МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н.Э. БАУМАНА

ЦЕНТР ДОВУЗОВСКОЙ ПОДГОТОВКИ

«ШАГ В БУДУЩЕЕ, МОСКВА»

НАУЧНО-ОБРАЗОВАТЕЛЬНЫЕ СОРЕВНОВАНИЯ МОЛОДЫХ ИССЛЕДОВАТЕЛЕЙ

«ШАГ В БУДУЩЕЕ, МОСКВА» и «ШАГ В БУДУЩЕЕ, КОСМОНАВТИКА»

СБОРНИК ЛУЧШИХ РАБОТ

Москва
2019

ДОРОГИЕ ДРУЗЬЯ!

Вы держите в руках сборник, в который вошли лучшие статьи участников Научно-образовательных соревнований молодых исследователей «Шаг в будущее, Москва» и «Шаг в будущее, Космонавтика», проходившего 14-19 марта 2019 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, Москва» 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, Москва» является уникальным методом профессиональной подготовки, которая осуществляется круглогодично в научно-исследовательских лабораториях школьников, созданных на базе кафедр Университета и ведущих предприятий Москвы, 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; Радио-опто-электроника; Биомедицинская техника; Специальное машиностроение, Энергетика и экология, Автоматизация, робототехника и механика; Инженерный бизнес и менеджмент; Фундаментальные науки; Образовательно-реабилитационные 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. Также проводились региональные и окружные научные конференции, экскурсии на ведущие

предприятия Москвы и в музей МГТУ им. Н.Э. Баумана, научно-методические совещания и семинары для профессиональных работников, другие интеллектуальные и творческие мероприятия.

Благодаря научно-образовательной и профессионально-ориентированной подготовке на базе научно-исследовательской работы молодежи в стенах МГТУ им. Н.Э. Баумана, наша страна получает целеустремленных и энергичных молодых специалистов, способных создавать высокие технологии, новую технику, проводить фундаментальные научные разработки.

Научно-образовательное соревнование «Шаг в будущее, Москва» 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. Победители на торжественном закрытии награждаются дипломами и ценными подарками от МГТУ им. Н.Э. Баумана, промышленных предприятий и спонсоров.

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», олимпиада школьников по черчению и компьютерному моделированию, олимпиада по программированию для школьников «ТехноКубок» совместно с Компанией Mail.Ru Group 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 сегодняшней 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.

СОДЕРЖАНИЕ:

ШАГ В БУДУЩЕЕ, МОСКВА

СЕКЦИЯ I. Образовательно-реабилитационные технологии 10

Запруднов Владислав Алексеевич
Реабилитация инвалидов 10

СЕКЦИЯ II. Машиностроительные технологии.....50

Соколова Надежда Дмитриевна
Исследование технологического процесса и режимов ультразвуковой сварки изделий из лавсановой ПЭТФ-пленки 50

СЕКЦИЯ III. Информатика и системы управления… 80

Коробова Екатерина Дмитриевна
Графические возможности программного пакета LabVIEW 80

Расюк Александр Александрович
Автономный контроллер дистанционного мониторинга и управления с функцией осциллографа 99

Моторкин Владимир Максимович
Метеостанция для измерения параметров атмосферы у земли и на высотах 174

Чудновский Даниэль Евгеньевич
Полнофункциональный мобильный редактор Polo с элементами социальной сети 217

Шумнов Пётр Алексеевич
Умный аранжировщик…….249

СЕКЦИЯ IV**.** Системы безопасности……273

Бушин Олег Алексеевич
Портативный сканер беспроводных сетей……..273

СЕКЦИЯ V. Радио-опто-электроника 307

Мифтахов Руслан Борисович
Использование датчиков мышечной активности для опознания напряжения мышц 307

СЕКЦИЯ VI. Биомедицинская техника..... 379

Осипкова Мария Викторовна
Разработка прототипа датчика для анализа движения опорно-двигательного аппарата 379

СЕКЦИЯ VII. Специальное машиностроение 432

Бондарь Георгий Евгеньевич
Мобильная роботизированная платформ для помощи в сельском хозяйстве «Siberian Tiger»..... 432

СЕКЦИЯ VIII. Энергетика и экология 504

Курочкина Кристина Александровна
Анализ работы двигателя Стирлинга 504

СЕКЦИЯ IX. Автоматизация, робототехника и механика 534

Сафронов Петр Андреевич
Автоматическая линия производства вязких лекарственных препаратов 534

СЕКЦИЯ XI. Фундаментальные науки…….612

Симановский Марк Анатольевич
Совершенствование численных методов решения многомерных оптимизационных задач…….612

ШАГ В БУДУЩЕЕ, КОСМОНАВТИКА

СЕКЦИЯ № 1. Космические аппараты и ракеты-носители……631

Иванов Никита Павлович
Ракеты-носители сверхлегкого класса……...631

СЕКЦИЯ № 2. Аэрокосмические системы……...662

Баринова Екатерина Александровна
Транспортно-пусковое устройство для запуска малых космических спутников при помощи пушки Гаусса…….662

СЕКЦИЯ № 4. Автономные радиоэлектронные устройства управления, автоматические системы и робототехника…….682

Затекин Дмитрий Владимирович
Система позиционирования мобильных роботов, воспроизводимая в учебных лабораториях цифрового производства…….....682

СЕКЦИЯ № 8. Ракетно-космические композиционные конструкции…….....715

Крутских Анна Андреевна
Эффективность использования электрического паруса……..……..715

СЕКЦИЯ I. Образовательно-реабилитационные технологии

Реабилитации инвалидов

Запруднов Владислав Алексеевич,

г. Москва, ГБОУ Школа №1501, 12 класс

Научный руководитель:

Константинов Михаил Дмитриевич,

ГУИМЦ МГТУ им. Н.Э. Баумана,

заведующий лабораторией

Оглавление

Введение

1 Постановка задачи и выбор устройства для иллюстрации процесса его создания

2 Описание процесса прототипирования печатной платы методом ЛУТ

2.1 Технология изготовления качественных печатных плат в домашних условиях

2.2 Инструмент и материалы

2.3 Процесс

3. Сценарии видеороликов. Выбор платформы для размещения видеокурса

4 Выводы

5 Список использованных источников

Введение

В XXI веке неотъемлемой частью многих приспособлений и предметов, окружающих человека в повседневной и профессиональной жизни, являются различные электронные устройства, начиная от домашнего телевизора, и кончая промышленными станками. В основе таких устройств лежат электрические схемы, состоящие из электронных компонентов, реализующих основную функцию данных устройств, и проводников, эти компоненты соединяющих. В подавляющем же большинстве подобных устройств связи между компонентами с помощью проводников делаются на основе так называемых печатных плат.

Целью данной работы является создание комплекса видеоинструкций по методике прототипирования и изготовления печатных плат электронных устройств, что позволит, во-первых, решить проблему получения прикладных навыков и понимания связи этих навыков с теоретическими знаниями, получаемыми студентами во время обучения. Во-вторых, как уже указывалось, печатные платы применяются повсеместно, и данный комплекс будет полезен для учащихся широкого круга технических специальностей.

Для достижения поставленной цели в работе планируется решить следующие **задачи**:

1. Выбор простого электронного устройства для дальнейшей иллюстрации процесса его создания.
2. Изучение и текстовое описание процесса прототипирования и изготовления печатных плат (так называемый метод «лазерного утюга»).
3. Написание сценариев для проведения видеосъемок процесса прототипирования и изготовления печатных плат.
4. Проведение съемок.
5. Монтаж и компоновка заснятых материалов в комплекс видеоинструкций.

1 Постановка задачи и выбор устройства для иллюстрации процесса его создания

В новом помещении факультета ГУИМЦ будет располагаться образовательное пространство нового поколение, где будут сочетаться и интегрироваться новейшие образовательные практики, ассистивные технологии и технические средства реабилитации инвалидов. В том числе там же будет находиться так называемый фаблаб – небольшая производственная и лабораторная площадка, где студенты-инвалиды совместно с разными преподавателями, специалистами и экспертами смогут заниматься проектной деятельностью. В настоящее время большинство проектов в инженерной области связано с созданием электронных устройств, как в виде отдельных девайсов, так и составных частей станков, машин и больших информационных систем. Электронные устройства в целом состоят из микросхем и электронных компонентов (резисторов, конденсаторов, транзисторов, диодов), источников электропитания и проводников, соединяющих все это вместе. Сейчас наиболее распространенной методикой (пример – все компьютеры и мобильные устройства) соединения этих электронных компонентов проводниками считается монтаж на так называемые печатные платы.

Печатная плата – пластина из диэлектрика, на поверхности и/или в объёме которой сформированы электропроводящие цепи электронной схемы. Печатная плата предназначена для электрического

и механического соединения различных электронных компонентов. Электронные компоненты на печатной плате соединяются своими выводами с элементами проводящего рисунка обычно пайкой. Печатные платы часто изготавливаются из фольгированного стеклотекстолита, где пластина диэлектрика (стеклотекстолита) с одной или обеих сторон покрыта тонкой фольгой (медной), из которой затем с помощью различных методов получается проводящий рисунок.

В проектной деятельности студентов-инвалидов будет присутствовать изготовление печатных плат, причем не в серийных, а в единичных количествах в рамках процесса создания прототипов электронных устройств (прототипирования).

Существует множество способов создания проводящих рисунков печатных плат из фольгированного стеклотекстолита, далее будут перечислены основные, которые могут использоваться для формата прототипирования.

1. Ручное нанесение рисунка краской с дальнейшим травлением незакрытой части фольги с помощью химических веществ. Трудоемкий, но самый дешевый метод. Не очень эффективен ввиду низкой точности получающегося рисунка.
2. Метод ЛУТ (лазерно-утюжная технология, «лазерный утюг»). Один из самых распространенных, дешевых и эффективных методов получения печатных плат не в промышленных условиях. Рисунок дорожек печатной платы распечатывается на глянцевой бумаге

лазерным принтером, затем переносится на фольгу посредством нагрева и давления утюгом при его совмещении с поверхностью фольги. Затем незакрытая часть фольги травится с помощью химических веществ.

3. Фрезерование печатной платы с помощью фрезерных станков. Фрезой убираются все лишние части фольгированной поверхности, остаются только проводящие дорожки.
4. Нанесение фоторезиста (вещества, которое затвердевает или наоборот делается растворимым при облучении электромагнитным излучением) на фольгу, его засветка через фотографию рисунка печатной платы, смывка лишней части фоторезиста, а затем дальнейшее травление в химическом растворе. Не бюджетный и гораздо более трудоемкий метод, чем ЛУТ.

В фаблабе ГУИМЦ на данный момент предполагается использование третьего метода из списка – фрезерования печатной платы. Тем не менее будущим участникам проектной деятельности очень желательно освоить более бюджетный способ создания печатных плат, чтобы они не были привязаны к оборудованию фаблаба при развитии своих проектов.

По причинам, обозначенным выше в списке, для работы и создания цикла обучающих видеороликов был выбран метод ЛУТ.

Для иллюстрации самого метода нет необходимости изготавливать сложное электронное устройство, поэтому была выбрана

простая схема, представляющая собой два автоматически мигающих светодиода. Эта схема показана на рисунке 1.

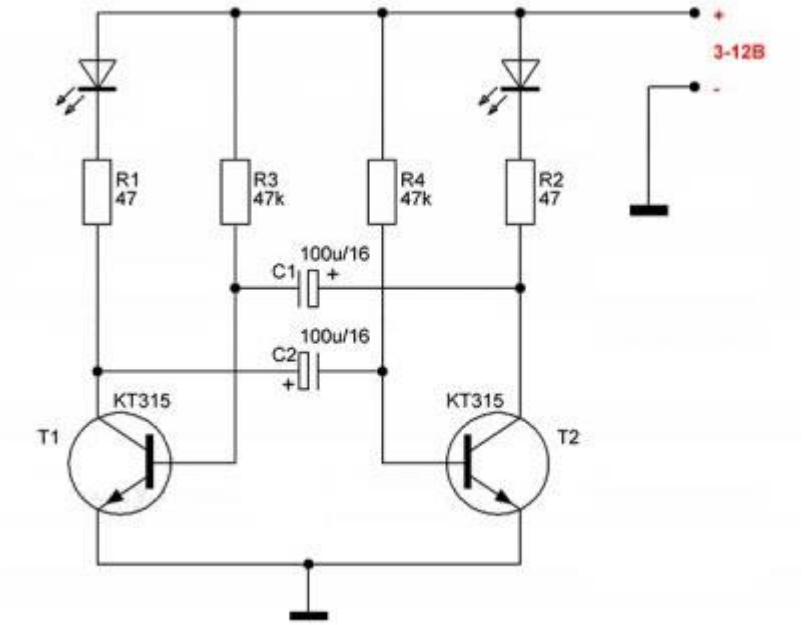


Рисунок 1 – Электрическая схема демонстрационного устройства

2 Описание процесса прототипирования печатной платы методом ЛУТ

2.1 Технология изготовления качественных печатных плат в домашних условиях

Метод ЛУТ основан на том, что защитный рисунок образуется тонером, который посредством нагревания переносится на текстолит. Так что нам потребуется лазерный принтер. Заправленные картриджи подходят крайне плохо, так как у них недостаточная плотность и равномерность выдачи тонера. В свойствах печати надо выставить максимальную плотность и контрастность тонера, обязательно отключить все режимы экономии – не тот случай.

2.2 Инструмент и материалы

Помимо фольгированного текстолита нам потребуется еще лазерный принтер, утюг, фотобумага, ацетон, мелкая шкурка, щетка для замши с металлопластиковым ворсом.

2.3 Процесс

Дальше рисуем рисунок платы в любой удобной для нас программе и печатаем его. В качестве такой программы можно использовать Sprint Layout. Чтобы нормально напечаталось надо слева цвета слоев выставить черным.

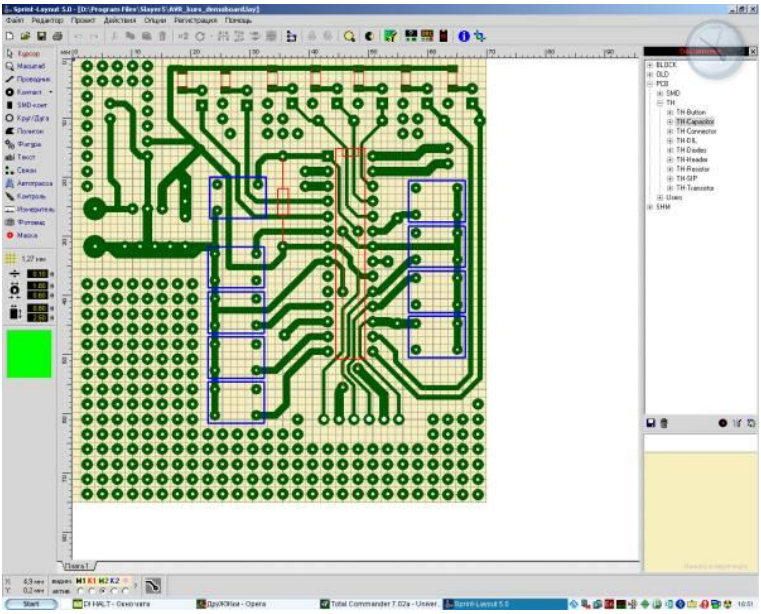


Рисунок 2 – Вид рабочей области программы Sprint Layout

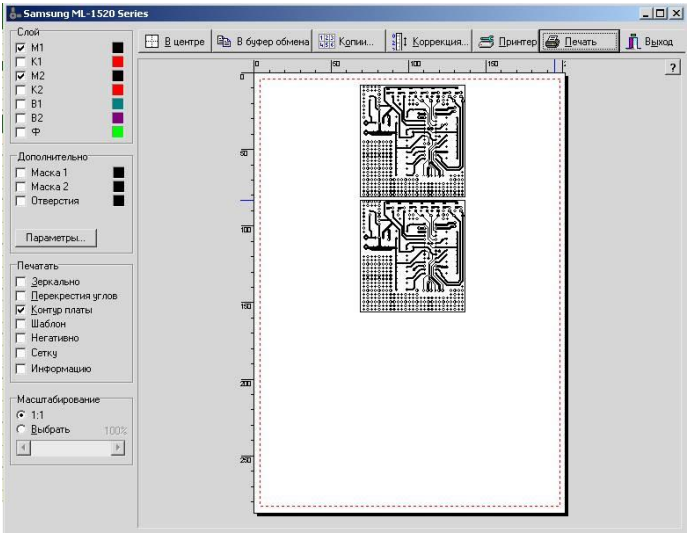


Рисунок 3 – Вид окна печати рисунка программы Sprint Layout

Вот тут заключается главная тонкость технологии **ЛУТ** из-за которой у многих возникают проблемы с выходом качественных плат, и они бросают это дело. Путем множества экспериментов было выяснено, что самый лучший результат достигается при печати на глянцевой фотобумаге для струйных принтеров. Идеальной некоторые практики называют фотобумагу **LOMOND 120г/м^2**.

Заряжаем бумагу в принтер и смело печатаем **на глянцевой стороне**. Печатать нужно в зеркальном отображении, чтобы после переноса картинка соответствовала действительности.

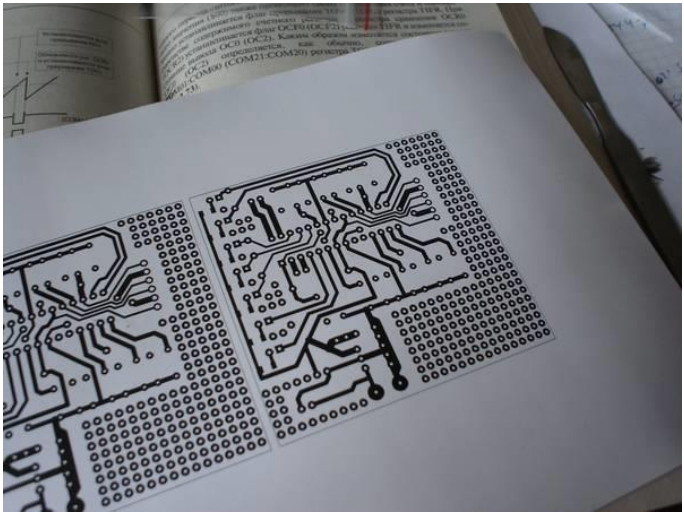


Рисунок 4 – Распечатанный рисунок печатной платы

После печати картинку ни в коем случае **нельзя хватать руками и желательно беречь от пыли**. Чтобы ничто не мешало соприкосновению тонера и меди. Далее вырезаем рисунок платы точно по контуру. Без каких-либо запасов – бумага жесткая, поэтому все будет хорошо.

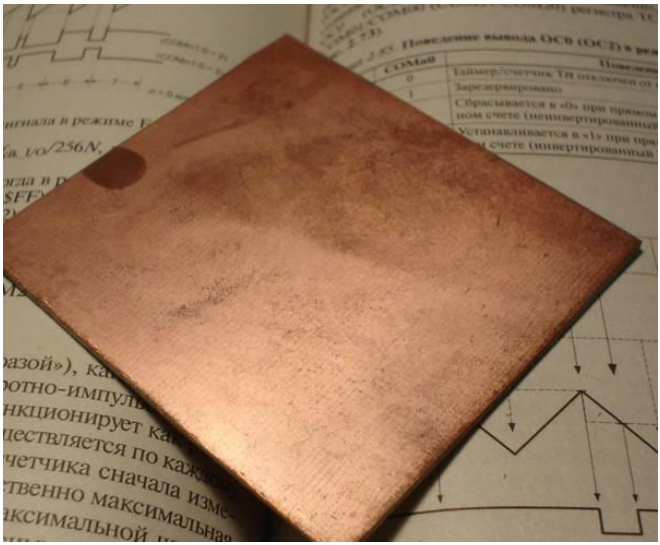


Рисунок 5 – Исходная заготовка печатной платы

Его надо хорошенько зашкурить. Тщательно, стараясь содрать весь окисел, желательно круговыми движениями. Немного шершавости не повредит – тонер будет лучше держаться. Можно взять не шкурку, а абразивную губку «эффект». Только брать надо новую, не жирную.

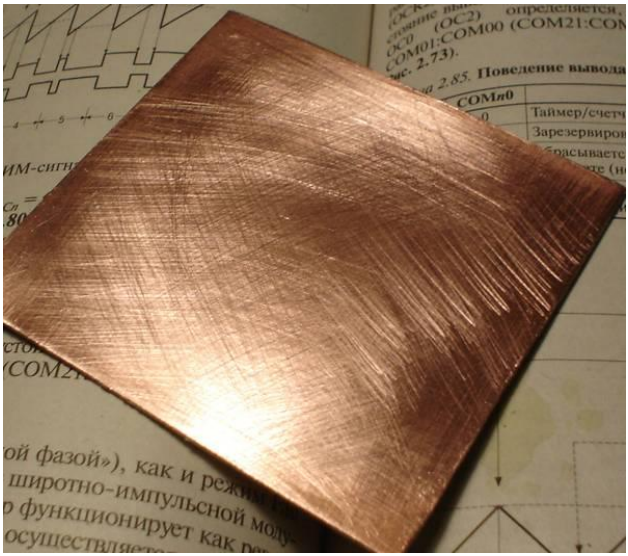


Рисунок 6 – Зашкуренная заготовка печатной платы

После зашкуривания его надо тщательнейшим же образом обезжирить, например, ацетоном.

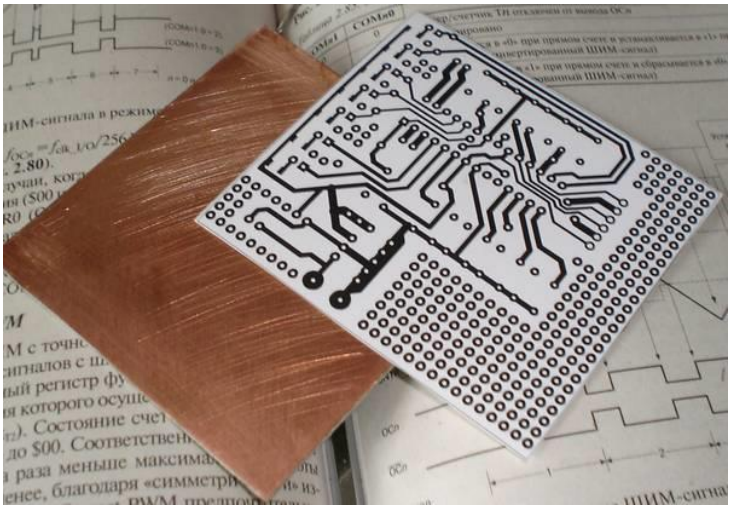


Рисунок 7 – Очищенная заготовка печатной платы и бумага с распечатанным рисунком

Накладываем наш рисунок на плату, естественно тонером вниз. Разогрев **утюг на максимум**, придерживая бумагу пальцем, хорошенько прижимаем и проглаживаем одну половину. Надо чтобы тонер прилип к меди.

Рисунок 8 – Процесс переноса рисунка с бумаги на фольгу стеклотекстолита

Далее, не допуская сдвижения бумаги, проглаживаем всю поверхность. Давим изо всех сил, полируем и утюжим плату. Стараясь не пропустить ни миллиметра поверхности. Это самая ответственная операция, от нее зависит качество всей платы. Тонер не поплывет и не размажется, так как фотобумага толстая и отлично защищает его от расползания.

Гладим до тех пор, пока бумага не пожелтеет. Впрочем, это зависит от температуры утюга.

Рисунок 9 – Заготовка с прикипевшей к ней бумагой после проглаживания

После можно дать плате немного остыть. А затем, схватив пинцетом, размещаем под струей воды. И держим некоторое время в воде, обычно минуты две три.

Взяв щетку для замши, под сильной струей воды, начинаем усиленно задирать внешнюю поверхность бумаги. Нам надо покрыть ее множественными царапинами, чтобы вода проникла в глубь бумаги. В подтверждение твоих действий будет проявление рисунка через плотную бумагу.

Рисунок 10 – Щетка для удаления остатков глянцевой бумаги

И вот этой щеткой трем плату, пока не сдерем верхний слой.

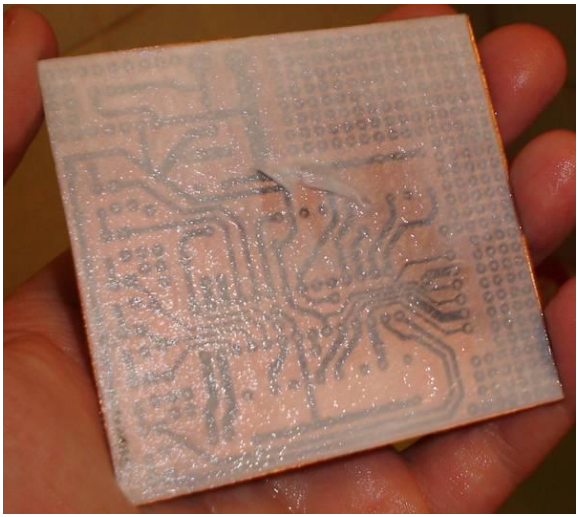


Рисунок 11 – Заготовка с промоченной глянцевой бумагой

Когда рисунок будет весь явно виден, без белых пятен, то можно начинать аккуратно, скатывать бумагу от центра к краям. Бумага **Lomond** скатывается великолепно, практически сразу же оставляя 100% тонера и чистую медь.

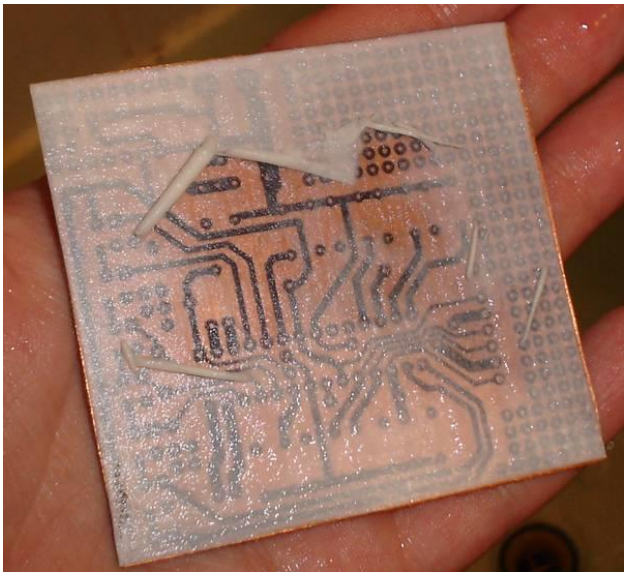


Рисунок 12 – Снятие бумаги с заготовки

Скатав пальцами весь рисунок можно зубной щеткой хорошенько продраить всю плату, чтобы вычистить остатки глянцевого слоя и ошметки бумаги. Зубной щеткой отодрать хорошо прижаренный тонер практически нереально.

Рисунок 13 – Очищенная заготовка с рисунком дорожек

Вытираем плату и даем ей просохнуть. Когда тонер высохнет и станет серым, то будет явно видно где осталась бумага, а где все чисто. Белесые пленочки между дорожками надо убирать. Можно разрушить их иголкой, а можно продрать зубной щеткой под струей воды. Вообще полезно пройтись щеткой вдоль дорожек. Из узких щелей белесый глянец можно вытаскивать с помощью изоленты или малярного скотча. Он липнет не так плотно, как обычный, и не срывает тонер. А вот остатки глянца отрывает без следа и сразу же.

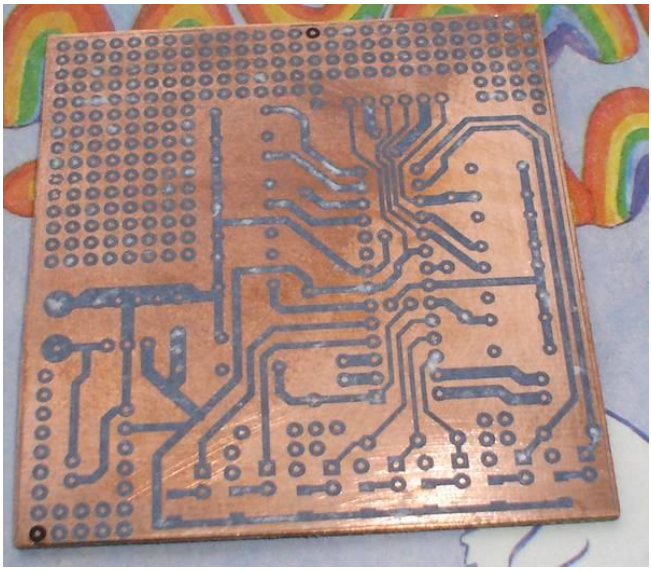


Рисунок 14 – Просушенная заготовка с рисунком дорожек и остатком глянцевого слоя

Под светом яркой лампы внимательно оглядываем слои тонера на разрывы. Дело в том, что при охлаждении он может потрескаться, тогда в этом месте останется узкая трещина. Под светом лампы трещины поблескивают. Эти места стоит подкрасить перманентным маркером для CD. Даже если есть лишь подозрение, то лучше все же прокрасить. Этим же маркером можно дорисовать и некачественные дорожки, если таковые возникли. Маркер **Centropen 2846** – он дает толстый слой краски и, фактически, им можно вообще рисовать дорожки.

Далее производится травление незакрытой рисунком части медной фольги. Для этого используют разные химические активы.

Медный купорос представляет собой голубоватые кристаллы, легко растворяющиеся в воде. Продается в любых магазинах для садоводов, так как является одним из самых ходовых садовых реактивов и удобрением. Так что проблем с его добычей не возникнет.

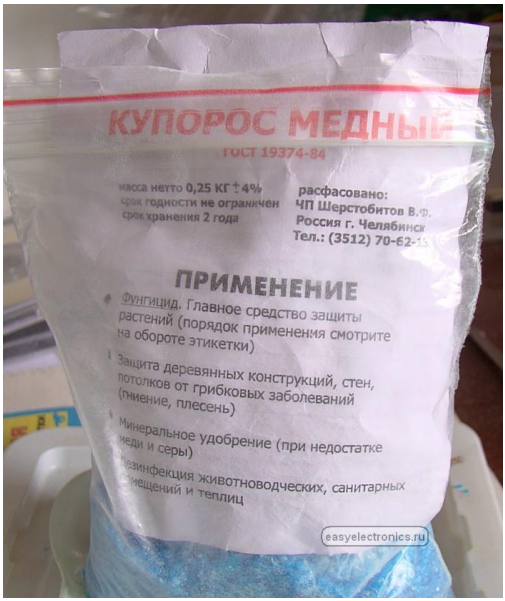


Рисунок 15 – Упаковка медного купороса

Как второй вариант, можно использовать для травления печатной платы хлорное железо.

Рисунок 16 – Упаковка хлорного железа

Для травления его надо смешать с повареной солью, в соотношении 1/3 купороса и 2/3 соли.

Рисунок 17 – Размещение реактивов в емкости

После чего все активно помешивалось для лучшего растворения солей. Впрочем, если емкость небольшая, то вода быстро остынет и почти весь процесс пойдет при комнатной температуре.

Рисунок 18 – Процесс травления платы

И полученным раствором можно травить десятки плат, пока не перестанет травить. Порошок надо сыпать в воду. И необходимо следить за тем, чтобы вода не перегревалась, а то реакция идет с выделением большого количества тепла.

Когда вся медь стравится, то аккуратно вынимаем плату и промываем под струей воды. Дальше смотрим на просвет, чтобы нигде не было недотравленных кусков фольги. Если таковые есть, то кидаем еще минут на десять в раствор. Если дорожки подтравились или возникли разрывы, то значит тонер криво лег и эти места надо будет пропаять медной проволокой.

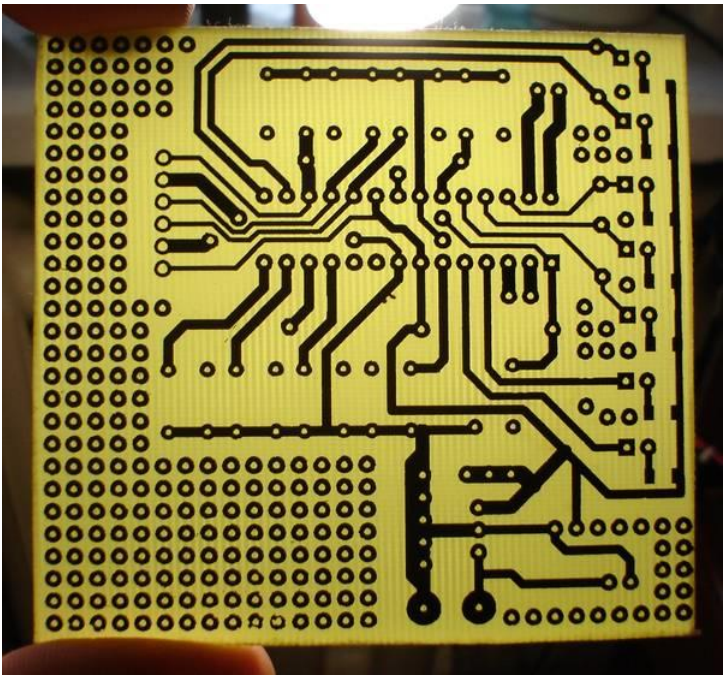


Рисунок 19 – Плата после травления

Если все хорошо, то можно смывать тонер. Для этого потребуется ацетон. Вместо ацетона вполне подходит **646 растворитель**.

Рисунок 20 – Плата после снятия с проводящих дорожек тонера

Берем кусок бинта и, хорошенько смочив его ацетоном, начинаем смывать тонер. Сильно давить не надо, главное двигать не слишком быстро, чтобы растворитель успевал впитываться в поры тонера, разъедая его изнутри. На смыв тонера уходит минуты две-три.

Отмытую плату можно сверлить.

Рисунок 20 – Микродрель

При сверловке нужно стараться держать сверло строго перпендикулярно. Иначе потом не получится в отверстия расположить, например, микросхему. А с двусторонними платами этот принцип становится основным.

Рисунок 21 – Плата с просверленными отверстиями

Изготовление двусторонней платы происходит также, только тут делаются три реперных отверстия, как можно меньшего диаметра. И после вытравливания одной стороны (другую в это время заклеивают скотчем, чтобы не стравилась) по этим отверстиям совмещают и накатывают вторую сторону. Первую заклеивают наглухо скотчем и травят вторую.

На лицевую сторону можно тем же ЛУТ методом нанести обозначение радиодеталей, для красоты и удобства монтажа.

3. Сценарии видеороликов. Выбор платформы для размещения видеокурса

Для съемки видеороликов необходимо разбить весь технологический процесс создания печатной платы методом ЛУТ на отдельные, законченные фрагменты (будущие ролики). К каждому фрагменту затем следует прописать небольшое описание (сценарий), где будет кратко рассказано, какие действия необходимо представить на ролике. Кроме того, к каждому фрагменту прописывается перечень инвентаря и техники, участвующих в съемке. Вся перечисленная выше информация представлена в таблице 1.

Таблица 1 – Сценарный план курса

№ ролика	Название	Описание (краткий сценарий)	Набор инвентаря	Примечание
1	Пилотный ролик	В ролике рассказывается про назначение всего обучающего цикла, показываются кадры из всех остальных роликов.	Изображение электрической схемы. Видеокадры работы сделанного устройства.	
2	Подбор компонентов, тестировани	Демонстрируются электронные компоненты, необходимые	Изображение электрической схемы.	Отметить пункт о технике безопасност

	е на макетной плате	для создания устройства по представленной электрической схеме. Затем схема собирается на макетной плате и запускается.	Все необходимые электронные компоненты (резисторы, конденсаторы, транзисторы, светодиоды). Источник напряжения. Макетная плата, проводящие перемычки.	и при работе с электричеством.
3	Создание рисунка печатной платы из электрической схемы	На основе электрической схемы и выбранных электронных компонентов в программе Sprint Layout делается разводка дорожек печатной платы. Выясняются габаритные размеры печатной платы (длина и ширина).	Изображение электрической схемы. Компьютер. Программа Sprint Layout.	Отметить пункт про минимальную ширину проводящих дорожек.

4	Подготовка заготовки к нанесению рисунка	По габаритным размерам из предыдущего фрагмента производится отрез заготовки от листа стеклотекстолита. Лист зачищается шкуркой. Лист очищается ваткой, промоченной в ацетоне.	Лист фольгированного стеклотекстолита. Ножницы по металлу. Шкурка с малым размером зерна («нулевка»). Ацетон. Вата (либо тряпка). Резиновые перчатки.	Отметить пункт о правилах работы с ацетоном.
5	Распечатка и нанесение рисунка	Из программы Sprint Layout лазерным принтером распечатывается рисунок дорожек на глянцевой бумаге (в зеркальном виде). Прогревается утюг. Полученный рисунок плотно прикладывается к заранее очищенной поверхности заготовки, и	Заготовка печатной платы. Компьютер. Программа Sprint Layout. Лазерный принтер. Глянцевая бумага для струйной печати. Утюг. Деревянная дощечка.	Отметить пункт о технике безопасности при работе с нагревательными элементами.

		медленно прокатывается утюгом. Бумага остается прикипевшей к поверхности заготовки.		
6	Очистка заготовки от следов бумаги	Заготовка с бумагой помещаются под струю теплой воды, через полминуты снимается бумага. Затем щеткой деликатно проходится поверхность с рисунком. В конце, участки заготовки, где остались следы глянцевого слоя бумаги, аккуратно очищаются подушечками пальцев рук. Производится контроль, что на поверхности фольги остались только дорожки, закрытые тонером	Заготовка печатной платы с рисунком. Раковина. Щетка. Резиновые перчатки.	

		лазерного принтера.		
7	Травление печатной платы	В ванночку наливается теплая вода (около 40-50 градусов). В нее ложечкой засыпается хлорное железо до момента, когда оно перестает растворяться в воде. Аккуратными движениями плата кладется на поверхность получившегося раствора и остается там плавать. Каждые пять минут производится контроль незакрытой рисунком фольги, которая травится раствором. Как только слой фольги полностью пропадает, плата	Заготовка печатной платы с рисунком. Ванночка. Вода. Хлорное железо (порошок). Пластмассовая ложка. Резиновые перчатки.	Отметить пункт про утилизацию хлорного железа.

		вынимается из раствора.		
8	Очистка заготовки от химическог о состава и тонера	Заготовка промывается под водой и вытирается. Затем ваткой или тряпкой, промокнутой в ацетоне, производится очистка дорожек печатной платы от тонера лазерного принтера.	Заготовка печатной платы с рисунком. Раковина. Ацетон. Вата (либо тряпка). Резиновые перчатки.	
9	Нанесение флюса и облуживани е дорожек	Заготовка платы размещается на держателе. Включается и прогревается паяльник. На дорожки платы наносится жидкий флюс. После этого жало паяльника помещается в сухой флюс, затем касается припоя, и производится нанесение тонкого слоя припоя на дорожки	Заготовка печатной платы с рисунком. Держатель платы («третья рука»). Паяльник. Флюс (спиртовой раствор канифоли и сухая канифоль). Припой. Тряпка. Ацетон. Вата (либо тряпка).	Отметить пункт о технике безопасност и при работе с паяльником.

		печатной платы. В конце, производится очистка поверхности платы от остатков флюса ваткой с ацетоном.	Резиновые перчатки.	
10	Сверление отверстий	Заготовка размещается на дощечке. Мини-дрелью производится сверление всех монтажных отверстий сверлом 0,7 мм. Проверяется, что в эти отверстия могут засовываться выводы электронных компонентов. При необходимости монтажные отверстия расширяются сверлами большего диаметра.	Заготовка печатной платы с рисунком. Деревянная дощечка. Мини-дрель. Сверла диаметров 0,7 – 1 мм.	
11	Монтаж и пайка	Заготовка платы размещается на держателе. Все	Печатная плата. Электронны	Отметить пункт о технике

	компонентов	электронные компоненты монтируются согласно электрической схеме в отверстия. Производится нанесение жидкого флюса на места соприкосновения выводов электронных компонентов и контактных площадок. Производится пайка выводов к контактным площадкам. Лишняя длина выводов отрезается кусачками. В конце, производится очистка поверхности платы от остатков флюса ваткой с ацетоном.	е компоненты. Держатель платы («третья рука»). Паяльник. Флюс (спиртовой раствор канифоли). Припой. Тряпка. Кусачки. Ацетон. Вата (либо тряпка). Резиновые перчатки.	безопасности при работе с паяльником.
12	Запуск устройства	Полученное устройство подключается к	Готовое устройство.	

		источнику питания и запускается. Кратко перечисляются все пройденные этапы.	Источник питания.	

В соответствии с представленным сценарием была произведена съемка видеороликов. Все пояснения в видеороликах делаются в виде текстовых подписей и субтитров.

Затем была выбрана платформа для публикации видеокурса: Eduardo. На этой платформе был выбран бесплатный тариф для небольших некоммерческих курсов. Платформа позволяет разбивать видеокурс на уроки, а каждый урок представляется в виде видеоролика и текстовых пояснений (будут браться из пункта 2 настоящей работы). Пример внешнего вида урока на платформе Eduardo представлен на рисунке 22.

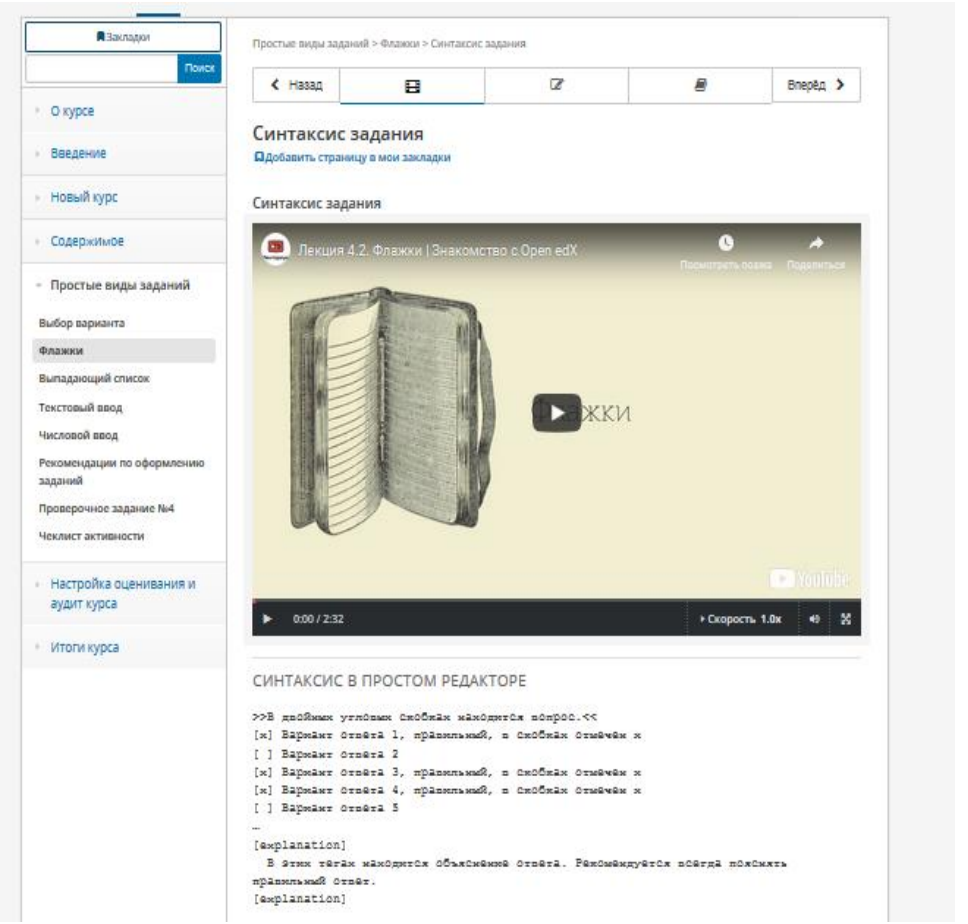


Рисунок 22 – Внешний вид урока на платформе Eduardo

4 Выводы

В результате данной работы я познакомился в теории и на практике с методом ЛУТ для создания печатных плат при прототипировании электронных устройств. Было получено:

1. Текстовое и графическое описание процесса создания печатных плат методом ЛУТ.
2. Сценарный план курса видеороликов по обучению методу ЛУТ.
3. Видеоролики, где наглядно демонстрируется применение вышеуказанного метода.
4. Создан курс на платформе Eduardo для более удобного просмотра информации и освоения навыков по применению метода ЛУТ.

Видеокурс будет полезен для студентов ГУИМЦ, желающих активно заниматься проектной деятельностью, а также просто для студентов, занимающихся или желающих заниматься в своей учебной и профессиональной деятельности созданием электронных устройств.

5 Список использованных источников

1. Ревич Ю.В. Занимательная электроника. 2-е издание – СПб. : БХВ-Петербург, 2009. – 722 с.
2. Лазерно-утюжная технология — Режим доступа: http://cxem.net/master/45.php (дата обращения 12.12.2018).
3. Методы изготовления печатных плат — Режим доступа: http://mirznanii.com/a/193218/metody-izgotovleniya-pechatnykh-plat (дата обращения 09.12.2018).
4. Печатная плата – изготовление в домашних условиях — Режим доступа https://ydoma.info/tehnologii-remonta/tehnologii-izgotovlenie-pechatnyh-plat.html (дата обращения 09.12.2018).
5. Изготовление высококачественных печатных плат в «домашних» условиях — Режим доступа: https://www.ixbt.com/mainboard/pcb-at-home.shtml (дата обращения 10.12.2018).

СЕКЦИЯ II. Машиностроительные технологии

Исследование технологического процесса и режимов ультразвуковой сварки изделий из лавсановой ПЭТФ- пленки

Соколова Надежда Дмитриевна
г. Мытищи, МБОУ СОШ №6, 11 класс
Научный руководитель:
Волков Станислав Степанович
к.т.н., профессор кафедры МТ-7,
МГТУ им. Н.Э. Баумана

СОДЕРЖАНИЕ:

ВВЕДЕНИЕ
2. ОБЗОР СУЩЕСТВУЮЩИХ СПОСОБОВ УЛЬТРАЗВУКОВОЙ ШОВНОЙ СВАРКИ
3. ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ
4. МЕТОДИКА ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТОВ
5. ОБОРУДОВАНИЕ ДЛЯ УЛЬТРАЗВУКОВОЙ СВАРКИ
6. ТЕХНОЛОГИЯ УЛЬТРАЗВУКОВОЙ СВАРКИ ЭЛАСТИЧНЫХ ЕМКОСТЕЙ
7. ВЫВОДЫ
8. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1.Введение

Стремительное развитие химии высокомолекулярных соединений привело к тому, что изготовленные на их основе полимерные материалы из заменителей превратились в незаменимые материалы, обладающие исключительно ценными свойствами.

Сварка полимерных материалов - это процесс получения неразъемного соединения, при котором поверхности соединяемых поверхностей активируют путём нагрева или введения растворителя с последующим приложения давления. При этом исчезает физико - химическая граница между свариваемыми поверхностями и получается соединение, по своим физическим свойствам приближающееся к основному материалу.

Сварка, как технологический процесс изготовления различных конструкций из полимерных материалов, позволяет в наилучшей степени реализовывать положительные свойства полимеров и способствует их интенсивному внедрению во всех областях народного хозяйства.

Полимерные материалы позволяют создавать сварные конструкции принципиально нового типа, которые невозможно или весьма затруднительно создавать из других материалов и другими способами. Это так называемые эластичные, оболочковые конструкции, это каркасные и бескаркасные пневмоопорные здания и

палатки, антифильтрационные экраны для сельскохозяйственных полевых целей и различные бассейны, временные плотины на мелких реках из тканных и нетканых материалов. Все эти конструкции могут быть сварены полностью на заводе, упакованы и доставлены на место монтажа.

С помощью сварки можно создавать сварные и ленточные конструкции, обладающие герметичностью, что их отличает от конструкций, полученных другими способами соединения.

С внедрением полимерных материалов в производство возникла необходимость совершенствования технологических процессов изготовления изделий из них. Это вызвано тем, что в большинстве случаев традиционные методы не обеспечивают качество изделий, соответствующего требованиям в процессе эксплуатации.

Анализ литературных источников показывает, что наиболее приемлемым способом сварки является ультразвуковой. Однако существующие схемы ультразвуковой сварки полимерных пленок не всегда обеспечивают предъявляемые к изделиям требования по прочности, стабильности качества сварных соединений [1,2].

В связи с этим совершенствование процесса ультразвуковой сварки изделий из указанных пленок и создание оборудования для его осуществления является весьма своевременной и важной задачей.

В нашей работе необходимо было решить следующие задачи:

1. Разработать ультразвуковые сварочные волноводы – инструменты, позволяющие стабилизировать процесс сварки и повысить прочность сварных полиэтилентерефталатных пленок.

2. Разработать технологический процесс, то есть отработать оптимальные режимы ультразвуковой сварки эластичных ёмкостей из полиэтилентерефталатных пленок с использованием разработанных устройств.

2.Обзор существующих способов ультразвуковой шовной сварки.

Для получения сварных швов на полимерных плёночных материалах необходимо относительное перемещение волновода и свариваемого изделия. Перемещение материала при получении протяжённых швов может вызвать повреждение разогретой плёнки. Поэтому при соединении полимерных пленок способ точечной ультразвуковой сварки получил дальнейшее развитие в следующих основных направлениях [1].

Непрерывная сварка обеспечивает получение непрерывных протяжённых сварных швов за счет относительного перемещения волновода и свариваемого изделия. Она используется для сварки изделий из полимерных пленок и синтетических тканей: мешков, непроницаемой одежды, фильтров и. т. д. Для непрерывной сварки используют схемы с фиксированной осадкой и фиксированным зазором [2].

По степени механизации непрерывная сварка делится на ручную и механизированную. При ручной сварке для получения непрерывных швов любой протяженности оператор вручную перемещает сварочную головку, а изделие остается неподвижным. При этом за счет клинообразной заточки инструмента и незначительной толщины свариваемых материалов инструмент контактирует с небольшим объектом полимера. Таким способом можно соединять многослойные пакеты из полиэтилентерефталатной плёнки, а также полипропиленовые и полиамидные плёнки, имеющие металлизированное, фотоэмульсионное или ферролаковое покрытие, без предварительного удаления последнего и. т. п. Такую разновидность сварки называют также сваркой скользящим инструментом [3]. Ручная сварка полиэтиленовых пленок затруднена из-за периодического налипания размягченного полимера на инструмент, что в конечном счёте приводит к прожогам свариваемого материала. Для устранения этого недостатка можно применять при сварке технологической прокладки из полимерных материалов (фторопласт и др.), металлической фольги, чертёжной кальки и др. Ручную сварку можно использовать для соединения в труднодоступных местах, а также для прихватки деталей перед механизированной сваркой. В последнее время речные ультразвуковые пистолеты нашли применение для сварки, наплавки и резки костной и других биологических тканей [3].

Механизированная непрерывная сварка, как правило, осуществляется при перемещении свариваемого изделия, а сварочный волновод остается неподвижным. В зависимости от способа перемещения свариваемого материала механизированная сварка делится на шовно- шаговую и шовную.

3. Постановка задач исследования.

В настоящее время разработано и выпускается большое количество полимерных плёночных материалов самыми разнообразными сочетаниями физико-механических, термических, химических и реологических свойств [1,2]. Разнообразие свойств полимерных плёночных материалов позволило применять их для укупорки различных пищевых продуктов. Часто полимерные пленки применяются при изготовлении ответственных пневматических конструкций, например, метеорологических зондов, крупногабаритных ёмкостей для хранения сыпучих продуктов питания, и, в частности, для изготовления эластичных ёмкостей для забора проб газа в газоанализаторных системах. В таких системах требованиям химической инертности и стойкости к газонепроницаемости, предъявляемым к материалу ёмкостей, отвечают высокоориентированные полимерные плёнки из полиэтилентерефталата (ПЭТФ) [3].

Полиэтилентерефталатная плёнка является материалом с большой степенью кристалличности и представляет практический

интерес благодаря исключительно высокой прочности (20 МПа), широкому интервалу рабочих температур (от -150 до +150ºС), стойкости к воздействию растворителей агрессивных сред.

Высокие механические свойства рассматриваемой плёнки обусловлены взаимной ориентацией молекулярных цепей, нарушение которой приводит к изменению свойств материалов, что естественно исключает возможность её сварки с помощью различных методов, основанных на нагреве до температуры плавления. Единственным методом сварки, приемлемым для соединения полиэтилентерефталатных пленок является ультразвуковая сварка, при которой материал нагревается до температуры более низкой, чем температура плавления [4,5].

Одним из основных преимуществ ультразвуковой сварки является возможность соединения ориентированных полимерных пленок, особенно из ПЭТФ, которые практически не могут свариваются другими, например, тепловыми процессами, так как длительное воздействие высоких температур приводит к нарушению молекулярной ориентации и повышение степени кристалличности, то есть к потере плёнками их ценных свойств. Это подтверждается сравнением данных измерения молекулярной ориентации акустическим методом [6] по изменению скорости звука в исходном материале и сварных швах, полученных ультразвуковой и термоимпульсной сварками. Было показано что, скорость звука в швах,

выполненных термоимпульсной сваркой, на 37 % ниже, чем в исходной плёнке, в то время как после ультразвуковой сварки она понижается на 09 и 11 процентов для плёнки из ПЭТФ толщиной 20 и 40 микрометров соответственно.

4. Методика проведения экспериментов

Ультразвуковая сварка обеспечивает расплавление в локальной зоне соединяемых поверхностей при сохранении относительно низких температур в слоях, контактирующих с ультразвуковым волноводом-инструментом и опорой, обеспечивая сохранение ориентации внешних слоев. Однако и она, зачастую, не обеспечивает предъявляемых к указанному изделию требования по прочности и плотности сварных соединений из-за значительных деформаций сварного шва и околошовной зоны.

Исходя из этого, решение задачи совершенствования технологического процесса ультразвуковой сварки данных изделий из ПЭТФ пленок и создание оборудования для его осуществления представляется весьма актуальным.

Ультразвуковая сварка ПЭТФ-пленок может осуществляться по различным технологическим схемам [1,2]- сварка по схеме «напротяг» ножевым волноводом- инструментом на неподвижной или вращающейся, настроенной или не настроенной опоре, шовная сварка вращающимся торцом ультразвукового инструмента на вращающейся

роликовой опоре, либо сварка точечным наклонным «скользящим» инструментом на неподвижной опоре и др. Для ПЭТФ- пленок наиболее приемлема схема сварки «напротяг» на вращающейся опоре. Однако ввиду того, что толщина плёнки в указанных изделия обычно составляет не менее 120 мкм, то из соображений прочности показателей, схема сварки «напротяг», на вращающейся опоре также не дает необходимого результата из-за недостатка энергии вводимых, как правило на частоте 22 кГц ультра - звуковых колебаний, амплитуда которых (30-50 мкм) соизмерима с толщиной плёночного материала. Снижение качества сварки при этом связано также и со значительными усадочными деформациями ПЭТФ- пленок свариваемых толщин.

При сварке ПЭТФ -пленок особое внимание следует обращать на геометрические размеры, материал и форму заточки рабочего торца волновода – инструмента. Оптимальная сферическая форма заточки рабочего торца волновода приведена на рис. 1.

Оптимальной является заточка типа «отвертки» с радиусом закругления около 1 мм. При большем радиусе требуется большое усилие прижима для создания необходимого статического давления на свариваемые плёнки. Сварку можно выполнять как широкой гранью при перемещении материала в направлении **а** (см. рис.1), так и узкой при перемещении материала в направлении **б**. В последнем случае скорость процесса несколько выше, однако больше вероятность образования подрезов материала в околошовной зоне. В первом случае

свариваемость в значительной степени зависит от радиуса закругления конца инструмента [6].

В нашей работе использована сферическая заточка торца волновода- инструмента с широкой гранью. Такая форма заточки позволяет производить сварку с наименьшим ослаблением околошовной зоны и более равномерно передавать усилия прижатия на свариваемые поверхности плёночного материала.

Определяющими технологическими параметрами процесса ультразвуковой сварки ПЭТФ - пленок являются: амплитуда колебаний волновода- инструмента, определяющая при заданной постоянной частоте энергию вводимых в материал колебаний; сила прижатия волновода-инструмента к материалу (сварочное статическое давление); скорость перемещения свариваемого материала под волноводом - инструментом или время пропускании ультразвукового импульса, зазор, установленный между рабочим торцом волновода – инструмента и опорой на которой располагается свариваемый материал и форма рабочего торца волновода- инструмента; частота ультразвуковых колебаний [1,2] .

Важнейшим энергетическим параметром процесса ультразвуковой сварки является частота ультразвуковых колебаний, повышение которой с 20 до 70 кГц может позволить значительно увеличить количество вводимой ультразвуковой энергии при

относительно небольших значениях амплитуды колебаний волновода и, следовательно, повысить качество сварных соединений.

Исходя из этого, одним из путей решения данной задачи является исследование влияния амплитудно-частотных характеристик ультразвуковых акустических систем, работающих на различных частотах на процесс сварки ПЭТФ- пленок и, по результатам этих исследований, создание акустических систем повышенной частоты, а также разработка технологии ультразвуковой сварки эластичных ёмкостей из ПЭТФ- пленок и ультразвукового сварочного оборудования для реализации данного процесса при изготовлении этих изделий.

Как следует из источников [1,6]- амплитуда ультразвуковых колебаний волноводов и частота акустических систем находятся во взаимосвязи, поскольку эти два параметра определяют величину энергии высокочастотных механических колебаний, вводимых в свариваемый материал. Известно также, что с повышением частоты ультразвуковых колебаний необходимое количество ультразвуковой энергии для сварки может быть получено при меньших значениях амплитуды колебаний рабочего торца волновода - инструмента, что имеет практический интерес при ультразвуковой сварке ПЭТФ - плёнок с толщинами, соизмеримыми со значениями амплитуды ультразвуковых колебаний волноводов- инструментов.

5. Оборудование ультразвуковой сварки пластмасс.

Исследование по ультразвуковой сварке ПЭТФ - пленок проводили в диапазоне частот от 20 до 70 кГц на шовной сварной машине модели УПШ – 19, разработанной на кафедре «Технологии сварки и диагностики» МГТУ имени и. Н. Э. Баумана. Машина предназначена для шовной ультразвуковой сварки ПЭТФ - пленок, а также дублированных плёночных материалов типа полиэтилен-целлофан толщиной 100 – 200 мкм. В машине используется схема автостабилизации толщины сварного шва. Сварка пленок производится по загрязненным поверхностям. Машина позволяет сваривать швы любой протяженности; мощность электродвигателя 0,5 кВт. Машина УПШ – 19 применяется для сварки рукавов, мешков, пакетов, а также эластичных ёмкостей из ПЭТФ - пленок.

Машина комплектуется никелевым магнитострикционным преобразователем мощностью 0,4 кВт с частотой от 20 до 70 кГц; амплитуда колебаний рабочего сферического торца сварочного волновода- инструмента А = 25 мкм; опора- ролик из стали СТ.3 размером ¼ длины волны, распространяющейся в стали; площадь пятна контакта сферического сварочного волновода S = 8 мм² (1мм- ширина торца волновода, 8 мм - его длина в направлении сварки).

Сварочный узел включает в себя магнитострикционный преобразователь (пакет); трансформатор упругих колебаний, являющихся согласующим акустическим звеном между волноводом и

преобразователем, и сферический волновод- инструмент, через рабочий торец которого ультразвуковые колебания вводятся в свариваемые изделия.

Схема машины УПШ – 19 приведена на рис. 2. На станине 1 сварочной машины крепятся следующие узлы и механизмы: преобразователь 2 и протягивающие ролики 4, связанные посредством цепной передачи с электродвигателем 8. Свариваемая плёнка, перемещаясь между роликом – опорой 5 и волноводом 3, поджимается к последнему пружиной 6. Сваренная плёнка наматывается на приемный барабан или поступает для дальнейшей обработки на рабочий стол. Сварочное статическое давление регулируется пружиной 6. Сварка с роликом при постоянном давлении позволяет на машине УПШ-19 выдерживать стабильную толщину сварного шва при колебаниях исходной толщины материала до 25 %, скорости сварки до 15 % и амплитуды колебаний волновода до 10 %. Охлаждение водяное; статическое сварочное давление от 0 до 2 МПа; скорость сварки 0,13-0,16 м/с.

В качестве источника питания магнитострикционного преобразователя (сварочного узла) мы использовали модернизированный серийно выпускаемый фирмой «Ультразвуковая техника ИНЛАБ» г. Санкт-Петербург ультразвуковой генератор типа НЛ10 –0,4 с расширенным диапазоном частот от 20 до 70 кГц. Генератор оснащен плавной регулировкой мощностью от 0,4 кВт, цифровым частотомером

и аналоговом индикатором резонанса акустической системы. Генератор имеет фазовую автоподстройку частоты и источник поляризации с выходным током до 30 А. Охлаждение генератора воздушное, принудительное. Выходная частота может меняться в любую сторону. Масса генератора 6,2 кг, габаритные размеры 115 х 150 х 60 мм.

Конструкция машины УПШ - 19 позволяет проведение экспериментов по схеме шовной сварки ПЭТФ - пленок с ультразвуковыми колебательными системами на всём диапазоне указанных частот. Для проведения экспериментов были спроектированы и изготовлены ультразвуковые сварочные узлы (магнитострикционные с акустической мощностью 0,4кВт с идентичным соотношением геометрических параметров рабочих сферических волноводов- инструментов по каждой частоте.

В процессе экспериментов частоту ультразвуковых колебаний контролировали с помощью цифрового частотомера, вмонтированного в ультразвуковой генератор ИЛ10 – 0,4, амплитуду колебаний волноводов- инструментов с помощью специально разработанных датчиков.

Влияние частоты на процесс сварки оценивали по прочностным показателям сварных соединений на образцах ПЭТФ - плёнки путём испытания сварных шов на расслаивание.

Нами были исследованы (рис. 3) выходные характеристики ультразвуковых акустических систем и определены максимальные значения амплитуды ультразвуковых колебаний волноводов-инструментов от частоты ультразвукового генератора.

6 Технология ультразвуковой сварки эластичных ёмкостей.

Были проведены опытные ультразвуковые сварки (рис. 4) с расширенным диапазоном частот от 20 ГЦ до 7 кГц. Результаты показали, что с повышением частоты ультразвуковых колебаний при прочих равных условиях, прочность сварных соединений повышается. Максимальные значения прочности сварных соединений лежат в приделах от 40 до 50 кГц. При частоте 45 кГц наблюдается значительное понижение значений прочностных показателей сварных соединений вследствие выхода акустической системы из резонанса при приложении рабочих значений сварочного статического давления, необходимого для получения качественного соединения, что подтверждается замерами амплитуд колебаний волновода в процессе сварки.

В результате проведённых нами экспериментов были установлены приделы значений параметров процесса ультразвуковой сварки, необходимые для разработки технологии сварки эластичных ёмкостей из ПЭТФ - пленок, основной из которых- частота ультразвуковых колебаний, составляет 40 кГц.

Эластичная ёмкость (рис. 5) служит для временного хранения выхлопных газов автомобильной и др. техники для последующего изучения их состава, в том числе и на содержание CO в стационарных анализаторных системах. Ёмкость выполнена из ПЭТФ - плёнки толщиной 120 мкм и представляет собой прямоугольный пакет 1 с размерами 1580х880, внутри которого располагается система трубок 2, соединённых со штуцером 3, который крепится к одной из стенок пакета в центральной его части. Пакет образован путём шовной ультразвуковой сварки по контуру двух слоев плёнки и нахлесточными швами 4 и 5, при этом внутренний шов обеспечивает герметичность ёмкости, внешний придает плёночной конструкции необходимую жёсткость. В верхней части ёмкости имеется петля крепления её на траверсе при транспортировании и фиксации ее в шкафу газоанализатора.

Основными требованиями, предъявляемыми к изделию, работающего под избыточным давлением 1,5 МПа и, следовательно, к сварным соединениям, является герметичность и прочность швов по всей их длине, работающих на расслаивание.

Разработанный технологический процесс состоит из следующих операций: 1 – раскрой плёнки, 2 – сварка продольных внутренних и наружных рантовых швов, 3 – сварка поперечных внутренних и наружных рантовых швов, 4 – контроль на герметичность швов, 5 – устранение дефектов сварных швов и материала изделия.

Перед сваркой полотнища заготовок необходимо очищать от загрязнения и пыли и обезжиривать ацетоном, затем их необходимо закрепить в сварочном приспособлении, обеспечивающем распрямление полотен изделий и фиксации рабочих зон для сварки.

На машине УПШ – 19 были проведены эксперименты по определению оптимальных технологических параметров процесса

ультразвуковой сварки эластичных ёмкостей из ПЭТФ – пленок. Влияние параметров оценивали по относительной прочности сварных соединений, выраженных в процентном (%) отношении к прочности основного материала в зависимости от сварочного давления (Рсв.) и скорости сварки при фиксированных значениях амплитуды рабочего торца волновода- инструмента.

Анализ результатов проведённых экспериментов позволил выявить некоторую особенность. Толщина ПЭТФ – плёнки весьма значительная и, как уже говорилось выше, составляет 120 мкм. В то же время, при толщине от 100 мкм она, являясь высокоориентированной, приобретает достаточную жёсткость. Вследствие этого, при ультразвуковой сварке ПЭТФ – плёнки требуется более значительное усилие для обеспечения необходимого акустического контакта в системе «торец ультразвукового волновода- инструмента – свариваемые плёнки – опора». А «добротная», то есть остро настроенная ультразвуковая колебательная система на повышенных частотах (40 кГц) достаточно чувствительна к внешним механическим

усилиям, что приводит к выводу её из резонансной частоты, а, следовательно, к значительному снижению амплитуды рабочего торца волновода- инструмента.

Для обеспечения устойчивой эффективной её работы возможны варианты – либо применение акустической обратной связи, либо использование менее «добротной» колебательной системы с относительно пологой амплитудно-частотной характеристикой. Применение акустической обратной связи приводит к усложнению конструкции малогабаритной акустической системы, то есть необходимо наличие датчика обратной связи [3,6].

Был принят второй вариант, при этом как, показали измерения, зависимость амплитуды торца волновода- инструмента от его длины имеет экстремальный характер (рис. 6), причём амплитуда рабочего торца волновода, длина которого короче резонансной, снижается более стремительно, чем при большей его длине. Исходя из этого длина волновода – инструмента была выбрана длиннее резонансной на 3.4 мм.

Анализ амплитудно-частотных характеристик (рис. 7), остронастроенной, «добротной» 1 и менее «добротной» 2 – с удлинённым рабочим волноводом – инструментом акустических систем при создании сварочного давления равного 1,2 МПа, показал, что частота менее «добротной» колебательной системы изменялась на несколько десятков Гц и амплитуда колебаний рабочего торца волновода снижалась весьма незначительно- на 2..3 мкм.

Как уже сказано выше, в процессе ультразвуковой сварки требуется создать сварочное давление величиной необходимой и достаточной для обеспечения акустического контакта, и создания условий протекания диффузионных процессов в локальной зоне контакта свариваемых пленок, что приводит к значительной деформации и появлению концентраторов напряжения и, как следствие, к снижению эксплуатационных характеристик сварного соединения. При получении сварного соединения таким концентратором является сварной шов и околошовная зона (ОШЗ).

Требования, предъявляемые к сварным соединениям указанных изделий, является не только герметичность сварного шва по всей его протяженности, но и прочностные характеристики. Снижение оптимальных значений сварочного давления способствует уменьшению деформации сварного шва и околошовной зоны и, следовательно, к уменьшению концентраторов напряжений. На рис. 8 представлены результаты экспериментов по определению области оптимальных значений параметров процесса сварки – скорости сварки и величины сварочного давления на прочность сварных соединений. Повышение скорости перемещения плёнки к уменьшению тепловыделения в объеме плёнки, расположенной со стороны опоры, и, соответственно, к снижению прочностных характеристик сварного соединения. Величина сварочного давления ограничивается условием нормальной работы акустической системы сварочной головки, которая

с превышением определённых значений приводит к выходу её из резонанса.

При сварке поперечных швов в зонах их пересечения с продольными швами необходимо снижать скорость сварки до 0,7-0,8 от номинальной, что необходимо для плавного перехода через уже сваренный шов и надёжного проплавления на этом участке шва.

После сварки этих сварных швов устанавливается трубчатая система подвода и отбора рабочего газа из всего её объема и производится наложение завершающего сварного шва, после чего ёмкость заполняется рабочим газом и подвергается контролю сварных соединений путём повторной сварки дефектного места на сварочной установке.

Разработанная технология, ультразвуковое сварочное оборудование и технологическая оснастка могут быть весьма полезными для предприятий, занимающимися вопросами изготовления пневматических плёночных конструкций из ориентированных ПЭТФ и других трудносвариваемых полимерных пленок.

7 Выводы.

1. В работе разработана технология и оборудование для шовной ультразвуковой сварки полиэтилентерефталатной плёнки.

2. С целью повышения прочностных характеристик сварных соединений были разработаны и изготовлены ультразвуковые сварочные устройства и рабочие волноводы – инструменты со сферической формой рабочего торца для сварки ПЭТФ пленок.

3. Были исследованы выходные характеристики разработанных ультразвуковых акустических систем и определенны максимальные значения амплитуд колебаний волноводов - инструментов от частоты ультразвукового генератора.

4. Проведён анализ амплитудно-частотных характеристик акустических систем.

5. Определены зависимости прочности сварных соединений ПЭТФ пленок от частоты, сварочного давления, скорости сварки и амплитудных и ультразвуковых колебаний.

8 Список используемой литературы.

1. Волков С.С. Сварка и склеивание полимерных материалов: учебное пособие для вузов. – М.: Химия, 2001 г., 376 с.

2. Волков С.С., Черняк Б.Ю. Сварка пластмасс ультразвуком, 2-е издание, переработанное и дополненное. – М.: Химия, 1986 г., 254 с.

3. Зайцев К.И., Мацюк Л.Н., Богдашевский А.В и др. Сварка полимерных материалов: справочник. – Под общей редакцией Зайцева К.И., Мацюк Л.Н – М.: Машиностроение, 1988 г., 312 с.

4. Гуль В. Е., Акутин М. С. Основы переработки пластмасс. — М.: Химия, 1985 г., 399 с.

5. Каменев Е.И., Мясников Г.Д., Платонов М.П. Применение пластических масс. справочник. Л.: Химия, 1985 г., 448 с.

6. Донской А.В., Келлер О.К., Кратыш Г.С. Ультразвуковые электротехнологические установки, 2-е издание, переработанное и дополненное. Л.:Электроиздат,1982 г., 208 с.

