментом 15 кг·см.

Дополнительные откидные колёса

В ходе испытаний выяснилось, что для повышения проходимости робота можно установить дополнительные откидные колеса (на хвосте), благодаря которым робот сможет преодолевать высокие препятствия. Также появится возможность опустить хвост для увеличения площади соприкосновения с поверхностью и более равномерного распределения давления.

Чтобы облегчить конструкцию, колёса не имеют привода и закреплены на жёсткой раме, которую приводит в движение сервопривод RDS5160 с крутящим моментом 60 кг·см.

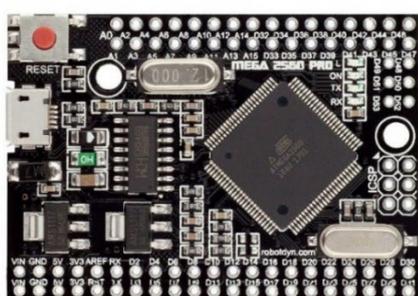


Рисунок 4 – Робот после установки дополнительных колёс

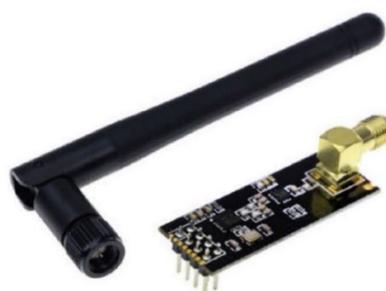
Электронные компоненты

Для обработки сигналов с пульта необходим микроконтроллер (МК) или плата управления. Я решил использовать МК ATmega2560 (Arduino Mega 2560 Pro) [2][3]: он прост в использовании, имеет большие возможности, а также достаточно много пинов (вход/выход) по сравнению с другими МК линейки Arduino.

Для связи между роботом и пультом будет использоваться совместимый с Arduino модуль NRF24L01+PA+LNA [4], работающий на частоте 2.4 ГГц, с заявленной дальностью действия до 1 км на открытом пространстве.



(а)



(б)

а – Arduino Mega 2560 Pro, б – NRF24L01+PA+LNA

Рисунок 5 – Микроконтроллер и радиомодуль

Моторы ходовой части робота рассчитаны на напряжение 12 V, поэтому был выбран аккумулятор со схемой 3S2P ёмкостью 6000 мАч. Он был самостоятельно собран из 6 Li-ion аккумуляторов 18650. Логическая часть робота и все сервоприводы питаются от 5 V, поэтому

на работе установлено две понижающих преобразователя напряжения: Mini360 (на питание логической части) и XL4015E1 (на сервоприводы).

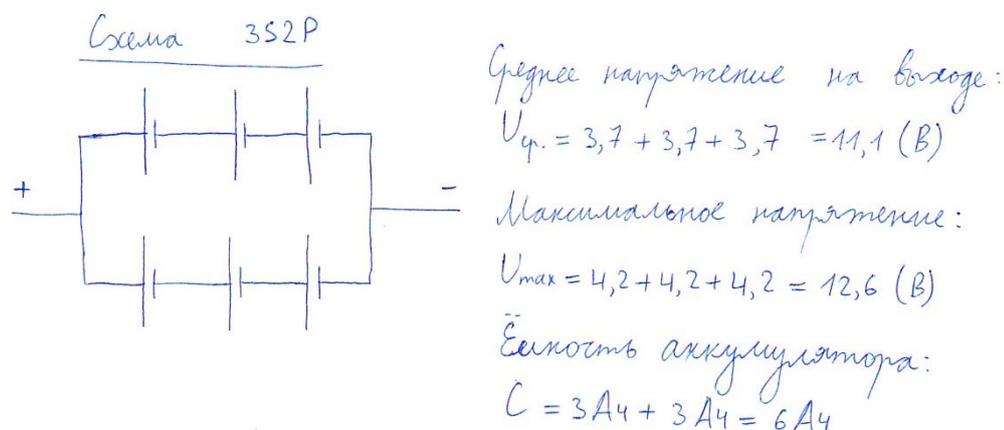


Рисунок 6 – Расчёт параметров аккумулятора

Электроника робота распаяна на макетной плате. Для удобства разводки питания имеются шины 12 V, 5 V, а также шина I2C для общения между МК и датчиками.

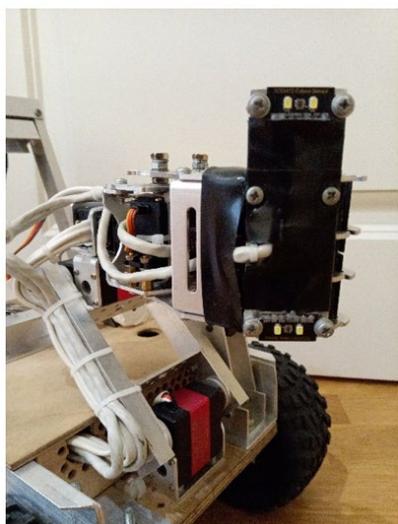
Датчики и автономные программы

Для повышения удобства управления роботом и возможности автономного выполнения действий было решено установить на робота гироскопический датчик, дальномер, а также два датчика цвета.

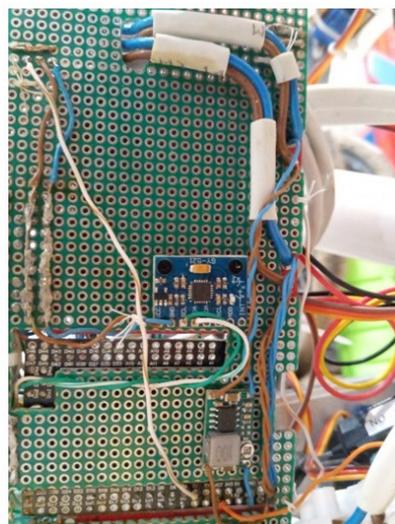
В качестве гироскопического датчика я выбрал гироскоп-акселерометр MPU6050 [7]. Он установлен в точке геометрического центра робота. С помощью гироскопического датчика можно отслеживать положение робота в пространстве. Это позволяет удерживать заданный курс, а также отслеживать изменение уклона местности.

В качестве дальномера выбрал лазерный дальномер VL53L0X [6]. Я закрепил дальномер на клешне манипулятора. Он позволяет роботу захватывать предметы в автоматическом режиме.

Также на роботе установлено два датчика цвета TCS3472 [5]. Датчики позволяют роботу отслеживать контрастную линию и автономно двигаться по ней. Это может быть полезно при езде в помещениях с разметкой на полу в условиях плохой связи. Датчики цвета закреплены на клешне робота.



(а)



(б)

а – пластина с датчиками цвета, б – гироскопический датчик

Рисунок 7 – Датчики цвета и гироскоп



Рисунок 8 – Лазерный дальномер

Программа для робота

Для написания программ робота и пульта использовалась среда разработки Arduino IDE. Передача управляющих команд и телеметрии осуществляется путём обмена сообщениями. Эти сообщения представляют из себя пакеты данных размером 32 байта. Робот принимает пакеты, затем обрабатывает данные и подаёт сигналы на моторы, сервоприводы и другие модули. Данные телеметрии отправляются с робота на пульт.

Ссылка на GitHub с программами для робота, пульта и приёма видеоизображения с камер <https://github.com/VladimirMukhin21/Totoshka>.

Видеосвязь с роботом

Для управления роботом вне зоны видимости необходима система установленных на роботе видеокамер, а также устройства для передачи и приёма видеосигнала. Для этой цели я

выбрал технологию FPV (First person view), работающую на частоте 5.8 ГГц. Эта технология передачи сигнала бюджетна и удобна тем, что при передаче видекартинки по аналоговому сигналу нет торможения и зависания изображения, что очень важно при управлении роботом в экстремальных ситуациях.

На роботе установлены две камеры: основная обзорная на специальной раме и вспомогательная спереди. Камеры приводятся в движение при помощи сервоприводов SG90, так что при необходимости их можно повернуть, изменив тем самым угол обзора.

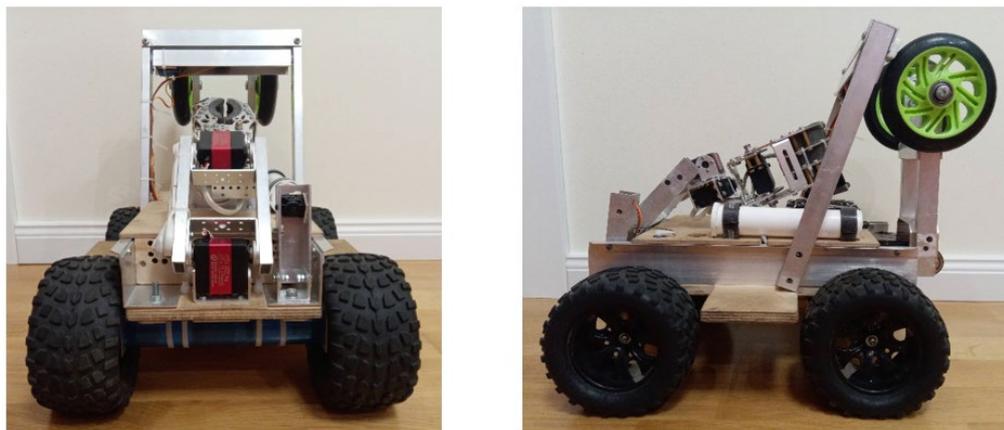


Рисунок 9 – Робот с видеосистемой

После анализа различных видеопередатчиков и приёмников были выбраны передатчик TS832 и приёмник видеосигнала Skydroid, который подключается к ноутбуку через USB-кабель. Камеры должны иметь небольшие размеры, чтобы их можно было установить в алюминиевый профиль п-образного сечения 20x25 мм. Я выбрал камеры Foxeer Razer Micro: они имеют небольшие размеры и дают при этом достаточно качественное видеоизображение. Так как видеопередатчик может передавать сигнал только с одной камеры, то для переключения между двумя камерами робота используется специальный трёхканальный видеокоммутатор. Изображения используемых элементов видеосистемы приведены в Приложении Б.

Обработка видеоизображения

Для вывода видеоизображения с камер робота на рабочем месте оператора используется программа на языке Python и библиотека OpenCV [8]. Программа позволяет отображать данные телеметрии, вспомогательные линии на кадре, производить запись в файл и имеет другие полезные функции.

Питание и взаимодействие электронных компонентов

Приведу схему электрического питания и взаимодействия электронных компонентов.

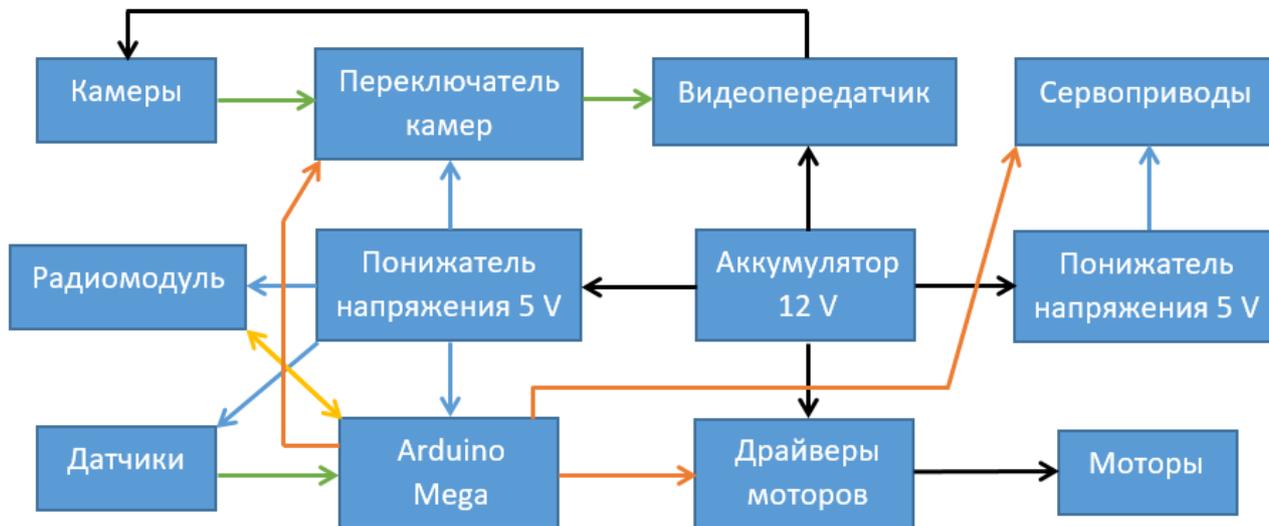


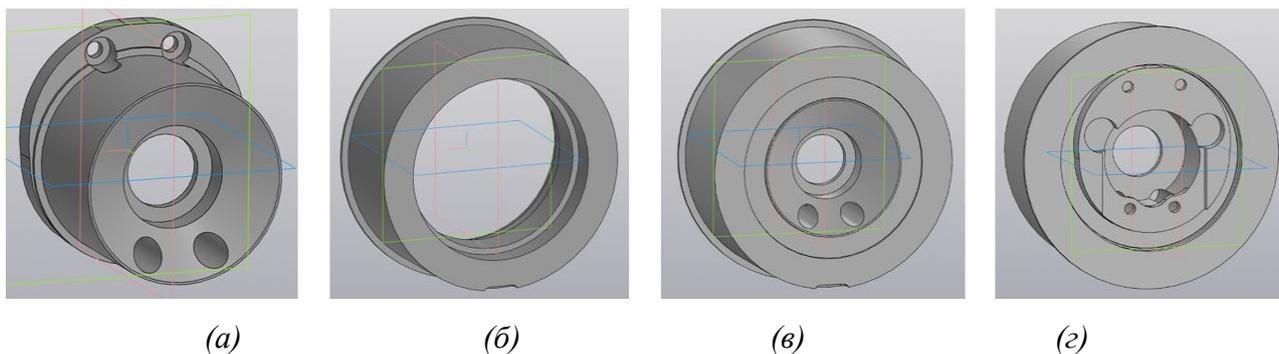
Рисунок 10 – Схема электрического питания и взаимодействия электроники робота

Условные обозначения:

- напряжение 12 V;
- напряжение 5 V;
- управляющие сигналы;
- передача показаний датчиков и сигнала с камер;
- ↔ двусторонняя связь между микроконтроллером и модулем радиосвязи.

Доработки робота

Изначально весь вес робота приходился на валы редукторов моторов. Для снятия нагрузки с валов и продления срока службы редукторов я решил добавить подшипники на колёса. Подшипники выбрал 6808. 3D-модель для их крепления сделал в САПР Компас-3D.



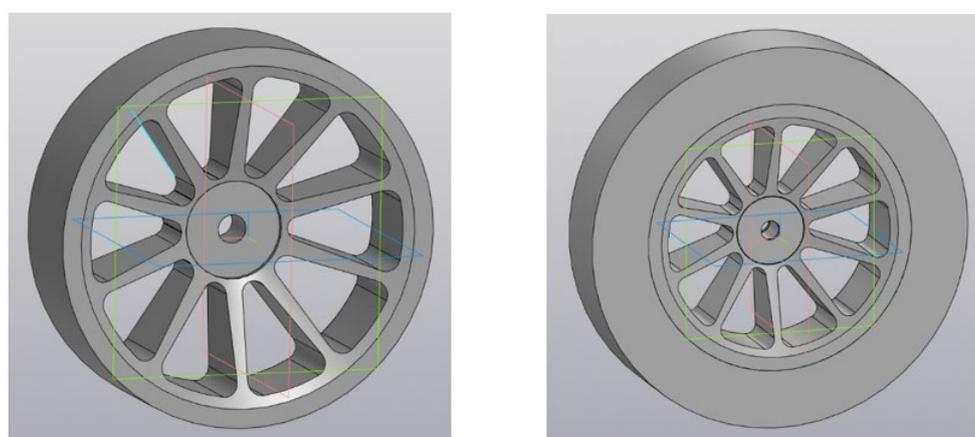
а – втулка, б – диск-вкладыш в колесо, в и г – сборки узла (с разных сторон)

Рисунок 11 – 3D-модель ступицы колеса с посадочным местом под подшипники

После этой доработки вес робота стал приходиться на подшипники, а валы стали выполнять только функцию передачи крутящего момента.

Установленные изначально дополнительные откидные колёса оказались слишком тяжёлыми, из-за чего центр массы робота был смещён к задней части. Установка колёс меньшего диаметра привела к снижению проходимости. Чтобы решить эту проблему, я решил напечатать на 3D-принтере облегчённые колёса большого диаметра.

Печать облегчённых колёс позволила сместить центр тяжести робота ближе к передней части. Важно, что после этой доработки робот не потерял своей проходимости.



(а)

(б)

а – 3D-модель диска колеса, б – сборка колеса

Рисунок 12 – 3D-модель облегчённого колеса

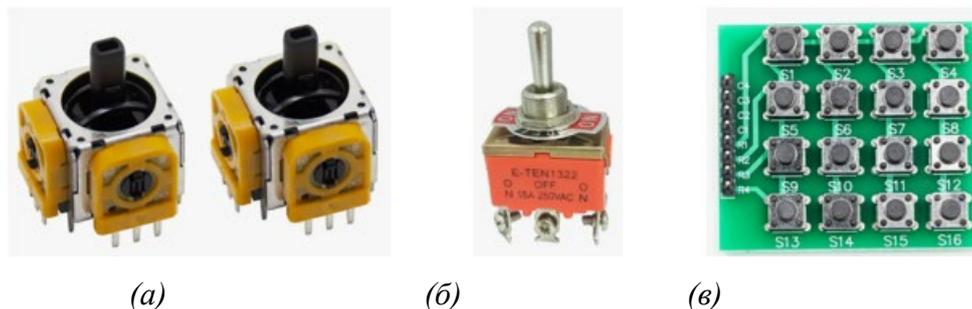
Разработка пульта управления

Определение принципа управления и необходимых электронных компонентов

Стандартные пульты для радиоделей имеют ограниченное количество каналов связи, которых попросту не хватит для управления моим роботом. Поэтому я решил разработать собственную аппаратуру на базе микроконтроллера ATmega2560 (Arduino Mega 2560 Pro), аналогичном установленному на работе, и радиомодуля NRF24L01+PA+LNA, имеющего 128 каналов.

Сначала я определил принцип управления роботом и основные электронные компоненты пульта. Для управления ходовой частью, манипулятором и хвостом нужно 2 джойстика (1 на колёсную тележку, 1 на манипулятор и хвост). Так как для управления манипулятором и хвостом будет использоваться общий джойстик, то для переключения

режима управления необходим дополнительный тумблер. Также необходимы кнопки для запуска автономных программ и действий.



(а) (б) (в)
а – джойстики, б – тумблер, в – кнопочная клавиатура

Рисунок 13 – Джойстики, тумблер и кнопочная клавиатура

Схема электрического питания и взаимодействия электроники

Электронные компоненты пульта взаимодействуют по несложной схеме (приведена на рисунке 14): информация о состоянии джойстиков, кнопок и тумблера передается на микроконтроллер, МК формирует пакеты данных и отправляет их по радиосвязи на робота. В обратных пакетах данных с робота он принимает показания телеметрии и передает их по USB на компьютер.

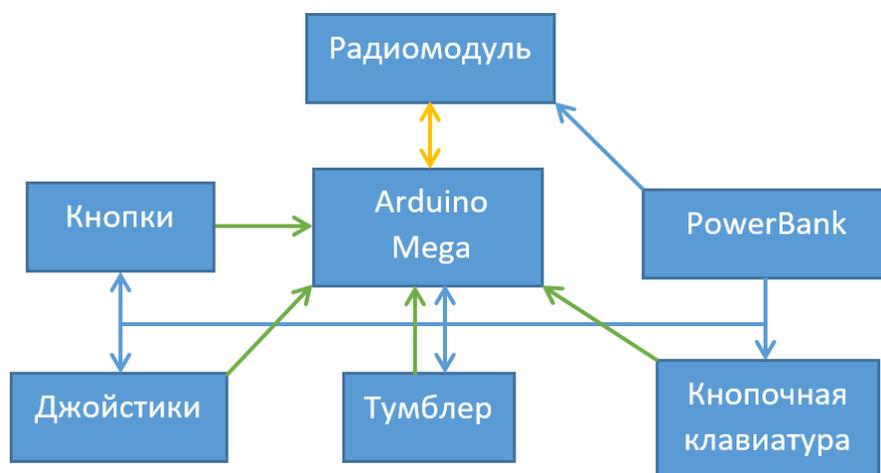


Рисунок 14 – Схема электрического питания и взаимодействия электроники пульта

Условные обозначения:

- напряжение 5 V;
- передача сигналов с элементов управления;
- ↔ двусторонняя связь между микроконтроллером и модулем радиосвязи.

Сборка пульта управления

Корпус пульта я изготовлен из фанеры толщиной 4 мм. Вся электроника распаяна на макетной плате, которая расположена внутри корпуса. Питается пульт от PowerBank. Также есть возможность подключения пульта к ноутбуку по USB-кабелю, при этом через USB идёт как подача питания, так и передача информации. Внешний вид пульта управления приведён в Приложении В.

Написание программы для пульта

Программа пульта выполняет опрос джойстиков, кнопок и переключателя, упаковывает их показания в пакеты данных, которые отсылаются на робота.

Краткая инструкция по управлению роботом

Ходовая часть робота управляется при помощи правого джойстика. За движение манипулятором и хвостом отвечает левый джойстик. Есть 3 режима левого джойстика, для переключения которых используется 3-х позиционный тумблер. В положении тумблера от себя идёт управление локтем манипулятора и вращением клешни. Среднее положение тумблера отвечает за движение руки с сохранением параллельности, а также за разжатие/сжатие захвата. Поднимание и опускание хвоста с дополнительными колёсами осуществляется в положении тумблера на себя. Также на пульте есть 6 кнопок разных цветов, имеющих следующие функции:

- синяя – кнопка-предохранитель от случайного запуска автономных программ (автономные программы запускаются только в сочетании с нажатой синей кнопкой), нажатая синяя кнопка в сочетании с левым джойстиком отвечает за поворот камеры;
- зелёная – подъём манипулятора на спину/хвоста (в зависимости от положения тумблера);
- красная – стоп всех узлов робота и остановка автономных программ;
- жёлтая – запуск программы для автономного захвата маяка;
- белая – ещё одна кнопка-предохранитель (аналогично синей кнопке), используется для увеличения количества автономных программ;
- чёрная – переключение камеры.

Для запуска остальных автономных программ есть кнопочная клавиатура 4x4 кнопки.

Испытания, сравнение, планы на будущее

Сравнение разработки с уже существующими роботами

Чтобы проверить работоспособность системы, я принимал участие в соревнованиях по экстремальной робототехнике «Кубок РТК» и неоднократно занимал призовые места. Испытания подтвердили удачность принятых технических решений и надёжность конструкции.

Дополнительно к испытаниям на специальном полигоне я провёл тесты устройства по нескольким параметрам и составил таблицу 2 для сравнения с другими роботами [9][10].

Таблица 2 – сравнительная таблица

Название робота	Тотошка	РТК Teodor	РТК Telemax	МРК-35	ТРОПА-ЗРОП
Тип шасси	4-х колесная база + 2 доп. колеса	гусеничн.	гусеничн.	гусеничн.	6-ти колесная база
Мин. габариты (ДхШхВ), см	39х30х34	-	80х40х75	80х65х90	182х128х56
Масса, кг	5,4	375,0	80,0	320,0	350,0
Грузоподъемн. робота, кг	5	100	20	-	200
Грузоподъемн. манипулятора, кг	1	30	5	40	-
Макс. скорость, км/ч	2,7	3,0	4,0	2,3	20,0
Макс. угол подъема, град.	40	45	45	25	-
Преодолеваемое препятствие, см	22	-	50	50	-
Дальность связи, м	80-90 в городской застройке	-	-	200 по кабелю, 500 по радиосвязи	-
Время работы без подзарядки, ч	2-4	-	2-4	4	10

По данным таблицы несложно заметить, что разработанный робот сильно отличается от остальных своей небольшой массой и габаритами. При этом по проходимости, скорости и углу подъёма по наклонной робот не сильно уступает остальным. Манипулятор робота достаточно функционален для данных габаритов робота, но по грузоподъёмности и количеству подвижных суставов не может сравниться с манипуляторами РТК Teodor и МРК-35.

Планы на будущее

В дальнейшем я планирую улучшить подсистему технического зрения робота для возможности автономной работы робота без участия оператора.

ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта был разработан и создан прототип робота для помощи спасателям при устранении последствий ЧС. Робот соответствует изначально запланированным сферам применения, таким как разведка местности, доставка полезной нагрузки в зону ЧС, поиск людей и животных и другим. Проведенные испытания подтвердили удачность принятых технических решений и надежность конструкции. Робот может использоваться в качестве помощника спасателям МЧС, а также в военной сфере.

Поставленные в начале работы задачи можно считать выполненными, а цель – достигнутой.

СПИСОК ЛИТЕРАТУРЫ

Драйвер коллекторного двигателя VNH2SP30. - URL: <https://arduinolab.pw/index.php/2017/06/20/drajver-kollektornogo-dvigatelya-vnh2sp30/> (дата обращения 10.02.25)

Arduino Documentation. - URL: <https://docs.arduino.cc/> (дата обращения 10.02.25)

ATMEGA2560 Datasheet. - URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/107092/ATMEL/ATMEGA2560.html> (дата обращения 10.02.25)

Урок 26.5 Соединяем несколько arduino по радиоканалу через nRF24L01+. - URL: <https://lesson.iarduino.ru/page/urok-26-5-soedinyaem-neskolko-arduino-po-radiokanalu-cherez-nrf24l01/> (дата обращения 10.02.25)

Обзор датчик цвета на TCS3472. - URL: <https://robotchip.ru/obzor-datchik-tsveta-na-tcs3472/> (дата обращения 10.02.25)

Arduino VL53L0X: лазерный дальномер. - URL: <https://arduinoplus.ru/arduino-vl53lox/> (дата обращения 10.02.25)

Гайвер, А. Работа с Arduino и MPU6050. - URL: <https://alexgyver.ru/arduino-mpu6050/> (дата обращения 10.02.25)

OpenCV Tutorials. - URL: https://docs.opencv.org/4.x/d9/df8/tutorial_root.html (дата обращения 10.02.25)

Роботы МЧС. - URL: https://asu100.ru/catalog/CO/M4C/index_m4c.html (дата обращения 10.02.25)

Робототехнические комплексы (РТК): основные модели, описание и ТТХ. - URL: <https://fireman.club/statyi-polzovateley/robototekhnicheskie-kompleksyi-mchs-osnovnyie-modeli-opisanie-i-tth/?ysclid=m53zhgnxic465064398> (дата обращения 10.02.25)

Приложение А

3D-модели колёсной тележки робота

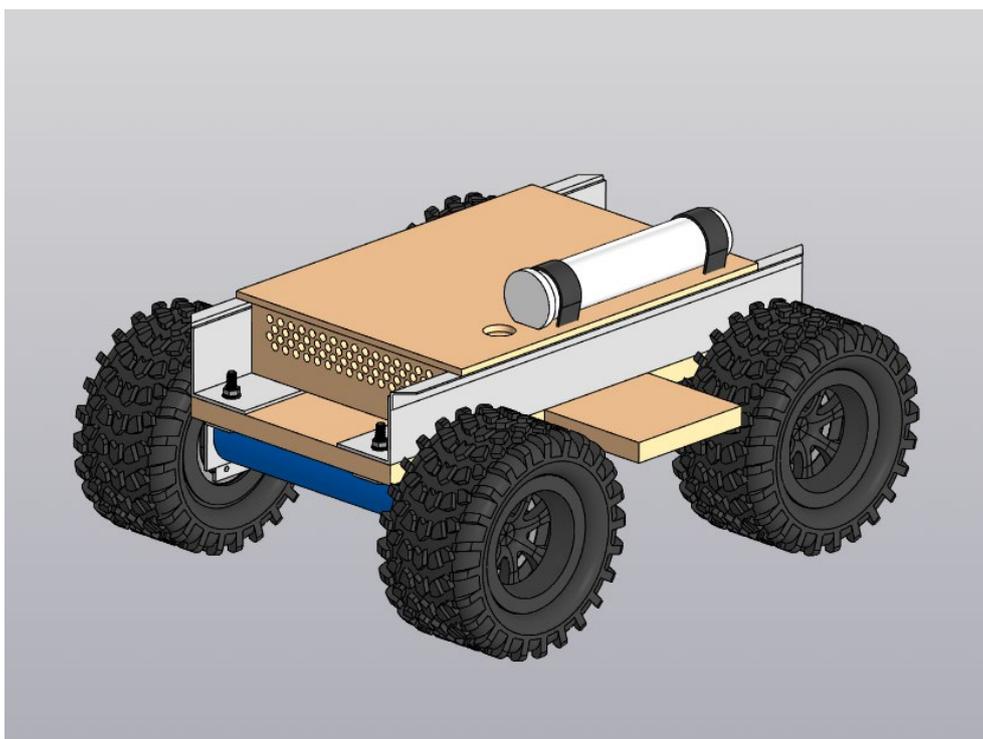


Рисунок А.1 – Колёсная тележка, общий вид

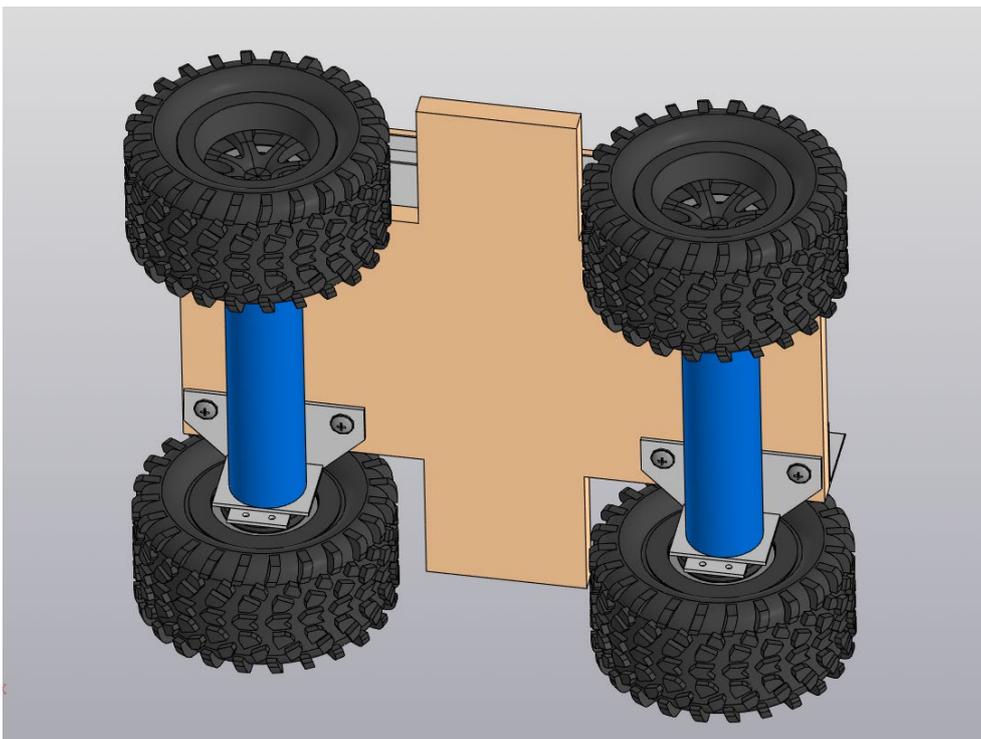


Рисунок А.2 – Колёсная база, вид снизу

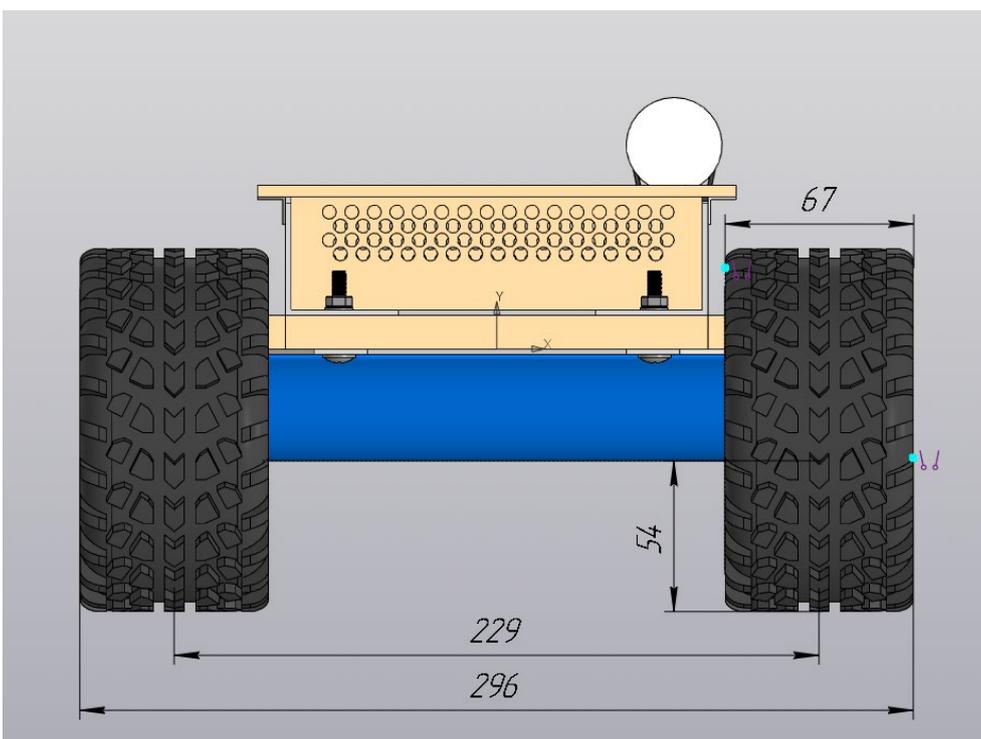


Рисунок А.3 – Колёсная тележка, фронтальная проекция

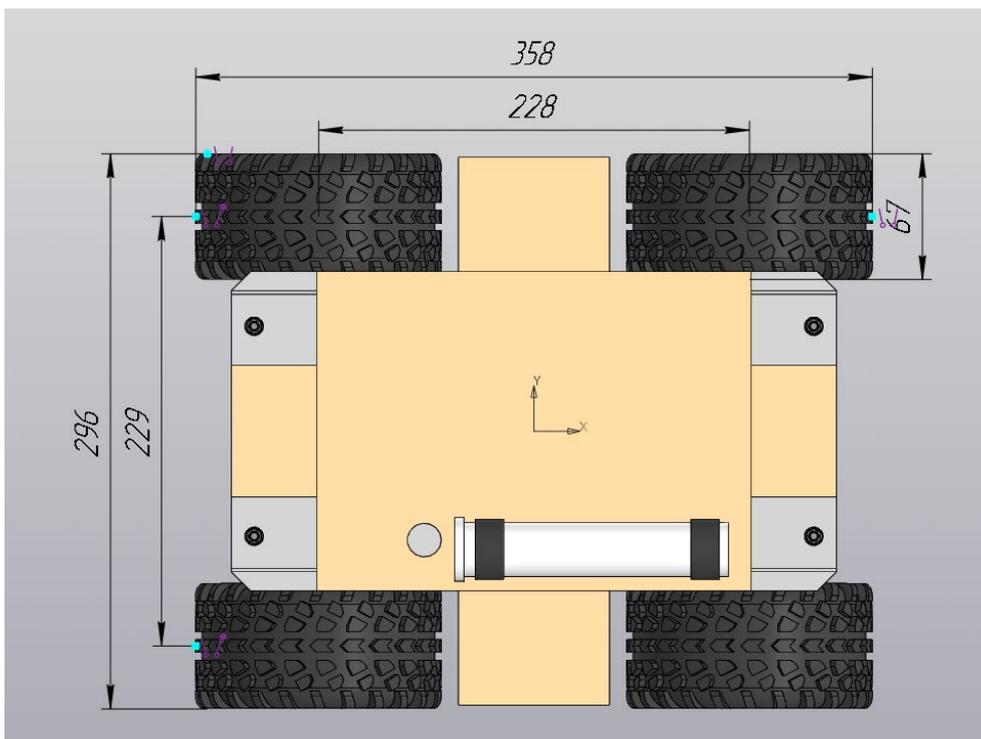


Рисунок А.4 – Колёсная тележка, горизонтальная проекция

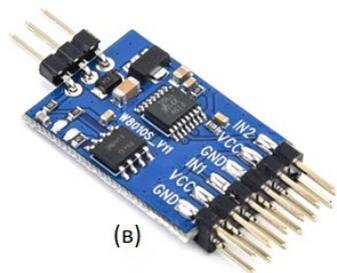
Приложение Б Элементы видеосистемы



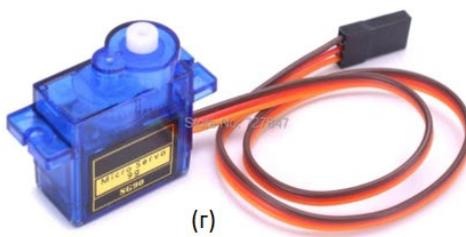
(а)