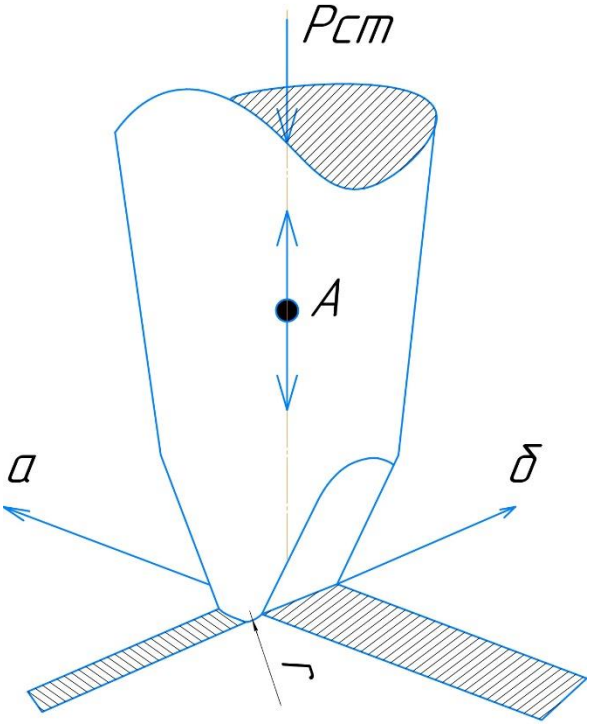


Рис .1 Сферическая заточка рабочего торца волновода-инструмента

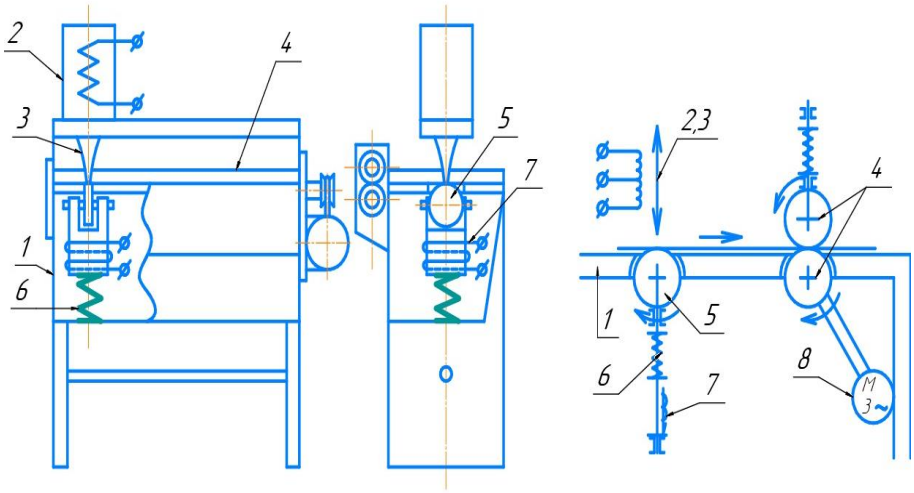


Рис. 2 Схема ультразвуковой шовной сварочной машины УПШ-19

1-станикна; 2- преобразователь; 3-волновод; 4-протягивающие ролики; 5-ролик опора; 6-пружина; 7-электромагнит опоры; 8-электродвигатель

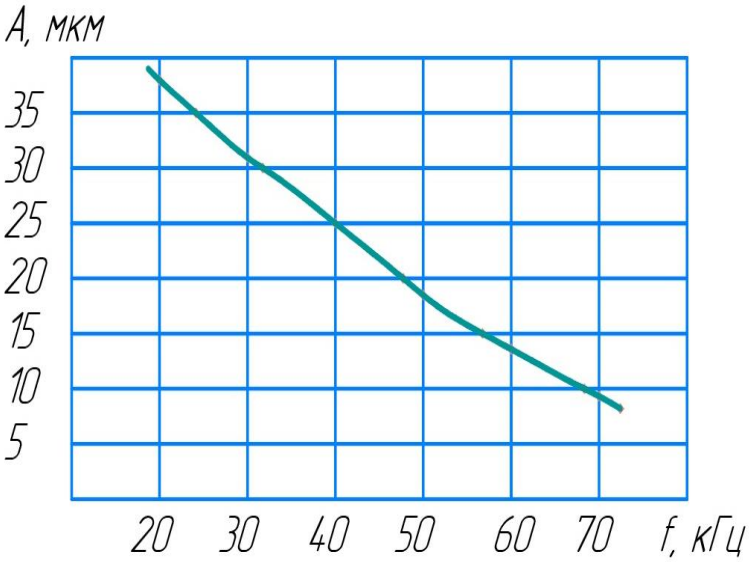


Рис. 3 Зависимость амплитуды ультразвуковых колебаний волноводов-инструментов от частоты ультразвуковых генераторов

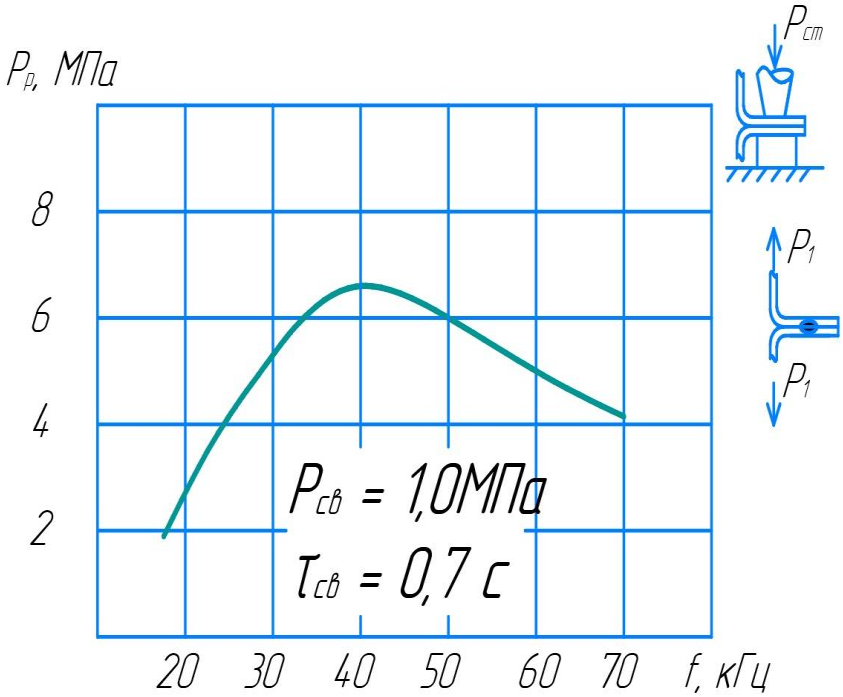


Рис.4 Влияние частоты ультразвуковых колебаний на прочность сварных соединений ($P_{ст}$=1 МПа, $t_{св}$=0,7 с)

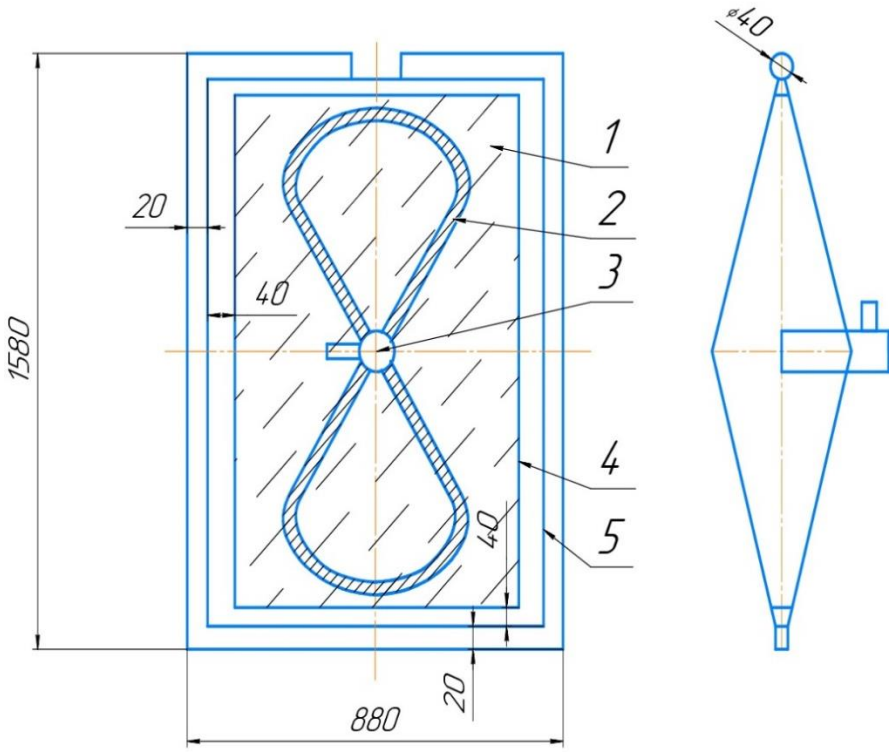


Рис. 5 Эластичная емкость из ПЭТФ-пленки.

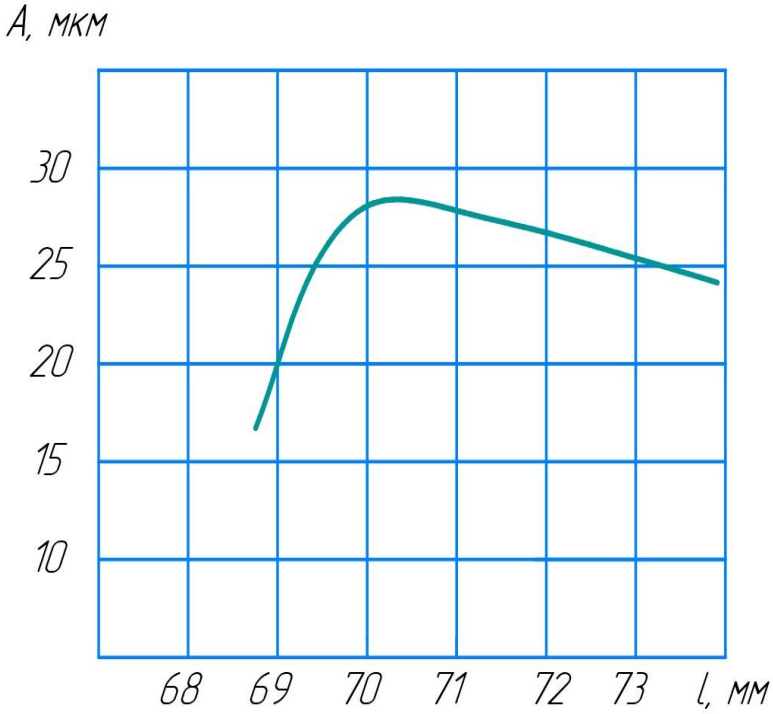


Рис. 6 Зависимость амплитуды колебаний рабочего торца волновода-инструмента

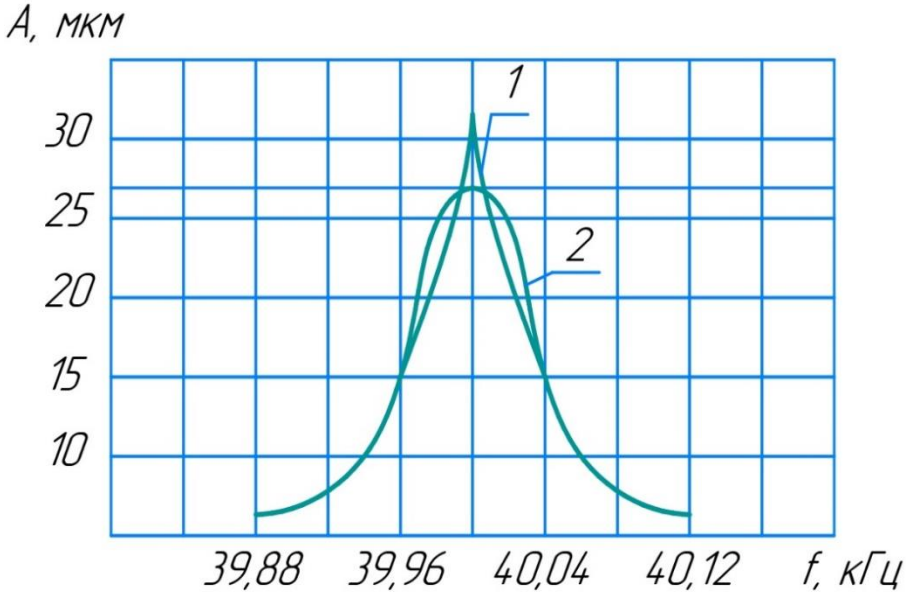


Рис. 7 Амлитудно частотные характеристики акустических систем.

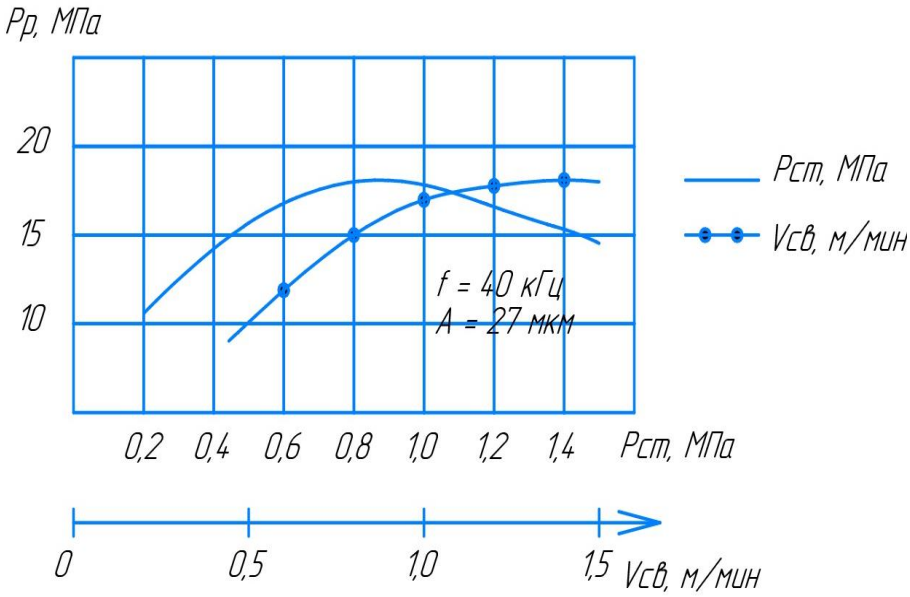


Рис.8 Зависимости прочности сварных соединений ПЭТФ – пленок от сварочного давления и скорости сварки.

СЕКЦИЯ III. Информатика и системы управления

Графические возможности программного пакета LabVIEW

Коробова Екатерина Дмитриевна

г. Москва, ГБОУ школа №1517, 11 класс

Научный руководитель:

Петросян Олег Гарегинович, с.н.с., доцент,

Каф. ИУ-3

В работе использованы возможности программного редактора LabVIEW для визуального представления графиков функций различной сложности. Программный пакет LabVIEW - это доступный графический редактор с набором встроенных функций, которые доступны для самостоятельного изучения в старших классах средней школы и, в частности, для вычисления площади, находящейся под кривой $y = F(x)$ и длины самой линии, ограниченной аргументами x_{min} и x_{max}, с использованием для этой цели встроенной единичной функции. Таким образом, представленный проект позволит заинтересованным учащимся познакомиться с основами численного интегрирования и некоторыми другими элементами из курса высшей математики.

Содержание

Введение

1. Краткое описание работы LabVIEW
2. 1.1 Лицевая панель и блок-диаграмма
3. 1.2 Проводники данных
4. 1.3 Плавающие палитры
5. 1.4 Контекстное меню
6. 1.5 Окно контекстной помощи
7. 1.6 Встроенные функции блок-диаграммы
8. Описание работы схемы
9. Заключение

Список использованных источников

Введение:

LabVIEW – это среда разработки виртуальных приборов (в дальнейшем LV). LV – является средой программирования, с помощью которой можно создать приложение. При реализации «алгоритма приложения» используется графическое представление всех элементов алгоритма. С помощью LV возможно создание необходимых типов виртуальных приборов. В LV используется язык графического программирования, алгоритм создаётся в графической иконной форме, образуя блок-диаграмму, что позволяет исключить множество синтаксических деталей.

Применяя этот метод, можно сконцентрировать внимание лишь на программировании потока данных. Основополагающий для LV принцип потока данных (dataflow) однозначно определяет порядок исполнения алгоритма, и согласно которому функции выполняются лишь тогда, когда они получают на вход необходимые данные.

Программы LV называются виртуальными приборами, т. к. они функционально и внешне подобны реальным приборам.

1.1 . ВП состоит из двух основных частей:

Лицевая панель (Front Panel) представляет собой интерактивный пользовательский интерфейс ВП и названа так, потому что имитирует лицевую панель традиционного прибора. На ней могут находиться ручки управления, графические индикаторы и другие *элементы управления* (controls), которые являются средствами ввода данных со стороны пользователя, и *элементы индикации* (indicators) – выходные данные из программы.

Пользователь вводит данные, используя мышь и клавиатуру, а затем видит результат действия на мониторе.

Блок-диаграмма (Block Diagram) является исходным программным кодом ВП, созданного на языке графического программирования LV. Блок-диаграмма представляет собой реально исполняемое приложение LV. Компонентами блок-диаграммы являются *виртуальные приборы более низкого уровня, встроенные функции* LV*, константы и структуры управления* выполнением программы. Для того, чтобы задать поток данных между определёнными объектами и создать связь между ними, необходимо нарисовать соответствующие проводники (wires). Объекты на лицевой панели представлены на Блок-диаграмме в виде соответствующих терминалов, через которые данные могут поступать от пользователя в программу и обратно.

Окно Блок-диаграммы содержит исходный графический код ВП. Блок-диаграмма LV соответствует строкам текста в обычных языках программирования. Конструирование Блок-диаграммы осуществляется путём соединения между собой объектов, выполняющих определённые функции.

Элементы управления и индикаторы: лицевая панель в основном состоит из совокупности элементов управления и индикаторов. Элементы управления имитируют типичные органы управления, которые имеются у любого измерительного прибора (кнопки, переключатели и т. д.). ЭУ позволяют пользователям ввести данные, которые передаются в Блок-диаграмму. Индикаторы отображают выходные данные, являющиеся результатом выполнения программы. ЭУ – это данные, вводимы пользователем, терминалы – источники данных. Индикаторы - данные, выводимые пользователю - приёмники данных. ЭУ и Индикаторы на лицевой панели выбираются из подпалитры плавающей палитры ЭУ и размещаются в нужном месте. После появления объекта на лицевой панели можно поменять его размеры, цвет, местоположение и другие атрибуты.

При использовании ВП в качестве подпрограммы в Блок-диаграмме другого ВП необходимо определить его иконку и соединительную панель. Виртуальный прибор, который применяется внутри другого ВП, называется виртуальным подприбором, который аналогичен подпрограмме в традиционных алгоритмических языках. Иконка является однозначным графическим представлением ВП и может использоваться в качестве объекта на Блок-диаграмме другого ВП. Соединительная панель представляет собой механизм передачи данных в ВП из другой Блок-диаграммы, когда он применяется в качестве подприбора – ВПП. Подобно аргументам и параметрам подпрограммы, соединительная панель определяет входные и выходные данные виртуального прибора. Подприбор имеет клеммы входа и клеммы выхода, так что его удобно использовать в качестве готового инструмента в любой программе.

***Терминалы данных*:** при помещении ЭУ на лицевую панель LV автоматически создаёт на Блок-диаграмме соответствующий терминал. Терминал нельзя удалить с Блок-диаграммы. Он исчезнет лишь тогда, когда будет удалён соответствующий элемент управления или индикатор с лицевой панели.

Узлы данных: это обобщающее название любого исполняемого элемента программы. Узлы аналогичны операторам, функциям, подпрограммам в традиционных языках программирования. Функции «*сложить» и «вычесть»* представляют один вид узла. Другим видом узла является *структура* (Structure), которая может выполнять код циклически или по условию. LV содержит и специальные типы узлов, например узел Формула (Formula Node), предназначенный для работы со сложными математическими формулами и выражениями.

1.2. Проводники данных:

ВП в LV представляет собой единое целое за счёт проводников данных, соединяющих узлы и данные. Проводники являются каналами прохождения данных от терминала-источника к одному или нескольким терминалам-приёмникам. При попытке присоединить к проводнику более одного источника или вообще ни одного источника будет генерироваться «повреждение» проводника.

Принцип соединения источников и приёмников проводником объясняет, почему управляющие элементы и индикаторы не могут заменить друг друга. Управляющие элементы –это источники, а индикаторы- приёмники данных. Каждый проводник имеет свой стиль и цвет в зависимости от типа данных, проходящих по нему.

1.3. Плавающие палитры:

LV имеет три часто используемые палитры, которые помещаются в любом удобном месте на экране: палитры *Инструменты, Элементы управления и Функции*. На палитре ЭУ выбирают *Элементы Управления и Индикаторы* для расположения их на лицевой панели. Если провести курсором по палитрам ЭУ и Ф, можно заметить, что название подпалитры появляется в верхней части окна. Можно осуществить поиск любого специфического объекта, щёлкая мышью по иконке с изображением лупы, а также отредактировать собственные палитры, щёлкая мышью по кнопке *Опции*. Закрепление палитры осуществляется щелчком мышью по изображению *кнопки - thumbtack,* расположенной в верхнем левом углу палитры.

1.4. Контекстное меню:

Практически каждый объект LV имеет контекстное меню опций и команд. Вызов контекстного меню осуществляется правой

кнопкой мыши. Например, если текущий инструмент *Цвет,* то при вызове контекстного меню появляется цветовая палитра. Большинство объектов контекстного меню распадается на подменю, которые называются иерархическими и которые обозначаются стрелкой «вправо». Выбранная текущая опция обозначается галочкой при работе с текстом или окружена прямоугольником для графики. Некоторые объекты меню имеют диалоговые окна, содержащие опции, которые можно настраивать. Объекты меню, ведущие к диалоговым окнам, показаны в виде *(...)* многоточий. Объекты меню без троеточий и без стрелок являются командными и выполняются сразу после выбора. В названии команды обычно записано, что будет выполнено при её выборе. Например, *Заменить индикатор (Change indicator).* После выбора некоторые команды заменяются в меню на противоположные. Например, после реализации команды Заменить на индикатор появляется команда *Заменить на элемент управления (Change to Control).*

Видимые объекты: многие объекты имеют меню *Видимые элементы (Visible Items),* с помощью которых можно показать или скрыть определённые элементы оформления, например, ярлыки, заголовки, полосы прокрутки или соединительные терминалы. При выборе *Видимые элементы* получаем другое меню, в котором перечисляются все элементы, которые могут быть показаны (этот перечень меняется в зависимости от объекта). И если элемент имеет рядом галочку, то он является видимым, если галочки нет, то он скрыт. Для изменения статуса элемента надо щёлкнуть кнопкой по соответствующей опции.

Опции Описание и Подсказка: выбор опции позволяет описать объект и сделать подсказку к нему.

Опция Создать: облегчает создание узла данных, локальной переменной или ссылки на данный объект.

Опция Заменить: позволяет входить в палитры ЭУ и Ф (в зависимости от того с чем мы работаем: с лицевой панелью или блок диаграммой) и заменить выбранный объект другим по желанию.

Опция Операция с данными:

- ***Установить в значение по умолчанию (Reinitialize to Default)*** возвращает объект к его значению по умолчанию, тогда как ***Сделать текущую величину значением по умолчанию (Mace Current Value Default)*** устанавливает текущие данные в качестве значения по умолчанию.

- **Опции** ***Удалить данные (Cut Data), Копировать данные (Copy Data) и Вставить данные (Past Data)*** используются для извлечения или ввода данных в элементы управления или индикатора.

Опция Дополнительно: элементы опции дают доступ к некоторым редко используемым параметрам, которые употребляются для тонкой настройки поведения элементов управления и индикаторов.

- ***Управление клавишами (Key Navigation***) применяется для создания комбинации клавиш с целью вызова объекта лицевой панели. При вводе в действие этой комбинации в процессе работы ВП LV действует так, как будто мы щёлкнули мышью по объекту и курсор становится активным в поле данного объекта.

- ***Синхронное отображение (Suncronous Display)*** – элемент, который заставляет LV обновить изображение элемента управления или индикатора при вводе/выводе новой информации.

- ***Настройка (Customize***) выводит редактор элементов управления для настройки графического представления ЭУ.

- ***Скрыть ЭУ/И (Hide Control/Indicator)*** и наоборот ***Показать ЭУ/И (Show Control/Indicator)***.

1.5. Окно контекстной помощи:

Окно контекстной помощи (Context Help Window) LV предлагает необходимую информацию по функциям, константам, подприборам, ЭУ и И. Для того, чтобы задействовать окно, выбираем опцию *Show Context Help* из меню справка. При удержании курсора на функции, узле подприбора или иконки ВП, в окне контекстной помощи появляется иконка для этой функции или ВВП с проводниками соответствующего типа данных. Проводники входных данных направлены влево, а проводники выходных данных – вправо. Для некоторых подпрограмм или функций окно контекстной помощи показывает имена обязательных входных данных жирным шрифтом, а значения по умолчанию заключает в скобки. Имеется возможность блокировки окна помощи. Если поместить инструмент соединения на каком-нибудь вводе функции или ВВП, то в окне будет мигать точка на соответствующем входе. Переключение между типами просмотра осуществляется путём нажатия кнопки *Простой/Детальный просмотр (Simple/Detailed Diagram Help)* в нижнем левом углу окна помощи.

1.6. Встроенные функции Блок-диаграммы:

ВП из палитры генерации сигналов и шумов (рис.1) используются для формирования детерминированных и случайных сигналов с заданным набором параметров. Из этой палитры я использую следующие функции: единичный импульс *(Impulse Pattern)* и треугольник *(Triangle Pattern).*

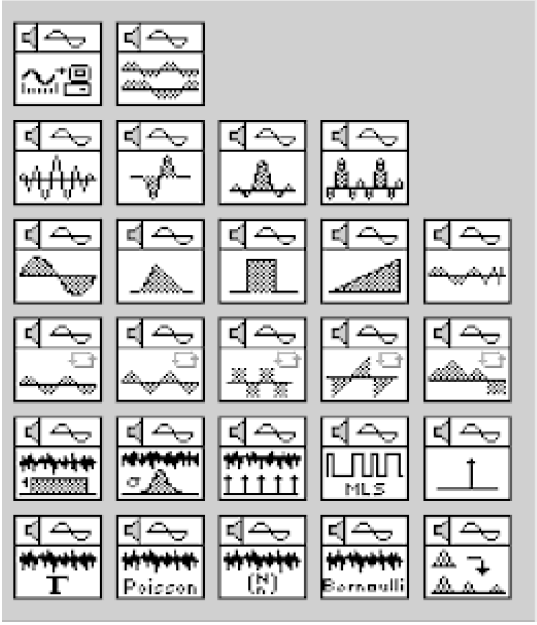


Рис.1 Палитра функций генерации сигналов и шумов

В графическом редакторе в подпалитре *Signal Processing>>Signal Generation>>**Impulse Pattern*** имеется единичная функция (рис.2), использование которой с помощью ряда других функций LabVIEW позволит провести численное интегрирование.

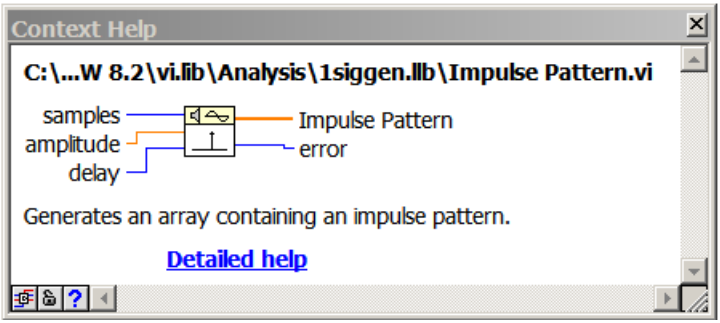


Рис.2 Иконка встроенной функции *Impulse Pattern*

Вход *задержка* (delay) задаёт задержку единичной функции Impulse Pattern по длительности импульса относительно начала сигнала. С помощью структуры For Loop начальное положение импульса можно перемещать с дискретностью $\Delta x=1$.

Выход *импульсного сигнала* (Impulse Pattern) – это массив, у которого все элементы равны 0, кроме элемента n_i , на абсциссе которого в данный момент будет находиться импульс. Численное значение этого элемента равно 1. Очевидно также, что по мере смещения импульса индекс единичного элемента массива будет приближаться к N-1, т. е. станет последним элементом массива после завершения программы.

Точность численного интегрирования с использованием единичной функции зависит от числа итераций. С ростом числа итераций растёт и точность конечного результата. Здесь также используется тот факт, что нижний предел интеграла равен 0.

Числовые функции и функции манипуляции данными: числовые функции используются для выполнения арифметических и комплексных операций с числовыми данными, а также для преобразования типов числовых данных. На рис. 3 показан вид основной палитры числовых функций (рис. 3а) и дополнительных подпалитр. Преобразование (рис. 3б), Функции манипуляции данными (рис. 3в), Комплексные (рис. 3г) и Математические и научные константы (рис. 3д).

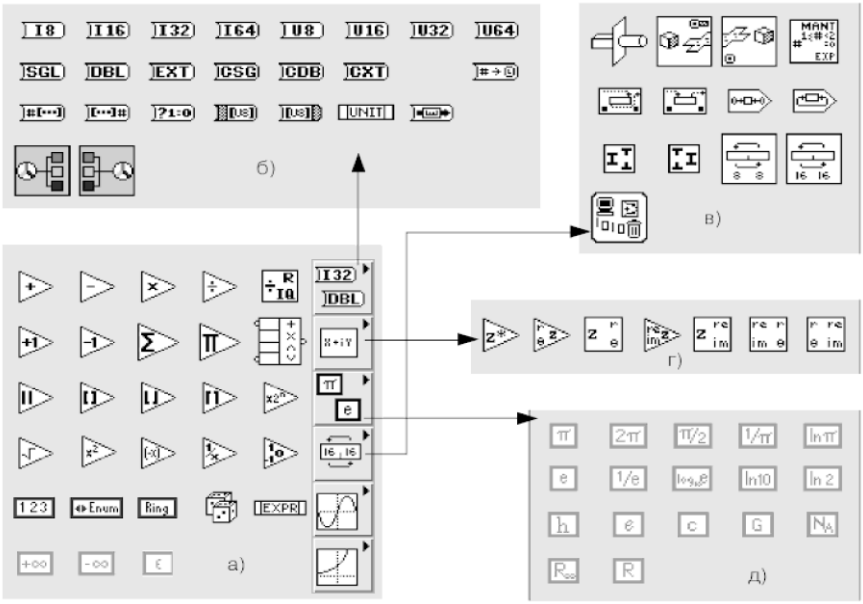


Рис.3 Вид основной палитры (а) и дополнительных подпалитр (б)–(д) числовых функций структуры

В составе подпалитры *Структуры* среды LabVIEW можно выделить две группы структур, имеющих ряд общих черт. Первая группа включает *Цикл* с фиксированным числом итераций *(For Loop)*, с которым я работаю в программе. Помимо этого в состав палитры также входит структура *Формула (Formula Node)*. Структура *Цикл* с фиксированным числом итераций *(For Loop)* создает массив данных от i = 0 to i=N-1, N – число итераций. Количество циклов задается с помощью константы N, которая находится в левом верхнем углы структуры *(For Loop).*

8. Описание работы схемы:

Описание работы схемы для вычислений площади и длины полуокружности:

В качестве примера использования единичной функции была поставлена задача измерения площади полуокружности и длины ее дуги. Для этого располагаю на Блок-диаграмме структуру For Loop и задаю уравнение полуокружности y=(1-x**2)$^{1/2}$. Аргумент x должен изменяться от -1 до +1. Графическое изображение полуокружности представлено на рис.4. Управляющим является алгоритм реализации создания полуокружности, который находится в нижней структуре рисунка 5.

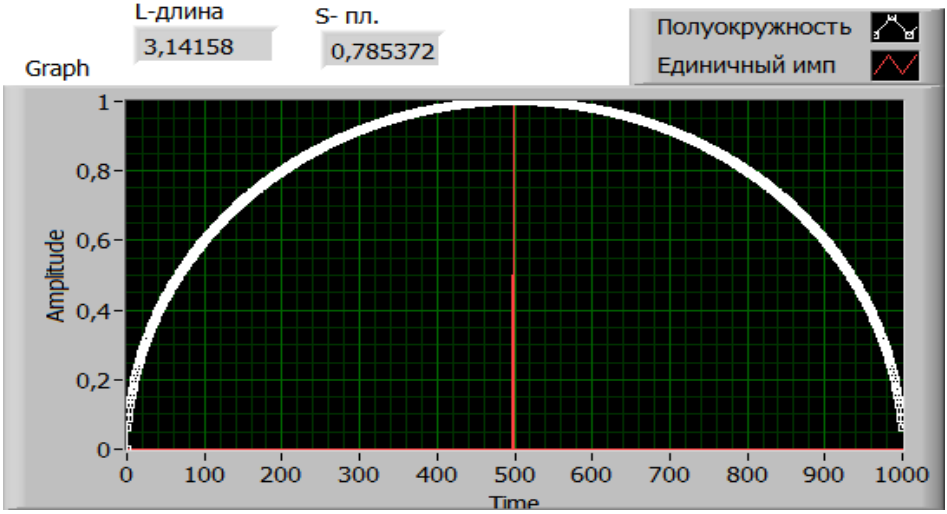
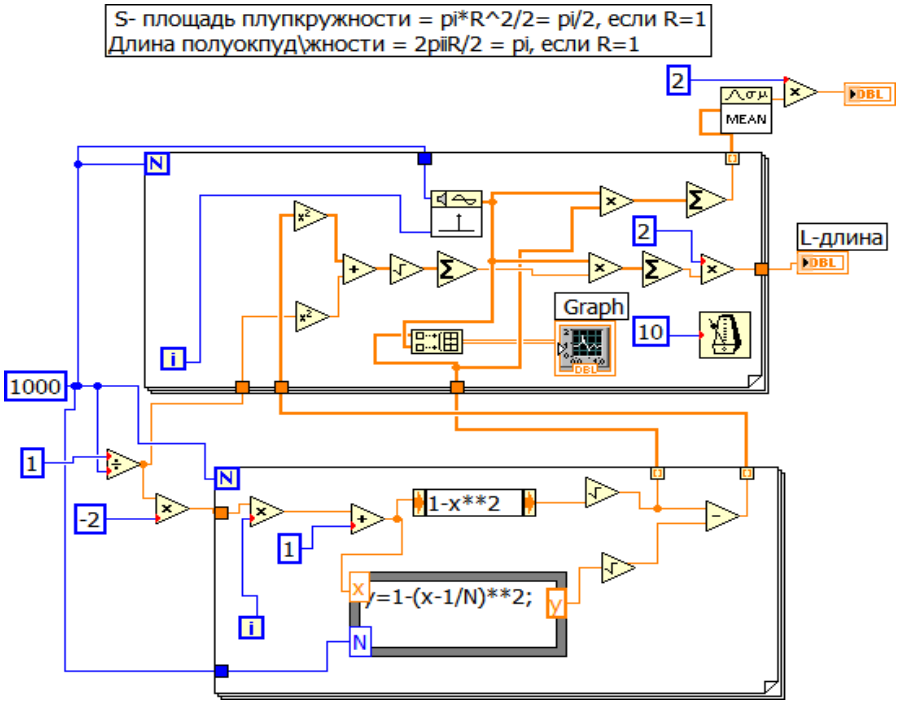


Рис.4 Фронт панель с изображением полуокружности и единичного



импульса

Рис.5 Блок-диаграмма измерения площади и длины полуокружности

В структуру for loop помещаю функцию Expression Node, с вписанной в неё формулой полуокружности и Formula Node, куда вписывается та же формула окружности с смещение в виде 1/N. Далее извлекаю квадратные корни и вывожу полученные потоки данных в другую структуру for loop, нахожу разность между ординатами $\Delta y = y_{i+1} - y_i$, возвожу её в квадрат и, суммируя квадраты катетов треугольника ABC вычисляю гипотенузу $AB = (\Delta y^2 + (1/N))^2$ (рис.6). Результат умножаю на единичный импульс. При прохождении единичного импульса вдоль оси абсцисс он пересекается с линией полуокружности с дискретностью $\Delta x_i = 1/N$, где N – частота дискретизации (Samples) и фиксирует отрезок полуокружности с помощью гипотенузы AB.

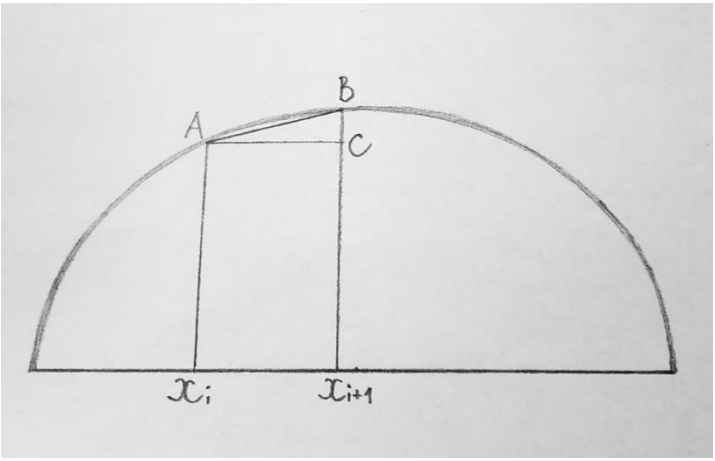
Результаты суммируются, и можно найти длину полуокружности. Здесь надо отметить следующий момент. Полученный результат надо привязать к какой-нибудь метрике. Например, если на графике строится окружность с единичным радиусом, то основание, на которое опирается полуокружность – есть 2R.

Рис. 6 Наглядная схема для вычисления длины окружности

Для вычисления площади полуокружности перемножаю ординаты с единичным импульсом, закрепляю и складываю выведенные значения на сумматоре, нахожу среднюю высоту, которую перемножаю на диаметр.

Результаты измерений приведены на индикаторах Lokr и Sokr

Описание работы схемы для вычислений площади и длины треугольника:



Измерение площади треугольника и длины боковых сторон данного треугольника провожу по той же схеме, что и в предыдущем случае с полуокружностью. Треугольник вписываю в полуокружность,

как это показано на рис. 7 и по схеме блок-диаграммы представленной на рис. 8 с помощью единичной функции провожу измерения.

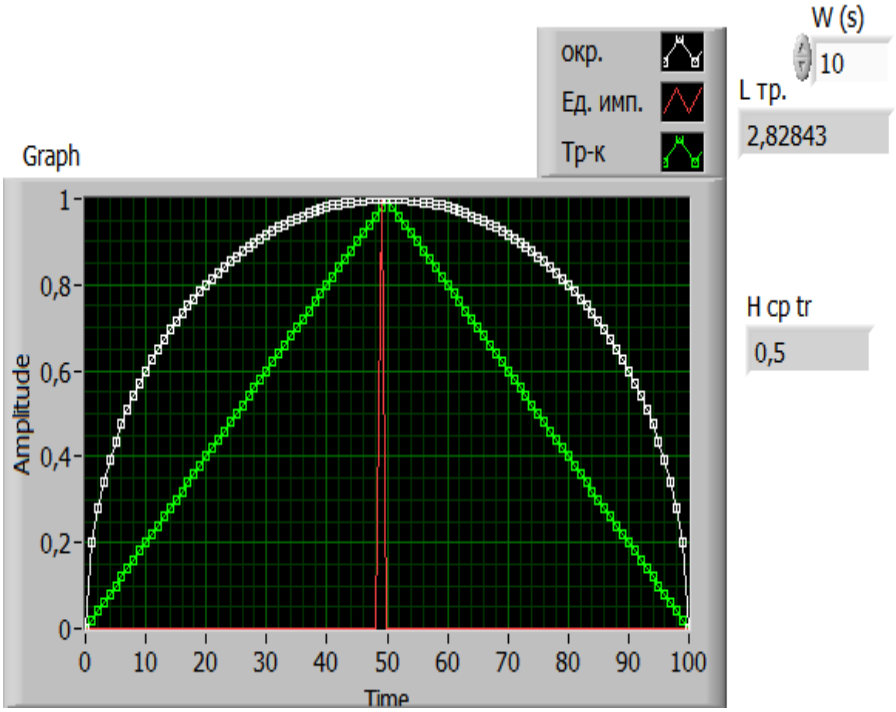


Рис.7 Фронт панель с изображением треугольника, полуокружности и единичного импульса

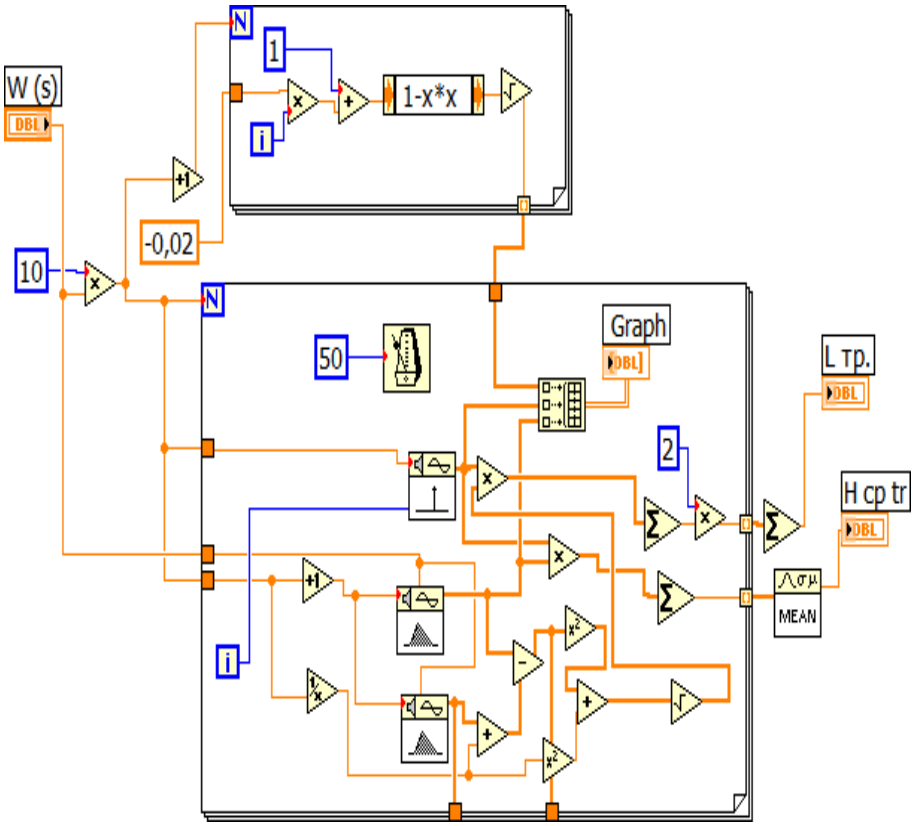


Рис.8 Блок-диаграмма для вычисления площади треугольника и длины боковых сторон равнобедренного треугольника

В блок-диаграмме (рис. 8) две структуры for loop, одна из которых строит окружность, другая служит для запуска единичного импульса и считывания числовых характеристик, полученных из массивов треугольников. При этом числовые характеристики одного из треугольников смещены на величину 1/N. Частота дискретизации привязывается к ширине основания треугольника. Схема построена

таким образом, чтобы изменение ширины основания треугольника приводило одновременно к изменению всех параметров схемы. Разность числовых массивов, снятых с треугольников возводится в квадрат и суммируется с квадратом смещения, извлекается квадратный корень и результат перемножается с единичным импульсом. В точке пересечения с дискретными значениями x_i абсциссы снимаются численные значения ординаты y_i и запоминаются и суммируются.

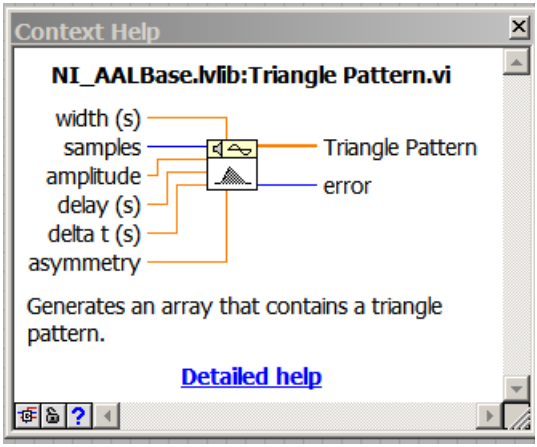


Рис. 9 Развёрнутая схема управления треугольником из подпалитры Signal Processing.

Асимметрия равнобедренного треугольника по умолчанию 0,5;
Width – ширина основания треугольника, устанавливаем -10;
Amplitude, delay, delta по умолчанию: соответственно 1, 0, 0,1.

Исходя из принятой системы, безразмерного исчисления основание треугольника ABC полагаю равным AC = 2R, а высота H= R. Полагая R=1, площадь треугольника $S=2R*R/2=R^2$. Боковые стороны треугольника $AB=AC=R\sqrt{2}$ безразмерных единиц, что совпадает с полученными результатами, которые проведены на схеме (Str и Ltr). Очевидно, что для повышения точности можно увеличить частоту

дискретизации, однако для нашего случая частота дискретизации=1000 достаточно точный результат до 5ого знака.

Заключение:

Таким образом, используя встроенную единичную функцию в LabVIEW, реализуется возможность измерения площади полуокружности и длины ее дуги, а также площади и длины дуги треугольника. Данная работа может использоваться в качестве наглядного пособия для ознакомления школьников с элементами численного интегрирования.

Список использованных источников:

1. Суранов А. Я. LabVIEW 8.2: Справочник по функциям. – Москва, 2007.
2. Дж. Трэвис Дж. Кринг LabVIEW для всех (3 издание, переработанное и дополненное) – Москва, 2008.

СЕКЦИЯ III. Информатика и системы управления

Автономный контроллер дистанционного мониторинга и управления с функцией осциллографа

Расюк Александр Александрович
г. Москва, ГБОУ «Лицей № 1580», 11 класс
Научный руководитель:
Зуйков Василий Васильевич,
Преподаватель дополнительного образования
ГБОУ г. Москвы «Школа №1580 при МГТУ им. Н.Э. Баумана»

В работе рассматривается автономный контроллер дистанционного мониторинга и управления с функцией осциллографа. Устройство состоит из измерительного модуля и модуля мониторинга, соединенных посредством беспроводного канала связи Bluetooth, обеспечивающего автономность при использовании. Для конвертирования и вывода информации используется микроконтроллер Atmega328. В качестве устройства сбора информации выступает отдельная платформа, содержащая микроконтроллер Atmega328, электронный щуп (сигнал с которого считывается с помощью АЦП), Bluetooth модуль и литий-полимерный аккумулятор. Так же для сбора дополнительной информации использованы отдельные автономные датчики температуры, влажности, освещенности. Данное устройство может быть

использовано для измерения показателей электронного оборудования в режиме реального времени при проектировании или настройке.

СОДЕРЖАНИЕ

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ВВЕДЕНИЕ

1 ОБЩЕТЕХНИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ УНИВЕРСАЛЬНОГО УСТРОЙСТВА СБОРА ИНФОРМАЦИИ С БЕСПРОВОДНЫХ ДАТЧИКОВ

1.1 Анализ существующих устройств сбора данных с датчиков

1.2 Принципы преобразования физических величин в цифровые значения

1.3 Методы беспроводной передачи информации

1.4 Концепция реализации универсального устройства сбора информации с беспроводных датчиков

Вывод по главе 1

2 МАТЕМАТИЧЕСКИЕ МЕТОДЫ И МОДЕЛИ СБОРА ИНФОРМАЦИИ

2.1 Преобразование полученных значений

2.2 Анализ методов фильтрации сигнала

2.3 Анализ существующих средств представления информации

Вывод по главе 2

3 РАЗРАБОТКА УНИВЕРСАЛЬНОГО УСТРОЙСТВА СБОРА ИНФОРМАЦИИ С БЕСПРОВОДНЫХ ДАТЧИКОВ

3.1 Проектирование устройства

3.2 Макетирование устройства

3.3 Разработка управляющего кода

3.4 Изготовление печатных плат

Вывод по главе 3

4 ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ УНИВЕРСАЛЬНОГО УСТРОЙСТВА СБОРА ИНФОРМАЦИИ

4.1 Тестирование датчиков температуры, освещенности

4.2 Испытательный стенд для измерения ускорения

4.3 Испытательный стенд для тензометрического датчика

Вывод по главе 4

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ 1 - 4

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ADC	Analog-to-digital converter	АЦП	Аналого-цифровой преобразователь
DAC	Digital to analog converter	ЦАП	Цифро-аналогового преобразователя
FFT	Fast Fourier transform	БПФ	Быстрое преобразование Фурье
FHSS	Frequency-hopping spread spectrum	ППРЧ	Псевдослучайная перестройка рабочей частоты
GPS	Global Positioning System	-	Система глобального позиционирования
GSM	Groupe Special Mobile	-	Глобальный стандарт цифровой мобильной сотовой связи
ISM	Industry, Science and Medicine	-	Промышленность, наука и медицина
LCD	Liquid crystal display	ЖКД	Жидкокристаллический дисплей
PC	Personal computer	ПК	Персональный компьютер

PIC	Peripheral interface controller	-	Контроллер интерфейса периферии
SPS	Samples per second	-	Частота дискретизации
USB	Universal Serial Bus	-	Универсальная последовательная шина
VGA	Video Graphics Array	-	15-контактный аналоговый разъём
-	-	МЧС	Министерство РФ по чрезвычайным ситуациям
-	-	СМИК	Система мониторинга инженерных конструкций

ВВЕДЕНИЕ

Работа посвящена разработке и изготовлению универсального устройства сбора информации с беспроводных датчиков: температуры, влажности, освещенности, ускорения, сжатия/растяжения. Отдельное внимание уделено созданию осциллографа с беспроводным щупом, спроектированного в качестве автономного измерительного устройства.

Актуальность. В настоящее время важным является сбор актуальных показателей электронно-технических устройств в режиме реального времени. Неотъемлемой частью современных технологий выступает обработка и конвертирование информации в удобный для восприятия вид. Для создания и обслуживания различной аппаратуры как в производственном, так и бытовом секторе используются устройства контроля множества характеристик. В частности, для диагностики электронного оборудования применяются такие приборы как: вольтметры, амперметры, осциллографы, частотомеры. А сбор данных о температуре, влажности и освещённости применяется в большом количестве областей - начиная от производства электроники и заканчивая сферой хозяйственной деятельности.

В настоящее время производителями электронного оборудования представлен большой выбор осциллографов:

- аналоговые;
- цифровые;
- портативные.

Наибольшее распространение получили цифровые осциллографы, так как современные технологии позволяют сделать их относительно недорогостоящими. Не зависимо от фирмы производителя, практически каждый осциллограф имеет цветной дисплей, два сигнальных канала, чувствительность 0,002-0,5 В, частоту пропускания 40-60 МГц. Также производят большое количество высокоточных осциллографов, стоимость которых достаточно велика.

Компания **Batronix** предлагает надежное измерительное устройство -цифровой запоминающий осциллограф **Owon SDS6062V** [1]. Цветная TFT LCD панель с диагональю 8 дюймов (20,3 см) и разрешением 800 × 600 обеспечивает отличную читаемость графика, особенно когда применяют режим наложения нескольких сигналов. В комплект поставки входят 2 щупа, USB-кабель и программное обеспечение для ПК.

Так же существует вариант «V», в котором предложена дополнительная функция - выход VGA, использующийся для подключения полноценного компьютерного монитора.

Осциллографом с особыми характеристиками может послужить **IkaScope WS200** [2] от **IKALOGIC** - это полнофункциональный осциллограф, со встроенным датчиком и подключением к WiFi. IkaScope имеет небольшие габариты и работает до одной недели без подзарядки, так же работа от батареи обеспечивает

гальваническую развязку от сети. Он подключается к компьютеру, смартфону или планшету для отображения измеряемых сигналов.

Благодаря частоте дискретизации 200 МГц, ширине полосы 30 МГц и входному диапазону 80 В, IkaScope идеально подходит для быстрой диагностики электронных систем и предназначен для инженеров, работающих со встраиваемыми системами.

Технология **ProbeClick®** автоматически удерживает захваченные сигналы при отпускании наконечника зонда. А программное обеспечение, работающее на настольном компьютере, ноутбуке или мобильном устройстве, позволяет легко хранить и обмениваться данными измерений и обеспечивает сохранение автоматической истории.

Для измерения температуры и влажности используют портативные или стационарные метеостанции, которые имеют выносные (подключаются посредством кабеля или по радиоканалу) или встроенные датчики. Стоимость данных устройств не велика, но их невозможно использовать в качестве управляющего контроллера, так как они подразумевают только вывод значений на дисплей.

Цифровой термометр **KEYNICE** с датчиком на кабеле длиной 1,5 м имеет корпус с пазами для установки заподлицо. В качестве особенностей можно выделить: переключатель единиц измерения °F/°C (расположен на задней панели устройства), авто обновление показания температуры каждые 2,5 секунды, антикоррозийный датчик (изготовлен из нержавеющей стали), подсветка дисплея.

Промышленный термогигрометр **Testo 645** автоматически отображает параметры относительной влажности, абсолютной влажности, температуры точки росы, уровня влаги и температуры. Диапазон измеряемых показателей очень велик: для температуры -200…+1370°C, при погрешности 0.1°C. Возможен выбор различных зондов (разная длина кабеля, с защитой от влаги и без, разная рабочая температура), что делает устройство более универсальным, но влечет резкое увеличение цены.

Отдельно можно выделить измерительные устройства компании **ZETLAB**, специализирующейся на изготовлении автоматизированных стационарных система мониторинга инженерных конструкций (СМИК). СМИК представляет собой набор подсистем, отвечающих за контроль параметров в реальном режиме времени, направленных на предупреждение перехода объекта мониторинга в ограниченно работоспособное или аварийное состояние, которое может повлечь за собой гибель людей.

СМИК позволяет обнаружить на ранней стадии изменения несущей способности грунтового основания, а также конструктивных элементов и своевременно информировать персонал дежурно-диспетчерской службы объекта о критическом изменении параметров состояния несущих конструкций объекта. Система мониторинга инженерных конструкций может включать в себя как все подсистемы мониторинга (ZETLAB предоставляет большое количество

измерительного оборудования: виброакустическое, тензометрическое, сейсмическое, термометрическое, датчики давления, наклономеры), так и их произвольный набор, отвечающий требованиям заказчика. Данную систему можно использовать для измерения ускорений на аттракционах, диагностики лифтов, сейсморазведочных работ, диагностики зданий и сооружений по требованиям МЧС, мониторинга мостов. Количество точек и вид мониторинга зависит от типа конструкции объекта и выбирается на этапе проектирования каждой подсистемы расчетным способом, на основе проектной документации.

Также ZETLAB на базе компании ЭТМС организовала серверную база для хранения, вычисления и обработки результатов измерений с объектов наблюдения и дальнейшей передачей готовых значений непосредственно клиенту.

Исходя из характеристик существующего оборудования можно выделить следующие недостатки:

- отсутствие единого контролера для разных задач;
- недостаток автономности;
- реализация дополнительных функций, требует приобретения отдельного устройства.

Постановка задачи: спроектировать и изготовить макет автономного контроллера дистанционного мониторинга и управления с функцией осциллографа.

Цель работы: разработать универсальное устройство сбора информации с датчиков, имеющее минимальные масса-габаритные

характеристики, обладающее автономностью и легко адаптируемое под конкретные области применения.

В рамках работы были решены следующие задачи:

- анализ имеющихся на рынке прототипов;
- исследование принципа работы АЦП;
- анализ схемы работы АЦП;
- разработка принципиальной схемы и конструкции контроллера дистанционного мониторинга и управления с функцией осциллографа;
- разработка программного обеспечения для микроконтроллера;
- анализ существующих способов беспроводной передачи информации;
- изучение и использование Bluetooth передатчиков.

Методы проведения исследований: для изучения основных методов сбора информации с датчиков были рассмотрены существующие аналоги, так же в ходе работы вносились изменения в проектируемое устройство, чтобы учесть появившиеся недостатки. После конечной сборки, был проведен ряд экспериментов, направленных на установление точности датчиков.

Структура и объем работы: работа состоит из четырех основных разделов.

В первом разделе рассматриваются существующие методы и готовые устройства сбора информации. Выделены основные достоинства и недостатки аналогов. Сформирована основная архитектура.

Во втором разделе описаны принципы математических преобразований, используемые в конвертировании сигнала с датчиков в удобный для пользователя вид.

В третьем разделе дано описание проектирования, моделирования, изготовления и проверки работоспособности устройства, а также показан окончательный вид и функционал.

В четвертом разделе изложены методы апробации и тестирования датчиков с использованием испытательных стендов.

На основе вышеизложенных разделов сделано заключение о результате проделанной работы.

Общий объем работы: 46 страниц, 18 рисунков.

Результаты работы: в ходе работы был разработан и изготовлен цифровой осциллограф с диапазоном измерений -12…12 В и частотой до 20 МГц с дополнительными датчиками: температуры (-40…+80°C, точность 0.5°C), влажности (0…100 %, точность 4 %), освещенности, ускорения и малых деформаций. Габариты устройства 14x11 см (главный модуль), 5x6 см (модули датчиков). Возможно, как подключение внешнего источника питания, так и автономная работа от встроенного аккумулятора. Поддерживается Bluetooth канал связи, с радиусом действия 6–12 м. Реализован графический интерфейс,

включающий вкладки датчиков (отображение графиков и максимальных/минимальных значений) и настройки. Для удобства использования добавлена функция авто определения подключенных датчиков.

1 ОБЩЕТЕХНИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ УНИВЕРСАЛЬНОГО УСТРОЙСТВА СБОРА ИНФОРМАЦИИ С БЕСПРОВОДНЫХ ДАТЧИКОВ

1.1 Анализ существующих устройств сбора данных с датчиков

Осциллографы, пробники, тестеры

Вольтметры, амперметры, осциллографы, частотомеры служат неотъемлемыми приборами электронной промышленности. Осциллографы можно выделить в отдельную группу, так как они способны частично, а иногда и полностью, решать задачи остальных вышеприведенных приборов. Характеристики осциллографов очень сильно отличаются в зависимости от класса устройства (см. таблица 1.1).

Таблица 1.1 – Особенности осциллографов

Особенности	**Преимущества**	**Недостатки**
Малогабаритные цифровые осциллографы	Удобство в использовании	Урезание функционала
Многоканальные осциллографы	Одновременное отслеживание	Увеличение стоимости

	нескольких сигналов	
Сенсорные графические дисплеи	Понятный, простой интерфейс	Увеличение габаритов, стоимости
Функция записи сигнала	Удобство в отслеживании сигнала в течении продолжительного времени	Отсутствуют
Дополнительные выходы для подключения к ПК или монитору	Улучшение интерфейса	Сложность в поддержании форматов/стандартов современных ПК

Из выше приведённой таблицы становиться ясно, почему двухканальные электронные осциллографы с дисплеем и кнопками управления (не сенсорными) получили наибольшее распространение – они имеют достаточно широкий функционал при малой стоимости.

Комбинированные измерители температуры и влажности

Широкое распространение получили электронные термометры со встроенными датчиками влажности, так как они позволяют точно определять главные характеристики воздушной среды. На данный момент существуют различные конфигурации данных устройств (см. таблица 1.2), которые можно разделить на условные категории.

Таблица 1.2 – Классификация электронных термометров

Особенности	**Преимущества**	**Недостатки**
Датчики встроены в устройство	Малые габариты, простота использования	Невозможность удаленных измерений
Выносные датчики	Удаленные измерения	Наличие проводов или радиоканал (зачастую приводит к резкому увеличению цены)
Измерение одного или двух значений	Простота использования (для бытовых целей)	Отсутствие вариативности, нет возможности расширить функционал
Графический дисплей	Удобный формат информации	Отсутствуют

Высокая точность датчиков	Подходит для промышленных целей	Резкое увеличение цены

Таким образом, решено объединить особенности разных устройств в одном для компенсации недостатков одних за счет других: при использовании выносных датчиков можно уменьшить размер управляющего блока, но оставить в нем дисплей, а многоканальность обеспечить за счет радиоканала, не создавая лишних проводных соединений.

1.2 Принципы преобразования физических величин в цифровые значения

Общим для всех вышеперечисленных примеров является конвертирование аналоговых сигналов, полученных с датчиков, в цифровые показатели.

Аналого-цифровое преобразование (АЦП) [3] – это процесс преобразования входной физической величины в ее числовое представление. Для получения точной информации о каком-либо физическом процессе необходимо установить зависимость между измеряемой величиной и величиной, изменение которой легко зарегистрировать. Входной величиной АЦП может быть любая

физическая величина – напряжение, ток, сопротивление, емкость, частота следования импульсов и т.п.

Однако, в данном случае, под АЦП представляют преобразователь напряжение-код, так как в цифровых датчиках температуры, влажности, освещенности используют резисторы, сопротивление которых пропорционально измеряемой характеристики.

АЦП имеет множество характеристик, из которых основными можно назвать частоту преобразования и разрядность. Частота преобразования обычно выражается в отсчетах в секунду (samples per second, SPS [4]), разрядность – в битах. Современные АЦП могут иметь разрядность до 24 бит и скорость преобразования до единиц GSPS. Чем выше скорость и разрядность, тем труднее получить требуемые характеристики, тем дороже и сложнее преобразователь. Скорость преобразования и разрядность связаны друг с другом определенным образом, и можно повысить эффективную разрядность преобразования, пожертвовав скоростью.

Типы АЦП:

- АЦП параллельного преобразования (прямого преобразования);
- АЦП последовательного приближения;
- дельта-сигма АЦП (АЦП с балансировкой заряда).

Наибольшим быстродействием и самой низкой разрядностью обладают АЦП прямого (параллельного) преобразования. Из-за одновременной работы всех компараторов достигается высокая

скорость обработки, но разрядность ограничивается их количеством. Например, АЦП параллельного преобразования TLC5540 фирмы Texas Instruments [5] обладает быстродействием 40MSPS при разрядности всего 8 бит.

Среднюю нишу в ряду разрядность-скорость занимают АЦП последовательного приближения. В данном методе величину входного сигнала последовательно сравнивают с рядом генерируемых величин. Распространенным значением является разрядность 12-18 бит при частоте преобразования 100KSPS-1MSPS. Многие микроконтроллеры AVR содержат встроенный АЦП последовательного приближения, имеющие в восьмиразрядном режиме 256 отсчетов. Достоинством является относительно высокая скорость.

Наибольшей точности достигают сигма-дельта АЦП, имеющие разрядность до 24 бит включительно и скорость от единиц SPS до единиц KSPS, но принцип действия данного метода несколько более сложен, чем у других типов АЦП.

1.3 Методы беспроводной передача информации

Не маловажным выступает передача данных с датчиков, расположенных в разных, возможно удаленных, местах на главное устройство. На данный момент распространёнными являются следующие технологии передачи информации, принцип действия которых основан на использовании радиоволн:

- Wi-Fi [6];

- Bluetooth [7];

- радио модули.

Радиосвязь Bluetooth осуществляют в ISM диапазоне (англ. Industry, Science and Medicine [8]), который используют в различных бытовых приборах и беспроводных сетях (свободный от лицензирования диапазон 2,4-2,4835 ГГц). В Bluetooth применяют метод расширения спектра со скачкообразной перестройкой частоты (англ. Frequency Hopping Spread Spectrum, FHSS [9]). Метод FHSS прост в реализации, обеспечивает устойчивость к широкополосным помехам, а оборудование недорогое.

В качестве канала связи выбран Bluetooth модуль в силу его достоинств: хорошая устойчивость к широкополосным помехам, низкое энергопотребление и простота реализации.

1.4 Концепция реализации универсального устройства сбора информации с беспроводных датчиков

Исходя из преимуществ и недостатков описанных выше методов работы с датчиками, разработана концепция универсального устройства. Основной задачей является объединение датчиков широкого спектра измерений, достижение автономности использования и реализация удобного интерфейса. Для достижения данных характеристик решено разделить устройство на несколько отдельных частей: блок управления с крупным дисплеем и отдельные модули датчиков, работающие независимо.

Архитектура всей системы представлена на рисунке 1.1, главного модуля на рисунке 1.2 и обобщенная для разных датчиков на рисунке 1.3.

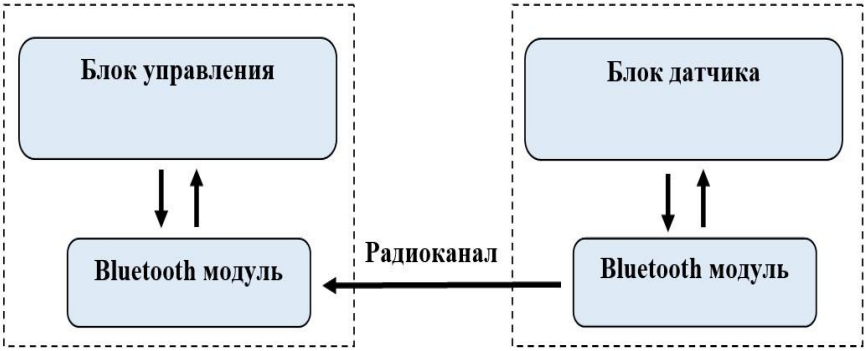


Рисунок 1.1 – Общая архитектура

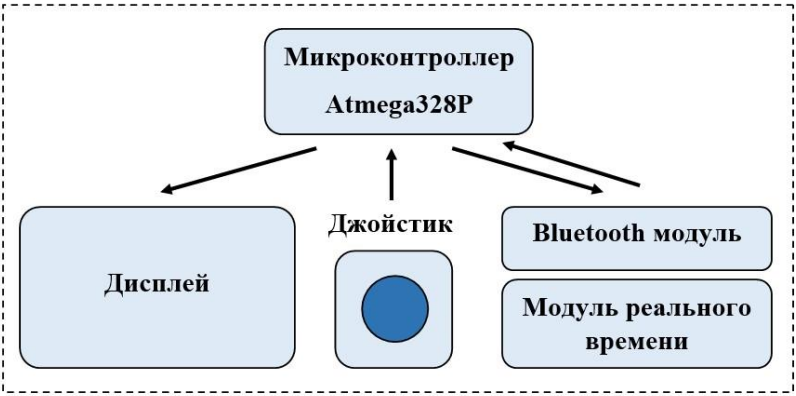


Рисунок 1.2 – Блок управления

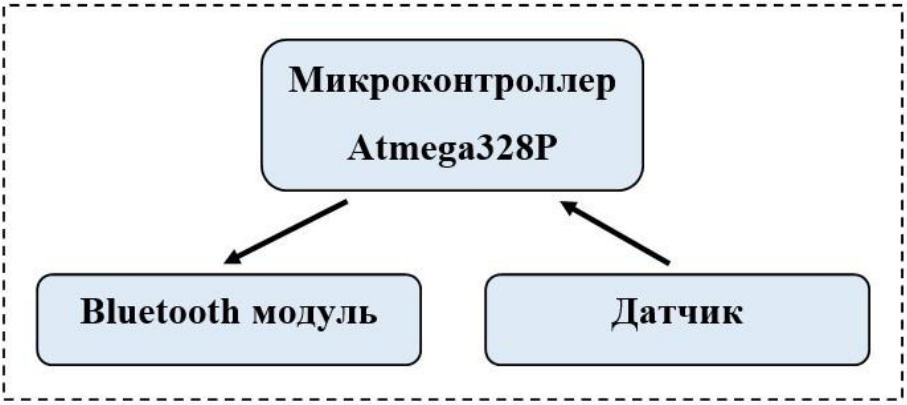


Рисунок 1.3 – Обобщенная архитектура для разных датчиков

Таким образом, достигнута модульность, за счет которой возможно быстрое изготовление и настройка дополнительных датчиков, что позволяет мониторить большое количество характеристик или одинаковые показатели в разных частях контролируемого объекта (если он достаточно велик).

Вывод по главе 1

Проанализировав существующие устройства сбора информации, можно увидеть, что существует большое количество различных решений, направленных на конкретные задачи, но не всегда пригодных для широкого спектра измерений или быстрой модификации под новую задачу. Так же мало распространенным является оборудование с высокой точностью измерений, но доступное для использования непрофессионалами. По ряду вышеперечисленных причин ставится задача по созданию устройства, обладающего характеристиками точного измерительного прибора с малыми габаритами и энергозатратами. Для решения данной задачи можно использовать микроконтроллер Atmega328P [10], имеющий встроенный АЦП, и отдельный Bluetooth передатчик. Совокупность данных модулей позволяет сделать устройство автономным и легко перенастраиваемым. Использование распространённого Bluetooth канала расширяет функционал, так как возможно подключение к любым другим внешним устройствам, поддерживающим данный протокол.

2 МАТЕМАТИЧЕСКИЕ МЕТОДЫ И МОДЕЛИ СБОРА ИНФОРМАЦИИ

2.1 Преобразование полученных значений

Информация, собранная с датчиков, зачастую характеризует не саму измеряемую величину, а значение другой физической величины, измерение которой легкодоступно. Для получения нужных данных необходимо произвести математические вычисления по заранее известной зависимости прямых и косвенных величин.

Например, получая значения АЦП, можно узнать значение сигнала только относительно опорного напряжения. Для расчета настоящего значения можно воспользоваться простым преобразованием:

$$Vin = ADC * Vref / 1024,$$

где Vin – вычисляемое напряжение;

Vref – опорное напряжение;

ADC – значение, полученное с АЦП;

1024 – степень дискретизации (2^10 = 1024, для 10 битного АЦП).

Другим примером может послужить процесс преобразований для измерения температуры, влажности, малых деформаций. Эти методы можно объединить в одну группу, так как изменение этих величин влечет за собой изменение сопротивления резистора нужного датчика (терморезистор, резистивный датчик влажности, тензорный датчик). Для каждого типа измерения устанавливается линейная или логарифмическая зависимость.

Данные преобразования предоставляются производителями датчиков или реализованы в микроконтроллере, установленном на самом датчике. Также существует большое количество библиотечных программ, использование которых удобно при быстрой реализации устройства. Но для более точных измерений зачастую требуется вносить изменения в изначальные коэффициенты пропорциональности, так как любой датчик имеет отклонения от указанных технических параметров и требует калибровки.

2.2 Анализ методов фильтрации сигнала

Полученное значение с датчика всегда содержит некоторую погрешность относительно настоящей, измеряемой величины. Для компенсации отклонений можно применять как электронные аналоговые или цифровые фильтры, так и математические модели.

Фильтр Калмана [11] разработан для сглаживания полученных данных в физико-математической модели с известными параметрами. Но на практике очень часто бывает, что о физической модели, значения которой фильтруют, мало что известно.

К примеру, необходимо отфильтровать показания с акселерометра. Заранее неизвестно по какому закону будет перемещаться акселерометр. Максимум информации, которую можно узнать - это дисперсия ошибки сенсора. И соответственно составить зависимость по формуле

$$x_{k+1} = x_k + \varepsilon,$$

где x_{k+1} – новое значение;

x_k – предыдущее значение;

ε – погрешность.

Такая система не удовлетворяет тем условиям, которые налагаются на физику движения и случайную величину, поэтому нельзя говорить, что в этом случае выполняется теория фильтра Калмана. Но можно заметить, что коэффициент Калмана с увеличением номера шага всегда приближается к определенному значению. Поэтому вместо того, чтобы подбирать коэффициенты и находить их по точным формулам, можно считать этот коэффициент всегда константой, и

подбирать только эту константу. Это допущение почти ничего не изменит. И в итоге вычисления очень упростятся:

$$M_n = k * A_n + (1 - k) * M_{n-1},$$

где M_n - результирующее значение вычисления на шаге n;

k - коэффициент стабилизации;

A_n - исходное значение текущего измерения.

На рисунке 2.1 показаны отфильтрованные двумя разными способами данные с некоторого сенсора, при условии того, что ничего не известно о физике явления.

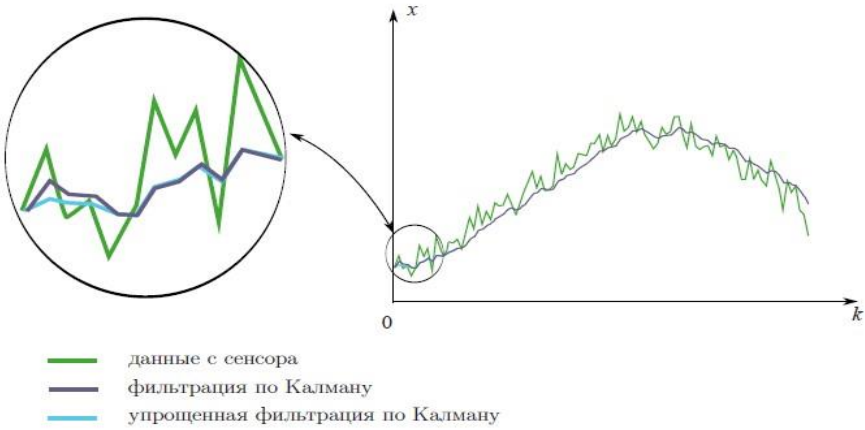


Рисунок 2.1 – Сравнение упрощенной и полной фильтрации по Калману

Первый способ - со всеми формулами из теории Калмана, второй - упрощенный. Таким образом, результаты почти ничем не отличаются. Маленькое отличие наблюдается только вначале, когда коэффициент Калмана еще не стабилизировался.

2.3 Анализ существующих средств представления информации

Так как в данном проекте для вывода информации применяют дисплей, то рассмотрены основные графические средства представления информации:

- график;
- гистограмма;
- таблица;
- числовое отображение.

Графики удобны, когда необходимо отслеживать не только значение, но и характер изменения этих значений во времени, именно поэтому графики применяют для отображения осциллограммы. На них

легко увидеть форму сигнала, что зачастую и является основной интересующей информацией.

Для отображения температуры и влажности удобно использовать гистограммы, так как эти величины изменяются только на протяжении достаточно большого времени (исключение могут составить, например, химические производства, на которых используют мощные нагреватели), поэтому удобно производить замеры через определенные промежутки времени и составлять картину изменений за 24 ч (или подходящий срок). Так же можно отдельно хранить максимальные и минимальные значения за весь период измерений.

В таблицах наглядно видны характеристики одной величины, собранные с разных датчиков, это можно использовать для сравнения показателей в разных комнатах одного помещения.

Вывод по главе 2

Вышеперечисленные методы преобразования информации достаточно легко реализуемы, они позволяют достичь желаемого результата (удобное отображение точных величин) с небольшими затратами как памяти, так и вычислительной мощности микроконтроллера, именно по этой причине они получили широкое распространение в микропроцессорной технике и были выбраны в качестве основных методов в данном проекте.

3 РАЗРАБОТКА УНИВЕРСАЛЬНОГО УСТРОЙСТВА СБОРА ИНФОРМАЦИИ С БЕСПРОВОДНЫХ ДАТЧИКОВ

3.1 Проектирование устройства

В качестве основной идеи архитектуры устройства выбрана модульность. Поэтому было решено отделить главное устройство с крупным дисплеем от датчиков по средствам Bluetooth модуля. Это сформировало общее представление о дальнейшей архитектуре. После были подобраны датчики, микроконтроллер, дисплей (128x64 пикселя) и дополнительные радиодетали. Основными характеристиками при выборе были малые габариты, легкодоступность, высокая точность и скорость работы.

3.2 Макетирование устройства

Изначально все блоки конечного устройства были собраны на беспаячной макетной плате (рисунок 3.1), проверена их работоспособность, внесены изменения в конфигурацию: установка модуля реального времени (для отслеживания даты/времени измерений), переход с управления кнопками на более удобный джойстик.

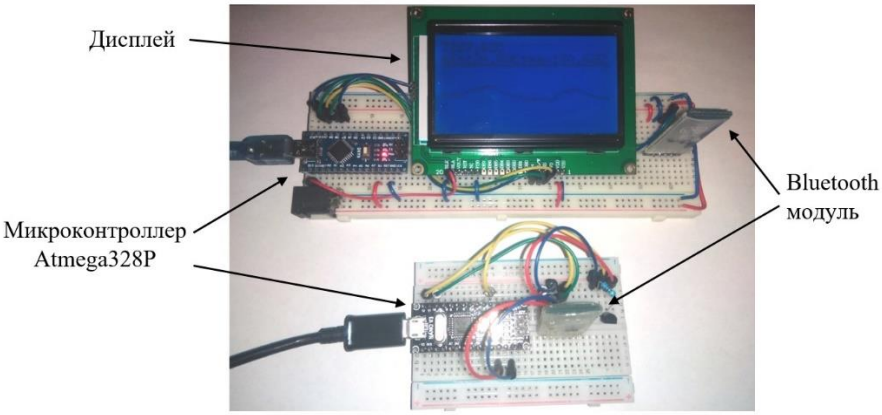


Рисунок 3.1 – Макет устройства

Во время макетирования были замечены некоторые неточности в работе устройства: отклонения показания датчиков, мерцание дисплея, несрабатывание кнопок. Это обусловлено плохим контактом соединений макетной платы.

3.3 Разработка управляющего кода

Для работы микроконтроллеров разработан код (приложение 1.1 – 1.4). Код можно разделить на две группы: управление модулями датчиков, управление главным устройством.

Основные части кода для работы с датчиками:

- конвертация полученной величины в значение измеряемой;

- формирование посылки данных с указанием типа передаваемого значения;

- отправка данных через определенный промежутки времени.

Основные части кода главного устройства:

- прием данных с канала Bluetooth;

- обработка информации (формирование массива данных, фильтрация, вычисление минимума/максимума);

- создание графического интерфейса, включающего возможность пользовательских настроек (характеристики способа отображения данных, настройка яркости/контрастности дисплея);

- обработка действий пользователя (управление джойстиком).

3.4 Изготовление печатных плат

Так как во время макетирования были исправлены недочеты схемы и выбрано удобное расположение главных элементов устройства, то было решено изготовить печатные платы [12] для достижения устойчивой работы системы.

Основные этапы изготовления печатной платы:

- разводка соединений [13] в трассировщике Sprint Layout (рисунок 3.2 – 3.5);

- печать графических дорожек на лазерном принтере;

- термоперенос тонера с бумаги на фольгированный стеклотекстолит с помощью прокатного ламинатора;
- травление персульфатом аммония (рисунок 3.6);
- удаление тонера;
- лужение сплавом розе в растворе глицерина (рисунок 3.7);
- сверловка отверстий;
- пайка электронных деталей.

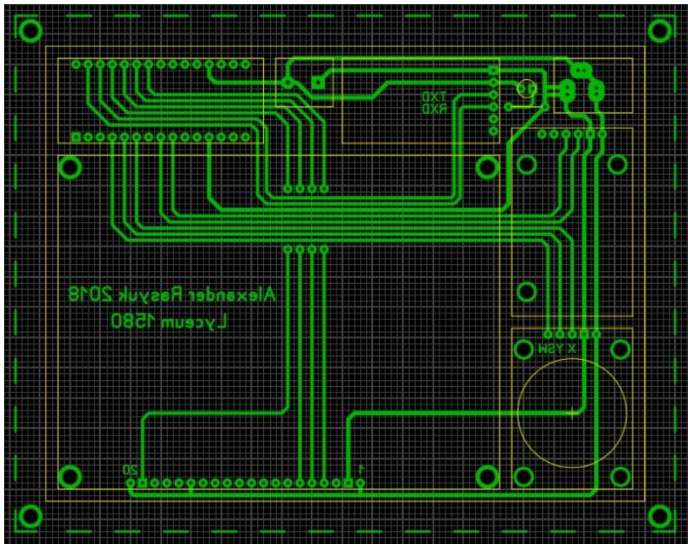


Рисунок 3.2 – Разводка управляющего блока

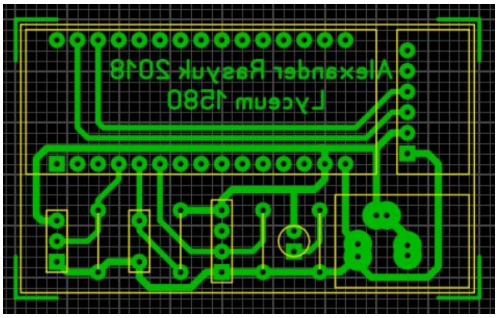


Рисунок 3.3 – Разводка модуля термодатчика

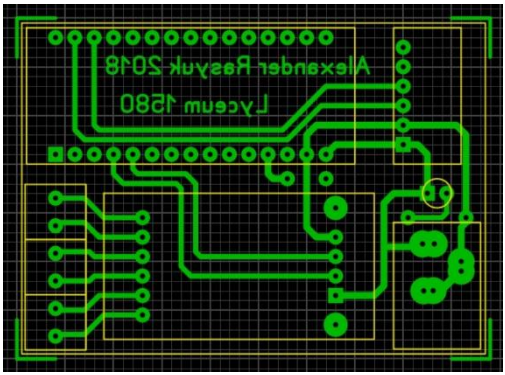


Рисунок 3.4 – Разводка модуля тензорного датчика

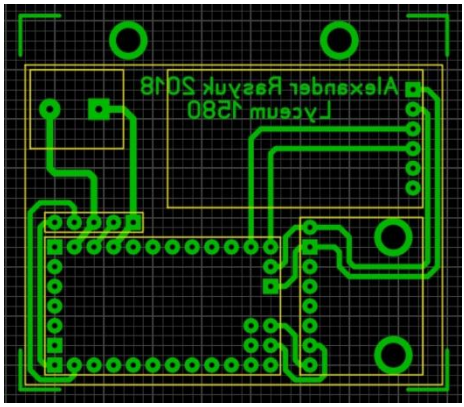


Рисунок 3.5 – Разводка модуля акселерометра

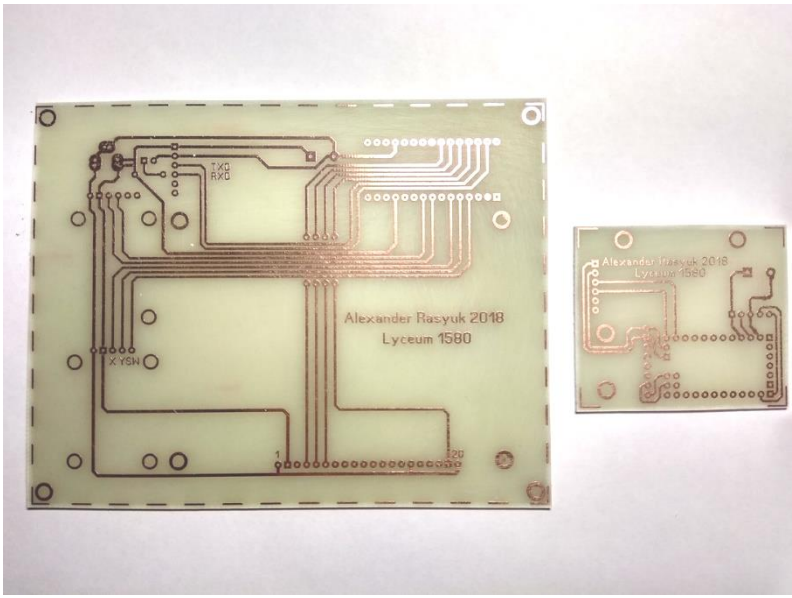


Рисунок 3.6 – Печатная плата после травления

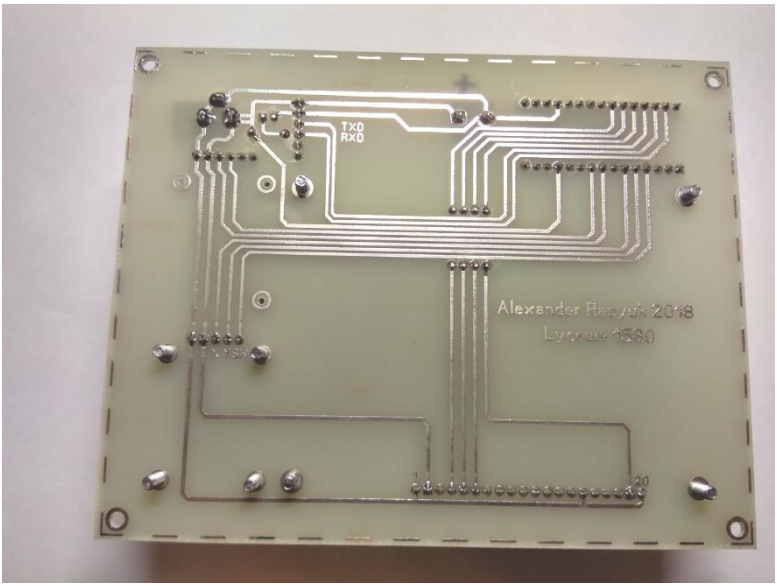


Рисунок 3.7 – Печатная плата после лужения

После изготовления плат произведено их тестирование и отладка по средству прозвонки. На рисунке 3.6 – 3.10 представлены собранные модули.

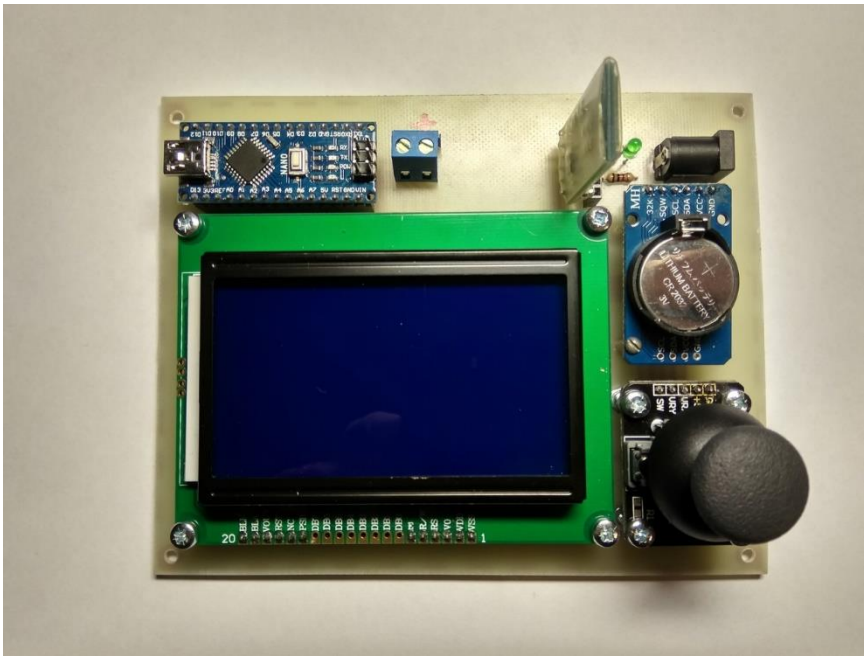


Рисунок 3.6 – Управляющий блок

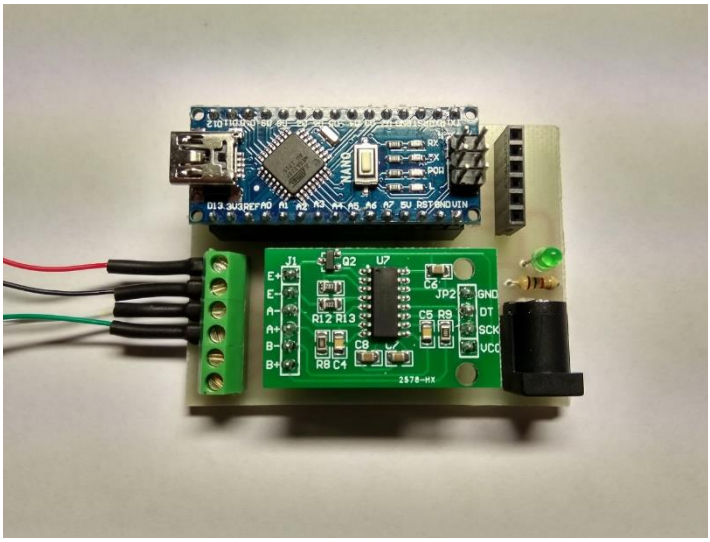


Рисунок 3.7 – Блок тензорного датчика

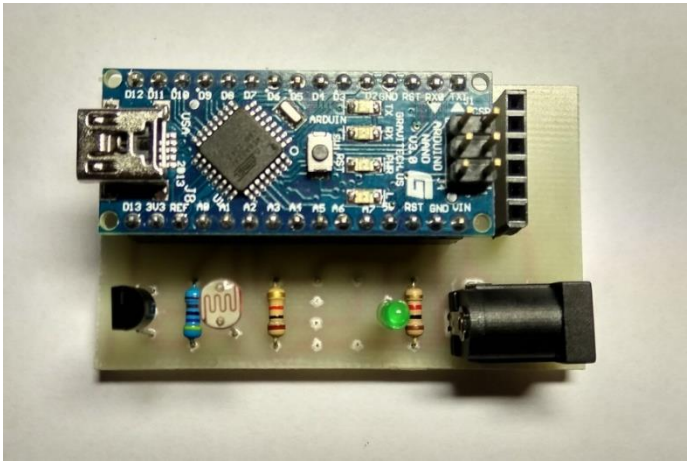


Рисунок 3.8 – Блок температурного датчика

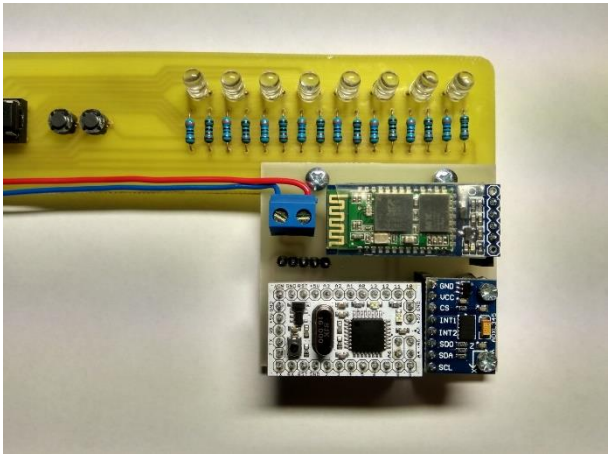


Рисунок 3.9 – Блок акселерометра

Тестирование печатных плат показало, что устранен дребезг контактов, наблюдаемый в макете. Так же появилась возможность жесткого закрепления блоков датчиков на испытательном стенде или установка в удобном месте помещения.

Вывод по главе 3

Таким образом, последовательные этапы планирования, моделирования, макетирования и изготовления устройства позволили учесть ошибки начальной архитектуры и привели к реализации готового, работоспособного прототипа.

4 ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ УНИВЕРСАЛЬНОГО УСТРОЙСТВА СБОРА ИНФОРМАЦИИ

4.1 Тестирование датчиков температуры, влажности, освещенности

Для проверки работоспособности и точности датчиков использовался готовый цифровой домашний термометр с функцией измерения влажности. Для сравнения была проведена серия экспериментов в разное время суток и в разных комнатах помещения, результаты представлены в таблице 4.1.

Таблица 4.1 – Результаты измерения температуры

		Температура, °С	
Комната	№	Готовый термометр	Изготовленный прототип
1	1	26,4	26,9
	2	25,8	26,5
	3	26,7	25,9
	4	25,9	26,4
	5	26,3	25,5

2	1	22,5	21,4
	2	22,7	24,0
	3	23,4	22,3
	4	21,9	21,8
	5	23,1	23,3
3	1	23,6	22,4
	2	24,5	25,7
	3	23,8	22,6
	4	22,9	23,6
	5	24,1	23,4

Из измерений вычислена средняя погрешность отклонений – 3%.

4.2 Испытательный стенд для измерения ускорения

В качестве испытательного стенда для датчика ускорения, было выбрано светодинамическое устройство (изготовлено в рамках предыдущих проектов, рисунок 4.1), в котором графическая информация формируется за счет вращения полоски светодиодов.

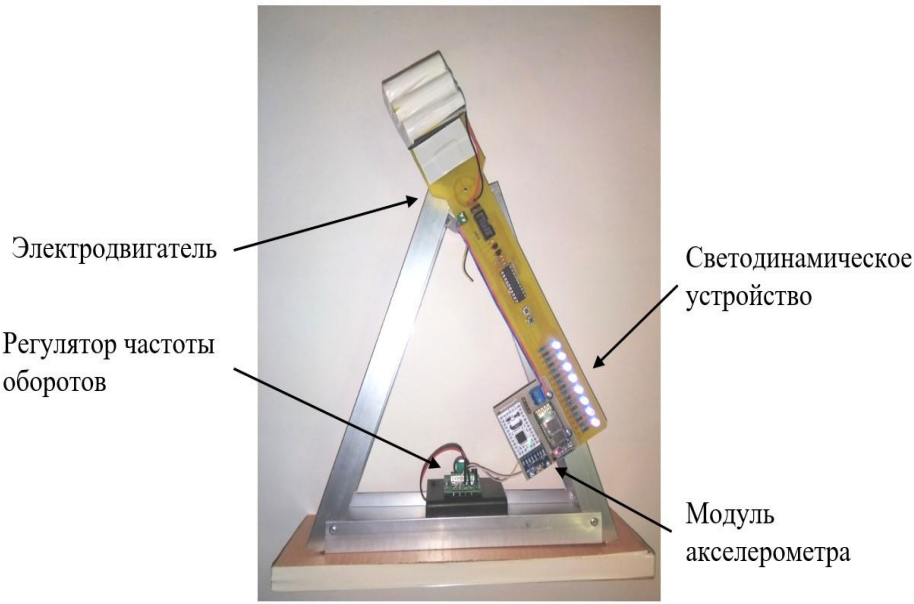


Рисунок 4.1 – Испытательный стенд для измерения ускорения

На данном устройстве был закреплен модуль с датчиком ускорения, в режиме реального времени снимались и отображались показания на дисплее. Во время вращения создается ускорение, которое в проекции на оси измеряется акселерометром.

4.3 Испытательный стенд для тензометрического датчика

Для тестирования тензометрического датчика взята алюминиевая балка, на которую эпоксидной смолой (необходима монолитная фиксация для работы с микродеформацией) наклеен датчик (рисунок 4.2, 4.3).

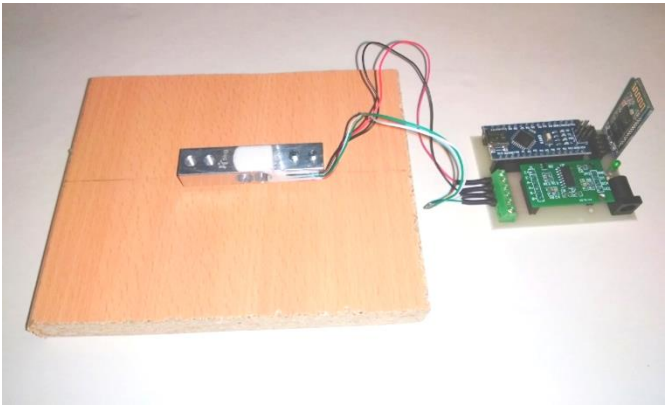


Рисунок 4.2 – Датчик для измерения веса

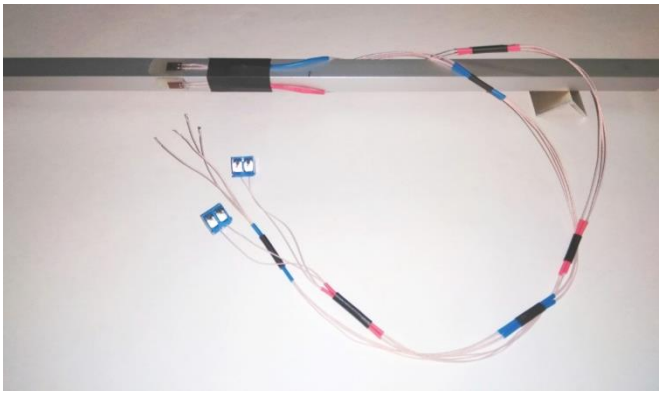


Рисунок 4.3 – Балка с наклеенным тензорным датчиком

На балку ставился груз с заранее измеренной массой на бытовых электронных весах и сравнивался с показаниями устройства (таблица 4.2).

Таблица 4.2 – Тестирование тензометрического датчика

	Масса, г		
№	Электронные весы	До калибровки	После Калибровки
1	76,4	86,3	73,3

2	152,8	135,9	151,3
3	322,3	277,3	335,2
4	120,7	133,9	118,3
5	176,9	168,1	185,7

Была получена погрешность 10-15 % до калибровки и 3-5 % после.

Вывод по главе 4

В ходе апробации изготовленного устройства были получены данные, которые сравнены с легкодоступными бытовыми измерителями. Полученные отклонения достаточно малы (для прототипа) и легко устранимы либо более точной калибровкой, либо переходом на датчики более высокого класса. Таким образом, можно считать, что прототип полностью выполняет поставленные перед ним задачи.

ЗАКЛЮЧЕНИЕ

Анализ существующих устройств сбора информации показал, что большое количество различных решений направлено на конкретные задачи, но не всегда пригодно для широкого спектра измерений или быстрой модификации под новую задачу. По этой причине создание устройства, обладающего характеристиками точного измерительного прибора с малыми габаритами и энергозатратами полезно как в бытовой сфере, так и в промышленности. Для решения данной задачи можно использовать микроконтроллер Atmega328P и отдельный Bluetooth передатчик. Совокупность данных модулей позволяет сделать устройство автономным и легко перенастраиваемым.

Распространенные методы преобразования, фильтрации информации достаточно легко реализуемы, и они позволяют достичь желаемого результата с небольшими затратами как памяти, так и вычислительной мощности микроконтроллера, именно по этой причине они были выбраны в качестве основных методов в данном проекте.

Последовательные этапы планирования, моделирования, макетирования и изготовления устройства позволили учесть ошибки начальной архитектуры и привели к реализации готового, работоспособного прототипа.

Апробация изготовленного устройства показала, что полученные отклонения достаточно малы (для прототипа) и легко устранимы либо более точной калибровкой, либо переходом на датчики более высокого класса.

Таким образом, можно считать, что прототип полностью выполняет поставленные перед ним задачи – устойчиво опрашивает датчики с приемлемой точностью (3 – 5 % погрешности с частотой обновления 0.1 – 5 сек.). Реализован удобный интерфейс в автономном устройстве (размеры: 14x11 см и 5x6 см, питание от аккумулятора). А в сравнении с аналогами габариты меньше в 1.5 – 2 раза, стоимость меньше в 3 – 5 раз. Так же стоимость можно менять в зависимости от важности объекта, что позволяет реализовывать установку в ранее экономически не целесообразных точках.

В дальнейшем возможен переход на более мощную микроконтроллерную платформу (STM или PIC) с расширенной памятью, так же планируется увеличить диапазон работы беспроводного канала посредством использования GSM связи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Fujian Lilliput Optoelectronics Technology Co.,Ltd: SDS Series Smart Digital Storage Oscilloscopes, June. 2011
2. IKALOGIC S.A.S: IkaScope WS200 product data sheet, Mar. 2018
3. Й. Янсен: Курс цифровой электроники. Сложные ИС для устройств передачи данных, том 3. Москва: “Мир” 1987. Стр. 218-235.
4. Martin H. Weik: Communications Standard Dictionary, 1995
5. Texas Instruments Incorporated: 8-bit high-speed Analog-To-Digital Converter, April 2004
6. Пролетарский А.В., Баскаков И.В., Федотов Р.А.: Беспроводные сети Wi-Fi(2-е изд.), Москва: НОУ "ИНТУИТ", 2016
7. В. Кессених, Е. Иванов, З. Кондрашов: Bluetooth: принципы построения и функционирования, 2007
8. Даниил Петров: Стандарты беспроводной связи диапазона ISM. Электронный ресурс. Режим доступа: http://www.russianelectronics.ru/leader-r/review/2187/doc/53409/. Дата обращения 17.10.2018.

9. Petar Popovski; Hiroyuki Yomo; Ramjee Prasad: Strategies For Adaptive Frequency Hopping In The Unlicensed Bands, Dec. 2006

10. Atmel Corporation: 8-bit AVR Microcontroller DATASHEET, Jan. 2015

11. Ingvar Strid, Karl Walentin: Block Kalman Filtering for Large-Scale DSGE Models, June 2008

12. Й. Янсен: Курс цифровой электроники. Проектирование устройств на цифровых ИС, том 2. Москва: “Мир” 1987. Стр. 9-59.

13. Й. Янсен: Курс цифровой электроники. Основы цифровой электроники на ИС, том 1. Москва: “Мир” 1987. Стр. 322-324.

ПРИЛОЖЕНИЕ 1

КОД ПРОГРАММ ДЛЯ МИКРОКОНТРОЛЛЕРА

ПРИЛОЖЕНИЕ 1.1

ОСНОВНОЕ УСТРОЙСТВО

```
#include <U8glib.h>

#include <SoftwareSerial.h>

#include <iarduino_RTC.h>
```

```
#define btTX 10

#define btRX 11

#define u8gE 8

#define u8gRW 7

#define u8gRS 6

#define pinX A1

#define pinY A2

#define pinBut A0

#define pinBacklight 9

U8GLIB_ST7920_128X64 u8g(u8gE, u8gRW, u8gRS,
U8G_PIN_NONE);

SoftwareSerial BT(btTX, btRX);

iarduino_RTC time(RTC_DS3231);

const byte pageCount = 6;

byte page = 0, sensor[pageCount] = {0, 0, 0, 0, 0, 0}, backlight = 150,
settingScroll = 0, timeSelect = 0;

bool grid = 1;

String val = "0#";

long long timeBT = 0;
```

```
void initBoard() {
  Serial.begin(9600);
  BT.begin(9600);
  time.begin();
  pinMode(pinX, INPUT);
  pinMode(pinY, INPUT);
  pinMode(pinBut, INPUT);
  digitalWrite(pinBut, HIGH);
  pinMode(pinBacklight, OUTPUT);
  analogWrite(pinBacklight, backlight);
}
String getBT() {
  String s = "";
  //while (!BT.available());
  while (BT.available()) {
    s += char(BT.read());
    delay(10);
  }
  return s;
```

```
}
char joystick() {
  short int x = analogRead(pinX);
  if (x < 100) return 'R';
  if (x > 900) return 'L';
  short int y = analogRead(pinY);
  if (y < 100) return 'U';
  if (y > 900) return 'D';
  if (!digitalRead(pinBut)) return 'B';
  return 'N';
}
void initSoft() {
  int t = millis();
  byte i = 0;
  while (!BT.available() && millis() - t < 10000) {
    u8g.firstPage();
    do {
      u8g.setColorIndex(1);
```

```
    u8g.setFont(u8g_font_6x10);
    u8g.drawRFrame(32, 14, 64, 15, 2);
    u8g.setPrintPos(35, 24);
    u8g.print("Loading...");
    u8g.drawRFrame(32, 35, 64, 15, 2);
    u8g.setColorIndex(0);
    u8g.drawRBox(33 + i, 37, 8, 11, 2);
    u8g.setColorIndex(1);
    u8g.drawRBox(34 + i, 37, 8, 11, 2);
  } while (u8g.nextPage());
  ++i;
  if (i == 53) i = 0;
  delay(20);
}
if (BT.available()) {
  val = getBT();
  byte c = val[0] - '0';
  page = c;
  sensor[c] = 1;
```

```
  }
}
void graph(String s, byte c = 0) {
  u8g.setColorIndex(1);
  u8g.setFont(u8g_font_6x10);

  u8g.drawFrame(0, 0, 33, 12);
  u8g.setPrintPos(2, 9);
  u8g.print(time.gettime("H"));
  u8g.print(":");
  u8g.print(time.gettime("i"));
  u8g.drawFrame(113, 0, 15, 12);
  u8g.drawBox(115, 2, 3, 8);
  u8g.drawBox(119, 2, 3, 8);
  u8g.drawBox(123, 2, 3, 8);
  u8g.drawLine(36, 3, 36, 8);
  u8g.drawLine(35, 4, 35, 7);
  u8g.drawLine(34, 5, 34, 6);
```

```
  u8g.drawLine(109, 3, 109, 8);
  u8g.drawLine(110, 4, 110, 7);
  u8g.drawLine(111, 5, 111, 6);
  if (c)
    u8g.drawBox(38, 0, 69, 12);
  else
    u8g.drawFrame(38, 0, 70, 12);
  u8g.setColorIndex(!c);
  u8g.setPrintPos(38 + (70 - s.length() * 6) / 2, 9);
  u8g.print(s);
  if (sensor[page] == 0 && page) {
    u8g.setColorIndex(1);
    u8g.setPrintPos(30, 30);
    u8g.print("This sensor");
    u8g.setPrintPos(15, 45);
    u8g.print("is not connected");
  }
}
```

```
void setBacklight() {
  switch (joystick()) {
    case 'L': backlight = (backlight == 0 ? 0 : backlight - 10); break;
    case 'R': backlight = (backlight == 250 ? 250 : backlight + 10); break;
  }
  analogWrite(pinBacklight, backlight);
}
void set_Contrast() {
  //No pin on display (
}
void setGrid() {
  if (joystick() == 'B')
    grid = !grid;
}
void set_Time() {
  switch (joystick()) {
    case 'L': timeSelect = (timeSelect == 0 ? 0 : timeSelect - 1); break;
    case 'R': timeSelect = (timeSelect == 3 ? 3 : timeSelect + 1); break;
```

```
  }
  settingScroll = (timeSelect == 0 ? 4 : 5);
  switch (joystick()) {
    case 'U':
      switch (timeSelect) {
        case 1: time.settime(-1, -1, time.Hours == 23 ? 0 : time.Hours + 1);
break;
        case 2: time.settime(-1, time.minutes == 59 ? 0 : time.minutes + 1);
break;
        case 3: time.settime(0); break;
      }
      break;
    case 'D':
      switch (timeSelect) {
        case 1: time.settime(-1, -1, time.Hours == 0 ? 23 : time.Hours - 1);
break;
        case 2: time.settime(-1, time.minutes == 0 ? 59 : time.minutes - 1);
break;
        case 3: time.settime(0); break;
      }
```

```
      break;
  }
}
void setting() {
  u8g.firstPage();
  do {
    graph("Settings", settingScroll == 0);
    u8g.setFont(u8g_font_6x10);
    u8g.setColorIndex(1);
    u8g.setPrintPos(59, 22);
    u8g.print(map(backlight, 0, 250, 0, 100));
    u8g.print("%");
    if (settingScroll == 1)
      u8g.drawBox(0, 13, 57, 12);
    else
      u8g.drawFrame(0, 13, 57, 12);
    u8g.setColorIndex(settingScroll != 1);
    u8g.setPrintPos(2, 22);
```

```
u8g.print("Backlight");
u8g.setColorIndex(1);
if (settingScroll == 2)
  u8g.drawBox(0, 26, 57, 12);
else
  u8g.drawFrame(0, 26, 57, 12);
u8g.setColorIndex(settingScroll != 2);
u8g.setPrintPos(2, 35);
u8g.print("Contrast");
u8g.setColorIndex(1);
u8g.setPrintPos(59, 48);
if (grid)
  u8g.print("ON");
else
  u8g.print("OFF");
if (settingScroll == 3)
  u8g.drawBox(0, 39, 57, 12);
else
  u8g.drawFrame(0, 39, 57, 12);
```

```
u8g.setColorIndex(settingScroll != 3);
u8g.setPrintPos(2, 48);
u8g.print("Grid");
u8g.setColorIndex(1);
u8g.setPrintPos(59, 61);
u8g.print(time.gettime("H"));
u8g.print(":");
u8g.print(time.gettime("i"));
u8g.print(":");
u8g.print(time.gettime("s"));
u8g.drawLine(40 + timeSelect * 18, 63, 50 + timeSelect * 19, 63);
if (settingScroll == 4)
  u8g.drawBox(0, 52, 57, 12);
else
  u8g.drawFrame(0, 52, 57, 12);
u8g.setColorIndex(settingScroll != 4);
u8g.setPrintPos(2, 61);
u8g.print("Time");
```

```
  } while (u8g.nextPage());
  if (!timeSelect) {
    switch (joystick()) {
      case 'U': settingScroll = (settingScroll == 0 ? 4 : settingScroll - 1);
break;
      case 'D': settingScroll = (settingScroll == 4 ? 0 : settingScroll + 1);
break;
    }
    switch (settingScroll) {
      case 1: setBacklight(); break;
      case 2: set_Contrast(); break;
      case 3: setGrid(); break;
      case 4: set_Time(); break;
    }
  }
  else
    set_Time();
}
void temperature() {
  u8g.firstPage();
```

```
  do {
    graph("Temperature");
  } while (u8g.nextPage());
}
void light() {
  u8g.firstPage();
  do {
    graph("Light");
  } while (u8g.nextPage());
}
void scale() {
  u8g.firstPage();
  do {
    graph("Scale");
    if (sensor[page]) {
      u8g.setColorIndex(1);
      u8g.setPrintPos(15, 30);
      u8g.print("Weight: ");
```

```
      u8g.print(val);
      u8g.print(" g");
    }
  } while (u8g.nextPage());
}
void oscill() {
  u8g.firstPage();
  do {
    graph("Oscill");
  } while (u8g.nextPage());
}
void accel() {
  u8g.firstPage();
  do {
    graph("Accel");
    if (sensor[page]) {
      u8g.setColorIndex(1);
      u8g.setPrintPos(1, 30);
      byte i = 0;
```

```
      u8g.print("X:");
      while (val[i] != '#') {u8g.print(val[i]); ++i;}
      ++i;
      u8g.print(" m/s^2");
      u8g.setPrintPos(1, 40);
      u8g.print("Y:");
      while (val[i] != '#') {u8g.print(val[i]); ++i;}
      ++i;
      u8g.print(" m/s^2");
      u8g.setPrintPos(1, 50);
      u8g.print("Z:");
      while (i < val.length()) {u8g.print(val[i]); ++i;}
      u8g.print(" m/s^2");
    }
  } while (u8g.nextPage());
}
void pageSelect() {
  switch (page) {
```

```
    case 0: setting(); break;
    case 1: temperature(); break;
    case 2: light(); break;
    case 3: scale(); break;
    case 4: oscill(); break;
    case 5: accel(); break;
  }
}
void scroll() {
  if (!settingScroll) {
    switch (joystick()) {
      case 'L': page = (page == 0 ? pageCount : page - 1); break;
      case 'R': page = (page == pageCount ? 0 : page + 1); break;
    }
    settingScroll = 0;
  }
}
void getsensor() {
  String s = getBT();
```

```
  if (s.length() == 0) {
    if (millis() - timeBT > 10000) {
      for (byte i = 0; i < pageCount; ++i)
        sensor[i] = 0;
      val = "0#";
    }
  }
  else {
    timeBT = millis();
    sensor[s[0] - '0'] = 1;
    val = "";
    for (byte i = 2; i < s.length(); ++i) val += s[i];
  }
}
void setup() {
  initBoard();
  initSoft();
}
```

```
void loop() {
  pageSelect();
  scroll();
  getsensor();
  delay(20);
}
```

ПРИЛОЖЕНИЕ 1.2

МОДУЛЬ АКСЕЛЕРОМЕТРА

```
#include <Wire.h>
#include <SoftwareSerial.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
SoftwareSerial BT(11, 10);
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(1);
void setup(void) {
  Serial.begin(9600);
  BT.begin(9600);
  accel.begin();
```

```
  accel.setRange(ADXL345_RANGE_16_G);}
void loop(void) {
  sensors_event_t val;
  accel.getEvent(&val);
  BT.print("5#");
  BT.print(val.acceleration.x);
  BT.print("#");
  BT.print(val.acceleration.y);
  BT.print("#");
  BT.print(val.acceleration.z);
  Serial.print("X: ");
  Serial.println(val.acceleration.x);
  Serial.print("Y: ");
  Serial.println(val.acceleration.y);
  Serial.print("Z: ");
  Serial.println(val.acceleration.z);
  Serial.println("  m/s^2 ");
  delay(500);}
```

ПРИЛОЖЕНИЕ 1.3

МОДУЛЬ ТЕМПЕРАТУРЫ

```
#include <DallasTemperature.h>
#include <SoftwareSerial.h>
SoftwareSerial BT(11, 10);
#define ONE_WIRE_BUS A0
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
void setup() {
  sensors.begin();
  Serial.begin(9600);
  BT.begin(9600);
  pinMode(A1, INPUT);
}
void loop() {
  sensors.requestTemperatures();
  float temp = sensors.getTempCByIndex(0);
  int light = map(analogRead(A1), 0, 1023, 0, 100);
```

```
  BT.print("1#");
  BT.print(temp);
  delay(250);
  BT.print("2#");
  BT.print(light);
  Serial.print("Temperature: ");
  Serial.println(temp);
  Serial.print("Light: ");
  Serial.println(light);
  delay(250);
}
```

ПРИЛОЖЕНИЕ 1.4

МОДУЛЬ ТЕНЗОРНОГО ДАТЧИКА

```
#include "HX711.h"
#include <SoftwareSerial.h>
HX711 sensor(A1, A0);
SoftwareSerial BT(11, 10);
```

```
float calibration = 6.7;
void setup() {
  Serial.begin(9600);
  BT.begin(9600);
  sensor.set_scale();
  sensor.tare();
  sensor.set_scale(calibration);
}
void loop() {
  float weight = 0;
  for (int i = 0; i < 5; i ++) {
    weight += sensor.get_units();
    delay(2); }
  weight = weight / 10 * 0.035274;
  BT.print("3#");
  BT.print(weight);
  Serial.println(weight);
  delay(500);
}
```

СЕКЦИЯ III. Информатика и системы управления

Метеостанция для измерения основных параметров атмосферы у земли и на высотах

Моторкин Владимир Максимович

г. Москва, ГБОУ №1580, 11 класс

Содержание работы:

Введение

Основная часть

1. Аналитический раздел
   1.1. Выбор платформы для создания метеостанции
   1.2. Выбор датчиков
   1.3. Выбор основы для наземной станции
   1.4. Выбор полетного контроллера БПЛА
   1.5. Выбор модуля для передачи данных
   1.6. Выбор рамы для БПЛА
   1.7. Программное обеспечение
2. Технологическая часть
   2.1. Сборка метеостанции
   2.2. Сборка наземной станции
   2.3. Написание программы
   2.4. Сборка БПЛА

Заключение

Ссылки на источники

Введение.

Зондирование атмосферы (ЗА) - определение вертикального или горизонтального распределения температуры, влажности, давления, ветра и других физических параметров атмосферы. Наибольшее значение имеет вертикальное ЗА.

Методов вертикального ЗА существует много: зондирование с помощью радиозондов, оптическое — лучом лазера, акустическое (звуком), радиолокационное, ракетное и др. Радиозондирование представляет наиболее точные результаты непосредственных контактных измерений термодинамических параметров атмосферы на высотах от уровня земли до 35…40 км, поэтому данный вид зондирования является наиболее распространённым.

Эти данные содержат информацию о вертикальных профилях температуры, влажности, скорости и направлении ветра, а также о давлении воздуха на заданных уровнях. Для получения информации в атмосферу выпускаются в свободный полёт небольшие лёгкие измерительные приборы, снабжённые датчиками различных метеорологических параметров и радиопередатчиком. Такие приборы, называемые радиозондами, поднимаются до больших высот с помощью специальных латексных шаров (оболочек), наполняемых лёгким газом – водородом или гелием.

Радиозондирование до сих пор в мире является основным средством определения вертикального профиля ветра. Но радиозондирование, наряду с высокой стоимостью расходных материалов, имеет недостаток, связанный с редкой частотой выпуска радиозондов (выпускаются радиозонды одновременно во всём мире дважды в день в 00.00 и 12.00 мирового времени).

Учитывая, что ветровое поле может изменяться кардинально за несколько минут, такая низкая оперативность получения ветровых данных не устраивает многих потребителей. Так, например, при организации прыжков с парашютом, необходимо знание среднего ветра от поверхности земли до высоты десантирования. Для чего используется в настоящее время - шаропилотный теодолитный метод (суть метода: наблюдая за перемещением шара-пилота в аэрологический теодолит и зная его вертикальную скорость, исходя из угловых координат б и в местоположения шара на различных высотах подъема, определяется направление и скорость его горизонтального перемещения. Поскольку шар-пилот перемещается вместе с воздушными течениями в атмосфере, полученные значения будут выражать скорость и направление ветра на высотах).

Данный метод имеет следующие существенные недостатки:

резкое снижение эффективности в тёмное время суток и в условиях низкой облачности;

громоздкость оборудования необходимого для проведения наблюдений (нужны баллоны с гелием или водородом);

необходимость проведения множества вычислений и как следствие большая вероятность ошибки.

Также быстро изменяется распределение температуры по высотам, которое необходимо знать для оценки условий распространения загрязняющих веществ в условиях крупного мегаполиса, прогноза тумана.

Так при наличии инверсии (роста температуры с высотой) образуется слой (рис.1), препятствующий вертикальным перемещениям воздуха и способствующий образованию дымки, тумана, смога, облаков, миражей.

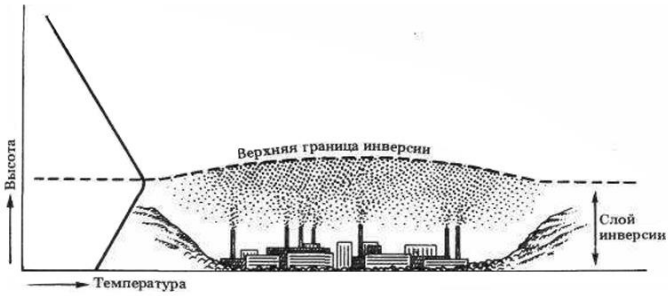


Рисунок 1.

Целью данного проекта является создание метеостанции, поднимаемой в атмосферу при помощи беспилотного летательного аппарата (БПЛА), которая способна оперативно (в реальном времени) предоставлять пользователю необходимую информацию о параметрах атмосферы от поверхности земли до заданной высоты.

В процессе разработки были поставлены следующие задачи:

- создание метеостанции, способной измерять и передавать на землю основные параметры атмосферы: температура и влажность воздуха, скорость и направление ветра, давление, высота нижней и верхней границ облачности
- изготовление наземной станции, способной принимать данные от метеостанции;
- написание программного обеспечения для сбора, обработки и архивирования полученных данных, а также предоставление их пользователю в удобном для него виде;
- проектирование и изготовление БПЛА, способного поднимать метеостанцию на заданную высоту.

Основная часть.

1. Аналитический раздел.

Для создания метеостанции было необходимо выбрать основу (платформу) и датчики.

1.1. Выбор платформы для создания метеостанции.

- Arduino UNO R3: контроллер построен на ATmega328 Платформа имеет 14 цифровых вход/выходов, 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки.
- RobotDYN : Плата - бюджетный аналог Arduino UNO R3 от российского производителя и конструктивно практически идентична оригинальной плате. В отличии от оригинала плата RobotDyn, имеет более распространенный разъем Micro USB.
- Teensy 3.6 : Teensy — это мощная Arduino-совместимая платформа для разработки в компактном форм-факторе. Главное отличие старших представителей версии 3.x — высокопроизводительный ARM-процессор Cortex-M4, большое количество памяти, встроенный слот для SD-карты и широкий набор интерфейсов.

Плата Arduino UNO R3 имеет все необходимые компоненты для обеспечения работы микроконтроллера. Благодаря наличию DIP панели, можно менять микроконтроллер. Данная плата выдерживает резкие перепады температур, имеет низкое энергопотребление, кроме

этого является одной из самых распространённых в своём классе и для неё создано наибольшее количество уроков.

Характеристики Arduino UNO R3

Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА

Флеш-память	32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика
ОЗУ	2 Кб (ATmega328)
EEPROM	1 Кб (ATmega328)
Тактовая частота	16 МГц

Выбор датчиков.

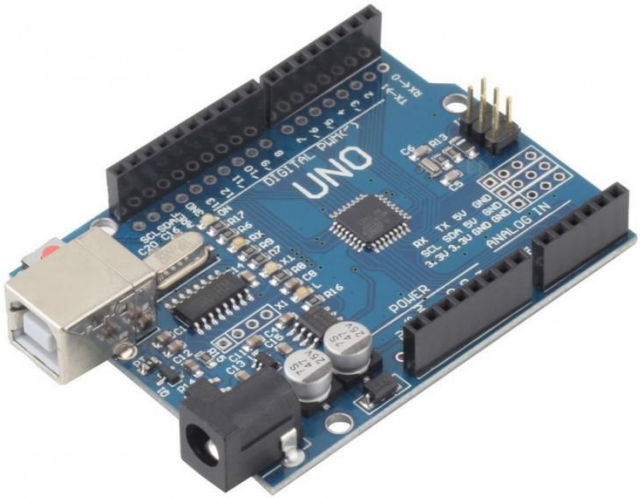


Рисунок 2. Arduino UNO R3.

1.2. Выбор датчиков.

При выборе датчиков основными характеристиками являлись точность измерений, устойчивая работа при низких температурах и пределы измерений. На основании этого были выбраны следующие датчики.

DHT-22 (также называемый AM2302) представляет собой датчик влажности и температуры с цифровым выходом, для измерения используется емкостной датчик влажности и термистор, все показания передаются по цифровой информационной шине.

- Технические параметры:
- Модель: DHT22
- Напряжение питания: 3.3 В … 5 В
- Выходной сигнал: цифровой
- Чувствительный элемент: полимерный конденсатор
- Диапазон измерения влажности: 0 … 100%, погрешность ±2%
- Диапазон измерения температуры: -40°C … +80°C, погрешность ±0.5°C
- Задержка: 2с
- Габариты: 15.1 мм x 25.1 мм x 7.7 мм

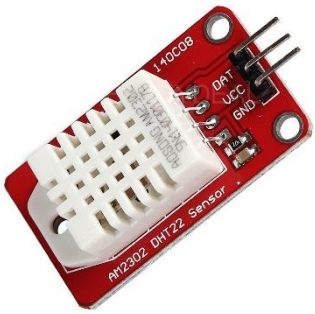


Рисунок 3. Датчик температуры и влажности DHT 22.

Спецификация BMP280:

- Напряжение питания модуля: 3,3 или 5 В постоянного тока (поддерживаются оба уровня).
- Потребляемый ток: до 2 мА во время измерений (зависит от режима точности).
- Потребляемый ток: до 0,2 мА в режиме ожидания.
- Измеряемое давление: от 30'000 до 110'000 Па (разрешение 0,16 Па)
- Измеряемая температура: от 0 до +65 °C (разрешение 0,01°C)
- Рабочая частота шины I2C: до 3,4 МГц.
- Адрес модуля на шине I2C: 0x77.
- Уровень логической «1» на шине I2C: от 0,7*Vcc до Vcc (где Vcc это напряжение питания модуля)
- Подготовка к первому запуску после подачи питания: не менее 2 мс.
- Рабочая температура: -40 … +85 °C
- Габариты: 30x30 мм.

BMP280 (рис. 4) — это цифровой модуль на базе чипа BMP280, подключаемый по шине I2C (адрес 0x77), позволяющий получить текущие значения атмосферного давления и температуры окружающей среды. Еще одним применением данного модуля является определений

высоты, которая зависит от давления и рассчитывается по международной барометрической формуле. Точность Trema барометра позволяет фиксировать изменение высоты от 20 см.

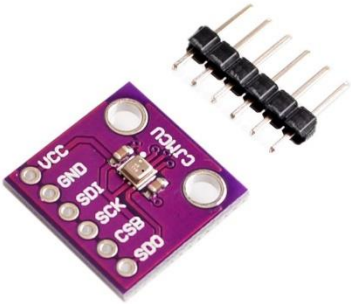


Рисунок 4. Датчик атмосферного давления BMP280.

HC-SR04 (рис. 5) - Ультразвуковой датчик для измерения расстояния и обнаружения предметов в диапазоне от 2 сантиметров до 4метров. В рамках данного проекта используется для измерения высоты нижней границы облачности.

Характеристики:

- Напряжение питания: 5 В
- Потребление в режиме тишины: 2 мА
- Потребление при работе: 15 мА
- Диапазон расстояний: 2–400 см
- Эффективный угол наблюдения: 15°

- Рабочий угол наблюдения: 30°

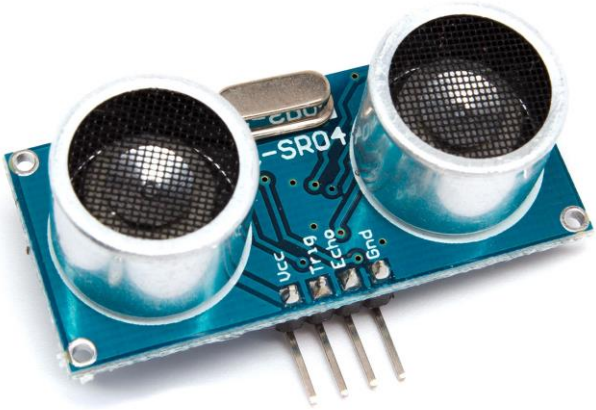
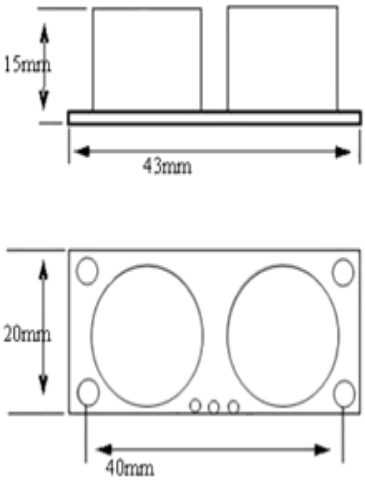


Рисунок 5. Датчик приближения HC-SR04.



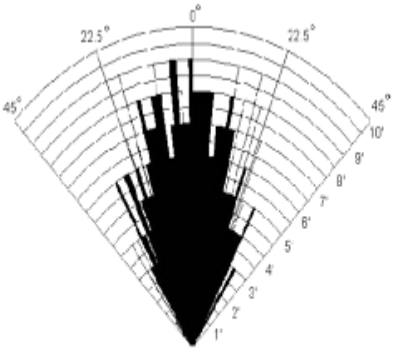


Диаграмма направленности

Рисунок 6. Размеры и диаграмма направленности HC-SR04.

Датчика скорости ветра

Характеристики:

- Точность: +-1 м/с
- Начальный ветер: 0,2 м 0,4 м/с
- Диапазон: 0 ~ 32,4 M/s
- Выходные сигналы: 0-5 В
- Скорость ветра значения:(Выходное напряжение)/5* 32,4

Рисунок 7. Датчик скорости ветра.

Компас на LIS3MDL.

Характеристики:

- Напряжение питания: 3.3–5 В
- Диапазон погрешности измерения: ±4/±8/±12/±16 Гаусс
- Выходной интерфейс: I²C
- Частота обновления сигнала: 0.625…80 Гц
- Температурный диапазон: −40…+85 °C
- Габариты: 25×25 мм

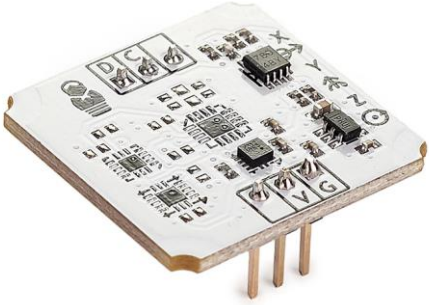


Рисунок 8. Магнитометр (компас) на LIS3MDL.

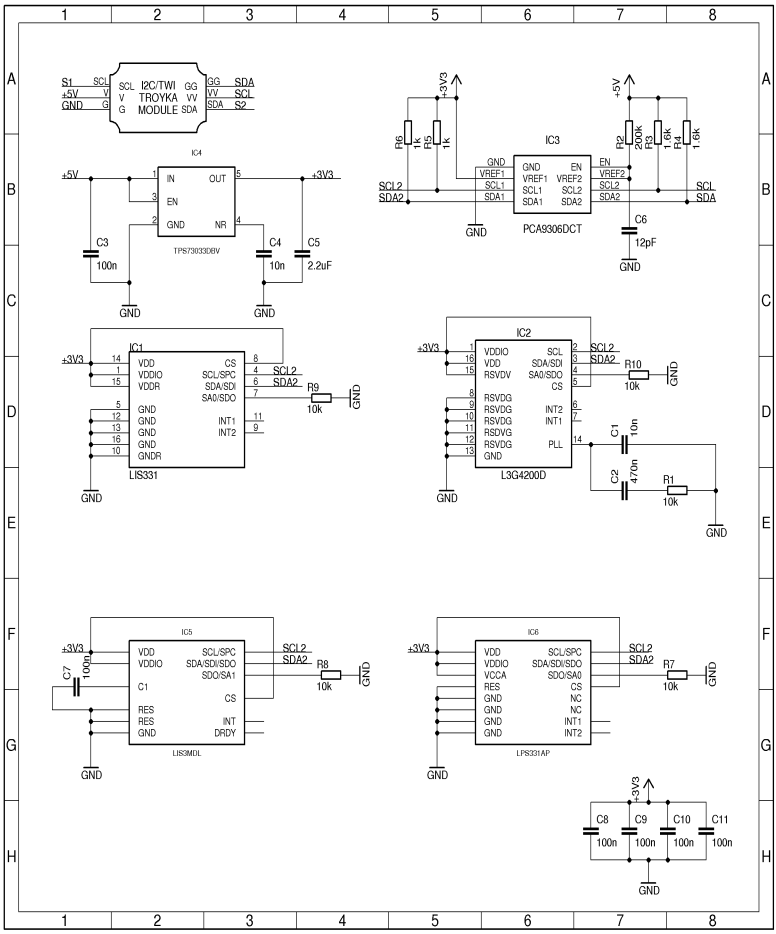


Рисунок 9. Принципиальная схема магнитометра.

1.3. Выбор основы для наземной станции.

Для основы наземной станции была выбрана плата Arduino Nano.

Nano – одна из самых миниатюрных плат Ардуино. Она является полным аналогом Arduino Uno – так же работает на чипе ATmega328P (хотя можно еще встретить варианты с ATmega168), но с меньшим форм-фактором. Из-за своих габаритных размеров плата часто используется в проектах, в которых важна компактность. На плате отсутствует вынесенное гнездо внешнего питания, Ардуино работает через USB (miniUSB или microUSB). В остальном параметры совпадают с моделью Arduino Uno.

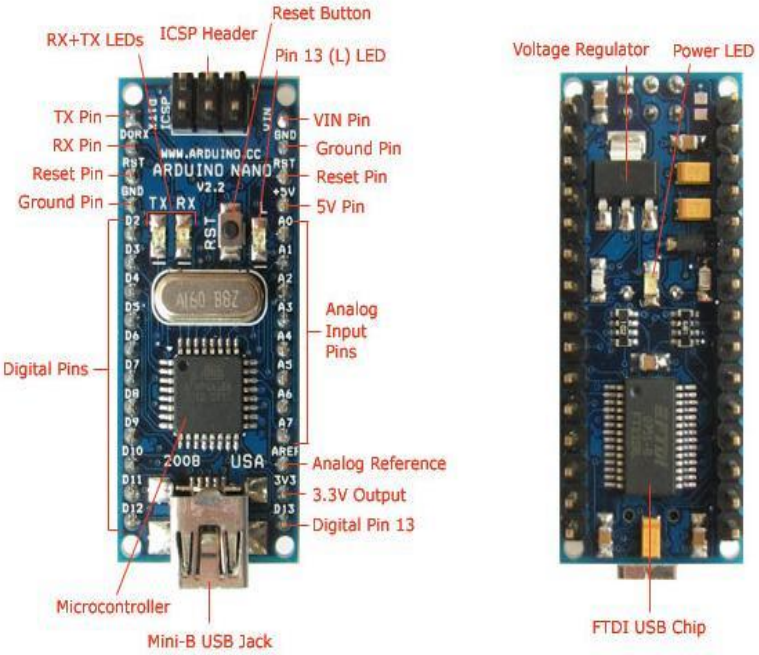


Рисунок 10. Arduino Nano.

Технические характеристики Arduino Nano:

- Напряжение питания 5В;
- Входное питание 7-12В (рекомендованное);
- Количество цифровых пинов – 14, из них 6 могут использоваться в качестве выходов ШИМ;
- 8 аналоговых входов;
- Максимальный ток цифрового выхода 40 мА;

- Флэш- память 16 Кб или 32 Кб, в зависимости от чипа;
- ОЗУ 1 Кб или 2 Кб, в зависимости от чипа;
- EEPROM 512 байт или 1 Кб;
- Частота 16 МГц;
- Размеры 19 х 42 мм;
- Вес 7 г.

1.4. Выбор полетного контроллера БПЛА.

Полетный контроллер ArduPilot Mega является полноценным решением БПЛА, который позволяет помимо радиоуправляемого дистанционного пилотирования - автоматическое управление по заранее созданному маршруту, т.е. полет по точкам, а также обладает возможностью двухсторонней передачей телеметрических данных с борта на наземную станцию (телефон, планшет, ноутбук, DIY) и ведение журнала во встроенную память.

Данный контроллер является наиболее оптимальным решением в рамках данного проекта. Кроме того, возможность автоматического полета и передачи телеметрических данных позволяет полностью автоматизировать процесс измерения параметров атмосферы.

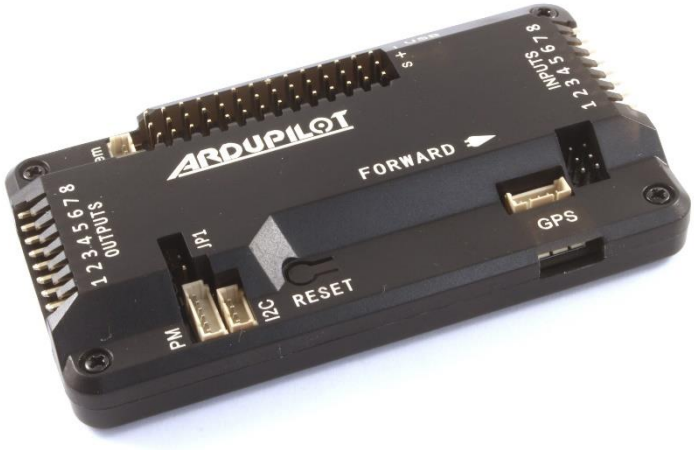


Рисунок 11. Полётный контроллер ArduPilot.

Сравнительная таблица №1. Преимущества автопилота Ardupilot Mega перед DJI NAZA, Wookoong-M, ZeroUAV (YS-X4,YS-X6), Mikrocopter, Hovefly Pro, uThree

CONTENTS	**APM** AUTOPILOT,	**DJI (Naza w/GPS)**	**DJI (WOOKONG-M WAYPOINT)**	**ZEROUAV (YS-X4 LOW**	**ZEROUAV (YS-X6 HIGH**	**MIKROKOPTER** AUTOPILOT, GPS,	**HOVERFLY PRO w/GPS**	**UTHREE (RUBY)**

	GPS, POWER MONITOR, LONG-RANGE TELEMETRY	AUTOPILOT, GPS, POWER MONITOR	——— AUTOPILOT, GPS, POWER MONITOR, TELEMETRY, 50 WAYPOINT PACKAGE	**END)** ——— AUTOPILOT, GPS, POWER MONITOR, TELEMETRY, 8 WAYPOINT PACKAGE	**END)** ——— AUTOPILOT, GPS, POWER MONITOR, LONG-RANGE TELEMETRY, 16 WAYPOINT PACKAGE	POWER MONITOR, LONG-RANGE TELEMETRY, 100 WAYPOINT PACKAGE	AUTOPILOT, GPS ONLY	AUTOPILOT, GPS
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

LIST PRICE	**$249**	**$399**	**~$3000**	**$639**	**~$2700**	**~$2500**	**$900**	**$345**
FULL AUTONOMY	Yes	RTL only	Yes	Yes	Yes	Yes	Yes	Yes
WAYPOINTS (# SUPPORTED)	166	1	50	8 w/in 500m (Navigator)	16 w/in 500m (Standard)	100 w/in 250m	1	1
IN-FLIGHT ROUTE EDITING	Yes	No	Yes	Guided Mode Only	Guided Mode Only	Yes	No	No
WARELESS CONFI	Yes (Radio Tele	No	Yes (Bluetooth)	No	No	No	No	No

GURATION	metry)							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
GEO-FENCE SUPPORT	Yes	No	No	No	No	No	No	No
ROBUST SIMULATION (HIL/SIL)	Both	No	HIL Only	No	No	SIL Only	No	No
OPEN SOURCE TOOLS AND CODE	Yes	No	No	No	No	No	No	No
OPEN SOURCE COMM	Yes (MAVLink)	No	No	No	No	No	No	No

UNICATION PROTOCOL								
TRICOPTER	Yes	No	No	No	No	No	No	No
QUADCOPTER (+ AND X)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
HEXACOPTER	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
OCTOCOPTER	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Y6	Yes	Yes	Yes	Yes	Yes	No	Yes	No
X8	Yes	Yes	Yes	Yes	Yes	No	Yes	No

TRADITIONAL HELICOPTERS	Yes	No	Yes (Ace One)	No	No	No	No	No
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
FIXED-WING AIRCRAFT	Yes	No	No	No	No	No	No	Yes
ROVERS	Yes	No	No	No	No	No	No	No
BOATS	Yes	No	No	No	No	No	No	No

Особенности:

- 3 осевой гироскоп, акселерометр, магнитометр (до 2.5.2) и высокоточный барометр
- Система стабилизации с возможностью воздушной акробатики

- Удержание позиции по GPS, полет по точкам и возврат на точку старта
- Возможность использования инфракрасного датчика для обхода препятствий
- Поддержка ультразвукового датчика (Sonar sensor) для автоматического взлета и посадки
- Автоматическое следование по маршрутным точкам
- Управление двигателями посредством ШИМ (PWM) с использованием дешевых регуляторов скорости (ESC)
- Собственная система стабилизации для камеры (функция контроллера подвеса)
- Радиосвязь и телеметрия с борта
- Поддержка множества рам и конфигураций летающих и ездящих аппаратов
- Поддержка датчика уровня заряда батареи
- Настраиваемая световая индикация при полетах
- Совместим с многими радиоуправляемыми приемниками PWM и PPM сигналов
- Передача в реальном времени телеметрических данных
- Поддержка OSD телеметрии (наложение на видеопередачу телемерических данных) используя протокол MAVLINK
- Конфигурирования точек полета посредством Google Maps
- Бортовая флеш память 16Мбит для автоматической регистрации данных

- Цифровой компас работает на HMC5883L (до версии 2.5.2)
- 6 степеней свободы в InvenSense акселерометре, гироскоп MPU-6000
- Датчик барометрического давления обновлен до MS5611-01BA03, от Measurement Specialties
- Контроллер Atmel ATmega2560-16AU и ATMEGA32U-2 чип для обработки и функции USB
- Возможна загрузка обновлений встроенного программного обеспечения и конфигурации

Режимы полетов:

- Стабилизация (Stabilize) - Удержание горизонта
- Удержание высоты (AltHold)
- Loiter - замри и слоняйся
- RTL (Return-to-Launch) - вернутся на точку старта
- Auto - выполнение заданного маршрута в автоматическом режиме
- Acro - акробатика
- Sport - для FPV
- Circle - облет по кругу, радиус задается. включая режим Panorama и ROI
- Drift - полет как у самолета
- Follow Me - следуй за мной, доступно при телеметрии со своей GPS

- Guided - указание пункта назначения, доступно в телеметрии
- Position - фиксация в воздухе с ручным газом взлета
- Land - автоматическая посадка
- Simple and Super Simple - легкий и суперлегкий полет, подходит для новичков
- Квадрокоптер летает за тобой, как AirDog, Hexo+ и другие экшен версии

1.5. Выбор модуля для передачи данных.

Основные условия при выборе модуля: возможность работы на большой высоте, вне зоны действия сотовых вышек, при низких температурах; надежная передача малых объемов данных на большое расстояние при низком энергопотреблении.

LoRa SX1278: современный RF модуль, работающий на частоте 433МГц и имеющий поддержку стандарта LoRa. Особенность стандарта LoRa — это передача небольших пакетов данных с невысоким энергопотреблением. По заверениям производителя, дальность на открытом воздухе может достигать 10км, а время работы от батареи может составлять несколько лет.

Данный модуль имеет уникальную в своем роде технологию, которая идеально подходит для передачи малых объемов данных по воздуху.

Таблица №2а. Характеристики модуля LoRa SX1278.

Параметр РФ	Значение	Примечание
Рабочая частота	410 ~ 441 МГц	По умолчанию: 433 МГц
Мощность передатчика	10 ~ 20 дБм	По умолчанию: 20 дБм (около 100 мВт)
Чувствительность приема	-146 дБм	Скорость передачи данных: 0,3 кбит/с
Скорость передачи данных воздуха	0.3k ~ 19,2 кбит/с	По умолчанию: 2,4 кбит/с
Тесты расстояние	3000 метра	В открытом и прозрачном воздухе, с максимальной мощностью, 5dBi усиление антенны, высота 2 м, Скорость передачи данных: 2,4 кбит/с

Таблица №2б. Характеристики модуля LoRa SX1278.

Параметр оборудование	Значение	Примечание
Размер	21*36 мм	Без SMA
Тип антенны	SMA-K	
Связь интерфейс	UART	Скорость передачи данных: 1200 ~ 115200, по умолчанию: 9600
Посылка	DIP	
Буфер	512 байт буфера	Автоматическая подупаковка с 58 байтами посылка

Таблица №2в. Характеристики модуля LoRa SX1278.

Электронный параметр	Мин.	Тип.	Макс.	Блок	Условие
Мощность питания	2,3	3,3	5,5	V	

Связь уровня	3,0	3,3	3,6	V	
Передача ток	102	110	118	МА	20дбм (100 мВт)
	90	93	96	МА	10 дБм (50 мВт)
Получение текущей	12	15	18	МА	
Сна	3	5	8	МкА	
Рабочая температура	-40	20	+85	℃	
Рабочая влажность	10	60	90	%	
Температура хранения	-40	20	+125	℃	

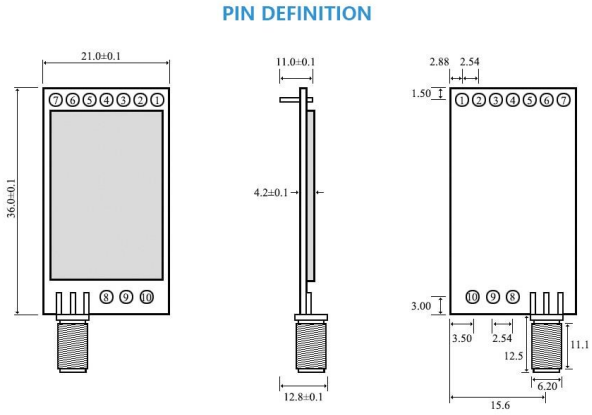


Рисунок 12. Распиновка модуля LoRa SX1278.

Рисунок 13. Модуль LoRa SX1278.

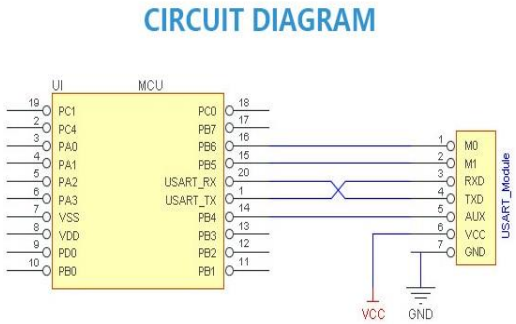


Рисунок 14. Схема модуля LoRa SX1278.

1.6. Выбор рамы для БПЛА.

Рама для БПЛА выбиралась исходя из её размеров, а именно расстояния между винтами, так как для работы датчика скорости ветра с минимальной погрешностью необходимо минимизировать влияние вертикальных потоков воздуха, создаваемых лопастями квадрокоптера. В то же время она должна быть удобной для переноски.

Материал рамы – стекловолокно.

Рисунок 15. Рама F450 БПЛА.

1.7. Программное обеспечение.

Для написания программы был выбран язык C#, так как его поддерживает интегрированная среда разработки Visual Studio, благодаря которой можно удобно разработать простой и понятный интерфейс. Кроме того, она имеет множество встроенных библиотек, в том числе для работы с графиками и COM-портами.

2. Технологическая часть.

2.1. Сборка метеостанции.

При помощи монтажной платы к микроконтроллеру Arduino были присоединены датчик температуры и влажности воздуха, датчик атмосферного давления и датчик приближения. Датчик скорости ветра был подключён через конденсатор 100нФ и резистор 10кОм. Это необходимо для точных показаний.

Была написана программа для микроконтроллера и произведена проверка датчиков путем присоединения по COM-порту к ПК.

Метеостанция без корпуса изображена на рисунке 15.

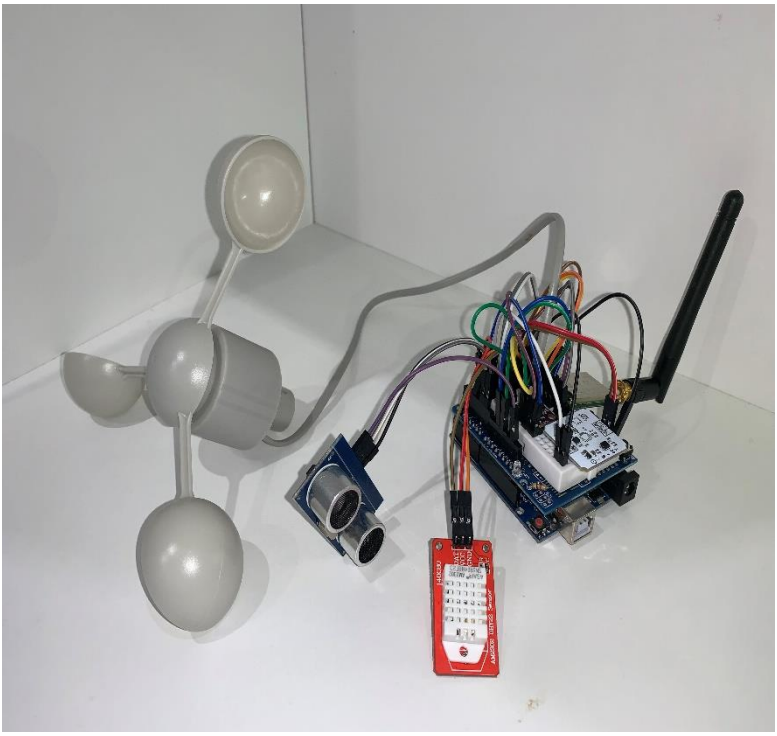


Рисунок 15. Arduino Uno с датчиками и модулем связи.

Далее была изготовлена 3D модель корпуса для метеостанции в виде флюгера. Такая форма была использована для того, чтобы корпус, вращаясь вокруг свой оси, являлся указателем

направления ветра (рис. 16а, 16б).