(б)



(в)



(г)



(д)

*а – видеопередатчик TS832, б – приёмник видеосигнала Skydroid, в – видеокоммутатор,
г – сервопривод MG90, д – камера Foxeer Razer Micro*

Рисунок Б.1 – Элементы видеосистемы

Приложение В
Пульт управления



Рисунок В.1 – Внешний вид пульта управления

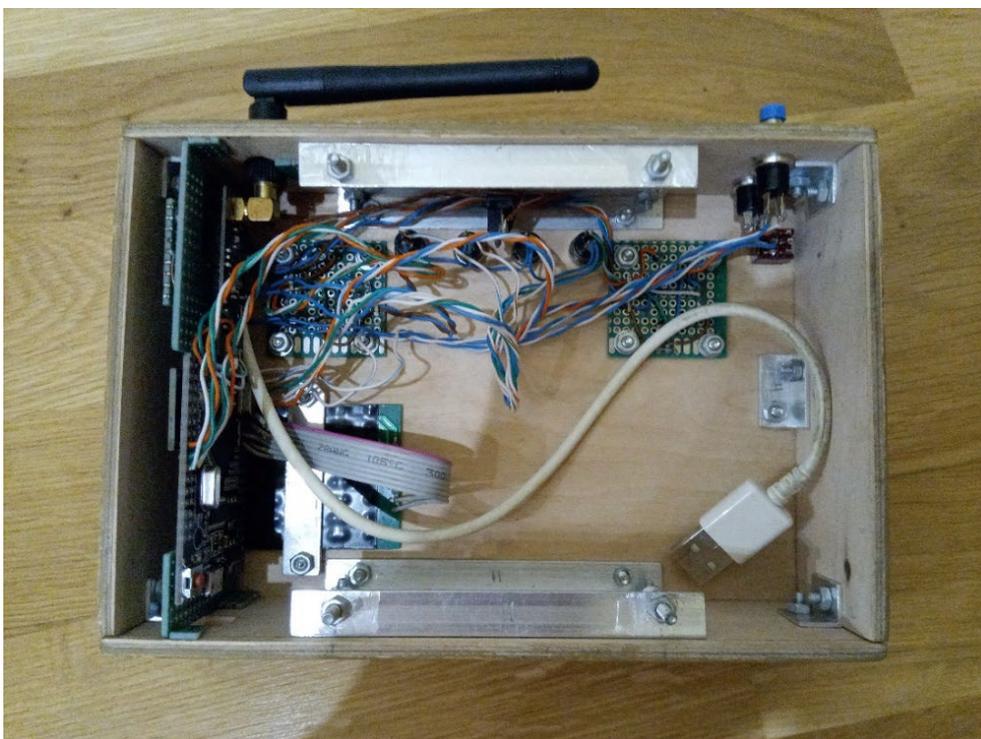


Рисунок В.2 – Пульт «изнутри»

СРЕДСТВА КИНЕМАТИЧЕСКОЙ ОБРАТНОЙ СВЯЗИ В ИМИТАТОРАХ ПЕРЕГРУЗОК ДЛЯ ТРЕНАЖЁРОВ ПИЛОТИРУЕМЫХ КОСМИЧЕСКИХ КОРАБЛЕЙ

Пономарев Никита Константинович

Балашиха г, МБОУ "Школа №30", 11 класс

Секция: Навигация и системы управления летательными аппаратами

*Научный руководитель: Пономарев Константин Владимирович, заместитель начальника
службы информационных технологий ФГБУ «НИИ ЦПК имени Ю.А.Гагарина»*

Аннотация

Предметом исследования данной работы является средство кинематической обратной связи.

Цель: выяснение возможности и необходимости модернизации тренажёров пилотируемых космических кораблей.

Методы: анализ литературы, поиск информации в Интернете, беседа, макетирование/прототипирование, эксперимент.

Результаты: выполнен поиск и анализ информации, собран экспериментальный стенд, состоящий из платформы с закреплёнными на ней аудиопреобразователем с сердечником (вибрационный компонент), источником аудиосигнала и его усилителем (акустический компонент), источником видеосигнала и телевизором (визуальный компонент). Выполнена презентация экспериментального стенда на Международной научно-практической конференции и его апробация с участием действующих космонавтов и специалистов Центра подготовки космонавтов.

Выводы: несмотря на простоту стенда, обратная кинематическая связь получилась достаточно ощутимой. Средство кинематической обратной связи получило положительный отклик среди космонавтов и специалистов, что делает его применимым для модернизации тренажёров пилотируемых космических кораблей после выполнения обследования на предмет безопасности использования.

Введение

В повседневной жизни каждый человек испытывает перегрузки, которые появляются при воздействии внешних сил. Мы регулярно сталкиваемся с перегрузками малых значений, например, в момент начала движения лифта (до 1,2 G), ускоряясь в автомобиле (от 0,28 до 1,65 G), взлетая на самолёте (до 1,5 G), приземляясь на парашюте (до 1,8 G) и другие.

Многие космонавты в своих интервью рассказывают о своих ощущениях от перегрузок в ходе старта и посадки.

Из воспоминаний космонавта О.В. Котова: «Ощущения во время выведения на орбиту смазываются очень напряжённой работой: всё время занят контролем приборов, связью с Землей, просмотром бортовой документации. Единственное, что замечаешь, — отделение ступеней. Первые две отделяются сравнительно мягко, масса оставшейся ракеты ещё велика. А вот отделение третьей пропустить трудно — сравнимо с хорошим пинком под зад. Срабатывают пироболты, отбрасывают назад остатки ракеты, и начинается состояние невесомости». [4]

Из воспоминаний космонавта А.М. Самокутяева: «Кроме перегрузки вход в атмосферу сопровождался сильной вибрацией. Представьте, что Ваша машина едет по хорошим кочкам на одних дисках, без шин. Вот так примерно трясло спускаемый аппарат. И сильнейший гул, больше, чем в метро. Затем характерный звук и удар – это вышли стропы и открылся парашют, сначала вытяжной, затем основной. Снова перегрузка, сильнейшая болтанка и раскачка. Затем перецепка и закрутка – мы все это чувствуем физически и понимаем, что происходит». [6]

Данные обстоятельства натолкнули меня на мысль о том, как можно помочь космонавту подготовиться к данным воздействиям и сделать тренировки ещё чуть более похожими на реальный полёт.

1. Средства подготовки к космическим полётам

1.1 Действующие технические средства подготовки космонавтов

Подготовка космонавтов – это длительный и трудоёмкий процесс, в котором задействовано большое количество специалистов космической отрасли. Сегодня, как и 65 лет назад, космонавт — это штучная профессия, поэтому отдельного вуза для подготовки будущих покорителей звёзд в России не существует. [5]

На протяжении всей своей профессиональной карьеры космонавту приходится непрерывно получать новые знания из разных областей науки и техники. Космонавты шуточно говорят о себе, что они «вечные студенты», но это и на самом деле так.

Полноценной подготовкой космонавтов в мире занимается всего несколько организаций, в России этим занимается ФГБУ «НИИ ЦПК имени Ю.А.Гагарина», известный как Центр подготовки космонавтов. Кроме теоретической подготовки, значительная часть времени отводится на практические занятия и тренировки на тренажёрах.

Тренажёры космические – специальные установки, имитирующие условия космического пространства и космического полёта. Используются для подготовки космонавтов, которые будут участвовать в космических полётах.

Специалисты по подготовке космонавтов стремятся к тому, чтобы тренажёры давали космонавтам максимально полную картину происходящего в реальном полёте, и для этого тренажёры постоянно модернизируются и дорабатываются.

В настоящее время в Центре подготовки космонавтов используются следующие технические средства подготовки космонавтов: [8]

- комплекс тренажёров российского сегмента Международной космической станции;
- комплекс тренажёров транспортного пилотируемого корабля «Союз»;
- различные функционально-моделирующие стенды и учебно-тренировочные натурные макеты;
- средства медицинской подготовки космонавтов;
- центрифуги для воспроизведения перегрузок;
- тренажёры и макеты для выживания в различных климато-географических зонах;
- гидролаборатория для подготовки к внекорабельной деятельности в открытом космосе;
- специальная авиационная техника для тренировок в условиях пониженной гравитации.

Воздействие на космонавтов перегрузок, возникающих во время полёта, исследуют в центрифугах. Кабина центрифуги имеет три степени свободы и обеспечивает создание перегрузок, действующих на космонавтов в любом направлении. Изменяя частоту вращения, можно моделировать условия старта, отделения ступеней ракеты-носителя и многие другие ситуации. В имитаторах линейных ускорений моделируют перегрузки торможения, которые могут возникать при входе в плотные слои атмосферы, при возврате на Землю, перегрузки при очень высоких скоростях и их быстром нарастании. [7]

Космический полёт является сложным испытанием для человеческого организма, и первое, с чем сталкивается космонавт при выполнении своего первого космического полёта – это совершенно новые ощущения собственного организма. При старте ракеты с космическим кораблём – это перегрузки около 3 G, в ходе космического полёта – ощущение невесомости, а при возвращении на Землю – перегрузки от 5 до 8 G.

Для того чтобы познакомить космонавтов с тем, с чем им предстоит столкнуться, а заодно проверить и натренировать вестибулярный аппарат, используют центробежные машины – центрифуги. Они могут создавать перегрузки от 1 до 30 G с ориентацией вектора перегрузки в любом заданном направлении. Кроме этого, в кабине создаются те условия, которые будут в условиях настоящего полёта – это температура, давление, влажность и газовый состав. [3]



Рисунок 1.1 Центрифуга с плечом 18 м (ЦФ-18) в Центре подготовки космонавтов.

1.2 Концепция модернизации центрифуг с применением вибрационной отдачи

С момента создания в тренажёры на базе центрифуг вносились доработки. Эти изменения вносились в соответствии с решаемыми задачами и изменяющимися требованиями к подготовке космонавтов, а также в связи с развитием технологий.

Одним из направлений доработок тренажёров является введение управляемых колебаний рабочего места оператора по трём осям, так называемая кинематическая обратная связь в виде вибрационной отдачи.

Однако кинематическая обратная связь, присутствующая в виде вибраций при старте ракеты-носителя и при посадке спускаемого аппарата, в настоящее время не применяется. А для космонавтов, особенно для тех, кто ещё не имеет опыта космических полётов, было бы полезно понимать, какой характер вибраций является нормальным в ходе выведения и спуска. [3] При выполнении послеполётных исследований с прилетевшими космонавтами установлено, что отсутствие вибраций «размывает» ощущения реального полёта и вносит когнитивный диссонанс при выполнении послеполётных тестов. [2] Кроме того, появление кинематической обратной связи, возможно, положительно скажется на сопротивляемости перегрузкам, так как космонавт будет переключать внимание и на виброотдачу. Исследования по теме виброотдачи уже проводились. В материалах XII Международной научно-практической конференции «Пилотируемые полёты в космос» в 2017 году опубликованы тезисы с применением «актуаторов» на основе электродвигателей с эксцентриками, наподобие вибромоторов у мобильных телефонов. [3]

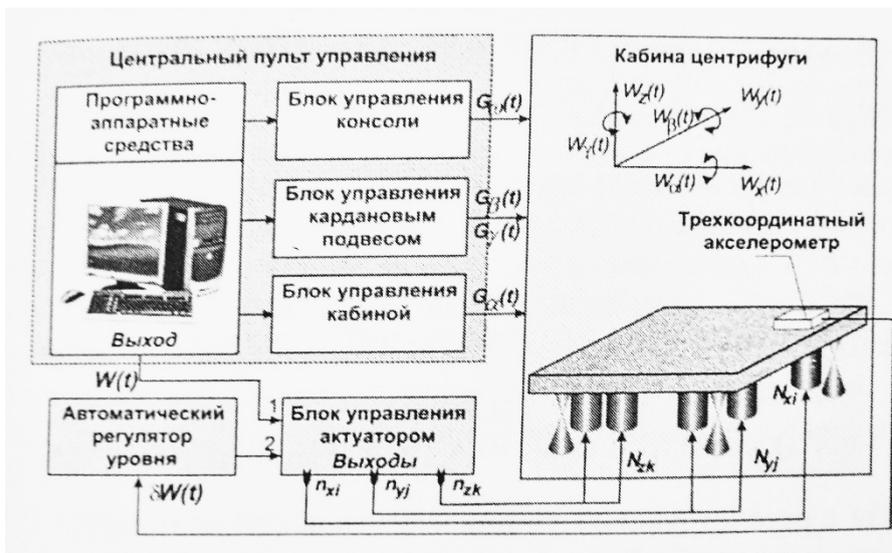


Рисунок 1.2 Структурно-функциональная блок-схема динамического испытательного стенда на основе электродвигателей с эксцентриками.

Суть данного подхода заключается в создании специализированных платформ-актуаторов, воспроизводящих пространственно-временные микродинамические воздействия (вибрацию с заданным спектром) при направленных перегрузках. Такие платформы-актуаторы должны обеспечивать генерацию и модуляцию управляемых быстроизменяющихся возвратно-поступательных и круговых орбитальных колебаний. Однако в условиях макродинамических нагрузок, вызванных вращением центрифуги, быстроизменяющиеся виброускорения претерпевают искажения, относительно ранее заданных акселерограмм, вплоть до полного их гашения. [1]

Отдельной задачей, которую нужно будет решить при использовании актуаторов в виде вибромоторов, это управление направлением вибрационных воздействий, так как конструктивно данные актуаторы блокируют вибрации только по оси вращения эксцентрика (ось Z), а по другим осям (X и Y) потребуется применять программно-механические методы управления, так как направление резонансных колебаний будет зависеть от начального положения эксцентрика при запуске двигателя.

Ещё одной особенностью воздействия вибраций на организм человека является то, что вредное действие вибрации на организм человека имеет механическую природу, по крайней мере, в диапазоне тех частот колебаний, которые возникают во время космического полёта. Космический корабль на разных этапах выведения на орбиту имеет резонансную частоту от 4,5 Гц до 9 Гц. Это очень важно, так как резонансная частота тела человека в зависимости от его положения и способа фиксации лежит в диапазоне от 3 до 12 Гц, а отдельные органы имеют более высокие собственные резонансные частоты.

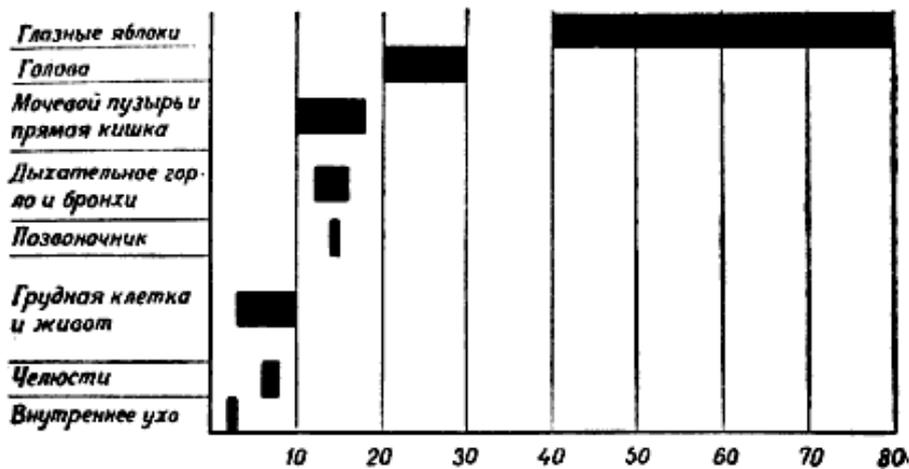


Рисунок 1.2 Диаграмма, показывающая, на каких частотах резонируют отдельные части тела человека [в Гц]

Когда космический корабль вибрирует на какой-либо из этих частот, вибрация соответствующих органов человека резонансно увеличивается, эти органы деформируются, смещаются или теряют фиксацию, то есть происходит их механическое повреждение. Однако до этого в большинстве случаев возникает ощущение дискомфорта. Отмечалось, что при частоте колебаний 50 Гц пилоты не могли считывать показания приборов, так как именно при этой частоте начинают вибрировать глазные яблоки и глаза словно застилают пеленой. [9]

Несмотря на потенциальную вредность вибраций космонавты должны понимать, с чем они столкнутся в ходе реального космического полёта, и ощутить это ещё на этапе

наземной подготовки. Долю тренировок с применением виброотдачи определяют специалисты медицинского управления Центра подготовки космонавтов.

2. Кинематическая обратная связь с использованием низкочастотных аудиопреобразователей

Предлагаемый мной подход является развитием идеи управляемой вибрационной отдачи, но на основе низкочастотного аудиопреобразователя, закреплённого к ложементу космонавта.

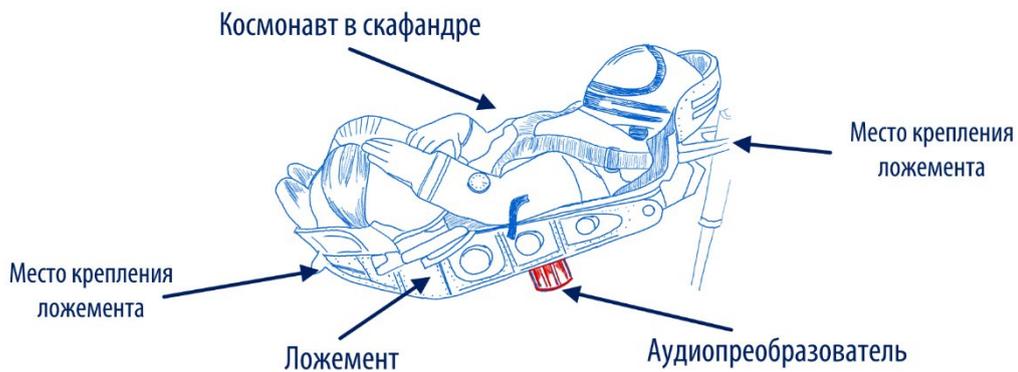


Рисунок 2.1 Предлагаемый вариант крепления аудиопреобразователя к ложементу космонавта.

Общая идея заключается в том, чтобы получить данные мгновенных ускорений от акселерометров, установленных на борту космического корабля, с привязкой к времени, затем обработать полученные данные и преобразовать их в низкочастотный аудиосигнал, который потом подавать через усилитель на аудиопреобразователь при выполнении тренировок.

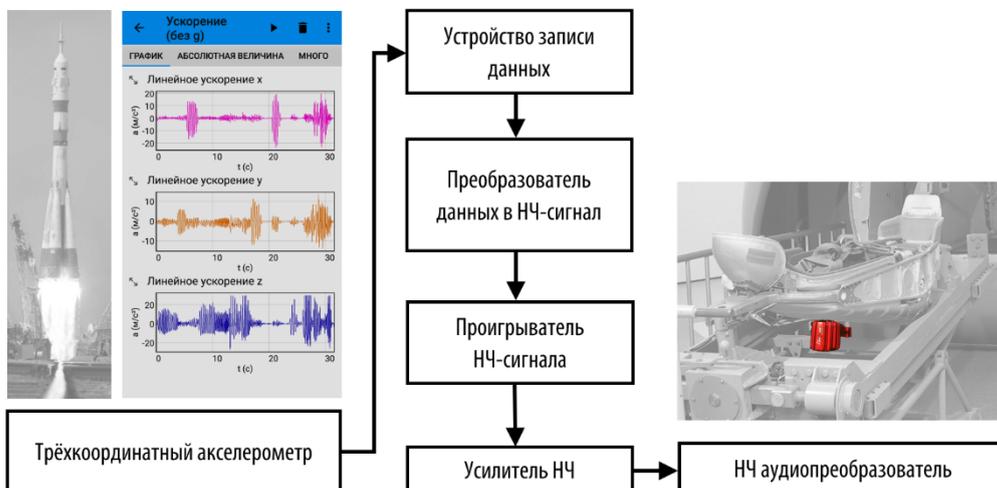


Рисунок 2.2 Предлагаемый подход по работе с данными по ускорению

Аудиопреобразователь представляет собой некоторое подобие звукового динамика, однако здесь движется не диффузор, а массивный неодимовый магнит, который и передаёт свою энергию на ту конструкцию, к которой закреплён корпус аудиопреобразователя. Чем больше масса неодимового сердечника, тем большую энергию он может передать конструкции, но и тем большее количество энергии он потребляет сам. Отличительной особенностью данного устройства является чёткая фиксация оси вибрационных воздействий, которая определяется возвратно-поступательными движениями магнитного сердечника. Подбор оптимального аудиопреобразователя и усилителя к нему является предметом отдельного исследования.

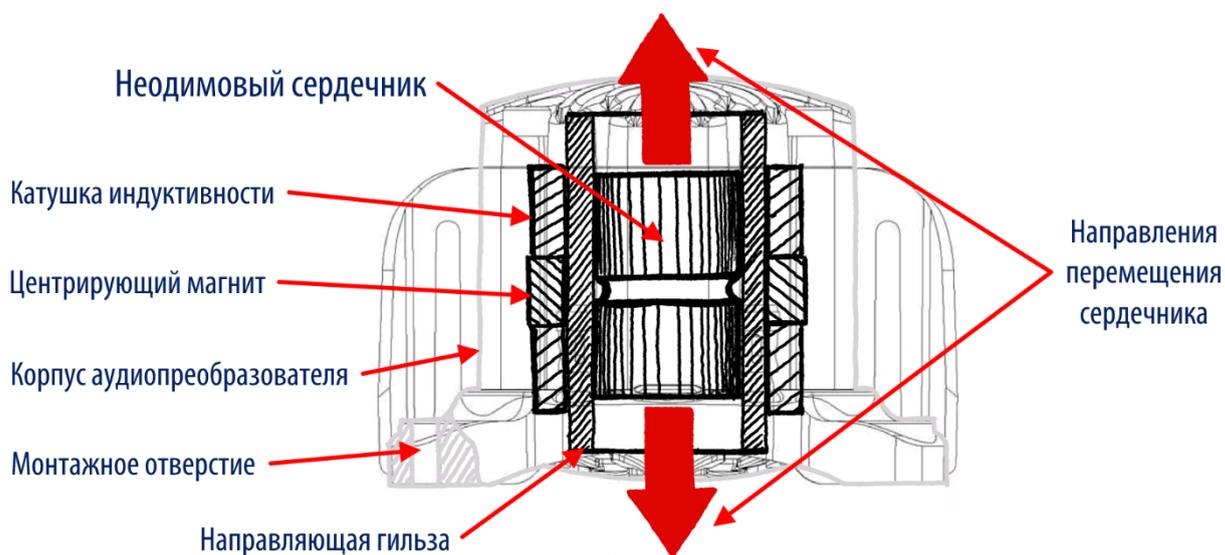


Рисунок 2.3 Конструкция аудиопреобразователя

Для проверки данной идеи был собран простой тестовый стенд, который содержит только основные элементы: аудиопреобразователь с сердечником массой 450 г, закреплённый к листу фанеры толщиной 2 см, усилитель низкой частоты мощностью 300 Вт и компьютер с телевизором. В качестве источника звукового сигнала выбран обыкновенный видеоролик с записью старта ракеты-носителя «Союз» без дополнительной обработки звука. Несмотря на простоту стенда, кинематическая обратная связь получилась достаточно ощутимой.

При использовании аудиодорожек с выделенной низкочастотной составляющей в канале сабвуфера, кинематическая обратная связь получается ещё более выраженной, так как отсекаются «паразитные» высокочастотные составляющие.

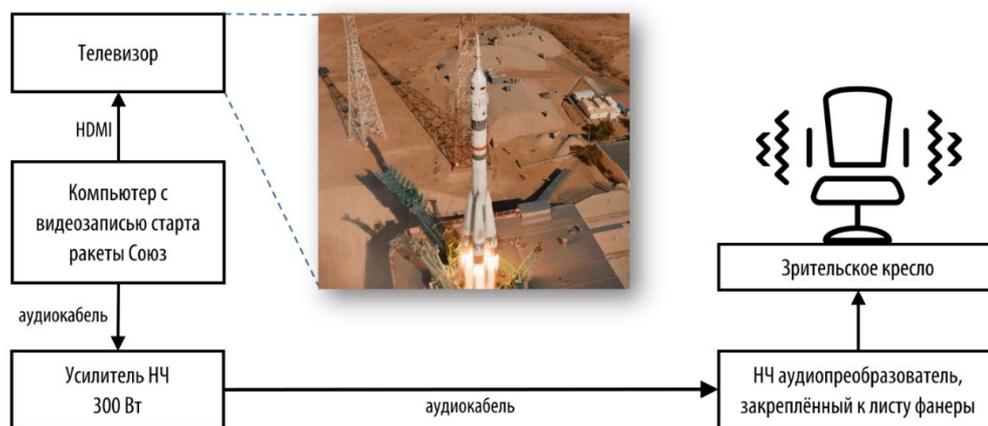


Рисунок 2.4 Схема экспериментального стенда

Однако при выполнении тестирования аудиопреобразователей были выявлены некоторые конструктивные недостатки, выражающиеся в ударах сердечника о демпферы верхней и нижней крышек корпуса на больших амплитудах сигнала, что сопровождается неприятным звуком.

Для полной реализации данного проекта потребуется решить ряд задач, которые позволят предлагаемой доработке стать соответствующей реальной картине событий при выведении и спуске космического корабля:

- провести натурный космический эксперимент по формированию необходимого набора вибрационных данных, заключающийся в размещении на ложементе космонавта внутри спускаемого аппарата 3D-акселерометра и устройства записи его показаний. При этом потребуется получить данные на этапах старта ракеты-носителя и выведения на орбиту, а также при возвращении спускаемого аппарата с экипажем на Землю;

- доработать аудиопреобразователи для минимизации призвуков на высоких амплитудах сигнала;

- разработать программное обеспечение для конвертации записанных показаний акселерометра в аудиофайл, пригодный для дальнейшего использования с аудиопреобразователем;

- модернизировать тренажёры для применения кинематической обратной связи как с точки зрения конструктива, так и с точки зрения программного обеспечения тренажёра.

Заключение

Работа над проектом началась в конце 2023 года. Был выполнен поиск и анализ информации. Выполнена презентация экспериментального стенда на Международной научно-практической конференции. Опубликована статья в сборнике тезисов конференции.

В 2024 году было проведено несколько экспериментальных тестирований с участием действующих космонавтов и специалистов Центра подготовки космонавтов.

В качестве устройства с акселерометром для снятия показаний вибрации в космическом корабле предполагается использовать бортовой планшет, но предварительно необходимо проверить, насколько качественно работает акселерометр при воздействии перегрузок. Для этого потребуется провести эксперимент на центрифуге.

Производятся расчёты для оценки безопасности работы конструкции центрифуги после установки моего устройства.

Для передачи в кабину центрифуги видео высокого разрешения (запись взлёта ракеты) не хватает скорости канала связи. Связано это с тем, что центрифуги для передачи сигналов между статическими и вращающимися узлами, оборудованы специальными вращающимися контактными устройствами, которые имеют ограниченное количество контактов и практически все из них уже задействованы. Сейчас ведутся работы по решению данной задачи.

Для того чтобы разработать конвертер «сырых» текстовых данных от акселерометра в полезный сигнал для аудиопреобразователей я изучаю язык программирования Python, который мне видится подходящим для решения данной задачи.

Выводы и перспективы:

- Несмотря на простоту стенда, обратная кинематическая связь получилась достаточно ощутимой.
- Средство кинематической обратной связи получило положительный отклик среди космонавтов и специалистов, что делает его применимым для модернизации тренажёров пилотируемых космических кораблей после выполнения обследования на предмет безопасности использования.
- В перспективе возможно применение данного вибрационного устройства на комплексных тренажёрах пилотируемых космических кораблей.

Список литературы

1. Крылов А.И., Спиринов Е.А., Киришанов В.Н. Динамический испытательный стенд с комбинированными микро- и макродинамическими воздействиями /Материалы XI Международной научно-практической конференции «Пилотируемые полёты в космос». – Звёздный городок, 2015.
2. Крылов А.И., Спиринов Е.А., Киришанов В.Н., Гаврик И.Н. Подход к реализации пространственно-временных микродинамических воздействий при перегрузках в центрифугах /Материалы XII Международной научно-практической конференции «Пилотируемые полёты в космос». – Звёздный городок, 2017.
3. Пономарев Н.К. Средства кинематической обратной связи в имитаторах перегрузок для тренажёров пилотируемых космических кораблей /Материалы XV

- Международной научно-практической конференции «Пилотируемые полёты в космос». – Звёздный городок, 2023.
4. Как проходит полет в космос от старта до посадки: рассказывает 100-й космонавт России [Электронный ресурс]: Сетевое издание Men Today. URL: <https://www.mentoday.ru/life/experience/polet-v-kosmos-ot-starta-do-posadki-rasskazyvaet-kosmonavt-oleg-kotov/> (дата обращения: 07.03.2025).
 5. Космические тренажеры: репетиция полета на Земле [Электронный ресурс]: Госкорпорация Ростех. URL: <https://rostec.ru/media/news/kosmicheskie-trenazhery-repetitsiya-poleta-na-zemle/#middle> (дата обращения: 07.03.2025).
 6. «Космос не пахнет» [Электронный ресурс]: ФГБУ «НИИ ЦПК имени Ю.А.Гагарина». URL: <https://www.gctc.ru/print.php?id=1003> (дата обращения: 07.03.2025).
 7. Тренажеры космические [Электронный ресурс]: ВикиЧтение. Большая энциклопедия техники. URL: <https://info.wikireading.ru/84170> (дата обращения: 07.03.2025).
 8. ТСПК Центра [Электронный ресурс]: ФГБУ «НИИ ЦПК имени Ю.А.Гагарина». URL: <https://www.gctc.ru/main.php?id=110> (дата обращения: 07.03.2025).
 9. Факторы космического полета [Электронный ресурс]: ФГБУ «НИИ ЦПК имени Ю.А.Гагарина». URL: <https://www.gctc.ru/print.php?id=940> (дата обращения: 07.03.2025).