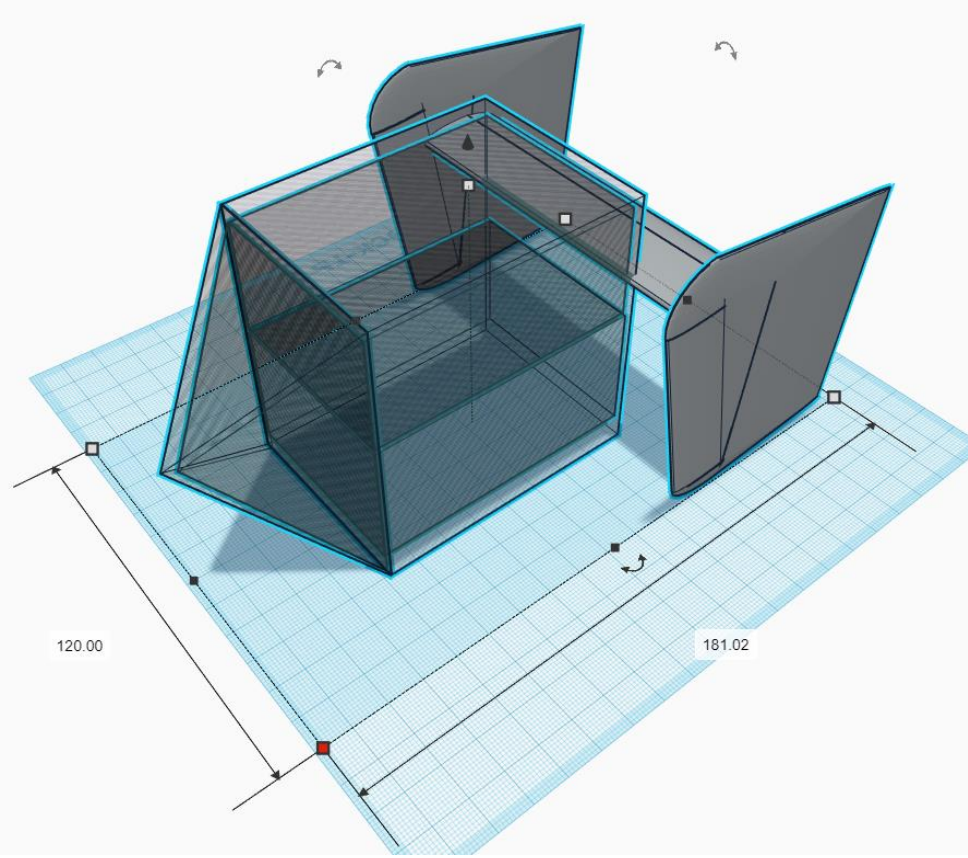


Рисунок 16а. Корпус метеостанции в разрезе.

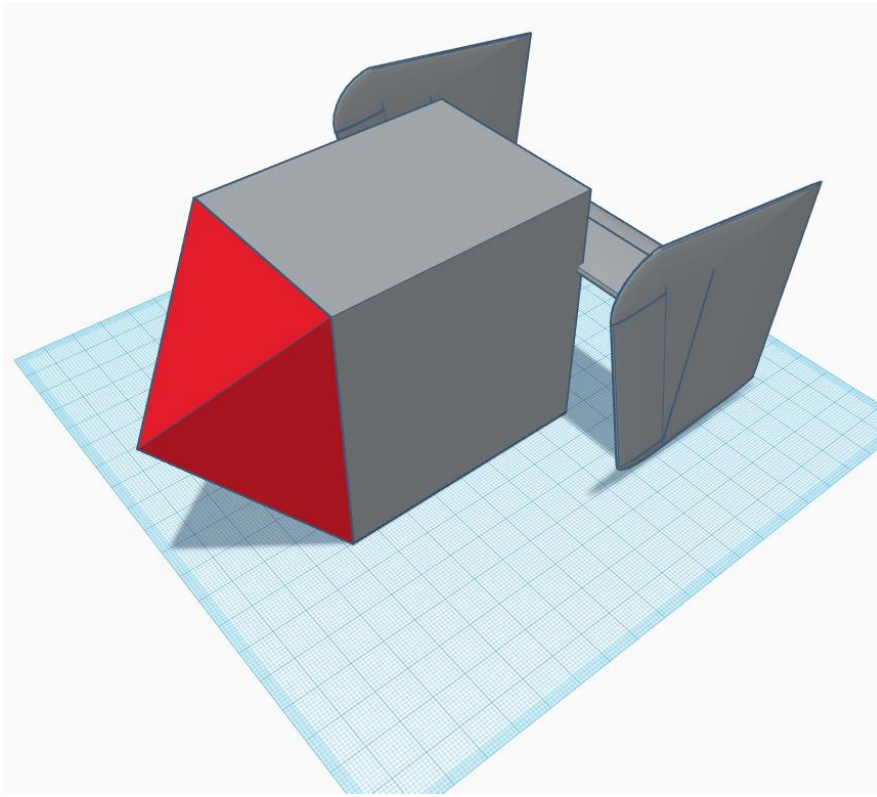


Рисунок 16б. Внешний вид корпуса метеостанции.

2.2. Сборка наземной станции.

Был присоединен модуль передачи данных к Arduino Nano. Была произведена проверка передачи данных с метеостанции на наземную станцию.

Рисунок 17. Наземная станция.

2.3. Написание программы

Была написана программа на языке C# и разработан удобный интерфейс в среде программирования Visual Studio.

В качестве базы данных был использован простой текстовый файл, так как доступ к данным осуществляется последовательно и объем данных сравнительно небольшой.

2.4. Сборка БПЛА.

На раму F450 были установлены полётный контроллер ArduPilot, бесколлекторные электродвигатели на 920 кВ, регуляторы напряжения на 30А, аккумулятор ZOP Power 11.1V 6300mAh 35C 3S Lipo Battery, обеспечивающий полет до 30 минут. Для удобства управления была установлена FPV (First Person View) система, состоящая из передатчика Eachine TS832 Boscam FPV 5.8G 48CH 600mW и малогабаритной камеры и GPS модуль для полета по точкам (координатам).

Рисунок 18. БПЛА в сборе.

Заключение.

В процессе разработки поставленные цели были достигнуты. Был изготовлен беспилотный летательный аппарат с метеостанцией, способный производить вертикальное зондирование атмосферы путем передачи её основных параметров: температуры и влажности воздуха, скорости и направления ветра, давления, высоты нижней и верхней границ облачности.

В процессе работы над проектом в него были внесены некоторые дополнения, метеостанция была изготовлена как самостоятельное устройство, способное передавать данные не только будучи установленной на БПЛА, но и самостоятельно. Этого удалось достичь путем установки дополнительного питания. Таким образом, метеостанция может использоваться как самостоятельное устройство без БПЛА для наблюдения за параметрами атмосферы в приземном слое.

Возможности применения.

Данное устройство может быть использовано для решения следующих задач:

- Определение наличия инверсии для оценки условий распространения загрязняющих веществ в условиях крупного мегаполиса, прогноза тумана.

- Определение среднего ветра от поверхности земли до заданной пользователем высоты.
- Определение наличия сдвига ветра в приземном слое атмосферы.
- Измерение основных параметров атмосферы у поверхности земли.

  Дальнейшее развитие системы:

- Возможность определять точку десантирования, с отображением её на карте.
- Осуществление автоматизированных полетов от поверхности земли до заданной пользователем высоты.

Ссылки на источники.

1. Виктор Петин. Проекты с использованием контроллера Arduino, 2-е издание - БХВ-Петербург, 2015 г.
2. Улли Соммер. Программирование микроконтроллерных плат Arduino/Freeduino- БХВ-Петербург, 2012 г.
3. Теро Карвинен, Киммо Карвинен, Вилле Валтокари. Делаем сенсоры. Проекты сенсорных устройств на базе Arduino и Raspberry Pi – Вильямс, 2015 г.
4. Валерий Яценков. Электроника. Твой первый квадрокоптер. Теория и практика - БХВ-Петербург, 2017 г.
5. В. Н. Киселев, А. Д. Кузнецов. Методы зондирования окружающей среды (атмосферы) - РГГМУ Санкт-Петербург, 2004 г.

6. В.В.. Стерлядкин, А.Г. Горелик, Г.Г.Щукин. Обзор методов и средств ветрового зондирования атмосферы – МГУПИ, МФТИ, ВКА им. Можайского, 2011 г.
7. Сравнение AtduPilot с аналогами. http://ardupilot.su
8. Дальномер http://wiki.amperka.ru/продукты:hc-sr04-ultrasonic-sensor-distance-module
9. Датчик давления BMP280 https://arduinomaster.ru/datchiki-arduino/datchiki-atmosfernogo-davleniya-bmp280-bmp180-bme280/
10. Датчик температуры и влажности воздуха DHT22 http://robotchip.ru/obzor-datchika-temperatury-i-vlazhnosti-dht22/
11. Компас http://wiki.amperka.ru/продукты:troyka-compass
12. Электродвигатели, регуляторы напряжения https://m.aliexpress.com/item/32837080614.html
13. Модуль связи https://m.aliexpress.com/item/32791728376.html
14. Датчик скорости ветра https://m.ru.aliexpress.com/item/32711809008.html
15. Рама F450 https://m.aliexpress.com/item/32609712090.html

СЕКЦИЯ III. Информатика и системы управления

Полнофункциональный мобильный фоторедактор Polo с элементами социальной сети

Чудновский Даниэль Евгеньевич
г. Москва, ГБОУ Школа №138, 11 класс
Научный руководитель: Степанов Павел Валерьевич, преподаватель кафедры ИУ-6, МГТУ им. Н.Э. Баумана

Polo - полнофункциональный мобильный фоторедактор – полезный инструмент в создании своих мини-историй в социальных сетях. С его помощью люди могут оперативно делиться эмоциями и новостями. Мое приложение максимально разнообразит это общение.

При его создании я решил добавить, помимо стандартных функций редактора, новые, отсутствующие у других приложений, среди которых: редактор градиентов, готовые паттерны для рисования, пользовательская матрица свертки, готовые цитаты и стихи, пипетка и онлайн лента публикаций. Малый вес также является дополнительным преимуществом и отличием.

Проект реализован на языке программирования java на платформе android. При разработке функций, связанных с обработкой фотографий, мною не использовались сторонние библиотеки. Каждый элемент, добавляемый пользователем, является объектом класса со своими характеристиками. Работа фильтров выполняется в нескольких

потоках для ускорения. Данные для заполнения онлайн ленты отправляются в формате json.

Результатом работы над данным проектом стал полноценный фоторедактор. Полученный продукт обладает преимуществом перед аналогами за счет новых и свежих функций, а также быстрой обучаемости работы.

Фоторедактор Polo получил Золотой диплом IT школы Samsung и первое место в номинации “Социальное приложение”. В том же 2018 году он получил Гран-При на конкурсе “IT школа Samsung выбирает сильнейших”. Все это вдохновляет меня на продолжение разработки по пути усовершенствования продукта.

Содержание

Введение
1 Используемые методы
2 Приветственный экран
3 Экран редактирования
4 “Кисть”
5 “Текст”
6 “Стикеры”
7 “Фильтры”
8 “Уровни”
9 “Дизайн”
10 “Микс”

11 Онлайн лента публикаций

12 Сравнение Polo и Retrica

Заключение

Список использованной литературы

Введение

Одним из важнейших вещей в жизни человека является общения.

Его значение трудно переоценить.

Мы не можем обойтись без него. Общение – это сложная, многофункциональная и многогранная составляющая человеческой жизни. Процесс коммуникации в современном мире стремительно развивается, предоставляя нам всё новые, улучшенные возможности. Интернет открывает в этом смысле значительные перспективы, позволяя человеку общаться в любой момент и с любой аудиторией.

Для реализации этих возможностей были созданы социальные сети, которые занимают всё больше место в жизни современного общества. Они плотно вошли в нашу жизнь.

Исследования показывают, что люди во всем мире предпочитают выходить в интернет со смартфонов. Они генерируют больше веб-трафика, чем все прочие устройства суммарно. Смартфон с приложениями пользователь носит с собой и практически не выпускает их из рук.

Мое приложение позволит оперативно делиться эмоциями и создавать истории в популярных социальных сетях. При его создании закладывалось 5 идей:

1) Всегда под рукой. Достаточно нажать на иконку и начать редактирование.
2) Быстрое редактирование. Всего в пару тапов можно преобразить фото.
3) Разнообразие общения. Картинки и фото добавляют эмоции вашим сообщениям. Готовые изображения можно использовать в качестве stories.
4) Удобный функционал. Каждый инструмент обладает набором характеристик, которыми можно управлять.
5) Интуитивно-понятный интерфейс. Разобраться в программе не составит труда, так как каждый элемент подписан. Именно эта понятность и простота использования приложения расширяет возрастную аудиторию (от детей до пенсионеров). Мне было важно, чтобы это приложение было действительно социальным и доступным даже пожилым людям, которым общение с близкими им людьми в первую очередь по-человечески жизненно необходимо. А не только приятное время препровождение и развлечение в часы досуга как у молодежи.

Внимание пользователей все чаще перетягивают мобильные решения. За телефоном они проводят больше времени, чем за компьютером и ноутбуком. На этом поле требуется простое приложение, которым можно было бы пользоваться одним пальцем. Не нужно больше сидеть с книжкой и инструкцией или смотреть обучающие видеоролики в интернете. Все устали от программ с нагромождённым функционалом, где больше 10% функций никогда не используются, а от обилия кнопок разбегаются глаза.

В отличие от других фоторедакторов, в моем приложении присутствует гибко-настраиваемый редактор градиентов с выбором направления, цветов и типа наполнения.

Мой редактор также выгодно отличается от сторонних приложений тем, что во вкладке “Текст” можно напечатать готовую умную мысль или цитату известного человека.

В программе присутствует возможность создать свой фильтр, воспользовавшись собственной матрицей свертки.

Также все стикеры разделены на группы по тематике, а последние использованные, чтобы не было путаницы, отображаются в отдельной вкладке.

Новшеством приложения стала лента. Готовым изображением можно поделиться с друзьями через социальные сети или выложить в ленту, которую я написал для приложения и ее видят все пользователи.

Polo кардинально отличается от других приложений своим весом. Если другие фоторедакторы весят 45+ МБ, Polo не достигает и 6. Память телефона не забивается, а сама скачка приложение происходит в разы быстрее.

Значит ли это, что у него меньше функций?

Вовсе нет!

1 Используемые методы

Мобильный фоторедактор polo был написан на языке программирования Java с использованием Android SDK в интегрированной среде разработки Android Studio. Для написания приложения был выбран именно этот язык из-за гибкой системы безопасности, инфраструктуры JVM, автоматического управления памятью, встроенных средств создания многопоточных программ и огромнейшего количества библиотек.

Google давно полюбил этот язык и именно поэтому решил сделать свою мобильную ОС на этом языке. Android построен на особом ядре Linux и переработанной виртуальной машине Java. Она и является прослойкой между приложением и ядром системы. В итоге, почти любое приложение, которое запускается там, должно быть написано на этом языке. Конечно, можно использовать и другие (Python, C++, JavaScript и даже PHP), но они будут работать медленнее. Также под эти задачи написано гораздо меньше библиотек.

Приложение состоит из нескольких окон: EditActivity, MainActivity, FeedActivity, FinalActivity и другие. Их подробное описание и назначение будет дано далее.

2 Приветственный экран

При открытии программы пользователю показывается сплошной экран с логотипом приложения. Стилем программы является Theme.AppCompat.Light.NoActionBar. Он убирает стандартный титул с названием, который занимает много места на экране. Фоном окна является фигура, описанная в файле shape.xml. В нем задаются форма, цвет, контур и прочие данные объекта. Благодаря этому приему на любом экране с любым разрешением картинка будет оставаться четкой и без пикселизации.

С помощью класса Android.os.Handler, который является следующей ветвью развития потоков, реализуется выполнение кода в некоторый момент в будущем. Также он используется для передачи кода, который должен выполняться в другом потоке. Переменная SPLASH_TIME_OUT со значением 2000 задает время отображения приветственного экрана равное двум секундам, так как оно указывается в миллисекундах.

После показа логотипа выполняется метод start(); из библиотеки imagepicker. Он позволяет отобразить в виде сетки все медиа файлы пользователя, а также моментально сделать новое с помощью камеры. Данное средство было выбрано из-за внешней

простоты в использовании и возможности сохранения в кэш миниатюр. Эта мера позволяет не создавать их каждый раз, нагружая телефон, а вынимать их из промежуточного буфера с быстрым доступом.

Возвращенные данные проверяются на корректность с помощью кодов состояния. Полученное изображение добавляется в списочный массив. Перед переходом в другое окно фоторедактор вкладывает путь к изображению в явное намерение, а принимающая активность должна уметь обработать ее. С помощью уникального ключа ей удастся вытащить содержимое.

3 Экран редактирования

Главным окном в приложении Polo является EditActivity, в котором происходит редактирование выбранной пользователем фотографии. При запуске оно с помощью метода getIntent().getExtras().getString(“path”); пытается сохранить в строковую переменную путь к оригинальному изображению. В случае удачной попытки, создается ее копия. Если путь не был получен, то пользователя возвращают на главный экран выбора файла.

Специальный метод принимает путь к изначальному файлу. Он получает характеристики экрана. В переменной scaleW и scaleH сохраняются отношения длин сторон дисплея и изображения. Для сохранения пропорции и правильного масштабирования умножаем

обе стороны на наименьший коэффициент. С помощью оригинального изображения и измененных размеров мы создаем новую картину. В конце возвращается переменная типа Bitmap.

Удобная реализация прокручиваемого меню была сделана с помощью библиотеки horizontalScrollMenuLibrary. Она позволяет быстро добавлять и удалять пункты, а также назначать им иконку и название. Все графические ресурсы хранятся в папке drawable. Реализованный в меню слушатель определяет нажатый элемент и выполняет характерный для него действия.

За отображение всех выполняемых пользователем процедур отвечает класс SurfaceView, который предоставляет объект Surface. Данное решение было выбрано из-за поддержки рисования в фоновом потоке. Оно отлично подходит для насыщенных графикой элементов, которые нуждаются в частых обновлениях. В методе surfaceCreated(); описываются все изменения, которые должны быть применены к поверхности для рисования.

4 “Кисть”

Рисование является основной функцией в редакторе. Пользователь чаще всего работает именно с этим инструментом. Кисть обладает рядом функций, которые позволяют видоизменять и настраивать ее. Меню кисти имеет следующие пункты:

1) Закрыть

2) Цвет
3) Толщина
4) Отменить
5) Ок

Все функции выделены в отдельный класс Brush.java. В нем все следы пользователя сохраняются в виде двух автоматически расширяемых массивах. Paths сохраняют фигуры, нарисованные человеком. Paints запоминает свойства кисти для каждого отдельного штриха.

При перемещении пальца по экрану вызывается метод brushMove();. В него мы передаем координаты касания по x и y. Все новые данные добавляются в конец. Приложение проходит циклом по всему списку и отображает все прямые, кривые и прочие линии. При касании и отпускании рисуется круг. Это позволяет создать округлые следы от кисти в соответствии с рисунком 1.

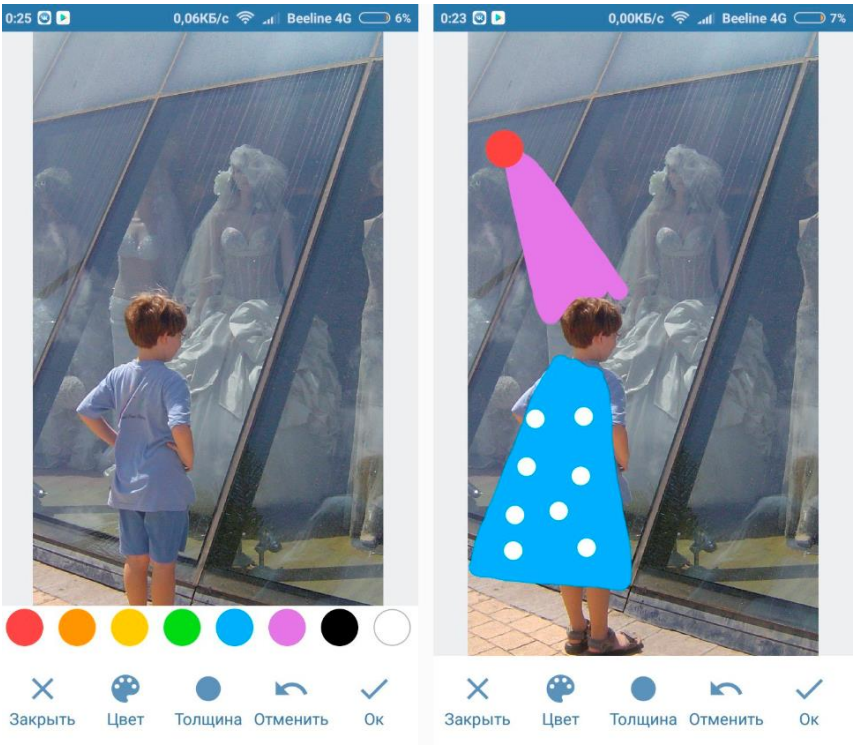


Рисунок 1 – Панель меню инструмента “Кисть” и пример обработки изображения

У пера можно поменять цвет и толщину. Для этого создано два дополнительных подменю. Пользователю доступно 8 базовых цветов. Их цвет передается в формате RGB. Каждая из трех переменных отвечает за один из каналов (красный, зеленый, синий) и может принимать значения от 0 до 255.

Если среди основного набора нет нужного цвета, то его можно взять из изображения с помощью пипетки. На холсте отрисовывается, залитый цветом выбранного участка, круг, который следит за пальцем пользователя. После отпускания новый цвет передается в прежний метод.

В меню выбора ширины кисти доступно 5 размеров. Их значение хранится в виде целого числа.

При закрытии панели все нанесенные метки остаются на экране, а последние использованные параметры пера сохраняются и не сбрасываются. В следующий раз не придется заново их устанавливать.

5 “Текст”

На редактируемом изображении можно оставлять текстовые сообщения. Это хороший способ передачи информации и своих чувств. Панель текста имеет следующие пункты:

1) Закрыть / Отменить
2) Цвет
3) Добавить
4) Шрифт
5) Мысли
6) Ок

Методы для работы с текстом находятся в отдельном классе Text.java. Сообщения пользователя сохраняются в четырех автоматически расширяемых массивах. Paints запоминает свойства каждой строки. Strings хранит их содержимое. Xs и ys указывает их расположение на экране.

К тексту мы применяем сглаживание и выравнивание по центру. На выбор доступно 10 различных шрифтов. От каллиграфических до готических. Все шрифты хранятся в подпапке fonts папки assets в формате ttf и otf. Настройки цвета идентичны тем, что используются в кисти.

В фоторедакторе Polo есть целый список мыслей и цитат великих людей. Пользователь может использовать любую из них и добавить ее на холст. Выбор происходит на отдельном экране. В нем размещён виджет ListView, который заполняет собой все окно. В его адаптер мы передаем текущий контекст, XML-разметку для отдельного элемента списка и массив данных в соответствии с рисунком 2.

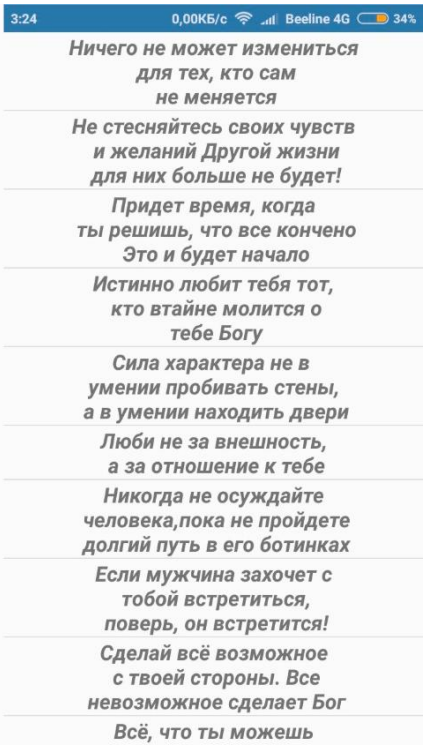


Рисунок 2 – Панель меню инструмента “Текст” и список готовых цитат

6 “Стикеры”

Пользователь может добавлять на изображение тематические картинки – стикеры. С их помощью можно сделать маску для человека или придать определённый окрас фотографии в соответствии с рисунком 3.

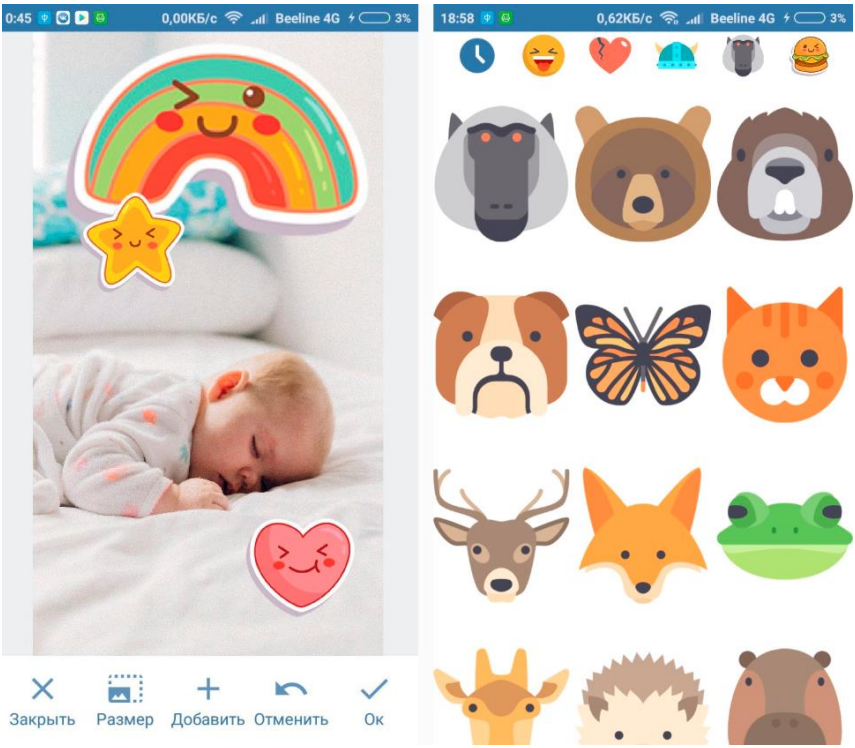


Рисунок 3 – Панель меню инструмента “Стикеры” и наборы готовых наклеек

Все функции выделены в отдельный класс Sticker.java. Каждая наклейка представляет собой объект этого класса со своим изображением, идентификатором и координатами.

Кнопка “Добавить” переносит пользователя в отдельное окно. В его верхней части находится панель переключения между наборами стикеров, разбитыми по тематикам: эмодзи, романтика, маски,

животные и еда. Также доступна умная вкладка с последними использованными изображениями. Благодаря экземпляру класса SharedPrefernces данные не пропадут даже после закрытия приложения. Список будет находиться в файле, доступ к которому будут иметь только компоненты моего приложения.

Виджет GridView позволяет расположить в удобной сетке все переданные ему элементы. Для работы с ним был создан уникальный GridAdapter, наследуемый от BaseAdapter. На вход он получает контекст и массив целочисленных чисел. Каждое число является идентификатором изображения в ресурсах.

Перемещая ползунок пальцем по экрану, пользователь можем менять размер стикера. Изменение параметра отслеживается с помощью метода onProgressChanged();. Новое значение передается в класс Sticker.java. Например, когда ширина больше высоты, в нем высчитывается новый размер изображения по формуле (1)

$$outWidth = postSize; \quad (1)$$

$$outHeight = (inHeight * postSize) / inWidth; ,$$

где outWidth – новая ширина;

outHeight – новая высота;

inWidht – прежняя ширина;

inHeight – прежняя высота;

postSize – размер, переданный пользователем.

7 “Фильтры”

Приложение Polo умеет накладывать 20 различных фильтров. С их помощью можно преобразовать изображение, придать ему стиля. Также пользователю доступна матрица свертки, в которую он может вбить самостоятельно данные для обработки.

Для инверсии, Ч-Б и сепии применяется класс PictureThread.java. В нем у каждого эффекта есть собственный ColorMatrix. Данная матрица заполняется по принципу указанному в таблице 1.

Таблица 1 – Фильтр для изменения цвета, использующий значения 4x5 матрицы

rR	rG	rB	rA	rT
gR	gG	gB	gA	gT
bR	bG	bB	bA	bT
aR	aG	aB	aA	aT

Новый красный канала будет высчитываться, как значения исходного цвета (R, G, B, A), перемноженные на первые четыре значения (rR, rG, rB, rA) из первой строки. К результату добавляется rT. С остальными каналами также. Данная матрица устанавливается в качестве параметра для ColorMatrixColorFilter.

Выполнение других фильтров происходит одновременно в пяти потоках. Реализация каждого скрыта в классе Filter.java. Программа проходит попиксельно изображение, меняя его согласно двумерному массиву или с помощью математических операций в соответствии с рисунком 4.

Рисунок 4 – Панель меню инструмента “Фильтры” и пример обработки изображения

Матрица свертки – это матрица коэффициентов, которая “умножается” на значение пикселей изображения для получения требуемого результата. На вход она получает матрицу 3 на 3 (классический размер), а также значения div(делитель) и offset(коэффициент). Результат преобразования делится на div и нему прибавляется offset.

8 “Уровни”

С помощью уровней можно тонко отредактировать изображение. Ползунок может увеличить или ослабить силу любого из каналов, яркость и насыщенность. Данный инструмент использует тот же класс PictureThread.java, что и фильтры. При вызове setLevel(); передается номер эффекта, который будет применен. Конструкция switch case выбирает нужную матрицу и устанавливает ее в соответствии с рисунком 5.

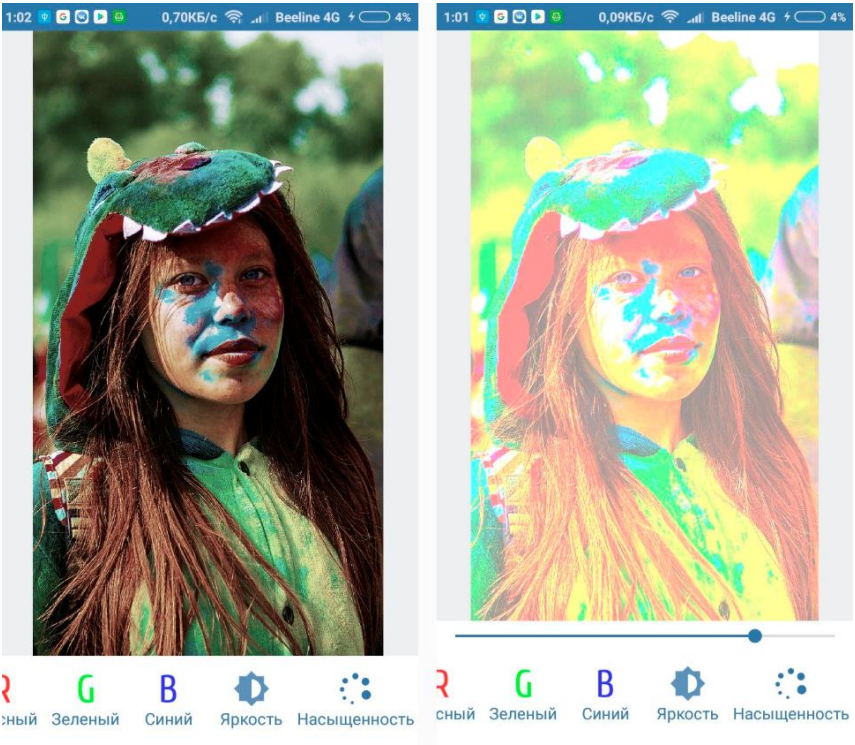


Рисунок 5 – Панель меню инструмента “Уровни” и пример обработки изображения

9 “Дизайн”

У данного пера очень богатый функционал. В отличие от кисти, дизайн рисует не цветом, а узорами и градиентом. Пользователь может выбрать любой из 8 готовых иллюстраций и 5 градиентов. Изображения хранятся в подпапке drawable папки res. Для преобразования ресурса в Bitmap мы используем метод decodeResource();. Он получает доступ к ресурсам и вынимает указанный элемент.

Расширения класса Shader позволяют создавать объекты Paint, которые закрашивают элементы более сложным образом, чем заливка сплошным цветом. Чтобы использовать их при рисовании, следует применить к объекту Paint метод setShader(); Наш шейдер будет состоять из узора, повторяющегося по горизонтали и по вертикали.

Для градиентов используется LinearGradient. В его конструкторе указываются координаты линии, по которой градиент будет идти, а его размер станет равным длине этой линии. Также передается массив цветов и режим отображения. Цвета задаются целыми числами. Как положительными, так и отрицательными.

Если среди существующих градиентов не найдется подходящий, то можно создать свой собственный в редакторе. С

помощью трех ползунков (красный, зеленый и синий каналы) можно точечно выбрать нужный цвет. Создаваемый оттенок отображается в верхнем окне. При нажатии кнопки “Добавить” его код добавляется в целочисленный массив. Готовый градиент отображается ниже.

В дополнительном меню можно выбрать тип и направление градиента. Цвета могут зеркально повторяться или следовать друг за другом один раз. Превью цвета и градиента представляют собой ImageView с наложенным через метод setImageDrawable(); фоном в соответствии с рисунком 6.

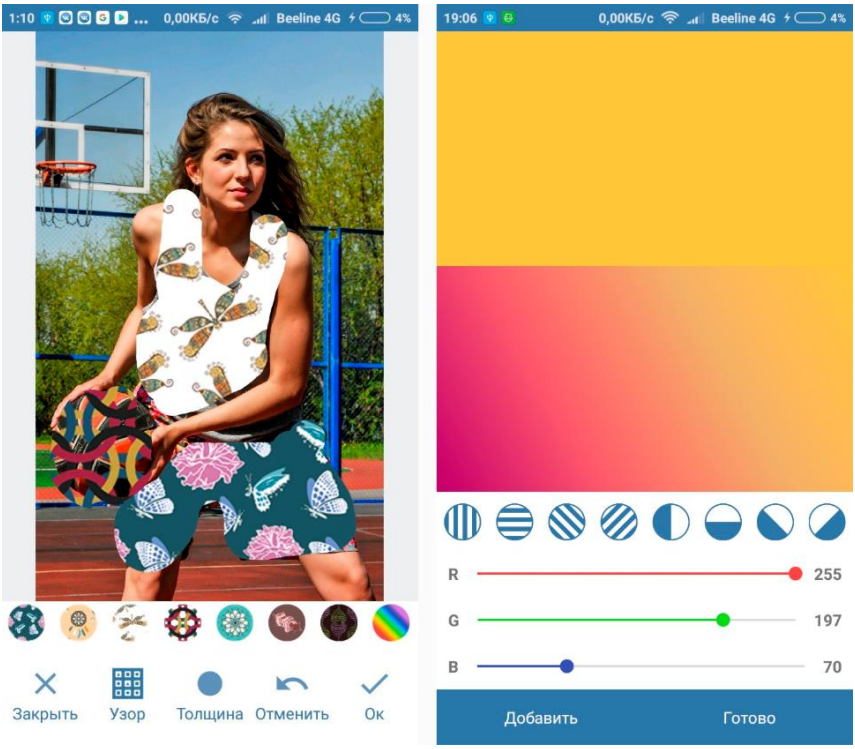


Рисунок 6 – Панель меню инструмента “Дизайн” и редактор градиентов

10 “Микс”

В меню “Микс” пользователь может воссоздать свое изображение из набора элементов или добавить ему глитч-эффектов и помех. Фотографию можно составить из текста, точек и геометрических фигур.

Мы разделяем изображение на блоки с равными сторонами. Программа вычисляет средний цвет всех пикселей на полученном участке и запоминает этот оттенок. Дальше в качестве фона устанавливается сплошной черный цвет и рисуется элемент установленного цвета в соответствии с рисунком 7.

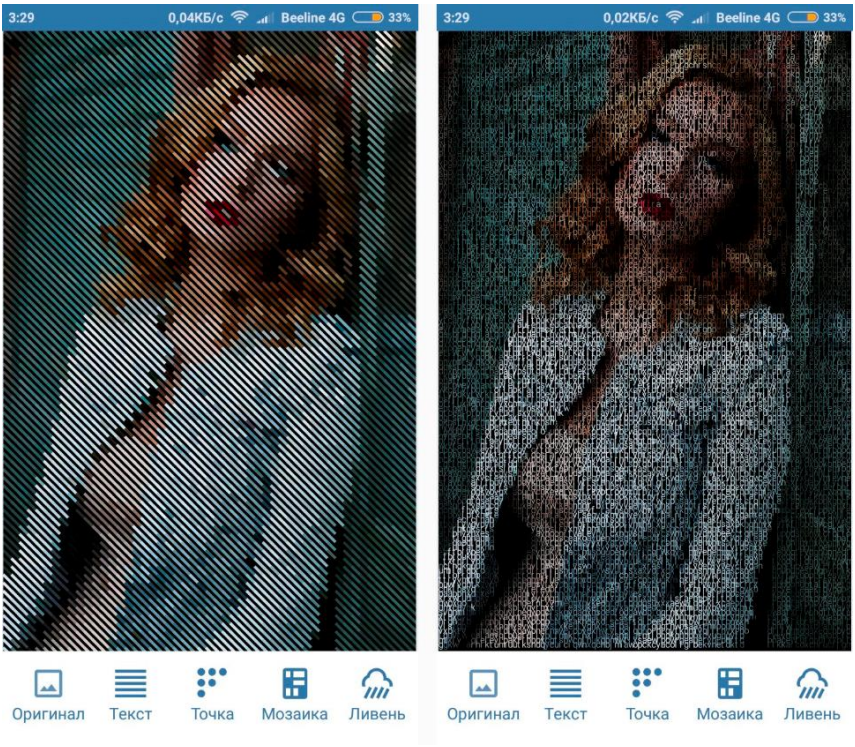


Рисунок 7 – Панель меню инструмента “Микс” и пример обработки изображения

Например, для генерации случайного символа я использую рандомное число, полученное с помощью метода random nextInt(25);. Он возвращает целое число в диапазоне от 0 до указанного числа, которое в этот промежуток не входит. Чтобы получить числовой код строчной буквы латинского алфавита, в конце мы увеличиваем получившиеся число на 97 (в таблице ASCII литеры латиницы стоят с 97 по 122 место). Если присвоить это значение переменной типа char, то она сохранит в себе символ под этим порядковым номером.

Для создания глитч-эффекта программа в цикле со случайным количеством выполнений вставляет поверх оригинальной фотографии вырезанные из нее прямоугольники. Также присутствует возможность создания “Штрих-кода”. Polo следит за y-координатой касания пальца. Все пиксели, находящиеся на том же уровне, растягиваются до самого низа картины.

11 Онлайн-лента публикаций

Свои работы пользователи могут в любое время выложить в онлайн-ленту публикаций. Достаточно указать только никнейм и содержание. Приложение связывается с сервером с помощью библиотеки OkHttp. Загрузка файла происходит в нескольких потоков. На сервере uploadImages.php сохраняет файл в формате png. Автоматически записываются данные о фотографии, пользователе и дате публикации. Приложение умеет работать со всеми языками мира, смайлами и эмоджи.

При формировании ленты приложение обращается к getImagesPolo.php. Он создает строку в формате JSON, которая имеет ссылку на оригинал изображения, никнейм, содержание и дату. Для создания особой выборки фотографий используются два значения: count и offset. Первый отвечает за количество возвращаемых объектов, а второй за смещение от первой фотографии.

Фоторедактор Polo получает эту строку и с помощью библиотеки GSON от Google превращает ее в удобный для работы массив данных. Вся полученная информация делится на блоки, после чего вставляется в свой контейнер и выводится на экран. В качестве контейнера используется ListView, который заполняет собой все окно. Для него был разработан новый адаптер и единичный элемент. Он включает в себя: никнейм, изображение, описание, дату и время выкладывания. Доступ к элементам происходит в адаптере через LayoutInflater Публикация новой записи и просмотр онлайн-ленты происходит в соответствии с рисунком 8.

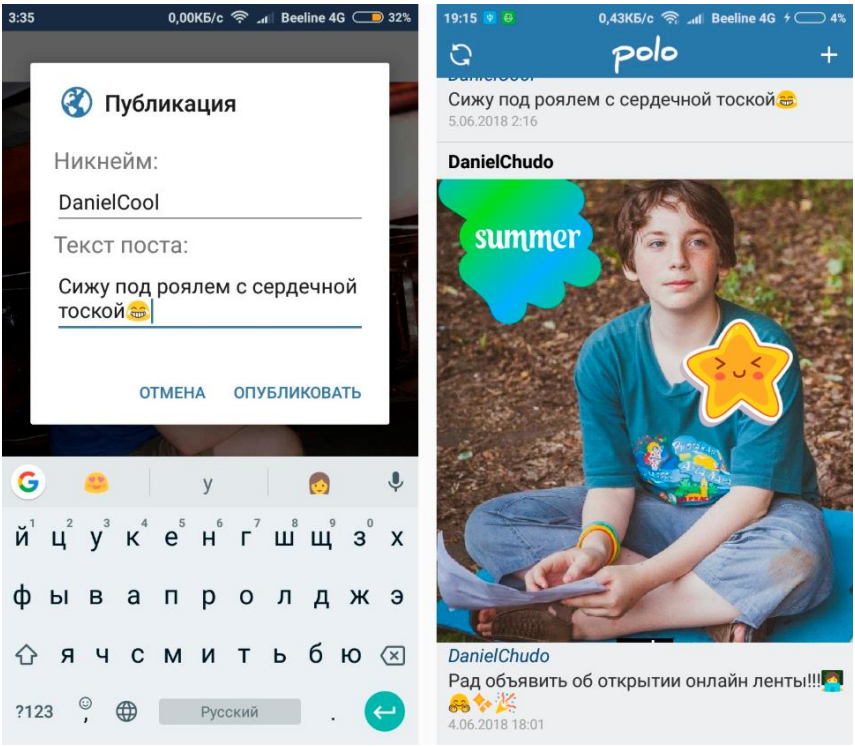


Рисунок 8 – Форма создания публикации и онлайн-лента с постами

Изображение загружается в пост с помощью библиотеки Glide. Мы даем ему ссылку на картинку, и он устанавливает ее в назначенное место. Также фотография в посте центрируется и обрезается. Если нажать на пост, то откроется оригинал изображения. Для этого используется всплывающий AlertDialog с нулевыми отступами. Все его содержимое занимает ImageView.

В панели ленты доступны две кнопки: обновить страницу и добавить новый материал. Содержимое листается плавно, без задержек из-за кэширования уже ранее открытого материала.

12 Сравнение Polo и Retrica

Если смотреть на сторонние редакторы, то можно заметить, что многие из них весят более 50 МБ. Они занимают ценное место на телефоне, сильно нагружают оперативную память. Часто эти приложения используют уже готовые решения в виде написанных другими людьми библиотек. В крупных проектах их количество трудно сосчитать. Но часто сама программа использует только пару методов из них. В итоге более 90% существующих в них функций никогда не используется, увеличивая конечный вес приложения.

В качестве примера для сравнения я взял популярный среди молодежи фоторедактор Retrica. Сравнение двух приложений можно увидеть в таблице 2.

Таблица 2 – сравнительная характеристика Polo и Retrica

Название	Polo	Retrica
Вес (МБ)	5.8	45.29
Редактор градиентов	Есть	Нет

Паттерны для рисования	Есть	Нет
Share в социальных сетях	Есть	Есть
Пользовательская матрица свертки	Есть	Нет
Онлайн лента публикаций	Есть	Нет
Настраиваемые кисти	Есть	Есть
Готовые цитаты и стихи	Есть	Нет
Пипетка у кисти и цвета	Есть	Нет
Разнообразные фильтры	Есть	Есть

Сравнительный анализ наглядно показывает преимущества Polo. Разумеется у Retrica есть удобные функции, но у Polo их больше, включая более широкий набор инструментов и значительно более легкий вес.

Заключение

Работать над созданием приложения было интересно и познавательно. В итоге у меня получалось создать полноценный фоторедактор, который бы удовлетворял желания пользователя.

Были добавлены такие функции как: стикеры, текст, кисть, фильтры, уровни, дизайн, микс, share в социальные сети и онлайн-лента публикаций.

Приложение работает в связке с веб-сервером. Именно это позволило создать ленту с фотографиями пользователей.

В будущем хотелось бы расширить возможности приложения. Помимо фотографий можно было бы создавать собственные короткие анимации. А видео получалось бы при добавлении аудио - дорожки к ним. Музыку можно будет выбирать из трек-листа пользователя из его любимой социальной сети или создать самому из набора готовых сэмплов.

Также ясно, что увлеченному пользователю всегда будет мало предоставленных функций, поэтому появится возможность создания собственных алгоритмов для фильтров и стикеров. Последние можно будет искать в поисковой выдаче google, не покидай приложение.

В идеале планируется создать систему, где в формате drag'n'drop можно было бы перемещать блоки с действиями и составлять готовые сценарии для редактирования.

В ходе данной исследовательской работы были рассмотрены и проанализированы разные алгоритмы преобразования фотографий,

налажена отправка и получение данных по интернету и получена плавная работа программы.

Фоторедактор Polo получил Золотой диплом IT школы Samsung и первое место в номинации “Социальное приложение”. В том же 2018 году он получил Гран-При на конкурсе “IT школа Samsung выбирает сильнейших”. Все это вдохновляет меня на продолжение разработки по пути усовершенствования продукта.

Список использованной литературы

1 SurfaceView [Электронный ресурс] - http://developer.alexanderklimov.ru/android/views/surfaceview.php
2 Список с множественным выбором и собственным адаптером ArrayAdapter [Электронный ресурс] - http://developer.alexanderklimov.ru/android/listview/multiselect_customarrayadapter.php
3 Диалоговое окно AlertDialog [Электронный ресурс] - http://developer.alexanderklimov.ru/android/alertdialog.php
4 OkHttp [Электронный ресурс] - http://square.github.io/okhttp/
5 Android OkHttp Example [Электронный ресурс] - http://www.zoftino.com/android-okhttp-example
6 Работа с JSON [Электронный ресурс] - http://java-help.ru/android-json/

7 JSON: основы использования [Электронный ресурс] - https://ruseller.com/lessons.php?id=1212&rub=28

8 Библиотека GSON [Электронный ресурс] - http://developer.alexanderklimov.ru/android/library/gson.php

9 Glide [Электронный ресурс] - https://github.com/bumptech/glide

10 Android Glide Example [Электронный ресурс] - https://www.dev2qa.com/android-glide-example/

11 Матричные фильтры обработки изображений [Электронный ресурс] - https://habr.com/ru/post/142818/

12 Графические фильтры на основе матрицы скручивания [Электронный ресурс] - http://wb0.ru/articles/120.htm

13 ColorMatrixColorFilter [Электронный ресурс] - https://developer.android.com/reference/android/graphics/ColorMatrixColorFilter

СЕКЦИЯ III. Информатика и системы управления

Умный аранжировщик

Шумнов Пётр Алексеевич

г. Москва, ГБОУ Школа №1533 «ЛИТ»

Научный руководитель: Завриев Николай Константинович, ГБОУ Школа №1533 «ЛИТ», учитель по информатике

Цель проекта – разработать программу, которая бы позволяла автоматически генерировать аккомпанемент к гитарной партии. В частности, писала бы партии для барабанов и бас-гитары, требуя при этом от пользователя лишь загрузить свою гитарную партию и выбрать жанр и стиль. Программа может быть использована для создания «минусовок» для игры или же для того, чтобы музыкант мог оценить, как придуманная им партия будет звучать в различных аранжировках.

Так как задача носит творческий характер, было решено использовать методы машинного обучения. Было опробовано более 60 моделей, различающихся методом преобразования входных и выходных данных, типом нейронной сети, количеством слоев и нейронов. На данный момент наилучший результат достигнут при помощи нейронной сети с одним скрытым LSTM-слоем с 50 нейронами. В этой модели партии инструментов представлены как последовательности нот, и каждый такт обрабатывается отдельно.

Конечная точность (accuracy) нейронной сети на валидационной выборке – 80%. Однако этот показатель не позволяет оценить качество работы программы. Дело в том, что данный метод вычисления точности предполагает для каждого входного вектора – партии гитары – единственно верный правильный выходной вектор –

партии барабанов и бас-гитары. Соответственно, оценивается приближенность сочиненной нейросетью партии к некому правильному ответу. Однако задача сочинения музыки творческая, «правильных» вариантов аранжировки огромное множество. Так что оценить результат может только человек на слух.

На данный момент программа способна генерировать партии барабанов и бас-гитары в стилях «рок» и «рок`н`ролл».

Оглавление

1 Введение
2 Постановка задачи
3 Актуальность и целевая аудитория
4 Обзор аналогов
5 Математические модели и алгоритмы
6 Обученные модели
7 Программная реализация
8 Результаты
9 Заключение
10 Список литературы
11 Приложение А
12 Приложение Б

Введение

В настоящее время из-за многих факторов (например, повышение доступности музыкальных инструментов) очень распространена музыкальная деятельность. Научиться играть на музыкальном инструменте при наличии интернета совсем нетрудно и не затратно. При этом одним из самых распространенных инструментов в мире является гитара, а программное обеспечение для гитаристов уже давно стало многомиллионной индустрией и включает в себя огромное количество разного рода приложений.

Однако существующие программы автоматического аккомпанемента для гитары имеют значительные недостатки. В данном проекте применяется принципиально новый в этой области подход – использование машинного обучения для создания аккомпанемента с нуля. Среди аналогов присутствует программа, в которой нейронная сеть подбирает для каждого такта композиции наиболее подходящий шаблон. Однако этот метод не позволяет создавать ничего кроме барабанов, в то время как модель, применяемую в данном проекте, можно обучить играть на любом инструменте в любом стиле.

Постановка задачи

Цель проекта – разработать программу, автоматически создающую партии для барабанов и бас-гитары на основе партии гитары. Предполагается, что данная программа будет использоваться вместе с табулатурным редактором Guitar Pro, так как она работает с форматами этого приложения.

На вход программа получает файл формата Guitar Pro любой из трех версий: .gp3, .gp4, .gp5 – содержащий некоторое число дорожек с табулатурами/нотами. Далее пользователь имеет возможность добавить произвольное число новых дорожек и сохранить файл в одном из перечисленных выше форматов.

При добавлении новой дорожки пользователь должен указать следующие ее параметры:

- «Родительскую» дорожку – ту, на основе которой будет генерироваться новая
- Тип: барабаны или бас-гитара
- Желаемый музыкальный жанр/стиль
- Имя

Актуальность и целевая аудитория

Табулатура – аналог нотной записи музыки, широко распространенный среди гитаристов. Guitar Pro является одним из самых популярных табулатурных редакторов, предоставляющих пользователю огромный функционал для чтения, воспроизведения и редактирования как табулатур, так и нот.

Целевая аудитория проекта – гитаристы, в частности пользователи Guitar Pro и его аналогов (большинство подобных программ поддерживают формат файлов Guitar Pro). Данный проект может быть использован несколькими способами:

- При помощи программы гитарист может создать «минусовку» для игры.
- Если музыкант придумал гитарную партию, он может посмотреть, как она будет звучать в различных аранжировках.
- В дальнейшем при улучшении качества возможно третье применение: создание аранжировки для конечного продукта. При таком использовании могут помочь VST-плагины, способные «оживить» стандартный звук midi и сделать его приемлемым для конечной записи песни.

Обзор аналогов

Данный проект имеет множество аналогов, однако значительно отличается от них как областью применения и целевой аудиторией, так и реализацией.

Все аналоги можно разделить на три группы:

1. **Ручные**, например:
    - Guitar Pro.
    - TuxGuitar.

    В основном это редакторы табулатур или нот. Все партии там создаются вручную и никакой автоматической генерации они не включают. Однако, они могут являться аналогами, т.к. имеют отчасти похожие с данным проектом цели - создание партий инструментов. Тем не менее, это наиболее отдаленные аналоги проекта.

2. **Полуавтоматизированные**, например:
    - LA Scoring Strings.
    - Band-in-a-Box.

    В этих программах процесс создания аккомпанемента автоматизирован лишь частично: пользователю требуется ввести в программу последовательность аккордов, которые уже будут обыгрываться автоматически.

    Такой подход имеет некоторые минусы:

        - Пользователю требуется умение гармонизировать мелодию.

- Пользователю придется тратить на этот процесс время.
- Результат всегда будет представлять собой последовательность аккордов.

Также, не было найдено ни одной подобной программы, позволяющей генерировать ударные.

3. **Автоматизированные**, например:
   - Microsoft SongSmith.
   - «Automatic Accompaniment Generation with Seq2Seq».
   - GarageBand

Этот вид аналогов ближе всего к данному проекту. Было найдено всего 3 подобные программы. Они позволяют полностью автоматически создавать аккомпанемент, однако тоже имеют ряд отличий от данного проекта. Стоит подробно рассмотреть каждый из них.

Microsoft Songsmith - это программа для создания аккомпанемента к вокалу, и работает она только с ним. Отсутствует возможность генерации барабанов, так как вокал изначально записывается под один из шаблонов ударных. Так же, результат опять же представляет собой только обыгровку аккордов.

«Automatic Accompaniment Generation with Seq2Seq» – это проект Калифорнийского университета в Беркли, основанный на работе Стэнфордского университета. Целью этого проекта было проверить, могут ли нейросети генерировать аккомпанемент к мелодии. Можно сказать, что

ответ был получен положительный, однако результат очень примитивен, в особенности ударные.

И, наконец, GarageBand – программа от Apple, которая позволяет автоматически генерировать барабаны. О недостатке данного ПО говорилось во введении – программа лишь подбирает наиболее подходящие «сэмплы» (шаблоны барабанных партий). Таким образом можно сгенерировать только ударные.

После анализа аналогов был выявлен ряд преимуществ данного проекта:

- Полная автоматизация, требующая от пользователя только выбора музыкального жанра.
- Возможность создания партии ударных.
- Гибкость применяемого метода – возможность обучить модель сочинять партии любых инструментов в любом жанре.

Математические модели и алгоритмы

В данном проекте было решено использовать методы машинного обучения, так как задача носит творческий характер. Сначала было опробовано использовать многослойный перцептрон, однако не удалось добиться приемлемого результата. Поэтому следующим методом было глубокое обучение, в частности, рекуррентные нейронные сети (РНС).

Для применения РНС входные данные и выходные данные – партии инструментов – представляются в виде последовательностей нот. Данная модель называется (sequence-to-sequence, последовательность к последовательности). Подробное описание принципа работы РНС представлено в приложении А.

Несмотря на все преимущества, которые дает возможность «запоминать» информацию, у РНС есть большой недостаток: проблема долговременных зависимостей. Нейроны хорошо «помнят» недавно полученную информацию, но не имеют возможности надолго сохранить в памяти что-то, что обработали много циклов назад, какой бы важной та информация ни была. Для решения этой проблемы используют частный вид РНС – LSTM (Long short-term memory, долгая краткосрочная память). Это особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям.

В данном проекте LSTM-сети показали себя лучше, чем классические РНС. В приложении Б представлено описание принципа работы LSTM-сети.

Обученные модели

На данный момент для генерации барабанов было опробовано 66 моделей, различающихся входными/выходными данными, типом нейросети и количеством слоев и нейронов. Самым главным параметром является способ преобразования входных и выходных данных – партий инструментов – в векторы, пригодные для нейросети. По данному параметру все 66 моделей можно разделить на 5 общих категорий:

1. Самый простой и очевидный вариант, который на данный момент и показал себя лучше всех. Использованы LSTM-сети, за единицу последовательности взята одна нота и каждый такт песни анализируется отдельно и независимо. Также, для этого варианта пробовалось использовать многослойный перцептрон.
2. Аналогичный вариант, но анализируется сразу вся песня – последовательность состоит из нот не одного такта, а всего произведения.
3. Снова LSTM-сети, но за единицу последовательности взята не нота, а целый такт. Песня анализируется целиком.
4. Задача делится на 2 части: сначала по партии гитары первая нейросеть выдает только ритм, а затем вторая по этому ритму пишет партию ударных. Это была попытка упростить задачу, ведь нейросети не нужно будет учиться и выделять ритм, и писать партию конкретно барабанов. Было опробовано брать за единицу последовательности как ноту, так и такт, как в первом

и третьем вариантах. Для каждой из двух нейросетей были опробованы 2 типа: LSTM и многослойный перцептрон.

5. Партия барабанов разделялась на партии каждого барабана отдельно, и обучалось 6 нейросетей, каждая училась играть на определенном барабане. Далее по определенному алгоритму эти партии объединялись.

Также, были опробованы разные способы описания ноты. В конечном итоге гитарная (входная) нота описывается 3 параметрами:

- Нота/пауза – логическая переменная
- Количество сыгранных струн. Введение этого параметра позволило определить, была сыграна отдельная нота или аккорд, что сильно улучшило результат.
- Длительность ноты.

Добавлялись такие параметры, как:

- Высота ноты.
- Разность высот текущей и предыдущей нот.

Нота барабанов описывает 7 параметрами: первый – длительность, каждый из следующих 6 отвечает за какой-либо барабан и равен 1, если удар в барабан был, и 0, если нет.

Для каждого их этих вариантов варьировалось также количество слоев и нейронов в сети. Были попытки заменять LSTM на GRU и классическую RNN, но к улучшению результатов это не привело.

В большинстве случаев нейронная сеть почти не обучалась (находила лишь самые простые закономерности, как, например, выделение первой доли), либо работала хуже первого варианта.

Для бас-гитары входные данные преобразуются следующим образом: все ноты «переносятся» в одну октаву, далее нейросети даются те же параметры нот, что и нейросети для барабанов, с добавлением высоты (число от 0 до 11 – в пределе одной октавы). Выходная же нота имеет 3 параметра: нота/пауза, длительность и высота ноты.

Программная реализация

Для разработки программы использовался язык python, так как он считается основным языком для работы с нейросетями. На рис. 1 изображена схема компонентов программы.

Программу можно условно разделить на 3 блока:

1 Графический интерфейс пользователя.

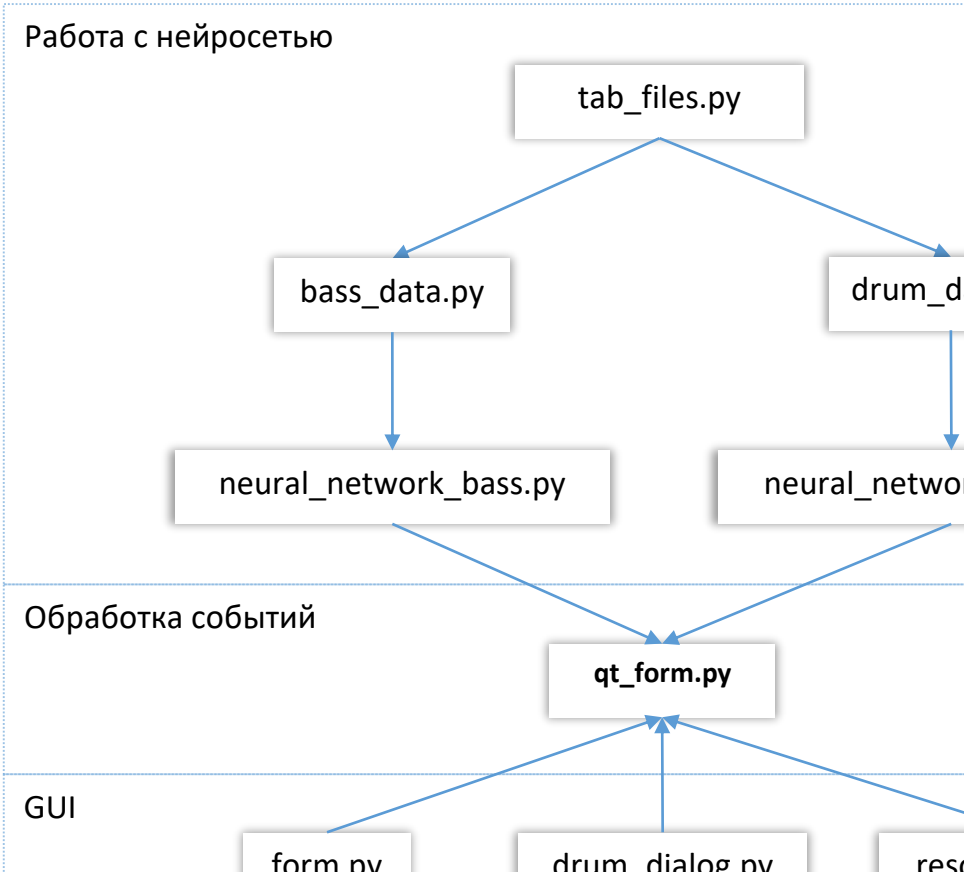


Рис. 1: компоненты программы

При создании графического интерфейса использовались среда Qt Designer для разработки дизайна и библиотека pyqt5 для импорта и работы с ним в python-коде.

В данный блок можно включить 3 файла, которые были сгенерированы автоматически при экспорте из Qt Designer:

- Form.py – файл с дизайном основной формы.
- Drum_dialog.py – файл с дизайном диалогового окна создания новой дорожки.
- Resources_cs.py – файл с ресурсами.

2 Обработка событий.

В этом блоке описаны все взаимодействия пользователя с программой. Он состоит из одного файла – qt_form.py – который запускается первым при открытии программы.

3 Работа с нейросетью.

Самая сложная часть программы, на создание которой ушла подавляющая часть времени работы над проектом. Блок включает 5 файлов:

- Tab_files.py – файл, в котором описаны классы для работы с табулатурами (см рис. 15).
- Bass_data.py и Drum_data.py – файлы, в которых происходит подготовка входных данных (как для обучения, так и для валидации и тестирования) и преобразование выхода нейросети в табулатуру.
- Neural_network_bass.py и Neural_network_drums – файлы с кодом нейросетей (для бас-гитары и барабанов соответственно).

Список используемых библиотек:

- Keras + TensorFlow для реализации нейросети.
- PyGuitarPro для чтения файлов табулатур.
- Numpy для работы с массивами.
- MsgPack для сериализации данных.

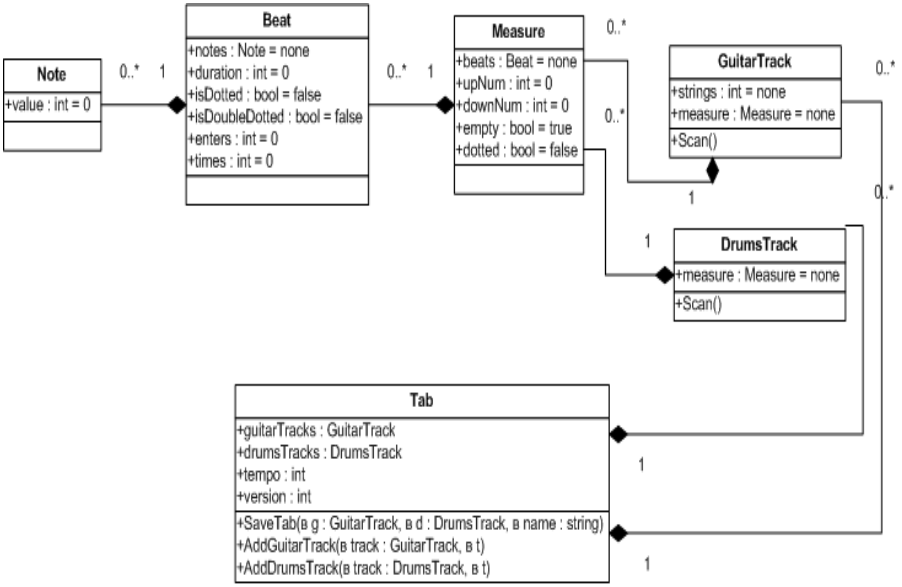


Рис. 15: диаграмма классов файла tab_files.py

Результаты

Поставленная задача была реализована частично: реализована генерация барабанов и бас-гитары в двух жанрах: «рок» и «рок`н`ролл».

Однако благодаря гибкости разработанной системы, для добавления новых жанров требуется лишь обучить уже существующую модель на новых данных и поместить файл с весами в папку с проектом.

На рис. 2 изображены 2 графика: синий – accuracy (точность), оранжевый – loss (ошибка). По горизонтальной оси отложены эпохи от начала обучения. Видно, что ошибка падает, что является основным признаком успешного обучения нейросети.

Итоговая точность сети ~80% для бас-гитары и ~65% для барабанов. Однако эта оценка весьма условна, так как задача носит

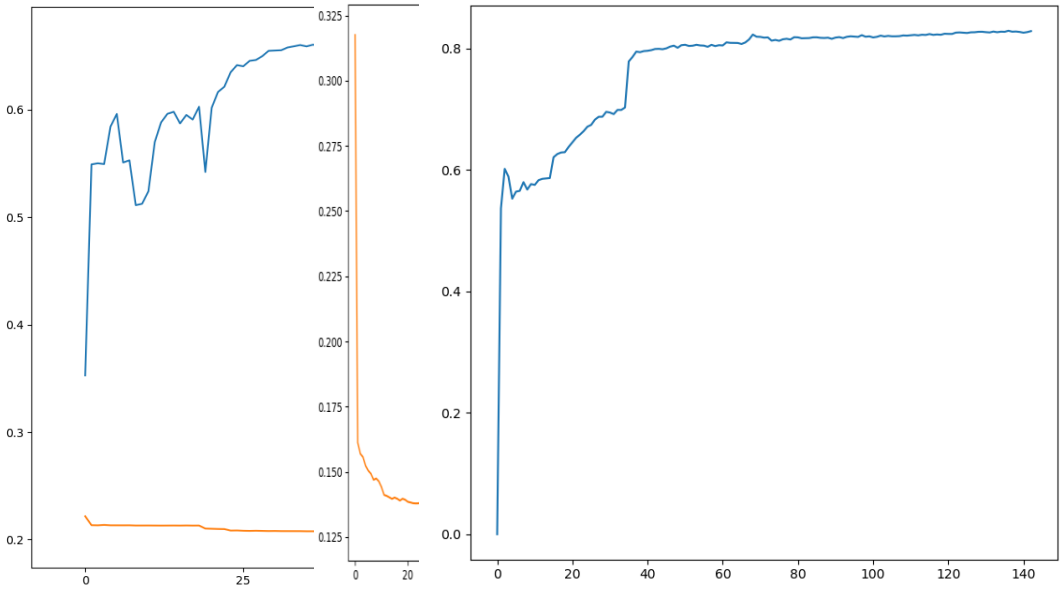


Рис. 2: график loss (ошибки) – оранжевым и accuracy (точности) - синим

творческий характер. Для одной партии гитары может быть сочинено огромное количество разных удовлетворительных партий барабанов, в то время как подобные методы предполагают только один правильный

ответ. Таким образом, оценить работу программы может только человек на слух.

Заключение

По итогам проделанной работы можно сказать, что нейронные сети определенно способны справиться с поставленной задачей. Предложенный метод превосходит по возможностям используемые в аналогах и имеет большой потенциал. В дальнейшем результат может быть значительно улучшен благодаря более оптимальному подбору параметров нейросети, корректировке формата входных и выходных данных и увеличению времени обучения.

Список литературы

1 Тадеусович, Р. Элементарное введение в технологию нейронных сетей с примерами программ / Р. Тадеусович. и др. – : Горячая Линия - Телеком, 2011. – 408 с.

2 Melody-to-Chord using paired model and multi-task learning language modeling [Электронный ресурс]. – Режим доступа: https://web.stanford.edu/class/cs224n/reports/2742800.pdf, свободный. – Загл. с экрана.

3 Automatic Accompaniment Generation with Seq2Seq [Электронный ресурс]. – Режим доступа: http://qihqi.github.io/machine/learning/music-generation-using-rnn/, свободный. – Загл. с экрана.

4 Нейронные сети для начинающих. Часть 1 [Электронный ресурс]. – Режим доступа: https://habr.com/post/312450/, свободный. – Загл. с экрана.

5 Нейронные сети для начинающих. Часть 2 [Электронный ресурс]. – Режим доступа: https://habr.com/post/313216/, свободный. – Загл. с экрана.

6 LSTM – сети долгой краткосрочной памяти [Электронный ресурс]. – Режим доступа: https://habr.com/company/wunderfund/blog/331310/, свободный. – Загл. с экрана.

7 Библиотеки для глубокого обучения: Keras [Электронный ресурс]. – Режим доступа: https://habr.com/company/ods/blog/325432/, свободный. – Загл. с экрана.

8 Understanding LSTM Networks [Электронный ресурс]. – Режим доступа: http://colah.github.io/posts/2015-08-Understanding-LSTMs/, свободный. – Загл. с экрана.

9 Keras documentation [Электронный ресурс]. – Режим доступа: https://keras.io, свободный. – Загл. с экрана.

10 PyGuitarPro documentation [Электронный ресурс]. – Режим доступа: https://pyguitarpro.readthedocs.io/en/stable/index.html, свободный. – Загл. с экрана.

Приложение А

Рекуррентные нейронные сети

На рисунке 3 изображена схема РНС с одним нейроном, получающей на вход последовательность x_t и дающая на выход последовательность h_t. Рекуррентные сети, в отличие от обычных, имеют обратную связь: синапс, который передает информацию от одного шага сети к другому.

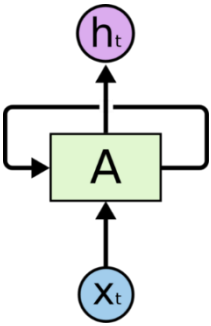


Рис. 3: рекуррентная нейронная сеть с 1 нейроном

Если «развернуть» обратную связь, получим рисунок 4.

На каждой итерации внутренний слой нейронов получает на вход новый член последовательности x_t и информацию о предыдущем состоянии внутреннего слоя A, на основании чего генерирует новый член последовательности h_t. Таким образом в сети реализуется «память», что принципиально меняет характер ее работы и позволяет анализировать любые последовательности данных, в которых важно, в каком порядке идут значения.

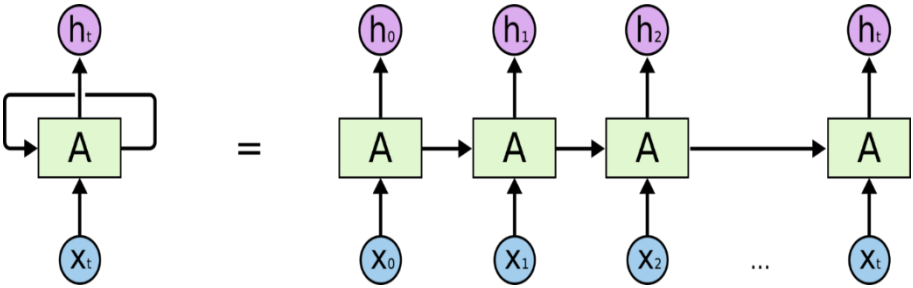


Рис. 4: рекуррентная нейронная сеть в развертке

Приложение Б

LSTM

Любая рекуррентная нейронная сеть имеет форму цепочки повторяющихся модулей нейронной сети (см. рис. 5). В обычной РНС структура одного такого модуля очень проста, например, он может представлять собой один слой с функцией активации tanh

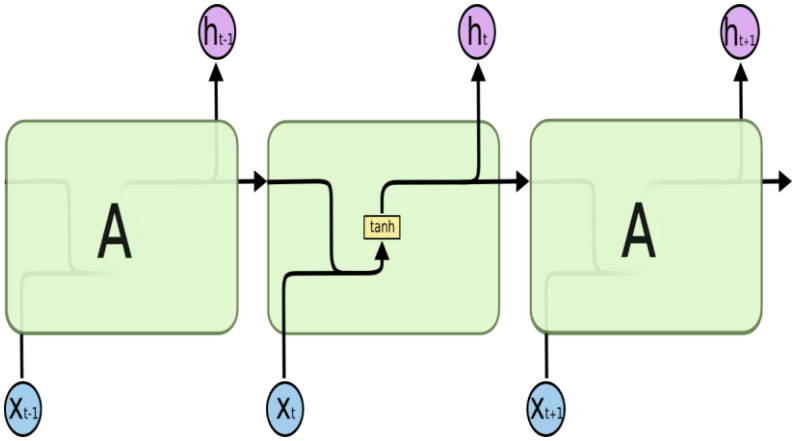


Рис. 5: слой рекуррентной нейронной сети

(гиперболический тангенс).

Структура LSTM также напоминает цепочку, но модули выглядят иначе (рис. 6). Вместо одного слоя нейронной сети они содержат целых четыре, и эти слои взаимодействуют особенным образом.

На рисунке 7 показаны специальные обозначения, используемые на схемах (слева направо: слой нейронной сети, поточечная операция, векторные перенос, объединение, копирование).

Ключевой компонент LSTM – это состояние ячейки (cell state) –

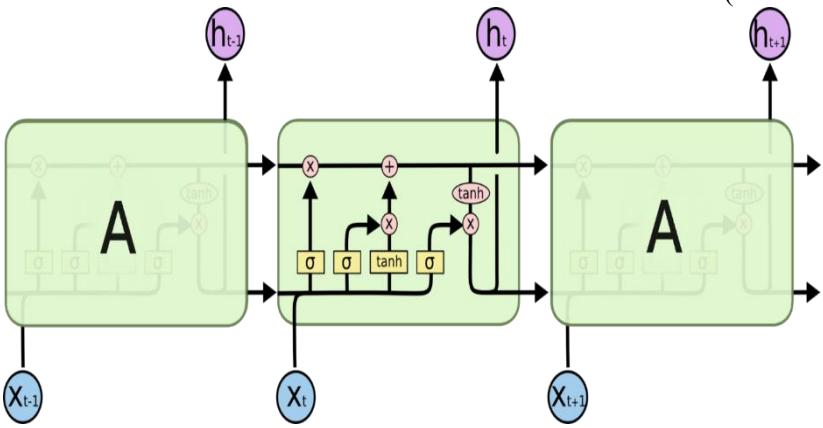


Рис. 6: слой LSTM

горизонтальная линия, проходящая по верхней части схемы (рис. 8). Состояние ячейки проходит напрямую через всю цепочку, участвуя лишь в нескольких линейных преобразованиях. Информация может легко течь по ней, не подвергаясь изменениям.

Тем не менее, LSTM может удалять информацию из состояния ячейки; этот процесс регулируется структурами, называемыми фильтрами (gates). Фильтры позволяют пропускать информацию на основании некоторых условий. Они состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения.

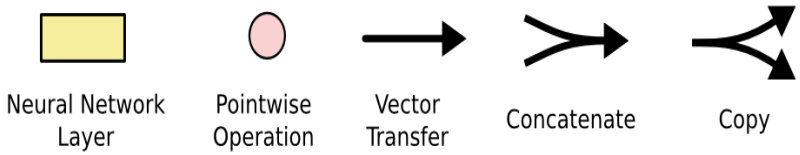


Рис. 7: условные обозначения

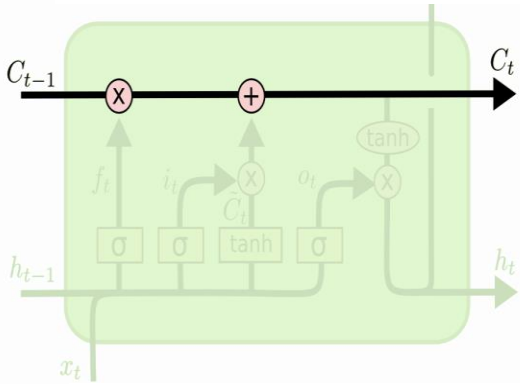


Рис. 8: состояние ячейки

Сигмоидальный слой возвращает числа от нуля до единицы, которые обозначают, какую долю каждого блока информации следует пропустить дальше по сети. Ноль в данном случае означает “не пропускать ничего”, единица – “пропустить все”.

В LSTM три таких фильтра, позволяющих защищать и контролировать состояние ячейки.

Секция IV. Системы безопасности

Портативный сканер беспроводных сетей

Бушин Олег Алексеевич

г. Москва, ГБОУ Школа №1501, 11 класс

Научный руководитель: Медведев Николай Викторович,

МГТУ им. Баумана, кандидат технических наук,

доцент кафедры «Информационная безопасность»

Цель работы: разработка и программная реализация портативного сканера для сбора информации из открытых беспроводных сетей.

В рамках исследования моделируется процесс сканирования беспроводных сетей. На основе разработанной модели реализуется устройство (сканер), запрограммированное на языке высокого уровня.

Сканер позволяет узнать тип шифрования, MAC-адрес раздающего устройства, RSSI, количество сетей.

Данные, полученные с устройства, могут быть использованы для технической разведки и для тестирования на проникновение.

Преимуществом данного устройства являются малые размеры и малое энергопотребление, что обеспечивает компактность и малый вес при длительном времени работы.

Теоретическая часть работы включает:

а) Изучение методов шифрования в беспроводных сетях.
б) Изучение языка программирования C++.
в) Разработка алгоритмов реализации.

Практическая часть включает:

а) Разработка модели сканирования беспроводных сетей.

б) Программная и аппаратная реализация устройства на основе разработанной модели.

Оглавление

1. Введение
2. Техническая характеристика модуля ESP8266
3. Распиновка модуля ESP8266 и micro card reader
4. RSSI
5. Протоколы безопасности беспроводных сетей
6. Моделирование корпуса
7. Изображения корпуса с габаритами размеров в программе “SolidWorks”288
8. Описание программы
   8.1 Блок-схема программы
   8.2 Заголовок и объявление библиотек
   8.3 Ввод данных для подключения
   8.5 Подключаемся к сети и к Blynk
   8.6 Инициализация SD-card
   8.7 Создание файла для записи результатов
   8.8 Объявляем и присваиваем значения переменным
   8.9 Вывод номера листа, времени и дату
   8.10 Проверяем наличие сети
   8.11 Сеть найдена
   8.12 Определение типов шифрования

8.13 Запись результатов сканирования
8.14 Вывод результатов
8.15 Разделяем сканирования
8.16 Ошибка при открытии файла
8.17 Частота повторного сканирования
8.18 Вывод результатов в виртуальный порт
9. Вывод результатов в COM- порт и запись на micro SD-карту
10. Вывод в терминал “Blynk”
11. Заключение
12. Литература

1. Введение

Цель работы: разработка и программная реализация портативного сканера для сбора информации из открытых беспроводных сетей. Сканер позволяет узнать:

- Тип шифрования сети.
- Количество сетей в радиусе действия сканера.
- Не скрытый SSID (имя, которое идентифицирует конкретную беспроводную локальную сеть).
- MAC-адрес раздающего устройства (уникальный идентификатор, присваиваемый каждой единице активного оборудования).
- RSSI. (Показатель уровня принимаемого сигнала).

Все эти данные необходимы при технической разведке и для тестирования на проникновение.

Устройство имеет рад преимуществ в сравнении с телефоном или с ноутбуком, на которых также могут запущенны программы для мониторинга сетей: малые размеры и компактность, небольшое энергопотребление**,** дешевизна, скрытность.

В процессе работы сканер не подключается к точке доступе, поэтому его невозможно обнаружить. Для сканирования сетей подключение не нужно. Чтобы объявить о наличии сети, все Wi-Fi точки непрерывно транслируют SSID. Если SSID скрыт, сканирование не произойдет. Данное решение необходимо для бескомпроматности функционирования разведывательного устройства.

Данный проект реализован на модуле ESP8266 на базе чипа **CP2102**. Результаты записываются на micro SD card и выводятся - при наличии подключения в интернет - в приложение Blynk.

Программирование осуществилось через программу “Arduino IDE” на языке C++.

Также был спроектирован корпус в программе “SolidWorks” и распечатан на 3D-принтере.

2. Техническая характеристика модуля ESP8266

ESP8266 - микроконтроллер китайского производителя Espressif. Отличительной особенностью которого является наличие радиоинтерфейса Wi-Fi. Ядро ESP8266 интегрировано в Tensilica L106 – 32-битный микроконтроллер с ультранизким

энергопотреблением. Поддержка тактовых частот 80 и 160 МГц; встроенная микрополосковая антенна на плате.

Пользователь может использовать модуль либо в качестве дополнения для подключения какого-либо устройства к сети, либо в качестве отдельного сетевого контроллера.

Характеристики:

- встроенный 32-битный MCU с низким энергопотреблением;
- встроенный 10-битный АЦП; Аналого-цифровой преобразователь - устройство, преобразующее входной аналоговый сигнал в дискретный код (цифровой сигнал).
- Wi-Fi 2,4 ГГц, поддержка WPA/WPA2;
- потребление в режиме глубокого сна < 10 мкА, ток утечки в выключенном режиме < 5 мкА;
- потребление до 215 мА в режиме передачи, 100 мА в режиме приема, 70 мА в режиме ожидания.
- просыпание и передача пакетов через < 2 мс;
- выходная мощность +20 дБм в режиме 802.11b;
- диапазон рабочих температур: -40 – 125 ºС.
- питание от 2,3 до 3,6 В.

1. Распиновка модуля ESP8266 и micro card reader

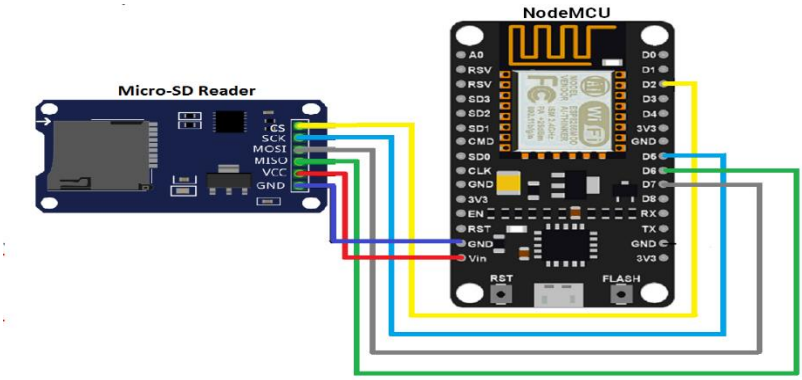


Рисунок 1. Схема подключения ESP8266 и card reader

Распиновка осуществляется следующим образом:

CS – D2

SCK – D5

MOSI – D7

MISO – D6

VCC – Vin

GND – GND

Для подключения платы и card reader используются следующие пины:

– GND: Земля (электроника) — узел цепи, потенциал которого условно принимается за ноль.
– VIN 5V - питание пять вольт.

– GPIO: D2, D5, D6, D7 - интерфейс для связи между компонентами компьютерной системы. Во встраиваемых системах широко используют GPIO для чтения информации от различных внешних датчиков, а также для управления двигателями постоянного тока, аудио, ЖК-дисплеями или светодиодами для индикации состояния чего-либо.

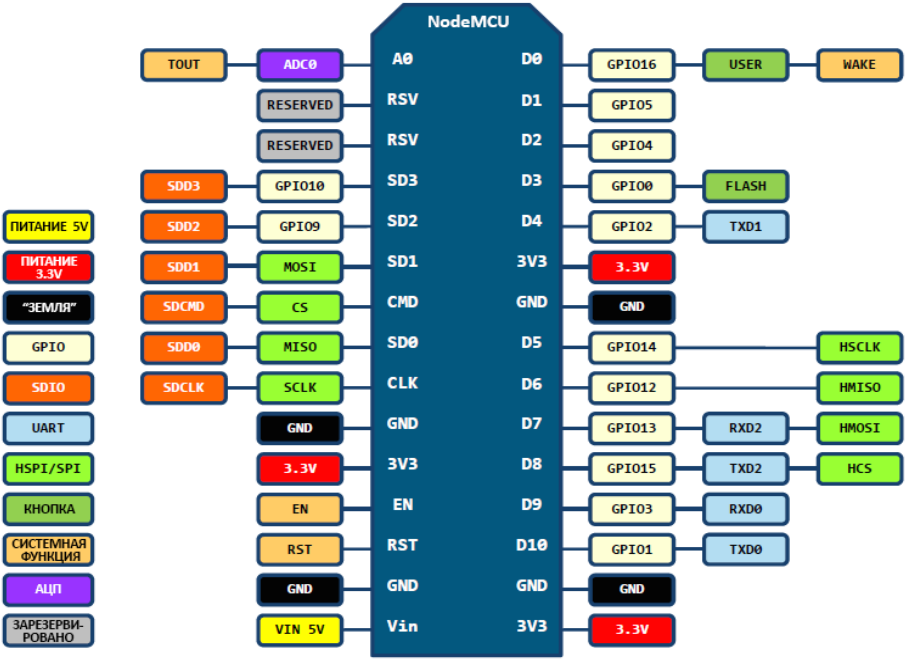


Рисунок 2. Распиновка ESP8266

2. RSSI

Показатель уровня принимаемого сигнала, RSSI (англ. received signal strength indicator) — полная мощность принимаемого приёмником сигнала. Измеряется приёмником по логарифмической шкале в дБм. (dBm, децибел относительно 1 милливатта).

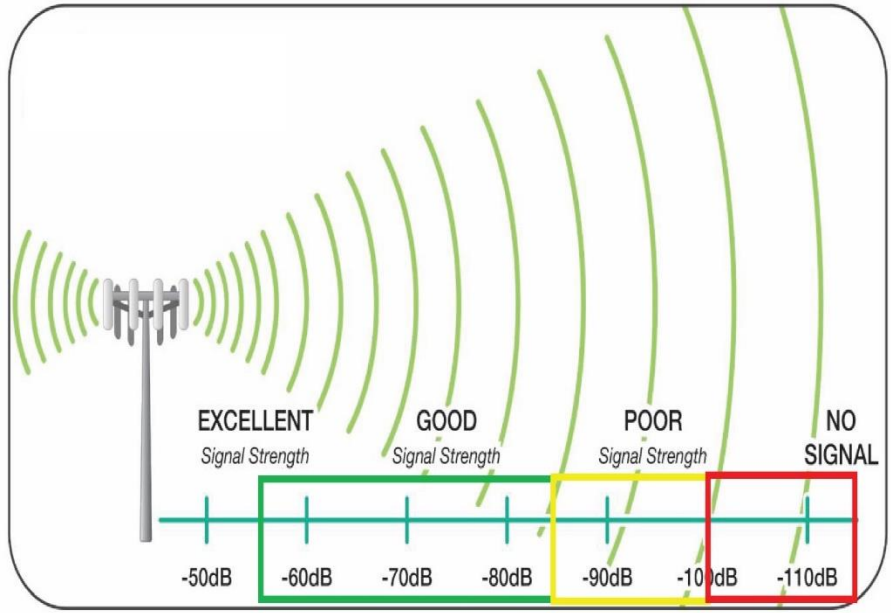


Рисунок 3. Корреляция RSSI и качества сигнала

3. Протоколы безопасности беспроводных сетей

Сканер может распознать следующие протоколы безопасности:

- WEP.
- WPA
- WPA2

WEP

В основе WEP используется симметричный потоковый шифр RC4, которые имеет высокую скорость работы. Есть два основные протокола WEP:

40-битный WEP (длина ключа 64 бита, 24 из них выделено на вектор инициализации, которые передается в открытом тексте).

104-битный WEP (длина ключа 128 бит, 24 из них выделено на вектор инициализации. Вектор инициализации зашифрован алгоритмом RC4.

Вектор инициализации — Initialization vector (IV), представляет собой произвольное число, которое может быть использовано вместе с секретным ключом для шифрования данных. Этот номер, также называемый одноразовым номером, используется только один раз в любом сеансе.
Использование IV предотвращает повторение шифрования данных, что делает процесс взлома более трудным для хакера с помощью атаки по словарю, в попытках найти шаблоны и сломать шифр.

Алгоритм RC4, как и любой потоковый шифр, строится на основе генератора псевдослучайных битов. На вход генератора записывается ключ, а на выходе читаются псевдослучайные биты. Длина ключа может составлять от 40 до 2048 бит.

WPA

Представляет собой обновлённую программу сертификации устройств беспроводной связи. Технология WPA пришла на замену технологии

защиты беспроводной Wi-Fi сети WEP. Плюсами WPA являются усиленная безопасность данных и ужесточённый контроль доступа к беспроводным сетям. Wi-Fi Alliance даёт следующую формулу для определения сути WPA:

WPA = 802.1X + EAP + TKIP + MIC.

IEEE 802.1X - это стандарт, который используется для аутентификации (процесс подтверждения, что этот человек именно тот, за кого себя выдает) и авторизации (процесс принятия решения о том, что именно этому аутентифицированному человеку разрешается делать) пользователей в сети передачи данных. Благодаря стандарту 802.1x можно предоставить пользователям права доступа к корпоративной сети и ее сервисам в зависимости от группы или занимаемой должности, которой принадлежит тот или иной пользователь.

EAP – расширяемый протокол аутентификации, который представляет собой набор из большого количества (порядка 40) методов аутентификации.

TKIP – механизм динамической генерации ключей шифрования, который позволяет сделать процесс обмена информацией более безопасным и исключает возможность перехвата данных.

MIC – система проверки целостности пакетов, позволяющая еще надежнее защитить данные от их возможного перехвата. MIC работает как на отправителе, так и на получателе, что позволяет максимально защитить передаваемые данные.

WPA2

Алгоритм WPA2 является дальнейшей модификацией алгоритма WPA. Все сертифицированные устройства, начиная с 2006 года выпуска, обязательно должны поддерживать этот алгоритм. WPA2 – наиболее защищенный алгоритм шифрования данных.

Как и WPA, WPA2 используется шифрование с помощью алгоритма AES со 128-битным ключом. Изменения коснулись только «напарника» AES – механизма управления ключами TKIP. Ему на смену пришел метод CCMP.

CCMP-шифрование —протокол шифрования, созданный для замены TKIP, обязательного протокола шифрования в WPA и WEP, как более надёжный вариант. CCMP является обязательной частью стандарта WPA2 и необязательной частью стандарта WPA. Алгоритм CCMP основан на алгоритме шифрования CCM AES.

4. Моделирование корпуса

Корпус был спроектирован в программе “SolidWorks”. Программа даёт возможность 3D проектирования изделий, деталей и сборок любой степени сложности с учётом специфики изготовления.

Корпус состоит из основной части и крышки прикрепленной на болтах.

Корпус вмешает в себя плату Esp8266 и card reader.

Основное требование при создании корпуса были:

- Надёжность конструкции.
- Малые размеры и вес.
- Дешевизна изготовления.
- Возможность извлечения SD- карты, не снимая крышки.
- Возможность подключение питания, не снимая крышки.

С учетом всех этих требований был спроектирован корпус и распечатан на 3D-принтера из ABS-пластика.

ABS-пластик имеет ряд преимуществ: коррозийная стойкость, ударопрочность, легкость материала, возможность доработки, доступность и относительно маленькая стоимость изготовления.

Технические характеристики корпуса

Размер: 32мм*22мм*52.5мм.

Материал: ABS-пластик.

Вес: 13.1 грамм

Объём: 12.5см^3

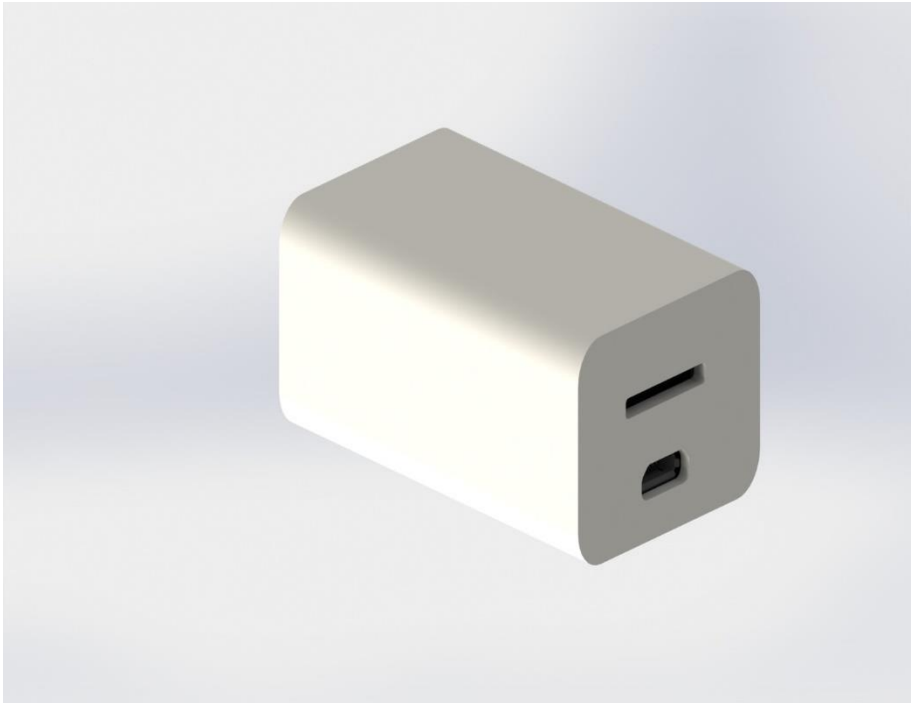


Рисунок 4. Общий вид.

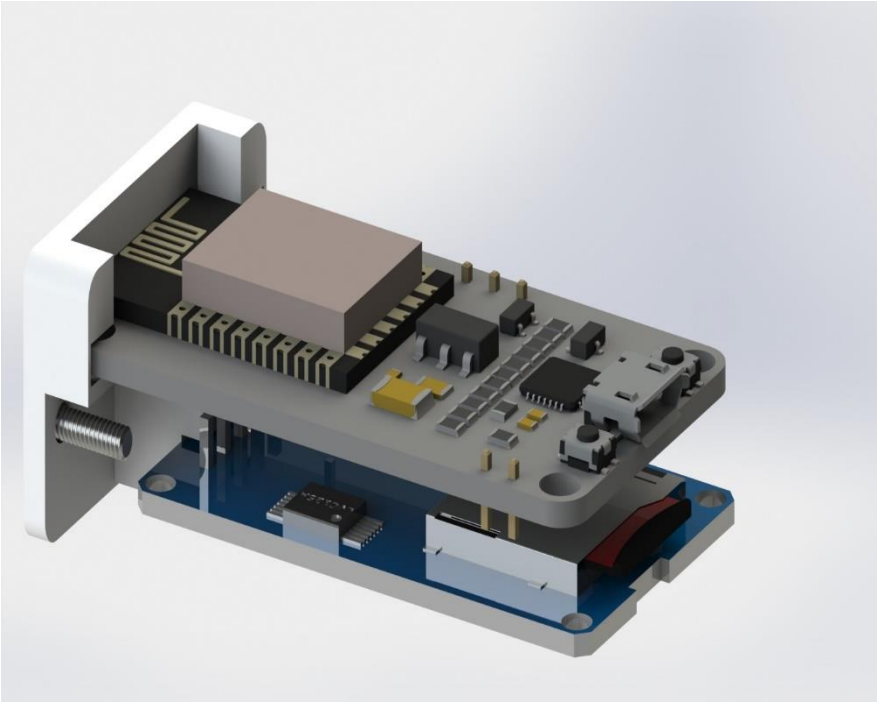
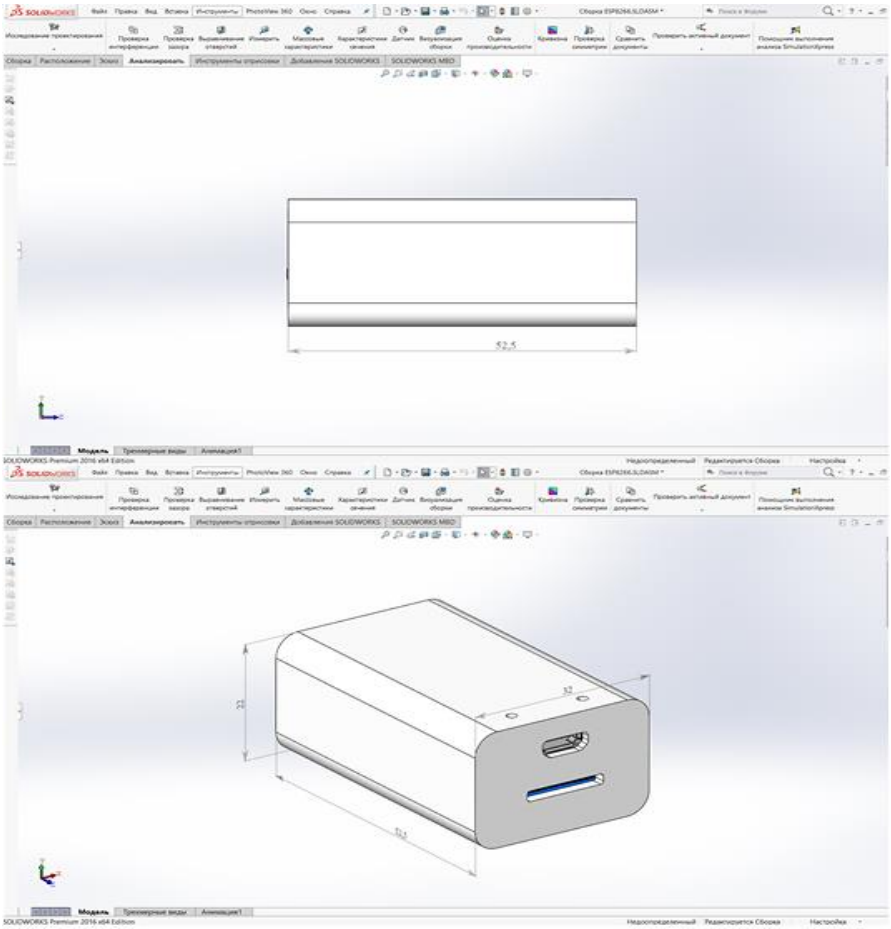
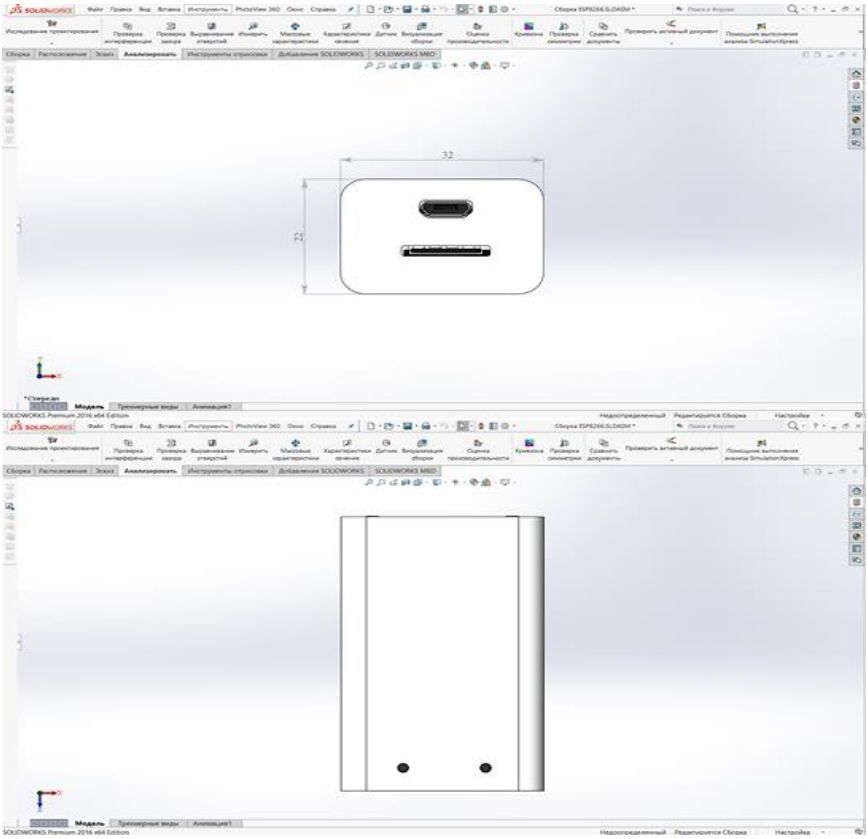


Рисунок 5. Задняя крышка с платой и micro card reader.

Изображения корпуса с габаритами размеров в программе “SolidWorks”



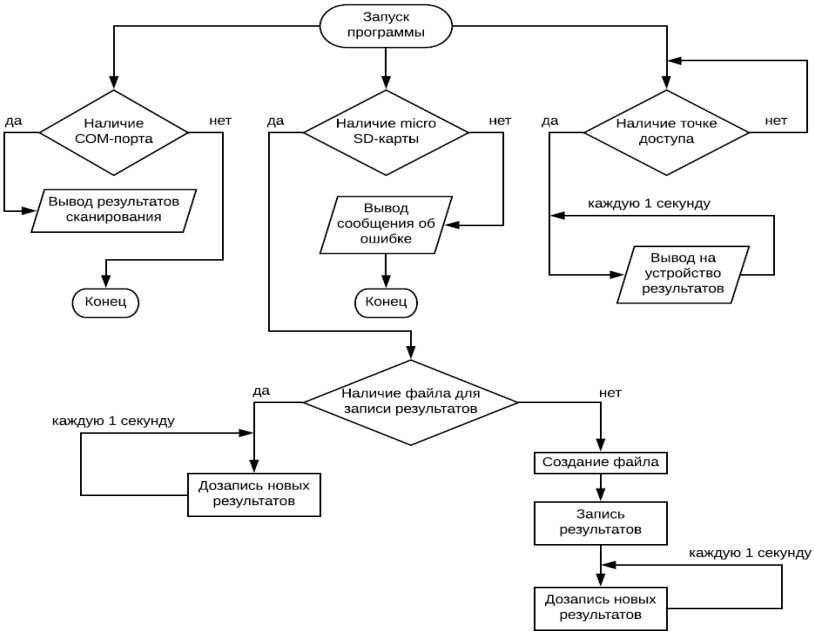
Изображение 1



Изображение 2

1. Описание программы

8.1 Блок-схема программы



8.2 Заголовок и объявление библиотек

```
/*
-----------------------------------|
Created by Oleg Bushin 07.01.2019  |
-----------------------------------|
Pinout Card reader and esp8266     |
                                   |
CS--D2                             |
SCK--D6                            |
MOSI--D7                           |
VCC--Vin                           |
GND--GND                           |
-----------------------------------|
*/

#include <SPI.h>
#include <SD.h>

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <TimeLib.h>
#include <WidgetRTC.h>
```

В 3-ей строчке указан автор и дата создания.

С 4-ой по 11-ую продублирована распиновка платы и модуля работы с micro SD-картой.

В 15-ей строчке находится директива препроцессора "**include**", что означает "включить", то есть - "включить в состав программы заголовочный файл "**SPI.h**" (h - header, заголовок). Данная библиотека позволяет вам взаимодействовать с SPI устройствами.

SPI - последовательный синхронный стандарт передачи данных в режиме полного дуплекса (Дуплекс — способ связи с использованием приёмопередающих устройств), предназначенный для обеспечения простого и недорогого высокоскоростного сопряжения микроконтроллеров и периферии.

В 16-ой строчке указана библиотека для работы с SD-card и micro SD-card.

С 18-ую по 22-ую указаны библиотеки программы “Blynk”, а именно библиотека работы с платой ESP8266 и библиотека часов реального времени.

8.3 Ввод данных для подключения

```
23 char auth[] = "token";
24 char ssid[] = "network name";
25 char pass[] = "password";
```

В 23-ей строчке вводим токен устройства для связи с программой Blynk.

В 24-ой строчке вводим имя сити, через которую будем подключаться.

В 25-ой строчке вводим пароль от сети.

8.4 Подключение виджетов

```
27 WidgetRTC rtc;
28 WidgetTerminal terminal(V2);
```

В 27-ой строчке подключаем виджет часов реального времени.

В 28-ой строчке выводим значение в виртуальный терминал с pin V2.

```
30 File myFile;
31
32 unsigned long int Network_list_number = 0;
```

В 30-ой строчке создаем объект класса File.

В 32-ой строчке присваиваем 0.

8.5 Подключаемся к сети и к Blynk

```
34 void setup()
35 {
36   Serial.begin(9600);
37   Blynk.begin(auth, ssid, pass);
```

В 36-ой строчке плата ESP8266 посылает по последовательному порту (COM-порту) данные со скоростью 9600 бит в секунду.

В 37-ой строчке подключаемся к сети и Blynk.

```
39   rtc.begin();
```

В 39-ой строчке начинаем синхронизировать время.

8.6 Инициализация SD-card

```
40
41    while (!Serial); {
42    }
43
44    Serial.print("Initializing SD card...");
45
46    if (!SD.begin(4)) {
47      Serial.println("Initialization failed!");
48      return;
49    }
50    Serial.println("Initialization done.");
51    WiFi.mode(WIFI_STA);
52
53    terminal.println(F("-----Scan Start-----"));
54    terminal.flush();
55
56  }
```

В 44-ой строчке выводим "Initializing SD card..." в COM-порт.

В 47-ой строчке выводим сообщение об ошибке, если не получилось.

В 50-ой строчке при удачной подготовке к работе выводим “Initialization done.”

И в 53-ей строчке печатаем о начале сканирования в терминал Blynk.

8.7 Создание файла для записи результатов

```
58 void loop()
59 {
60
61   myFile = SD.open("scan.txt", FILE_WRITE);
62   if (myFile) {
63
64   Blynk.run();
```

В 61-ой строчке создаем файл skan.txt и записываем на него результаты, если он был ранее создан дозаписываем результаты сканирования.

8.8 Объявляем и присваиваем значения переменным

```
66   int n = WiFi.scanNetworks();
67   char ssid[64];
```

В 66-ей строчке присваиваем n количество обнаруженных сетей.

В 67-ой определяем ssid как массив из 64 объектов типа данных char.

8.9 Вывод номера листа, времени и дату

```
69    ++Network_list_number;
```

В 69-ой строчке после каждого сканирования увеличиваем номер на один.

```
70    String NewListDataTime = "Network list number "
+ String(Network_list_number) + " ";
```

```
71    NewListDataTime += String(hour()) + ":" + minute() + ":"
+ second() + " " + String(day()) + " " + month() + " " + year() + "\n\n";
```

В 70-ой и в 71-ой строчке последовательно записываем в переменную NewListDataTime номер сканирования, время сканирования дату сканирования.

8.10 Проверяем наличие сети

```
73   if (n == 0) {
74
75     Serial.print(NewListDataTime);
76     Serial.print("No networks found\n");
77
78     myFile.println(NewListDataTime);
79     myFile.println();
80     myFile.println("No networks found\n");
81
82     sendUptime(NewListDataTime);
83     sendUptime("No networks found\n");
84   }
```

Данный блок выводит в случае отсутствия сети сообщение “No networks found”.

8.11 Сеть найдена

```
85    else
86    {
87      Serial.print(NewListDataTime);
88
89      myFile.println(NewListDataTime);
90      myFile.println();
91      sendUptime(NewListDataTime);
92
93      for (int i = 0; i < n; ++i)
94      {
```

Если сеть найдена, то выводим переменную NewListDataTime в сом-порт и Blynk. Записываем в файл.

Повторяем данный цикл ровно столько раз сколько сетей было найдено.

8.12 Определение типов шифрования

```
96        String EncryptionType = "";
97
98        if (WiFi.encryptionType(i)==2)
99          EncryptionType = "TKIP(WPA)";
100
101       else if (WiFi.encryptionType(i)==5)
102         EncryptionType = "WEP";
103
104       else if (WiFi.encryptionType(i)==4)
105         EncryptionType = "CCMP(WPA)";
106
107       else if (WiFi.encryptionType(i)==7)
108         EncryptionType = "NONE";
109
110       else if (WiFi.encryptionType(i)==8)
111         EncryptionType = "AUTO";
```

Данный блок программы определяет тип шифровании сети:

Если переменная “i” равна 2 тип шифрования сети "TKIP(WPA)"

Если переменная “i” равна 4 тип шифрования сети "CCMP(WPA)"

Если переменная “i” равна 5 тип шифрования сети "WEP"

Если переменная “i” равна 7 тип шифрования сети "NONE"

Если переменная “i” равна 8 тип шифрования сети "AUTO" (WPA/WPA2 PSK)

8.13 Запись результатов сканирования

```
113        String data = String(i+1) + " " + WiFi.SSID(i) +
" " + EncryptionType + " " + "(" + String(WiFi.RSSI(i))+")";
114       data += String((WiFi.encryptionType(i) == ENC_TYPE_NONE)?
" ":"* ")+ "MAC:"+ String(WiFi.BSSIDstr(i))+ "\n";
```

В 113-ей и 114-ой строчке присваиваем переменной Data номер строки, название точки доступа, тип шифрования, RSSI, MAC-адрес.

8.14 Вывод результатов

```
116         Serial.print(data);
117
118         myFile.println(data);
119
120         sendUptime(data);
121     }
```

Выводим значения в COM-порт.

Записываем на SD-карту.

Выводим в Blynk.

8.15 Разделяем сканирования

```
122     Serial.print("_____\n\n");
123
124     myFile.println("_____");
125     myFile.println();
126
127     sendUptime("_____\n\n");
128
129     }
130     }
131
```

После каждого сканирования выводим полосу для наглядности выводов результатов.

8.16 Ошибка при открытии файла

```
132   else {
133
134     Serial.println("error opening scan.txt");
135   }
136     myFile.close();
```

Если не поучается открыть файл, то выводим сообщение об ошибке. Закрываем файл.

8.17 Частота повторного сканирования

```
138 delay(10000);
139 }
```

Через процедуру delay выставляем время ожидания до следующего запуска. Время выставляется в миллисекундах.

8.18 Вывод результатов в виртуальный порт

```
141 void sendUptime(String data)
142 {
143   Blynk.virtualWrite(V2, data);
144 }
```

Выводим результаты в виртуальный терминал.

5. Вывод результатов в COM- порт **и запись на micro SD-карту**

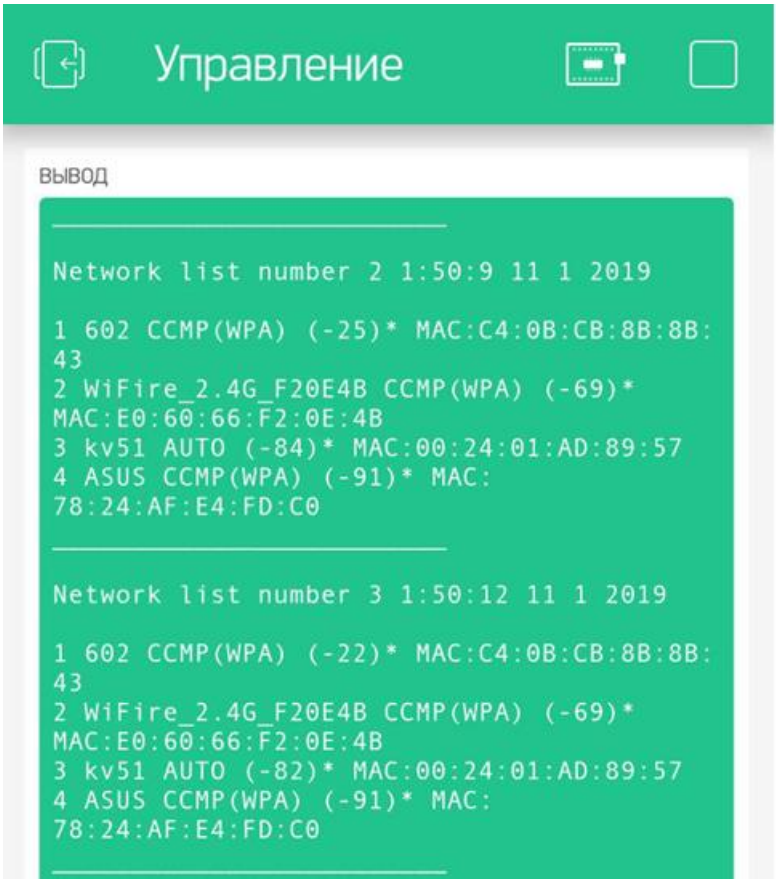
В начале каждого нового сканирования пишется дата и время санирования. Результаты выводятся в следующем порядке: номер строки, название точки доступа, тип шифрования, RSSI, MAC-адрес.

В таком виде результаты записываются на micro SD-карту и выводятся в COM-порт, если он есть в наличии.

```
Network list number 2 0:8:2 10 1 2019

1 602 CCMP(WPA) (-21)* MAC:C4:0B:CB:8B:8B:43
2 KiraMcpro_2G AUTO (-94)* MAC:78:94:B4:66:BF:84
3 ASUS CCMP(WPA) (-89)* MAC:78:24:AF:E4:FD:C0
4 kv51 AUTO (-78)* MAC:00:24:01:AD:89:57
_____________________________

Network list number 3 0:8:5 10 1 2019

1 602 CCMP(WPA) (-20)* MAC:C4:0B:CB:8B:8B:43
2 ASUS CCMP(WPA) (-89)* MAC:78:24:AF:E4:FD:C0
3 kv51 AUTO (-79)* MAC:00:24:01:AD:89:57
_____________________________
```

6. Вывод в терминал “Blynk”

При наличии выхода в интернет результаты выводятся в терминал “Blynk”.


Управление
ВЫВОД
Network list number 2 1:50:9 11 1 2019
1 602 CCMP(WPA) (-25)* MAC:C4:0B:CB:8B:8B:
43
2 WiFire_2.4G_F20E4B CCMP(WPA) (-69)*
MAC:E0:60:66:F2:0E:4B
3 kv51 AUTO (-84)* MAC:00:24:01:AD:89:57
4 ASUS CCMP(WPA) (-91)* MAC:
78:24:AF:E4:FD:C0
Network list number 3 1:50:12 11 1 2019
1 602 CCMP(WPA) (-22)* MAC:C4:0B:CB:8B:8B:
43
2 WiFire_2.4G_F20E4B CCMP(WPA) (-69)*
MAC:E0:60:66:F2:0E:4B
3 kv51 AUTO (-82)* MAC:00:24:01:AD:89:57
4 ASUS CCMP(WPA) (-91)* MAC:
78:24:AF:E4:FD:C0

7. Заключение

В ходе работы осуществлена полнофункциональная разработка портативного сканера беспроводных сетей.

8. Литература

1 Ю.К. Меньшаков “ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ТЕХНИЧЕСКИХ РАЗВЕДОК”, под редакцией Ю.Н. Лаврухина. 2008 год.
2 Программирование на языке С++: Практический курс. Огнева М. В., Кудрина Е. В. 2017 год.
3 https://community.blynk.cc - образовательная платформа.
4 https://esp8266.ru - интернет ресурс.
5 https://www.arduino.cc - интернет ресурс.
6 https://www.blynk.cc - интернет ресурс.
7 https://www.wikipedia.org - интернет ресурс.
8 https://circuits4you.com - интернет ресурс.

Секция V. Радио-опто-электроника

Использование датчиков мышечной активности для опознания напряжения мышц

Мифтахов Руслан Борисович

г. Москва, ГБОУ Школа №1581, 11 класс

Научный руководитель: Круглов Геннадий Валерьевич,

Кафедра РЛ-1, МГТУ им. Н.Э. Баумана

В работе проведено исследование методики считывания сигналов мышечной активности предплечья.

Целью работы является проектирование устройства, которое может осуществлять управление механическим объектом посредством мышечных сигналов. Основной подход к считыванию - измерение разности потенциалов в различных точках мышц. Наиболее распространены измерения сигналов с помощью миографических датчиков.

Рассмотрены поверхностные, неинвазивные электроды, проводился анализ существующих реализаций и обоснование необходимости в разработке собственной конструкции электродов и системы считывания.

Существующие решения нацелены, преимущественно, на получение единичных сигналов, и не предназначены для комплексного анализа пальцевых жестов. Спроектированы электрические схемы отдельных элементов и модульная схема устройства, разведены платы, изготовлен прототип.

Прототип представляет собой манжету с массивом электродов, датчиками, модулями питания, аналого-цифровыми преобразователями и центрального процессорного устройства на базе ESP-32.

Разработаны алгоритмы сбора сигналов и распознавания пальцевых жестов, программа для процессора.

Применимость данного решения продемонстрирована в виде стенда по управлению роботизированным манипулятором - пятипальцевой механической руки, реализованной по клиент серверной архитектуре с использованием произвольного протокола на базе http. Данная система может быть применена для управления исполнительными силовыми элементами в агрессивных средах, при протезировании, в виртуальной и дополненной реальности.

2 Содержание

Аннотация

Содержание

Введение

Биологическое обоснование

Миоэлектрический сенсор

Устройство электрода

Устройство датчика

Сборка и изготовление датчика

Структурная схема устройства

Программа контроллера миографических датчиков

Демонстрационный стенд

Взаимодействие стенда и основного устройства.

Заключение

Список используемых источников

3 Введение

В связи с развитием радиоэлектроники и робототехники, усложнением механизмов, которые требуют управления, в данное время проводится большое количество исследований, связанных с поиском дополнительных интерфейсов управления.

Одним из таких интерфейсов является прямое управление механизмами посредством прямого считывания миографических сигналов.

До недавнего времени большинство решений в этой области было связано с простейшим распознаванием состояния мышц. Как правило такие интерфейсы выдавали только два состояние - “ДА” и “НЕТ”.

Только недавно стали предприниматься попытки расширить функционал такого интерфейса. В большинстве случаев миографический интерфейс оснащался дополнительными источниками данных, например, аудиоканалом, различными датчиками перемещения в пространстве (гироскопы, акселерометры, магнитометры). Безусловно, данные решения заслуживают внимания, но в данной работе была предпринята попытка расширения функционала, основанного именно на чтении мышечных электрических управляющих воздействий.

В работе доказано, что на современной элементной базе возможно решение этой задачи с хорошим качеством.

Описана теория измерения сигналов мышечного тонуса, разработана схема и конструкция датчика. Выполнен рабочий экземпляр устройства, считывающего пальцевые жесты. Воплощен стенд демонстрирующий работоспособность.

В разделе “Биологическое обоснование” приведена краткая теория считывания биоэлектрических сигналов мышц пальцев.

Раздел “Миоэлектрический сенсор” содержит комплекс исследований, связанный с исследованием аналогов и прототипов электромиографических (ЭМГ) датчиков, выявлением их достоинств и недостатков.

Подраздел “Устройство электрода” описывает комплекс электродов, применяемый в данной работе.

Подраздел “Устройство датчика” содержит описание и расчет электрической схемы датчика.

“Сборка и изготовление датчика” - показывает этапы работы над датчиком.

“Структурная схема устройства” - разъясняет электрическую конструкцию устройства в целом.

“Программа контроллера миографических датчиков” - алгоритм и принцип работы микропроцессорного ядра устройства.

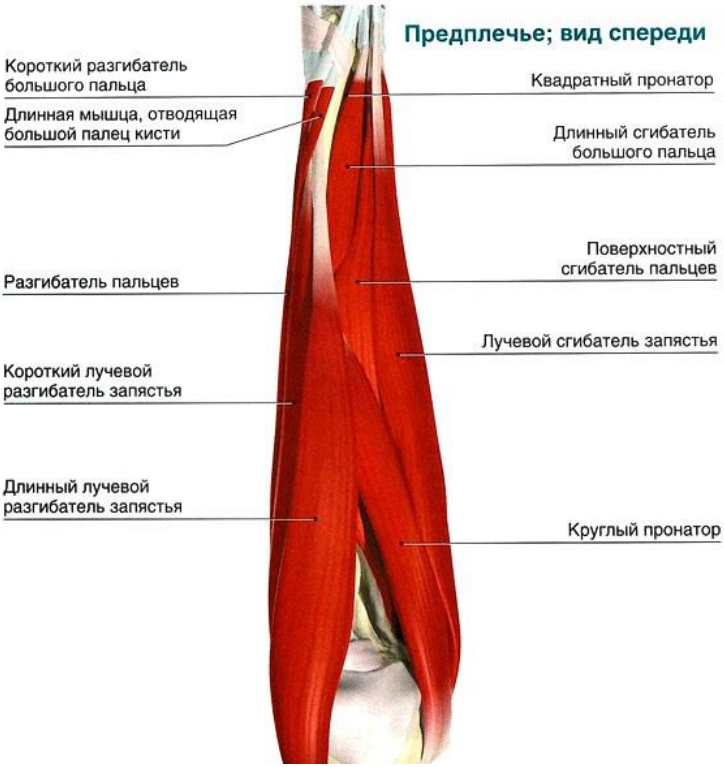
“Демонстрационный стенд” описывает структурную схему устройства визуализации.

“Взаимодействие стенда и основного устройства” описывает протокол связи.

4 Биологическое обоснование

Мышечное сокращение – это реакция мышечных тканей на нейромедиаторы. Нейромедиатор - биологически активные химические вещества, с помощью которых производится передача электрического импульса от нервной клетки. Такой импульс можно определить при замере разности потенциалов возникающей в мышце. Замеряя и обрабатывая разность потенциалов мышцы, можно определить в какой момент палец сгибается, напряжен или разгибается.

Мышцы предплечья состоят из двух групп мышц — передних и задних. Передние мышцы — сгибатели, а задние — разгибатели и супинаторы. Одни из этих мышц отвечают за движения запястья, а другие — за движения пальцев. Сокращающаяся мышца передает костям усилие через сухожилия. При сокращении мышц, если усилия достаточно, то пальцы перемещаются (1).



“Рис. 1”

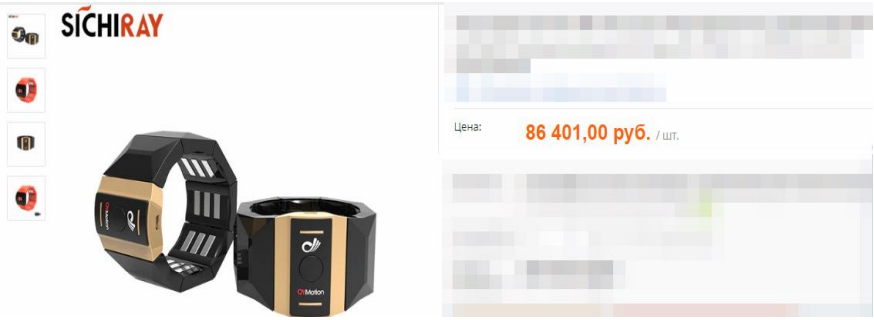
В движении пальцев участвуют, по крайней мере, 2 мышцы, одна из которых отвечает за сгибание, а другая- за разгибание соответственно (Расположение мышц - “Рис. 1”). Мышечный комплекс предплечья не имеет отдельных групп мышц для управления каждым пальцем. Движение пальцев осуществляется вариацией изгибов одной и той-же группы мышц. В работе выдвигается гипотеза, в соответствии с которой, в независимости от усилия, для конкретного типа движения пальца, общая картина сигналов снятых с группы мышц будет неизменна. Основываясь на этом, снятие множества сигналов с

различных участков группы мышц предплечья позволит однозначно определить сформированный жест, причём чем больше количество точек измерения сигналов, тем точнее можно определить жест и тем больше будет общее количество распознаваемых жестов.

5 Миоэлектрический сенсор

В данной части произведено рассмотрение имеющихся реализаций датчиков и возможности их использования в проекте. При подборе датчиков пришлось учитывать: доступность, стоимость, надежность, размеры и сложность интеграции.

1)Датчик компании SICHIRAY OY motion “GForce”, стоимостью 86 401 рублей.



“Gforce”

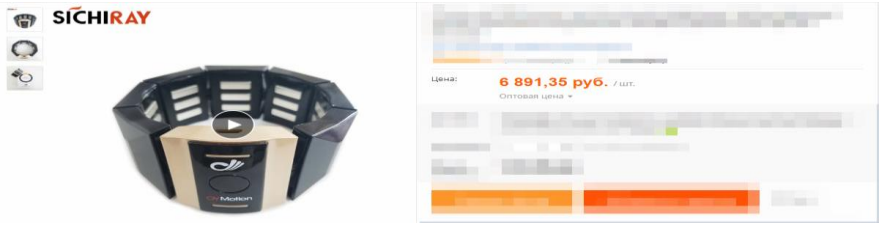
Датчик имеет ряд недостатков, такие как: высокая стоимость, количество распознаваемых жестов (8 жестов). Датчик не позволяет определить какие-либо жесты, кроме предусмотренных изначально.

2)Датчик “Thalmic Labs Myo Gesture Control”, стоимостью 20 990 рублей.

“Thalmic Labs Myo Gesture Control”

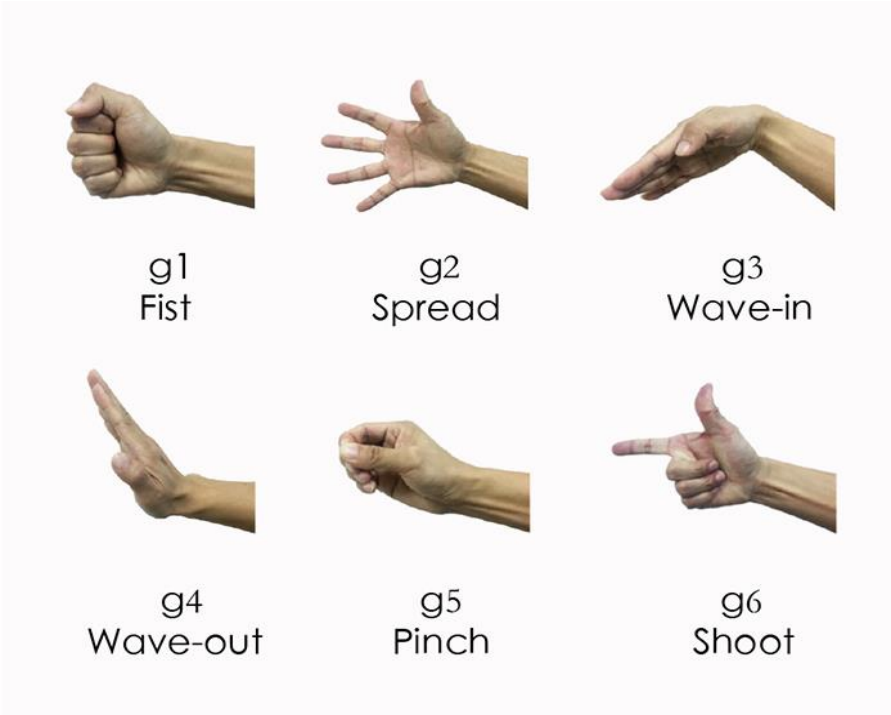
“Браслет Myo позволяет использовать электрическую активность в мышцах для беспроводного управления компьютером, телефоном и другими цифровыми приборами.” В датчик встроен гироскоп, но он также не позволяет определять большое количество жестов.

3) Датчик компании SICHIRAY OY motion “GForce”, стоимостью 6 891. 35 рублей.



“GForce используется для определения жестов в основном через сигналы EMG.

Gforce позволяет определять 6 жестов, имеет модуль Bluetooth, что позволяет управлять смарт-системами.” Браслет имеет те же недостатки, которые были описаны ранее (2).



“Жесты, поддерживаемые браслетом”

Приведенные миографические сенсоры имеют малое количество считываний разности потенциалов с мышц, также скорость считывания низкая. Чтобы восполнить недостаток количества измерений, используется гироскоп. Используя подобную схему измерений, суть снимаемых показаний меняется, исключая возможность увеличения количества распознаваемых жестов при использовании пальцев руки. Гироскоп возможно использовать в датчиках, но как дополнительный источник данных.

4) ЭМГ-датчик компании “BitronicsLab”, стоимостью 25 000 рублей (3).

Данный ЭМГ-сенсор, имеет всего один датчик - два электрода. Скорость измерений датчика низка. Датчик необоснованно дорогой.

На основе произведенного анализа параметров доступных датчиков, было принято решение, спроектировать и изготовить собственный сенсор.

5.1 Устройство электрода

Электрод представляет собой контактную площадку, с минимальным электрическим сопротивлением, которая должна максимально плотно прилегать к участку кожи, с которой будет снята разность потенциалов. Электрод состоит из двух шайб, одна из которых имеет 3-х миллиметровый внутренний диаметр и 9-ти миллиметровый внешний диаметр, а другая - 5-ти миллиметровый внутренний и 15-ти миллиметровый внешний, которая и является основной проводящей

частью прилегающей площадки. Разность потенциалов снимается между двумя электродами, расстояние между которыми 83мм.

Сами электроды закреплены на пластмассовом, диэлектрическом основании. Всего имеется 10 комплектов электродов, по три на датчик. Два электрода - измерительные, третий - заземляющий. Как показали эксперименты, такая конструкция не требует контактных гелей, которые обычно используются.

5.2 Устройство датчика

На вход датчик получает 3 сигнала с электродов, между двумя из которых измеряем разность потенциалов, усиливаем и фильтруем ее, а третий сигнал является “заземлением”.

Первый каскад схемы построен на инструментальном (измерительном) усилителе U1, марки AD8220. В соответствии с руководством по применению, изменяя сопротивление R1, можно изменить коэффициент усиления G по формуле (4) :

$$R1=49.4k\Omega /(G-1),$$

где R1 - сопротивление резистора R1;

G - коэффициент усиления.

При проведении экспериментов оказалось, что оптимальный коэффициент усиления должен быть 200, проведя расчеты, можно вычислить сопротивление R1=248Ω. Ближайший резистор из таблицы номиналов 240Ω, его и используем, при этом коэффициент усиления первичного каскада G составил 206.8, что вполне соответствует требованиям.

Конденсатор C3 предназначен для фильтрации постоянной составляющей сигнала, кроме того R-C-цепочка, работающая как

фильтр высоких частот; образует завал амплитудно-частотной характеристики в области нижних частот. Емкость C3 [мкФ] была рассчитана по формуле (5):

$$C3=1/(2*Pi*R7*f),$$

где R7 - сопротивление резистора R7;

f-частота аналогового сигнала равная 100 Гц.

Такая частота обоснована проведенными исследованиями диапазона частот входного сигнала от 100 Гц до 1000 Гц.

Последующие каскады схемы устроены на счетверенном операционном усилителе TL084 (6). Второй каскад использует один из операционных усилителей U2:B, включенный с коэффициентом усиления 1,фактически инвертор (7), рассчитан по формуле

$$G=R8/R7,$$

где R8 и R7 - сопротивления резисторов R7 и R8 соответственно.

Следующим каскадом схемы являются диоды: D1, скомпонованные в одном корпусе - BAS40W-04. Диоды предназначены для разделения сигнала на положительный и отрицательный полупериоды. Сигналы попадают на неинвертирующий и инвертирующий входы операционного усилителя U2:C соответственно. Данный ОУ складывает сигналы, без усиления.

Схема на U2:D является фильтром низких частот, пропуская сигнал до 1 кГц.

$$C4=1/(2*Pi*R13*f),$$

где R13 - сопротивление резистора R13;

f-частота переменного тока.

Ближайшие к расчетным значения: R13=82kΩ, C4=1мкФ.

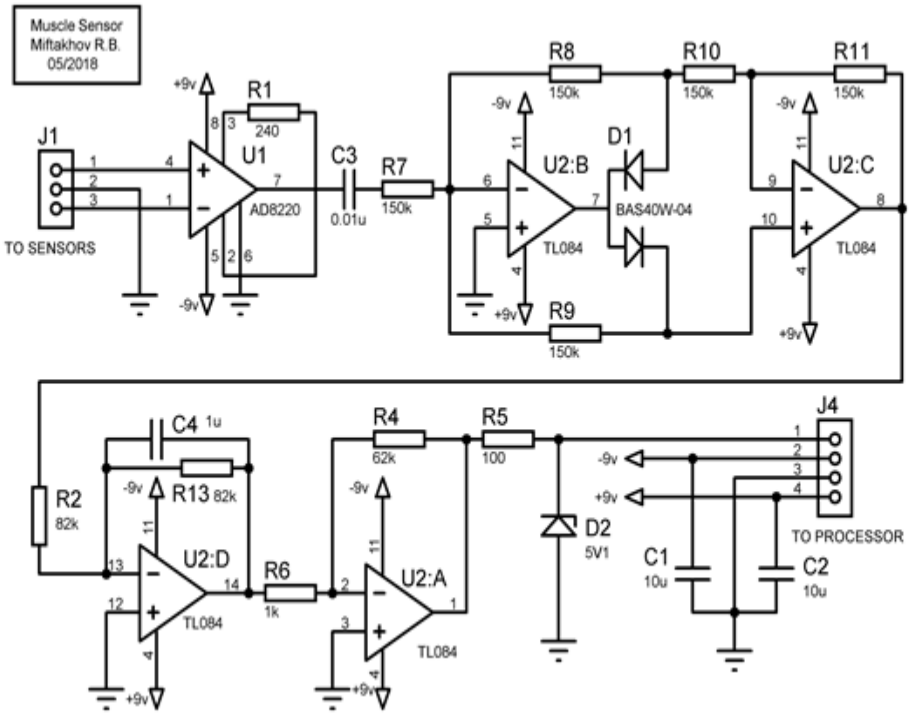
Следующий каскад имеет коэффициент усиления G

$$G=R4/R6=62,$$

где R4 и R6 - сопротивления резисторов R4 и R6 соответственно.

Сигнал на нем инвертируется. Данный каскад предназначен для усиления сигнала до уровня возможного для оцифровки (от 0 до 5 вольт) .

Резистор R5 является гасящим резистором и предназначен для согласования уровня, который будет ограничен (не выше 5.1 вольт) стабилитроном D2. Данное ограничение призвано защитить последующие блоки схемы.



"Схема электрическая принципиальная"

5.3 Сборка и изготовление датчика

Для минимизации габаритов прототипа, было решено изготовить датчик используя поверхностные монтаж, для этого были подобраны детали соответствующих номиналов.

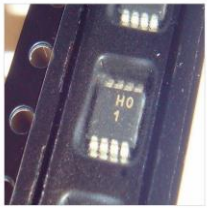
Таблица 1 - используемые радиодетали.

Наименование,	Номинал	Корпус	Количество	Обозна-чение на

тип				схеме
Инструментальный ОУ	AD8220	MSOP	1	U1
ОУ	TL084	S0-14	1	U2
Резистор	240Ω	0805	1	R1
Резистор	82kΩ	0805	2	R2, R13
Резистор	150kΩ	0805	5	R7, R8, R9, R10, R11
Резистор	1kΩ	0805	1	R6
Резистор	62kΩ	0805	1	R4
Резистор	100Ω	0805	1	R5
Диодная сборка	BAS40W-04	SOT323	1	D1
Стабилитрон	BZX84B5	SOT23	1	D2

	V1			
Конденсатор	0.01uF	0805	1	C3
Конденсатор	1uF	0805	1	C4
Конденсатор	10uF	0805	2	C1, C2

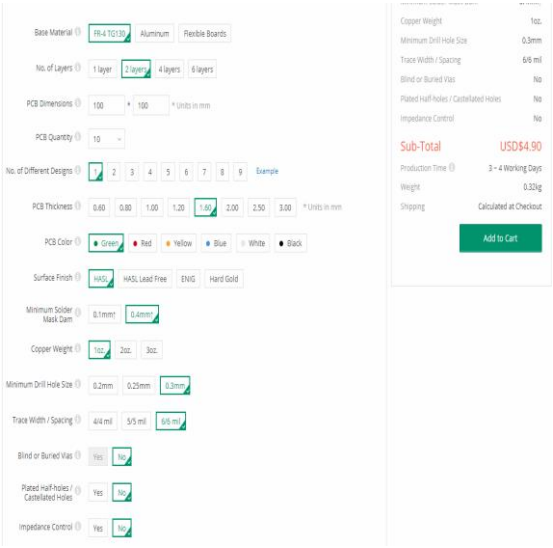
В программе “Proteus” была разведена плата, которую изготовили на заводе “Itead”. Комплектующие приобретены на “aliexpress.com” и в магазине “Чип и Дип”.

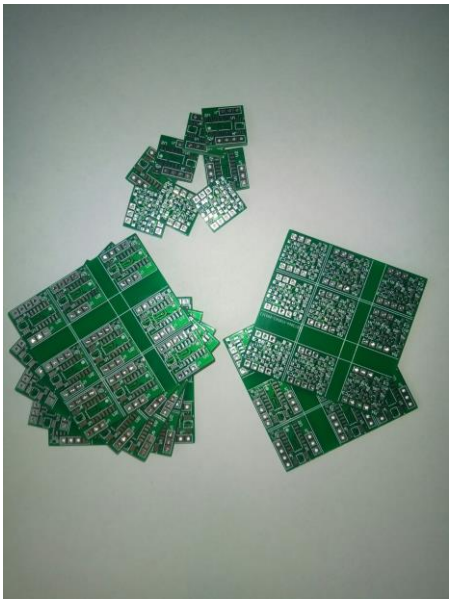


Наименование		Условия поставки*	Цена	Кол-во	Сумма
BZX84B5V1LT1G, Диод Зенера, BZX84C, 5.1 В, 225 мВт, SOT-23, 2 %, 3 вывод(-ов), 150 °C ON Semiconductor, Одиночные Диоды Зенера		2-3 недели *Потребуется 100% предоплата.*	18 руб.	10	180 руб.
BAS40W-04, Диод Шоттки малого сигнала, Двойной Последовательный, 40 В, 200 мА, 1 В, 600 мА, 150 °C Multicomp, Диоды - Шоттки Маленьких Сигналов		2-3 недели *Потребуется 100% предоплата.*	9 руб.	10	90 руб.
0.125Вт 0805 82 кОм, 1%, Чип резистор (SMD) Тайвань, Чип резисторы (SMD, для поверхностного монтажа)		1 рабочий день	0.90 руб.	20	18 руб.
0.125Вт 0805 150 кОм, 1%, Чип резистор (SMD) Тайвань, Чип резисторы (SMD, для поверхностного монтажа)		1 рабочий день	0.90 руб.	50	45 руб.
0.125Вт 0805 240 Ом, 5%, Чип резистор (SMD) Тайвань, Чип резисторы (SMD, для поверхностного монтажа)		1 рабочий день	0.90 руб.	10	9 руб.
0.125Вт 0805 8.2 кОм, 1%, Чип резистор (SMD) Тайвань, Чип резисторы (SMD, для поверхностного монтажа)		1-2 рабочих дня	0.90 руб.	10	9 руб.
0.125Вт 0805 1 кОм, 1%, Чип резистор (SMD) Тайвань, Чип резисторы (SMD, для поверхностного монтажа)		4 рабочих дня	0.90 руб.	10	9 руб.
0.125Вт 0805 100 Ом, 1%, Чип резистор (SMD) Тайвань, Чип резисторы (SMD, для поверхностного монтажа)		2-3 недели *Потребуется 100% предоплата.*	5 руб.	10	50 руб.
Кер.ЧИП конд. 0.01 мкФ X7R 50В.10%, 0805 Тайвань, Конденсаторы керамические SMD		1 рабочий день	3 руб.	10	30 руб.
Кер.ЧИП конд. 1 мкФ X7R 10% 16В 0805, GRM21BR71C105K Murata, Конденсаторы керамические SMD		1 рабочий день	4 руб.	10	40 руб.
Кер.ЧИП конд. 10мкФ X5R 10%6.3В 0805, GRM21BR60J106K Murata, Конденсаторы керамические SMD		1 рабочий день	4 руб.	20	80 руб.
TL084CD, Универсальный 4-х канальный операционный усилитель с полевым входом [SO-14] Texas Instruments, Усилители – Инструментальные, Операционные, Буферные		1 рабочий день	16 руб.	10	160 руб.

Всего 12 товаров на сумму:
720 руб.
К оформлению

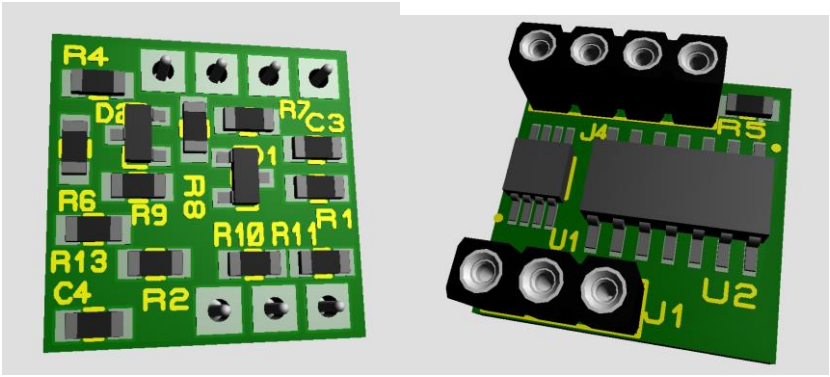
“Приобретение деталей”


Base Material
FR-4 TG130
Aluminum
Flexible Boards
No. of Layers
1 layer
2 layers
4 layers
6 layers
PCB Dimensions
100
100
* Units in mm
PCB Quantity
10
No. of Different Designs
1
2
3
4
5
6
7
8
9
Example
PCB Thickness
0.60
0.80
1.00
1.20
1.60
2.00
2.50
3.00
* Units in mm
PCB Color
Green
Red
Yellow
Blue
White
Black
Surface Finish
HASL
HASL Lead Free
ENIG
Hard Gold
Minimum Solder Mask Dam
0.1mm↑
0.4mm↑
Copper Weight
1oz
2oz.
3oz.
Minimum Drill Hole Size
0.2mm
0.25mm
0.3mm↑
Trace Width / Spacing
4/4 mil
5/5 mil
6/6 mil
Blind or Buried Vias
Yes
No
Plated Half-holes / Castellated Holes
Yes
No
Impedance Control
Yes
No
Copper Weight
1oz.
Minimum Drill Hole Size
0.3mm
Trace Width / Spacing
6/6 mil
Blind or Buried Vias
No
Plated Half-holes / Castellated Holes
No
Impedance Control
No
Sub-Total
USD$4.90
Production Time
3 – 4 Working Days
Weight
0.32kg
Shipping
Calculated at Checkout
Add to Cart



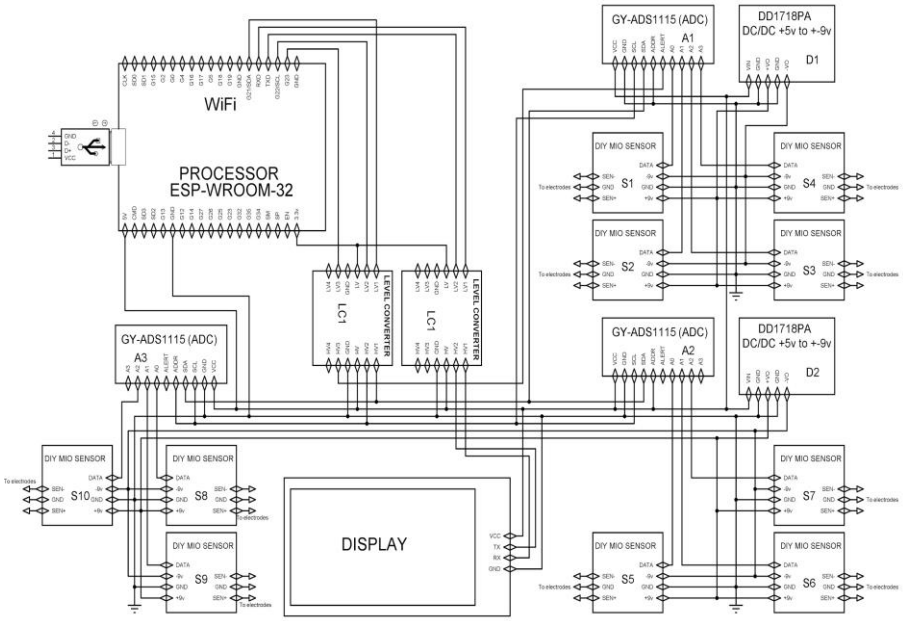
“Заказ и изготовленные платы датчика”

Сборка датчика произведена в домашних условиях. Не учитывая сборку, себестоимость собранной платы датчика составила 277 рублей.



“Датчик”

5.4 Структурная схема устройства



“Структурная схема”

Более подробно Приложение А.

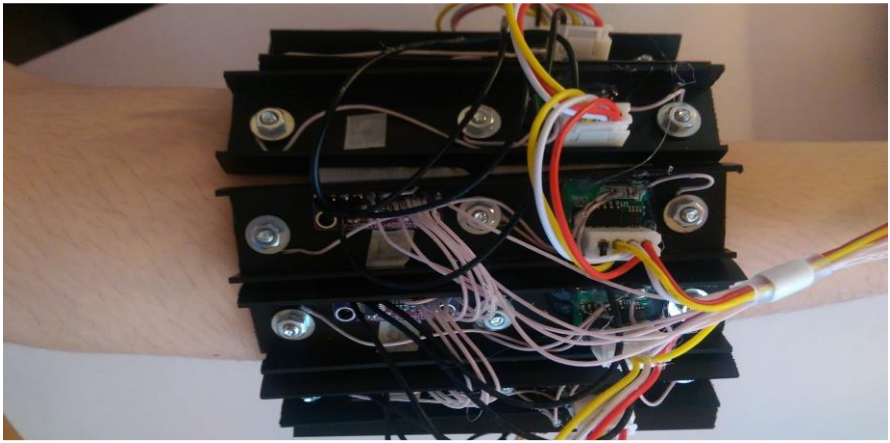
Сигналы полученный с электродов обрабатывается платами датчика (S1-S10). Выходы датчиков подключены к 4-х канальным аналого-цифровым преобразователям (АЦП), на схеме: A1-A3, каждый датчик к отдельному каналу. АЦП по шине I2C, передают оцифрованную информацию на процессорное устройство, по запросу процессора. Дисплей отображает информацию по команде ЦПУ. В связи с тем, что уровни сигналов дисплея и АЦП 0-5 вольт, а уровни процессора 0-3.3 вольта, необходимо применить согласующие устройства в виде преобразователей уровня (LC1, LC2).

Основной источник питания схемы имеет напряжение в 5 вольт, плата процессора содержит встроенный преобразователь из 5 в 3.3

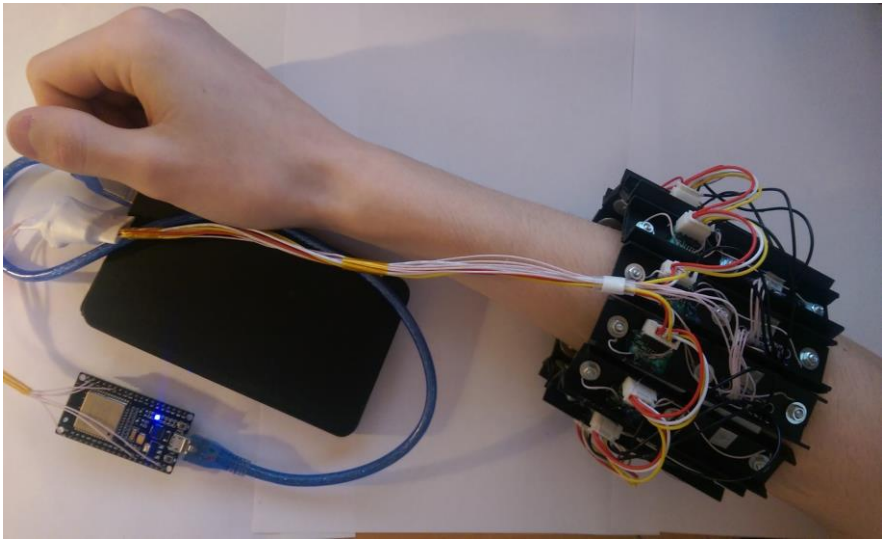
вольт. Модулям датчиков необходимо двуполярное питание 9 вольт. Поэтому в схеме используется 2 преобразователя питания (D1, D2) из 5 вольт в +-9 вольт.