**СПУТНИК ОБНАРУЖЕНИЯ ОПАСНЫХ АСТЕРОИДОВ И КОМЕТ,
НАПРАВЛЯЮЩИХСЯ К ЗЕМЛЕ СО СТОРОНЫ СОЛНЦА**

Руппель Максим Дмитриевич

Великий Новгород г, МАОУ "Гимназия №2", 10 класс

Секция: Космические аппараты и ракеты-носители

Научный руководитель: Товарных Геннадий Николаевич, к.т.н., доцент каф. СМ1
МГТУ им. Н.Э. Баумана

Аннотация

В настоящее время плотность населения Земли значительно выросла, а последствия катастрофы, вызванной падением крупного космического тела, могут быть крайне серьёзными. Мировое сообщество предпринимает усилия по контролю за космическим пространством, включая объекты как искусственного, так и естественного происхождения. Однако существует категория опасных объектов, которые невозможно обнаружить с Земли — это астероиды и кометы, приближающиеся к нашей планете под значительным углом со стороны Солнца. Спутник, расположенный в точке Лагранжа L1 системы Земля–Солнце, способен эффективно решать задачи обнаружения таких угроз с минимальными затратами энергии на корректировку орбиты. Кроме того, он может выполнять дополнительные научные задачи, такие как наблюдения Солнца и Земли.

Целью исследований в работе является разработка эскиза и макета спутника для размещения его в первой точке Лагранжа системы Земля-Солнце.

Методы исследования: анализ литературы, исторический метод, математические методы: метод уравнений и неравенств, построение графиков, подготовка эскиза и практическое моделирование.

Выводы и результаты исследований:

Рассмотрены основные методы контроля космического пространства.

Разработаны: эскиз и макет спутника для проведения прикладных и научных исследований в первой точке Лагранжа системы Земля-Солнце.

Произведен расчёт затрат топлива.

Проведены расчеты электронно-оптических инструментов спутника.

Произведен расчет обнаружительной способности.

Произведены оценочные расчеты по дополнительным инструментам и научным задачам спутника.

Введение

Астероиды представляют серьезную опасность для Земли в общем и для хозяйственной деятельности человека в частности. Несмотря на то, что событие, когда крупный астероид врежется в Землю не очень частое, но регулярное, оказывается важным иметь возможность вовремя обнаружить такой опасный космический объект и по возможности предупредить о нём и подготовиться. На 2024 год известно около восьми случаев, когда удалось заблаговременно обнаружить астероид и предсказать его место падения. Из недавних случаев – астероид, упавший в Якутии 4 декабря 2024 года [1]. Несмотря на то, что много обсерваторий занимаются поиском астероидов, нерешенной проблемой является поиск астероидов, летящих со стороны Солнца. В данном проекте предлагается разработать космический аппарат (КА), который сможет обнаруживать такие астероиды из точки Лагранжа L1, наблюдая в сторону Земли, он получит отличные условия освещенности таких объектов и лучшие возможности по их обнаружению. В качестве дополнительных возможностей, такой аппарат сможет выполнять прикладные задачи по контролю вывода спутников на высокоэллиптические и геостационарные орбиты, наблюдения освещенной стороны Земли, а также научные задачи по наблюдению Солнца.

Основная часть

Историческая часть

На заре активного освоения космического пространства возникла необходимость создания специальных средств наблюдения и обработки измерительной информации. Эти средства позволяли бы определять орбиты иностранных и отечественных КА с отказавшей или отработавшей свой ресурс бортовой аппаратурой, а также фрагменты ракет-носителей, вышедшие на орбиту. Совокупность таких инструментов получила название системы контроля космического пространства (СККП).

В настоящее время СККП сопровождает более 50 000 объектов и космических аппаратов и оценивает количество опасных объектов на орбите Земли размером более 1 см в количестве 1 млн. объектов.

Мировые обсерватории, занимающиеся обнаружением астероидов [2]:

- Pan-STARRS и Catalina Sky Survey при поддержке НАСА;
- Инфракрасный телескоп NEOWISE;
- Проекты Лаборатории реактивного движения и система из четырёх телескопов Atlas;

- Смитсонская астрофизическая обсерватория (SAO) и Центр малых планет (Minor Planet Center, MPC). Вместе они запустили проект «Hubble Asteroid Hunter», с помощью которого любой желающий может помочь астрономам найти треки новых астероидов.

Российские обсерватории:

- Глобальная сеть телескопов-роботов МАСТЕР под руководством профессора В. М. Липунова; [3]
- Телескоп «Синтез» (он же «АСТ-1200»), который с 1990 до февраля 2018 года находился на консервации в Крымской астрофизической обсерватории. В 2018 году младший научный сотрудник Сергей Назаров поставил задачу восстановить его. В настоящее время телескоп «Синтез» стал полноценным научным прибором, в котором сочетаются аналоговая оптика и современные электронные компоненты. [4]

Проблематика

Несмотря на то, что первый астероид (Церера) был обнаружен в 1801 году, оценить реальное количество астероидов удалось только с появлением фотографии и телевизионных камер, которые сильно упростили задачу фиксации и обнаружение астероидов.



Рисунок 2.1 – Количество обнаруженных астероидов по годам [5]

После 2000 года количество обнаруженных астероидов стало стремительно расти и стало очевидным, что их численность огромна. При этом оказалось, что потенциально опасных околоземных астероидов также очень много.

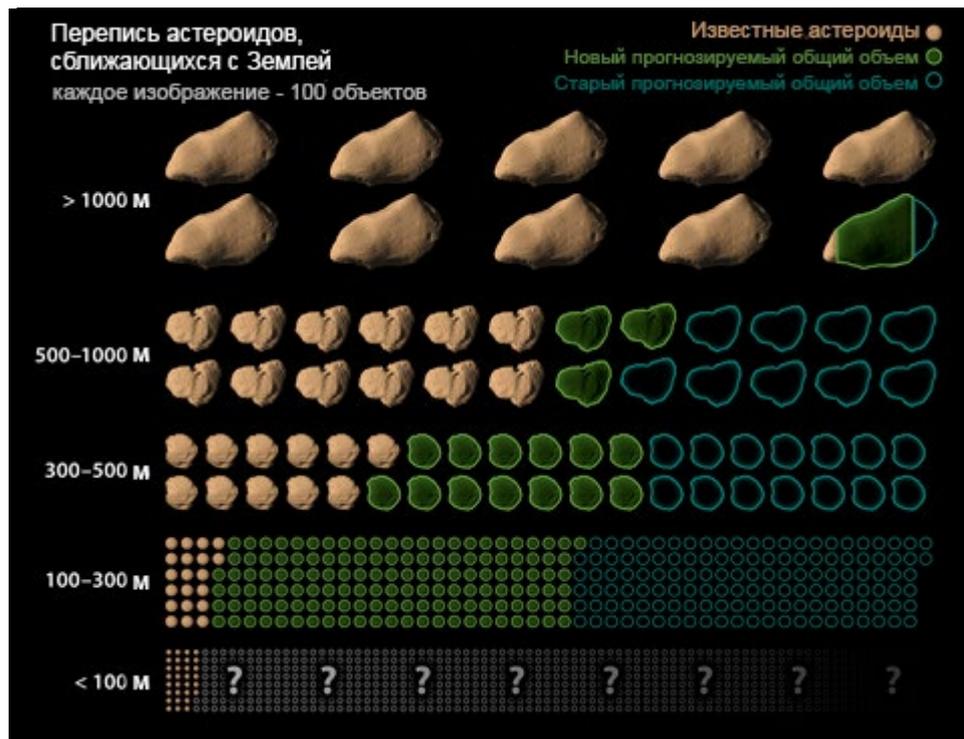


Рисунок 2.2 – Количество и размер астероидов (потенциальных и уже обнаруженных) [5]

Разумеется, одно из самых страшных событий, связанных с проблемой контроля космического пространства, стало падение Челябинского метеорита [6] — каменного, обыкновенного хондрита, упавшего 15 февраля 2013 года в Челябинской области. На высоте 23,3 км полёт метеороида завершился взрывом мощностью 100–200 кт в тротиловом эквиваленте. Последствия падения были серьёзными:

- Взрывной волной были выбиты окна, двери;
- Удар пришёлся на системы вентиляции зданий;
- Рухнула часть стены постройки на территории цинкового завода.

Всего пострадало более семи тысяч зданий. Чрезвычайное происшествие (ЧП) не привело к гибели людей, но травмы получили свыше 1,6 тысячи человек — в основном из-за порезов стёклами выбитых окон. Объект приблизился к Земле незамеченным до входа в атмосферу, отчасти потому, что его радиант (направление источника) находился близок к Солнцу.

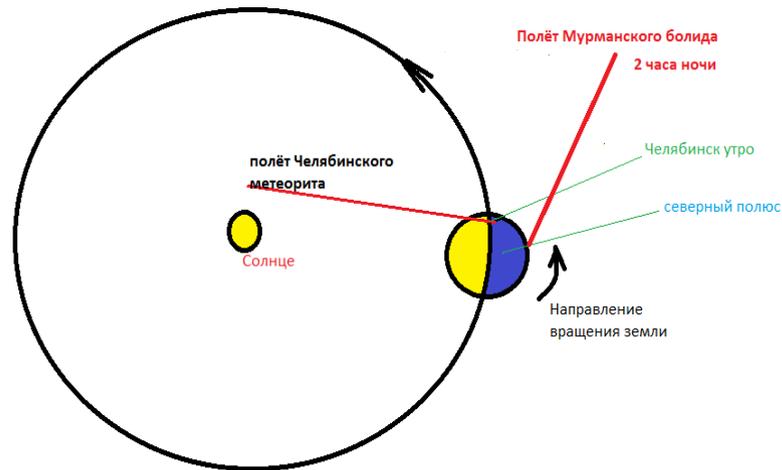


Рисунок 2.3 – Эскиз орбиты Челябинского метеорита

Если вспоминать историю открытия известной межзвездной кометы Г. В. Борисова, то она также направлялась со стороны Солнца. Именно это позволило её обнаружить астроному-любителю, а не автоматизированным системам.

Таким образом, астрономы приходят к выводу, что значительная часть неба, особенно вблизи Солнца, остается вне зоны контроля наземных инструментов.

Варианты решения

На текущий момент для обнаружения строятся телескопы с большим углом зрения, но ни один из них не сможет обнаруживать астероиды, направляющиеся к Земле со стороны Солнца. Но для проработки такого решения необходимо разобрать существующие варианты решения задачи обнаружения астероидов.

Современный телескоп для обнаружения астероидов (максимально производительный): Pan-STARRS представляет из себя комплекс из четырех телескопов (в настоящее время работает два) каждый зеркалом размером 1.8 метра. Для обработки изображений привлекаются мощные сервера с огромной производительностью.

Аппаратное обеспечение IPP-кластера

Ядра = 820 (сейчас) + 564 (к ноябрю 2011) = всего 1480 в МНРСС

Объем хранилища = 1.34 петабайт (сейчас) + 52 Тбайт (к ноябрю 2011) = 1.86 петабайт в МНРСС

Медленное отображение необработанных данных = 360 терабайт (сейчас) + 400 (к 2012) = 760 Тбайт в АTRC



Рисунок 3.1 – Фото сервера обработки телескопа LINEAR. Количество процессоров более 1400, объём хранилища более 1.8 петабайт [7]

Метод работы таких широкопольных телескопов заключается в том, что телескоп снимает отдельную площадку в течение 30-180 секунд и переводится на следующую площадку. Астероиды обнаруживаются как разность изображений площадки между ночами.

На орбиту вывести телескоп с зеркалом 1.8 метра, а также суперкомпьютер не представляется возможным, поэтому отдельно имеет смысл рассмотреть решения, которые применяют астрономы любители для обнаружения астероидов. Таким телескопом, который в последние годы успешно обнаруживает новые объекты, является телескоп «Синтез» Крымской астрофизической обсерватории. Для обнаружения используется телескоп с зеркалом 350 мм, а для обработки достаточно современного компьютера [3]. Сравнение с каталогом PS1 позволило обнаружить на сумме большинство звезд яркостью 22m. Основой алгоритма является непрерывное наблюдение за одним полем космического пространства в течение ночи.

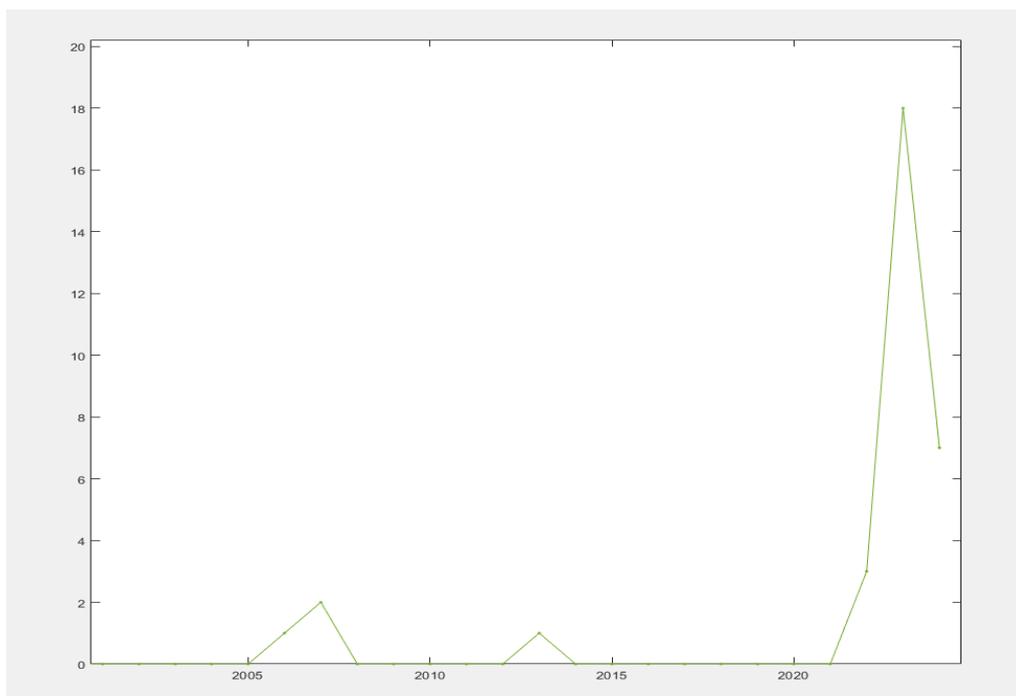


Рисунок 3.2 - Количество открытых астероидов КРАО по годам. Всплеск открытий связан с работой телескопа «Синтез» [4]

Предлагаемое решение (исследовательская часть)

Выбор орбиты спутника

Наилучшие условия обнаружения наблюдения и астероидов это:

- максимальная фаза освещенности;
- наибольший угол наблюдения от Солнца;
- постоянство орбиты и режимов работы.

Из множества возможных орбит наиболее стабильными и представляющими интерес для рассмотрения являются орбиты в точках Лагранжа.

Точка L2 и L3 не подходят для рассмотрения, так как в одном случае (точка L2) потенциально опасные объекты не освещены Солнцем, а в случае точки L3 Солнце будет в значительной степени мешать задаче обнаружения.

Точки L4 и L5 позволяют обнаружить объекты на значительном удалении от Земли, а также наблюдать объекты, которые только приближаются к Солнцу. Точки имеют хорошую радиосвязь с пунктами приёма данных на Земле. В то же время фаза освещенности объектов не может быть более 50%, а для надежного обнаружения потребуется (желательно) два спутника по одному в точке L4 и L5.

Наилучшие условия обнаружения предлагает точка Лагранжа L1. К сожалению, стоит отметить, что радиосвязь из этой точки будет в значительной степени осложнена Солнцем, но задача радиосвязи решалась ранее другими аппаратами.

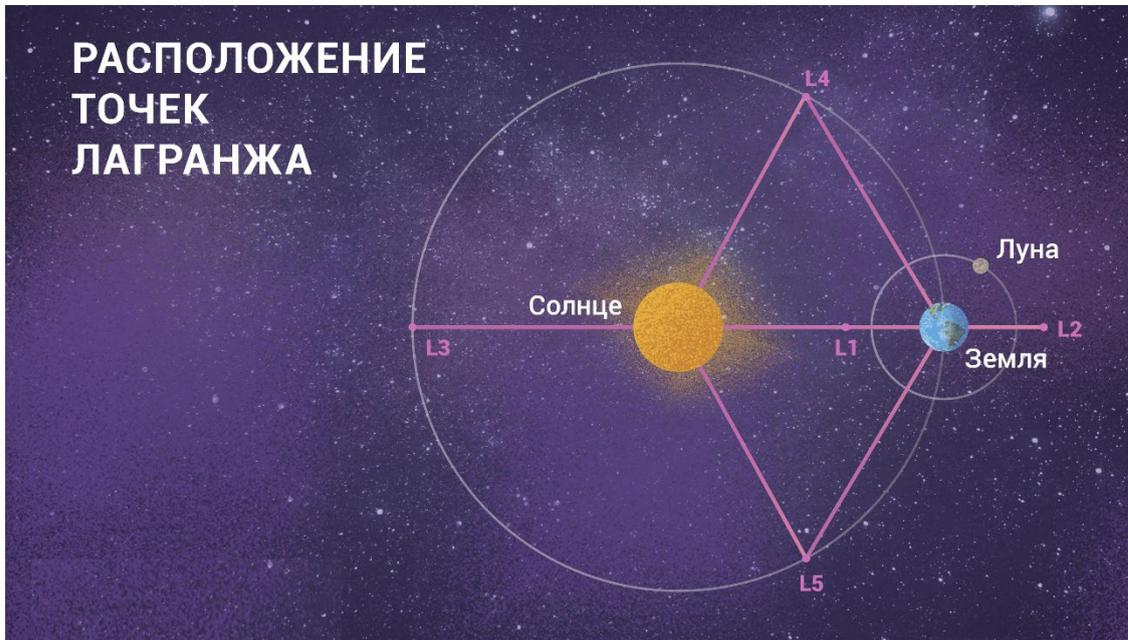


Рисунок 4.1 – Расположение точек Лагранжа системы Земля-Солнце

В системе Солнце — Земля точка L1 может быть идеальным местом для размещения космической обсерватории для наблюдения Солнца, которое в этом месте никогда не перекрывается ни Землёй, ни Луной. Первым аппаратом, работавшим вблизи этой точки, был запущенный в августе 1978 года аппарат ISEE-3. Аппарат вышел на периодическую гало-орбиту вокруг этой точки 20 ноября 1978 года и был сведён с этой орбиты 10 июня 1982 года (для исполнения новых задач). На такой же орбите с мая 1996 года работает аппарат SOHO. Аппараты ACE, WIND и DSCOVR находятся на квазипериодических орбитах Лиссажу близ этой же точки, соответственно, с 12 декабря 1997, 16 ноября 2001 и 8 июня 2015 года. В 2016—2017 годах также в окрестностях этой точки проводил эксперименты аппарат LISA Pathfinder.

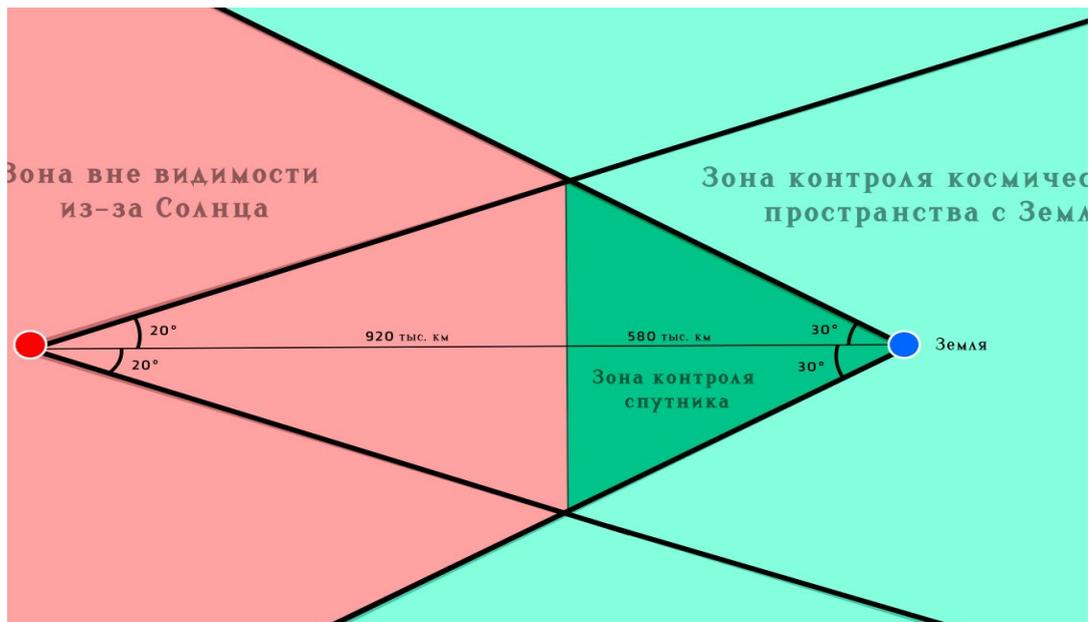


Рисунок 4.2 – Визуализация поля зрения спутника из точки Лагранжа L1

Методы обнаружения

Пороговый метод обнаружения: благодаря отсутствию атмосферы, можно применять пороговый метод обнаружения астероидов. Все объекты, которые не относятся к звездам и имеют яркость выше некоторой величины, можно считать астероидами и обрабатывать. Минусом алгоритма является низкая чувствительность, высокая вероятность ложного обнаружения.

Метод перебора всех вероятных скоростей объекта в серии кадров (метод Чернышева-Назарова на телескопе «Синтез») [4]. Метод очень вычислительно затратный и требует относительно длительного наблюдения одной и той же части сферы для накопления статистики (от нескольких десятков минут до часов). Метод показал свою высокую эффективность и реализуемость на мощностях персонального компьютера.

Углы обзора

Угол обзора должен обеспечивать контроль пространства, который нет возможности контролировать с Земли, но при этом не быть очень большим, чтобы не усложнять спутник.

Минимальное время предупреждения об астероиде определяется исходя из удаления спутника L1 и максимальной скоростью астероида равным 30 км/с (скорость на орбите Земли). Таким образом, время предупреждения будет от 20 часов (при обнаружении объекта сразу на удалении L1) до 10 часов, при обнаружении объекта на половине дистанции от спутника до Земли.

Будем считать, что с Земли есть возможность контролировать объекты с удалением до 30 градусов от Солнца. Таким образом угол обзора телескопа спутника должен составлять

60 градусов. Объем контролируемого пространства составит: около одного экса кубокилометров (10^{18} км).

Определим для каждой из камер полезной нагрузки фокусное расстояние телескопа, а также для удобства составим таблицу со всеми этими данными для каждой камеры в зависимости от её предназначения. Для этого выведем формулы.

Чтобы определить угол обзора камеры, достаточно знать радиус обзораемого объекта (области) и расстояние до него. Это следует из равнобедренного треугольника, высота в котором является также медианой и биссектрисой.

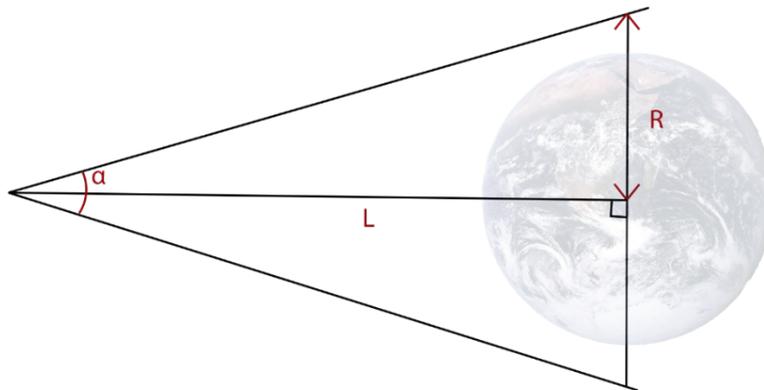


Рисунок 4.3 – Эскиз расчета угла поля зрения телескопа при заданном радиусе обзора R и расстоянии L

$$\alpha = 2 \arctg \frac{R}{L} \quad (3.1)$$

- α – угол обзора камеры;
- R – радиус обзораемой области;
- L – расстояние до неё.

Теперь, выведем формулу для расчета фокусного расстояния. В телескопе у нас собирающая линза, а фокусное расстояние в нём равно расстоянию до матрицы камеры (т.е. $F = d$). Так как оптическая ось объектива/телескопа проходит через центр матрицы камеры, то мы должны взять только половину стороны матрицы. Соответственно для этого нам также необходимо знать размер каждого пикселя (a). Выходим снова на прямоугольный треугольник.

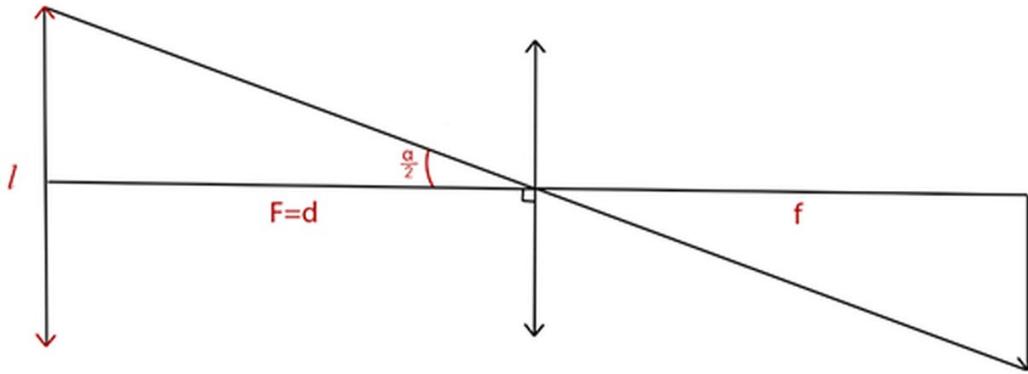


Рисунок 4.4 – Эскиз расчета фокусного расстояния F телескопа при заданном угле поля зрения α

$$F = \frac{la}{2tg \frac{\alpha}{2}}, \quad (3.2)$$

где

- l – сторона матрицы;
- a – сторона одного пикселя в миллиметрах;
- F – фокусное расстояние в миллиметрах.

Используя данные формулы, рассчитаем параметры полезной нагрузки КА.

Таблица 3.1 – Результаты расчета оптической схемы ПН КА

Камера	Назначение	Обозреваемый радиус	Угол	Фокусное расстояние
К1..К4	обнаружение	-	20 град	100 мм
К0	наблюдение Земли и геостационара	42378 км	7 град	300 мм
ИК	наблюдение Земли в инфракрасном диапазоне	12756 км	1.15 град	2000 мм
S0	наблюдение Солнца	1.4 млн. км	1.17 град	1800 мм
S1	наблюдение Солнечной короны, коронограф	3.5 млн. км	3 град	700 мм

Телескопы и матрицы

Для расчета рассмотрим пример барьерного телескопа Геннадия Владимировича Борисова (открывшего первую и единственную межзвездную комету) Genop представленного на сайте Астрономического Научного Центра.

Характеристики отдельного инструмента:

- Диаметр входного зрачка – 192;
- Эффективное фокусное расстояние - 296 мм (1:1,54);
- Угловой диаметр поля зрения - 10° ;
- Линейный диаметр поля зрения - 52,1 мм;
- Диапазон длин волн - 4500-8500 Å;
- Задний отрезок - 31 мм;
- Среднеквадратический диаметр изображений звезды в интегральном диапазоне 4500-8500 Å, оптическая ось – край поля - 3,0- 4,0 мкм;
- Коэффициент центрального экранирования по диаметру - 0,50;
- Относительная освещенность по полю - 1.00-0,75;
- Максимальная дисторсия изображения - 0,5%;
- Длина оптической схемы - 280 мм;
- Масса телескопа (труба с оптикой в сборе) - 21 кг.

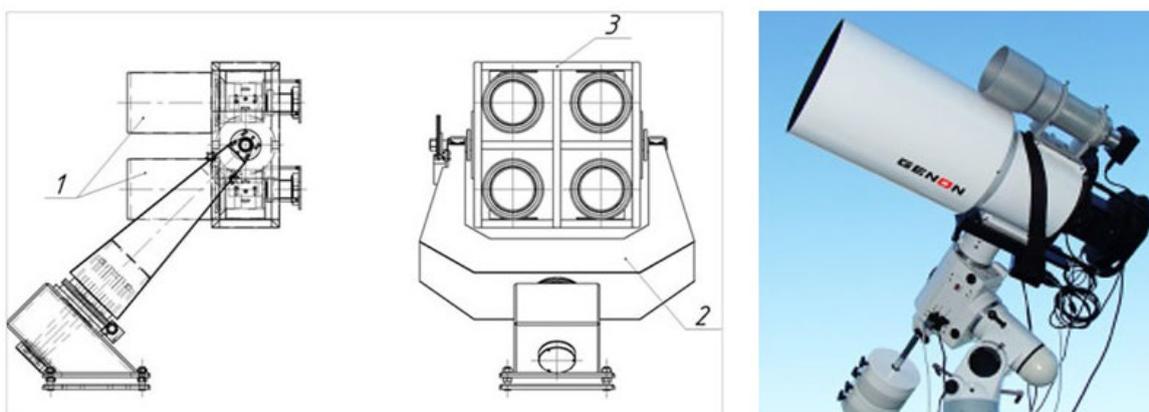


Рисунок 4.5 – Телескоп Genon АНЦ [8]

В качестве камер обнаружения рассмотрим камеры Нева4040 НПК «Фотоника», разрешение каждой из которых составляет 4096x4096 элементов.



Рисунок 4.6 – Астрокамера Нева4040 [9]

Угол поля зрения каждой камеры составит 6 градусов. Что обеспечит при применении четырех телескопов (по аналогии с представленным на сайте АНЦ) угол контроля: четыре поля по 7x7 градусов.

Алгоритм обнаружения

Алгоритм обнаружения предполагает наблюдение за одной и той же точкой пространства с небольшим временем экспозиции (10-30 секунд). Непрерывно получаемый набор кадров подвергается разделению на серии по 30-100 кадров. В каждом кадре выделяются потенциальные объекты и проверяется, что они находятся на одной траектории [4].

Оценка размера обнаруживаемых астероидов

Для оценки размера и определения обнаружительной способности воспользуемся специализированной программой Чернышева-Назарова, применяемой для оценки телескопа «Синтез» [4]. Для этого нам придётся задать данные объектива Genon и камеры Нева4040. Расчет позволяет оценить предельную яркость объекта, от времени наблюдения. Для сравнения добавим камеру Нева400 (разрешением 2048x2048 пикселей). Отметим, что камеру меньшего разрешения имеет смысл применить в случае недостаточности вычислительных мощностей спутника для задачи обнаружения астероидов.

Задаваемые параметры для расчёта:

- порог соотношения сигнал объекта/шум: 10;
- яркость фона: 23m;
- угловой размер размытия в оптике: 1.5 пикселя или 10 угловых секунд;
- время экспозиции: от 30 до 300 секунд.

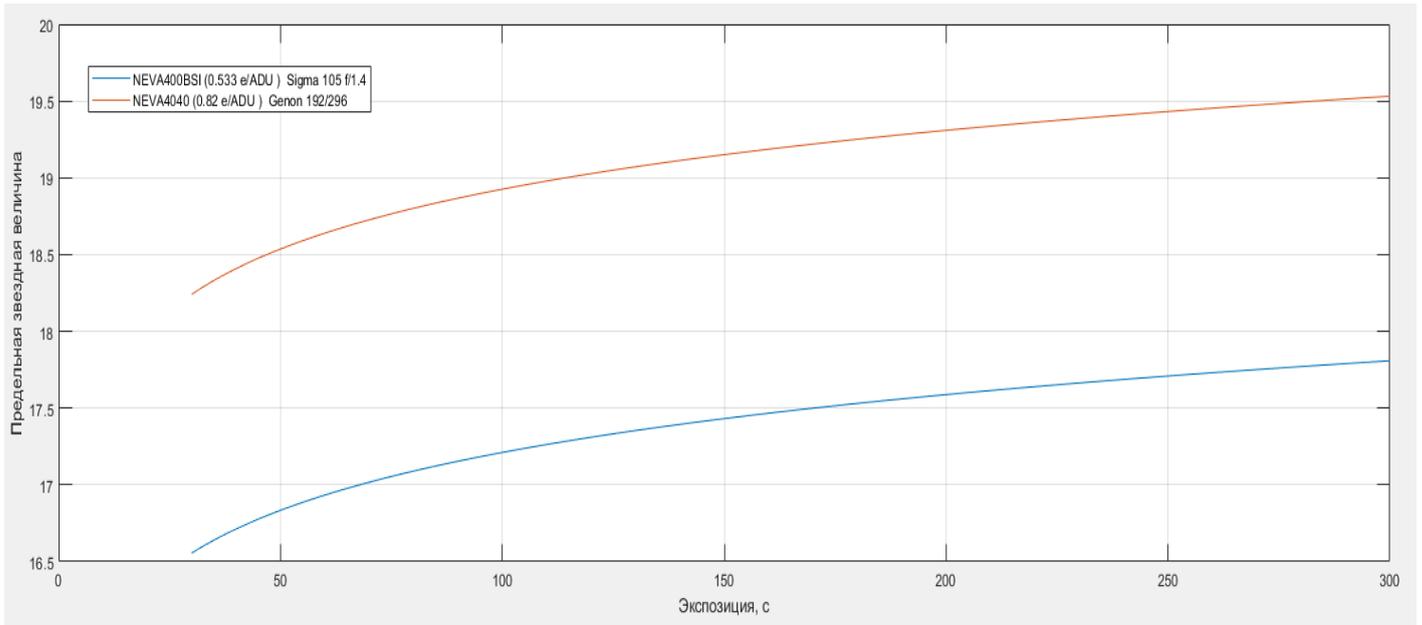


Рисунок 4.7 – Результат расчета предельной яркости объекта в программе «Синтез» для камер Нева4040 и Нева400

Результат расчета показывает (для камеры Нева4040), что при длительности наблюдения 30 с можно обнаружить объекты яркостью 18 зв.в. А при длительности наблюдения 300 с слабее 19 зв.в. Следует отметить, что так как спутник наблюдает неподвижно одну и ту же область пространства (с незначительным дрейфом), то имеется возможность накапливать суммарные длительные экспозиции.