Таблица 2 - блоки структурной схемы “Комплекс датчиков мышечной активности”.

Название блока	Назначение	Количество	Обозначение
ESP-WROOM-32	Центральный процессор	1	PROCESSOR
GY-ADS1115(ADC)	Аналого-цифровой преобразователь	3	A1, A2,A3
DIY MIO SENSOR	Измерение и первичная обработка сигнала	10	S1, S2, S3, S4, S5, S6, S7, S8, S9, S10
NX4832T035_011	Дисплей	1	DISPLAY
LEVEL CONVERTER	Преобразователь уровней	2	LC1, LC2
DD1718PA	Преобразование напряжения +5 вольт в +-9 вольт	2	D1, D2



“Устройство на руке”



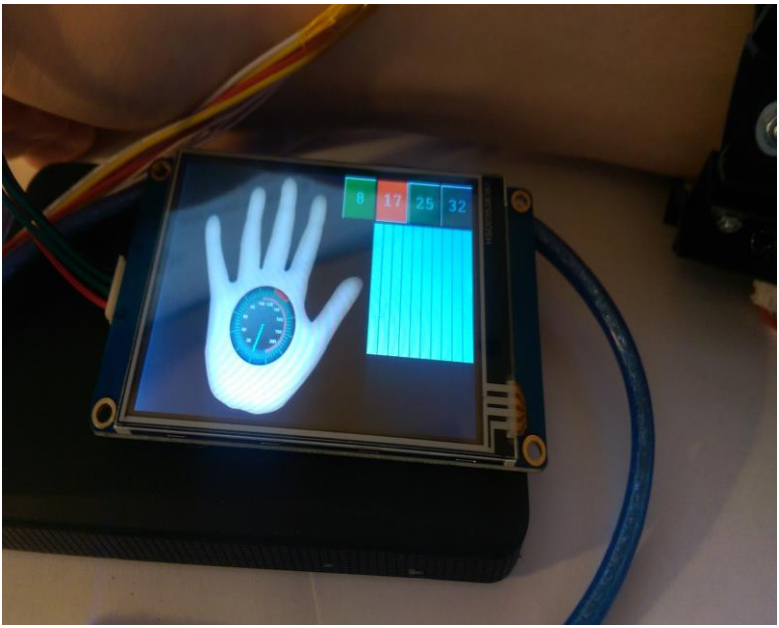
“Включенное устройство”

5.5 Программа контроллера миографических датчиков

Программа написана на языке C++ в среде разработки Arduino IDE (11). Приложение Б. Программа имеет два режима: режим настройки “калибровка” и основной режим “функционирование”. Микропроцессор, вне зависимости от режима работы, с частотой 30 Гц получает измерения с 10-ти сенсоров.

5.5.1 Этап калибровки

На этапе калибровки оператору требуется, в соответствии с сценарием, производить жесты пальцами. Показания датчиков при каждом жесте объединяются в n-мерный массив значений. Полученные данные проходят нормировку относительно максимального значения в последовательности, обработанный массив сохраняется в матрице как ее строка, где номер строки равен соответствует жесту в словаре. Среднеарифметическое значение показаний всех датчиков записывается в массив параллельно калибровочной матрице и будет использовано в дальнейшем для определения усилия сжатия. Словарь представляет из себя массив из 32 жестов (2^5=32 комбинации) отсортированный по частоте использования. В конце калибровки получается матрица 32x10. В дальнейшем для удобства обработки, строка матрица воспринимается как 10-мерный вектор. Калибровка осуществляется с помощью дисплея и демонстрационного стенда. Во время калибровки желательно максимально напрягать мышцы для получения наилучшего распознавания.



“Дисплей на этапе калибровки”

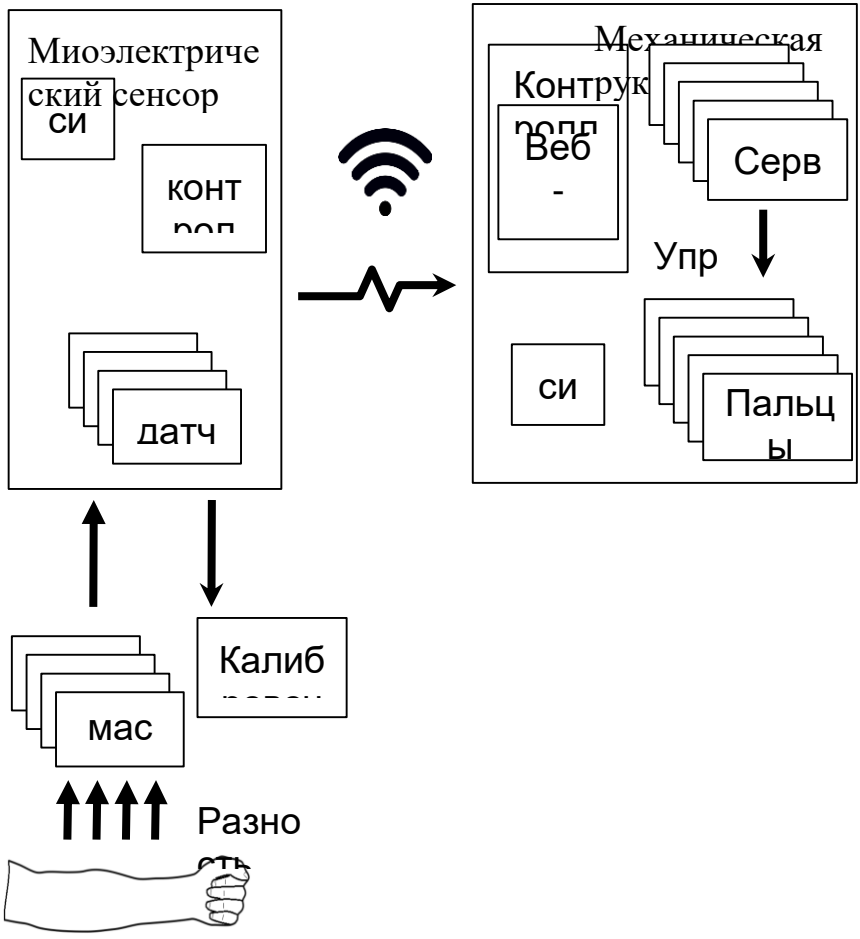
5.5.2 Основной цикл работы

На этом этапе программа циклически опрашивает каждый из датчиков и формирует текущий 10-ти мерный вектор, далее осуществляется поиск вектора с наименьшим отклонением от исходного. Графически процесс нормализации можно представить, как определение точки на поверхности гиперсферы 10-мерного пространства, соответствующей определенному вектору. Помимо текущего измерения на сфере можно также отобразить точки, соответствующие различным жестам, вычисленные на этапе калибровки. Найдя точку в калибровочной матрице, которая в совокупности с измеренной даст вектор с наименьшей длиной, можно определить какому жесту соответствует полученный вектор, а сила

сжатия руки влияет на усредненные значения сигналов и соответственно угол отклонения пальцев механической руки. Вычисленные данные визуализируются на дисплее и по http-протоколу отправляются на демонстрационный web-сервер.

5.6 Демонстрационный стенд

Демонстрационный стенд состоит из исполнительного устройства - механической руки с контроллером управления. Структурная схема системы представлена ниже.



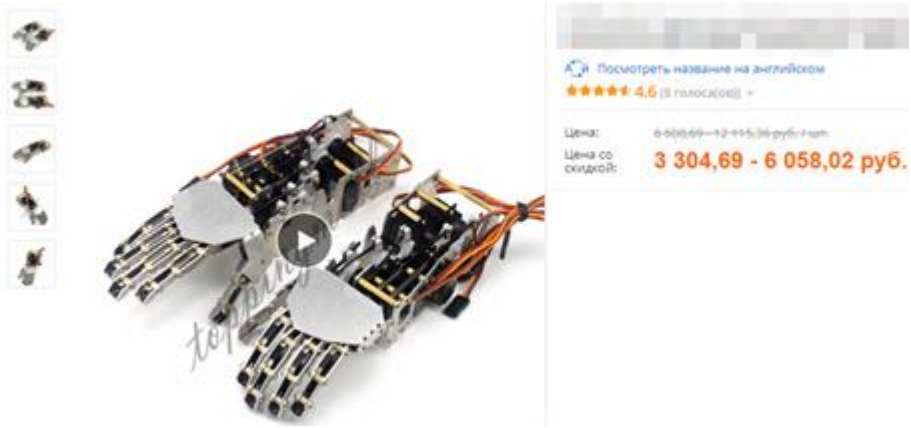
“Обобщенная блок-схема”

5.7 Описание механической руки

Механическая рука была заказана из Китая (2), но также полноразмерный протез или макет может быть напечатан на 3д-принтере. Помимо протезирования данная система может

использоваться для жестового управления другими машинами и механизмами, в виртуальной и дополненной реальности.

В движение пальцы механической руки приводятся с помощью сервоприводов. Они были заказаны отдельно, так как те, которые шли в комплекте оказались ненадлежащего качества.

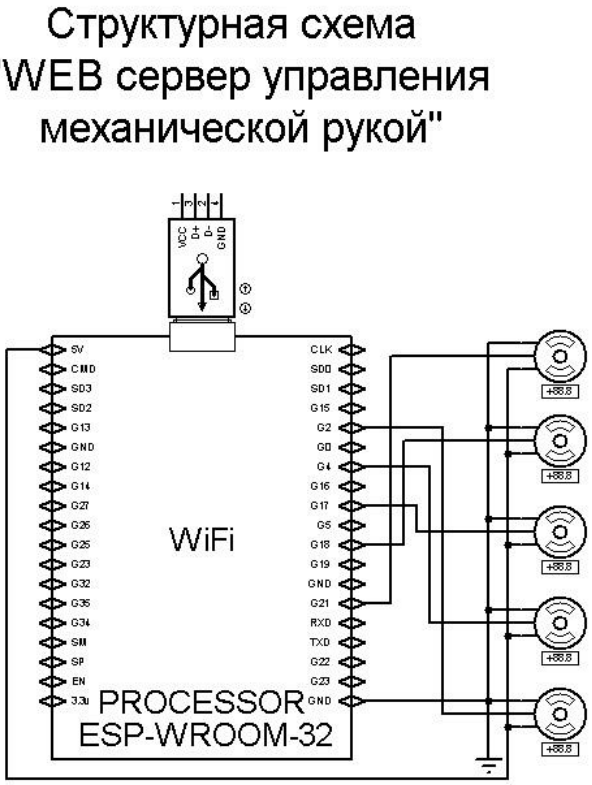


“Используемая механическая рука”

“Используемые сервоприводы”

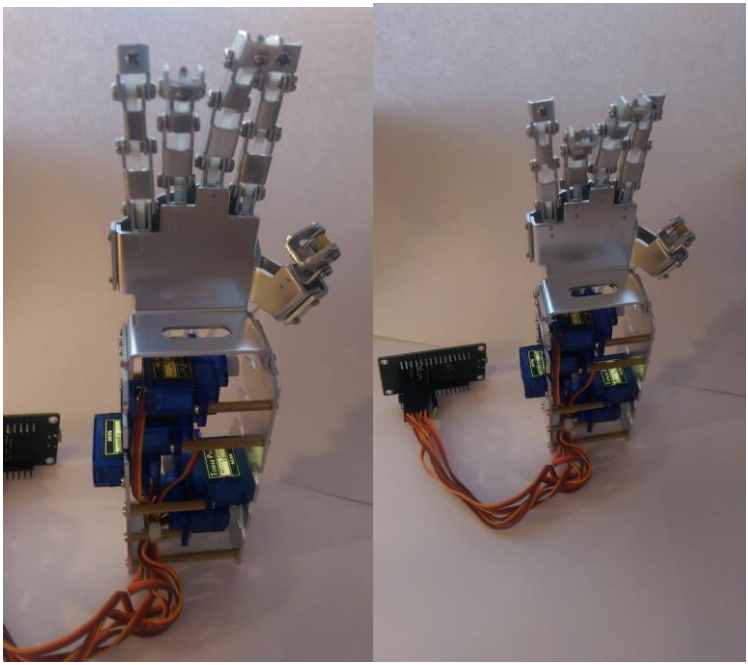
Была написана программа в среде разработки Arduino IDE для контроллера EPS32/ESP8266 (9), которая принимает команды через web-интерфейс и приводит пальцы в движение.

Сервопривод управляется с помощью библиотеки servo.h, прямым указанием угла отклонения пальца. Программа, работающая на сервере-руке, является вспомогательной и не заслуживает глубокого описания в рамках данной научно-исследовательской работы.



С программой можно ознакомиться в Приложении В к данной работе. Назначение программы - получить управляющую команду от “клиента”, которым является основная программа и выполнить ее.

Кроме того, как дополнение к основной программе визуализации, используется дисплей компании “Nextion” (10), который принимает простейшие управляющие команды от основной программы по проводному последовательному интерфейсу и дублирует показания датчиков и действия механической руки. К сожалению, текст программы из-за особенности организации дисплея, не может-быть распечатан в читаемом виде.



“Собранная механическая рука”

5.8 Взаимодействие стенда и основного устройства.

Основное устройство измеряет миографическое состояние мышц, производит расшифровку данных. Определяет жест и транслирует его на демонстрационный стенд в виде HTML POST запроса.

Формат запроса “POST/?N=XX...&N=XX”, где N-номер пальца (от 1 до 5), а XX - степень сгибания (от 0 до 100 %). Пальцы пронумерованы так, что 1 - мизинец, а 5 - большой. После получения запроса стенд преобразует его в управляющие воздействия и применяет их к системе сервоприводов, которые сгибают и разгибают пальцы механической руки.

```
l?1=100&2=100&3=100&4=100&5=100
```

“Запрос”

Finger server

FINGER	1	2	3	4	5	
MAX	103	123	141	0	0	=>
VAL	100	100	100	100	100	=>
MIN	35	65	65	65	60	=>

“Сервер”

6 Заключение

На основании проведенных работы и анализа можно утверждать, что использование ЭМГ датчиков, на современной радио-элементной базе, позволяет сконструировать недорогой высококачественный инструмент, дающий возможность создать “дружественный” и удобный интерфейс управления.

Проведенные исследования показали “жизнеспособность” решения на базе миографических датчиков. Невзирая на существующие сложности, однозначно можно утверждать, что применение этой разработки в ближайшем будущем рекомендовано и будет востребовано.

В результате получено уникальное устройство, способное корректно определять пальцевые жесты, степень напряжения мышц.

Дальнейшее исследование необходимо посвятить разработке более эргономичной конструкции устройства в целом и совершенствованию алгоритмов распознавания жестов. Идеально было бы достичь такого результата, когда обосновались-бы полусогнутые пальцы, что дало бы еще большее расширение сферы применения устройства.

7 Список используемых источников

1 - https://www.bsmu.by/downloads/kafedri/k_anatomia/stud/2017-2/mr13.pdf

2 - https://aliexpress.com/

3 - http://www.bitronicslab.com/

4 - Техническая документация на ОУ AD8220

©2006 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners. C03579-0-4/06(0)

5 - Р. Кофлин Ф. Дрискол ОПЕРАЦИОННЫЕ УСИЛИТЕЛИ И ЛИНЕЙНЫЕ ИНТЕГРАЛЬНЫЕ СХЕМЫ Издательство “Мир”, Москва

6 - Техническая документация на ОУ TL084

2001 STMicroelectronics - Printed in Italy - All Right Reserved

STMicroeelectronics GROUP OF COMPANIES

7 - Г.И. Пухальский Т.Я. Новосельцева Справочник: ”Проектирование дискретных устройств на интегральных микросхемах”

8 - С.Т. Хвощ Н.Н. Варлинский Е.А. Попов “МИКРОПРОЦЕССОРЫ и МИКРОЭВМ в системах автоматического управления”

9 - https://github.com/espressif/

10 - https://nextion.itead.cc/

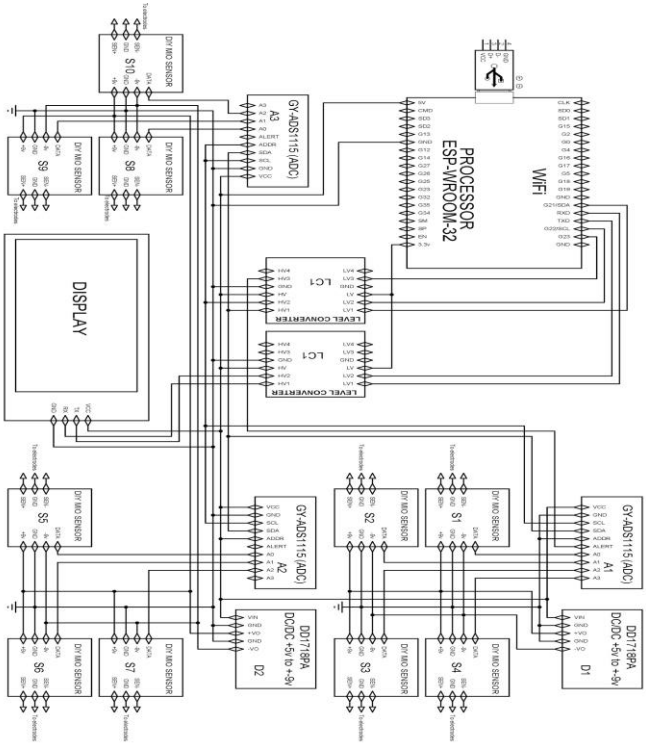
11 - https://www.arduino.cc/

https://ru.wikipedia.org/

http://www.lomonosov-fund.ru/enc/ru/encyclopedia/

Приложение А

“Блок схема основного устройства”



Приложение Б

“Программа основного устройства”

#include "ads1115.h" // чтение показаний 12 каналов датчиков

#include "radiosend.h" // передача команд о состоянии пальцев на манипулятор

#include "screenobj.h" // для подключения дисплея

#define SENSORS 10 // всего 10 датчиков

#define MATRSIZE 32 // размер матрицы пальцы - по два состояния 2 в степени 5

#define VECTORS // поиск осуществляется с помощью векторной математики

#define THRESHOLD_OK 80 // порог опознавания комбинации при поиске по матрице

#define THRESHOLD_DEV 15 // порог допустимых колебаний показаний, при удержании комбинации

uint8_t oldGauge[SENSORS]={0}; // на дисплее есть всего 10 градусникjd

uint8_t matr[MATRSIZE][SENSORS]; // матрица значений датчиков в процентах от показаний, 0,1,2,3 - показания датчиков, 100% - выбирается датчик с самыми мощными показаниями, строка матрицы соответствует пальцу

int16_t matrPow[MATRSIZE]; // в этом буфере сохраняем максимальные показания датчика для данного слепка, используем для вычисления усилия

uint8_t power=0; // ориентировочное усилие мышцы относительно эталонного

uint8_t oldPower=255; // предыдущее (старое) ориентировочное усилие мышцы относительно эталонного, для исключения повторной обработки

uint8_t sysStat=0; // системный статус сработки датчиков

uint8_t oldSysStat=0; // старый статус сработки датчиков

uint8_t calibrate=0; // статус калибровки

uint32_t calibrateDelay=0; // количество времени на текущую операцию калибровки

uint32_t calibrateHalfDelay=0; //буфер половинного времени стадии, время для сохранения измерений

uint32_t calibrateTimer=0; // буфер засечки этапов калибровки

uint8_t maxDest=0; //количество жестов на которое настраиваем устройство

int16_t adc_data[12];   // полученные данные

int16_t adc_max[12];   // максимумы данных

uint8_t fingerStat[MATRSIZE]={   //карта состояния пальцев большой,указательный,средний,безымянный, мизинец

  B00001, // всего 8 простейших жестов

  B00010,

  B00100,

  B01000,

  B10000,

  B11111,

  B00000, // всего 17 простых жестов

B01111,

B00111,

B00011,

B11000,

B11100,

B01100,

B01001,

B10001,

B11001,

B01011,

// сложные жесты

B11110, // 25 простых+сложных

B11101,

B11011,

B01101,

B10011,

B01110,

B10111,

B10101,

```
  // неудобные жесты
  B11010, // всегшо 32 комбинации пальцев
  B01010,
  B00101,
  B00110,
  B10010,
  B10100,
  B10110,
};
// управление показом стадий калибровки
void calibrateStage(byte stage){
  calibrate=stage; // установить новый этап калибровки
  calibrateTimer=millis(); //засечь таймер этапа
  byte st=128; // буфер статуса пальцев для показа на экране
  word delayGaug=50; // шаг таймера градусника
  byte stepGaug=1; // шаг градусника
  String msg=""; //буфер сообщения для показа на экране
  if (stage==1) { // стадия запуска калибровки
    powerGauge.setValue(256); // погасить спидометр
```

```
    powerTimer.enable(); // запустить таймер обработки спидометра
    for(byte i=0;i<SENSORS;i++){
      //shift[i]=0; // сброс смещения графика относительно ноля
      gr[i].setValue(0); // сброс проресс баров на экране
      oldGauge[i]=0;
      //_max[i]=1;
    }
    msg=F("Сожмите руку в кулак");
    calibrateDelay=4000; // установить продолжительность этапа 5 сек
    delayGaug=80; // шаг таймера градусника каждые 80 us
    stepGaug=2; // шаг градусника по два деления (80*100/2 = 4000)
    st+=1+2+4+8+16;
  } else if (stage==2) { //
    msg=F("Полностью расслабьте руку");
    calibrateDelay=1500; // установить продолжительность этапа 1,5 сек
    delayGaug=60; // шаг таймера градусника каждые 60 us
    stepGaug=4; // шаг градусника по два деления (60*100/4 = 4000)
  } else if (stage==3) {
    //for(byte i=0;i<SENSORS;i++){
```

```
      //shift[i]=adc_data[i]; // запоминаем смещение графиков относительно ноля
      //adc_data[i]=0; // применяем смещение к текущим данным
      //_max[i]=0; // чистим буфер максимумов
    //}
  } else if (stage<(3+(maxDest*2))){
    stage-=3;
    if ((stage&1)==0){ // четные стадии (сгибания)
      msg=F("Согните пальцы выделенные красным");
      calibrateDelay=4000;
      delayGaug=80; // шаг таймера градусника каждые 80 us
      stepGaug=2; // шаг градусника по два деления (80*100/2 = 4000)
      st+=fingerStat[stage/2]; // картина пальцев из таблицы
    } else {
      msg=F("Разогните...");
      calibrateDelay=1500;
      delayGaug=60; // шаг таймера градусника каждые 60 us
      stepGaug=4; // шаг градусника по два деления (60*100/4 = 4000)
    }
```

```
} else { // окончание калибровки
  msg=F("Калибровка окончена");
  calibrateDelay=1000;
  delayGaug=50; // шаг таймера градусника каждые
  stepGaug=100; // шаг градусника по два деления
  LearnButton8.setValue(1); // "отпустить кнопку обучения"
  LearnButton17.setValue(1); // "отпустить кнопку обучения"
  LearnButton25.setValue(1); // "отпустить кнопку обучения"
  LearnButton32.setValue(1); // "отпустить кнопку обучения"
  calibrate=0; // переход в нормальную работу
}
calibrateHalfDelay=calibrateDelay/2;//половина времени стадии
UniToIso(&msg); // перекодировать сообщение в буфер
ShowText.setText(msg.c_str()); // напечатать на дисплее
fingerScreenStat.setValue(st); // установить статус показа пальцев
metalHandSetFingers(st);
nexClick(ShowText.getObjName(),0); //активировать статус
gaugeStep.setValue(stepGaug);
gaugeTimer.setCycle(delayGaug);
```

```
  gaugeTimer.enable();
}
// запуск калибровки
void startCalibrate8(void *ptr){
  if (calibrate==0) { //калибровка не запущена
    calibrateStage(1); //запускаем калибровку первая стадия калибровки
    maxDest=8; // учимся всего восьми комбинациям
  }
}
void startCalibrate17(void *ptr){
  if (calibrate==0) { //калибровка не запущена
    calibrateStage(1); //запускаем калибровку первая стадия калибровки
    maxDest=17; // учимся всего 17 комбинациям
  }
}
void startCalibrate25(void *ptr){
  if (calibrate==0) { //калибровка не запущена
    calibrateStage(1); //запускаем калибровку первая стадия калибровки
    maxDest=25; // учимся всего 25 комбинациям
```

```
  }
}
void startCalibrate32(void *ptr){
  if (calibrate==0) { //калибровка не запущена
    calibrateStage(1); //запускаем калибровку первая стадия калибровки
    maxDest=32; // учимся 32 комбинациям
  }
}

// расчет в процентах
uint8_t getProc(int16_t val, int16_t maxVal){
  if (val<0) { // отрицательные значения недопустимы, делить на 0 нельзя
    return 0;
  } else if (maxVal<1) {
    return val;
  }
  return (((long)val*100)/maxVal);
}
```

```
void setup(){
  Serial.begin(115200);
  Serial.println(F("Hallo"));
  nexInit();// инициализация дисплея
  String msg=F("Подключаемся к серверу !");
  ShowText.setText(msg.c_str());
  uint8_t test=1;
  while(test==1){ // запуск связи с сервером-рукой
    test=metallHandConnect();
    if(test==1) {
      msg=F("Попытка подключения к серверу");
      ShowText.setText(msg.c_str());
    }
  }
  if(test==2) {
    msg=F("Ошибка. Работаем без сервера !");
    delay(5000);
  } else {
    msg=F("Подключение успешно !");
```

```
      metalHandSetFingers((uint8_t)0); // выпрямить пальцы руки
    }
    ShowText.setText(msg.c_str());
    adcBegin(20); //запуск системы модулей ADC, с правильным коэфициентом измерений, для получения показаний
//    adc_data=getData();
//    adc_max=getMax();
    LearnButton8.attachPop(startCalibrate8);       // подключаем на виртуальную кнопку на дисплее подпрограмму запуска калибровки на 8 жестов
    LearnButton17.attachPop(startCalibrate17);     // подключаем на виртуальную кнопку на дисплее подпрограмму запуска калибровки на 17 жестов
    LearnButton25.attachPop(startCalibrate25);     // подключаем на виртуальную кнопку на дисплее подпрограмму запуска калибровки на 25 жестов
    LearnButton32.attachPop(startCalibrate32);     // подключаем на виртуальную кнопку на дисплее подпрограмму запуска калибровки на 32 жеста
}
void loop() {
  // постоянно проверяем нет ли какого действия с элементами экрана
  nexLoop(nex_listen);
```

```
  // показываем прогресс бары на дисплее
  static uint8_t pbCount=0;
  long temp=getProc(adc_data[pbCount],adc_max[pbCount]); // 
расчитываем значение показаний датчика для вывода на экран в виде 
прогрессбара, в процентах от максимума
  // показываем показания на градуснике на экране
  if (oldGauge[pbCount]!=temp) {
    if (temp<=100) { // значения больше 100 воспринимаются 
монитором как ошибка
       gr[pbCount].setValue(temp);
       oldGauge[pbCount]=temp;
    }
  }
  if (pbCount<(SENSORS-1)){ // счет от 0 до 11
    pbCount++;
  } else {
    pbCount=0;
  }
  // обработка данных с датчиков
  while (adcAvailable()){ // данные готовы
```

```
  for(uint8_t i=0; i<10; i++){
    Serial.print(adc_data[i]);
    Serial.print(' ');
  }
  Serial.println();
   // если идет этап калибровки калибровки
   if (calibrate) {// идет калибровка
    if (millis()-calibrateTimer > calibrateDelay) {  // конец стадии калибровки
      calibrateStage(calibrate+1); // начало следующего этапа калибровки
    } else if ((calibrateHalfDelay!=0) && (millis()-calibrateTimer>calibrateHalfDelay)){ // середина стадии
      calibrateHalfDelay=0; // обнулить таймер сохранения данных, что бы не повторять операции
      // ищем максимумы измерений среди датчиков
      int16_t _maxBuff=1; // максимум стабилизации датчиков, не может быть меньше 1, поскольку на него будем делить и это проценты
      int32_t _maxAbs=0;  // абсолютный максимум усилия для матрицы
```

```
uint8_t k=(calibrate-3)/2; // вычисляем номер комбинации в таблице

if (k<MATRSIZE) {

  for(uint8_t i=0;i<SENSORS;i++){ // перебираем показания датчиков

    if (_maxBuff<adc_data[i]) { // ищем максимальные

      _maxBuff=adc_data[i];

    }

    _maxAbs+=(adc_max[i]); // суммируем все максимумы показаний датчиков для усреднения

    adc_max[i]=0; // сбрасываем максимумы для сл. измерения

  }

  _maxAbs/=SENSORS; //получаем среднее среди всех показаний

  // заполняем матрицу измерений датчиков в процентах от максимума

  for (uint8_t i=0;i<SENSORS;i++){

    matr[k][i]=getProc(adc_data[i],_maxBuff);

  }

  // сохраняем показания соответствующие абсолютному максимуму данной матрицы пальца
```

```
        if (_maxAbs<1){
          matrPow[k]=1; // поскольку на это число будем делить и поскольку это проценты то не может быть меньше 1
        } else {
          matrPow[k]=_maxAbs;
        }
      } else {
        Serial.println(F("Error calc matrix size !"));
      }
    }
  } else { // процесс работы (калибровка отключена)
    // ПОДБИРАЕМ НАИБОЛЕЕ ПОДХОДЯЩУЮ СТРОКУ МАТРИЦЫ ТЕКУЩЕМУ СЛЕПКУ ИЗМЕРЕНИЙ
    // ищем максимум измерений среди датчиков
    int16_t _maxBuff=1; // на это число будем делить, не может быть 0 или отрицательным, потому что это процентные вычисления
    int32_t _maxAbs=0; // тут будем вычислять средние показания датчиков
    for(uint8_t i=0;i<SENSORS;i++){
      if (_maxBuff<adc_data[i]) { // ищем максимум
        _maxBuff=adc_data[i];
```

```
    }
    _maxAbs+=adc_data[i]; // накапливаем сумму для вычисления среднего
  }
  _maxAbs/=SENSORS;
  // заполняем текущий слепок (матрицу) измерения датчиков в процентах от максимума
  uint8_t currMatr[SENSORS];
  for (uint8_t i=0;i<SENSORS;i++){
    currMatr[i]=getProc(adc_data[i],_maxBuff);
  }
  #ifdef VECTORS
    // считаем, что каждая строка матрицы n-мерный вектор, поскольку все вектора нормализованы (имеют одинаковую длинну,
    // связано это с тем, что в матрице хранятся показания в процентах от максимального) можно искать ближайший вектор
    // по наименьшему расстоянию между концами векторов, которое можно найти через корень суммы квадратов разностей координат
    // одноименных плоскостей. Раз мы ищем минимум, а сами показания нас не интересуют, можно не брать корень (это долгая операция)
```

```
        uint8_t num=maxDest; // буфер номера ближайшей комбинации
        uint32_t min=4294967295;
        for (uint8_t j=0;j<maxDest;j++) { // перебираем образцовые матрицы
          int32_t test=0;
          for(uint8_t i=0;i<SENSORS;i++){
            int16_t buf=currMatr[i]-matr[j][i];
            test+=buf*buf;
          }
          if (min>test){
            min=test;
            num=j;
          }
        }
      #else
        // расчитываем веса отклонения полученой матрицы от образцовых и ищем с минимальным отклонением
        uint16_t min=65535; // для поиска минимумов
        uint8_t num=maxDest; // буфер номера ближайшей комбинации
```

```
        for (uint8_t j=0;j<maxDest;j++) { // перебираем образцовые матрицы

          uint16_t test=0; // тут будем накапливать вес отклонения

          for(uint8_t i=0;i<SENSORS;i++){  // перебираем датчики в матрице и текущие

            if (currMatr[i]>matr[j][i]){ // суммируем отклонения всех датчиков от эталона в процентах

              test+=currMatr[i]-matr[j][i];

            } else {

              test+=matr[j][i]-currMatr[i];

            }

          }

          if ((test<THRESHOLD_OK) && (min>test)) { // ловим минимум отклонения и границу порога чувствительности

            min=test;

            num=j; // при минимуме отклонений запоминаем подходящую строку матрицы

          }

        }

      #endif
```

```
if(num<maxDest){ // был найден подходящий элемент матрицы

  sysStat=fingerStat[num] ; //устанавливаем соответствующий пальцевый статус

} else {

  sysStat=0; // либо нет никакого статуса (рука разогнута)

}

// как дополнительный фильтр коротких выбросов используем порог силы, как порог изменения статуса

power=getProc(_maxAbs,matrPow[num]); // вычисляем ориентировочное усилие

if ((power<THRESHOLD_DEV) && (oldSysStat!=sysStat)) { // если это начало напряжения группы мышц, отличной от предыдущей, но усилие не достигло 15% , учитывать будем (фильтр коротких выбросов)

  sysStat=0; // игнорируем измененение статуса

}

// показываем статус на экране

if (oldSysStat!=sysStat) { // статус изменился, нужно изменить данные на экране

  oldSysStat=sysStat; // уравниваем, что бы не повторять

  fingerScreenStat.setValue(sysStat); // установить статус показа пальцев
```

metalHandSetFingers(sysStat,power); // отправить показания на сервер для работы манипулятора

nexClick(ShowText.getObjName(),0); //активировать статус пальцев на экране

// текстовое сообщение

String msg=""; // создать правильное сообщение о согнутых пальцах

```
if ((sysStat & (1+2+4+8+16))==(1+2+4+8+16)) {
  msg=F("КУЛАК");
} else {
  if (sysStat & 1) {
    msg+=F("МИЗИНЕЦ, ");
  }
  if(sysStat & 2){
    msg+=F("БЕЗЫМЯННЫЙ, ");
  }
  if(sysStat & 4){
    msg+=F("СРЕДНИЙ, ");
  }
  if(sysStat & 8){
```

```
            msg+=F("УКАЗАТЕЛЬНЫЙ, ");
          }
          if(sysStat & 16){
            msg+=F("БОЛЬШОЙ, ");
          }
          if (msg.length()>2){ // если в строке что то есть, удалить два последних символа (запятую и пробел)
            msg.remove(msg.length()-2, 2);
          }
        }
        UniToIso(&msg); // перекодировать сообщение в буфер
        ShowText.setText(msg.c_str()); // напечатать на дисплее
      }
      // отображение усилия на спидометре
      if (sysStat==0){ // если статус не опознан,
        power=0; // то и силу сжатия не показываем
      }
      if (oldPower!=power) { // только если усилие изменилось и не выходит за разумные пределы из-за очень маленького максимума
```

metalHandSetFingers(sysStat,power); // отправить показания на сервер для работы манипулятора, с указанием мощности сжатия

oldPower=power; // что бы не повторять дважды передачу одинаковых данных на экран

powerGauge.setValue(power); // передать данные спидометру на экране, коэфициент введен для увеличения размаха показаний, просто так нагляднее

powerTimer.enable(); // запустить таймер обработки спидометра

}

}

}

Приложение В

“Программа демонстрационного стенда”

```
#include "EEPROM.h"

#ifdef ESP8266 // для процессора esp8266

  #include <ESP8266WiFi.h>

  #include <WiFiClient.h>

  #include <ESP8266WebServer.h>

  #include <Servo.h>

  #define PIN_FINGER_0 D4 // пины подключения сервоприводов пальцев
```

```
  #define PIN_FINGER_1 D3

  #define PIN_FINGER_2 D2

  #define PIN_FINGER_3 D1

  #define PIN_FINGER_4 D0

  Servo serv[5];

#else  // для процессора ESP32

  #include <WiFi.h>

  #define DUTY_LOW 2000 // заполнение ШИМ при минимальном занчении сервопривода

  #define DUTY_HIGH 7800 //при максимальном

  #define PWM_FREQUENCY 50 //частота PWM

  #define PWM_RESOLUTION 16 //Разрешение PMW

  #define PIN_FINGER_0 19 // пины подключения сервоприводов пальцев

  #define PIN_FINGER_1 18

  #define PIN_FINGER_2 5

  #define PIN_FINGER_3 17

  #define PIN_FINGER_4 16

#endif
```

```
#define AP_WIFI // если эта строка раскоментирована, то наше устройство - точка доступа

#ifdef AP_WIFI

  #define AP_SSID   "FingerServer"  // параметры точки доступа

  #define AP_PASS   "fspassword"

#else

  #define STA_SSID   "Test"  // измените эти параметры на параметры вашей сети

  #define STA_PASS   "password"

#endif

#define EEPROM_SIZE 32 // максимальный размер сохраняемых в независимой памяти данных

WiFiServer server(80); //экземпляр класса веб сервер

byte f_max[5]={180,180,180,180,180}; //массив содержащий маскимальные значения положения серв пальцев (градусы)

byte f_min[5]={0,0,0,0,0};//массив содержащий минимальные значения положения серв пальцев (градусы)

byte fval[5]={50,50,50,50,50}; //массив содержащий текущие значения положения серв пальцев (проценты)

byte f_pin[5]={PIN_FINGER_0, PIN_FINGER_1, PIN_FINGER_2, PIN_FINGER_3, PIN_FINGER_4}; //массив содержаищй контакнты подключений сервомашин пальцев
```

```
// переменные для клиента:

char linebuf[80] ;// буффер получения данных от клиента

int charcount=0; // указатель на текущее место в буфере

void setFingerPos(byte n, byte pos){//подпрограмма установки пальца n в значение c (проценты)

  static uint16_t oldPos[5]={0};   // буфер хранения старых показаний

  if ((n<5) && (pos<=100)){

    #ifdef ESP8266

      uint8_t value=map(pos,0,100,f_min[n],f_max[n]);    //Текущее положение в градусах

      if (oldPos[n]!=value) {

        oldPos[n]=value;

        serv[n].attach(f_pin[n]);           // активируем серву

        serv[n].write(value);

      }

    #else

      uint16_t maximum=map(f_max[n],0,180,DUTY_LOW,DUTY_HIGH); //максимум в единицах PWM

      uint16_t minimum=map(f_min[n],0,180,DUTY_LOW,DUTY_HIGH); //Минимум в единицах PWM
```

```
      uint16_t value=map(pos,0,100,minimum, maximum);          //Текущее положение в единицах PWM

      if (oldPos[n]!=value) {

        oldPos[n]=value;

        ledcSetup(n,PWM_FREQUENCY,PWM_RESOLUTION); // настриваем канал PWM

        ledcAttachPin(f_pin[n], n); //подключаем каналы к выходам

        ledcWrite(n, value); //Устанавливаем серву в необоходимое положение

      }

   #endif

  }

}

void setup() {

  // инициализируем последовательную коммуникацию

  // и ждем открытия порта:

  Serial.begin(115200);

  while(!Serial) {

    ; // ждем подключения последовательного порта

     // (нужно только для штатного USB-порта)
```

```
  }
  EEPROM.begin(EEPROM_SIZE);
  for (byte i = 0; i < sizeof(f_max); i++){
    //настраиваем сервоприводы
    #ifdef ESP8266
      serv[i].attach(f_pin[i]);
    #else
      ledcSetup(i,PWM_FREQUENCY,PWM_RESOLUTION); //
настриваем канал PWM
      ledcAttachPin(f_pin[i], i); //подключаем каналы к выходам
    #endif
    //считываем сохраненные данные привода из eeprom
    byte temp=EEPROM.read(i);
    if (temp<181){
     f_max[i]=temp;
    }
    temp=EEPROM.read(sizeof(f_max)+sizeof(fval)+i);
    if (temp<181){
     f_min[i]=temp;
```

```
    }
    setFingerPos(i,fval[i]);
  }
  // начинаем с подключения к WiFi-сети:
  Serial.println();
  Serial.println();
  WiFi.disconnect();
  WiFi.softAPdisconnect();
  #ifdef AP_WIFI
    // работа как точка доступа
    WiFi.mode(WIFI_AP); // режим как точка доступа
    Serial.print("Create WIFI net: ");  //  "Подключаемся к "
    Serial.println(AP_SSID);
    WiFi.softAP(AP_SSID,AP_PASS); // запускаем нашу сеть
    Serial.print("Password for connect: ");  //  "Подключаемся к "
    Serial.println(AP_PASS);
    Serial.print("IP address: ");    //  "IP-адрес: "
    Serial.println(WiFi.softAPIP());
  #else
```

```
    WiFi.mode(WIFI_STA); // режим как станция
    Serial.print("Connecting to ");  //  "Подключаемся к "
    Serial.println(STA_SSID);
    WiFi.begin(STA_SSID, STA_PASS); // если к открытой сети, то заменить пароль на пустой
    while(WiFi.status() != WL_CONNECTED) {// пытаемся подключиться к WiFi-сети:
      delay(500);
      Serial.print(".");
    }
    Serial.println();
    Serial.println("WiFi connected");  //  "Подключение по WiFi выполнено"
    Serial.print("IP address: ");    //  "IP-адрес: "
    Serial.println(WiFi.localIP());
  #endif
  server.begin(); // запуск веб сервера
}
void loop() {
  WiFiClient client = server.available(); // анализируем канал связи, ждем подключения клиента:
```

```
 if (client) {
   Serial.println("New client");  //  "Новый клиент"
   memset(linebuf,0,sizeof(linebuf));// чистим буфер для клиента
   charcount=0;
   // HTTP-запрос заканчивается пустой строкой:
   boolean currentLineIsBlank = true; // переменная для контроля пустой строки
   while (client.connected()) {
     if (client.available()) {
       char c = client.read();
       linebuf[charcount]=c;// считываем HTTP-запрос, символ за символом в буфер
       if (charcount<sizeof(linebuf)-1) charcount++;
       // если добрались до конца строки (т.е. получили
       // символ новой строки) и строка пуста,
       // это значит, что HTTP-запрос закончился;
       // следовательно, можно отправлять ответ:
       if (c == '\n' && currentLineIsBlank) {
         // Формируем HTML страницу
         // отправляем стандартный заголовок HTTP-ответа:
```

```
client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

        //  "Тип контента: text/html"

client.println("Connection: close");

        //  "Соединение: отключено";

        //  после отправки ответа связь будет отключена

client.println(); // ниже отправка содержимого HTML страницы

client.println("<!DOCTYPE HTML><html><head>");

client.println("<meta name=\"viewport\"content=\"width=device-
width, initial-scale=1\"></head>");

client.println("<h1>Finger server</h1>");

client.println("<form>FINGER       1    
            2        
      3             
&nbsp4               
&nbsp5<BR>");

client.print("MAX     <input
type=\"number\"size=\"3\"name=\"1x\"min=\"0\"max=\"180\"value=\"");

client.print(f_max[0]);

client.print("\"><input
type=\"number\"size=\"3\"name=\"2x\"min=\"0\"max=\"180\"value=\"");

client.print(f_max[1]);
```

```
        client.print("\"><input
type=\"number\"size=\"3\"name=\"3x\"min=\"0\"max=\"180\"value=\"");

        client.print(f_max[2]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"4x\"min=\"0\"max=\"180\"value=\"");

        client.print(f_max[3]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"5x\"min=\"0\"max=\"180\"value=\"");

        client.print(f_max[4]);

        client.println("\"><input
type=\"submit\"value=\"=>\"title=\"send\"></form><form>");

        client.print("VAL       <input
type=\"number\"size=\"3\"name=\"1\"min=\"0\"max=\"180\"value=\"");

        client.print(fval[0]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"2\"min=\"0\"max=\"180\"value=\"");

        client.print(fval[1]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"3\"min=\"0\"max=\"180\"value=\"");

        client.print(fval[2]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"4\"min=\"0\"max=\"180\"value=\"");
```

```
        client.print(fval[3]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"5\"min=\"0\"max=\"180\"value=\"");

        client.print(fval[4]);

        client.println("\"><input
type=\"submit\"value=\"=>\"title=\"send\"></form><form>");

        client.print("MIN       <input
type=\"number\"size=\"3\"name=\"1n\"min=\"0\"max=\"180\"value=\"");

        client.print(f_min[0]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"2n\"min=\"0\"max=\"180\"value=\"");

        client.print(f_min[1]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"3n\"min=\"0\"max=\"180\"value=\"");

        client.print(f_min[2]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"4n\"min=\"0\"max=\"180\"value=\"");

        client.print(f_min[3]);

        client.print("\"><input
type=\"number\"size=\"3\"name=\"5n\"min=\"0\"max=\"180\"value=\"");

        client.print(f_min[4]);
```

```
        client.println("\"><input
type=\"submit\"value=\"=>\"title=\"send\"></form></html>");
        break;
      }
      if (c == '\n') {// началась новая строка
        currentLineIsBlank = true; // начинаем новую строчку:
        if (strstr(linebuf,"GET /?") > 0){// если это строка GET запроса
          char* start=strstr(linebuf,"?"); // ищем в ней наличие данных
          while (start) {
            byte number=0; // сюда считаем номер пальца
            byte type=0; // тут будет признак данных
            int value=255; // сюда считаем значение параметра
            if ((start[0]>='1') && (start[0]<='5')){ // считываем номер
пальца
              number=start[0]-'0'-1;
              if ((start[1]=='x') && (start[2]=='=')){ //если второй смвол 'x',
то это максимум
                type=1; // тип данных максимум
                start+=2; //значит обработали два символа
              } else if (start[1]=='='){ //если второй смвол '=', то это
положение пальца
```

```
                type=2; // тип данных текущее значение
                start+=1; //значит обработали один символ
            } else if ((start[1]=='n') && (start[2]=='=')) {//если второй
смвол 'n', то это минимум
                type=3; // тип данных минимум
                start+=2; // обработали два символа
            } else { // значит ошибочные данные
                break; // сбрасываем анализ
            }
            if (type>0){ // если тип был считан верно
                if (start[0]=='='){//считываем значение параметра
                  start++; //указатель на символ
                  if ((start[0]>='0') && (start[0]<='9') ){ //проверим что
цифра
                    value=start[0]-'0';
                    start++;
                    if ((start[0]>='0') && (start[0]<='9')) { //проверим что
цифра
                      value*=10;
                      value+=start[0]-'0';
```

```
                start++;
                if ((start[0]>='0') && (start[0]<='9')) { //проверим что
цифра
                  value*=10;
                  value+=start[0]-'0';
                 }
               }
              }
             } else { // ошибка данных
              break;
             }
             if ((value<=180) && (type>0)){ // параметр не может быть
больше 180 градусов
              if (type==1){ // если считывали максимум
               f_max[number]=value; // обновить данные максимума
               Serial.print(number);
               Serial.print(" MAX=");
               Serial.println(f_max[number]);
               EEPROM.write(number,f_max[number]); // сохранить
параметр в епроме
```

```
                setFingerPos(number,fval[number]); // установить палец
в соотвествии с новыми значениями

               } else if (type==2){ // положение пальца в процентах

                if (value>100){ // проценты не должны быть более 100

                  break;

                }

                fval[number]=value;  // обновить данные текущего
положения пальца

                Serial.print(number);

                Serial.print(" Value=");

                Serial.println(fval[number]);

                // EEPROM.write(number+sizeof(f_max),fval[number]);

                setFingerPos(number,fval[number]); // установить палец
в соотвествии с новыми значениями

               } else if (type==3){

                f_min[number]=value;// обновить минимум

                Serial.print(number);

                Serial.print(" MIN=");

                Serial.println(value);
```

```
EEPROM.write(number+sizeof(f_max)+sizeof(fval),f_min[number]);// сохранить параметр в епроме

                    setFingerPos(number,fval[number]); // установить палец в соотвествии с новыми значениями

                  }

                } else { // если параметр ошибочный, далее не анализируем

                  break;

                }

              }

            }

            start++;

            if ((start[0]=='\n')|| (start[0]=='\r')|| (start[0]==0)){ // если строка закончилась

              break; // больше параметры не ищем

            }

          }

          EEPROM.commit();//сохранить изменения епрома

        }

        currentLineIsBlank = true; // начинаем новую строчку:
```

```
          memset(linebuf,0,sizeof(linebuf)); // очищаем буфер приема
          charcount=0;
        } else if (c != '\r') { // в строке попался новый символ:
          currentLineIsBlank = false;
        }
      }
    }
    // даем веб-браузеру время, чтобы получить данные:
    delay(1);
    // закрываем соединение:
    client.stop();
    Serial.println("client disconnected");  //  "клиент отключен"
  }
}
```

Секция VI. Биомедицинская техника

Разработка прототипа датчика для анализа движения опорно-двигательного аппарата

Осипкова Мария Викторовна
г. Москва, ГБОУ №1525, 11 класс
Научный руководитель: Аполлонова Ирина Анатольевна
К.т.н., доцент, заместитель декана, кафедра БМТ-1, МГТУ им. Н.Э. Баумана

Целью проекта «Разработка прототипа датчика для анализа движения опорнодвигательного аппарата» является создание медико-технической системы сбора и обработки информации для исследования кинематики движения нижних конечностей, позвоночного столба и головы.

Разрабатываемая медико-техническая система основана на **сквозных технологиях** «интернет вещей» и «обработка и анализ больших массивов данных, искусственный интеллект» [3].

Необходимость накапливать мощный технологической потенциал будущего и развивать сквозные технологии отмечена в послании Президента Федеральному Собранию 01.12.2016 года одновременно с задачей «запустить масштабную системную

программу развития экономики нового технологического поколения, так называемой цифровой экономики» [4].

Использованный научный метод исследования движений основан на классических трудах по биомеханике выдающегося советского психо-нейрофизиолога, основоположника физиологии активности – Николая Александровича Бернштейна [1, 2], и непосредственно связан с развитием и внедрением цифровых технологий телемедицины. Исследование сочетает научные методы «наблюдение» и «эксперимент».

В ходе работы проведен анализ существующих и перспективных устройств и методов исследования двигательной активности; разработана медико-техническая система сбора и обработки информации о движении опорно-двигательного аппарата; разработана конструкция крепления датчиков; изготовлен

экспериментальный прототип автономного носимого устройства для сбора данных; разработана программа сбора данных; проведено экспериментальное исследование особенностей движения; проанализированы полученные результаты и предложены способы использования разработанной системы для диагностики заболеваний и восстановления двигательных функций.

Содержание

1 Введение

2 Перечень используемых сокращений

3 Основная часть

3.1 Анализ медико-биологической проблемы

3.1.1 Сведения об опорно-двигательном аппарате человека

3.1.2 Методы исследования двигательной активности

3.2 Анализ уровня техники

3.2.1 Устройства с датчиками движения

3.2.2 Устройства с услугой экстренной помощи

3.2.3 Устройства крепления различных датчиков на тело

3.2.4 Устройства с маркерной системой видеозахвата

3.2.5 Другие устройства для исследования движения человека

3.3 Экспериментальный прототип медико-технической системы

3.3.1 Принцип действия

3.3.2 Выбор комплектующих

3.3.3 Ориентация осей

3.3.4 Крепление датчиков

3.3.5 Места крепления датчиков

3.3.6 Схема соединений

3.3.7 Программа для сбора экспериментальных данных

3.3.8 Подготовка к сбору данных

3.3.9 Способ сбора экспериментальных данных

3.4 Результаты экспериментов

3.4.1 Условия проведения эксперимента

3.4.2 Планирование экспериментов

3.4.3 Упражнения, проделанные в ходе экспериментов

3.4.4 Анализ результатов эксперимента

3.5 Варианты использования

3.6 Выводы

4 Заключение

4.1 Дальнейшие шаги

5 Список использованных источников

6 Приложения

ПРИЛОЖЕНИЕ 1 Код модификации прошивки полетного контроллера

ПРИЛОЖЕНИЕ 2 Программа агрегирования данных датчиков для OrangePi PC+ 35

ПРИЛОЖЕНИЕ 3 Схема соединений медико-технической системы мониторинга движений

ПРИЛОЖЕНИЕ 4 Гониограммы движений

1 ВВЕДЕНИЕ

Основной целью работы является создание системы датчиков для анализа движения опорно-двигательного аппарата человека.

Задачи:

- Изучить методы исследования движения человека

- Изучить опорно-двигательный аппарат человека,
- Изучить основные заболевания опорно-двигательного аппарата
- Провести анализ уровня техники
- Разработать медико-техническую систему для осуществления поставленной цели
- Разработать и изготовить экспериментальный прототип автономного носимого устройства исследования движения:
    - Выбрать места крепления и определить количество датчиков; о Выбрать датчики; о Разработать схему соединения датчиков и сбора данных; о Изготовить прототип;
    - Изготовить устройства крепления датчиков к телу человека; о Разработать программу для сбора данных.
- Провести апробацию прототипа и собрать экспериментальные данные;
- Проанализировать собранные данные;

- Предложить способы использования разработанной системы для диагностики заболеваний и восстановления двигательных функций человека.

2 ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

ГНСС – глобальная навигационная спутниковая система;

ГСК - географическая система координат;

Гр – грудной отдел позвоночника;

Гл – голова;

ДЦП – детский церебральный паралич;

ИИМ – инерциальный измерительный модуль;

ЛБ – левое бедро;

ЛГ – левая голень;

ЛС – левая стопа;

ОДА – опорно-двигательный аппарат

ПБ – правое бедро;

ПГ – правая голень;

ПС – правая стопа;

ССК - связанная система координат;

Тз – таз;

ЧСС – частота сердечных сокращений;

Ш – шейный отдел позвоночника;

3 ОСНОВНАЯ ЧАСТЬ

Данная работа основана на трех научных методах: «эксперимент», «наблюдение» и «дедукция».

3.1 Анализ медико-биологической проблемы

3.1.1 Сведения об опорно-двигательном аппарате человека

Опорно-двигательный аппарат – совокупность костей скелета, суставов и мышц, позволяющая осуществлять движения человека в последствии нервной регуляции. [5]

3.1.1.1 Теория построения движений Бернштейна

Теория построения движений Н.А.Бернштейна [1,2,37] была впервые озвучена в начале XX века. Основные положения данной теории: о единстве онто и филогенеза движений; о ступенчатом развитии движений, об уровнях построения движений в центральной

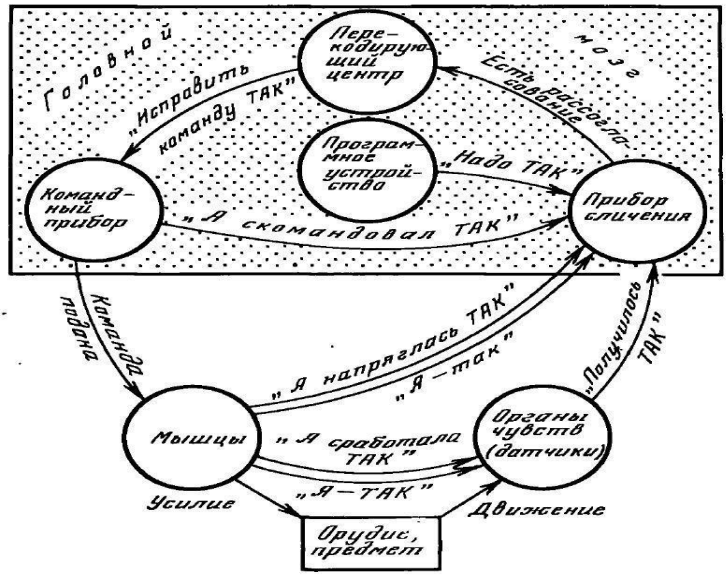


Рисунок 1 Блок-схема рефлекторного кольца типовой управляемой подсистемы

нервной системе; о рефлекторном кольце и сенсорных коррекциях; о двух циклах взаимодействия; о развитии навыка.

Рецепторы вестибулярного аппарата призваны сигнализировать о положении частей тела в пространстве и

скорости и направлении движений. *двигательного аппарата [37]* Шагательная синергия таламо-

паллидарного уровня координации, включающая почти всю скелетную мускулатуру, служит фундаментом организации двуногой ходьбы.

Развитие дифференцировки в пространстве, связано с созреванием полосатого тела (стриатума) – высокодифференцированного образования головного мозга, играющего роль регулятора и тормоза грубой рефлекторной деятельности бледного шара (паллидума).

Периферический двигательный аппарат осуществляет свою деятельность путем взаимодействия с внешней средой. Здесь имеется сложная циклическая форма взаимодействия: мышечные напряжения влияют на протекание движения, а движения влияют на мышечное напряжение (Рисунок 1).

Для того, чтобы правильно решить двигательную задачу, необходимо в течение всего двигательного акта выверять его с помощью органов чувств, в случае необходимости вносить необходимые поправки. Автоматизация движения – это процесс переключения технических компонентов управления движением в

низовые, фоновые, уровни – автоматизация движения [37]. Сущность процесса автоматизации, требующего иногда длительного времени и настойчивого упражнения, состоит именно в определении состава движении.

Первоначально для удержания стойкой длины шага ребёнок использует проприоцептивные механизмы. Аналогично, в начале освоения движения новичок напрягает все мышцы-антагонисты, заранее и с запасом выводит из строя все степени свободы, оставляя одну-две, необходимые для данной базы движений.

На следующей ступени упражнения, когда человек освоится, когда и с какой силой последует толчок реактивной силы, появляется возможность позволить себе постепенно высвобождать фиксированные до того степени свободы, чтобы экономить энергию в результате борьбы с реактивными силами.

Третья ступень развития движения наступает, когда борьба с реактивными силами превращается в полезные силы.

По Бернштейну, существует три ступени возрастной инволюции ходьбы:

1. Снижение функции структурных механизмов ходьбы с увеличением контроля сознания;

2. Настороженная сознательность сменяется суетливостью, мелкими шаговыми движениями;
3. Явный распад двигательных структур.

3.1.1.2 Основные выводы из теории построения движений Бернштейна

1. Уровень координации движений морфологически строго обозначен и включает в себя определенные структуры головного мозга;
2. Качества движений не наследуются, а приобретаются;
3. В онтогенетическом развитии природа использует все приобретенные ранее механизмы координации от примитивных до высших социальных действий;
4. Основой координации являются механизм сенсорной коррекции, два цикла взаимодействия и механизм развития навыка;
5. Ребенок в норме рождается «таламо-паллидарным». В течении первых двух лет он проходит две стадии прелокомоторного периода развития координаторных механизмов.
6. Прелокомоторный период заканчивает свое развитие ко второму году жизни в норме. Другими словами, в 2 года у ребенка есть все элементы, необходимые для удержания устойчивой вертикальной

позы и ходьбы. По Бернштейну, эта способность окончательно формируется к 5 годам.

3.1.1.3 Заболевания ОДА

3.1.1.3.1 Сколиоз

Сколиоз – трехплоскостная деформация позвоночника человека. Может быть, как врожденным, так и приобретенным заболеванием. Существует четыре различных стадии сколиоза: I – отклонение на 1-10 градусов, II – отклонения на 11-

25 градусов, III – отклонения на 26-50 градусов, IV степень – более 50. [6]

3.1.1.3.2 Плоскостопие

Плоскостопие - изменение формы стопы. Бывает двух видов – продольное и поперечное. При I степени деформации угол между первой и второй плюсневыми костями составляет 10—12 градусов, а угол отклонения первого пальца — 15—20 градусов; при II степени эти углы соответственно увеличиваются до 15 и 30 градусов; при III степени — до 20 и 40 градусов; при IV степени — превышают 20 и 40 градусов. [7]

3.1.1.3.3 Остеоартроз

Остеоартроз – группа заболеваний, при котором поражается хрящевая ткань суставов. В среднем от остеоартроза страдает около 7% населения развитых стран.

В настоящее время используется рентгенологическая классификация:

Сомнительный — сомнительное сужение суставной щели, возможны остеофиты;

Мягкий — определённые остеофиты и сомнительное сужение суставной щели; Умеренный — умеренные остеофиты, определённое сужение суставного пространства, возможная деформация костей; Тяжёлый — большие остеофиты, сужение суставного пространства, тяжёлый остеосклероз, определённая деформация костей. [8]

3.1.1.3.4 Переломы костей

Переломы костей – нарушение целостности кости в результате травмы или нагрузки, превышающую прочность травмированного участка. [9]

3.1.1.3.5 Судорога (спазм)

Судорога – непроизвольное сокращение мышц. Спазмы бывают двух типов:

поперечнополосатых и продольных мышц. [10]

3.1.1.3.6 Детский церебральный паралич (ДЦП)

Детский церебральный паралич - группа хронических непрогрессирующих заболеваний двигательных нарушений [37]. Распространённость детского церебрального паралича среди новорождённых: 2 к 1000 живорождённых (у мальчиков чаще в 1,33 раза) [38].

У ребенка с осложненным рождением, у которого будет диагностирован детский церебральный паралич, дефектен уже самый древний механизм координации движения. Родовые травмы – периферические параличи рук, переломы ключиц, гематомы и другие осложнения можно было рассматривать не как причину, а как следствие нарушенной программы движений плода [37].

Такие особенности позы и ходьбы при детском церебральном параличе, как нестабильность голеностопного сустава в саггитальной плоскости, значительное ослабление заднего толчка при ходьбе, поза тройного сгибания при стоянии, вызваны неполноценным контролем пирамидального уровня координаций с последующим нарушении работы тибиальной синкинезии. Тибиальная синкинезия Штрюмпела клинически трактуется как автоматическое тыльное сгибание и супинация стопы с одновременным подошвенным сгибанием первого пальца этой стопы. Данная синкинезия является элементом локомоции здорового человека, служит для переноса стопы над опорой.

Для улучшения состояния больных ДЦП применяются специальные костюмы. Важным фактором является влияние ношения костюма на движения. С помощью продольных эластичных тяг, идущих от надплечий к поясу и от пояса к стопам, усиливается сенсорная импульсация в координационные структуры мозга, ответственные за регуляцию антигравитационных функций. При применении костюма наблюдается повышение устойчивости позы и ходьбы детей с ДЦП.

Из теории Бернштейна следует:

1. Онтогенез повторяет филогенез, а значит, у каждого человека индивидуальная схема построения движений. Следовательно, можно применять общие законы как для здоровых людей, так и с людьми с различными нарушениями в двигательной сфере, в том числе и ДЦП;
2. Сравнительный анализ качественных характеристик движений в процессе их онтогенеза в норме и клинических симтомокомплексов расстройств позы и ходьбы при ДЦП позволяет провести отчетливые параллели. Исходя из этого, ДЦП это не болезнь с резидуальной стадией, а результат созревания изначально дефектного мозга, проявляющий себя уже внутриутробно;

3. У детей с ДЦП пирамидальный уровень не достигает своей полноценной зрелости;
4. У детей с ДЦП происходит задержка в развитии двигательных функций в 2-3 года. Из шестого пункта выводов из теории построения движений Бернштейна следует что различное хирургическое вмешательство ранее шести-семи лет бесполезно.

3.1.2 Методы исследования двигательной активности

Электромиография — регистрация биоэлектрических зарядов, возникающих в мышцах при напряжении. Исследование проводится при помощи электромиографа или электронейромиографа. Несмотря на простоту и широкий спектр заболеваний, которые можно выявить данным способом, существует огромное количество факторов, которые приводят к серьезным ошибкам в диагностике.

Кинезиология – наука о механике мышечного движений человека. В Советском Союзе, а затем и в России это направление называлось «биомеханика» [11]. Теоретическую основу кинезиологии составляет концепция «физиологии активности» советского психофизиолога и физиолога Н. А. Бернштейна [2].

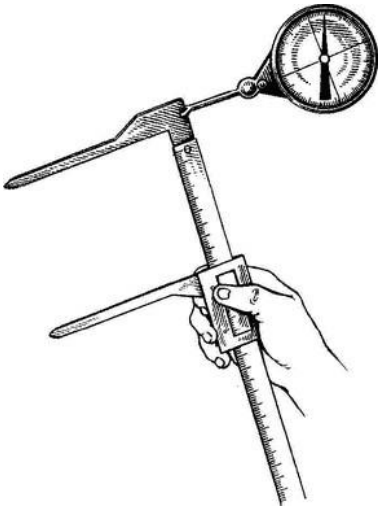
Актография – исследование двигательной активности во времени [12]. Наиболее широко применяется исследование двигательной активности во время сна. При анализе учитывается

продолжительность засыпания, длительность состояния полного покоя, общее время сна.

3.1.2.1 Методы исследования угловых перемещений

Гониометрия (угломерия) – единая методика определения угловой подвижности суставов и сравнения ее со стандартной для данного пола, возраста и других признаков [13].

В 1894 г. А.Н.Крылов предложил для изучения позы школьников за партами и искривлений позвоночника у детей прибор — школьно-гигиенический гониометр, состоящий из двух брусочков и транспортира с отвесом.



Для проведения различных угломерных исследований В.А.Гамбурцев в 1934 г. предложил новый прибор – скользящий циркуль-гониометр, который позволял производить измерения кривизны позвоночника, углов наклона таза, амплитуд движения суставов конечностей и позвоночного столба. Методики

гониометрических исследований описаны в [14]. Для измерения амплитуд ротации - углов поворота

кисти и стопы (пронации и супинации) применяют прибор «ротаметр».

Ангулография — запись углов сгибания и разгибания в суставах конечности с обозначением межзвенных углов [15].

Циклография - метод изучения движений человека посредством фотографирования светящихся точек на движущихся частях тела. Н. А. Бернштейн и его коллеги в качестве основного метода регистрации циклических движений использовали циклографию [2, 16].

Циклограммы позволяют видеть целостное пространственное движение тела, образующееся в результате сложения угловых движений множества звеньев тела относительно друг друга.

Ихнометрия — измерение пространственных характеристик шага, таких как длина шага, угол поворота и т.д. [17].

Основным методом получения измерений является ихнография — метод записи следов от обеих ног при ходьбе с учетом длины шага каждой ноги, разворота стопы, ширины шага. Пример - анализ ходьбы и бега по положениям отпечатков ног [15].

3.1.2.2 Методы исследования ходьбы

Подография – метод регистрации длительности отдельных периодов шага на основе исследования распределения давления на подошве [18].

Подометрия — измерение временных характеристик шага, таких как длительность цикла, интервалы опоры на пятку и носок, коэффициент ритмичности и другие [19].

3.1.2.3 Методы исследования позы тела

Постурография – совокупность методов оценки способности человека управлять позой тела.

Треморография - метод исследования дрожания тела. Запись колебаний частей тела производится при помощи акселерометра на ЭКГ аппарате [20].

Стабилография – метод диагностики нарушений функции опорнодвигательной, центральной и периферической нервной системы на основе стабилометрии — регистрации положения и движений проекции общего центра давления на плоскость опоры при стоянии. Для исследования используются координационные пробы:

- Вращение в кресле Барани;
- Проба Ромберга - постепенное уменьшение площади опоры. Фиксируются пошатывания и дрожание рук или век (тремор);
- Тест Яроцкого – определение порога чувствительности вестибулярного анализатора. Определение времени от начала вращательных движений головой с закрытыми глазами в позе стоя до потери равновесия. Норма здорового человека – 28 с, тренированного – до 90 с;
- Оптокинетический тест (выявление изменения функции равновесия, связанные с влиянием оптокинетического нистагма, вызванного движением по экрану черных и белых полос) [21].

-Для того, чтобы

3.1.2.4 Системы видеозахвата движений

Захват движения (Motion Capture) – это процесс регистрации движений человека с последующим использованием полученных данных для анимирования образа или персонажа, созданного ЭВМ [36].

Два основных вида систем захвата движения:

1. Маркерная система. На человека надевают специальные костюмы, данные с датчиков и с камер поступают на ЭВМ. Компьютер создает трехмерную модель изображения на

основе поступивших данных. На основе полученной модели создается анимация персонажа. Подобным образом можно создать и мимику персонажа.

2. Безмаркерная технология не требует специального оборудования. Представляет собой комплекс из высокоскоростных камер и специального программного обеспечения. Позволяет снимать сложные движения.

3.2 Анализ уровня техники

3.2.1 Устройства с датчиками движения

Умная трость разработки Mediq-Care (The Smart Cane from Mediq-Care) представляет собой трость, в которую встроено устройство для определения движения и ориентации в пространстве. Ведется постоянный мониторинг ГНСС, а через рукоятку ведется фиксация ЧСС, температура человека и сила давления на трость. В случае, если трость находится в горизонтальном положении дольше допустимого времени, то она автономно отправляем сигнал тревоги врачу и близким родственникам [22].

Умная трость, разработанная Patrick Aubin, снабжена вибромотором для организации обратной связи [23].

Умная скакалка «Smart Rope» рекомендует график тренировок и подсчитывает потраченные калории при помощи

анализа высоты и количества прыжков. Внутри веревки протянута LED-лента, обеспечивающая «вау-эффект». В ручках размещены акселерометры с устройствами беспроводной связи для передачи данных в приложение смартфона [24].

Baby Check – одеваемый на плечо прибор приспособлен специально для младенцев и позволяет родителям следить за температурой и движением. В случае обнаружения каких-то отклонений от нормы, устройство подает сигнал тревоги по Bluetooth на смартфон родителя [25].

Senior Lifestyles предлагает устройство, которое при помощи технологий «умный дом» наблюдает за образом жизни пациента. При выявлении отклонений поведения от зарегистрированных шаблонов, устройство подает сигнал по беспроводной связи на смартфон или в социальный аккаунт близких или родственников [26].

3.2.2 Устройства с услугой экстренной помощи

Philips Lifeline AutoAlert - размещаемое на шее водозащищенное устройство с кнопкой, на которую необходимо нажать, чтобы попросить о помощи. При детектировании падения устройство делает автоматический вызов даже тогда, когда пациент не в состоянии нажать на кнопку. При подъеме после падения менее чем через 30 секунд тревожный сигнал не отправляется. Для передачи

вызова используется промежуточное устройство Lifeline Communicator [27].

SAMi 3 - с помощью инфракрасной камеры следит за движениями человека во время сна и фиксирует то, что происходит в комнате. Использует специальный алгоритм для фильтрации обычных движений и эпилептического приступа [28].

3.2.3 Устройства крепления различных датчиков на тело

Футболка Тришей Эндрю - на футболке или любой другой одежде зафиксированы миниатюрные суперконденсаторы с помощью электропроводных нитей с вакуумным напылением, «обернутых» в полимерную пленку. В ближайшее время ученые хотят «вышить» шагомер, сенсор для наблюдения за работой, а также датчик для получения данных о работе суставов. [29]

3.2.4 Устройства с маркерной системой видеозахвата

32 Neuron Edition V2 – набор датчиков для создания 3D модели анимированного персонажа с помощью техники анимации захвата движения.

Средняя стоимость подобных датчиков – 1500-4000$. [30]

Костюм Perception Neuron 32 – система из 32 датчиков, позволяющая полностью переноси свой аватар в виртуальную реальность. Максимальная задержка между движением и отображение составляем 17мс. В каждом датчике есть акселерометр, гироскоп и

магнетометр. Perception Neuron работает с Windows и IOS, был выпущен для Unity, Unreal и Oculus VR. Данные передаются по Wi-Fi или через USB в случае необходимости подзарядки. [31]

PrioVR – полноценный костюм виртуальной реальности, состоящий из 17 датчиков. Позволяет тренировать мышцы в ходе игры. Продается сразу в комплекте с играми. В каждый датчик встроен гироскоп и акселерометр. [32]

Senso Suit – костюм для отслеживания движений, состоящий из 15 модулей. Каждый модуль представляет собой коробочку размером 2,5х2,5см. В модули встроены датчики IMU и вибромоторы. Данные передаются через Wi-fi, есть возможность USB соединения. Костюм не требует дополнительных камер для отслеживания движений. Основная технология отслеживания движений – SteamVR/Lighthouse. [33]

3.2.5 Другие устройства для исследования движения человека

Стабилометрическая платформа AMTI AccuGaitACG – портативная дорожка для измерения биометрических параметров походки. Встроены датчики Холла для наиболее точного измерения прилагаемой силы на платформу. Встроена защита от перегрузок на всех осях. Информация передается по USB, RS-232 или через коммутатор. [34]

Умная школьная форма – разработка китайский ученых, на каждом плече ребенка расположены два чипа, позволяющих идентифицировать его и проследить за его местоположением. В случае попытки покинуть школу в учебное время издает сигнал. Также чипы позволяют определить, не заснул ли ученик на уроке.

Выдерживает температуру до 150 градусов Цельсия и до 500 стирок. [35]

IOTA Biosciences – сенсор, вживляемый в организм человека, работающий на ультразвуковом излечении. Производители считают, что для передачи сигнала ультразвук подходит идеально: длина волны в данном случае измеряется в миллионных долях метра, а значит его можно захватить — и захватить очень эффективно. Прибор способен работать удаленно.