Оценка яркости позволяет нам определить минимальный размер астероида. Для этого произведём расчёт абсолютной видимой величины H .

Воспользуемся формулой для вычисления видимой звёздной величины [11]:

$$H = m - 5 \lg \left(\frac{d_{BS} d_{BO}}{0_d} \right) - 2.5 \log_{10} q(\alpha) \quad (4.6.1)$$

из формулы, связывающей абсолютную звёздную величину и видимую звёздную величину тела, получаем:

$$H = m - 5 \lg \left(\frac{d_{BS} d_{BO}}{0_d} \right) + 2.5 \log_{10} q(\alpha) \quad (4.6.2)$$

где α – фазовый угол, т.е. угол между линиями Солнце-тело и тело-наблюдатель. Но, так как мы находимся в точке Лагранжа L_1 , то мы имеем отличную освещенность, поэтому возьмём фазовый угол равный 1, приведя формулу к виду:

$$H = m - 5 \lg \left(\frac{d_{BS} d_{BO}}{0_d^2} \right) \quad (4.6.3)$$

где:

- H – абсолютная звездная величина;
- m – видимая звездная величина (18 зв.в.);
- d_{BS} – расстояние от Земли до Солнца (150 млн. км);
- d_{BO} – расстояние от КА до Земли (1,5 млн. км);
- 0_d – коэффициент преобразования единиц измерения, равен 1 астрономической единице, среднему расстоянию между Землей и Солнцем (150 млн. км).

Для определения размера объекта, необходимо знать его геометрическое альbedo. Воспользуемся статьей Д.Ф. Лупишко [12] и возьмем из неё альbedo для астероида класса С, так как из всех известных астероидов они самые распространенные, а также их альbedo наименьший, что позволяет учесть наихудший исход – 0.06.

Для подсчета используем формулу зависимости абсолютной звездной величины, геометрического альbedo и диаметра космического объекта [11]:

$$H = -5 \lg \frac{D \sqrt{p}}{1329} \quad (4.6.4)$$

где

- D – диаметр космического объекта в километрах;
- p – геометрическое альbedo (0.06).

Из неё находим размер объекта:

$$D = 1329 \frac{10^{-\frac{H}{5}}}{\sqrt{p}} \quad (4.6.5)$$

Объединим формулы:

$$D = 1329 \frac{10^{-\frac{m-5 \lg \left(\frac{d_{BS} d_{BO}}{0_d^2} \right)}{5}}}{\sqrt{p}} \quad (4.6.6)$$

Подставив значения в формулу, мы получаем 13.6м. Для самопроверки воспользуемся таблицей [10]. Для абсолютной видимой величины $H=28$, даже при альbedo 5%, размер объекта составит порядка 15м. Что примерно соответствует оценке Челябинского метеорита диаметром примерно 20 метров и близко к полученному нами значению [6].

albedo	0.30	0.25	0.20	0.15	0.10	0.05
H						
30.0	0.0025	0.0027	0.0030	0.0035	0.0043	0.0060
29.5	0.0031	0.0034	0.0038	0.0044	0.0054	0.0076
29.0	0.0039	0.0043	0.0048	0.0055	0.0068	0.0096
28.5	0.0049	0.0054	0.0060	0.0069	0.0085	0.012
28.0	0.0062	0.0068	0.0076	0.0087	0.011	0.015
27.5	0.0078	0.0085	0.0095	0.011	0.013	0.019
27.0	0.0098	0.011	0.012	0.014	0.017	0.024
26.5	0.012	0.014	0.015	0.017	0.021	0.030

Рисунок 4.8 – Таблица приблизительных диаметров астероидов для диапазона значений H и альбедо [10]

Дополнительные задачи спутника

Значительным преимуществом проекта является то, что на орбите L1 спутник может решать научные и прикладные задачи, такие как:

Контроль вывода КА на высокоэллиптические и геостационарные орбиты

5.1 Основная камера обнаружения объектов может быть задействована для контроля вывода объектов на высокоэллиптическую и геостационарную орбиту. А также для контроля объектов на геостационарных орбитах.

5.2 Наблюдение Солнца (дополнительный телескоп);

Точка L1 является превосходным местом для наблюдения Солнца;

5.3 Датчики солнечного ветра

Позволят измерять и более точно предсказывать погоду на орбите Земли, что важно для прикладных КА. А также прогнозировать магнитные бури на самой Земле. Отлично показал себя датчик “Плазма-Ф” разработки ИКИ РАН на спутнике-обсерватории “Радиоастрон” [14].

5.4 Наблюдение освещенной части Земли и атмосферы (дополнительный телескоп)

При установке дополнительного телескопа можно решать научные задачи по наблюдению освещенной стороны Земли. Благодаря постоянству условий наблюдения можно оценивать такие параметры как интенсивность отражения солнечного излучения. Фактическое

измерение отражения поможет более точно спрогнозировать глобальное потепление или похолодание на Земле.

Расчет орбиты вывода аппарата в точку L1

Задачей расчета является:

- выбор орбиты выведения;
- расчет затрат скорости и массы топлива для перелета в точку L1 и поддержания орбиты;
- выбор ракеты-носителя и разгонного блока аппарата;
- оценка допустимой массы полезной нагрузки.

Для расчета орбиты будем исходить из опыта выведения и работы космической обсерватории «Спектр-РГ», работающей в точке L2. Известно, что обсерватория имеет примерную конечную массу в точке L2 ~2700 кг. Вывод обсерватории производится РН «Протон-М» и разгонным блоком «ДМ-03». На рисунке 6.1 представлена схема перелёта, в соответствии с которой произведём расчет.

Исходные данные для вычисления и константы:

- Гравитационная постоянная Земли:

$$G = 6,67430 \cdot 10^{-11} \text{ Н} \cdot \text{м}^2 \cdot \text{кг}^{-2};$$

- Масса Земли:

$$M_{\text{Земли}} = 5.97 \cdot 10^{24} \text{ кг};$$

- Гравитационный параметр Земли:

$$\mu_z = G * M_{\text{Земли}} = 3.988 \cdot 10^{14} \frac{\text{м}^3}{\text{с}^2}.$$

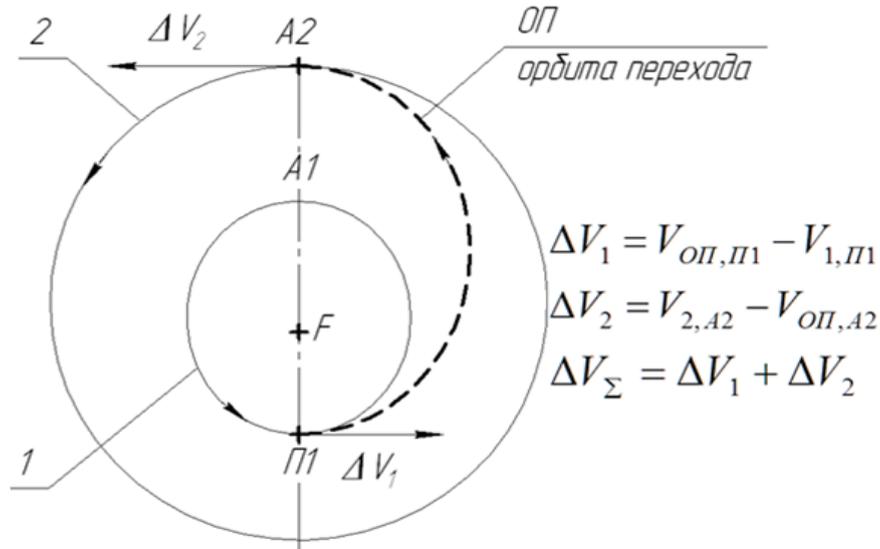


Рисунок 6.1 - Схема перелёта КА с опорной орбиты вокруг Земли (орбита 1) на орбиту в точку Лагранжа L1 (орбита 2)

Формула расчета скорости на орбите

$$V = \sqrt{\left(G(M_{\text{земли}} + m_{\text{ка}}) \left(\frac{2}{r} - \frac{1}{a} \right) \right)}, \quad (6.1)$$

где

- r – расстояние между объектами;
- a – большая полуось орбиты;
- $m_{\text{ка}}$ – масса КА.

Массой космического аппарата по сравнению с массой Земли можно пренебречь, а для круглой орбиты $r = a$, формула упрощается до:

$$V = \sqrt{\frac{GM_{\text{земли}}}{a_{\text{оп}}}} = \sqrt{\frac{\mu_3}{a_{\text{оп}}}} \quad (6.2)$$

Скорость на круговой (опорной) орбите вокруг Земли $V_{\text{кр1}}$ при $r_1 = 6571$ км (или 200 км над поверхностью Земли).

$$V_{\text{кр1}} = \sqrt{\frac{\mu_3}{r_1}} = 7,79 \text{ км/с} \quad (6.3)$$

Требуемое приращение скорости для перехода с опорной орбиты на отлетную (переходную) орбиту в точку L1 будет определяться большой полуось орбиты перехода с апогеем в точке L1 и перигеем на опорной орбите:

$$a_{\text{оп}} = \frac{r_1 + r_2}{2} \quad (6.4)$$

где $r_2 = 1.5$ млн. км – радиус орбиты точки Лагранжа L1

Требуемое изменение скорости для выхода на орбиту перехода:

$$\Delta V_1 = V_{\text{ОП.П1}} - V_{\text{кр1}} = \sqrt{\mu_3 \left(\frac{2}{r_1} - \frac{1}{a_{\text{оп}}} \right)} - \sqrt{\frac{\mu_3}{r_1}} = 3.2 \text{ км/с} \quad (6.5)$$

Требуемое изменение скорости для выхода в точку L1 с орбиты перехода:

$$\Delta V_2 = V_{\text{ОП.Л2}} = \sqrt{\mu_3 \left(\frac{2}{r_2} - \frac{1}{a_{\text{оп}}} \right)} = 48 \text{ м/с} \quad (6.6)$$

Формула Циолковского расчета позволяет определить количество топлива для обеспечения требуемого изменения скорости:

$$\Delta V = J \cdot \ln \frac{M_0}{M_k} \quad (6.7)$$

где

- J – удельный импульс двигателя;
- M_0 – начальная масса КА;
- M_k – конечная масса КА.

При известной конечной (требуемой) массе, можно рассчитать начальную и массу топлива:

$$M_0 = M_k \cdot e^{\frac{\Delta V}{J}} \quad (6.8)$$

$$M_T = M_0 - M_k \quad (6.9)$$

Удельный импульс двигателя на НДМГ- горючего (РБ – «Фрегат»):

$$J_{\text{НДМГ}} = 3200 \text{ м/с}$$

Удельный импульс двигателя на керосин-кислород горючего (РБ – «ДМ»):

$$J_{\text{керосин}} = 3000 \text{ м/с}$$

Примем затраты скорости на проведение коррекции по аналогии с расчетами обсерватории «Спектр-РГ» [15] по 24 м/с в год, исходя из срока активного существования 10 лет и двукратным запасом:

$$\Delta V_{\text{коррекции}} = 480 \text{ м/с}$$

Результаты расчета сведём в таблицу 6.1. Учтем, что разгонный блок отделяется после выдачи импульса при переходе с круговой на орбиту перехода.

Таблица 6.1 – Результат расчета массы КА, ПН и топлива

	Вариант запуска на РН «Протон-М»	Вариант запуска на РН «Союз-2»
Масса КА+РБ на НОО	18500	8500 кг
Масса, тип РБ, удельный импульс	3250 кг, «ДМ-03», 3000 м/с	1045 кг, «Фрегат», 3200 м/с
Масса топлива для перехода на ОП	12160 кг	5387 кг
Масса КА на ОП	3088 кг	2068
Масса топлива для перехода с ОП в L1	46 кг	31 кг
Масса топлива для проведения коррекции	440 кг	500 кг
Масса платформы [16]	850 кг	850 кг
Масса ПН (за вычетом платформы и топлива)	1752 кг	890 кг

В связи с тем, что РН «Протон-М» является РН тяжелого класса, а также количество носителей ограничено, то целесообразно выбрать РН среднего класса «Союз-2», разгонный блок «Фрегат» и платформу «Навигатор». Таким образом, для полезной нагрузки масса составит:

$$m_{\text{ПН}} = 890 \text{ кг}$$

Также посчитаем время перелёта в точку L1 с околоземной орбиты. Это время равно половине одного оборота КА на переходной орбите:

$$T = \pi \sqrt{\frac{a^3}{\mu}} \quad (6.10)$$

Время перелёта составит около 38 суток.

Для иллюстрации отобразим схему перелёта и рабочую орбиту по аналогии со схемой перелёта «Спектр-РГ» в точку Лагранжа L2 [20].



Рисунок 6.2 – Схема перелета космического аппарата с орбиты Земли в точку Лагранжа L1

Состав и структура космического аппарата

Космический аппарат должен содержать в своём составе следующие обязательные элементы:

- Навигационная платформа;
- Датчики ориентации;
- Широкопольный телескоп контроля космического пространства;
- Бортовой вычислитель;
- Солнечные панели;
- Аккумуляторные батареи;
- Баки с топливом,

А также может содержать дополнительную полезную нагрузку:

- Телескоп наблюдения Солнца;
- Телескоп наблюдения солнечной короны (коронограф);
- Телескоп наблюдения Земли в ИК-диапазоне;

- Датчики солнечного ветра.

Навигационная платформа “Навигатор” является центральным “элементом” космического аппарата. Предлагается применить проверенную и надежную платформу производства НПО им. Лавочкина [16]. Данная платформа уже применялась в космических проектах, таких как: “Электро-Л”, “Арктика-М”, “Спектр-Р”, “Спектр-РГ” [16] и планируется к применению в будущих проектах. Платформа обеспечивает (по примеру КА «Спектр-РГ»):

- Мощность электропитания (СБ+АКБ): от 600 до 1150 Вт;
- Массу полезной нагрузки: до 2600 кг;
- Заправка топливом: до 570 кг;
- Дальность связи: до 2 млн. км.

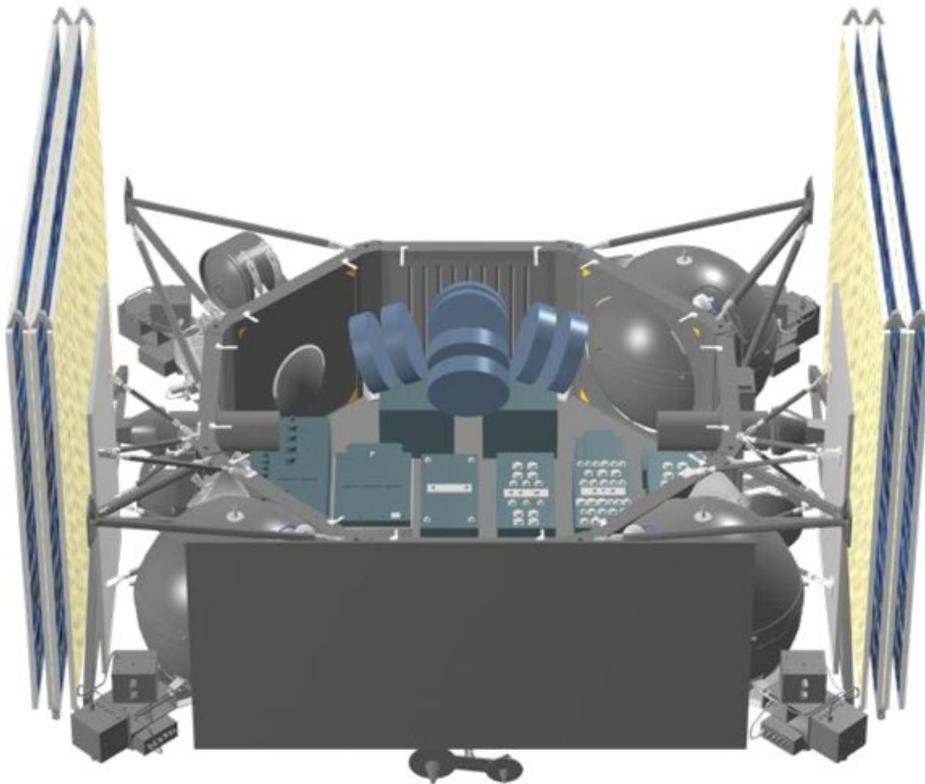


Рисунок 7.1 – Космическая платформа Навигатор [15]

Широкопольный телескоп является основной полезной нагрузкой, его предлагается собрать из пяти идентичных телескопов разработки Геннадия Борисова (расчет телескопов КрАО, Теребиж), которые при необходимости могут быть доработаны Крымской Астрофизической обсерваторией [17] с камерами разрешением 4096x4096 пикселей на базе изделий производства АО «Нева Электроника» Нева4040[9]. Полезная нагрузка является специализированным инструментом, поэтому телескоп и камеры нужно будет адаптировать под задачи космического аппарата.

Солнечные телескопы также оснастить аналогичными камерами Нева4040 с целью унификации проекта.

В качестве камеры наблюдения Земли в инфракрасном диапазоне можно использовать болометрическую камеру разработки ИФП СО РАН разрешением 2000x2000 элементов [18].

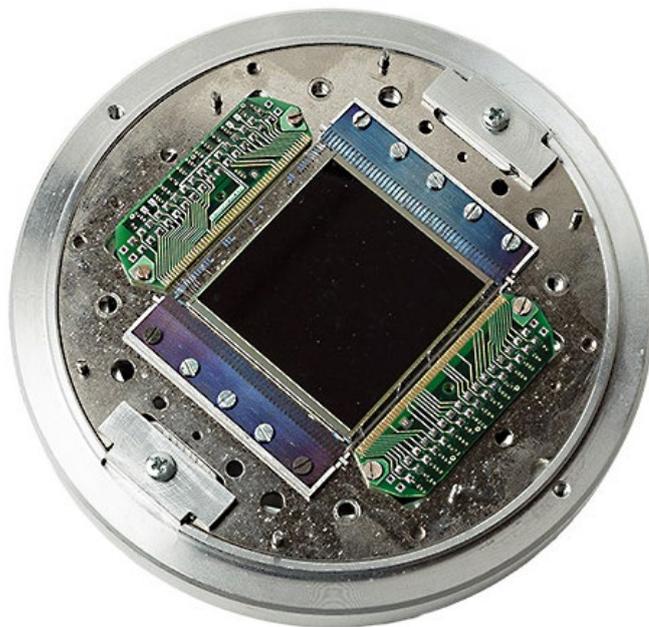


Рисунок 7.2 – Инфракрасная камера высокого разрешения ИФП СО РАН

С целью обеспечения точной ориентации использовать малогабаритные датчики астроориентации АЗДК АО «Азмерит» [19].



Рисунок 7.3 – Датчик астроориентации АЗДК

Остальное оборудование выбрать из ранее применённых в космических проектах «Коронас-Фотон», «Фобос-Грунт», «Радиоастрон», «Спектр-РГ» и др.

Список оборудования космического аппарата представлен в таблице 7.1.

Таблица 7.1 – Предлагаемый состав КА

	Элемент КА	Модель/прототип	Кол-во	Производитель
1	Платформа КА	Навигатор	1	НПО им. Лавочкина
2	Камеры обнаружения	Нева4040 разрешение 4096x4096	5	Нева Электроника
3	Камера инфракрасного диапазона	ГЭС КРТ разрешение 2000x2000	1	ИКИ РАН
4	Камеры солнечного телескопа	Нева4040 разрешение 4096x4096	2	Нева Электроника
5	Астродатчик	АЗДК-2.3	2	Азмерит
6	Датчик солнечного ветра	Плазма-Ф	2	ИКИ-РАН
7	Телескопы обнаружения	Genon	5	КрАО/АНЦ
8	Телескопы наблюдения Солнца	Genon Genon с коронографом	2	КрАО/АНЦ
9	Телескоп ИК	нет прототипа	1	КрАО/ИКИ РАН

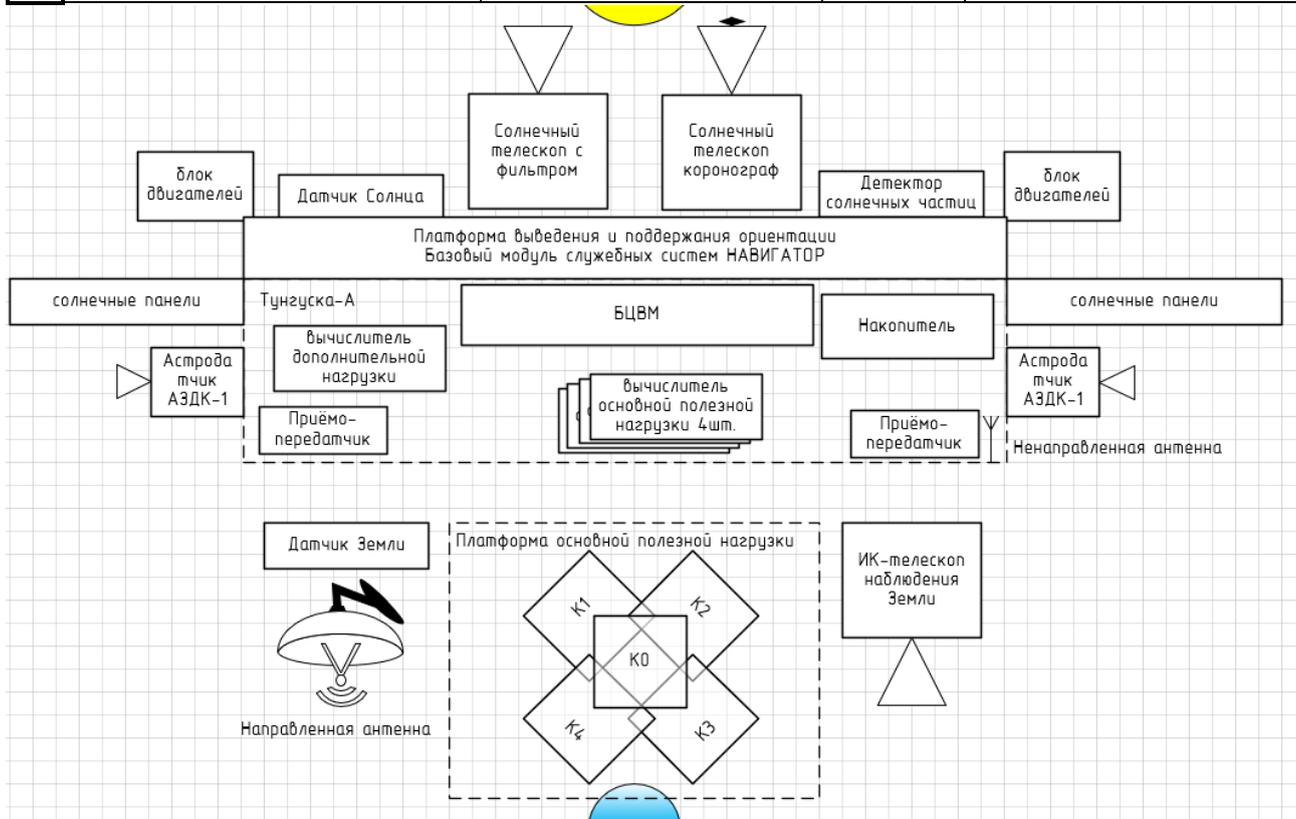


Рисунок 7.4 – Эскизная схема космического аппарата

Стоимость проекта

Оценка ущерба Челябинского метеорита составляет около 1 млрд.рублей, но следует отметить, что главной задачей контроля космического пространства является своевременное предупреждение жителей и исключение человеческих жертв, а ценность человеческих жизней переоценить невозможно.

К сожалению, в свободном доступе очень ограниченное количество информации по стоимости. Оценку стоимости РН «Союз» примем в 1.3 млрд. рублей (исходя из опубликованной суммы в 10.4 млрд. рублей за 8 РН [21]), стоимость РБ - 600 млн. рублей [22], стоимость научной миссии - 1 млрд. рублей.

Оценка наземного сегмента из расчета 10 серверов мощностью хранения информации в объёме 100 ТБ данных по 5 млн. рублей, а также трех станций приёма стоимостью 10 млн. рублей.

Оценку обслуживания примем как 5% от стоимости КА, так как объективных данных для его расчета нет.

Таблица 9.1 – Примерная стоимость проекта

	Элемент миссии	Стоимость
1	РН «Союз 2-1»	1.2 млрд. рублей
2	РБ «Фрегат»	600 млн. рублей
3	стоимость пусковых услуг	1 млрд. рублей
3	КА с проектированием	1 млрд. рублей
4	Наземный сегмент	100 млн. рублей
5	Обслуживание в течении 5 лет	50 млн. рублей
	итого	~4 млрд. рублей (~40 млн. \$)

Стоит отметить, что стоимость реализации в первую очередь определяется стоимостью пусковых услуг и разработки КА. И в общем случае не превышает типовых затрат на космический аппарат связи, но значительно менее затратный по сравнению с научными миссиями, такими как “Спектр-РГ”, “Спектр-Р”, за счет применения типовых решений и российских комплектующих, унификации оборудования.

Заключение

В ходе работы были проведены исследования в объёме:

- рассмотрения проблематики обнаружения космических объектов приближающихся к Земле со стороны Солнца

- рассмотрены современные средства обнаружения космических объектов;
- предложен вариант решения обнаружения объектов, направляющихся к Земле со стороны Солнца, с помощью спутника, располагающегося в точке Лагранжа L1, между Землей и Солнцем на расстоянии 1.5 млн. км.;
- произведён расчёт размера обнаруживаемых объектов. Оценка размера обнаруживаемых астероидов составляет около 15-20 метров и более, что соответствует размеру разрушительного Челябинского метеорита;
- приведён пример алгоритма обнаружения небольших объектов малыми телескопами на базе успешного опыта работы уникального телескопа «Синтез» КрАО и алгоритма Чернышева-Назарова;
- произведен расчет орбиты, скоростей и затрат топлива для достижения точки L1;
- произведен расчет массы космического аппарата, разгонного блока;
- произведен расчет топлива;
- предложены дополнительные функции и оборудование КА, такие как телескопы наблюдения Земли, Солнца и солнечной короны;
- предложен состав космического аппарата;
- Исследование демонстрирует востребованность, а также реализуемость проекта по контролю космического пространства Земли со стороны Солнца, а так же наблюдения Солнца, Земли, космического ветра, на основе уже отработанных технических решений.

Основные критические элементы космического аппарата уже успешно были применены в проектах гамма-обсерватории «Спектр-РГ», радиообсерватории «Радиоастрон», «Коронас-Фотон», Космический аппарат может быть вполне реализован на данном техническом уровне.

Перечень сокращений и обозначений

АКБ – аккумуляторные батареи

ИК – инфракрасная

КА – космический аппарат

КрАО – Крымская астрофизическая обсерватория

НДМГ - Несимметричный диметилгидразин

НПО – научно-производственное объединение

НОО – низкая околоземная орбита

ОП – орбита перехода

РБ – разгонный блок

РН – ракета-носитель

СБ – солнечные батареи

СККП - Система контроля космического пространства

Список использованных источников

1. Гроховский В. РАН назвали успехом предсказанное учеными падение астероида в Якутии. URL: <https://rg.ru/2024/12/04/reg-dfo/v-ran-rasskazali-v-chem-unikalnost-upravshego-v-iakutii-asteroida.html>
2. Еленин Л. Существующие и будущие оптические обзорные системы. URL: <https://astronomer.ru/publications.php?act=view&id=47>
3. Глобальная сеть телескопов МАСТЕР. URL: <https://sai.msu.ru/devices/master.html>
4. С. В. Назаров, А. С. Чернышев. Поиск неизвестных астероидов на телескопе «Синтез» /— Физика космоса: труды 50-й Международной студенческой научной конференции (Екатеринбург, 30 января — 3 февраля. 2023 г.). — Екатеринбург: Издательство Уральского университета, 2023. — С. 404-407.
5. NASA. Discovery Statistics. URL: https://cneos.jpl.nasa.gov/stats/site_all.html
6. М. Я. Маров, Б. М. Шустов ЧЕЛЯБИНСКОЕ СОБЫТИЕ КАК АСТРОНОМИЧЕСКОЕ ЯВЛЕНИЕ // ГЕОХИМИЯ, 2013, № 7, с. 647–649
7. Астероиды и мы URL:<https://habr.com/ru/articles/400317/>
8. Астрономический научный центр. URL: https://www.ancprotek.ru/pg_2_6.html
9. НПК Фотоника. Каталог продукции.
URL:https://nevael.ru/catalog2/astronomicheskie_kamery/834/?ysclid=m76d2g0qox425419058
10. NASA. JPL. Asteroid Size Estimator. https://cneos.jpl.nasa.gov/tools/ast_size_est.html
11. Оценка размера астероида от яркости и расстояния.
URL:https://en.wikipedia.org/wiki/Absolute_magnitude
12. Лупишко Д.Ф. Физические свойства астероидов // Вестник астрономической школы, том 1, 2000
13. Расчет размера астероида. URL: https://cneos.jpl.nasa.gov/tools/ast_size_est.html

14. Зеленый Л.М., Застенкер Г.Н., Петрукович А.А. и др. Эксперимент ПЛАЗМА-Ф на спутнике «Спектр-Р» // Космич. исслед. 2013. Т. 51. № 2. С. 83-87. (Cosmic Research. P. 73–79)
15. Е. А. Михайлов, С. А. Аксенов, Г. С. Заславский, П. В. Мжельский, А.В.Погодин
МЕТОДИКА РАСЧЕТА ПАРАМЕТРОВ СЕРИИ «БОЛЬШИХ» КОРРЕКЦИЙ
ТРАЕКТОРИИ ПОЛЕТА КА «СПЕКТР-РГ» ДЛЯ УЛУЧШЕНИЯ ЕГО
РАДИОВИДИМОСТИ// ПИСЬМА В АСТРОНОМИЧЕСКИЙЖУРНАЛ, 2022, том
48,№1, с. 61–74
16. Космическая платформа Навигатор
URL:<https://www.laspace.ru/ru/activities/products/platformy/bmss-navigadtor/>
17. В.Ю. Теребиж, Н.М. Стешенко, А.В. Долгополов, К.А. Антонюк, А.Н. Ростопчина-
Шаховская. Новая оптика 1.25-м телескопа АЗТ-11 Крымской астрофизической
обсерватории // Изв.Крымск. Астрофиз. Обсерв. 120, № 3, 22–31 (2024)
18. ИК-матрица высокого разрешения ИФП СО РАН
URL:<https://ktipm.isp.nsc.ru/product/ik-matricza-hd/>
19. Аббубекеров М.К., Тучин М.С., Стекольников О.Ю., Потанин С.А., Захарова А.И.,
Прохоров М.Е., Гостев Н.Ю. Малогабаритный звездный датчик АЗДК-2 // Системный
анализ, управление и навигация: Тезисы докладов. Сборник, 2023, с. 59-60.
20. Роскосмос. «Спектр-РГ» 100 дней полёта URL:<https://www.roscosmos.ru/26971/>
21. Сделано у нас. Роскосмос заказал восемь ракет «Союз-2.1а» за 10,4 млрд рублей.
URL:<https://sdelanounas.ru/blogs/120069/>
22. ТАСС. Роскосмос заказал три разгонных блока "Фрегат" для спутников Глонасс" и
"Гонец-М". URL:<https://tass.ru/kosmos/5835007?ysclid=m7269jrh7j422241008>

«ЗАГОР» – ЛЕГКОАТЛЕТИЧЕСКИЙ ИННОВАЦИОННЫЙ МЕТАТЕЛЬНЫЙ ДИСК

Рюмина Милана Николаевна

Москва г, лицей "Вторая школа" имени В.Ф.Овчинникова, 11 класс

Секция: Машиностроительные технологии

Научный руководитель: Сулягин Андрей Романович, методист кафедры физики, лицей "Вторая школа" имени В.Ф.Овчинникова

Аннотация

«Загор» – инновационный метательный диск.

Объектом разработки является легкоатлетический диск для метаний, с интегрированной системой сбора данных.

Цель работы – разработать и реализовать решение для облегчения тренировочного процесса и автоматизации проведения соревнований.

В процессе работы проводилось исследование проблемы с помощью анализа и сравнения, а также разработка, создание и тестирование диска.

Новизна проекта в интегрированной системе измерения расстояния и использовании поликарбоната для пластин.

Значимость проекта заключается в отсутствии аналогов изделия и возможности применения диска как инновационной технологии для проведения соревнований.

Основные конструктивные и технико-эксплуатационные показатели: достаточная эксплуатационная стойкость, практичность применения для пользователя.

Степень внедрения – изделие в процессе производства.

Областью применения является тренировочная и соревновательная деятельность спортсменов легкоатлетов.

Перспектива развития данного проекта состоит в разработке приложения для метателей, получении сертификата IAAF, расширении модельного ряда дисков для использования всеми категориями спортсменов и работе над распространением данной технологии на другие метательные снаряды

Сокращения

АКБ – аккумуляторная батарея

БЗУ – беспроводное зарядное устройство

ВФЛА – всероссийская федерация лёгкой атлетики

ВЦИОМ – всероссийский центр изучения общественного мнения

ГОСТ – государственный стандарт

ЗУ – зарядные устройства

КЗ – короткое замыкание

ПК – поликарбонат, полимер

ПММА – полиметилметакрилат, полимер

ПП – полипропилен, полимер

ПЭЭК – полиэфирэфиркетон, полимер

САПР – система автоматизированного проектирования

СОЖ – смазочно-охлаждающая жидкость

ТЗ – техническое задание

ТХ – технические характеристики

УП – управляющая программа

УФ – ультрафиолетовое излучение (солнечный свет)

ЦА – целевая аудитория

ЧПУ – числовое программное управление

ЧР – Чемпионат России по лёгкой атлетике

B2G (англ. Business-to-government) – бизнес-модель, в которой компания продаёт свои услуги государственным учреждениям.

IAAF (англ. International Amateur Athletics Federation) – международная ассоциация легкоатлетических федераций. Она же World Athletics.

IoT (англ. Internet of Things) – интернет вещей, то есть множество физических объектов, подключенных к интернету и обменивающихся данными.

MVP (Minimum Viable Product) — минимально жизнеспособный продукт, ранняя версия товара/сервиса с небольшим набором функций.

USB (англ. Universal Serial Bus – универсальная последовательная шина) –последовательный интерфейс для подключения периферийных устройств к вычислительной технике

Wi-charge (англ. Wireless charge) – беспроводное зарядное устройство

ВВЕДЕНИЕ

Лёгкая атлетика – один из старейших видов спорта. Её история начиналась с проведения в Древней Греции Олимпийских игр в 776 году до нашей эры. Программа соревнований древности была очень разнообразна: в неё входили беговые, прыжковые и метательные дисциплины, – и по сей день IAAF сохраняет часть традиционных видов. В то же время, чтобы в современном быстроразвивающемся мире спорт не выродился и продолжал быть интересен, она работает над внедрением новых технологий для анализа и автоматизации проведения соревнований.

Метание диска на дальность – один из сохранившихся изначальных видов лёгкой атлетики (рисунок 1), инноваций в котором IAAF пока что предложено не было [13]. Поэтому мой диск следует трендам в данном виде спорта и поможет развитию метания диска в России, поддержав спортсменов, тем более находящихся в изоляции от мирового спорта.



Рисунок 1 – Античная статуя «Дискобол»

Мой проект нацелен на решение основных проблем метателей – потеря инвентаря и невозможность быстро узнать результат попытки на тренировке. Таким образом, он позволит улучшить спортивные результаты за счет возможности быстро провести аналитику полёта диска и увереннее чувствовать себя на соревнованиях благодаря отсутствию потерь диска и необходимости привыкать к новому инвентарю.

Кроме того, изделие снижает травмоопасность работы судей, так как автоматически замеряет расстояние.

Изделие не имеет аналогов, так как на сегодняшний день любой диск представляет собой лишь тяжёлый корпус для метания. Загор – первый диск с встроенной электроникой, это не просто снаряд, а полноценное устройство для тренировок.

Описание цели проекта и постановка задач

Цель работы – разработать и реализовать решение для облегчения тренировочного процесса и улучшения спортивных результатов метателей диска, а также автоматизации проведения соревнований в этой дисциплине.

Работа над проектом была разделена на следующие задачи:

- Выявить проблему в тренировочном процессе метателей;
- Оценить актуальность и востребованность проектного изделия;
- Изучить высокотехнологичные разработки в лёгкой атлетике;
- Провести сравнительный анализ дисков разных производителей;
- Сделать выводы о необходимых конструкционных изменениях;
- Сформулировать ТЗ;
- Создать эскиз и модель изделия;
- Выбрать компоненты для создания продукта;
- Разработать технологию изготовления корпуса и электронных компонентов;
- Изготовить корпус и плату;
- Установить электронные компоненты в корпус диска;
- Разработать и реализовать мобильное приложение для передачи данных с диска;
- Провести апробирование и оценить точность работы устройства;
- Внедрить изделие в тренировочный процесс и получить отзыв прямого потребителя;
- Провести экономический и экологический анализ проекта;
- Определить вектор дальнейшего развития.

В данной пояснительной записке подробно описаны идея, цели, исследования, ход и результаты работ, а также дальнейший вектор развития проекта.

1 ОСНОВНАЯ ЧАСТЬ

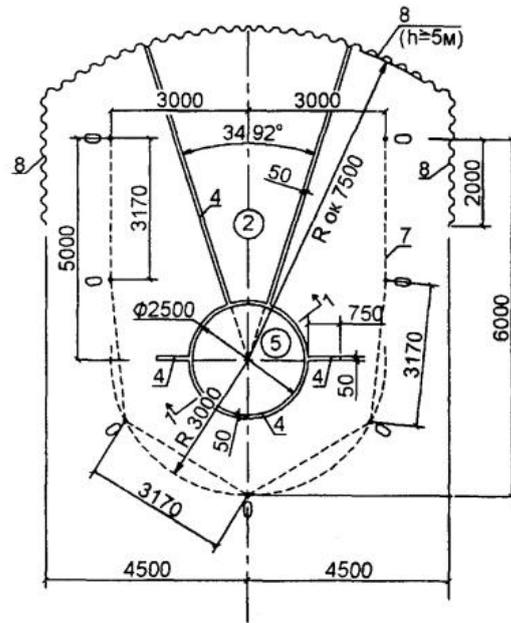
Постановка проблемы

«Сделай на один шаг больше соперника,
предусмотри на одну мелочь больше»

Метание диска – технический вид лёгкой атлетики, соревнования по которому заключаются в выполнении 6 попыток метнуть диск максимально далеко из специального сектора. На данный момент результаты чемпионов России составляют порядка 60м и 65м у женщин и мужчин соответственно.

Сектор состоит из круга диаметром 2.5м, U-образной высокой заградительной сетки и площади для приземления (рисунок 2). В соревновательные секторах ее роль обычно играет поле с газоном, расположенное в центре легкоатлетического стадиона. Его размеры на классическом летнем ядре составляют 68м на 105м, а рельеф регламентирован (рисунок 3). Это делается для удобства измерения результата попытки, его обычно производят рулеткой или лазерным дальномером. Таким образом, становится понятно, что устройство полноценного сектора требует больших пространств, что существенно усложняет реализацию как крытых площадок, так и секторов под открытым небом, ведь для этого требуется выравнивание ландшафта на большой площади [14].

При общении с легкоатлетами одной из спортивных школ выяснилось, что из-за недостатка места у них нет крытого сектора для метания снаряда, а по причинам, описанным выше, на улице они имеют лишь стартовый круг. Тренировки проходят в таких условиях практически в любую погоду, поэтому снаряды: и диски, и копья, и ядра приземляются в траву и снег и часто теряются. Учитывая длину светового дня в России (между 50°с.ш. и 70°с.ш. в зимнее время световой день составляет примерно 6-8 часов) и распорядок дня школьников, большая часть тренировок проходит в темное время суток, что еще больше осложняет поиск инвентаря.



ПО 1-1



Рисунок 2 – Чертеж сектора для метания диска

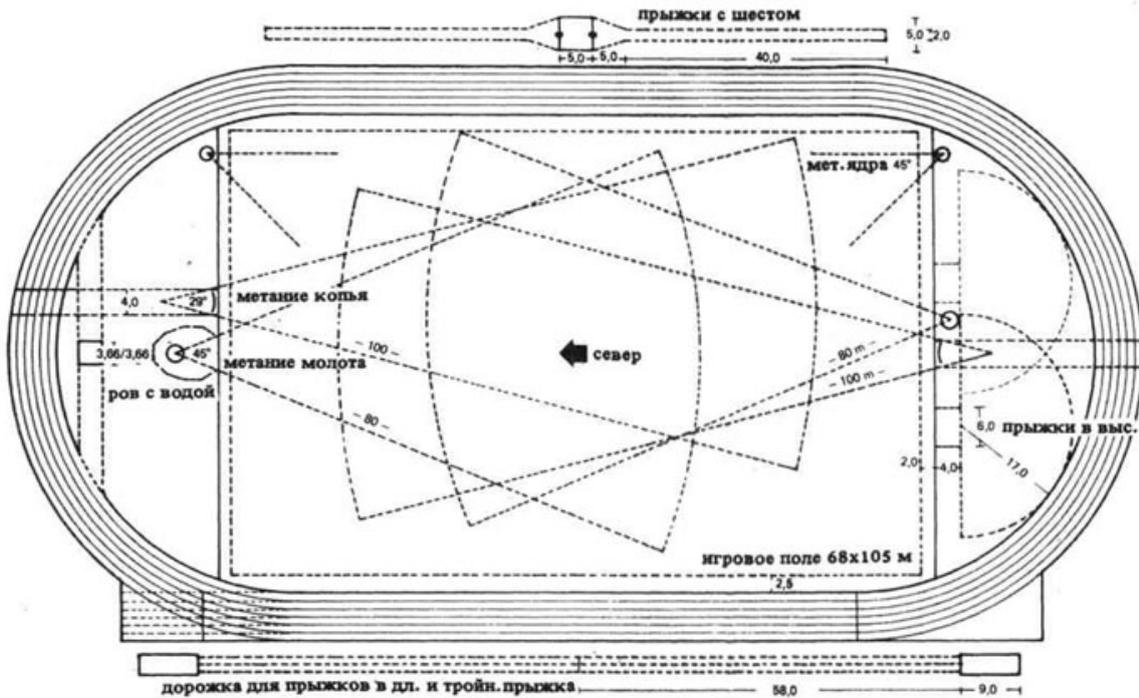


Рисунок 3 – Чертеж стадиона

У части спортсменов принято тренироваться и выступать на соревнованиях со своим

собственным диском. Для них недопуск личного снаряда на соревнования является стрессом, так как придётся привыкать к новому снаряду, что может привести к демонстрации худшего результата. Также и поломка/потеря диска может деморализовать спортсмена. Диск – дорогостоящий инвентарь, а не расходный материал, а потому все спортсмены хотят сохранить его на максимально возможный срок.

Кроме того, в отсутствие сектора для приземления снаряда из-за неровностей становится невозможно измерить дальность попытки на тренировке, а ведь этот показатель является ключевым в данном виде спорта.

Вариант решения проблемы

Решать рассмотренную проблему можно двумя способами: постройкой стадионов с секторами или совершенствованием инвентаря для метания. Я могу помочь спортсменам вторым путем, поэтому проведя опрос большого количества метателей, я выявила самые распространенные проблемы и убедилась в востребованности тех или иных функций изделия. После я сделала выводы о необходимости создания снаряда, удовлетворяющего следующим критериям:

1. Имеет систему измерения расстояния, не требующую ровной поверхности приземления;
2. Подключается к приложению для отображения местоположения диска на карте для поиска.

Согласно описанному в подпункте 1.3 опросу потенциальных пользователей (Приложение А), эти изменения в конструкции легкоатлетического диска удовлетворяют потребностям ЦА, а значит позволят спортсменам тренироваться с большим комфортом.

Актуальность проекта

Легкая атлетика – вид спорта, не требующий так много дорогостоящего оборудования как например, хоккей, баскетбол или водное поло. Поэтому секций этого вида спорта очень много. По данным опроса ВЦИОМ на 2023 год легкая атлетика – самый распространенный в России вид спорта, им занимаются 37% опрошенных [15]. Со слов замминистра спорта Алексея Морозова более 1 миллиона человек систематически занимаются лёгкой атлетикой, из них более 200000 человек юных спортсменов [16].

Большинство спортивных школ используют оставшуюся, как наследие СССР, инфраструктуру, а потому во многом устаревшую или даже требующую ремонта (согласно

данным того же А.Морозова в подготовке спортивного резерва участвует 1323 отделения, расположенные почти во всех субъектах РФ [16]). Бегуны, конечно, могут проводить часть сезона на необорудованных площадках, но метателям, в сравнении с ними, требуются более специализированные условия.

После выявления проблемы, мной был проведен опрос (Приложение А) среди 141 человека: метателей, их родителей и тренеров. Он показал, что у 44% нет полноценных секторов для метания, то есть данная проблема актуальна не только для моих знакомых.

Кроме того, изделие можно отнести к концепции IoT, с 2019 года набирающей популярность в WA. Для привлечения спонсоров и повышения зрелищности федерация постепенно внедряет новые технологии в некоторые виды лёгкой атлетики [13].

В 2024 году прыжках в длину с разбега было предложено измерять дальность попытки не от специальной планки до места приземления (рисунок 4), учитывая только попытки без заступов за эту планку, а от точки отрыва ступни. Это положение фиксируется камерами в области, большей чем планка, а поэтому в зачёт идут все попытки несмотря на заступы. Тестирование на коммерческих соревнованиях показало, что это позволяет сделать соревнования динамичнее и интереснее для зрителя. Поэтому на данный момент в WA идёт обсуждение изменения правил для проведения всех соревнований в таком формате.

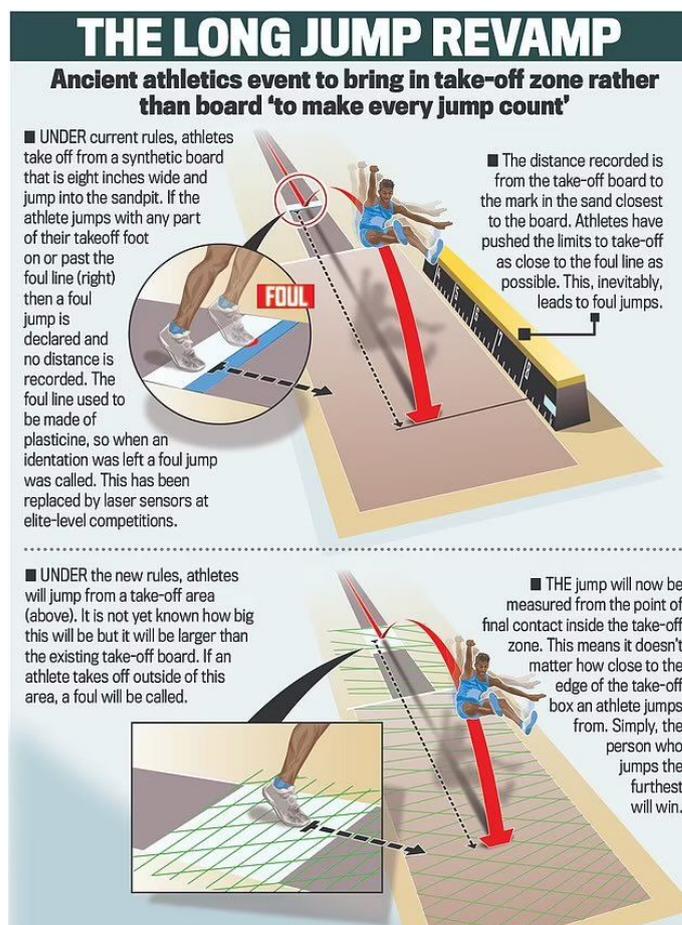


Рисунок 4 – Визуализация инновации в прыжках в длину

Для большей привлекательности в трансляциях соревнований высокого уровня строится инфографика ключевых показателей: удалённости от планки нижней точки тела спортсмена в прыжках с шестом и в высоту, скорости спортсмена по дистанции в спринте, траектория полёта снаряда в метаниях. Все эти показания получают за счёт обработки дорогостоящей системой изображений со многих камер [13]. Мой разработка позволит получать эти данные не используя внешнюю аппаратуру. Поэтому такой его функционал может быть полезен для использования на менее престижных соревнованиях, проводимых на хуже оборудованных стадионах.

На сегодняшний день Россия находится в изоляции от мирового спорта, а потому как никогда важно поддерживать спортсменов и развивать внутренний спорт. Внедрение нового будет полезно профессиональным спортсменам, сейчас сталкивающимися с застоем в результатах и потерей мотивации. Также более удобный инвентарь способен привлечь в спорт больше молодых атлетов, чаще других тренирующихся на неполноценных полях и сталкивающимися с потерями снарядов. Значит шансы вырастить чемпионов, достойно представляющих Россию на международной арене увеличиваются. Таким образом, продукт

поможет развитию метания диска в России

Кроме того, как было отмечено опрошенными, большая часть используемого сейчас оборудования зарубежного производства. Поэтому из-за санкций сейчас есть сложности с его приобретением. Создание отечественного диска, сопоставимого по качеству с зарубежными аналогами, отвечает задачам импортозамещения, активно реализуемым в современной России.

В результате реализации проекта получаем идеальный с точки зрения функционала продукт, который актуален в наше время

Круг потенциальных пользователей

Таким образом, целевую аудиторию продукта можно разделить на три сегмента:

- юные атлеты, которые чаще других тренируются на необорудованных для метания полях;
- спортивные школы, несущие денежные убытки из-за потери инвентаря;
- профессиональные спортсмены, которые сейчас находятся в изоляции от мирового спорта.

В реализации дисков следует говорить о модели B2G.

Во-первых, в России большинство спортивных школ государственные, поэтому чаще они, а не спортсмены занимаются закупкой инвентаря (подтверждается опросом из Приложения А). Готовый продукт заинтересует такие учреждения, имеющие проблемы с организацией тренировочного пространства либо столкнувшиеся с частыми потерями инвентаря, влекущими денежные убытки. Продукт также будет привлекателен для спортивных школ, из-за санкций столкнувшихся с сложностью заказа зарубежных дисков надлежащего качества.

Во-вторых, возможность использовать изделие на соревнованиях в качестве нового метода для снятия измерений делает продукт привлекательным для ВФЛА.

Итоговый продукт имеет себестоимость, сравнимую с дисками, уже представленными на рынке, а учитывая функциональное преимущество и импортозамещающую функцию, будет привлекателен в модели B2G.

Анализ существующих решений проблемы и вариантов идей

Так как идея проекта заключается в объединении спортивного снаряда и системы измерения расстояния, то перед проектированием собственного изделия я провела исследование дисков, имеющих в продаже, и систем, использующихся для измерения дальности попыток на соревнованиях. Оно помогло мне определить приемлемую стоимость изделия, а также выявить особенности строения и приоритетные для материала корпуса

характеристики.

Рассмотрим легкоатлетические диски массой 1 кг:

- Polanik CCD20-1 и Polanik CCD14-1 как передовые модели, при этом последний создан «для самых требовательных метателей»;
- Polanik CPD11-1 и Polanik HPD11-1 как самые распространенные соревновательные модели;
- Glav как наиболее доступный диск.

В силу специфики продукта, на рынке представлено не очень много разновидностей дисков, поэтому я отобрала 5 соревновательных моделей из разных категорий, позволяющих узнать нужную мне информацию, и провела сравнительный анализ продуктов (Приложение Б).

Таким образом, прямых аналогов моему изделию на сегодняшний день не существует, поэтому в качестве конкурентов рассмотрим системы измерения расстояний, используемые для проведения соревнований.

На международной арене и крупнейших российских соревнованиях используются (Рисунок 5):

- лазерные дальнометры, переносимые вручную судьями на колышке, прикладываемом к месту касания снарядом земли;
- системы камер, в качестве дополнительного контроля.

На менее крупных соревнованиях в России (Рисунок 6) методы еще более простые: один судья следит за наличием заступа и после попытки сигнализирует об этом двум другим, они в свою очередь, держат измерительную ленту и прикладывают её концы центру метательного круга и точке первого касания диском земли (таким образом, измерение происходит от внутреннего края окружности метательного круга до точки пробития поля диском). Для быстроты передачи данных они кричат результат и его выводят на табло. Во-первых, для реализации такого метода один из судей должен находиться в поле, то есть прямо в секторе для приземления снаряда, что подвергает его жизнь опасности. Во-вторых, требуется судейская бригада минимум из 3 человек, моя автоматизированная система позволяет задействовать всего одного [14].

Таким образом, диск, измеряющий расстояние без использования дополнительной аппаратуры и не нуждающийся в ровном рельефе – это отличное решение для спортивных школ, не имеющих возможности организовать полноценный сектор. Используя современные технологии, сегодня можно создать не имеющий аналогов многофункциональный продукт,

содействующий развитию спорта и повышающий его зрелищность.

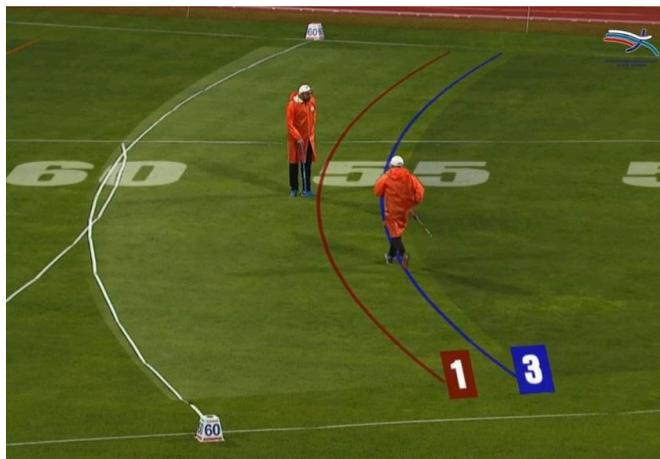


Рисунок 5 – Судейство с дальномером на ЧР



Рисунок 6– Судейство с лентой на первенстве города

Преимущества предлагаемого решения

Существующие продукты имеют право на жизнь, но моё изделие имеет следующие преимущества:

- имеет систему измерения расстояния, не требующую ровной поверхности приземления и системы камер;
- подключается к приложению для отображения местоположения диска и характеризующих полёт данных, полученных с датчиков;
- изготовлен из современных прочных материалов;
- безопасен для судей;
- имеет подсветку для тренировок в темное время суток.

Обоснование выбора используемых ресурсов, компонентов и технологий

1.1.1 Составление технического задания

Изучив рынок дисков, я пришла к выводу, что изделий, подобных моему, не существует. Кроме того, система не должна повлиять на весовые и размерные характеристики (Рисунок 7), чтобы соответствовать правилам соревнований [14, 17], поэтому было принято решение самостоятельно разработать конструкцию и произвести изделие.

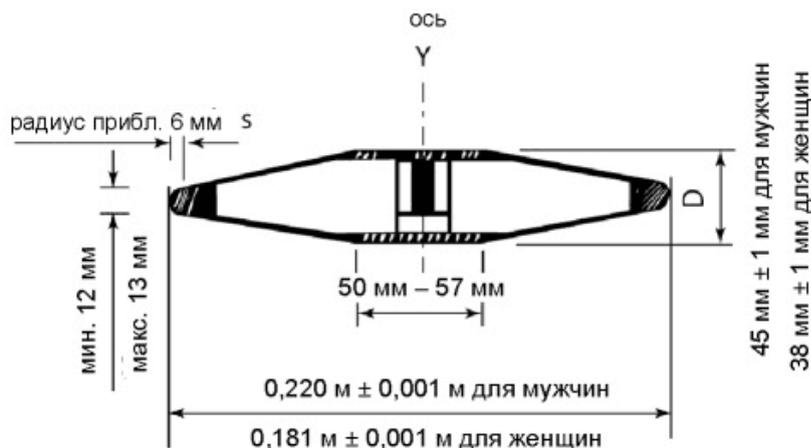


Рисунок 7 – технический рисунок диска из официальных правил

Перед формулировкой ТЗ были выделены ключевые особенности изготавливаемого устройства:

1. недопустимость погрешностей массы (отклонение соревновательных дисков массы 1 кг: $-0,005\text{кг}$, $+0,025\text{кг}$);
2. симметричность конструкции, так как центра масс должен попадать на ось диска дальность, ведь от попадания на ветер и распределения сил с точки зрения физики зависит дальность полёта полёта;
3. необходим демпфер для целостности платы под ударными нагрузками;
4. для работы подсветки корпус должен быть из прозрачного материала.

Таким образом, в идеале мне требуется создать изделие, максимально приближенное к представленным на рынке, но имеющее электронные компоненты внутри

Выбор материала

Выбор материала был осуществлен после анализа материалов дисков, представленных на рынке.

Для реализации для боковых пластин важно было подобрать прочный, прозрачный, устойчивый к минусовым температурам и по возможности легкообрабатываемый материал. Я провела сравнительный анализ прозрачных полимеров с пластиком, фанерой и углепластиком, которые чаще всего используются для производства боковых платин дисков (Приложение В) и выбрала поликарбонат, удовлетворяющий всем моим требованиям. Его единственным минусом поликарбоната является восприимчивость к УФ. Поэтому для защиты материала от

негативного воздействия солнечного света я использую лак Lens Clear.

Из-за строгих требований к массе изделия я рассчитала, что плотность металла для изготовления должна составлять менее $7,5 \text{ г/см}^3$. В то же время ударная вязкость должна быть не менее 20 Дж/см^3 . Единственным подходящим материалом оказался титан. Это дорогостоящий и очень сложный в обработке материал. Он требует резцов из сплавов высокой твёрдости, а также особых вязких охлаждающих жидкостей. Поэтому работать с ним нужно только на высококлассном оборудовании, и детали из него были отданы на производство. Из-за опасности возгорания от нагревания при обработке, обработкой занимался специалист имеющий достаточный уровень знаний.

1.1.2 Проектирование

Для проектирования я выбрала программу КОМПАС-3D [18], так как её функционал подходит для создания модели каркаса и я имею опыт работы в ней.

По аналогии с существующими дисками изделие должно быть разборным и состоять из двух боковых пластин, двух центральных, обода и стягивающего механизма. Я приобрела диск Nordic для снятия всех размеров и изучения строения.

Для системы измерения расстояния и определения местоположения в конструкцию добавляются печатная плата, АКБ, катушки Qi-charge, поэтому в модель добавлены и они.

Для удобства установки платы предусмотрен пазовый механизм с зазорами для меньшего воздействия на неё ударных нагрузок.

Я учла все особенности конструкции и создала модели всех деталей Таким образом я получила модель изделия (рис. 8) и чертежи для изготовления корпуса и контура платы (Приложения Г, Д)

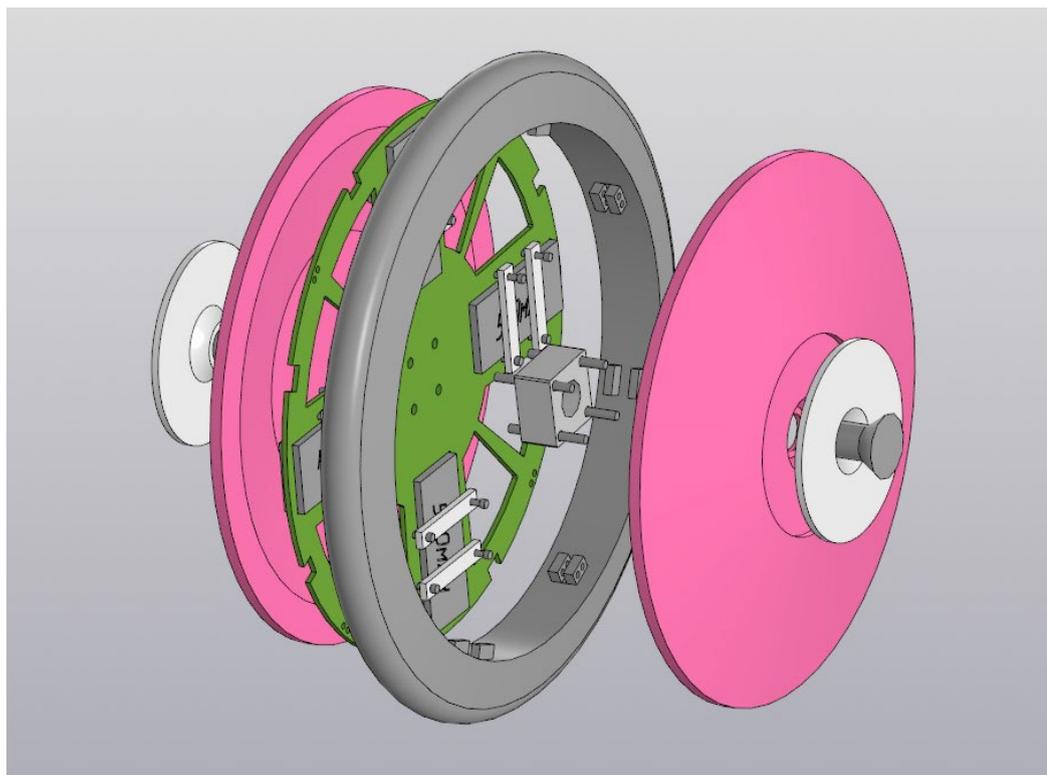


Рисунок 8 – взрыв-схема диска

1.1.3 Плата

Основной частью системы является плата. Проектирование проводилось в САПР KiCAD [19] (Приложение Е).

Для высокого качества и большей надежности производство было заказано на производстве [20], а не проведено самостоятельно. Кроме того, без дорогостоящего оборудования можно изготовить максимум двухслойную плату, а для качественного определения местоположения, определения координат и передачи этих данных на смартфон требуется минимизация помех работы датчиков, которая возможна лишь при добавлении слоев экранирования. Поэтому требовалась многослойная печатная плата.

Важнейшей частью любой платы является микроконтроллер, поэтому прежде всего нужно выбрать его. Микроконтроллеры STM делятся на 5 видов:

- STMF – стандартные
- STML – энергосберегающие
- STMG – массовые
- STMB – со встроенным Bluetooth и Wi-Fi
- STMH – высокопроизводительные

Из-за необходимости минимизации веса дополнительных компонентов я выбрала STM32WB55, так как он имеет низкое энергопотребление, а значит потребует меньше АКБ, и встроенный Bluetooth (в отдельности модули довольно громоздки).

С учетом погодных условий диск с электроникой внутри должен быть герметичен, поэтому используется беспроводная зарядка, энергия будет храниться на четырёх (для симметричности) литий-ионных аккумуляторах ёмкостью 500 мАч. Для симметрии беспроводных зарядок в конструкции две, и обе они расположены под центральными пластинами.

так как приёмник беспроводной зарядки требует ровной поверхности для лучшего функционирования. Принцип работы беспроводной зарядки основан на эффекте электромагнитной индукции, поэтому центральные пластины не могут быть изготовлены из металла, как это происходит у стандартных дисков. Поэтому они изготовлены из поликарбоната, как и боковые детали.

Передачик Qi-charge встроен в кейс для транспортировки диска. Для простоты эта часть напечатана на 3Д-принтере из ABS пластика. Кейс позволит дольше сохранить диск и зарядить датчики перед тренировкой. Расчетное время работы – 40 часов.

Для определения координаты в каждый момент времени имеется 9-осевой акселерометр и гироскоп BMX055. Он был выбран как наиболее точный согласно документации. Этот датчик позволит производить расчёты для измерения расстояния.

Для определения момента, когда спортсмен отпускает диск, а значит измерения должны начаться требуется сборка светодиодов и фототранзисторов, которые смогут фиксировать снижение светового потока из-за пропажи отражающей ладони спортсмена.

Для определения местонахождения поставлен GPS-модуль на базе ATGM336H.

Для подсветки диска при тренировках в тёмное время суток установлены неопиксели в большом количестве.

Также на плате установлена микросхема, представляющая из себя сборку диодов и отводящая статическое электричество от платы микроконтроллера.

Остальные компоненты являются вспомогательными: преобразователи напряжения, коннектор для загрузки программы, резисторы, конденсаторы.

1.1.4 Система

Мобильное приложение ребуется для отслеживания дисков для метания, которое сейчас находится в бета-версии. Оно позволяет спортсменам и тренерам регистрировать местоположение, траекторию и состояние дисков с помощью GPS и дополнительных датчиков.

Приложение написано на Swift для iOS и Kotlin для Android, а серверная часть реализована на Node.js с базой данных PostgreSQL. В бете уже доступны основные функции, а впереди – доработка алгоритмов отслеживания и интеграция с аналитическими инструментами.

Этапы работ

1.1.5 Выполненные процессы по изготовлению корпуса

- расчёт массы всех компонентов;
- 3D-модель в Компас 3D [18];
- Получение чертежей в Компас 3D;
- фрезеровка на станке с ЧПУ поликарбонатных деталей;
- фрезеровка металлического обода на станке;
- покрытие поликарбоната лаком с УФ-стабилизатором.