3.3 Экспериментальный прототип медико-технической системы

В данном разделе описан процесс разработки медико-технической системы для сбора информации о двигательной активности человека.

3.3.1 Принцип действия

Экспериментальный прототип собран из десяти датчиков углов наклона и ускорений на базе полетных контроллеров, объединенных в сеть сбора и передачи данных с Raspbery-Pi подобным микрокомпьютером «OrangePi-PC+». Автономное электропитание

нательной системы сбора данных осуществляется от аккумулятора («Банк Энергии» / «Power Bank»), исходно предназначенного для подзаряда смартфонов. Банк энергии имеет ёмкость 10 Ампер-часов, обеспечивает выходное напряжение 5 Вольт при токе нагрузки до 2 Ампер. Для обеспечения вывода и передачи данных произведена модификация программного обеспечения полетного контроллера (ПРИЛОЖЕНИЕ 1). Штатная функция полетного контроллера, основанная на алгоритме Махони производит вычисления при помощи кватернионов, выдает текущую ориентацию ССК относительно ГСК в виде трех углов Эйлера – курс (угол рыскания), крен, тангаж. Углы Эйлера вычисляем 120 раз в секунду.

Калибровка гироскопов производится в момент каждого запуска датчика.

Поток данных от датчиков передается по последовательному (UART 3.3В) порту со скоростью 500000 бит в секунду, данная скорость выбрана для обеспечения совместимости с микрокомпьютером OrangePi PC+.

Последовательные порты датчиков соединены друг с другом, так, что каждый из промежуточных датчиков работает в качестве ретрансляторов сигналов от предыдущего к последующему датчику. Каждый из «хвостовых» датчиков (их всего три) подключен к агрегатору сбора данных – микрокомпьютеру «OrangePi PC+», который записывает поступающую информацию на твердотельный

носитель “microSD”, а также через модуль Wi-fi передает информацию на управляющий компьютер.

3.3.2 Выбор комплектующих

В качестве основы для создания датчика угла наклона и ускорения выбран недорогой полетный контроллер, совместимый с «NAZE32». Десять таких устройств двух типов были приобретены на двух распродажах в интернетмагазине. Одно из основных назначений полетного контроллера – измерение ориентации и перемещения в пространстве. Для этого в авиации используется прибор «курсовертикаль», измеряющий углы между географической системой координат (ГСК) и связанной системой координат (ССК), начальная точка которой совпадает с начальной точкой ГСК, а оси соответствуют ориентации прибора.

Полетный контроллер содержит микропроцессор типа STM32F103, акселерометр и гироскоп, встроенные в микросхему MPU-6050 и датчик давления воздуха, способный определять изменение высоты с шагом в 0,1 метра.

3.3.3 Ориентация осей

Изначально полетный контроллер предназначен для установки в горизонтальной плоскости: направление оси X обозначено стрелкой,

нарисованной на плате, ось Y находится в плоскости печатной платы, на 90 градусов левее от оси X, а ось Z направлена вверх.

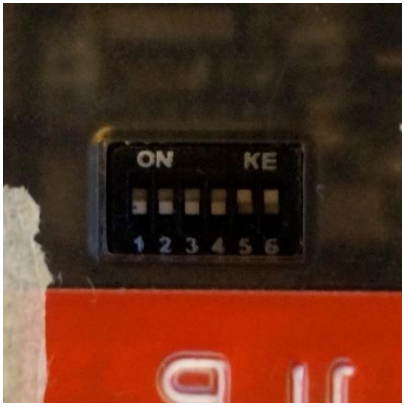


Рисунок 2 Положение ключей

Так как большинство датчиков удобнее располагать в вертикальной плоскости, поэтому требуется настройка изменения ориентации осей. Требуемое направление осей устанавливается при помощи интерактивной программы Betaflight Configurator (программа для настройки различных полетных контроллеров на основе прошивки Betaflight с открытым исходным кодом). Это бесплатное приложение, его можно установить, как дополнение к браузеру Google Chrome [39].

Для обеспечения возможности настройки параметров полетного контролера необходимо переключить ключи соединения USB порта согласно схеме: ключи 1, 2, 3, 4 выключены, ключи 5, 6 включены (Рисунок 2).

В Таблице 1 приведены сведения об изменении ориентации датчиков в соответствии с их расположением. Так, например, правая

ступня в нулевой момент времени расположена примерно параллельно поверхности земли, при чем стрелка ориентации направлена влево, поэтому для корректной работы угол рыскания повернут на 90 градусов по часовой стрелке.

Таблица 1 - Настройка ориентации осей датчиков

Название датчика	Крен	Тангаж	Рыскание
Левое бедро	0	90	180
Левая голень	0	90	180
Правое бедро	0	-90	0
Правая голень	0	-90	0
Таз	0	-90	0
Грудь	0	-90	0
Шея	0	-90	0
Правая ступня	0	0	90
Левая ступня	0	0	-90
Голова	0	0	90

3.3.4 Крепление датчиков

Для крепления датчиков используются различные устройства на основе эластичных материалов, исходно предназначенные для

фиксации суставов или коррекции осанки. Все датчики прикрепляются к устройствам крепления при помощи текстильной застежки-«репейника» «Velcro».

Устройства для крепления датчиков разработаны и изготовлены с учетом требования по возможности быстрого одевания, снятия, перемещения.

3.3.5 Места крепления датчиков

В результате исследования опорно-двигательного аппарата были выявлены основные звенья, имеющие собственные степени свободы. [40]

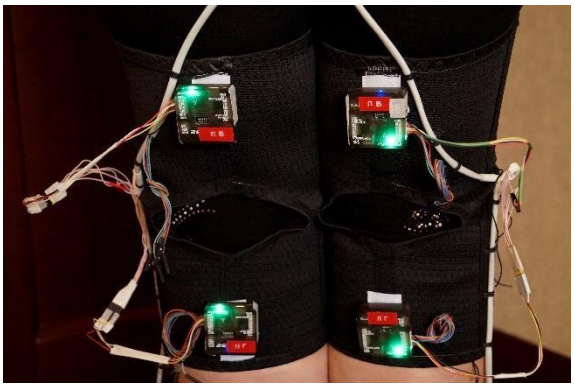
Было принято допущение о том, что стопа, большеберцовая и бедренная кости являются абсолютно твердыми телами, а значит, для исследования двигательной активности на каждой ноге достаточно разместить 3 датчика. Для жесткой фиксации было принято решение о креплении датчиков в местах с наименьшей толщиной мышечного слоя и кожно-жирового покроя. Для этого положения датчиков максимально приближены к кости. Для сокращения зависимости от электромагнитных помех и повышения качества передачи данных соединения датчиков выполнены при помощи экранированного двупарного провода «витая пара» категории 5е. Длина соединительных проводов выбрана таким образом, чтобы не стеснять

свободу движений человека. Прокладка проводов осуществлена так, чтобы минимизировать провисание провода в крайних положениях.

Для голеностопного сустава изготовлено устройство крепления из эластичного бинта (Рисунок 3. Рисунок 3 - Крепление датчика на стопе). Для датчика большеберцовой и бедренной костей приобретены в магазине коленные ортезы (Рисунок 4). Датчики прикреплены прямо на ортез с помощью Velcro выше и ниже колена соответственно.

Рисунок 3 - Крепление датчика на стопе.

Для исследования кинематики движений позвоночника применены три датчика, закрепленные на спине: на уровне талии, на уровне груди и на шее. Такое расположение выбрано по результатам исследования гибкости позвоночника (Рисунок 6).



Еще один датчик расположен на голове,

Рисунок 4 - Крепление датчиков на зафиксирован при помощи

специально сшитой *бедре и голени* шапочки (Рисунок 5)

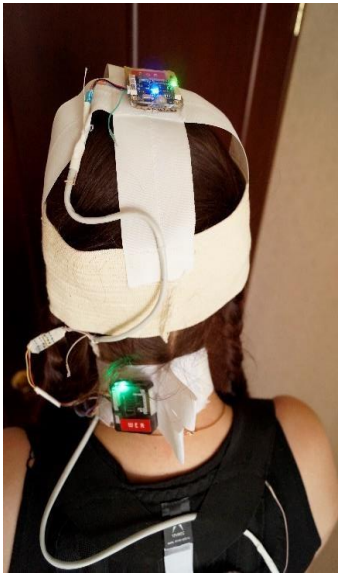


Рисунок 5 - Крепление датчиков на голове и шее

«OrangePi PC+» и аккумулятор расположены вместе с грудным датчиком на спине, прикреплены к корректору осанки (Рисунок 6).

3.3.6 Схема соединений

Датчики, аккумулятор и микрокомпьютер соединены в цепь, согласно схеме, приведенной в Приложении 3.

Датчик ЛС, размещенный на левой ступне, определяет ориентацию и вектор ускорения и передает эти данные совместно с информацией о времени и своем

названии по последовательному интерфейсу. Датчик ЛГ, размещенный на левой голени выполняет роль ретранслятора данных от датчика ЛС и добавляет

информацию о собственной ориентации и ускорении. Далее датчики ЛБ и Тз, размещенные на левом бедре и на пояснице соответственно, выполняют аналогичные функции: передают информацию от предыдущих датчиков, добавляя свою информацию о собственной ориентации и ускорении. Далее информация поступает на порт ttyS1 агрегатора данных на базе микрокомпьютера «OrangePi PC+», записывается на твердотельный носитель или по беспроводной линии связи Wi-fi передается на удаленный компьютер.

Аналогично, датчик ПС, размещенный на правой ступне передает информацию датчику ПГ, размещенному на правой голени, который добавляет свою информацию о времени и названии и выполняет свою роль ретранслятора и передает информацию датчику ПБ, размещенному на правом бедре, который в свою очередь передает пришедшую информацию с добавлением своей на порту ttyS2 микрокомпьютера «OrangePi PC+».

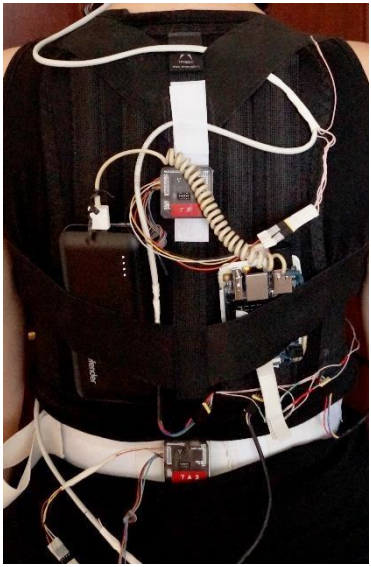


Рисунок 6- Крепление датчиков и устройств на спине

Датчик Гл, размещенный на голове, передает информацию об углах и ускорениях датчику Ш, размещенному на шее. Датчик Ш выполняет роль ретранслятора, добавляя свою информацию, и передает ее на порт ttyS3 микрокомпьютера «OrangePi PC+».

Передача данных осуществляется по последовательному интерфейсу UART с логическими уровнями 3.3В на скорости 500000 бит в секунду.

3.3.7 Программа для сбора экспериментальных данных

Программа агрегирования (сбора) экспериментальных данных выполняется на микрокомпьютере «OrangePi PC+», осуществляет чтение данных из последовательных портов ttyS1, ttyS2 и ttyS3, построчную сборку данных и публикацию данных с частотой 120 раз в секунду. Программа написана на языке С.

Исходный код программы приведен в Приложении 2.

3.3.8 Подготовка к сбору данных

Для удобного надевания разработана памятка-инструкция по надеванию системы датчиков:

1. Надевание осуществляется с помощью ассистента;
2. Надеть крепеж датчиков стопы;
3. Надеть корректор осанки, прикрепить аккумулятор, «OrangePi PC+» датчик груди;
4. Надеть крепеж для датчиков Ш и Гл, закрепить датчики, подсоединить к «OrangePi PC+»;
5. Надеть ортезы на колени, крепеж для датчика Тз;
6. Закрепить датчики на своих местах;
7. Подключить датчики к разъемам.

3.3.9 Способ сбора экспериментальных данных

1. Закрепить на теле датчики в соответствии с описанием в пункте 3.3.8;
2. Включить датчики, подождать 10-15 минут для термостабилизации датчиков необходимой для повышения стабильности показаний;
3. Встать в исходную позицию: руки по швам, ноги вместе, голова смотрит прямо, спина прямая;
4. Нажать на кнопку выключения питания микрокомпьютера «OrangePi PC+».

   Дождаться завершения работы микрокомпьютера и выключения зеленого светодиода;
5. Подождать 5 секунд после выключения зеленого светодиода;
6. Выключить и сразу включить разъем питания на аккумуляторной батарее;
7. Дождаться загрузки микрокомпьютера и загорания красного светодиода;
8. Совершить запланированные в эксперименте действия.

   Рекомендуется не использовать прототип дольше 3 минут;
9. После завершения эксперимента встать в исходную позу;

10. Перейти к следующему эксперименту, повторив действия пунктов 3-9;
11. По окончании цикла экспериментов перед выключением аккумулятора нажать кнопку на компьютере, подождать 5 секунд после выключения зеленого светодиода для полной остановки работы «OrangePi PC+», затем отключить аккумулятор.

3.4 Результаты экспериментов

3.4.1 Условия проведения эксперимента

Эксперименты проводились на одном человеке в помещении с искусственным освещением при температуре воздуха: +22С; атмосферном давлении 730 мм рт. ст., и относительной влажности воздуха 25%.

3.4.2 Планирование экспериментов

Для проверки работоспособности разработанной системы разработана совокупность упражнений, проведена серия экспериментов, осуществлена запись и анализ полученных данных. В результате были получены гониограммы движений, часть из которых приведена в Приложении 4.

3.4.3 Упражнения, проделанные в ходе экспериментов

1. Наклоны туловища до предельных положений вперед и назад;
2. Приседания;
3. Вращение туловищем вдоль вертикальной оси;
4. Сохранение положения равновесия при одной поднятой ноге;
5. Гимнастическое упражнение «Ласточка» (балансировка на одной ноге);
6. Ходьба на месте с высоким подниманием колена;
7. Обычная ходьба с перешагиванием через препятствие высотой 40 см;
8. Шаркающая ходьба;
9. Ходьба с правильной осанкой;
10. Ходьба вниз и вверх по лестнице;
11. Бег вниз и вверх по лестнице;

3.4.4 Анализ результатов эксперимента

В результате проведения серии экспериментов были получены файлы формата .csv, которые были преобразованы в графики

гониограмм при помощи электронной таблицы MS Excell - 2013 (Приложение 4).

На гониограмме «Ходьба вверх и вниз по лестнице» приведена зависимость от времени углов тангажа датчиков, размещенных на правой голени, левой голени, левом бедре и правом бедре. Обнаружено значимое различие двигательной активности с шестнадцатой по тридцать седьмую секунды (период 1 - спуск по четырем пролетам лестницы), с тридцать седьмой по пятьдесят восьмую секунды (период 2 - подъём по трем пролетам лестницы), с пятьдесят восьмой по семьдесят первую секунду (период 3 – подъем по пролету лестницы «приставным» шагом).

На гониограммах «Ходьба» и «Шаркающая ходьба» представлены зависимости от времени углов тангажа датчиков, размещенных на правой стопе, левой стопе, правой голени, левой голени, левом бедре, правом бедре и на тазу.

При сравнительном анализе этих гониограмм выявлены следующие различия:

1. Основное различие — темп ходьбы (число шагов в минуту): при обыкновенной ходьбе левая нога совершает 8 шагов за 9 секунд (53 шага в минуту), при шаркающей всего 6 за те же 9 секунд (40 шагов в минуту);

2. Амплитуда колебаний стоп: при обыкновенной ходьбе амплитуда равна 80 градусам, при шаркающей 60 градусов;
3. При обыкновенной ходьбе заметны удары при крайних положениях, приводящие к образованию двойных пиков, при шаркающей ходьбе двойные пики не зафиксированы.

Различие 3 подлежит дальнейшему исследованию для уточнения причины образования двойных пиков.

На гониограмме «Ласточка» представлены зависимости от времени углов тангажа датчиков, размещенных на левой стопе, правой стопе, левой голени, правой голени, левом бедре, правом бедре, тазе и голове. При выполнении этого упражнения человек стоит на правой ноге, левая отведена назад, корпус тела наклонен вперед. Графики левой голени и левого бедра практически совпадают так как угол в коленном суставе изменяется на малую величину (нога вытянута).

Высокочастотные колебания вызваны мышечным тремором в вытянутой ноге. Амплитуда угловых колебаний левой (находящейся в воздухе) ступни связана с нестабильностью удержания равновесия и может быть принята в качестве меры способности удерживать равновесие. Примечателен факт нестабильности по тангажу правой стопы, которая во время упражнения неподвижна относительно опорной поверхности: амплитуда угловых колебаний по составляет 9

градусов. Вероятнее всего, эти колебания вызваны мышечной активностью стопы в процессе удержания равновесия.

3.4.4.1 Вывод по итогам анализа результата эксперимента

В результате анализа полученных гониограмм подтверждена работоспособность предложенного метода и изготовленного прототипа медикотехнической системы.

3.5 Варианты использования

1. Исследования двигательной активности;
2. Диагностика и исследование патологий ОДА;
3. Восстановление двигательной активности;
4. Фитнесс, физкультура, спорт;
5. Системы дополненной реальности, индустрия игр и кинематограф.

3.6 Выводы

В данной работе проанализированы медико-биологическая проблема и уровень техники, предложен способ мониторинга двигательной активности, изготовлен прототип медико-технической системы исследования двигательной активности. По итогам работы сделаны следующие выводы:

1. Основным техническим решением для изучения кинематики движения опорно-двигательного аппарата является мониторинг ускорений и углов наклона конечностей;
2. На основании полученных экспериментальных данных можно сделать вывод об адекватной степени точности примененных датчиков MPU-6050 для распознавания различных положений тела и типов ходьбы;

4 ЗАКЛЮЧЕНИЕ

1. Проведено исследование медико-биологической проблемы. Изучена теория кинематики движения Николая Александровича Бернштейна, основоположника исследований опорно-двигательного аппарата (ОДА). Исследованы основные заболевания опорно-двигательной системы, анатомия и физиология ОДА. На основе анализа ОДА предложен способ крепления и обоснована схема месторасположения датчиков;
3. Проведен обзор и анализ методов исследования двигательной активности. На основе исследования сделан вывод о том, что перспективным техническим решением для изучения кинематики движения опорно-двигательного аппарата является мониторинг ускорений и углов наклона конечностей.

4. Проведен обзор и анализ устройств для исследования кинематики движения;
5. Разработана носимая медико-техническая система для мониторинга движения нижних конечностей, позвоночного столба и головы;
6. Выбраны и обоснованы места крепления датчиков;
7. Разработан способ крепления датчиков, изготовлены устройства крепления;
8. Изготовлен и собран экспериментальный прототип медико-технической системы для мониторинга и исследования движения человека;
9. Проведена серия экспериментов, проанализированы результаты и сделаны выводы по итогам экспериментов; экспериментальным путем подтверждена работоспособность прототипа и адекватность получаемых данных;
10. Предложены варианты использования.

4.1 Дальнейшие шаги

В дальнейшем планируется провести медицинские исследования различных патологий движений и ОДА, подобрать дизайнерские решения стильной одежды и модных аксессуаров с датчиками движения. Планируются дополнительные эксперименты

для проверки достоверности и доработки методов автоматического распознавания движений на большом количестве людей. Также планируется проведение исследований рынка, разработка бизнес-плана и подбор партнеров для стартапа.

5 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бернштейн Н. А. Биомеханика для инструкторов [Текст] : лекции на курсах инструкторов Промышленного Экономическ. Отд. Москпрофобра // Ник. Бернштейн. – М. : Новая Москва, 1926.
2. Бернштейн Н. А. Очерки о физиологии движений и физиологии активности. — М. : Медицина, 1966.
3. Пленарное заседание Петербургского международного экономического форума
   2 июня 2017 года [Электронный ресурс] // Интернет сайт Президента России.
   URL: http://kremlin.ru/events/president/news/54667/ (дата обращения: 12.01.2018).
4. Путин В.В. Послание Президента Федеральному Собранию 01.12.2016 г. [Электронный ресурс] // Интернет сайт Президента России. URL:
   http://kremlin.ru/events/president/news/53379 (дата обращения: 12.01.2018).
5. Опорно-двигательный аппарат [электронный ресурс] //Википедия https://ru.wikipedia.org/wiki/Опорно-двигательная_система (дата обращения:
   26.01.2019).

6. Сколиоз [электронный ресурс] //Википедия https://ru.wikipedia.org/wiki/Сколиоз (дата обращения: 26.01.2019).
7. Плоскостопие [электронный ресурс] //Википедия https://ru.wikipedia.org/wiki/Плоскостопие (дата обращения: 26.01.2019).
8. Остеоартроз [электронный ресурс] //Википедия https://ru.wikipedia.org/wiki/Остеоартроз (дата обращения: 26.01.2019).
9. Переломы костей [электронный ресурс] //Википедия https://ru.wikipedia.org/wiki/Перелом_кости (дата обращения: 26.01.2019).
10. Судорога [электронный ресурс] //Википедия https://ru.wikipedia.org/wiki/Спазм (дата обращения: 26.01.2019).
11. Кинезиология [Электронный ресурс] // Википедия — свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%BD%D0%B5%D0%B7%D0

    %B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F (дата обращения:

    05.01.2018).
12. Актография [Электронный ресурс] // Большой медицинский словарь. URL: https://dic.academic.ru/dic.nsf/medic2/934 (дата обращения: 12.01.2018).
13. Гониометрия [Электронный ресурс] // Википедия — свободная энциклопедия. URL:

https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%BE%D0%BC%D0 %B5%D1%82%D1%80%D0%B8%D1%8F (дата обращения: 05.01.2018).

14. Гамбурцев, В. А. Гониометрия человеческого тела / В.А. Гамбурцев. - М. : Медицина, 1973.
15. Дубровский В.И., Федорова В.Н. Биомеханика - М. : Владос-Пресс, 2003.
16. Циклография [Электронный ресурс] // Психологическая энциклопедия wiki.1vc0.ru. URL: http://wiki.1vc0.ru/enciklopediya/psixologiya/psyterms/ciklografiya.html (дата обращения: 05.01.2018).
17. Ихнометрия [Электронный ресурс] // Википедия — свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/%D0%98%D1%85%D0%BD%D0%BE%D0%BC%D0 %B5%D1%82%D1%80%D0%B8%D1%8F (дата обращения: 05.01.2018).
18. Подография [Электронный ресурс] // Открытая клиника. URL: https://www.dikul.org/diagnostycs/podografiya/ (дата обращения: 05.01.2018).
19. Подометрия [Электронный ресурс] // Википедия — свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%BE%D0%BC%D0 %B5%D1%82%D1%80%D0%B8%D1%8F (дата обращения: 05.01.2018).
20. Треморография [Электронный ресурс] // Интернет сайт «Sport2002.ru - Справочник тренера». URL: http://www.sport2002.ru/l/nervnomyshechn/Tremorografiya/ (дата обращения: 12.01.2018).

21. Слива С.С., Войнов И.Д., Переяслов Г.А., Слива А.С. Стабилографическая экспресс-оценка психофизиологического состояния человека. Методические рекомендации. – Таганрог : ЗАО «ОКБ «Ритм», 2011.
22. Comstock T. The Smart Cane from Mediq-Care [Электронный ресурс] // Kickstarter. URL: https://www.kickstarter.com/projects/634203661/the-smart-canefrom-mediq-care?ref=category (дата обращения: 12.01.2018).
23. Aubin P. Smart Cane [Электронный ресурс] // Aubin Patrick WordPress blog. URL: https://www.kickstarter.com/projects/634203661/the-smart-cane-from-mediqcare?ref=category (дата обращения: 05.01.2018).
24. Smart Rope [Электронный ресурс] // Kickstarter. URL: https://www.kickstarter.com/projects/1990698791/smart-rope?ref=category (дата обращения: 05.01.2018).
25. Baby Check - The Simplest Wearable Temp and Position Monitor [Электронный ресурс] // Kickstarter. URL: https://www.kickstarter.com/projects/1368111560/babycheck-the-simplest-wearable-temp-and-position?ref=category (дата обращения: 05.01.2018).
26. Senior Lifestyle System [Электронный ресурс] // Qorvo, Inc. URL: http://www.qorvo.com/applications/internet-of-things/lifestyle-systems (дата обращения: 05.01.2018).
27. Philips Lifeline AutoAlert Help Button Instructions for Use Model: FD100
[Электронный ресурс] // Philips lifeline. Medical alert service. URL:

https://www.lifeline.philips.com/content/dam/PLL/PLLB2C/PDFs/AutoAlertHelpButtonGuide_HomeSafeTechSupport.pdf (дата обращения: 10.01.2018).

28. Epilepsy Foundation - SAMi [Электронный ресурс] // Indiegogo, Inc. URL: https://www.indiegogo.com/projects/epilepsy-foundation-sami#/ (дата обращения:

05.01.2018).

29. Футболка Тришей Эндрю [Электронный ресурс] // URL: https://hi-news.ru/technology/tkanevaya-batareya-novoe-slovo-v-sozdanii-umnojodezhdy-i-nosimoj-elektroniki.html (дата обращения: 26.01.2019).

30. 32 Neuron Edition V2 [Электронный ресурс] // URL: https://neuronmocap.com/content/product/32-neuron-edition-v2-facerig-studio (дата обращения: 26.01.2019).

31. Костюм Perception Neuron 32 [Электронный ресурс] // URL: https://virtuality.club/store/kontrollery-virtualnoy-realnosti/kostjum-perception-

neuron?gclid=EAIaIQobChMIx9HB4qSg4AIVBOWaCh3J4gYbEAQYAyABEgK5z

PD_BwE (дата обращения: 26.01.2019).

https://www.youtube.com/watch?v=nP69c23vAcI (дата обращения: 09.02.2019).

32. PrioVR [Электронный ресурс] // URL: https://vr-magazine.ru/prio-vr (дата обращения: 26.01.2019).

33. Senso Suit [Электронный ресурс] // URL: https://vr-magazine.ru/senso-suit (дата обращения: 26.01.2019).

34. AMTI AccuGaitACG [Электронный ресурс] // URL: http://svga.ru/product_info.php?products_id=847 (дата обращения: 26.01.2019).

35. Умная школьная форма [Электронный ресурс] // URL: https://hi-news.ru/technology/odezhda-kotoraya-meshaet-progulivat-shkolu-i-spat-naurokax.html (дата обращения: 26.01.2019).

36. Александр Шведов, Системы видеозахвата движения за гранью индустрии развлечений, Санкт-Петербург, 2018 год. (дата обращения: 09.02.2019).

37. Перхурова И.С., Лузинович В.М., Сологубов Е.Г. Регуляция позы и ходьбы при детском церебральном параличе и накоторые способы коррекции. М. ; «Книжная палата», 1996.

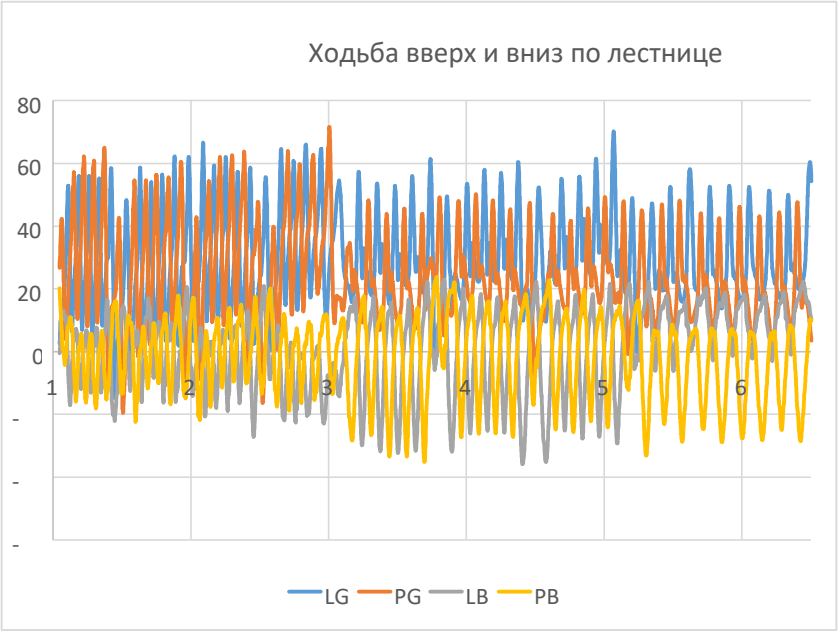
38. Детский Церебральный паралич [Электронный ресурс] // Википедия — свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%82%D1%81%D0%BA%D0%

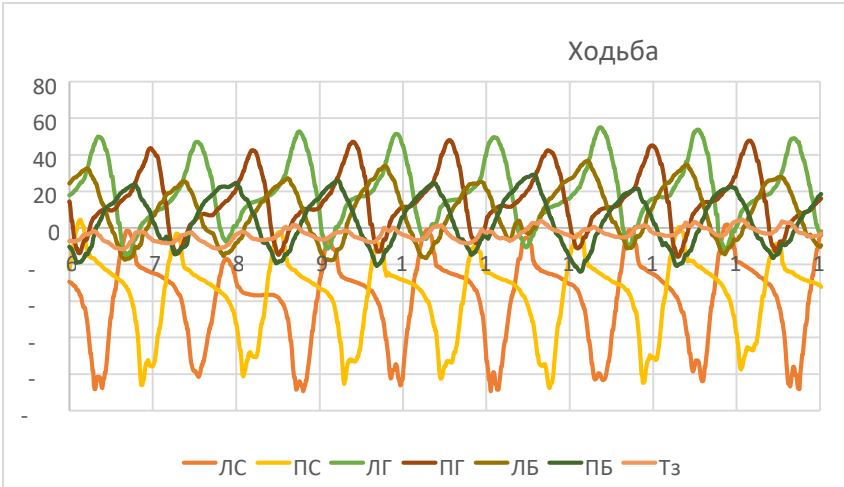
B8%D0%B9_%D1%86%D0%B5%D1%80%D0%B5%D0%B1%D1%80%D0%B0%
D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D0%B0%D1%80%D0
%B0%D0%BB%D0%B8%D1%87 (дата обращения 09.02.2019)

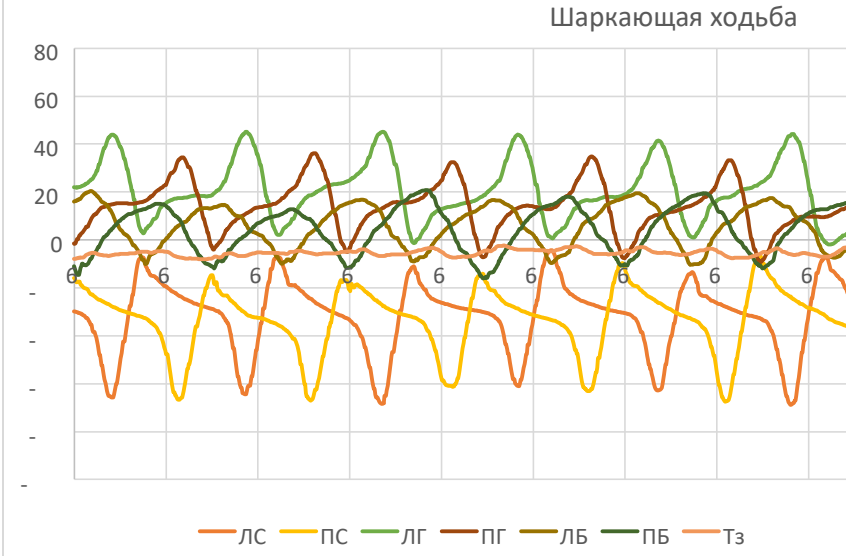
39. Betaflight Configurator [Электронный ресурс] // MulticopterWiki — свободная энциклопедия мультикоптеров. URL: http://multicopterwiki.ru/index.php/Betaflight_Configurator (дата обращения 09.02.2019)

40. Капанджи А.И. Нижняя конечность. Функциональная анатомия. – М. : Эксмо, 2017.

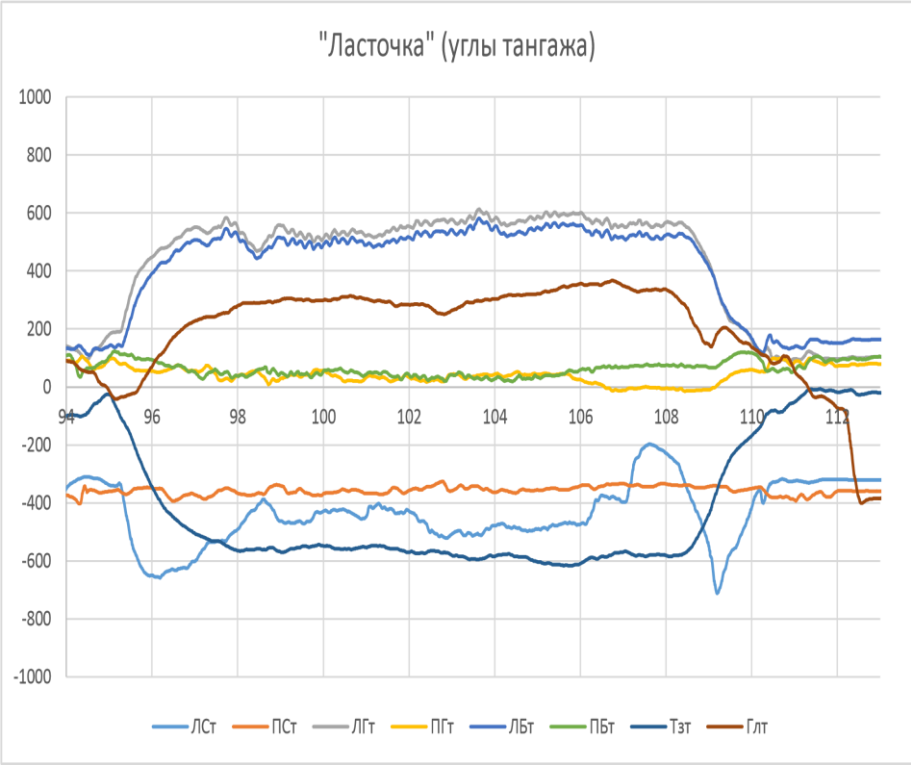
41. 6 ПРИЛОЖЕНИЯ




Ходьба
80
60
40
20
0
ЛС
ПС
ЛГ
ПГ
ЛБ
ПБ
Тз


Шаркающая ходьба
80
60
40
20
0
ЛС
ПС
ЛГ
ПГ
ЛБ
ПБ
Тз

Все величины углов приведены в десятых долях угловых градусов, время – в секундах.



Секция VII. Специальное машиностроение

Мобильная роботизированная платформа для помощи в сельском хозяйстве «Siberian Tiger»

Бондарь Георгий Евгеньевич

г. Москва, ГБОУ школа №354 им. Д.М. Карбышева, 10 класс

Научный руководитель: Богачева Татьяна Петровна, учитель информатики

СОДЕРЖАНИЕ

Введение

Глава 1. Теоретическая часть

1.1 Точное земледелие

1.2 Преимущества точного земледелия

1.3 Исследование 1

1.4 Исследование 2

1.5 Существующие аналоги

1.6 Эскизирование

1.7 Физический расчёт модели

1.8 Подбор компонентов

1.9 Методы изготовления печатных плат

Глава 2. Технологическая часть

2.1 Состав робота

2.2 Устройство робота

2.3 Общий чертёж робота

2.4 Таблица технологических карт

2.5 Программы и алгоритмы

2.5.1 Принцип управления

2.5.2 Алгоритмы Робота

Глава 3. Экономическая и экологическая оценка устройства

3.1 Материальные затраты

3.2 Расчёт экономической целесообразности

3.3 Экологическая оценка

Выводы

Заключение

Список источников информации

Приложения

Введение

Глобальная продовольственная проблема — одна из самых давних и важных проблем человечества. Устойчивое развитие сельскохозяйственного производства должно обеспечиваться не только организационно-экономическими мерами, но и уровнем научной обоснованности региональных систем земледелия. В настоящее время рекомендованные системы во многих случаях не обеспечивают рационального использования почвенно-климатических ресурсов, эффективного применения средств интенсификации земледелия, воспроизводства плодородия почв, экологической сбалансированности.

Цель проекта:

Рассмотреть возможность использования роботов и IT-технологий в сельском хозяйстве. Спроектировать и создать модель роботизированной мобильной платформы для решения задач диагностики и лечения растений, использующую современные технологии точного земледелия.

Гипотеза:

Я предполагаю, что предложенные технологии приведут к:

1. сокращению расхода лекарств и удобрений до 90%;
2. повышение урожайности и качества продукции;
3. минимизации негативного влияния на окружающую среду;
4. улучшению качества посевных площадей;
5. информатизации аграрного сектора;
6. контролю за урожаем;
7. составление разных типов карт полей с расширенным спектром визуальных возможностей;
8. мониторинг определенных участков почв для составления точной нормы удобрений.

Задачи:

1. Познакомиться с литературой по точному земледелию.
2. Спроектировать и собрать модель робота исследователя для ухода за растениями.
3. Спроектировать систему активной подвески для робота.
4. Написать программу управления роботом.
5. Создать чат-бота для робота-исследователя.
6. Изучить и применить нейронные сети для отличия здоровых и больных растений.
7. Изучить и применить машинное зрение для управления роботом.
8. Изучить и применить возможность общения объектов посредством сети WI-FI.

Оборудование

1. Raspberry pi3 B+
2. Микроконтроллеры ARDUINO
3. Web камера
4. Компьютер с ПО
5. 3D принтер
6. Роутер

Актуальность:

Современный мир, каким мы его знаем, во многом стал возможен благодаря революции в сельском хозяйстве. Технологический прогресс многократно повысил производительность труда в этой отрасли, и теперь достаточно небольшой процент людей, занятых в сельском хозяйстве способен прокормить все население планеты. Однако прогресс не стоит на месте, и находятся новые методики повышения эффективности отрасли. Одной из самых актуальных технологий современности является точное земледелие.

Глава 1. Теоретическая часть

1.1 Точное земледелие

Точное, или прецизионное земледелие, согласно определению, специалистов из РГАУ-МСХА им. К.А. Тимирязева, «представляет собой высшую форму адаптивно-ландшафтного земледелия, основанного на наукоемких агротехнологиях с высокой степенью

технологичности». В основе научной концепции точного земледелия лежат представления о существовании неоднородностей в пределах одного поля. Для оценки и детектирования этих неоднородностей используются новейшие технологии: системы глобального позиционирования, датчики, аэрофотоснимки и снимки со спутников, специальные программы для агроменеджмента на базе геоинформационных систем.

«Точное земледелие построено на получении, анализе, архивировании и консолидации разнообразных данных. Есть несколько групп датчиков: в сельхозтехнике, на полях (датчики влажности, температуры, уровня pH). Есть масса данных, которые поступают со спутников. Есть беспилотные технологии, которые все шире применяются в сельском хозяйстве: например, использование дронов позволяет перепроверить и уточнить информацию со спутников. Агрохимические данные позволяют строить послойные модели. Можно взять карту поля, точно его разметить, наложить показатели урожайности, через космос отследить, как вегетируют побеги, насколько однородны всходы, как нарастает биомасса с течением времени, соотнести эту информацию с соседними полями либо с показателями того же поля в прошлые годы и построить кривую, иллюстрирующую отставание, соответствие или опережение графика, - рассказывает Р. Куликов. – Понимая географические, ландшафтные и химические неоднородности поля, можно вносить удобрения или защитные препараты с учетом этих особенностей».

1.2 Преимущества точного земледелия

Следствием является значительная экономия - контроль и управление всеми этапами земледелия позволит экономить до 20% ресурсов (агрохимикатов, горюче-смазочных материалов, семенного материала), точнее учитывать сбор и логистику урожая. «Амортизация современной сельскохозяйственной техники - очень дорогое удовольствие. Методы точного земледелия позволяют сделать так, чтобы техника работала максимально эффективно и была возможность спланировать, на каких полях, в какой день она должна работать, какой препарат и куда именно вносить. Данные можно синхронизировать с управленческой отчетностью, что позволит, среди прочего, планировать закупку и использование удобрений».

Предлагаемые в России решения из области точного земледелия пока носят фрагментарный характер. Одни компании разрабатывают датчики, другие занимаются интеграцией, третьи - агрохимическими данными.

1.3 Исследование 1:

Развитие точного земледелия в нашей стране и за рубежем

Лидеры по количеству хозяйств, использующих элементы точного земледелия

Патентная активность стран в области точного земледелия

1 место	США
2 место	Германия Япония
3 место	Китай Франция Нидерланды

Описание проекта.

Проектируется небольшое фермерское хозяйство, которое имеет несколько полей в разной удаленности от фермерской усадьбы.

У фермера имеется:

- беспилотный летательный аппарат с видеокамерой,
- робот «Исследователь»,
- техническая станция
- обслуживающее программное обеспечение.

Использование БПЛА:

Сельское хозяйство является одной из перспективных сфер, применяющих беспилотные летательные аппараты для эффективного управления сельскохозяйственным производством.

Наземные исследования не всегда позволяют в полном объеме проанализировать и оценить состояние сельскохозяйственных угодий и проконтролировать процесс посева и уборки урожая. Наиболее рентабельным и действенным в данном случае является осуществление аэро-фото и видеосъемки с помощью беспилотных летательных аппаратов.

Применение БПЛА, по сравнению с аэрофотосъемкой, проводимой с помощью самолетов, имеет следующие преимущества:

- возможность съемки с небольших высот и вблизи объектов;
- оперативное получение снимков высокого разрешения;
- возможность применения в зонах чрезвычайных ситуаций без риска для жизни и здоровья пилотов.

1.4 Исследование 2:

Мы обратились в Российский государственный аграрный университет – Московская сельскохозяйственная академия имени К.А. Тимирязева («РГАУ-МСХА имени К.А. Тимирязева») к к.т.н, профессору Балабанову В.И., который познакомил нас с разработками роботизированных платформ для растениеводства и задачами в развитии точного земледелия.

Требования к точности систем позиционирования

Требуемая точность	Задача	Пример применения систем позиционирования
+/- 10м	Навигация	Определение координат полей
+\- 1 м	Выполнение операций Информация	· Мониторинг урожайности · Внесение удобрений · Автоматический сбор информации
+\- 10 см	Управление агрегатами	Контроль перекрытия смежных проходов при внесении удобрений и комбайновой уборке с.-х. культур
+\- 1 м	Контроль за выполнением точных операций	Механический способ борьбы с сорняками

Также я провел полевые испытания ходовой части робота на опытных полях Тимирязевской академии. (Фотографии в приложениях)

Существует система машинного зрения, позволяющая картировать поле. Система позволяет:

- обрабатывает снимок поля,
- следит за перемещением робота,
- строит карту поля по показаниям с датчиков, расположенных на роботе, строит маршрут роботу,
- дает задание роботу для внесения удобрений и полива растений.

Цифровая карта позволит:

- определять площади используемых пахотных земель и пустующих участков;
- осуществлять текущий контроль за состоянием посевов;
- оценивать всхожесть и степень спелости сельскохозяйственных культур, наблюдать за динамикой их развития;
- прогнозировать урожайность зерновых культур;
- оценивать состояние почвы;
- определять воздействие экологических и техногенных факторов на рост культур и урожайность;
- оценивать состояние посевов после стихийных явлений (пожаров, наводнений и т. п.);
- планировать комплекс агротехнологических работ для достижения планируемого урожая;
- проводить инвентаризацию и картографирование сельскохозяйственных угодий;
- привязывать изображения наиболее аномальных участков и проблемных территорий к их географическим координатам.

1.5 Аналоги:

У всех представленных агро-роботов имеются поворотные колёса, позволяющие им перемещаться всенаправленно. Мой робот отличается от представленных наличием активной подвески, позволяющей удерживать горизонтальное положение корпуса на холмистой/горной местности.

1.6 Эскизирование:

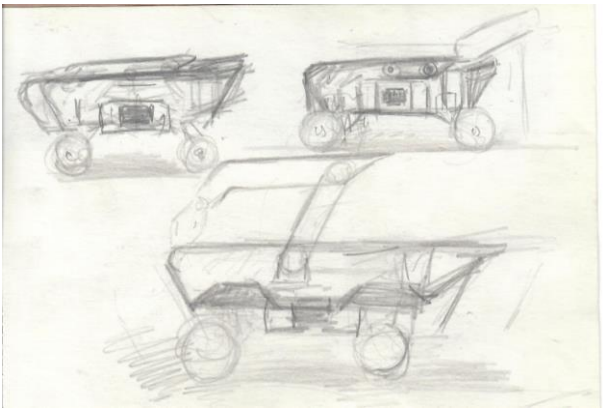
Я изучил организацию работ на полях: для проезда тракторов и прочей техники через каждые 15-20 метров есть технологические линии - полосы не засеянной земли, на разных полях они могут быть разной ширины, поэтому для подстройки ширины колёс и высоты клиренса у робота можно реализовать систему активной подвески. Система активной подвески позволит выравнивать корпус робота по горизонту, если тот будет работать в холмистой местности.

Для минимизации пространства поля, отведённого на технологические линии, я хочу реализовать всенаправленную платформу для робота, то есть каждое колесо будет иметь свой поворотный столик.

Чтобы сократить силы сопротивления поворота колёс, их вылет должен быть минимален или равен нулю.

Когда с функциональная часть была проработана, я начал искать стиль и форму. Я вдохновился такими качествами, как: Сила, целеустремлнность, брутальность и эстетика движения Сибирского тигра.

После приблизительных расчётов я нарисовал эскизы робота.



СЛИШКОМ

1.7 Физический расчёт модели

Я строю робота, то есть его прототип и уменьшенную копию, поэтому определил его размеры 500х500мм. Его приблизительный вес будет от 10 до 20 кг. Значит, на каждую из 4 опор будет приходится от 2,5 до 5 кг номинальной нагрузки. Каждая опора крепится к своему поворотному механизму, поэтому, для минимального сопротивления, вылет будет нулевой, получается, ширина колёс не должна мешать стойке, к которому это колесо крепится, и, центральная ось вращения колеса должна совпадать с центром ширины этого колеса.

Чтобы выиграть в крутящем моменте, в поворотной системе я поставлю ременную понижающую передачу.

Система активной подвески и изменения клиренса, по эскизам модели, должна изменяться на 100-120 мм. Для самоблокировки узла системы я буду использовать червячный механизм.

1.8 Подбор компонентов

Робот должен точно перемещаться на заданную величину, поэтому для точных движений я буду использовать шаговые двигатели Nema17 с крутящим моментом 4,5 кг/см, а для управления я буду использовать драйвера TB6600. В качестве маршевых двигателей я буду использовать Nema17 с планетарным редуктором 1 к 5.

Для управления роботом я использую микрокомпьютер raspberry pi 3 B+, для управления драйверами Arduino mega 2560 pro.

Глава 2. Технологическая часть

2.1 Состав робота:

- 2 шаговых моторов NEMO-17
- 12 драйверов TB6600
- плата Arduino MEGA 2560 pro
- Raspberry pi 3B+
- Raspberry связан с Arduino по шине I2C.
- Корпус:
  - конструкционный профиль 20x20 мм
  - детали, распечатанные на 3D-принтере:

- колеса
- элементы крепежа

2.2 Устройство робота

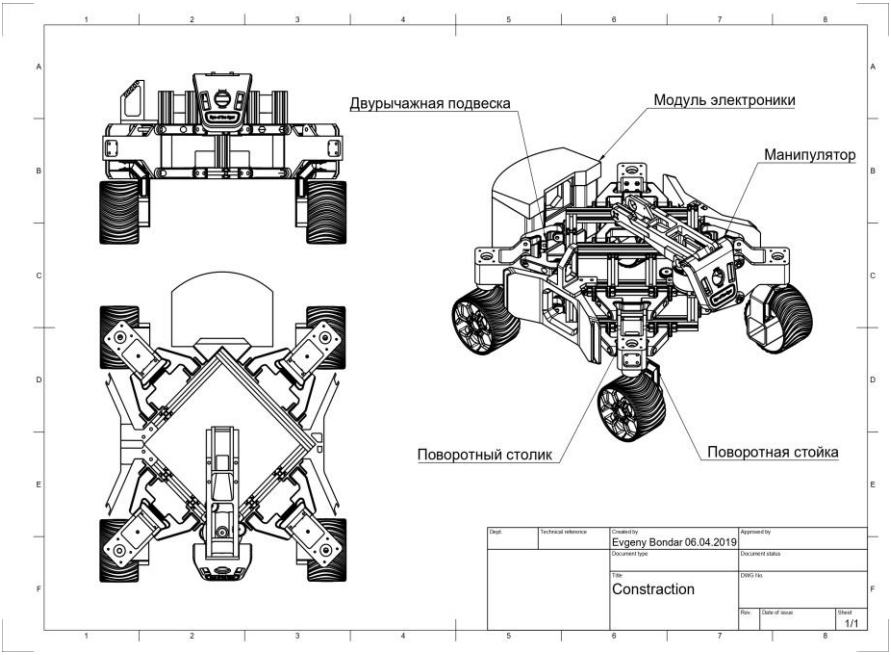
В конструкции робота использованы следующие типы передаточных механизмов:

- Винт-гайка для подъёма,
- ременная передача для поворота колёс,
- планетарный редуктор для движения.

Разработана активная система подвески, которая позволит выравнивать и удерживать горизонтальное положение корпуса робота.

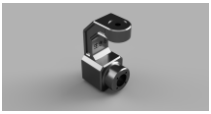
Система крепления колес робота позволяет поворачивать каждое колесо на 180 градусов, что позволяет осуществлять движение робота в любом направлении (вправо, влево, вперед, назад, на заданный угол) без поворота корпуса. Мы также разработали платы разводки питания для драйверов (TB6600) моторов (Nema17), собраны специальные высокоёмкостные аккумуляторные элементы питания (14.8V) для робота.

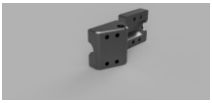
2.3 Общий чертёж робота:



2.4 Таблица технологических карт:

номер детали	Название	Количество	Технологическая карта	Вид
№ 1	Каркас из конструкционного профиля	1	ТК№1	

	20х20			
№ 2	Нижнее плечо подвески	4	ТК №2	
№ 3	Верхнее плечо подвески	4	ТК №3	
№ 4	Кулак	4	ТК №4	
№ 5	Стойка	4	ТК №5	
№ 6	Ступица	4	ТК №6	
№7	Колесо	4	ТК№7	
№ 7	Стенка, деталь корпуса	3	ТК №7	

№ 8	Крепление подвески	16	ТК №8	
№ 9	Кулак к шпильке	4	ТК №9	
№ 10	Плата разводки питания для драйверов шаговых двигателей	4	ТК №10	
№ 11	Матрица для Arduino Mega 2560	1	ТК №11	
№ 12	Крепление для плат	1	ТК №12	
№ 13	Крепление для бака с жидкостью	1	ТК №13	

№ 14	Высокоёмкостный аккумулятор	2	ТК №14	

2.5 Программы и алгоритмы:

2.5.1 Принцип управления:

У робота есть главный бортовой компьютер - Raspberry pi, который отдаёт общие приказы системам, задействует разные устройства на роботе. Управление моторами происходит, когда Главный компьютер посылает сигналы на Arduino Mega Pro, которая уже отдаёт приказы Драйверам шаговых двигателей. (Две моих схемы: одна для матрицы с Arduino Mega, а другая для разводки питания 3 драйверов TB6600)

Также я разработал плату управления шаговыми двигателями на основе расширителя портов MCP23S17. Это позволяет упростить систему управления двигателями и снизить её стоимость.

Языки программирования: – программирование бортового компьютера на языке Python, и Arduino программа написана на языке C++ для Arduino Mega Pro.

Я использовал библиотеку SMBus для связи с Arduino по I2C,

библиотеку robotNet для связи Raspberry pi 3 B + с компьютером по сети Wi-Fi.

2.5.2 Алгоритмы Робота.

Все действия робота задаются Байт-кодом, то есть каждое движение кодируется своим номером.

```
def runCode(code, place):     //Эта функция перебирает каждое число Байт-кода (движения)
        i = 01
        code.append(0)
        while code[i] != 0:
                i += runOperation(code[i],code[i+1],place)
                i += 1
        print("END")
        return place

def runOperation(c,n,place): //В этой функции описаны действия, соответствующие определённому числу.

        print('c=',c,' n=',n)
        if c == 0:
```

```
            print("OOPS: zero!")
            return -1
    elif c == 1:
            write_data2(c)
            return 0
```

Волновой алгоритм:

Для построения маршрута для робота на поле я использую волновой алгоритм.

В качестве карты поля использую ту, что составляет квадрокоптер, когда делает пролёт над полем.

map1 = [[0,0,0],[0,0,0],[0,0,0]] //Для использования этого алгоритма нам необходимо преобразовать технологические линии поля в вид графа (двумерного массива), чтобы программа могла создать маршрут

def path(xn,yn,xk,yk): //В этой функции простраивается маршрут, каждому направлению движения определёно число, каждое действие записывается в массив, а потом этот байт-код загружается в программе исполнения.

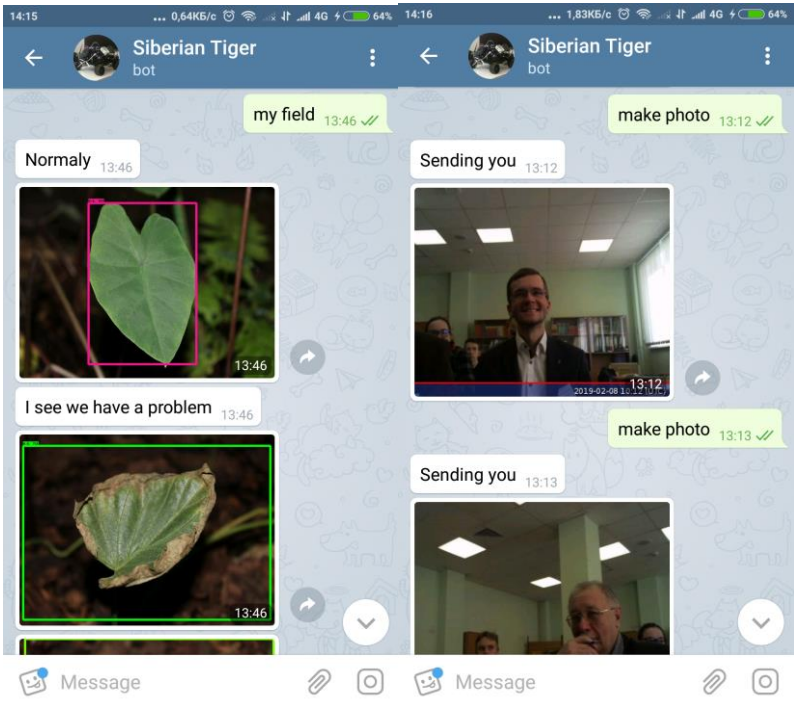
```
    PATH = []
    n = 1
    print(xn,yn,xk,yk)
```

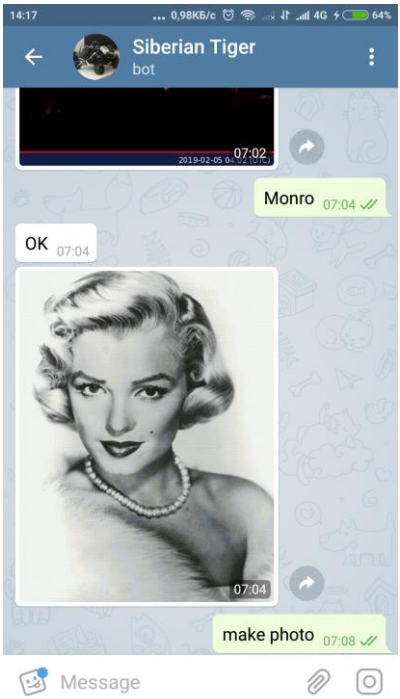
```
while not (xn==xk and yn==yk):
        if yn+1<3:
                if yk>yn and map1[xn][yn+1]==0:
                        yn+=1;n+=1
                        print("/FORWARD/")
                        PATH.append(5)
        if xn+1<3:
                if xk>xn and map1[xn+1][yn]==0:
                        xn+=1;n+=1
                        print("/RIGHT/")
                        PATH.append(7)
        if yn+1>0:
                if xk<xn and map1[xn-1][yn]==0:
                        xn-=1
                        n+=1
                        print("/LEFT/")
```

```
                    PATH.append(8)
            if yn+1>0:
                if yk<yn and map1[xn][yn-1]==0:
                    yn-=1;n+=1
                    print("/BACK/")
                    PATH.append(6)
    print(map1)
    PATH.append(0)
    print(PATH)
    return PATH
```

Чатбот робота:

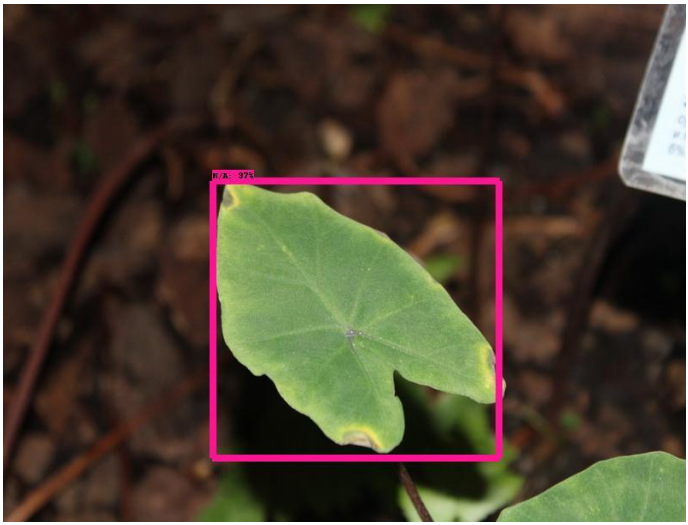
Для простоты работы фермера я написал программу Чатбота. Фермер сможет получать детальную информацию о состоянии полей, фотографии с места событий, конфигурацию системы в чат мессенджера где бы вы ни были.


14:15
0,64КБ/с
4G
64%
Siberian Tiger
bot
my field 13:46
Normaly 13:46
13:46
I see we have a problem 13:46
13:46
Message
14:16
1,83КБ/с
4G
64%
Siberian Tiger
bot
make photo 13:12
Sending you 13:12
13:12
make photo 13:13
Sending you 13:13
Message



Нейронные сети:

Для сравнения фотографий и выявления признаков заболеваний растений я
использую свёрточные нейронные сети. На данных примерах обученная
нейросеть нашла на фотографии листья съедобной Колоказии. Данная
программа предназначена для выявления вида растения, а после этого его
возможной болезни.


N/A: 97%

N/A: 53%

Глава 3. Экономическая и экологическая оценка устройства

3.1. Материальные затраты

Расчёт себестоимости робота

Комплектующее	**Количество**	**Стоимость за единицу**	**Общая стоимость**
Драйвера TB6600	12	660 р.	7920
ШД Nema17	8	1100 р.	8800
ШД Nema17 gearbox 1/5	4	1600 р.	6400
Сервопривод полного хода	3	1100	3300
Плата управления сервоприводами.	1	400	400
Поворотный столик манипулятора	1	500	500
Рычаги манипулятора	2	400	800
Плата разводки питания для ШД	4	Примерная оценочная стоимость 900Р	3600

Матрица для Arduino 2560	1	Примерная оценочная стоимость 800 р.	800
конструкционный профиль 20х20	2м	4500 р	4500
Бак для воды	1 шт	300 р.	300
Arduino 2560 pro	1 шт.	760 р.	760
Raspberry Pi 3 B+	1 шт.	3500 р.	3500
Комплект пластиковых деталей к роботу	40 деталей		15000
ступицы к колёсам из Д16Т	4 шт	200р.	800р
Колёса с протекторами	4шт	500 р.	2000
Болты и гайки	**	--	
Шлейфы, переходники, провода и шлейфы	**	--	
Аккумулятор 8000 mah	2 шт.	7 тыс.р.	7000
HD Камера	2 шт.	2.5 тыс.руб.	5000
Датчик дальности	3 шт	360 р.	1080

Датчик влажности	1 шт	600 р.	600
Датчик температуры	1 шт	700 р.	700
Радиатор для Raspberry	1 шт	1490 р.	1490
Стеклоткань	10 кв.м	100	1000
Эпоксидная смола	5л	500	2500
Эпоксидная шпатлёвка	3л	500	1500
Наждачная бумага	10 листов	20	200
Пластик чёрный PLA	5кг	1050	5250
Заготовка двусторонний текстолит	10	100	1000
		Общая стоимость:	76150 р

Это неполная стоимость материалов, из который собран робот, без учёта потраченного времени на обработку материалов, сборку, программирование. Также, это себестоимость одной модели, однако, было несколько версий каждого из элементов, поэтому цена разработки намного выше себестоимости одной модели.

3.2 Расчёт экономической целесообразности:

Задачей этого робота является регулярное наблюдение за посадками: определение болезни у растения, подбор лекарства, подбор удобрения, полив конкретных растений. То есть робот выполняет задачу агронома.

Средняя зарплата агронома в хозяйстве составляет 45 тыс.р., значит, при условии, что агроном работает по 8 часов в неделю, в среднем 22 дня в месяц, его по часовая зарплата 256 р. Агроном - человек, для его обучения требуется 4-5 лет в ВУЗе, время, деньги.

Допустим, что рыночная стоимость робота, без квадрокоптера, 1 млн. р. - стоимость трактора. Он на электроходу, его конструкция модульная, выполняет конкретные задачи агронома, но круглосуточно. Экономически, его создать проще, чем обучить агронома, поэтому робот более эффективен для конкретного фермерского хозяйства.

Круглосуточная работа моей системы позволит своевременно выявить заболевания и спасти растения, и таким образом спасти урожай. При росте культур бывают потери из-за вовремя не выявленных зон болеющих растений (Фитопатология как наука о болезнях растений, занимается снижением и ликвидацией потерь урожая от болезней растений. Многие исследователи считают, что болезни растений снижают общую продукцию сельского хозяйства

(включая хранение) примерно на 15—20%.) Поэтому, чем больше Робот спасёт растений, тем быстрее он окупится.

3.3 Экологическая оценка:

Робот используется для одной из задач точного земледелия, а именно, он дифференцированно осуществляет полив и обработку растения, что в целом влияет на экосистему поля.

Детали из PLA пластика можно легко утилизировать - он биоразлагаемый. Из минусов: металлический каркас, корпус, платы просто так не утилизируешь, тут больше затрат.

Название материала	**использование в роботе**	**экологичность**
конструкционный профиль, чёрный, анодированный.	Внутренний каркас	Алюминий можно использовать повторно, что экономит до 95 процентов энергии, потребляемой при первоначальной выработке.
PLA пластик	Детали робота	**Э**тот пластик биоразлагаемый, так как делается из крахмала, добытого, например, из кукурузы.
ШД Nema17	Все двигатели робота	Можно утилизировать, сдать в металлолом.

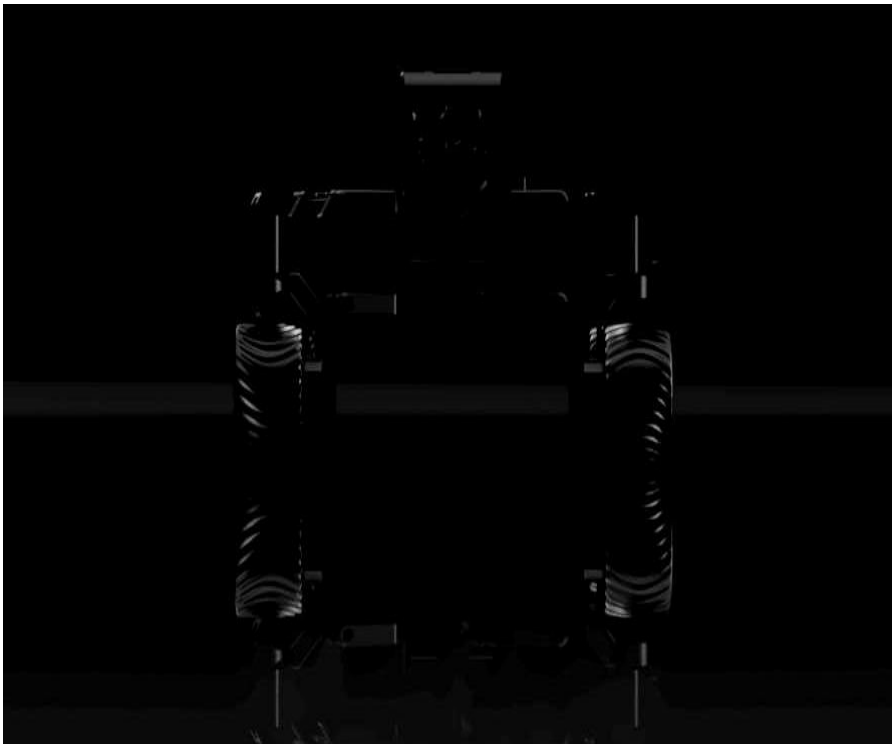
Аккумулятор Li-ion	Энергоноситель	Сдать на переработку после существенного сокращения запасающей способности. Сначала извлекается и нейтрализуется щелочной электролит,а далее ─ разделяются металл и пластик.

Реклама изделия:

Я рекламировал робота вместе с картиной на разных форумах и выставках, чтобы показать, что мой робот - это не просто проект, а нечто большее, как произведение искусства. Хотелось произвести резонанс, донести до публики идею красивой робототехники (искусства робототехники).

Для рекламы я сделал тизер проекта и назначил дату публичных презентаций, некоторые из которых прошли с 20-22 марта.

NIMATIC
IZ.RU



Выводы:

Разработанный прототип способен передвигаться по пересеченной местности и производить полив растений. Он оборудован камерой для наблюдения за работой платформы. Дальнейшая работа сосредоточена на переход к более мощным контроллерам, способным увеличить производительность и скорость анализа данных от различных датчиков, подбору специализированных датчиков. Отдельное

внимание уделяется распознаванию образов при помощи камеры. Определены задачи, которые требуется решить при помощи камеры: — определение растения; — анализ растения по форме листа, характеристическим углам; — определение диаметра стебля, выявление симптомов болезни или нарушения развития (очаги поражения, увядшие листья и др.). Одной из основных функций устройства является мониторинг растений и предупреждение фермера о возникновении исключительных состояний, например, о появлении вредителей.

Заключение:

В результате работы над проектом я:

1. Познакомился с литературой по точному земледелию.
2. Спроектировал и собрал модель робота исследователя для ухода за растениями.
3. Спроектировал систему активной подвески для робота.
4. Написал программу управления роботом.
5. Создал чат-бота для робота-исследователя.
6. Изучил и применил нейронные сети для отличия здоровых и больных растений.
7. Изучил и применил машинное зрение для управления роботом.
8. Изучил и применил возможность общения объектов посредством сети WI-FI.

В настоящее время проект выполнен на 80%.

Список источников информации

1. А.П.Алексеев, А.Н.Богатырев, В.А.Серенко, Робототехника, Москва «Просвещение», 1993г.
2. К.В. Рыжов, Сто великих изобретений, Москва «Вече», 1999 г.
3. И.М.Макаров, Ю.И.Топчеев, Робототехника: история и перспективы, Москва «Наука», 2003 г.
4. МакГрат, М. Программирование на Python для начинающих / М. МакГрат. - М.: Эксмо, 2015. - 192 с.
5. Саммерфилд, М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. - СПб.: Символ-плюс, 2015. - 608 с.
6. Балдин, К.В. Математическое программирование: Учебник / К.В. Балдин, Н.А. Брызгалов. - М.: Дашков и К, 2012. - 220 с.
7. Димитрин, Ю. Математическое программирование в примерах и задачах: Учебное пособие / Ю. Димитрин. - СПб.: Лань П, 2016. - 352 с.
8. https://kubsau.ru/upload/foresight/elements.pdf
9. http://mcx-consult.ru/d/77622/d/tochnoe-zemledelie.pdf
10. https://dic.academic.ru/dic.nsf/ruwiki/331043
11. https://dic.academic.ru/dic.nsf/ruwiki/331043
12. https://agronews.com/by/ru/news/agrosfera/2017-08-31/tochnoe-zemledelie

13. А. Барсуков, Компоненты и решения для создания роботов и робототехнических систем, Издательский дом «ДМК-пресс», 2005 г.

14. Навигационные технологии в сельском хозяйстве. Координатное земледелие. Учебное пособие / В.И. Балабанов, А.И. Беленков, Е.В. Березовский. – М.: Издательство РГАУ-МСХА имени К.А. Тимирязева, 2013. – 117 с.: ил.

Приложения:

Я проводил испытания активной подвески на опытных полях Тимирязевской Академии.

Технологические карты:

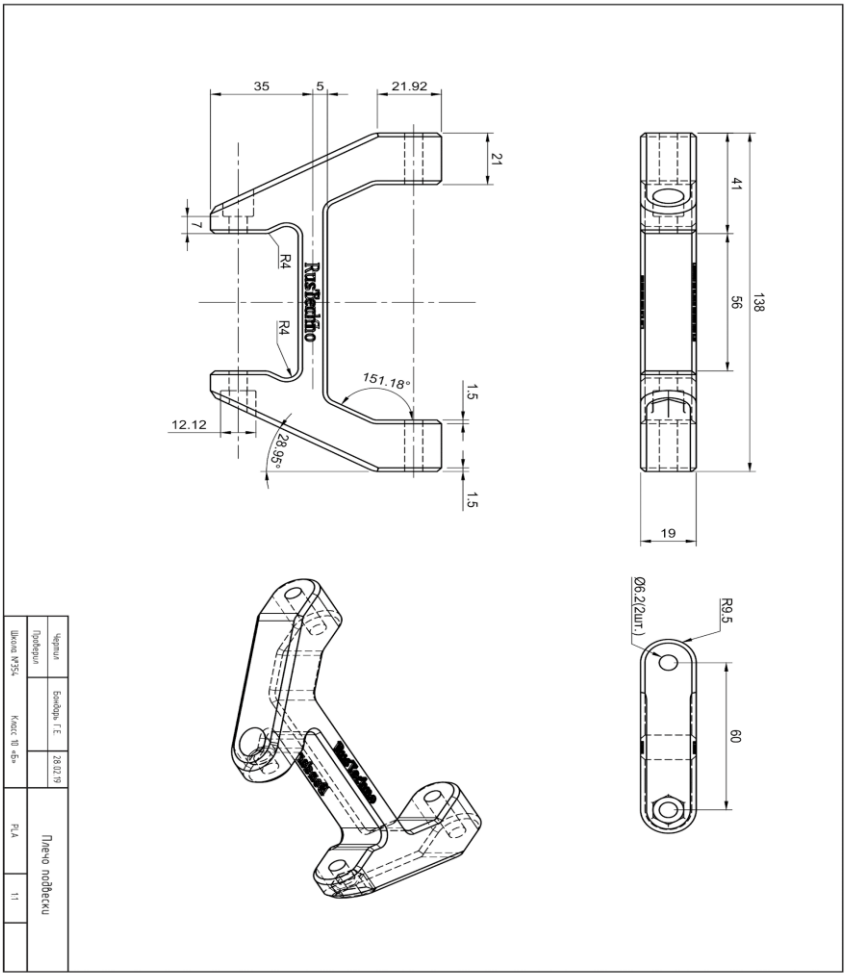
Следующие детали разработаны с помощью программы Fusion360 и напечатаны на 3D принтере (аддитивные технологии) Wanhao Duplicator 4S. В качестве пластика выбран биоразлагаемый PLA 1.75mm чёрного цвета.