1.1.6 Выполненные процессы по изготовлению платы

- 3D-модель контура в Компас 3D [18];
- Подбор всех датчиков и вспомогательных компонентов к ним;
- Создание схемы и трассировка в KiCad [19]
- Изготовление на производстве
- Монтаж компонентов к проводящему рисунку при помощи пайки
- Тестирование

1.1.7 Выполненные процессы по изготовлению кейса

- 3D-модель в Компас 3D
- -получен g-код
- печать на Creality SE
- постобработка от заусенцев

1.1.8 Сборка

- Загрузка на плату программы
- Крепление АКБ к плате
- Закрепление платы в ободу
- Сборка деталей и закрепление винтами и резьбовыми шпильками
- Нанесение герметика на стыки диска для предотвращения попадания воды

2 ЗАКЛЮЧЕНИЕ

Временные и денежные затраты

Для оценки вложений на изготовление прототипа устройства и MVP составлены таблицы 3 и 4.

Таблица 1 - Стоимость затрат на разработку.

Компонент	Стоимость	Кол-во	Итоговая стоимость
STM32WB55CEU6	1966	1	1966
Акселерометр	3030	1	3030
Зарядка BQ24617	575	1	575
Преобразователь в 3.3В	300	1	300
GPS	1905	1	1905
Аккумулятор	600	4	2400
Транзистор PNP	4	1	4
Неопиксели	27	20	540
Защита USB	200	1	200
USB 3.1		1	1
Вспомогательные компоненты			
Конденсатор 12pF	7	3	21
Конденсатор 4.7uF	13	3	39
Конденсатор 100nF	5	10	50
Конденсатор 1uF	1	3	3
Конденсатор 100pF	7	2	14
Резистор 220Ом	6	8	48
Резистор 100Ом	6	4	24
Резистор 47Ом	6	2	12
Резистор 12Ом	6	4	24
Резистор 10кОм	6	2	12
Резистор 0Ом	6	2	12
Кварц 16MHz	19	2	38
Катушка 2.7нГ	55	1	55
Катушка 47нГ	35	1	35
Плата основная	6000	1	6000
Корпус		0	
Беспроводная зарядка	169	1	169
Обод металлический	4000	1	4000

Покрышка поликарбонат	1500	2	3000
Лак	800	1	800
Крепежи	1000	-	1000
Пластик	500	-	500
ИТОГО	23844		

Таблица 2 - Затраты рабочих часов

Рабочие часы	Количество
Разработка	300 часов
Работа в мастерской	200 часов
Тесты	8 часов
Разработка ПО	20 часов
Итого:	528 часов

Таким образом, временные затраты составили 528 рабочих часов, стоимость компонентов и материалов при покупке в розницу 23 844 рублей.

Экологическая оценка

Я забочусь о состоянии окружающей среды, поэтому задумываюсь об утилизации материалов и применении их в альтернативных методах.

В то же время, важно помнить, что плату и ее компоненты, без которых невозможна реализация моего проекта, нужно утилизировать в специальных местах.

Результаты работы

2.1.1 Описание достигнутого результата

На данный момент имеется прототип корпуса и платы, а также все необходимые для производства изделия чертежи и расчеты. Этапы работ по месяцам представлены в Приложении .

Таким образом, можно говорить о успешном выполнении первой части целей и задач проекта.

2.1.2 Решение задач

- Проблема была изучена с помощью литературных и других источников;
- Был проведен опрос целевой аудитории;
- Была разработана конструкция устройства для решения поставленной проблемы, создана 3D-модель;
- Произведены расчеты массы всех компонентов;
- Выбраны материалы, компоненты и технологии изготовления;
- Произведен корпус;
- Плата создана и изготовлена;

2.1.3 Все электронные компоненты установлены Перспективы и развитие

Прямо сейчас плата находится в производстве и идет работа по разработке бета-версии приложения, являющегося немаловажной частью системы.

Перспектива развития данного проекта состоит в:

- разработке полноценного приложения для метателей, ведь цифровизация тренировок спортсменов и анализ их результатов - будущее элитного спорта
- расширении модельного ряда дисков для использования всеми категориями спортсменов
- распространении данной технологии на другие метательные снаряды.

В будущем, после получения сертификата IAAF, возможен запуск коммерческих продаж.

В организации проведения соревнований по метанию диска в отличие от других видов лёгкой атлетики ничего инновационного пока предложено не было [13], поэтому мой продукт, следующий тенденциям развития мирового спорта, будет еще одним примером применения современных технологий в спортивной отрасли. В качестве перспективы, сертифицированный Загор может быть востребован для проведения соревнований всероссийского или международного уровня.

Итоги

В результате проведенных работ я научилась работать с новым оборудованием и различными материалами. Приобрела ряд новых компетенций и создала прототип устройства.

Я детально изучила проблему своего проекта, грамотно подобрала компоненты и

материалы. Я не останавливаюсь на достигнутом и продолжаю работу по проекту в соответствии с обозначенным вектором развития.

Диск облегчит проведение тренировок, снизит денежные убытки, повысит доступность инвентаря, а также позволит увеличить зрелищность соревнований и снизить травмоопасность работы судей.

Изделие обладает большой социальной значимостью, так как способно стать дополнительной мотивацией как для начинающих, так и для профессиональных атлетов. Кроме того, диск позволит меньше нервничать из-за потерь инвентаря и необходимости привыкать к новому диску, а значит увереннее чувствовать себя на соревнованиях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Литература:

1. Глозман Е.С. Технология 7-8-9 класс: Учебник / Хотунцев [и др.]. – ФГОС, Просвещение, Дрофа, 2020.
2. Арзамасов Б. Н. Материаловедение: учебник для технических вузов / Б. Н. Арзамасов [и др.]. – Москва: МГТУ имени Н. Э. Баумана, 2001.
3. Якушев А. И. Допуски, посадки и линейные измерения: справочник – Москва: Машиностроение, 1970.
4. ГОСТ 2.747-68. Стандартизация в области электроники. Основные положения. – Москва: Издательство стандартов, 1968.
5. Соколов, П. И. Электронные печатные платы. – Москва: Энергоатомиздат, 2009.
6. ГОСТ 23752-79. Печатные платы. Общие технические условия. – Москва: Издательство стандартов, 1979.
7. Ширяев, В. А. Технология производства электронных плат. – Санкт-Петербург: Питер, 2015.
8. ГОСТ 23751-79. Печатные платы. Термины и определения. – Москва: Издательство стандартов, 1979.
9. Разумовский, В. Ф. Проектирование и производство многоканальных электронных плат. – Москва: Радио и связь, 2010.
10. ГОСТ 16368-2014. Элементы печатные. Основные нормы и требования. – Москва: Издательство стандартов, 2014.

Интернет-ресурсы:

11. ЕСКД – URL: http://www.electroeffect.ru/docs/databook/data_info/2.3012.321.pdf
12. ЕСТД – URL: <https://docs.cntd.ru/document/1200086391>
13. Использование новых технологий WA – URL: <https://sb-by.turbopages.org/turbo/sb.by/s/articles/tekhnika-dlya-bega.html>; URL: <https://cyberleninka.ru/article/n/vnedrenie-tehnologiy-v-olimpiadu-2024-novye-gorizonty-sportivnyh-sorevnovaniy>; URL: <https://worldathletics.org/news/news/high-tech-helps-judges-athletes-and-spectator>;

URL: [How Wavelight technology has opened up new possibilities in athletics | FEATURE | World Athletics](#);

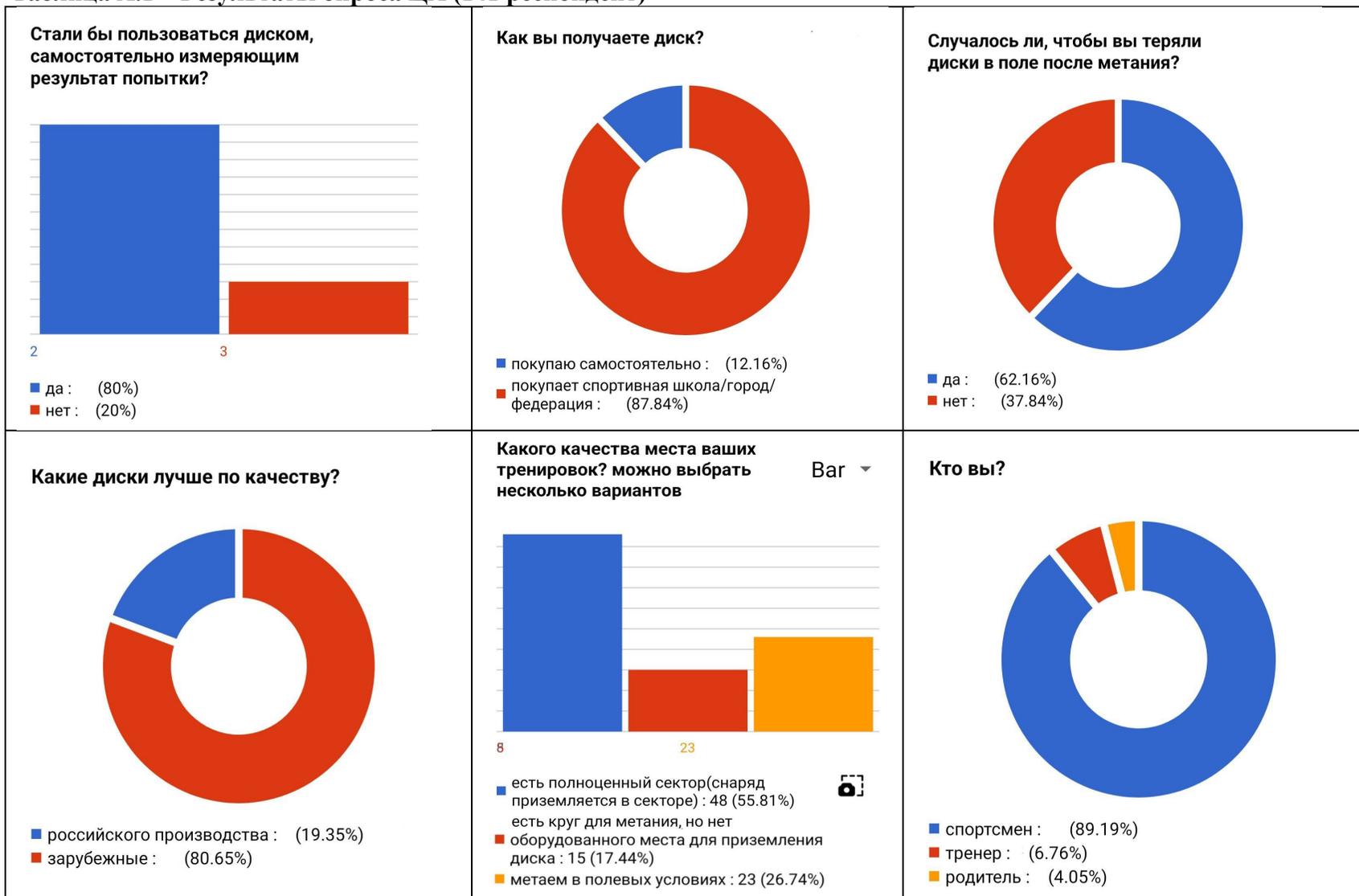
URL: [New competition formats trialled in Fribourg | REPORT | World Athletics](#);

URL: [EXPERIMENTS IN TRACK & FIELD](#)

14. Правила соревнований в метании диска – https://sudact.ru/law/pravila-vida-sporta-legkaia-atletika-utv-prikazom_1/pravila/glava-iv/c/
15. Статистика о занятиях спортом от ВЦИОМ – URL: <https://wciom.ru/analytical-reviews/analiticheskii-obzor/sport-dlja-vsekh>
16. Выставка в Госдуме, посвященная 135-летию российской легкой атлетики – URL: <https://clck.ru/3NBbPw>
17. Требования к соревновательному диску – https://sudact.ru/law/pravila-vida-sporta-legkaia-atletika-utv-prikazom_2/pravila/tekhnicheskie-pravila-sudeistva/chast-iii-tekhnicheskie-vidy/metaniia/34_1/disk/https://sudact.ru/law/pravila-vida-sporta-legkaia-atletika-utv-prikazom_2/pravila/tekhnicheskie-pravila-sudeistva/chast-iii-tekhnicheskie-vidy/metaniia/34_1/disk/
18. Аскон, САПР КОМПАС-3D – URL: <https://kompas.ru/>
19. KiCad IDE – URL: <https://www.kicad.org/>
20. Rezonit – URL: <https://www.rezonit.ru/>
21. Arduino – URL: <https://www.arduino.cc/>
22. Microsin – URL: <http://microsin.net/>
23. Амперка вики – URL: <http://wiki.amperka.ru/>
24. Arduino Master – URL: <https://arduinomaster.ru/>
25. Федеральный портал проектов нормативных правовых актов – URL: <https://regulation.gov.ru>
26. Хостинг IT проектов – <https://github.com>

ПРИЛОЖЕНИЕ А

Таблица А.1 – Результаты опроса ЦА (141 респондент)



ПРИЛОЖЕНИЕ Б

Таблица Б.1 – Сравнительный анализ легкоатлетических дисков и моего изделия

	Загор-1	Polanik CCD20-1	Polanik CCD14-1	Polanik CPD11-1	Polanik HPD11-1	Glav
Материал обода	титан	оцинкования сталь	Латунь с стальным сердечником	коррозионностойкий сплав	коррозионностойкий сплав	оцинкования сталь
Материал центральных пластин	титан	нержавеющая сталь	-	коррозионностойкий сплав	коррозионностойкий сплав	оцинкования сталь
Материал боковых пластин	поликарбонат	углеродное волокно	углеродное волокно	полипропилен	клееная фанера	клееная фанера
Наличие подсветки	+	-	-	-	-	-
Способность измерять дальность попытки	+	-	-	-	-	-
Возможность следить за характеристиками полёта диска со смартфона	+	-	-	-	-	-
Построение персональной зависимости результатов	+	-	-	-	-	-
Стоимость за шт в розницу, руб	30000	29630	25400	11250	12450	3570
Страна производства	Россия	Польша	Польша	Польша	Польша	Китай

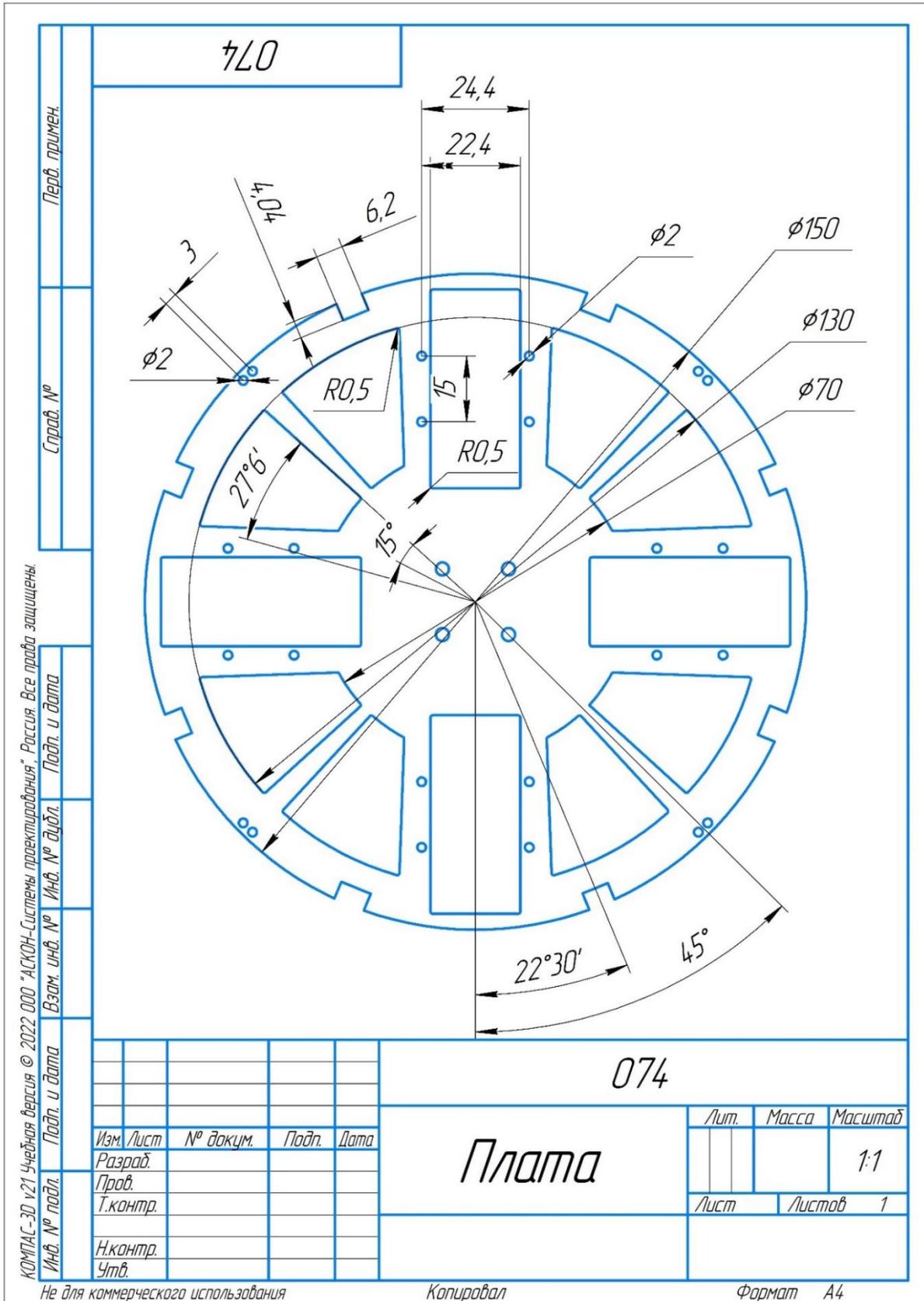
Сертификат IAAF	пока нет	+	+	+	+	-
-----------------	----------	---	---	---	---	---

ПРИЛОЖЕНИЕ В

Таблица В.1 – Сравнительная характеристика материалов для боковых пластин

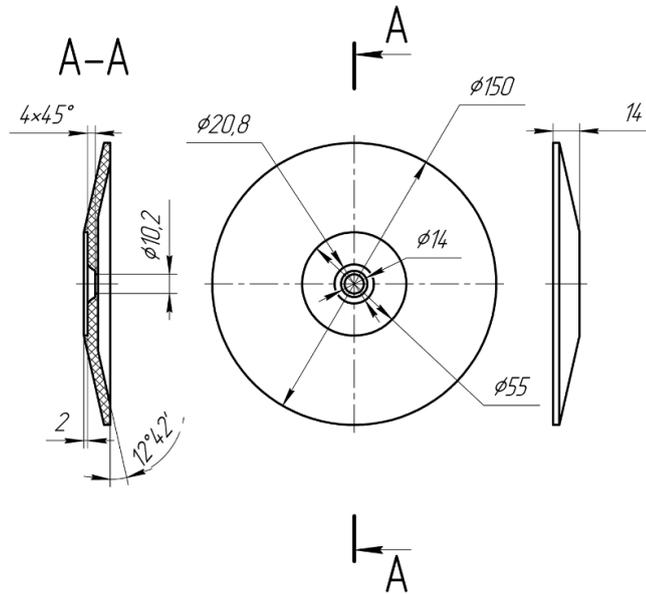
Характеристика	ПК	ПММА	ПЭЭК	Углепластик	ПП	Фанера
Прозрачность	До 88%	До 92%	низкая	Непрозрачный	низкая	Непрозрачный
Плотность, г/см ³	1,2	1,2	1,3	1,5-1,8	0,85-0,95	0,6-0,75
Прочность на разрыв, МПа	60-70	50-70	100-130	400-1500	30-40	30-60
Прочность на изгиб, МПа	90-100	80-100	150-180	400-1200	40-60	25-50
Ударная вязкость, Дж/см ³	60-80	15-20	8-12	До 90	5-10	35
Обрабатываемость	хорошая	высокая	хорошая	низкая	хороша	хороша
Стоимость	средняя	низкая	высокая	высокая	высокая	низкая

ПРИЛОЖЕНИЕ Г
Чертеж платы



ПРИЛОЖЕНИЕ Д
Чертёж и технологическая карта боковой пластины

Технологическая карта изготовления
пластины

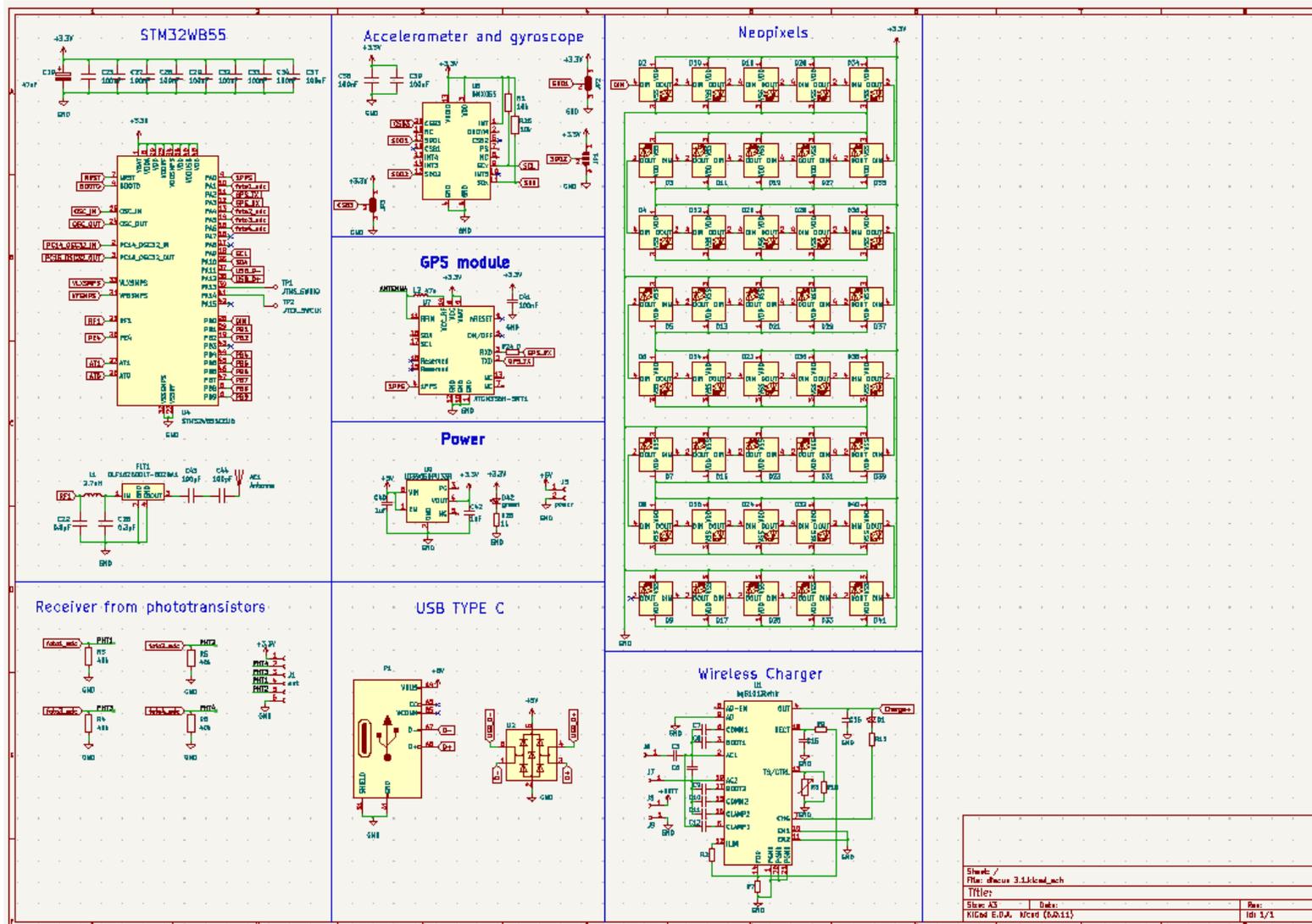


№	Технологическая операция
1	Подготовка УП для станка с ЧПУ
2	Подбор и подготовка режущего инструмента
3	Подготовка приспособления для крепления
4	Проверка работоспособности станка
5	Установка и выверка инструмента
6	Закрепление приспособления на столе станка
7	Закрепление заготовки в приспособлении
8	Обнуление координат
9	Предварительная обработка (черновая)
10	Получистовая обработка
11	Чистовая обработка
12	Снятие фасок
13	Сверление отверстий
14	Извлечение детали из приспособления
15	Удаление заусенцев
16	Очистка детали от стружки и СОЖ

Чертил	Рюмина М. Н.	13.01.24	Пластина
Проверил	Сутягин А. Р.	13.01.24	
Лицей "Вторая школа"			ПКБ

ПРИЛОЖЕНИЕ Е

Электрическая схема платы



ПРИЛОЖЕНИЕ Ж

Таблица Ж.1 – План работы над проектом

	Август	Сентябрь	Октябрь	Ноябрь	Декабрь	Январь
Появление идеи, изучение проблемы.						
Нахождение возможных способов решения поставленной проблемы						
Анализ систем измерения расстояния и рынка дисков						
Проведение опроса ЦА						
Создание 3D модели, прототипирование из пластика и фанеры						
Подбор материалов для корпуса						
Подбор компонентов для платы						
Создание схемы и производство платы						
Написание документации по проекту.						

УСТРОЙСТВО КОНТРОЛЯ ИНFUЗИИ

Скворцов Александр Сергеевич

Москва г, ГБОУ "Школа "Свиблово", 9 класс

Секция: Радиоэлектроника, лазерная и медицинская техника

Научный руководитель: Змиевской Григорий Николаевич, к.н., доцент каф. БМТ1

МГТУ им. Н.Э. Баумана

Аннотация

Непрерывность и скорость введения раствора при инфузионной терапии определяют эффективность лечения. Из-за высокой стоимости инфузионных насосов и широкого применения процедур инфузионной терапии в большинстве случаев на практике скорость введения раствора определяют визуально по частоте падения капель в камере инфузионной системы.

Целью данной работы является создание портативного недорогого устройства контроля инфузии для массового применения.

Для автоматического контроля скорости инфузии в данной работе предлагается использовать метод оптического зондирования капельной камеры.

Луч света распространяется от источника и принимается фотодатчиком на противоположной стороне камеры. Изменение сигнала фотодатчика происходит из-за поглощения света в капле, его рассеяния и отражения от поверхности капли.

Разработанная модель распространения света в капле и проведенное численное моделирование позволили определить параметры устройства.

В результате проведенных экспериментов продемонстрирована возможность контроля скорости инфузии путем оптического зондирования и скорректированы параметры устройства контроля инфузии.

Разработаны компоненты аппаратно-программных средств устройства контроля инфузии

Введение

Инфузией, или вливанием, называют введение в организм растворов лекарственных веществ, коллоидных, питательных растворов, крови и кровезаменителей, не используя пищеварительный тракт. Инфузия бывает подкожной, внутрикостной, внутрисуставной, внутрибрюшинной, ректальной, артериальной, но чаще всего используют внутривенную инфузию [1,2].

Инфузионная терапия проводится с целью восполнения потери жидкости, восстановления водно-солевого и электролитного баланса и введения в организм питательных веществ и лекарственных растворов.

Для проведения инфузии обычно вену прокалывают иглой (венопункцию). Если жидкость нужно ввести экстренно, то возможна струйная инфузия. В большинстве случаев инфузию проводят капельно. Скорость введения лекарств рассчитывают исходя из медицинских показаний в единицах массы на килограмм массы тела пациента. Затем, зная концентрацию лекарственного вещества в растворе, скорость введения раствора рассчитывают в мл/мин и в каплях в минуту.

От постоянства и скорости введения раствора лекарственного вещества зависит не только качество лечения, но и в ряде случаев жизнь пациента. На скорость введения влияет положение иглы, наличие пузырьков воздуха (воздушная эмболия), изгиб трубки для подачи раствора, количество оставшегося раствора в емкости и др. В ряде случаев требуется своевременная замена емкостей с растворами.

По данным Всемирной организации здравоохранения неисправность инфузионной системы приводит к ухудшению состояния здоровья почти каждого второго пациента отделений интенсивной терапии и реанимации, включая летальный исход [3].

Актуальность задачи поддержания заданной скорости инфузии в течение длительного времени следует из числа статей, посвященных этой теме [4]. На рисунке 1 показан рост количества статей в ведущих зарубежных журналах, посвященных точности и надежности инфузионных систем.

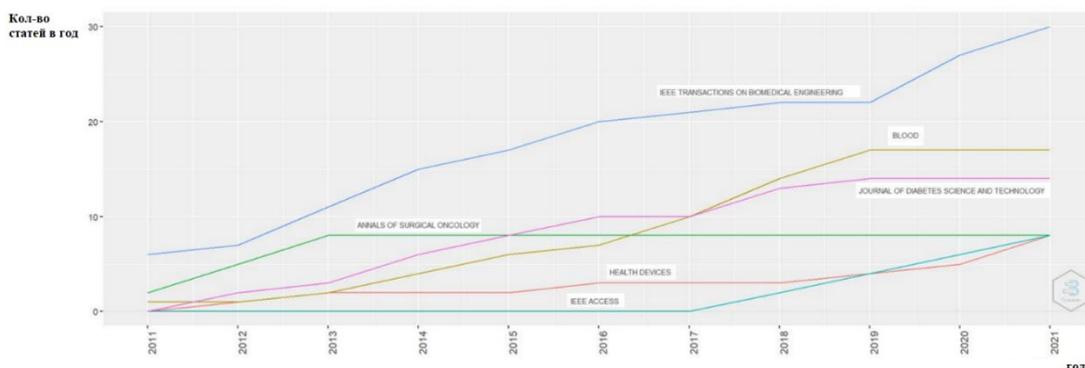


Рисунок 1 – Количество статей, посвященных проблеме контроля инфузии [4]

Обычно скорость инфузии не превышает 600-800 мл/час. Скорость введения раствора натрия хлорида (0,9%) обычно задают не более 180 капель/мин. Его применяют при состояниях обезвоживания и для разведения лекарственных веществ. Скорость введения раствора Рингера (содержит ионы натрия, калия, кальция и хлора) не превышает 70-80 капель/мин. Его применяют при потерях воды из желудочно-кишечного тракта или с мочой, а также для разведения лекарственных веществ.

Целью данной работы является создание портативного недорогого устройства контроля инфузии для массового применения.

В отличие от существующих прототипов, описанных, например, в [5], предполагается создание легкого датчика в виде прищепки, не вызывающего выпадение иглы из емкости с раствором. Устройство должно обеспечивать надежную передачу данных по радиоканалу и не требовать применения ретрансляторов радиосигнала в пределах больничного отделения. Для разработки такого датчика необходимо разработать модель рассеяния излучения в капле и выбрать параметры излучателя и приемника света.

1. Методы контроля инфузии

Для контроля инфузии применяются оптические, электрические (емкостные), вибрационные, гидравлические, температурные методы.

2.1. Оптические методы

Оптические методы основаны на регистрации оптического излучения, проходящего через капельную камеру инфузионной системы или через сосуд с лекарственным веществом. Оптическое излучение, проходя через каплю, преломляется и рассеивается в растворе лекарственного вещества, вызывая измерение сигнала фотодатчика. В одном из вариантов применения такого метода [5] используется два инфракрасных датчика и один фотоприемник, работающие в инфракрасном диапазоне (рисунок 2). Передача информации о скорости инфузии происходит по беспроводному каналу WiFi.

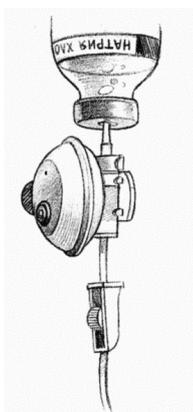


Рисунок 2 – Вариант датчика оптического контроля инфузии [5]

2.2. Емкостной метод

Емкостной метод основан на изменении емкости конденсатора при протекании жидкости или при наличии воздуха между его обкладками [6]. Данный метод позволяет определять появление пузырьков в инфузионной системе (рисунок 3).

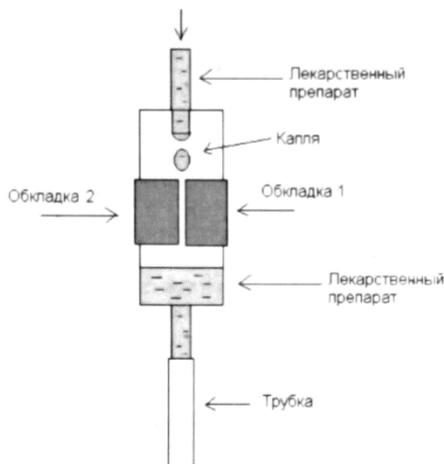


Рисунок 3 – Емкостной датчик контроля инфузии [6]

2.3. Вибрационный метод

Вибрационный метод основан на регистрации колебаний на поверхности жидкости при падении капли [7] общий вид датчика показан на рисунке 4

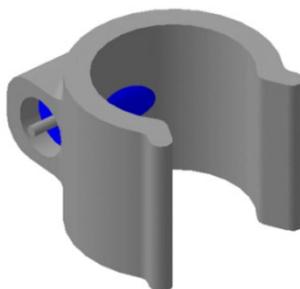


Рисунок 4 – Вибрационный датчик контроля инфузии [7]

2.4. Гидравлические методы

Гидравлические методы основаны на контроле давления, оказываемом поршнем на жидкость в инфузионной системе, или на контроле мембраны мембранных насосов [4].

2.5. Температурный метод

Температурный метод основан на измерении скорости тока жидкости по изменению температуры при прохождении через участок, где она подвергается нагреву [8]. При увеличении скорости жидкость не успевает нагреться, и разность температур до и после нагреваемого участка уменьшается.

Несмотря на существование указанных методов контроля инфузии [4-8], широкое применение нашли только инфузионные насосы с гидравлическим контролем. На практике из-за высокой стоимости инфузионных насосов и широкого применения процедур

инфузионной терапии скорость введения раствора определяют визуально по частоте падения капель в капельной камере инфузионной системы.

Целью данной работы является создание портативного недорогого устройства контроля инфузии для массового применения.

В отличие от существующих прототипов, описанных, например, в [5], предполагается создание легкого датчика в виде прищепки, не вызывающего выпадение иглы из емкости с раствором. Устройство должно обеспечивать надежную передачу данных по радиоканалу и не требовать применения ретрансляторов радиосигнала в пределах больничного отделения.

Для разработки такого датчика необходимо разработать модель рассеяния излучения в капле и выбрать параметры излучателя и приемника света.

2. Метод оптического контроля инфузии

Для автоматического контроля скорости инфузии в данной работе предлагается использовать метод оптического зондирования капельной камеры. Луч света распространяется от источника и принимается фотодатчиком на противоположной стороне камеры.

Капля, пересекая луч, изменяет сигнал фотодатчика. Изменение сигнала фотодатчика происходит из-за поглощения света в капле, его рассеяния и отражения от поверхности капли (рисунок 5).

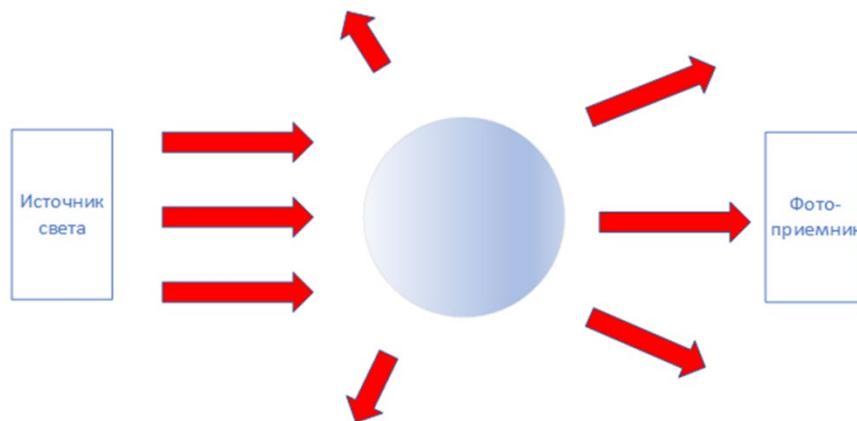


Рисунок 5 – Оптический метод контроля инфузии

2.1. Модель распространения светового луча через каплю

На рисунке 5 представлена разработанная модель рассеяния света в капле. В модели предполагается, что свет от источника распространяется в виде параллельного пучка, при этом диаметры источника и приемника равны.

Будем считать, что раствор прозрачный и рассеяние света на частицах лекарственного вещества отсутствует, поглощение также не рассматривается. Таким образом, единственной причиной уменьшения светового потока на фотоприемнике является отклонение лучей от прямолинейного распространения из-за преломления на границе двух сред.

отклонения от первоначального распространения луча. Используя равенство вертикальных углов, перпендикулярность нормали и касательной к поверхности сферы и равенство соответственных углов $\beta + \varepsilon$, получаем для угла отклонения от первоначального направления $\gamma = \varepsilon + \alpha = 2\alpha - 2\beta$. Подставляя выражение для β , полученное из закона Снеллиуса:

$$\gamma = 2(\alpha - \arcsin(\frac{n_1}{n_2} * \sin \alpha)). \quad (1)$$

2.2. Численное моделирование

2.2.1 Моделирование светорассеяния

Для численного моделирования рассматривалась двухмерная задача. Сечение луча представлялось N параллельными лучами, часть из которых или все лучи при движении капли отклонялись от первоначального направления. До попадания капли мощность принятого света принималась пропорциональной N . После отклонения каждого луча выполнялась проверка попадания в фотоприемник. Поскольку часть лучей не попадала в фотоприемник, суммарное количество принятых лучей уменьшалось и, следовательно, уменьшалась мощность. При расчетах было задано $N = 1000$, диаметр излучателя и фотоприемника принимались одинаковыми и равными D , диаметр капли всегда принимался равным 2,29 мм исходя из общепринятого объема капли воды 0,05 мл.

Для численного моделирования разработана программа в среде MathCAD (Приложение А).

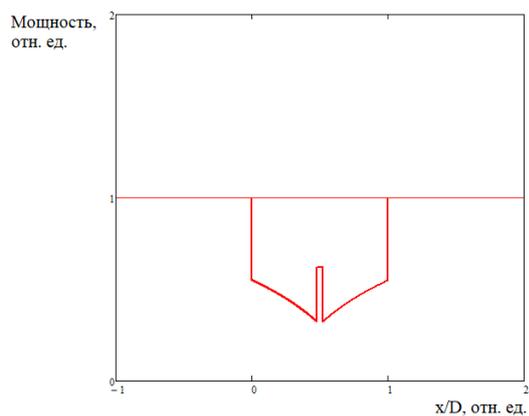
На рисунке 6 показаны результаты численного моделирования интенсивности при различных соотношениях между диаметром капли и диаметром луча.

На рисунке 6 обозначены случаи, соответствующие $D=2,5$ мм (а), 3,5 мм (б), 4,5 мм (в), 4,575 мм (г), 4,8 мм (д), 5,5 мм (е).

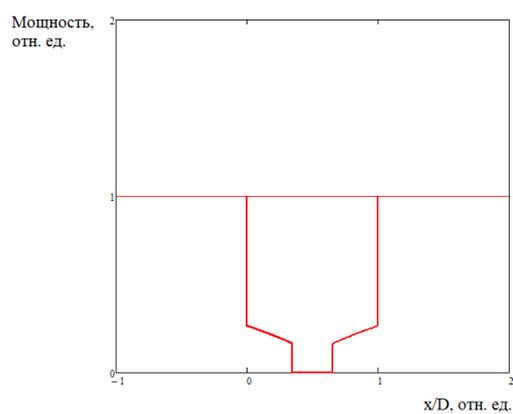
Из этих данных следует, что чем шире луч, тем уже импульс, возникающий на фотоприемнике, и тем труднее его зарегистрировать.

Кроме того, когда диаметр луча становится больше диаметра капли, импульс раздваивается, как показано на рисунке 7.

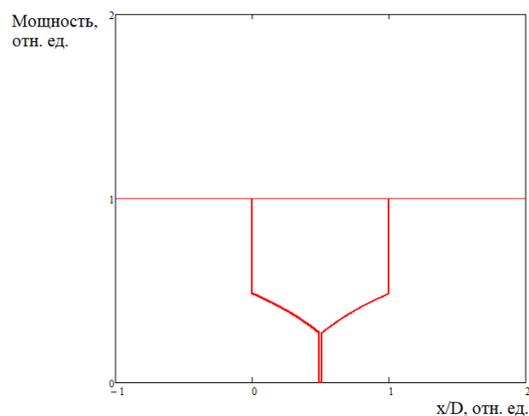
Поскольку при раздвоении импульса он может быть ошибочно принят за два, этот режим является нежелательным. Следовательно, в устройстве контроля инфузии диаметр луча должен быть меньше диаметра капли.



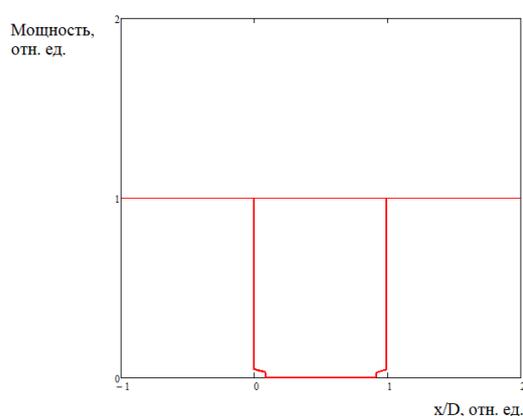
a



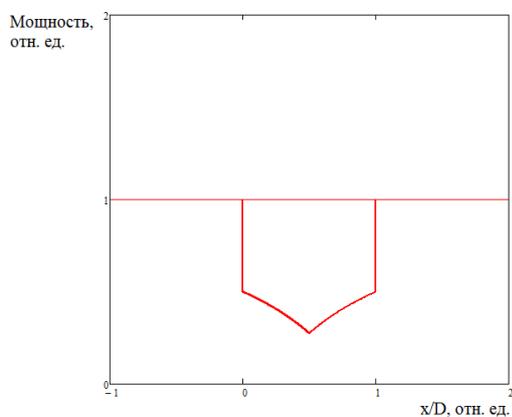
б



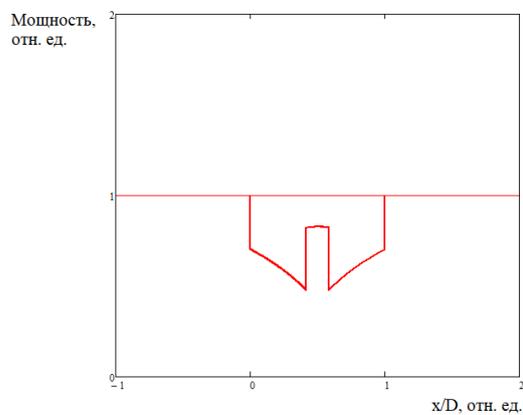
в



г



д



е

Рисунок 6 – Влияние диаметра луча на форму импульса фотоприемника

Исходя из полученных зависимостей, получено техническое требование к диаметру луча: от должен быть не больше диаметра капли, т.е. 4,6 мм. Чем меньше диаметр луча, тем лучше, так как ширина импульса увеличивается.

2.2.2 Исследование влияния коэффициента преломления

На рисунке 7 показана зависимость угла отклонения от угла падения для двух разных показателей преломления жидкости (1,333 для физраствора и 1,5 для 60% раствора глюкозы), рассчитанная по формуле (1).

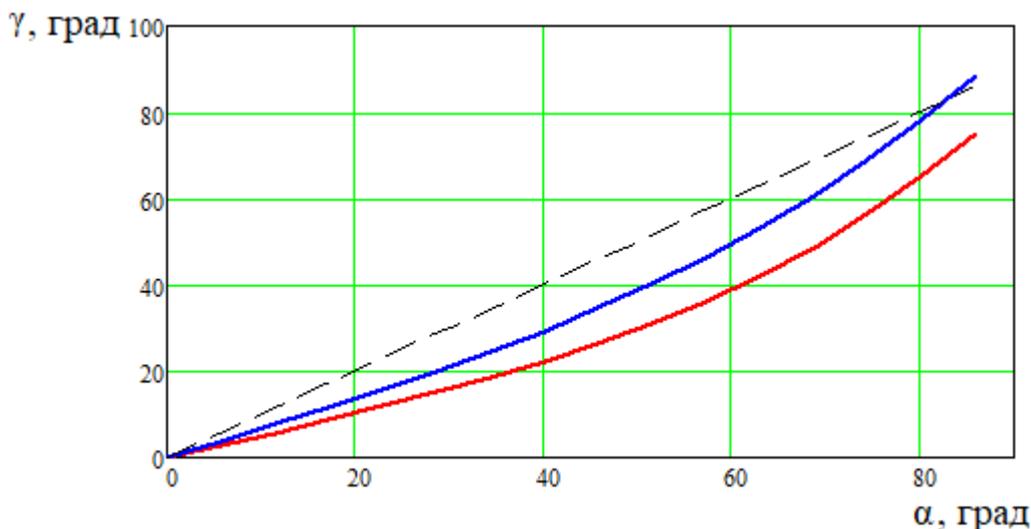


Рисунок 7 – Зависимость угла отклонения γ от угла падения α

Из этих данных следует, что показатель преломления жидкости несущественно влияет на характер преломления, поэтому принципиального отличия от зависимостей, показанных на рисунке 6, не произойдет.

2.2.3 Моделирование движения капли

Для того, чтобы оценить длительность импульса, необходимо решить уравнение движения капли и вычислить время пролета капли через область луча.

Используем формулы для равноускоренного движения материальной точки:

$$S_0 = gt_0^2, V_0 = gt_0$$

где V_0 – скорость при достижении верхней границы луча, g – ускорение свободного падения, t – время.

Задавая расстояние от крышки капельной трубки до верхней границы луча $S_0 = 26$ мм, получаем скорость $V_0 = 0,76$ м/с.

Задавая ширину луча $S = 4$ мм, решаем квадратное уравнение для времени пролета t через луч:

$$S = V_0 t + \frac{gt^2}{2}$$

В результате получаем $t = 0,0054$ с.

Для того, чтобы детально наблюдать форму капли, зададим минимальное число отсчетов по времени для считывания сигнала равное 10 за время пролета капли через луч.

Тогда получаем, что сигнал нужно считывать с интервалом времени не больше 0,00054 с. Частота считывания получается как обратная величина и равна 1850 Гц.

Таким образом, получено требование к частоте считывания сигнала, которая должна быть не меньше 1,9 кГц.

2.3. Экспериментальное исследование оптического метода контроля инфузии

Для исследования метода оптического зондирования собран экспериментальный стенд, в который входят (рисунок 8):

- 1) инфузионная система на штативе;
- 2) светодиод, имеющий максимум светимости на длине волны 590 нанометров;
- 3) фоторезистор, имеющий максимум чувствительности на длине волны 560 нанометров и кратность изменения сопротивления 8:1;
- 4) электрическая схема для подключения светодиода и фоторезистора;
- 5) модуль ввода данных производства компании L-card;
- 6) компьютер с установленной программой Lgraph для регистрации и отображения сигнала фотодатчика.

Электрическая схема обеспечивает подачу на светодиод постоянного тока 5 мА. Сопротивление фоторезистора регистрируется по падению напряжения от источника 5 В на резисторе 1 кОм. Это напряжение отображается в окне программы на компьютере в виде графика.



Рисунок 8 – Схема экспериментального стенда

Когда капля пересекает луч света, световой поток уменьшается, а сопротивление фоторезистора увеличивается. Это приводит к уменьшению регистрируемого напряжения и

появлению характерных провалов (рисунок 9). Удвоение импульса свидетельствует о том, что диаметр луча слишком большой: около 5 мм при диаметре капли около 4,6 мм.

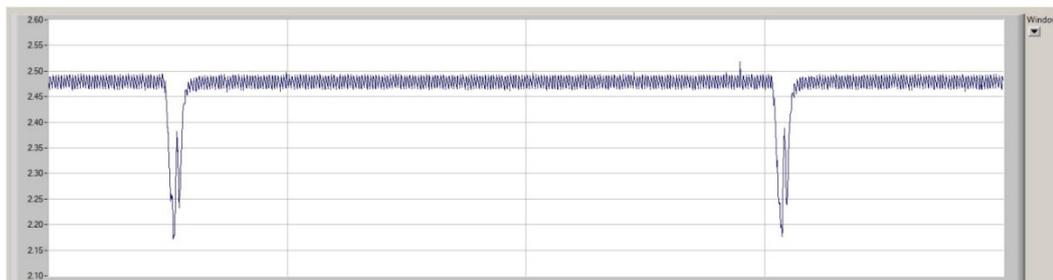


Рисунок 9 – Наблюдаемые импульсы в окне программы Lgraph

Полученные результаты показали, что изменение напряжения составляет около 30 мВ, а размах шума при этом не превышает 3 мВ. Это соответствует отношению сигнал-шум не менее 10.

Заключение

Выполненная работа показала возможность контроля скорости инфузии путем оптического зондирования:

1. Разработана модель преломления света в пузырьке с жидкостью
2. Численное моделирование позволило определить технические параметры устройства оптического контроля инфузии: диаметр луча не более 4,6 мм, скорость считывания данных с фотоприемника не менее 1,9 кГц.
3. Экспериментальные исследования подтвердили результаты численного моделирования и показали возможность достижения отношения сигнал-шум не менее 10.
4. Разработана структурная схема устройства и его отдельные компоненты.

На следующем этапе данной работы планируется изготовление элементов устройства и разработка программного обеспечения для микроконтроллера и компьютера.

Дальнейшее объединение устройств контроля инфузии в локальную беспроводную информационную сеть позволит уменьшить риск осложнений, повысить качество лечения и уменьшить нагрузку на медицинский персонал.

Список использованных источников

1. Большая российская энциклопедия. 2004-2019 [Электронный ресурс]: <https://bigenc.ru>
2. Саткеева А. Ж. Особенности инфузионной терапии // Бюллетень науки и практики. 2023. Т. 9. №4. С. 313-321. <https://doi.org/10.33619/2414-2948/89/38>
3. McCormick B. Control leaf of WHO — safety of surgical intervention // Update in Anaesthesia. 2009. Vol. 14. P. 5–6.
4. Silva et al. BioMedical Engineering OnLine (2023) 22:26 <https://doi.org/10.1186/s12938-023-01088-w>

5. Описание полезной модели к патенту RU 168516 U1, 2016 г.
6. Клестов С. А. Датчик автоматического подсчета объема вводимого лекарственного средства //Иноватика-2013. – 2013. – С. 73-75.
7. Барышева Е. Ю., Николаенко А. О., Кудашов И. А. Метод контроля объема вливания препаратов для безопасного использования инфузионных насосов //Политехнический молодежный журнал. – 2017. – №. 5. – С. 1.
8. Описание изобретения к патенту RU 2 745 581 C2, 2017 г.

РАЗРАБОТКА ПЛАТФОРМЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ДЛЯ ЗАДАЧ ПРОТИВОВОЗДУШНОЙ ОБОРОНЫ

Хаванов Иван

Москва г, ГБОУ Школа № 1409, 11 класс

Секция: Математика и естественные науки

Научный руководитель: Конев Станислав Андреевич, ассистент каф. ФН2

МГТУ им. Н.Э. Баумана

Аннотация

Защита воздушного пространства страны является одной из важных задач современной технической науки. Одним из способов защиты является разработка и совершенствование средств противовоздушной обороны (ракет, истребителей, средств радиоэлектронной борьбы и прочих).

Перспективным инструментом разработки и исследования средств ПВО является вычислительный эксперимент, когда натурные испытания заменяются компьютерными расчётами. Достоинствами этого подхода являются его относительная дешевизна, безопасность и возможность воссоздать некоторые критические режимы и экстремальные условия.

В данной работе построена математическая модель движения снаряда. Целью является разработка компьютерной платформы для расчёта траекторий снарядов и определение параметров, при которых происходит перехват. Базовой задачей, с которой начинается исследование, является задача перехвата неуправляемого снаряда противника с помощью такого же неуправляемого снаряда в трёхмерном пространстве. Условия перехвата снаряда формулируются в виде системы нелинейных уравнений, точное решение которой в общем случае затруднительно. Поэтому для решения используется метод градиентного спуска — с его помощью минимизируется ошибка решения системы нелинейных уравнений. Траектории снарядов находятся с помощью методов Рунге—Кутты интегрирования ОДУ. Эти задачи решаются совместно в рамках единого программного комплекса.

Созданный комплекс с высокой точностью находит параметры, при которых происходит перехват снаряда. Программная реализация метода выполнена с помощью языка C++. Полученная модель отличается гибкостью и имеет большой потенциал дальнейшего развития

ВВЕДЕНИЕ

В современном мире защита воздушного пространства является одним из актуальных аспектов национальной безопасности. Немаловажную роль играет уровень развития противовоздушной обороны государства. Противовоздушная оборона (ПВО) – это не просто ракеты и радары, это целая система, включающая в себя современные программные решения, способные в мгновение анализировать ситуацию.

Одним из перспективных инструментов разработки и исследования средств ПВО является вычислительный эксперимент, когда натурные испытания заменяются компьютерными расчётами. При таком подходе нет необходимости каждый раз проводить натуральный эксперимент, что позволяет существенно уменьшить затраты на разработку и тестирование новых технических образцов. Кроме того, если натурального эксперимента нет, то нет и опасности для исследователей и окружающей среды. Также в вычислительном эксперименте можно смоделировать поведение пока не существующих технических средств, чтобы оценить их возможную перспективность и целесообразность.

В основе любого вычислительного эксперимента лежит математическая модель и компьютерные алгоритмы. Модель определяет физику процесса, а компьютерные алгоритмы позволяют увидеть эту физику «в действии». Согласно [1, 2], вычислительный эксперимент — метод изучения устройств или физических процессов с помощью математического моделирования. Он предполагает, что вслед за построением математической модели проводится ее численное исследование, позволяющее «проиграть» поведение исследуемого объекта в различных условиях или в различных модификациях.

В вычислительном эксперименте можно выделить следующие этапы:



Рисунок 15 — Этапы вычислительного эксперимента

- Этап 1. Построение математической модели (составление уравнений, описывающих исследуемое явление).
- Этап 2. Выбор численных методов расчета (построение дискретной модели, аппроксимирующей исходную математическую задачу, разработка вычислительного алгоритма и так далее).
- Этап 3. Создание программы, реализующей вычислительный алгоритм.
- Этап 4. Проведение расчетов и обработка полученной информации.
- Этап 5. Анализ результатов расчетов, сравнение (если это возможно) с натурным экспериментом. [3]

Предметом исследования в рамках проекта является математическая модель движения снаряда. Целью работы является разработка платформы математического моделирования, программного комплекса, способного описывать процессы полета и перехвата снаряда.

Задачи проекта:

- провести теоретическое и численное исследование уравнений движения тела, учесть сопротивление среды,
- выбрать подходящие численные методы,
- разработать платформу на языке C++,
- провести верификацию,
- определить перспективы развития созданной платформы.

Базовой задачей, с которой начинается исследование, является задача перехвата неуправляемого снаряда противника с помощью такого же неуправляемого снаряда в трёхмерном пространстве.

Для описания движения снарядов противника и защитника используются дифференциальные уравнения движения материальной точки в трёхмерном пространстве. Для каждого из снарядов учитывается сопротивление среды.

Для решения задачи составляем уравнения движения снарядов в трёхмерном пространстве. Затем в рамках платформы математического моделирования реализуем численные методы решения систем нелинейных уравнений и методы интегрирования уравнений движения. Язык разработки — C++. Графические иллюстрации и верификация программы выполнены с помощью Wolfram Mathematica.

Модель имеет большой практический потенциал. При последующей доработке и развитии комплекса есть возможность учесть различные факторы (погодные условия, внешние ограничения и прочие). Область применения: разработанный программный комплекс может быть полезен как в учебных целях — например, в рамках лабораторного практикума для студентов соответствующих специальностей, а также как вспомогательный инструмент при анализе существующих систем ПВО, для повышения их эффективности и надёжности в заданных условиях.

ОСНОВНАЯ ЧАСТЬ

1 Базовая задача в трёхмерном пространстве

Начнём разработку модели с решения задачи о движении тела, брошенного под углом к горизонту, рассмотрим её в трёхмерном пространстве. Пусть нам необходимо бросить неуправляемый снаряд под определенным углом, чтобы сбить снаряд противника, летящий в нашу сторону. Определим угол, под которым нам нужно бросить снаряд.

На рисунке 2 представлена схема, иллюстрирующая ситуацию.

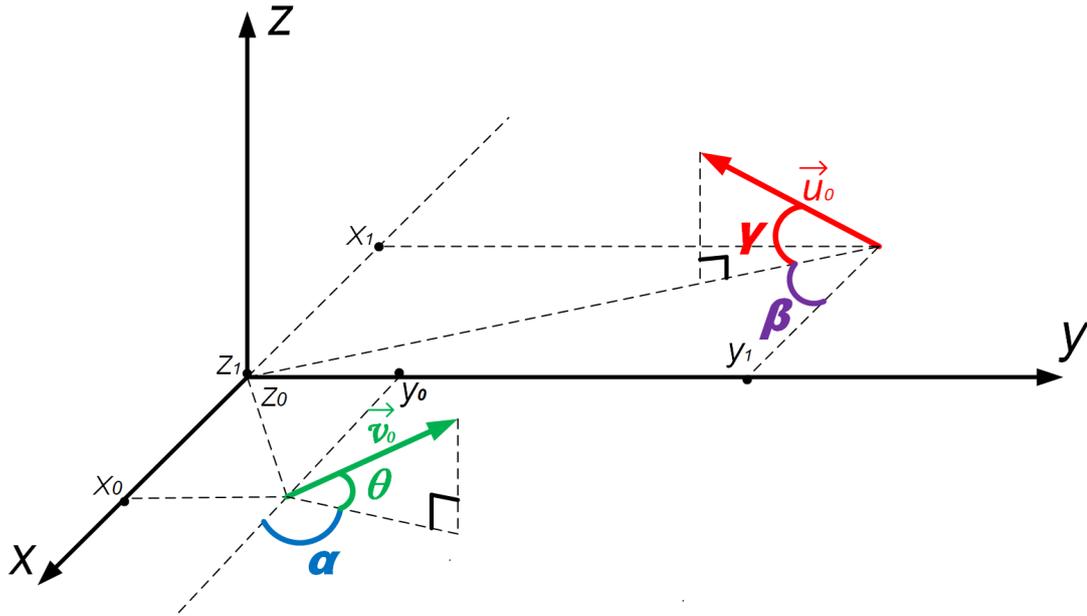


Рисунок 2 — Схема движения двух тел в трёхмерном пространстве

Составим уравнения движения тела в трёхмерном пространстве. Сопротивление среды пока не учитываем:

$$x''(t) = 0, \quad x(0) = x_0, \quad x'(0) = v_{0x}, \quad (1)$$

$$y''(t) = 0, \quad y(0) = y_0, \quad y'(0) = v_{0y}, \quad (2)$$

$$z''(t) = -g, \quad z(0) = z_0, \quad z'(0) = v_{0z}. \quad (3)$$

Представленные уравнения допускают точное решение. Тогда координаты снарядов по осям Ox , Oy , Oz выражаются формулами:

$$x = x_0 + v_0 \cos \theta \cos \alpha t, \quad (4)$$

$$y = y_0 + v_0 \cos \theta \sin \alpha t \quad (5)$$

$$z = z_0 + v_0 \sin \theta t - \frac{gt^2}{2} \quad (6)$$

$$x_e = x_1 + u_0 \cos \gamma \cos \beta t \quad (7)$$

$$y_e = y_1 + u_0 \cos \gamma \sin \beta t \quad (8)$$

$$z_e = z_1 + u_0 \sin \gamma t - \frac{gt^2}{2}, \quad (9)$$

где x_0, y_0, z_0 — координаты точки, из которой происходит запуск нашего снаряда,

x_1, y_1, z_1 — координаты точки, из которой происходит запуск снаряда противника,

v_0 — модуль начальной скорости нашего снаряда,

u_0 — модуль начальной скорости снаряда противника,

$\vartheta\alpha$ — угол запуска нашего снаряда в координатах Ox, Oy (в соответствии с рисунком 2),

$\vartheta\gamma$ — угол запуска нашего снаряда относительно горизонта (в соответствии с рисунком 2),

$\vartheta\beta$ — угол запуска снаряда противника в координатах Ox, Oy (в соответствии с рисунком 2),

$\vartheta\gamma$ — угол запуска снаряда противника относительно горизонта (в соответствии с рисунком 2),

g — ускорение свободного падения,

t — время полета.

Пусть $\Delta x = x_1 - x_0$, $\Delta y = y_1 - y_0$, $\Delta z = z_1 - z_0$, T — время полета тел до столкновения. Значит, $x(T) = x_e(T)$, $y(T) = y_e(T)$, $z(T) = z_e(T)$. Используя формулы (4), (5), (6), (7), (8), (9) получим систему уравнений

$$\begin{cases} \Delta x + (u_0 \cos \gamma \cos \beta - v_0 \cos \theta \cos \alpha)T = 0 \\ \Delta y + (u_0 \cos \gamma \sin \beta - v_0 \cos \theta \sin \alpha)T = 0 \\ \Delta z + (u_0 \sin \gamma - v_0 \sin \theta)T = 0, \end{cases} \quad (10)$$

2 Выбор метода

Полученная система является нелинейной, поэтому для ее решения необходимо использовать численные методы. Перепишем систему в виде

$$\begin{cases} f_1(\alpha, \theta, T) = 0, \\ f_2(\alpha, \theta, T) = 0, \\ f_3(\alpha, \theta, T) = 0; \end{cases}$$

где

$$f_1(\alpha, \theta, T) = \Delta x + (u_0 \cos \gamma \cos \beta - v_0 \cos \theta \cos \alpha)T,$$

$$f_2(\alpha, \theta, T) = \Delta y + (u_0 \cos \gamma \sin \beta - v_0 \cos \theta \sin \alpha)T,$$

$$f_3(\alpha, \theta, T) = \Delta z + (u_0 \sin \gamma - v_0 \sin \theta)T.$$

Определим функцию R ошибки решения системы следующим образом:

$$R(\alpha, \theta, T) = f_1^2(\alpha, \theta, T) + f_2^2(\alpha, \theta, T) + f_3^2(\alpha, \theta, T). \quad (11)$$

Условие $R(\alpha, \theta, T) = 0$ выполняется тогда и только тогда, когда тройка α, θ, T является решением системы (10). В то же время это решение является точкой минимума функции $R(\alpha, \theta, T)$. Следовательно, решение системы (10) эквивалентно поиску минимума функции $R(\alpha, \theta, T)$. Для поиска минимума функции $R(\alpha, \theta, T)$ будем использовать метод градиентного спуска.

2.1 Описание метода градиентного спуска

Градиент — это вектор, содержащий производные функции. Этот вектор всегда направлен в сторону скорейшего возрастания функции. Метод градиентного спуска — численный метод нахождения локального минимума функции с помощью движения против градиента. [8]

Алгоритм поиска минимума функции с помощью градиентного спуска:

- Выбрать начальную точку.
- Сделать шаг спуска — сдвинуться из этой точки против направления градиента. Получить новую точку.
- Повторять второй шаг, пока не подойдём к минимуму достаточно близко.

В рамках данной задачи на каждом шаге необходимо вычислять градиент функции $R(\alpha, \theta, T) = f_1^2(\alpha, \theta, T) + f_2^2(\alpha, \theta, T) + f_3^2(\alpha, \theta, T)$.

Суть метода градиентного спуска можно выразить формулой

$$\vec{x}_{n+1} = \vec{x}_n - w * \nabla f(\vec{x}_n), \quad (12)$$

где w — множитель, управляющий размером шага спуска, ∇f — градиент функции. Значение параметра w выбираем перед расчётом.

В нашей задаче формула (12) приобретает вид:

$$\begin{pmatrix} \alpha_{n+1} \\ \theta_{n+1} \\ T_{n+1} \end{pmatrix} = \begin{pmatrix} \alpha_n \\ \theta_n \\ T_n \end{pmatrix} - w * \begin{pmatrix} \frac{\partial R}{\partial \alpha}(\alpha_n, \theta_n, T_n) \\ \frac{\partial R}{\partial \theta}(\alpha_n, \theta_n, T_n) \\ \frac{\partial R}{\partial T}(\alpha_n, \theta_n, T_n) \end{pmatrix}, \quad (13)$$

где $\frac{\partial R}{\partial \alpha}$, $\frac{\partial R}{\partial \theta}$, $\frac{\partial R}{\partial T}$ — частные производные функции R (11), взятые по переменным α, θ, T соответственно.

Поскольку нам доступно выражение (11) для функции R , то и выражения для её производных тоже можно записать в формуле (13).

Теоретически минимум функции достигается ровно в той точке, где её градиент равен нулевому вектору. Поскольку на практике такой результат получить невозможно, будем останавливать итерации градиентного спуска, если длина вектора градиента на очередной итерации окажется достаточно малой.

Математически это условие записывается так:

$$|\nabla R(\alpha, \theta, T)| < tol, \quad (14)$$

где tol — достаточно малое число, которое выбирается перед расчётом.

3 Учёт сопротивления среды

В уравнениях движения (4), (5), (6), (7), (8), (9), мы не учитывали сопротивление среды (воздуха). Запишем уравнения движения с учётом сопротивления среды:

$$m x''(t) = -k v_x |\vec{v}|, \quad x(0) = x_0, \quad x'(0) = v_{0x}, \quad (15)$$

$$m y''(t) = -k v_y |\vec{v}|, \quad y(0) = y_0, \quad y'(0) = v_{0y}, \quad (16)$$

$$m z''(t) = -k v_z |\vec{v}| - g, \quad z(0) = z_0, \quad z'(0) = v_{0z}. \quad (17)$$

Здесь k — коэффициент лобового сопротивления снаряда. Он зависит от геометрии снаряда и описывает его аэродинамические свойства. Величина $|\vec{v}|$ — длина вектора скорости снаряда, v_x, v_y, v_z — проекции вектора скорости на оси координат Ox, Oy, Oz .

Формулу решения для этой системы можно выписать только в ограниченном числе случаев (например, в случае свободного вертикального падения или при $k = 0$). Но нам нужно

универсальное решение. Поэтому для решения системы дифференциальных уравнений (15), (16), (17) мы будем использовать численные методы Рунге—Кутты.

В данной работе используется явный 7-стадийный метод Рунге—Кутты с контролем точности, предложенный Дорманом и Принсом [12] в 1978 году.

Если решается задача вида:

$$u' = f(t, u),$$

то расчётные формулы метода имеют вид:

$$u_{n+1} = u_n + \tau \sum_{i=1}^s b_i k_i,$$

$$k_i = f(t_n + c_i \tau, u_n + \tau \sum_{j=1}^{i-1} a_{ij} k_j), i = \overline{1, s}$$

Где s – число стадий метода (у нас $s = 7$), $\{a_{ij}\}, \{b_i\}, \{c_i\}$ — коэффициенты метода. k_i – значения производных решения в промежуточные моменты времени; τ — шаг интегрирования задачи (выбирается автоматически в процессе счёта); u_n — значения приближённого решения в моменты времени t_n .

Коэффициенты метода, следующие [12]:

$$c_2 = \frac{1}{5}, c_3 = \frac{3}{10}, c_4 = \frac{4}{5}, c_5 = \frac{8}{9},$$

$$a_{21} = c_2, a_{31} = \frac{3}{40}, a_{32} = \frac{3}{40}, a_{33} = \frac{9}{40}, a_{41} = \frac{44}{45}, a_{42} = -\frac{56}{15}, a_{43} = \frac{32}{9},$$

$$a_{51} = \frac{19372}{6561}, a_{52} = -\frac{25360}{2187}, a_{53} = \frac{64448}{6561}, a_{54} = -\frac{212}{729}, a_{61} = \frac{9017}{3168},$$

$$a_{62} = -\frac{355}{33}, a_{63} = \frac{46732}{5247}, a_{64} = \frac{49}{176}, a_{65} = -\frac{5103}{18656},$$

$$b_1 = \frac{35}{384}, b_2 = \frac{500}{1113}, b_3 = \frac{125}{192}, b_4 = -\frac{2187}{6784}, b_5 = \frac{11}{84}.$$

Для применения метода Рунге—Кутты нам надо переписать наши дифференциальные уравнения 2-го порядка (15), (16), (17) в виде системы 1-го порядка. Получим систему из шести уравнений.

$$x' = v_x \quad (18)$$

$$y' = v_y \quad (19)$$

$$z' = v_z \quad (20)$$

$$mv_x' = -kv_x \sqrt{v_x^2 + v_y^2 + v_z^2} \quad (21)$$

$$mv_y' = -kv_y \sqrt{v_x^2 + v_y^2 + v_z^2} \quad (22)$$

$$mv_z' = -kv_z \sqrt{v_x^2 + v_y^2 + v_z^2} - mg, \quad (23)$$

где k – коэффициент лобового сопротивления снаряда,

v_x, v_y, v_z – проекции вектора скорости на оси координат Ox, Oy, Oz ,

m – масса снаряда, g — ускорение свободного падения.

3.1 Модификация метода градиентного спуска

Теперь у нас нет выражения для траекторий x, y, z . Мы можем получить только числовые значения в конкретный момент времени.

Это значит, что мы не можем вписать в формулу (13) выражения для производных функции R . Поэтому вернёмся к общему выражению для функции ошибки R :

$$R(\alpha, \theta, T) = |\vec{r}(\alpha, \theta, T) - \vec{r}_e(T)|^2 = (x(\alpha, \theta, T) - x_e(T))^2 + (y(\alpha, \theta, T) - y_e(T))^2 + (z(\alpha, \theta, T) - z_e(T))^2 \quad (24)$$

Как и ранее, мерой ошибки метода является длина отрезка, соединяющего центры масс снарядов. Здесь

$\vec{r}(\alpha, \theta, T)$ — радиус-вектор нашего снаряда,

$\vec{r}_e(T)$ — радиус-вектор снаряда противника,

$x(\alpha, \theta, T), y(\alpha, \theta, T), z(\alpha, \theta, T)$ — координаты снаряда защитника,

$x_e(T), y_e(T), z_e(T)$ — координаты снаряда противника.

Вектор градиента построим по приближённым формулам:

$$\text{grad } R = \left(\frac{\partial \bar{R}}{\partial \alpha}, \frac{\partial \bar{R}}{\partial \theta}, \frac{\partial \bar{R}}{\partial T} \right), \quad (25)$$

где

$$\frac{\partial \bar{R}}{\partial \alpha} = \frac{R(\alpha + \varepsilon, \theta, T) - R(\alpha - \varepsilon, \theta, T)}{2 \varepsilon}, \quad (26)$$

$$\frac{\partial \bar{R}}{\partial \theta} = \frac{R(\alpha, \theta + \varepsilon, T) - R(\alpha, \theta - \varepsilon, T)}{2 \varepsilon}, \quad (27)$$

$$\frac{\partial \bar{R}}{\partial T} = \frac{R(\alpha, \theta, T + \varepsilon) - R(\alpha, \theta, T - \varepsilon)}{2 \varepsilon}. \quad (28)$$

Параметр ε отвечает за точность приближённого вычисления производной. Он выбирается перед расчётом. Желательно, чтобы ε был достаточно мал: тогда качество приближений выше. В то же время ε нельзя брать слишком малым.

4 Программная реализация

Программная реализация выполнена на языке C++. Программа находит искомые величины (углы запуска снаряда и время перехвата), используя метод градиентного спуска. Соответствующий код представлен в файле «Gradient_3.0.cpp» (Приложение А).

Для повышения читаемости код программы разделён на несколько заголовочных файлов:

- «Vector_operations.h» (Приложение Б),
- «System_of_equations.h» (Приложение В).

Для ввода известных параметров используется файл Input.txt, где задаются такие параметры:

- углы β, γ ,
- начальные скорости v_0, u_0 ,
- координаты $x_0, x_1, y_0, y_1, z_0, z_1$;
- массы снарядов m_0, m_1 ;

- коэффициенты сопротивления снарядов k_0, k_1 ;

Для определения параметров расчёта используется файл `Inits.txt`, где указаны:

- *step_w* — шаг w градиентного спуска,
- *max_iters* — максимальное число итераций, при достижении которого программа прекращает вычисления и выводит результаты,
- *tolerance* — параметр tol в условии (14)
- начальные значения α, θ, T , необходимые для запуска метода градиентного спуска.

Программа считывает данные из этих файлов при помощи функций, определённых в файлах `Files_read.h` (Приложение Г) и `Inits_read.h` (Приложение Д), производит расчёт.

Заданы следующие условия прекращения вычислений:

1. Достижение максимального числа итераций *max_iters*.
2. Выполнение условия (14).

Если хотя бы одно из условий выполняется, происходит следующее:

1. программа завершает работу,
 2. выводит информацию о перехвате в файл `Output.txt`;
 3. выводит траектории снарядов защитника и атакующего в файлы `Trajectory_defender.txt` и `Trajectory_attacker.txt` соответственно.
- 5 Верификация

Для системы «Wolfram Mathematica» написан код (рисунок 3), выполняющий построение траекторий движения снарядов. Программа считывает данные из файлов `Trajectory_defender.txt` и `Trajectory_attacker.txt` и на основе этих данных строит траектории снарядов в трёхмерном пространстве на временном промежутке $[0, T]$.

Соединение траекторий в конечных точках означает успешный перехват вражеского снаряда.

```

In[ ]:= (* 09.02.2024. Пишем программу, которая будет рисовать графики пересечения траекторий снарядов. *)
ClearAll["Global`*"];
|очистить всё
SetDirectory[NotebookDirectory[]];
|здать рабочую... |директория файла блокнота

In[ ]:= (* Считываем траекторию атакующего. *)
AttackerFileName = "trajectory_attacker.txt";
AttackData = Import[AttackerFileName, "Table"];
|импорт
(* Зададим траекторию просто как массив трёхмерных точек. *)
AttackerPostns = {};
Do[AppendTo[AttackerPostns,
|... |добавить в конец к
AttackData[[i, 2 ;; 4]], {i, 1, Length[AttackData]}];
|длина

In[ ]:= (* Считываем траекторию защитника. *)
DefenderFileName = "trajectory_defender.txt";
DefenderData = Import[DefenderFileName, "Table"];
|импорт

(* Также задаём траекторию просто как массив трёхмерных точек. *)
DefenderPostns = {};
Do[AppendTo[DefenderPostns,
|... |добавить в конец к
DefenderData[[i, 2 ;; 4]], {i, 1, Length[DefenderData]}];
|длина

In[ ]:= (* Рисуем графики траекторий *)
ListLinePlot3D[
|линейный график данных в 3D
{AttackerPostns, DefenderPostns}, AxesLabel -> {Style["x, 10км", Large, Italic, Black], Style["y, 10 км",
|обозначения ... |стиль |крупный |курсив |чёрный |стиль

```

Рисунок 3 — Код для построения траекторий движения снарядов (Wolfram Mathematica).

В таблице 1 представлены результаты расчётов для модели без учета сопротивления среды.

Таблица 1— Результаты без учета сопротивления среды

Варианты	1	2	3
β	3.926	3.926	3.141
γ	1.047	1.047	0.523
u_0	3	3	3.25
x_1	2	2	4.4
y_1	2	2	3.3
z_1	0	0.55	0.55
v_0	6.5	6.5	5.3
x_0	0	0	2.25
y_0	0	0	1.3
z_0	0	0	0
α	0.785	0.785	1.187

θ	0.411	0.623	0.551
T	0.379	0.417	0.477

В таблице 2 представлены результаты расчётов для модели с учетом сопротивления среды. Добавлены такие параметры, как коэффициенты сопротивления для каждого из снарядов и массы снарядов. Первые 14 строк содержат исходные данные, а последние 3 — значения углов α , θ и время до перехвата T , найденные с помощью созданной программы.

Таблица 2— Результаты с учетом сопротивления среды

Варианты	1	2	3
β	3.926	3.926	3.141
γ	1.047	1.047	0.523
u_0	3	3	3.25
x_1	2	2	4.4
y_1	2	2	3.3
z_1	0	0.55	0.55
m_1	20	20	20
k_1	0.05	0.05	0.05
v_0	6.5	6.5	5.3
x_0	0	0	2.25
y_0	0	0	1.3
z_0	0	0	0
m_0	100	100	100
k_0	0.025	0.025	0.025
α	0.777	0.772	1.171
θ	0.421	0.649	0.551
T	0.381	0.423	0.475

При учете сопротивления среды значения углов α , θ и время перехвата T изменились. На рисунках 4, 5, 6 представлены траектории снарядов, соответствующие вариантам из таблицы 2. Как видно на рисунках 4, 5, 6, траектории пересекаются, т.е. происходит перехват вражеского снаряда. Следовательно, программа работает корректно.

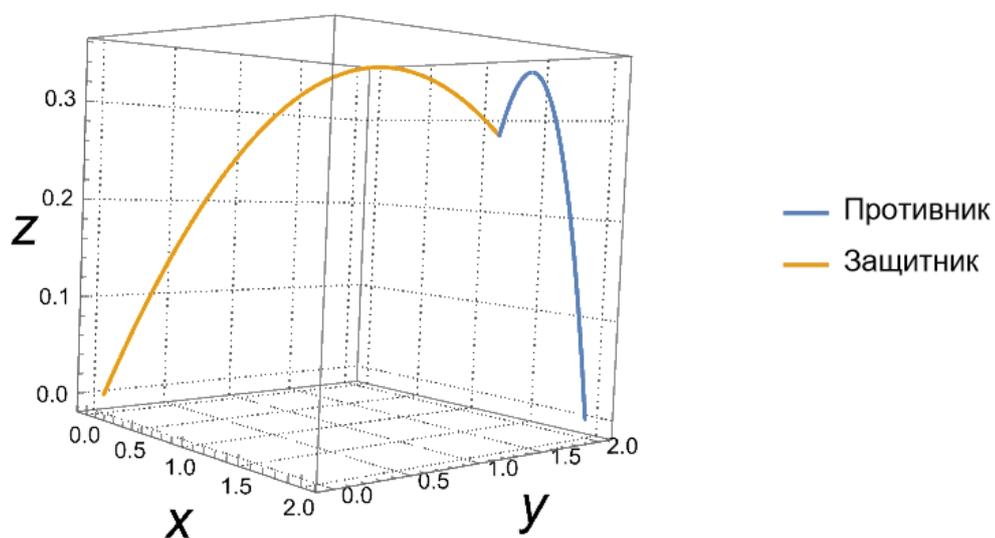


Рисунок 4 — Вариант №2 (Таблица 2)

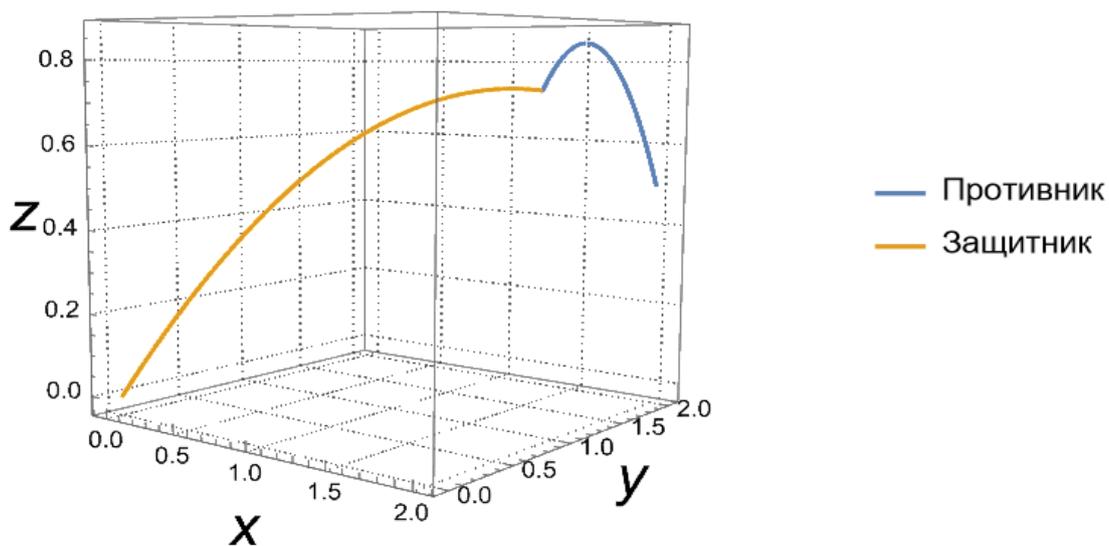


Рисунок 5 — Вариант №2 (Таблица 2)

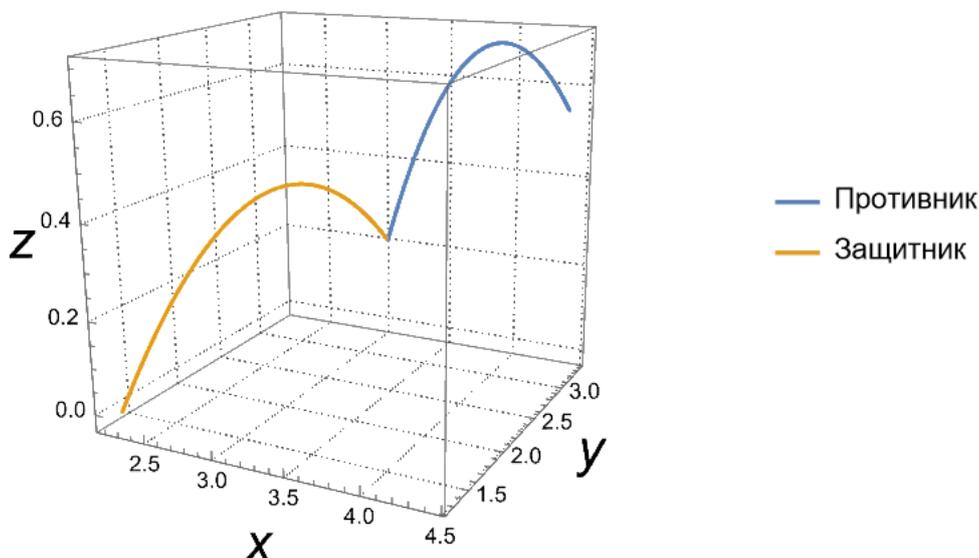


Рисунок 6 — Вариант №3 (Таблица 2)

ЗАКЛЮЧЕНИЕ

В результате работы над проектом построена модель полета неуправляемого снаряда в трехмерном пространстве. Решена задача о перехвате неуправляемого снаряда противника с учетом сопротивления среды.

Условия перехвата формулируются в виде системы нелинейных уравнений. По этим условиям построена функция ошибки системы, которая минимизируется с помощью метода градиентного спуска. Управляющим параметром метода является множитель w . Траектории снарядов находятся с помощью методов Рунге—Кутты интегрирования ОДУ. Эти задачи решаются совместно в рамках единого программного комплекса

Программная реализация выполнена на языке C++. Для облегчения работы и повышения читаемости код программы разделён на несколько заголовочных файлов.

С помощью программы Wolfram Mathematica проведена верификация: созданы графики траекторий движения неуправляемых снарядов до столкновения. Полученные результаты подтверждают работоспособность кода и обоснованность выбора метода

градиентного спуска. На графиках видно, что траектории пересекаются, то есть происходит перехват.

Платформа создана «с нуля», без использования готовых шаблонов. Разработан универсальный код для широкого использования, обладающий достаточной гибкостью.

В более сложной постановке можно рассмотреть ситуацию, когда вражеские снаряды сбиваются управляемыми и/или самонаводящимися ракетами. Такой подход потребует усложнения математической модели и программного кода. Однако это не является проблемой, поскольку существующая программная архитектура является гибкой.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Самарский А.А. Математическое моделирование и вычислительный эксперимент // Вестник АН СССР. — 1979.— № 5. — с.38-49.
2. Горбунов-Посадов М.М., Корягин Д.А., Мартынюк В.В. Системное обеспечение пакетов прикладных программ. — М.: Наука, 1990. — 208 с.
3. Горбунов-Посадов М.М. Расширяемые методы. М.: Полиптих, 1999. — гл.4.
4. Мякишев Г.Я., Буховцев В.В., Сотский Н.Н. Физика. 10 класс: учеб. для общеобразоват. организаций: базовый уровень; под ред. Н.А. Парфентьевой. — 3-е изд. — М.: Просвещение, 2017.
5. Колмогоров А.Н., Абрамов А.М., Дудницын Ю.П. [и др.] Алгебра и начала математического анализа: 10 — 11-е классы: учебное пособие; под ред. А.Н. Колмогорова. — 31-е изд., стер. — М.: Просвещение, 2023.
6. Галанин М.П., Савенков Е.Б. Методы численного анализа математических моделей. М.: Изд-во МГТУ им. Н.Э. Баумана. 2010. 591 с.
7. Галанин М.П., Лукин В.В., Щерица О.В. Методы вычислений. Задачи алгебры и анализа, учебное пособие. М.: Изд-во МГТУ им. Н.Э. Баумана. 2022. 376 с.
8. Калиткин Н.Н. Численные методы. СПб.: БХВ–Петербург. 2011. 592 с.
9. Прата С. Язык программирования C++. Лекции и упражнения, 5-е изд.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2007.
10. Стенли Б. Липпман, Жози Лажойе, Барбара Э. Му. Язык программирования C++. Базовый курс, 5-е изд.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2014.
11. Wolfram Language Documentation [Электронный ресурс].
[URL:https://reference.wolfram.com/language/](https://reference.wolfram.com/language/) (дата обращения 21.02.2025)
12. Dormand J.R., Prince P.J. New Runge—Kutta Algorithms for Numerical Simulation in Dynamical Astronomy // Celestial Mechanics, 18 (1978) pp. 223—232.
- 13.

Приложение А

Код программы Gradient_3.0.cpp

```
#include <iostream>  
#include <cmath>
```

```

#include <iomanip>
#include <functional>
#include <vector>
#include <fstream>
#include "system_of_equations.h"
#include "vector_operations.h"
#include "files_read.h"
#include "Inits_read.h"
struct Gradient_result
{
    std::vector<double> X{};
    double error = 0;
    unsigned iters = 0;
};
template <typename f_type>
auto Solver(const std::vector<double>& x0,
            const double step_w,
            f_type & R_function,
            const double eps,
            const unsigned long max_iters,
            std::ofstream & output_file,
            const double drv_step = 1e-5)
{
    auto x_old = x0;
    auto x_new = x0;
    auto grad = x0;
    auto temp_v1 = x0;
    auto error = 2 * eps;
    unsigned long iters = 0;
    while (error >= eps && iters < max_iters)
    {
        const auto& h = drv_step
        for (std::size_t i = 0; i < x_old.size(); ++i)
        {
            static std::vector<double> temp_arg{};
            vector_copy(x_old, temp_arg);
            temp_arg[i] += h;
            auto R_forw = R_function(temp_arg);
            vector_copy(x_old, temp_arg);
            temp_arg[i] -= h;
            auto R_back = R_function(temp_arg);
            grad[i] = (R_forw - R_back) / (2 * h);
        }
        error = R_function(x_old);
        vector_factor(grad, (-step_w), temp_v1);
        vector_sum(x_old, temp_v1, x_new);
        x_old = x_new;
        ++iters;
        if (iters % 100 == 0)
        {
            std::cout << "iters = " << iters << " error = " << error << '\t';
            std::cout << "|grad| = " << vector_length(grad) << '\n';
        }
    }
}

```

```

    }
    Gradient_result result;
    result.error = error;
    result.iters = iters;
    result.X = x_new;
    std::cout << "iters = " << iters << " error = " << error << '\n';
    vector_print(output_file, x_new);
    return result;
template <typename f_type>
auto Solver(Problem_inits& inits,
            f_type & R_function,
            std::ofstream & output_file)
{
    return Solver(inits.X0,
                  inits.step_w, R_function,
                  inits.tolerance, inits.max_iters, output_file,
                  inits.grad_step);
template <typename InputType>
auto argument_freezer(
    std::vector<InputType> flat_functions_array,
    std::vector<double> frozen_args,
    const std::size_t rows,
    const std::size_t cols
{
    auto test_function = std::bind(
        flat_functions_array[0],
        std::placeholders::_1,
        frozen_args
    );
    std::vector<std::vector<decltype(test_function)>>
        result;
    for (std::size_t i = 0; i < rows; ++i)
        result[i].resize(cols);
        for (std::size_t j = 0; j < cols; ++j)
            {
                result[i][j] = flat_functions_array[rows * i + j];
            }
    return result;
}
#include "rocket_science.h"
auto find_angles(
    double beta,
    double gamma,
    double v0,
    double u0,
    double x0,
    double x1,
    double y0,
    double y1,
    double z0,
    double z1,
    double m0,

```

```

    double m1,
    double k0,
    double k1,
    Problem_inits& inits,
    std::ofstream & output_file
)
{
    Rocket_Properties attacker;
    attacker.x0 = x1;
    attacker.y0 = y1;
    attacker.z0 = z1;
    attacker.alpha = beta;
    attacker.theta = gamma;
    attacker.v0 = u0;
    attacker.mass = m1;
    attacker.k_coeff = k1;
    Rocket_Properties protector;
    protector.x0 = x0;
    protector.y0 = y0;
    protector.z0 = z0;
    protector.v0 = v0;
    protector.mass = m0;
    protector.k_coeff = k0;
    auto R_function = std::bind(R_function_proto,
        std::placeholders::_1,
        attacker,
        protector);
    auto results = Solver(inits, R_function, output_file);
    ODE_options opts;
    opts.file_save = true;
    opts.filename = "trajectory_attacker.txt";
    double time = results.X[2];
    Get_Rocket_Position(attacker, time, opts);
    opts.filename = "trajectory_defender.txt";
    protector.alpha = results.X[0];
    protector.theta = results.X[1];
    Get_Rocket_Position(protector, time, opts);
    return results;
}
auto find_angles(
    Problem_parameters& values,
    Problem_inits& inits,
    std::ofstream & output_file
)
{
    double beta = values.beta;
    double gamma = values.gamma;
    double v0 = values.v0;
    double u0 = values.u0;
    double x0 = values.x0;
    double x1 = values.x1;
    double y0 = values.y0;
    double y1 = values.y1;

```

```

double z0 = values.z0;
double z1 = values.z1;
double m0 = values.mass0;
double m1 = values.mass1;
double k0 = values.k0;
double k1 = values.k1;
return find_angles(beta, gamma, v0, u0, x0, x1, y0, y1, z0, z1, m0, m1, k0,
    inits, output_file);
}
int main()
{
    std::string params_name = "Input.txt";
    auto params = parameters_read(params_name);
    std::string inits_name = "Inits.txt";
    auto inits = inits_read(inits_name);
    std::string output_name = "Output.txt";
    std::ofstream output_file;
    output_file.open(output_name.c_str());
    find_angles(params, inits, output_file);
    return 0;
}

```

Приложение Б

Код программы Vector_operations.h

```

#ifndef VECTOR_OPERATIONS_H__
#define VECTOR_OPERATIONS_H__

void vector_sum(const std::vector<double> & V1,
    const std::vector<double>& V2,
    std::vector<double> result)
{
    auto V1_sz = V1.size();
    if (V2.size() == V1_sz)
    {
        result.resize(V1_sz);
        for (std::size_t i = 0; i < V1_sz; ++i)
        {
            result[i] = V1[i] + V2[i];
        }
    }
}

void vector_factor(const std::vector<double>& V,
    const double factor,
    std::vector<double>& result)
{
    result.resize(V.size());
    for (size_t i = 0; i < V.size(); ++i)
    {
        result[i] = V[i] * factor;
    }
}

```

```

double vector_length(const std::vector<double>& V)
{
    double sqrt_sum = 0;
    for (std::size_t i = 0; i < V.size(); ++i)
    {
        sqrt_sum += V[i] * V[i];
    }
    return std::sqrt(sqrt_sum);
}
void vector_matrix_product(const std::vector<double>& L_Vec,
    const std::vector<std::vector<double>>& R_Matrix,
    std::vector<double>& result)
{
    if (R_Matrix.size() == L_Vec.size())
    {
        result.resize(R_Matrix[0].size());
        for (std::size_t i = 0; i < result.size(); ++i)
        {
            double temp = 0;
            for (std::size_t j = 0; j < L_Vec.size(); ++j)
            {
                temp += L_Vec[j] * R_Matrix[j][i];
            }
            result[i] = temp;
        }
    }
}
void vector_print(
    std::ofstream & output_file,
    const std::vector<double>& V)
{
    const short REQ_DIGITS = 16;
    for (std::size_t i = 0; i < V.size(); ++i)
    {
        output_file << std::setprecision(REQ_DIGITS)
            << V[i] << " ";
    }
    output_file << '\n';
}
#endif

```

Приложение В

Код программы System_of_equations.h

```

#ifndef SYSTEM_OF_EQUATIONS_H__
#define SYSTEM_OF_EQUATIONS_H__
template <typename T>
struct Args_Sliced
{
    T alpha = 0,
    theta = 0,

```

```

        t = 0,
        beta = 0,
        gamma = 0,
        v0 = 0,
        u0 = 0,
        dx = 0,
        dy = 0,
        dz = 0;
};
template <typename T>
Args_Sliced<T> args_slicer(
    std::vector<T> args,
    std::vector<T> froz_args
)
{
    Args_Sliced<T> result;
    result.alpha = args[0];
    result.theta = args[1];
    result.t = args[2];
    result.beta = froz_args[0];
    result.gamma = froz_args[1];
    result.v0 = froz_args[2];
    result.u0 = froz_args[3];
    result.dx = froz_args[4];
    result.dy = froz_args[5];
    result.dz = froz_args[6];
    return result;
}

double collision_X(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = dx;
    result += t * (
        u0 * std::cos(gamma) * std::cos(beta)
        -
        v0 * std::cos(theta) * std::cos(alpha)
    );
    return result;
}
double collision_X2(
    std::vector<double> args,

```

```

        std::vector<double> froz_args
    )
    {
        auto names = args_slicer(args, froz_args);

        return collision_X(
            names.alpha,
            names.beta,
            names.gamma,
            names.theta,
            names.v0,
            names.u0,
            names.dx,
            names.dy,
            names.dz,
            names.t
        );
    }
double collision_Y(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = dy;

    result += t * (u0 * std::cos(gamma) * std::sin(beta) - v0 * std::cos(theta) * std::sin(alpha));
    return result;
}

double collision_Y2(
    std::vector<double> args,
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return collision_Y(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,

```

```

        names.dy,
        names.dz,
        names.t
    );
}
double collision_Z(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = dz;
    result += t * (u0 * std::sin(gamma) - v0 * std::sin(theta));
    return result;
}
double collision_Z2(
    std::vector<double> args,
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return collision_Z(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}
double dcollision_X_dalpha(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t

```

```

)
{
    double result = 0;
    result += t * v0 * std::cos(theta) * std::sin(alpha);
    return result;
}
double dcollision_X_dtheta(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = 0;
    result += t * v0 * std::sin(theta) * std::cos(alpha);
    return result;
}
double dcollision_X2_dtheta(
    std::vector<double> args,
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return dcollision_X_dtheta(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}
double dcollision_X_dt(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,

```

```

    double t
)
{
    double result = 0;
    result += u0 * std::cos(gamma) * std::cos(beta) - v0 * std::cos(theta) * std::cos(alpha);
    return result;
}
double dcollision_X2_dt(
    std::vector<double> args,
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return dcollision_X_dt(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}
double dcollision_X2_dalpha(
    std::vector<double> args,
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return dcollision_X_dalpha(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}
double dcollision_Y_dtheta(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,

```

```

    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = 0;
    result += t * v0 * std::sin(theta) * std::sin(alpha);

    return result;
}
double dcollision_Y2_dtheta(
    std::vector<double> args,
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);

    return dcollision_Y_dtheta(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}
double dcollision_Y_dalpha(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = 0;
    result -= t * v0 * std::cos(theta) * std::cos(alpha);

    return result;
}
double dcollision_Y2_dalpha(
    std::vector<double> args
    std::vector<double> froz_args

```

```

)
{
    auto names = args_slicer(args, froz_args);
    return dcollision_Y_dalpha(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}
double dcollision_Y_dt(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = 0;
    result += u0 * std::cos(gamma) * std::sin(beta) + v0 * std::cos(theta) * std::sin(alpha);
    return result;
}
double dcollision_Y2_dt(
    std::vector<double> args
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return dcollision_Y_dt(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}

```

```

double dcollision_Z_dt(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = 0;
    result += u0 * std::sin(gamma) - v0 * std::sin(theta);
    return result;
}
double dcollision_Z2_dt(
    std::vector<double> args
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return dcollision_Z_dt(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}
double dcollision_Z_dalpha(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = 0;
    return result;
}

```

```

double dcollision_Z2_dalpha(
    std::vector<double> args,
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return dcollision_Z_dalpha(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,
        names.t
    );
}
double dcollision_Z_dtheta(
    double alpha,
    double beta,
    double gamma,
    double theta,
    double v0,
    double u0,
    double dx,
    double dy,
    double dz,
    double t
)
{
    double result = 0;
    result -= t * v0 * cos(theta);
    return result;
}
double dcollision_Z2_dtheta(
    std::vector<double> args,
    std::vector<double> froz_args
)
{
    auto names = args_slicer(args, froz_args);
    return dcollision_Z_dtheta(
        names.alpha,
        names.beta,
        names.gamma,
        names.theta,
        names.v0,
        names.u0,
        names.dx,
        names.dy,
        names.dz,

```

```

        names.t
    );
}
#endif

```

Приложение Г
Код программы Files_read.h

```

#ifndef FILES_READ_H__
#define FILES_READ_H__

#include <fstream>
#include <string>
struct Problem_parameters
{
    double beta = 0;
    double gamma = 0;
    double v0 = 0;
    double u0 = 0;
    double x0 = 0;
    double x1 = 0;
    double y0 = 0;
    double y1 = 0;
    double z0 = 0;
    double z1 = 0;
};

void parameters_read(
    std::istream& input,
    Problem_parameters& parameters
)
{
    input >> parameters.beta;
    input >> parameters.gamma;
    input >> parameters.v0;
    input >> parameters.u0;
    input >> parameters.x0;
    input >> parameters.x1;
    input >> parameters.y0;
    input >> parameters.y1;
    input >> parameters.z0;
    input >> parameters.z1;
}

auto parameters_read(
    const std::string & filename)
{
    ifstream input;
    input.open(filename.c_str());
    Problem_parameters params;
    if (input.is_open())
    {
        parameters_read(input, params);
    }
}

```

```

else
{
    std::cout << "COULD NOT OPEN FILE " << filename << ".\n";
}
return params;
}
#endif;

```

Приложение Д
Код программы Inits_read.h

```

#ifndef INITS_READ_H__
#define INITS_READ_H__

#include <fstream>
#include <string>
#include <vector>

struct Problem_inits
{
    double tolerance = 0;
    unsigned max_iters = 0;
    double step_w = 0;
    std::vector<double> X0 = {};
};

void inits_read(
    istream& input,
    Problem_inits& inits
)
{
    input >> inits.tolerance;
    input >> inits.max_iters;
    input >> inits.step_w;
    inits.X0.resize(3);
    input >> inits.X0[0];
    input >> inits.X0[1];
    input >> inits.X0[2];
}

auto inits_read(
    const std::string& filename)
{
    ifstream input;
    input.open(filename.c_str());

    Problem_inits inits;

    if (input.is_open())
    {
        inits_read(input, inits);
    }
    else

```

```
{  
    std::cout << "COULD NOT OPEN FILE " << filename << ".\n";  
}  
return inits;  
}  
#endif;
```