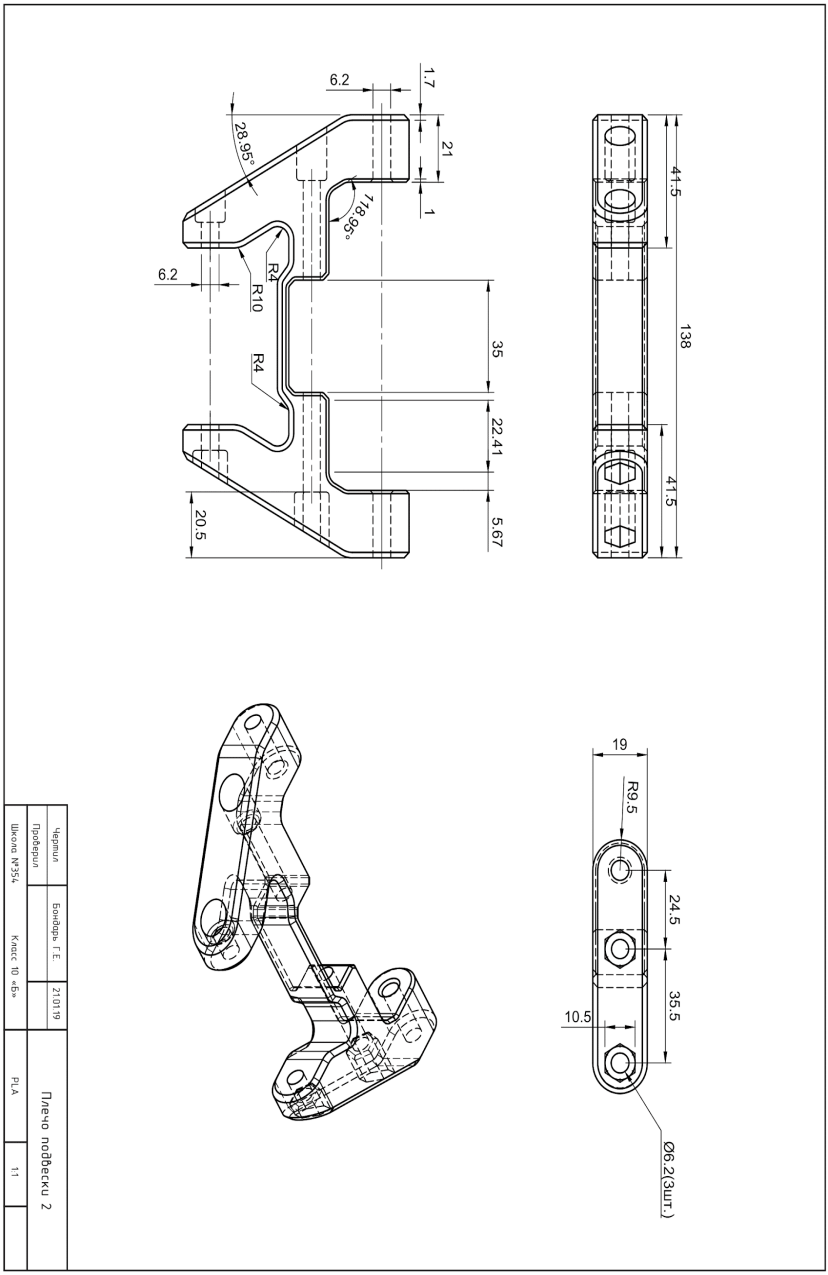
В процесс изготовления одной детали входят такие технологические операции, как установка модели этой детали в программу принтера, печать детали, съём готовой заготовки, рассверливание готовых отверстий, шлифовка, доводка правильной геометрии детали.

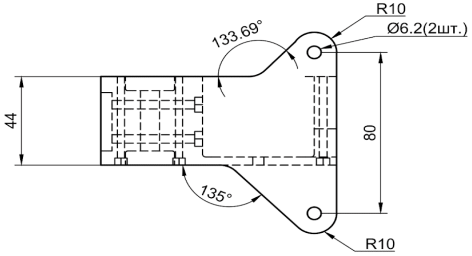
Универсальная ТК Изготовления деталей аддитивными технологиями (моделирование методом послойного нанесения)

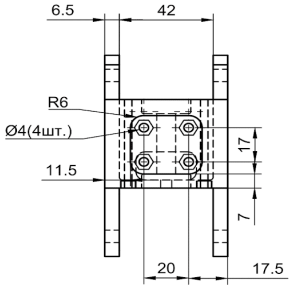
№	Название технологической операции	Инструменты и приспособления
1	Моделирование	компьютер, программа Fusion360
2	Подготовка задания станку	Компьютер, программа Simplify3D
3	Съем готовой модели со стола	Шпатель
4	Рассверливание	Дрель, сверло с конусом Морзе
5	Шлифовка	Наждачная бумага
6	Снятие фасок	Напильник

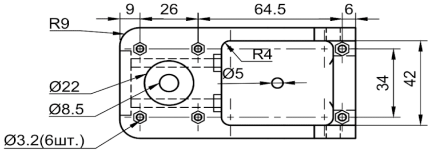
Далее представлены чертежи деталей: **Чертёжи нижнего плеча подвески, нижнего плеча подвески, кулака поворотной системы, стойки поворотной системы робота, ступицы для колеса, колеса для робота, Вала для системы стабилизации, бронзовой гайки для вала системы стабилизации**

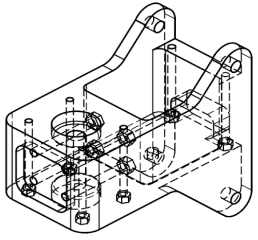

35
5
21.92
21
7
R4
R4
RusTechno
151.18°
1.5
12.12
28.95°
1.5
41
56
138
19
Ø6.2(2шт.)
R9.5
60
Чертил
Бондарь Г.Е.
28.02.19
Проверил
Школа №354
Класс 10 «Б»
PLA
1:1
Плечо подвески


6.2
1.7
28.95°
21
1
118.95°
6.2
R4
R10
R4
35
22.41
5.67
20.5
41.5
138
41.5
19
R9.5
24.5
35.5
10.5
Ø6.2(3шт.)
Чертил
Бондарь Г.Е.
21.01.19
Проверил
Школа №354
Класс 10 «Б»
PLA
1:1
Плечо подвески 2

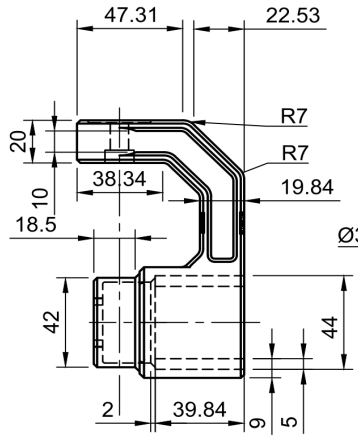


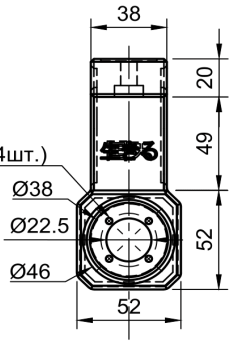


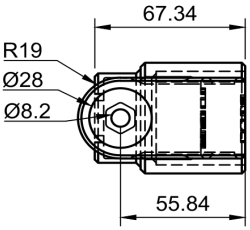


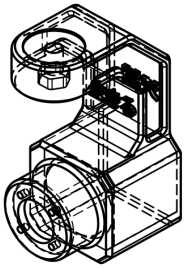


Чертил	Бондарь Г.Е.	28.01.19	Платформа суппорта	
Проверил				
Школа №354	Класс 10 «Б»		PLA	1:2

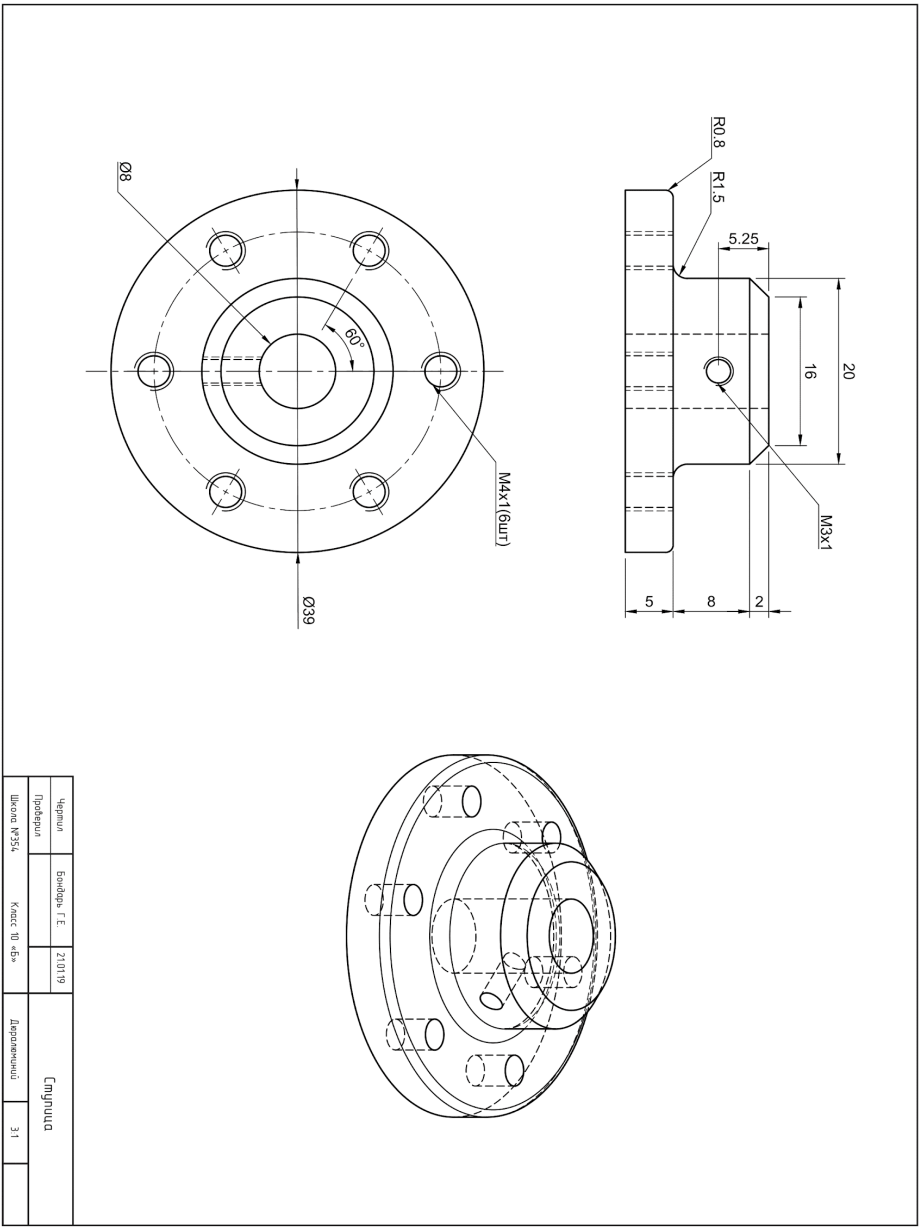


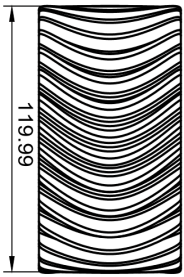


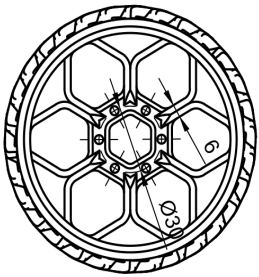


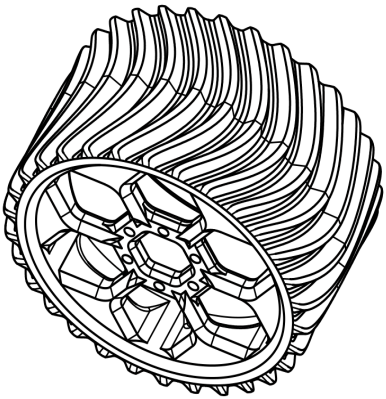


Чертил	Бондарь Г.Е.	21.01.19	Суппорт ходового двигателя		
Проверил					
Школа №354	Класс 10 «Б»		PLA	1:2	

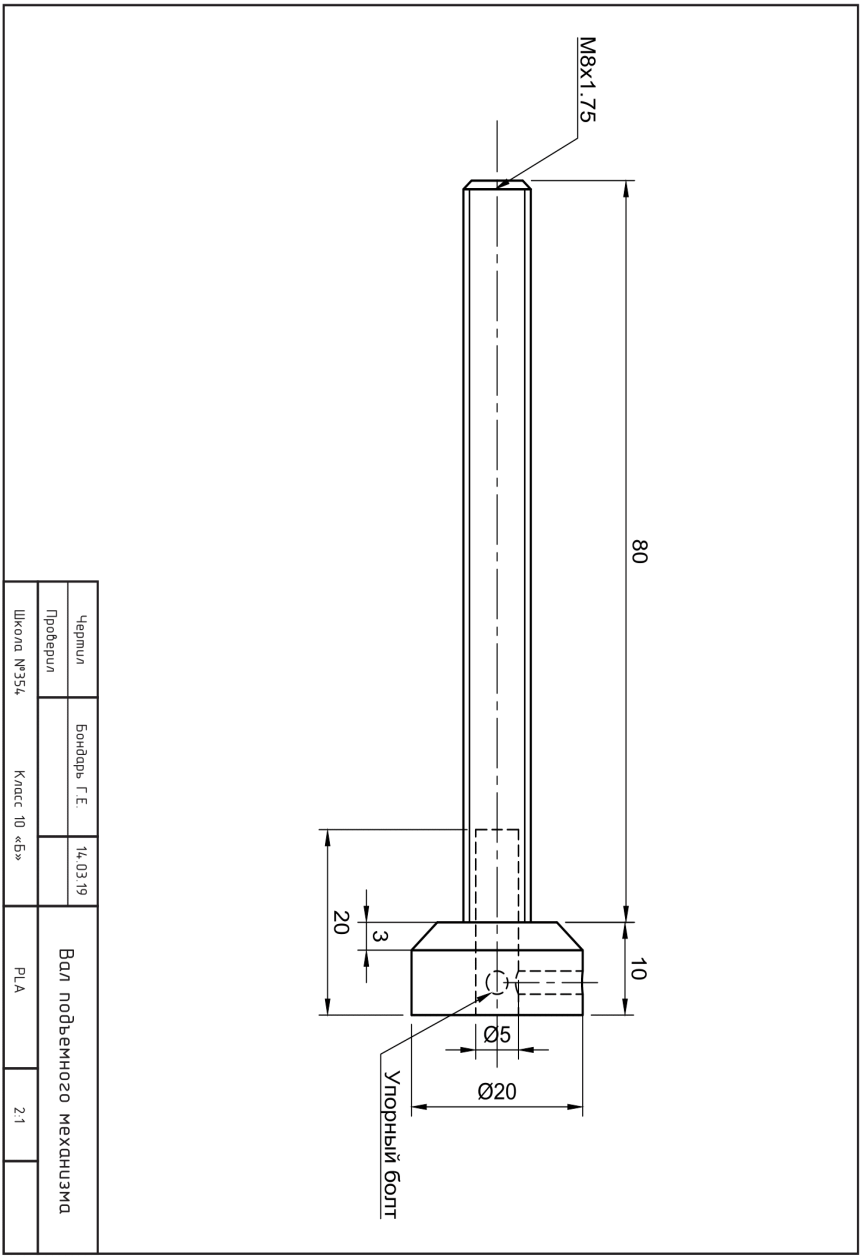

Ø8
60°
M4x1(6шт)
Ø39
R0.8
R1.5
5.25
16
20
M3x1
5
8
2
Чертил
Бондарь Г.Е.
21.01.19
Проверил
Школа №354
Класс 10 «Б»
Дюралюминий
Ступица
3:1

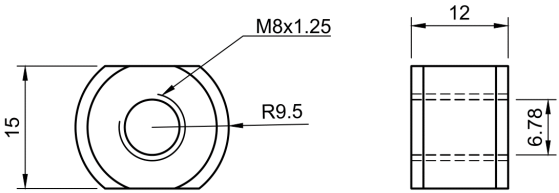




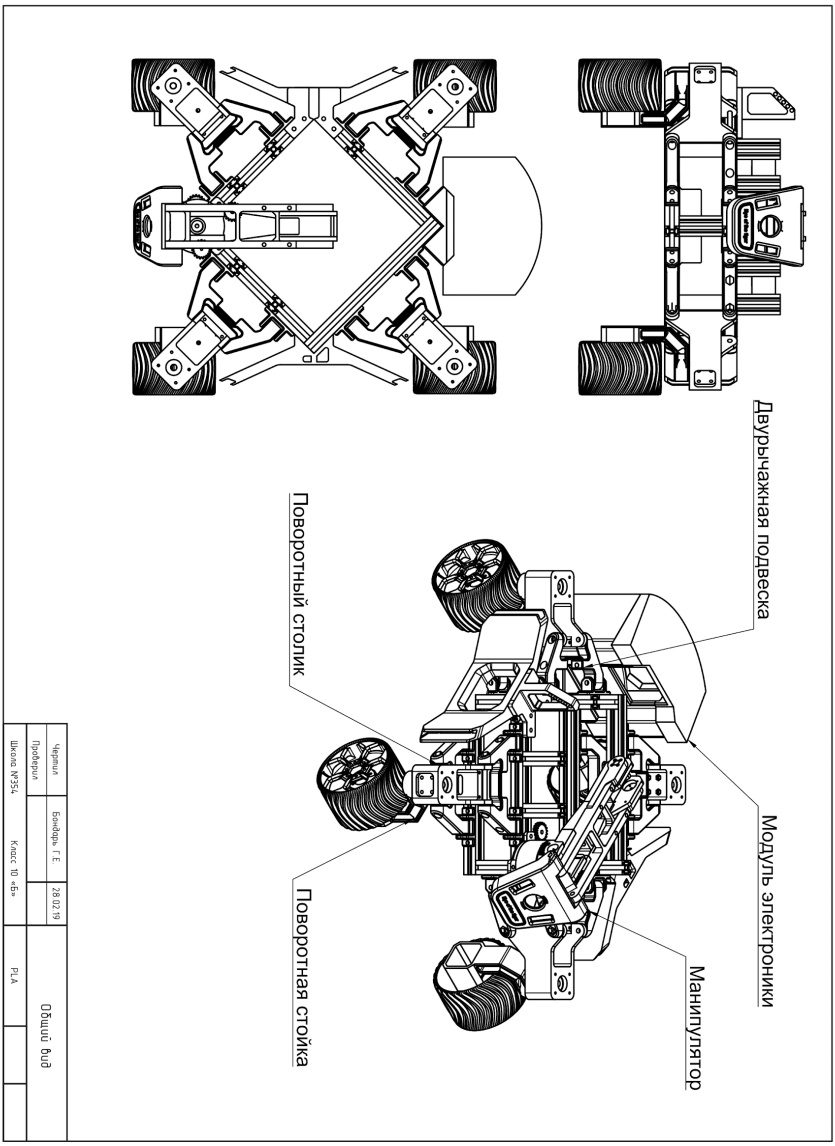


Школа №354	Класс 10 «Б»		Чертил	Бондарь Г.Е.
			Проверил	
PLA	1:2	210119	Колесо	


M8x1.75
80
20
3
10
Ø5
Ø20
Упорный болт
Чертил
Бондарь Г.Е.
14.03.19
Проверил
Школа №354
Класс 10 «Б»
Вал подъемного механизма
PLA
2:1



Чертил	Бондарь Г.Е.	06.04.19	Гайка подъемного механизма		
Проверил					
Школа №354	Класс 10 «Б»		Латунь	1:1	


Двурычажная подвеска
Поворотный столик
Модуль электроники
Манипулятор
Поворотная стойка
Чертил
Бондарь Г.Е.
28.02.19
Проверил
Школа №354
Класс 10 «Б»
PLA
Общий вид

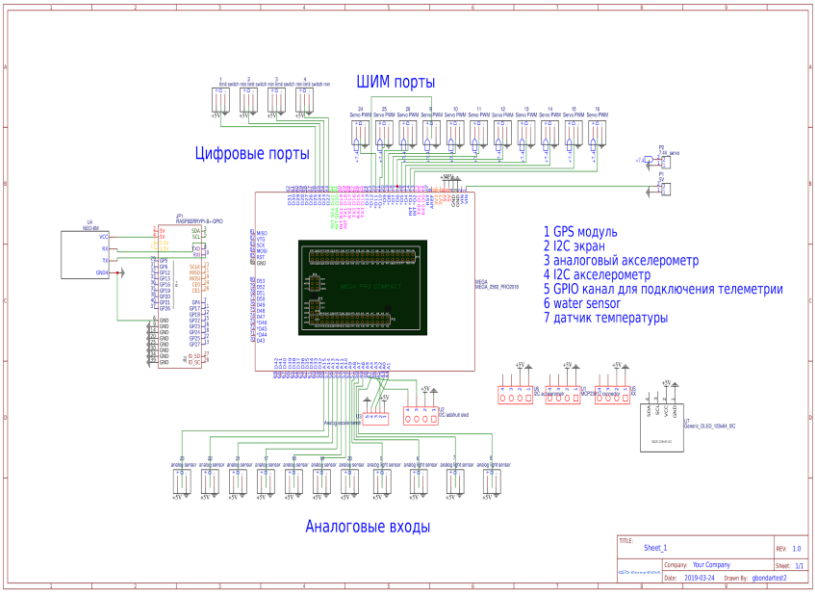
Универсальная ТК изготовления печатных плат

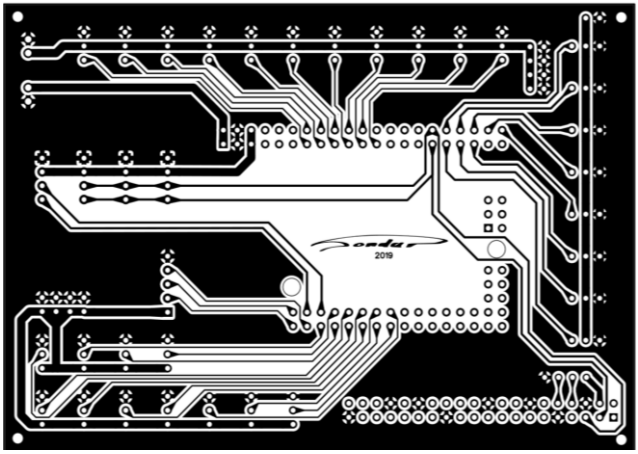
№	Название технологической операции	Инструменты и приспособления
1	Моделирование электрической схемы	EasyEda
2	Изготовление рисунка печатной платы	
	Выбор материала	Штангенциркуль, линейка
	Резка	Ножовка по металлу
3	Печать рисунка	УФ-принтер
4	Травление	Раствор персульфата аммония/хлорного железа/перекиси водорода+лимонной кислоты
5	Накернивание	Кернер, киянка
6	Сверление	Сверлильный станок,

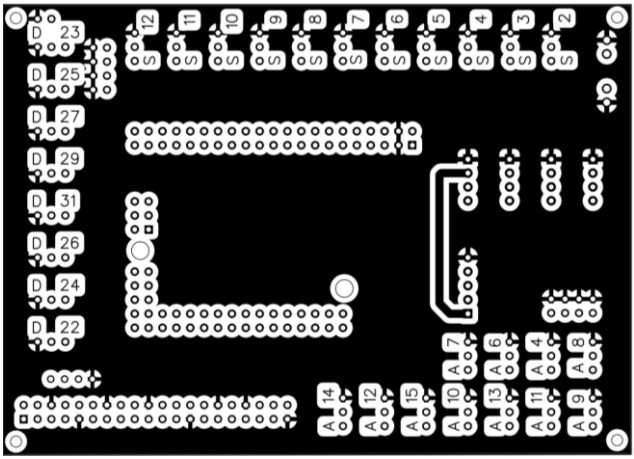
		стальная призма, сверло
7	Удаление защитного рельефа	Растворитель, ватный диск
8	Нанесение защитной маски	Акриловая маска, кисть
9	Монтаж элементов	Паяльник, флюс для пайки, радиоэлементы.

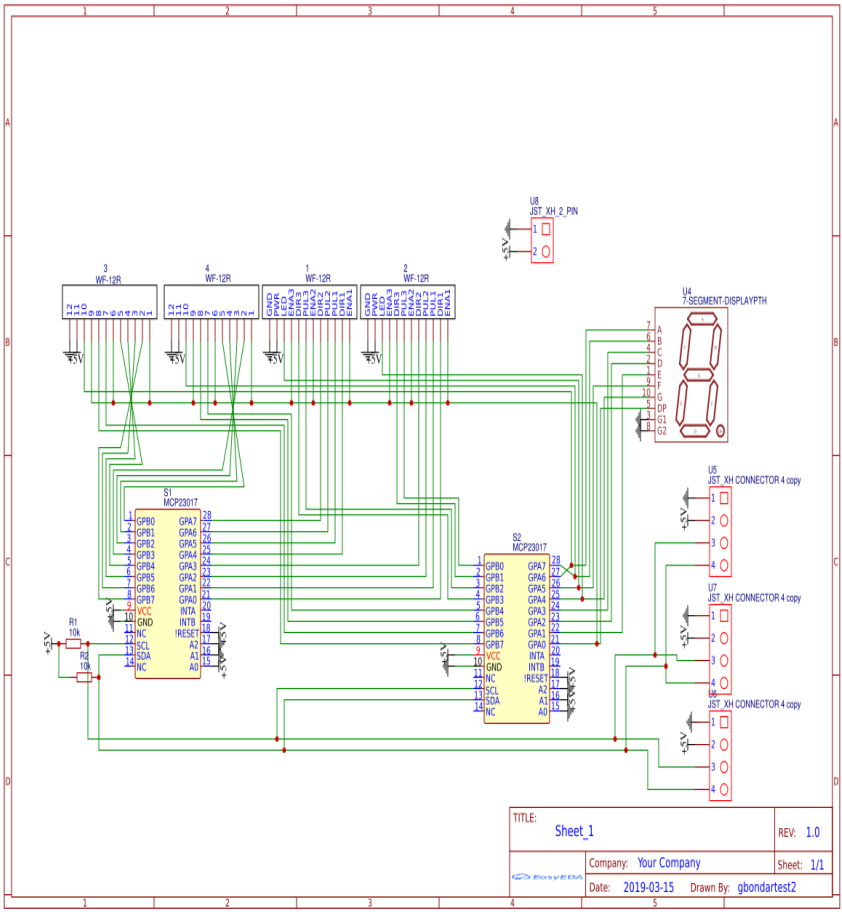
Электрическая схема и внешний вид платы управления моторами:

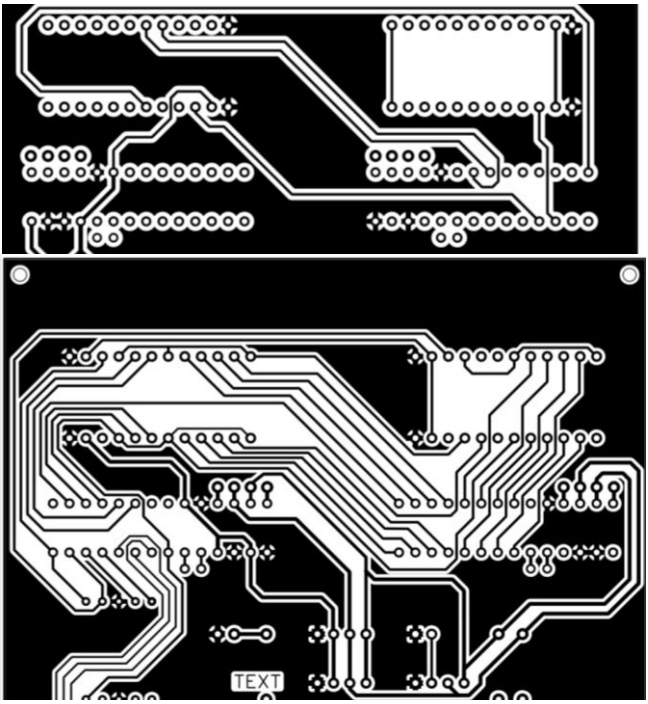
Электрическая схема платы разводки питания и её внешний вид:

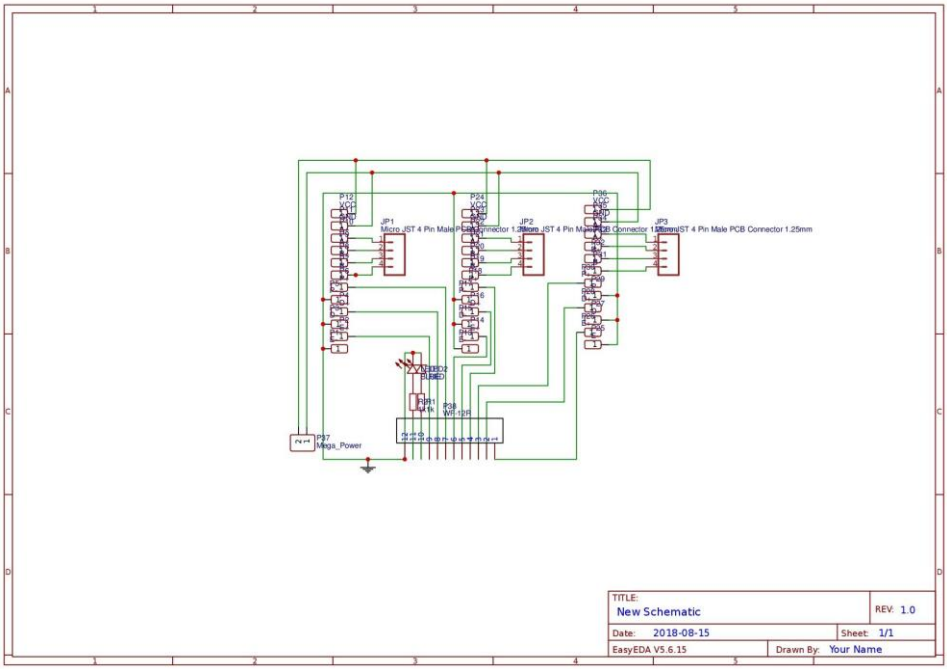


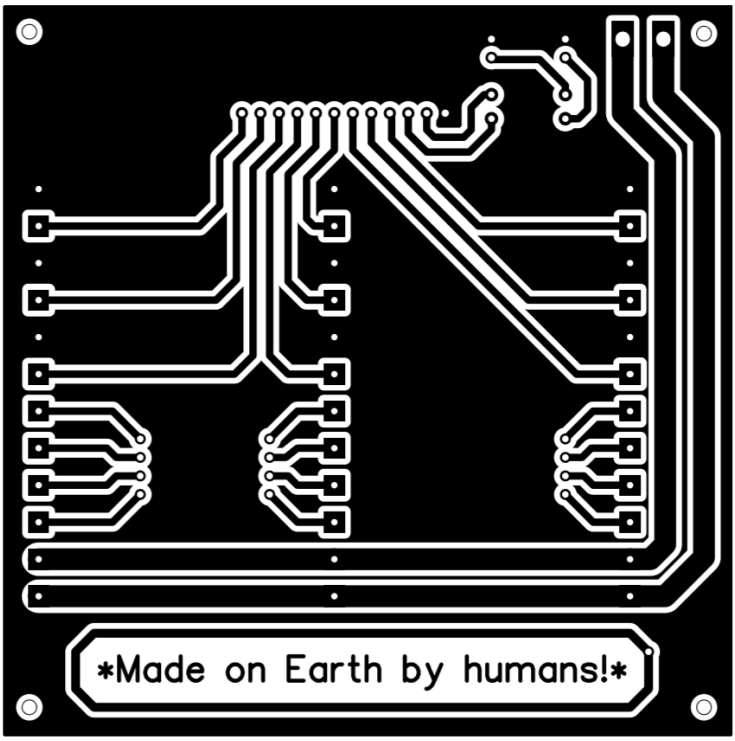


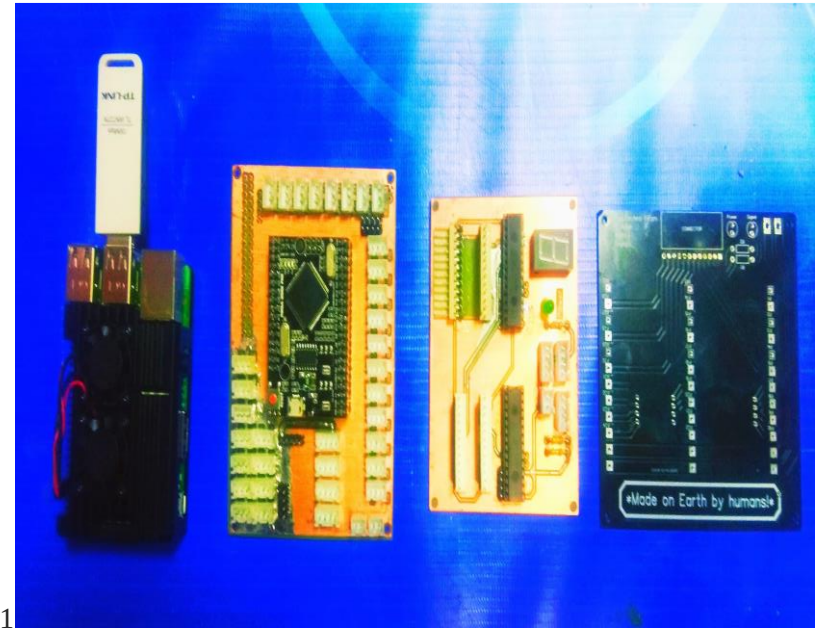

D 23
D 25
D 27
D 29
D 31
D 26
D 24
D 22
S 12
S 11
S 10
S 9
S 8
S 7
S 6
S 5
S 4
S 3
S 2
A 7
A 6
A 4
A 8
A 14
A 12
A 15
A 10
A 13
A 11
A 9


U8
JST_XH_2_PIN
3
WF-12R
4
WF-12R
1
WF-12R
2
WF-12R
U4
7-SEGMENT-DISPLAYPTH
S1
MCP23017
S2
MCP23017
R1
10k
R2
10k
U5
JST_XH CONNECTOR 4 copy
U7
JST_XH CONNECTOR 4 copy
U6
JST_XH CONNECTOR 4 copy
+5V
TITLE:
Sheet_1
REV: 1.0
Company: Your Company
Sheet: 1/1
Date: 2019-03-15
Drawn By: gbondartest2


TEXT


JP1
JP2
JP3
P37
Mega_Power
TITLE:
New Schematic
REV: 1.0
Date: 2018-08-15
Sheet: 1/1
EasyEDA V5.6.15
Drawn By: Your Name


Made on Earth by humans!



1

ТК гайки из (Бронзового прутка БрАЖ9-4 20мм)

Бронза марки БрАЖ9-4 зарекомендовала себя как отличный материал для подшипников скольжения, которым предстоит работать на высоких скоростях и с высокими ударными нагрузками.

№	Название Технологической операции	Инструменты и приспособления
1	Выбор материала	Штангенциркуль,

		линейка
2	Разметка	Штангенциркуль, линейка, чертилка.
3	Резка заготовки	Тиски, Ножовка по металлу
4	Черновое точение	Токарный станок по металлу, проходной резец
5	Сверление	Токарный станок, сверло с конусом Морзе
6	Разметка	Штангенциркуль
7	Резка	Отрезной резец, токарный станок
8	Снятие лысок	Напильник
9	Снятие фаски	Напильник
9	Нарезание внутренней резьбы	Тиски, метчик,метчикодержатель

10	Шлифование	Наждачная бумага

ТК изготовления вала

№	Название Технологической операции	Инструменты и приспособления
1	Выбор размера заготовки	Штангенциркуль, линейка
2	Разметка	Штангенциркуль, линейка, чертилка.
3	Резка заготовки	Тиски, Ножовка по металлу
4	Черновое точение	Токарный станок по металлу, проходной резец
5	Торцевание	Токарный станок, проходной резец
6	Центровка	Центровое сверло
7	Сверление	Токарный станок, сверло с конусом Морзе

8	Разметка	Штангенциркуль
9	Резка	Отрезной резец, токарный станок
10	Накернивание	Кернер
11	Сверление	Сверлильный станок, сверло с конусом Морзе
12	Снятие фаски	Токарный станок, проходной резец
13	Нарезание внутренней резьбы	Тиски, метчик, вороток
14	Нарезание внешней резьбы	Тиски, плашка, плашкодержатель
15	Шлифование	Наждачная бумага

ТК изготовления корпуса из композитного материала

№	Название технологической операции	Инструменты и приспособления
1	Моделирование	Пластилин, Стек, подставка
3	Покрытие стеклотканью	Стеклоткань, эпоксидная

	модели	смола, кисть
4	Шпатлёвка	Шпатель, эпоксидная шпатлёвка
5	Шлифовка	Наждачная бумага. плоскошлифовальная машинка
6	Покраска	Акриловая краска, кисть
7	Лакировка	Лак, кисть

ОКРУГ
ГОРОДА МОСКВЫ

Секция VIII. Энергетика и экология

Анализ работы двигателя Стирлинга

Курочкина Кристина Александровна
г. Москва, МОУ Лицей, 11 класс
Научный руководитель: Онищенко Дмитрий Олегович
Профессор кафедры «Поршневые двигатели»

Предмет: Двигатель Стирлинга

Цель работы: Посчитать выделенное количество теплоты. Сравнить работу двигателя на разном топливе.

Методы и приемы: В своем эксперименте я использовала модель двигателя Стирлинга, горючее и тепловизор. Первая часть работы заключается в том, что я проверяю, как работает двигатель Стирлинга на таком горючем, как парафиновая свеча. Здесь мы должны обратить внимание на продолжительность вращения коленчатого вала, другими словами, в течение какого времени двигатель будет работать. Также, обращаем внимание на его угловую скорость. С помощью тепловизора можно определить температуру. Таим же образом, анализирую работу двигателя на сухом спирте. Сравниваем два горючих вещества между собой. Находим количество теплоты, выделяющее в ходе работы.

Полученные результаты: $Q_2 = 0{,}1456$МДж- количество теплоты , выделившееся в ходе работы двигателя на парафиновой свече. $Q_1 = 0{,}5008$МДж- на сухом спирте.

$$Q_1 > Q_2$$

Время работы , $t_1 > t_2$

$\omega_1 > \omega_2$- скорость вращения на сухом спирте, так же превышает скорость вращения на парафиновой свече.

Содержание

1. Теоретическое введение.
   1.1 Двигатель внутреннего сгорания.
   1.2 Двигатель внешнего сгорания.
2. Конструкция двигателя Стирлинга
   2.1 Возрождение двигателя Стирлинга.
   2.2 Перспективы развития двигателей Стирлинга.
   2.3 Определение КПД.
   2.4 Классификация двигателей Стирлинга.
   2.5 Распределение температур и тепловой баланс двигателя Стирлинга
3. Анализ термодинамических циклов.
   3.1 Цикл Карно.
   3.2 Цикл Стирлинга.
   3.3 Цикл Отто.
   3.4 Цикл Тринклера.
   3.5 Цикл Дизеля.
4. Описание данной работы и её условий.
5. Методика эксперимента.
6. Результаты данного опыта.
7. Вывод.

Основные термины и определения.

Двигатель — устройство, преобразующее какой-либо вид энергии в механическую работу.

Двигатель внутреннего сгорания-тепловой двигатель, в котором топливо сгорает непосредственно в рабочей камере двигателя.

Двигатели внешнего сгорания — двигатель, где источник тепла или процесс сгорания топлива отделены от рабочего тела.

Двигатель Стирлинга — тепловая машина, в которой рабочее тело, в виде газа или жидкости, движется в замкнутом объёме, разновидность двигателя внешнего сгорания.

Поршневой двигатель — двигатель внутреннего сгорания, в котором тепловая энергия, образовавшаяся в результате сгорания топлива в замкнутом объёме, преобразуется в механическую работу поступательного движения поршня за счёт расширения рабочего тела (газообразных продуктов сгорания топлива) в цилиндре, в который вставлен поршень.

Газотурбинный двигатель (ГТД) — тепловой двигатель, в котором газ сжимается и нагревается, а затем энергия сжатого и нагретого газа преобразуется в механическую работу на валу газовой турбины.

Паровой двигатель-это устройство, которое превращает энергию пара в механическую работу.

Теоретическое введение.

Тепловыми машинами в термодинамике называются устройства, осуществляющие взаимное преобразование теплоты и работы в циклическом процессе. По своему назначению они подразделяются на 1) тепловые двигатели (цель– получение работы из теплоты), 2) холодильные машины (отбор теплоты от резервуара за счет совершения работы) и 3) разновидность холодильных машин – тепловые насосы (перенос тепла от одного резервуара в другой с большей температурой). В основе действия тепловой машины лежит круговой процесс (термодинамический цикл), совершаемый рабочим телом (газом, водяным паром и др.).

Рассмотрим далее только тепловые двигатели, поскольку холодильная машина и тепловой насос получаются обращением направления цикла двигателя.

Двигатель — устройство, преобразующее какой-либо вид энергии в механическую работу. Все двигатели делятся по устройству на двигатель внешнего сгорания, к ним относятся: поршневые паровые двигатели, паровые турбины, двигатели Стирлинга, паровой двигатель и внутреннего сгорания: газотурбинный двигатель, поршневой и роторный двигатели.

Двигатель внутреннего сгорания.

Работа двигателя внутреннего сгорания базируется на превращении тепловой энергии, образующейся в результате сгорания топлива, в механическую энергию, которая и применяется для привидения автомобили в движение. При этом двигатель включает в себя следующие агрегаты, детали, узлы: головка блока цилиндров, блок цилиндров, поршни, поршневые кольца, поршневые пальцы, шатуны,

коленчатый вал, маховик, распределительный вал с кулачками, клапана, свечи зажигания.

В двигателях внутреннего сгорания источником тепла является химическая энергия топлива, а его сгорание происходит внутри двигателя. Поэтому для таких двигателей не требуется котел или какой-то другой внешний нагреватель. Рабочим телом теоретически могут служить многие горючие вещества, однако практически все современные двигатели такого рода работают на бензине или дизельном топливе.

Тепловые циклы. Рабочий цикл любого двигателя внутреннего сгорания имеет четыре стадии: топливовоздушная смесь подается в цилиндр, затем она сжимается, сжигается, и, наконец, отработанные газы удаляются из цилиндра. После этого новый цикл начинается с подачи свежей порции смеси топлива и воздуха. В дизельных двигателях топливо и воздух подаются в рабочий цилиндр раздельно, но в остальном цикл тот же.

Автомобили малого и среднего класса оборудуются обычно четырехцилиндровыми двигателями внутреннего сгорания. Именно такими моторами оснащались «Москвичи» и «Жигули» – самые известные представители советского автопрома. Машины среднего и большого класса могут оснащаться и шести-, и восьми-, и двенадцатицилиндровыми моторами. Здесь прослеживается следующая закономерность: чем больше цилиндров – тем мощнее мотор, но, с другой стороны, и тем больше топлива он будет расходовать.

Преимущества:

1. Высокая дальность передвижения на одной заправке.
2. Малый вес и объем источника.

3. Автономность.
4. Умеренная стоимость.
5. Быстрый старт.
6. Возможность использования нескольких видов топлива.

Недостатки:

1. Низкий средний коэффициент полезного действия во время эксплуатации.
2. Высокое загрязнение окружающей среды.
3. Очень шумный.
4. Небольшой ресурс.

Двигатель внутреннего сгорания является основным видом автомобильного двигателя.

Двигатель внешнего сгорания.

Двигатель внешнего сгорания представляет собой тепловой двигатель, в котором происходит сжатие внутренней рабочей жидкости и ее нагревание за счет сгорания топлива через стенку двигателя или в теплообменнике. Далее жидкость расширяется и, действуя на механизм двигателя, приводит его в движение. Паровые двигатели и двигатели Стирлинга – наиболее известные двигатели внешнего сгорания.

Преимуществом двигателей внешнего сгорания перед двигателем внутреннего сгорания является их совместимость с широким спектром возобновляемых источников энергии и топлива. Они могут использовать тепло от любого источника: биомассы, продуктов, полученных из биомассы, бытовых отходов, ядерных, солнечных, геотермальных источников или экзотермических реакций без горения. Кроме того, важными преимуществами двигателей внешнего сгорания являются: низкий уровень выбросов благодаря непрерывному

внешнему сгоранию и низкий уровень шума благодаря удалению отработанных газов продуктов сгорания под высоким давлением. Перспективными двигателями внешнего сгорания являются двигатели Стирлинга, преобразующие тепловую энергию в механическую энергию поршней совершающих возвратно-поступательные движения.

Преимущества:

1.бесшумная работа (нет необходимости устанавливать глушитель);

2.отсутствие вибраций;

3.нет необходимости в создании высокого давления в системе;

4.универсальность, способность работать от различных источников тепла;

5.легкость регулировок.

К недостаткам:

1.сравнительно большой вес конструкции;

2.малая экономичность;

3.высокая себестоимость механизма.

Конструкция двигателя Стирлинга.

Возрождение двигателя Стирлинга.

Этот двигатель был изобретен и запатентован в 1816 г. Робертом Стирлингом, министром по делам церкви Шотландии, который долгие годы активно занимался разработкой этих двигателей, как альтернативой паровым машинам. Роберт Стирлинг был весьма обеспокоен травматизмом рабочих, работающих в его приходе с паровыми двигателями. Эти двигатели часто взрывались из-за низкого качества металла, из которого они изготавливались. Более прочного материала в те годы не существовало. Стирлинг решил усовершенствовать конструкцию теплового двигателя сделав его более безопасным.

Стирлинг придумал устройство, которое он назвал «эконом тепла» (сейчас такое устройство называют регенератором или теплообменником). Это устройство служит для повышения тепловой эффективности различных процессов. Стирлинг получил патент на двигатель с «экономом тепла» в 1816 году. Двигатель Стирлинга не может взорваться, потому что работает при более низком давлении, чем паровая машина, и не может причинить ожоги паром. В 1818 он построил первый практичный вариант своего двигателя и использовал его в насосе для откачки воды из карьера.

Теоретических основ работы двигателя Стирлинга — цикл Стирлинга — не существовало до тех пор, пока не появились работы Сади Карно. Карно разработал и опубликовал в 1825 году общую теорию работы тепловых двигателей — Цикл Карно, из которой цикл Стирлинга строится аналогичным образом.

Возрождение интереса к двигателям Стирлинга обычно ассоциируется с деятельностью фирмы «Филипс». Работы по

конструированию двигателей небольшой мощности Стирлинга начались там в середине 30-х гг. XX в. Целью работ было создание небольшого с низким уровнем шума электрического генератора с тепловым приводом для питания радиоаппаратуры в тех районах мира, в которых отсутствовали регулярные источники электроснабжения. Предполагают, что выбор между паровым двигателем и двигателем Стирлинга последовал после случайного посещения Музея техники в Париже одним из технических директоров лаборатории, когда он увидел некоторые старые двигатели, работающие на нагретом воздухе; директор справедливо предположил, что современные материалы и технология могут улучшить характеристики этих двигателей до значений, неосуществимых для моделей более раннего периода.

Работы по двигателям Стирлинга продолжались также и в 40-е гг. Впервые информация о них появилась в первый послевоенный год. Блестящие научные и инженерные исследования вылились в разработку двигателей, технические характеристики которых превосходили технические характеристики двигателей на нагретом воздухе, выпускаемых ранее.

В американская фирма «Дженерал Моторс» из г. Детройта заключила лицензионное соглашение с фирмой «Филипс», продолжавшееся до 1970 г. В работе Персиваля (1974 г.) 1958 г. приведены весьма интересные подробности о разработках фирмы «Дженерал Моторс» в этот период. Разработки были связаны с использованием двигателей Стирлинга для космических и подводных энергетических установок, автомобилей и судов, а также для систем стационарного энергоснабжения.

Вслед за отказом фирмы «Дженерал Моторс» от прав на лицензионное соглашение с фирмой «Филипс» лицензии в 1971 г. приобрела фирма «Форд Мотор» (г. Детройт).

Исследования двигателей Стирлинга для солнечных, космических и подводных энергетических установок, а также разработка базового лабораторного опытного двигателя в настоящее время проводятся в лаборатории реактивного движения Калифорнийского технологического института в Пасадене (шт. Калифорния).

Много новых и оригинальных разработок по теории и практике двигателей Стирлинга появляется и во все возрастающем числе промышленных фирм, университетов и правительственных исследовательских учреждений в странах Северной Америки, Европы и других странах.

Перспективы развития и применение двигателей Стирлинга.

В настоящее время будущее двигателей Стирлинга представляется более перспективным. В обзорах по различным двигательным установкам для автомобилей двигатель Стирлинга рассматривают как двигатель, обладающий наибольшими возможностями для дальнейшей разработки. Низкий уровень шума, малая токсичность отработавших газов, возможность работы на различных топливах, большой ресурс, сравнимые размеры и масса, хорошие характеристики в режимах частичной нагрузки и благоприятные характеристики крутящего момента — все эти параметры дают возможность «бросить вызов» ДВС. Можно гарантировать использование двигателей Стирлинга для стационарных энергетических систем в широком диапазоне мощностей. Очевидно, что эти двигатели найдут более широкое применение в тепловых насосах и холодильных системах.

Двигатель Стирлинга можно использовать во всех областях, где требуется преобразование тепловой энергии в механическую. В самом деле, почти нельзя назвать ни одной какой-либо серьезной области

потенциального применения двигателя Стирлинга, в которой уже не было бы предпринято попытки его использования или такая возможность не изучалась. При этом нельзя выделить каких-то необычных областей применения, поскольку во всех случаях имеются альтернативные источники механической энергии аналогичного назначения.

Одной из нетрадиционных областей применения двигателя Стирлинга есть медицина. Его применяют в системах искусственного сердца. Источником энергии в таких системах, как правило, есть радиоизотопы.

Двигатель Стирлинга уже сейчас применяют в когенерационных установках. Когенерационные установки предназначены для полного использования энергии, которая высвобождается во время сгорания топлива. Часть этой энергии превращается в электроэнергию, остальная – в теплоту, которая используется для удовлетворения бытовых нужд. Т.е. двигатель производит электроэнергию, а теплота из его системы охлаждения, смазки и выпускной системы утилизируется и обеспечивает горячее водоснабжение, отопление помещений и т.п. Благодаря общему производству электрической и тепловой энергии в когенерационных установках обеспечивается значительная экономия топлива – до 30%.

Криокулер Стирлинга также работает по принципу теплового насосу, но применяется как холодильная установка для получения очень низких температур. В широких масштабах их начали изготовлять уже 15-20 лет тому – преимущественно для использования в военной технике: на танках и самолетах необходимо было устанавливать высокочувствительные датчики и приемники, охлаждаемые до температуры -200°С. Для такого охлаждения и было разработано криокулери, работа которых основывается на обратном цикле Стирлинга.

Определение КПД.

В лаборатории кафедры теплоэнергетики имеется стенд двигателя Стирлинга, который может использоваться как в учебном процессе, так и в исследовательских целях. На данном стенде были проведены испытания, с целью определения его характеристик и выбора направлений возможного усовершенствования.

Одной из задач являлось определение КПД двигателя и сопоставление его с расчетом по выражению для цикла Карно, т.е. по температурам горячей и холодной частей цилиндра, а также зависимостей частоты вращения от перепада температур горячей и холодной части цилиндра, и от подводимой тепловой энергии. Для этого необходимо определять входную (подводимую) и выходную мощность двигателя.

Изначально нагрев предусматривался от сжигания сухого горючего, однако, при этом сложно точно определить подводимую к двигателю теплоту, т.е. учесть потери. Оптимальный, на наш взгляд, является электрический нагрев горячей части с помощью, нанесенной на нее обмотки. Обмотка накладывалась на горячую часть двигателя в виде ромовой спирали толщиной 0,5 мм. Напряжение, подаваемого на обмотку, варьировалось 75-100 В. Однако, нанесенной теплоизоляции из асбестового шнура оказалось недостаточно, поверхность теплоизоляции имела температуру от 60 до 80 и часть подводимой энергии рассеивалась. По известной формуле Стефана-Больцмана определялись потери на излучение. Определение выходной мощности на валу является следующим шагом исследования. Определялись следующие зависимости: температур горячей и холодной части цилиндра, частота вращения, подводимая мощность. К сожалению, пока не определена выходная мощность, необходимая для определения КПД двигателя и сопоставления с определенным по температурам горячего и холодного цилиндра. В данный момент

определение выходной мощности находится в стадии определения и обработки. Частоту вращения замеряли с помощью стробоскопа. Измерения температур проводились с помощью термопар.

В результате выполненных измерений следующие выводы:

1.С увеличением мощности растет частота вращения.

2.С увеличением времени нагрева начинает нагреваться холодная часть двигателя.

3.Также с увеличением мощности растет вибрация двигателя.

Классификация двигателей Стирлинга.

Двигатели Стирлинга по принципу действия могут быть классифицированы на двигатели одностороннего (простого) и двойного действия.

В двигателях одностороннего действия две полости (сжатия и расширения), соединяемые теплообменниками, могут находиться в одном или в двух цилиндрах. В одноцилиндровых двигателях предусмотрены два поршня — рабочий и вытеснительный, а в двухцилиндровых — два рабочих или рабочий и вытеснительный.

Двигатели двойного действия —это многоцилиндровые двигатели, в которых полости расширения каждого цилиндра последовательно соединены через ряд теплообменников с полостью сжатия соседнего цилиндра.

Инженеры подразделяют двигатели Стирлинга на три различных вида:

α-Стирлинг — содержит два раздельных силовых поршня в раздельных цилиндрах, один — горячий, другой — холодный. Цилиндр с горячим поршнем находится в теплообменнике с более

высокой температурой, с холодным — в более холодном. У данного вида двигателя отношение мощности к объёму достаточно велико, но, к сожалению, высокая температура «горячего» поршня создаёт определённые технические трудности. Регенератор находится между горячей частью соединительной трубки и холодной.

β-Стирлинг — цилиндр всего один, горячий с одного конца и холодный с другого. Внутри цилиндра движутся поршень (с которого снимается мощность) и вытеснитель, разделяющий горячую и холодную полости. Газ перекачивается из холодной части цилиндра в горячую через регенератор. Регенератор может быть внешним, как часть теплообменника, или может быть совмещён с поршнем-вытеснителем.

γ-Стирлинг — тоже есть поршень и вытеснитель, но при этом два цилиндра — один холодный (там движется поршень, с которого снимается мощность), а второй горячий с одного конца и холодный с другого (там движется вытеснитель). Регенератор может быть внешним, в этом случае он соединяет горячую часть второго цилиндра с холодной и одновременно с первым (холодным) цилиндром. Внутренний регенератор является частью вытеснителя.

Распределение температур и тепловой баланс двигателя Стирлинга.

Поступающий в двигатель воздух имеет на входе в подогреватель температуру, равную температуре окружающей среды. В подогревателе температура воздуха повышается; и он поступает в камеру сгорания, где происходит процесс сгорания топлива, сопровождающийся высокой температурой. Из камеры продукты сгорания проходят последовательно через нагреватель и подогреватель воздуха, где осуществляется нагрев рабочего тела и подогрев поступающего в двигатель воздуха, а затем выходят в

атмосферу. Обычно, исходя из требований охраны окружающей среды, до половины всего количества отработавших газов может быть возвращено (рециркулировано) в камеру сгорания.

Рабочее тело из регенератора поступает в полость расширения, его температура в нагревателе повышается от теплоты продуктов сгорания топлива. Так как обычно рабочим телом двигателей Стирлинга служит водород или гелий при высоком давлении, следует отметить их достаточно высокие свойства переноса теплоты по сравнению со свойствами переноса теплоты продуктов сгорания, находящихся при атмосферном давлении. Поэтому при заданном тепловом потоке разность температур между внешней стенкой нагревателя и продуктами сгорания достаточно значительная, в то время как между рабочим телом и внутренней стенкой нагревателя разность температур существенно меньше.

В холодильнике теплота от рабочего тела отводится циркулирующей в двигателе водой с небольшой разностью температур на входе и выходе, что обусловлено ее высокими свойствами переноса. Затем теплота рассеивается в атмосферу через охлаждаемый воздухом радиатор, для обдува которого предусмотрен вентилятор. Циркуляция воды в двигателе производится с помощью небольшого водяного насоса.

.

Диаграмма теплового баланса одного из двигателей Стирлинга

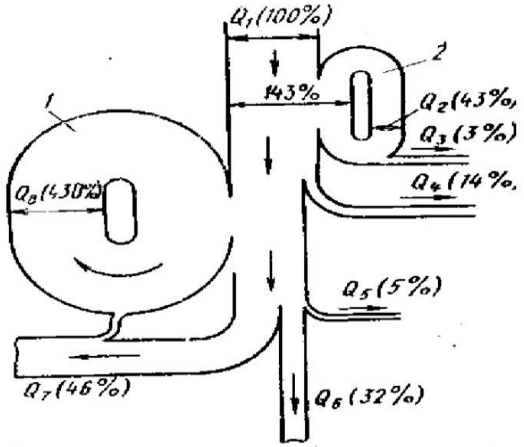


Рис.5.3. Тепловой баланс двигателя Стирлинга:

1 - регенератор; *2* - подогреватель; Q_1 -подводимая теплота топлива (100%); Q_2 -теплота рециркуляции отработавших газов (43% Q_1); Q_3 — тепловые потери в подогревателе воздуха и другие неучтенные потери теплоты(3 % Q_1); Q_4 - тепловые потери, отводимые с продуктами сгорания топлива (14 % Q_1); Q_5 - тепловые потери, эквивалентные работе сил механического трения (5 %Q_1); Q_6 — теплота, эквивалентная эффективной работе (32% Q_1); Q_7 — теплота, отводимая в систему охлаждения (46 % Q_1); Q_8 — теплота регенерации (430 %Q_1)

Подведенное количество теплоты принято за 100 %. Степень рециркуляции отработавших газов составляет 43 %, безвозвратные потери теплоты в подогревателе — 3 %, а потери с отработавшими газами — 14 % располагаемой теплоты.

Остальная часть баланса распределена следующим образом: потери с охлаждающей водой — 46 %, механические потери —5 %; примерно 32 % теплоты превращается в полезную работу двигателя. Необходимо отметить высокий уровень теплового потока в регенераторе (430 %), превышающего более чем в 4 раза подводимую теплоту и соответственно в 9 и 12 раз тепловой поток в холодильнике и полезную работу двигателя. Особенностью двигателя Стирлинга является и значительная степень рециркуляции отработавших газов.

Анализ термодинамических циклов.

Термодинамические циклы — круговые процессы в термодинамике, то есть такие процессы, в которых совпадают начальные и конечные параметры, определяющие состояние рабочего тела (давление, объём, температура и энтропия)

Цикл Карно.

В термодинамике цикл Карно или процесс Карно — это идеальный круговой процесс, состоящий из двух адиабатных и двух изотермических процессов. В процессе Карно термодинамическая система выполняет механическую работу за счёт обмена теплотой с двумя тепловыми резервуарами, имеющими постоянные, но различающиеся температуры. Пусть тепловая машина состоит из нагревателя с температурой Тн , холодильника с температурой Тх и рабочего тела.

Цикл Карно состоит из четырёх обратимых стадий, две из которых осуществляются при постоянной температуре (изотермически), а две

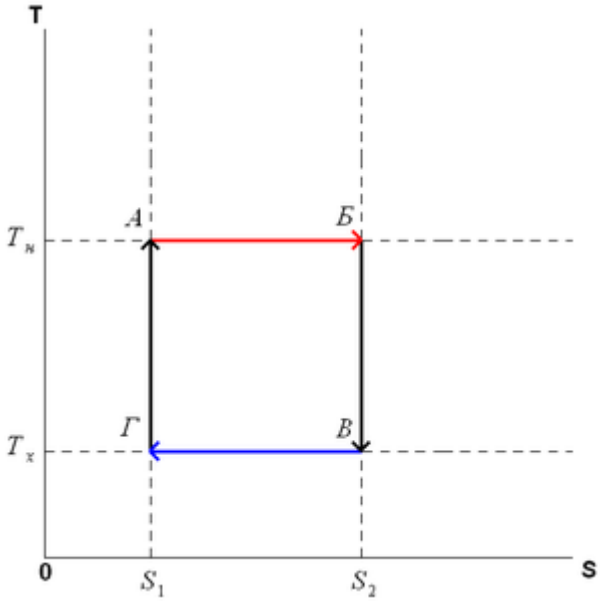
— при постоянной энтропии (адиабатически). Поэтому цикл Карно удобно представить в координатах T (температура) и S (энтропия).

1. Изотермическое расширение (процесс А→Б). В начале процесса рабочее тело имеет температуру Тн, то есть температуру нагревателя. Затем тело приводится в контакт с нагревателем, который изотермически (при постоянной температуре) передаёт ему количество теплоты Qн. При этом объём рабочего тела увеличивается, оно совершает механическую работу, а его энтропия возрастает.

2. Адиабатическое расширение (процесс Б→В). Рабочее тело отсоединяется от нагревателя и продолжает расширяться без теплообмена с окружающей средой. При этом температура тела уменьшается до температуры холодильника Тх тело совершает механическую работу, а энтропия остаётся постоянной.

3.Изотермическое сжатие (процесс В→Г). Рабочее тело, имеющее температуру Тх приводится в контакт с холодильником и начинает изотермически сжиматься под действием внешней силы, отдавая холодильнику количество теплоты Qх.Над телом совершается работа, его энтропия уменьшается.

4. Адиабатическое сжатие (процесс Г→А). Рабочее тело отсоединяется от холодильника и сжимается под действием внешней силы без теплообмена с окружающей средой. При этом его температура увеличивается до температуры нагревателя, над телом совершается работа, его энтропия остаётся постоянной.

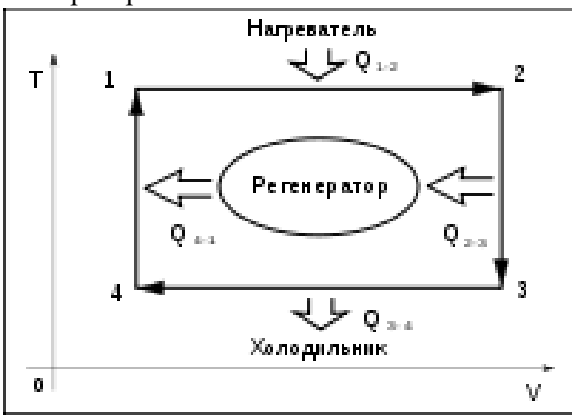


Цикл Стирлинга.

Цикл Стирлинга — термодинамический цикл, описывающий рабочий процесс машины Стирлинга, запатентованной в 1816 г. шотландским изобретателем Робертом Стирлингом, приходским священником по профессии. Помимо рабочего тела, нагревателя и холодильника абстрактная машина Стирлинга содержит ещё регенератор — устройство, отводящее тепло от рабочего тела на некоторых этапах цикла, и отдающее это тепло рабочему телу на других этапах. Идеальный цикл Стирлинга состоит из процессов:

- 1—2 изотермическое расширение рабочего тела с подводом тепла от нагревателя;
- 2—3 изохорный отвод тепла от рабочего тела к регенератору;

- 3—4 изотермическое сжатие рабочего тела с отводом тепла к холодильнику;
- 4—1 изохорический нагрев рабочего тела с подводом тепла от регенератора.

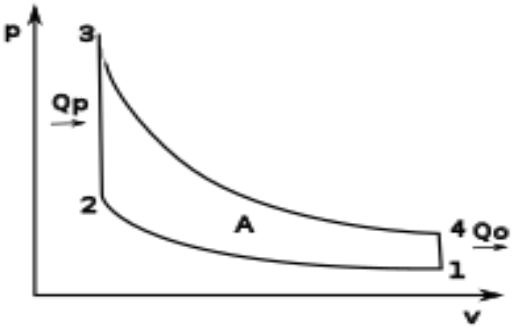


Цикл Отто.

Цикл Отто — термодинамический цикл, описывающий рабочий процесс двигателя внутреннего сгорания с воспламенением сжатой смеси от постороннего источника энергии, цикл бензинового двигателя. Назван в честь немецкого инженера Николауса Отто.

Идеальный цикл Отто состоит из четырёх процессов:

- 1—2 адиабатное сжатие рабочего тела;
- 2—3 изохорный подвод теплоты к рабочему телу;
- 3—4 адиабатное расширение рабочего тела;
- 4—1 изохорное охлаждение рабочего тела



КПД цикла Отто $\eta = 1 - \frac{1}{n^{k-1}}$,

где $n = {}^{V_1}/_{V_2}$,

k — показатель адиабаты.

Идеальный цикл лишь приблизительно описывает процессы, происходящие в реальном двигателе, но для технических расчётов в большинстве случаев точность такого приближения удовлетворительна.

Цикл Тринклера.

Цикл Тринклера — термодинамический цикл, описывающий рабочий процесс дизельного двигателя со смешанным сгоранием. Объединяет в себе цикл Отто и цикл Дизеля. Носит имя своего изобретателя Густава Тринклера.

Идеальный цикл Тринклера состоит из процессов:

- 1—2 В рабочем цилиндре воздух адиабатически сжимается за счет инерции маховика, сидящего на валу двигателя, нагреваясь при этом до температуры, обеспечивающей воспламенение топливно-воздушной смеси.
- 2—3 Сгорание части топлива в небольшом объеме форкамеры (V=const).

- 3—4 Догорание оставшегося топлива в рабочем цилиндре (P=const).
- 4—5 Адиабатическое расширение продуктов сгорания.
- 5—1 Удаление выхлопных газов (V=const).

Жидкое топливо, введенное в форкамеру при сравнительно невысоком давлении, распыляется струей сжатого воздуха, поступающего из основного цилиндра. Вместе с тем цикл со смешанным сгоранием частично сохраняет преимущества цикла Дизеля перед циклом Отто — часть процесса сгорания осуществляется при постоянном давлении.

Термический КПД цикла Тринклера $\eta = 1 - \frac{1}{n^{k-1}} \frac{\lambda m^k - 1}{\lambda - 1 + k\lambda(m-1)}$

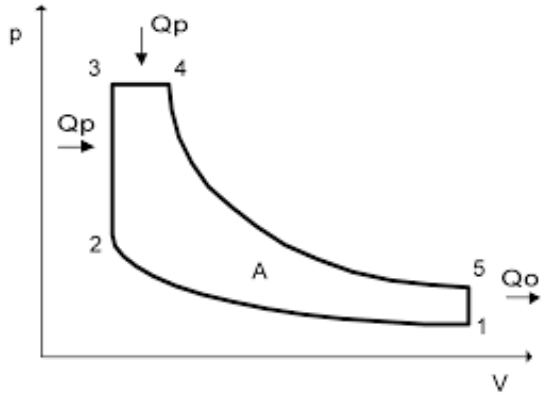
Где $n = {V_1}/{V_2}$ — степень сжатия

$m = {V_4}/{V_3}$ — степень предварительного расширения,

$\lambda = {p_3}/{p_2}$ — степень повышения давления при изохорном процессе сгорания,

k — показатель адиабаты.

Частными случаями цикла Тринклера являются цикл Отто (при m=1) ицикл Дизеля (при λ =1).



Цикл Дизеля.

Цикл Дизеля — термодинамический цикл, описывающий рабочий процесс двигателя внутреннего сгорания с воспламенением впрыскиваемого топлива от разогретого рабочего тела (сжатого поршнем воздуха), цикл дизельного двигателя.

Идеальный цикл Дизеля состоит из четырёх процессов:

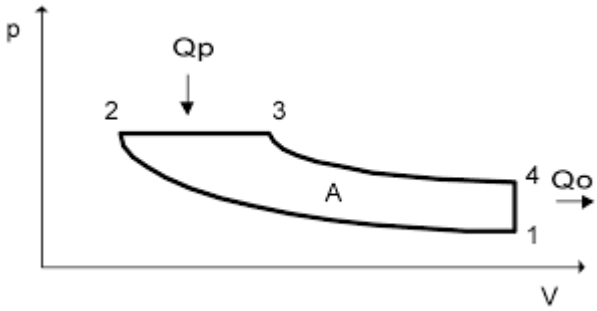
- 1—2 адиабатное сжатие рабочего тела;
- 2—3 изобарный подвод теплоты к рабочему телу;
- 3—4 адиабатное расширение рабочего тела;
- 4—1 изохорное охлаждение рабочего тела.

КПД цикла Дизеля $(= 1 - \frac{1}{к}(\frac{m^k - 1}{m - 1})\frac{1}{n^{k-1}}$

Где $m = {V_3}/{V_2}$ — коэффициент предварительного расширения,

k — показатель адиабаты.

Идеальный цикл лишь приблизительно описывает процессы, происходящие в реальном двигателе, но для технических расчётов в большинстве случаев точность такого приближения



удовлетворительна.

Экспериментальная установка.

В своем эскперименте я использую модель двигателя Стирлинга. Возьму первое горючее вещество- парафиновую свечу. Масса такой свечи равна 13г=0.013кг. Удельная теплота сгорания парафина равна 11,2 МДж/кг. Поджигаем горючее вещество, нагреваем рабочее тело двигателя. Коленчатый вал начинает вращаться. Угловая скорость двигателя начинает возрастать. Второе горючее вещество - сухой спирт. Масса одной таблетки равна 16г=0,016 кг. Удельная теплота сгорания сухого спирта равна 31,3 МДж/кг. Точно так же поджигаем горючее вещество, коленчатый вал начинает вращаться, но уже с другой скоростью.

Методика эксперимента.

Для чего нам нужны масса и удельная теплота сгорания ? С помощью этих данных можно найти выделенное количество теплоты (Q).

В первом случае берем парафин тогда,

$Q_2 = qm = 11{,}2 \times 0{,}013 = 0{,}1456$МДж – количество теплоты.

Возьмем сухой спирт,

$Q_1 = qm = 31{,}3 \times 0{,}016 = 0{,}5008$МДж – количество теплоты.

Вывод:

$$Q_1 > Q_2$$

Сравниваем результаты, полученные в ходе эксперимента.

Сухой спирт	Парафиновая свеча
Q=0,5008 МДж	Q=0,1456 МДж
$\omega_1 > \omega_2$	$\omega_1 > \omega_2$
$t_1 > t_2$	$t_1 > t_2$

Экспериментальная установка показала, что скорость вращения коленчатого вала двигателя, работающего на таком горючем

веществе, как сухой спирт, намного выше скорости двигателя, работающего на веществе- парафин.
Если обратить внимание на время вращения вала, то так же можно сделать вывод, что время работы двигателя на парафиновой свече меньше времени работы двигателя на сухом спирте.

Тепловые двигатели отличаются от других типов двигателей (электрических и др.) тем, что для всех них коэффициент полезного действия (КПД) имеет фундаментальное ограничение, устанавливаемое теоремами Карно, которые, в свою очередь, базируются на II начале термодинамики. Смысл второго начала термодинамики в формулировке Томсона (Кельвина) состоит в том, что невозможен циклический тепловой процесс, единственным результатом которого было бы производство работы за счет охлаждения (т.е. уменьшения внутренней энергии) теплового резервуара. Поскольку всю теплоту невозможно превратить в работу, часть ее должна быть отдана телу с меньшей температурой.
Если при осуществлении цикла теплового двигателя на одних его участках теплота Q+ подводится к рабочему телу, а на других отводится теплота Q- (при более низкой температуре), то рабочее тело совершает работу, равную (для идеальной тепловой машины) разности количеств подведенной и отведенной теплоты $A = Q_+ - Q_-$. Здесь количества теплоты берутся по модулю, а направление передачи учитывается знаком в уравнениях. Коэффициентом полезного действия (КПД) теплового двигателя называется отношение произведённой работы к подведённому извне количеству теплоты:

$$\eta = \frac{A}{Q+} = 1 - \frac{Q-}{Q+}$$

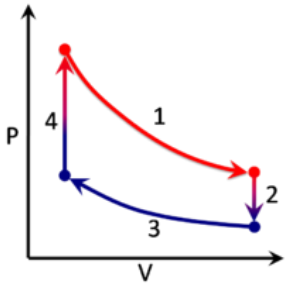
Теоремы Карно утверждают, что КПД η двигателя с любым термодинамическим циклом, проходящем в температурном интервале T2 ≤ T≤T1 , принципиально не может превысить максимального

значения

$$\eta_{\max} = 1 - \frac{T_2}{T_1}$$

которое, в свою очередь, теоретически реализуется только в обратимом цикле Карно, состоящем из двух изотерм при температурах T1 и T2, и двух адиабат (δQ = 0)

Коэффициент полезного действия цикла Стирлинга.



Обозначим отношение крайних объемов $n = \frac{V_2}{V_1}$, тогда, ввиду изотермичности процесса 1-2, $p_2 = \frac{p_1 V_1}{V_2} = \frac{p_1}{n}$

(1–2) Изотермическое расширение. Внутренняя энергия рабочего тела не изменяется, работа равна подводимой теплоте $Q_{12} = Q_+ = A_1$, энтропия увеличивается $\Delta S_{12} > 0$

(2–3) Изохорное охлаждение. Теплота уходит от рабочего тела – температура тела уменьшается от T_1 до T_2. Работа не производится (A23 = 0); внутренняя энергия и энтропия уменьшаются $Q_{23} <$

0 $\Delta S_{23} > 0$
(3-4) Изотермическое сжатие. Внутренняя энергия не изменяется. Работа равна отданной теплоте, энтропия уменьшается.

$$Q_{34} = A_{34} < 0 \qquad \Delta S_{34} < 0$$

(4–1) Изохорный нагрев. Работа в процессе не производится (A23 = 0). Теплота приходит к рабочему телу – температура тела увеличивается от T2 до T1. Внутренняя энергия и энтропия рабочего тела увеличиваются

$$Q_{41} > 0 \quad \Delta S_{41} > 0$$

Полная работа за цикл

$$A = Q_{12} + Q_{34}$$

Полученное рабочим телом тепло

$$Q_{+} = Q_{12} + Q_{41}$$

Коэффициент полезного действия

$$\eta = \frac{A}{Q_{+}}$$

Эта величина меньше КПД цикла Карно η_k.

Однако цикл Стирлинга обладает одной интересной особенностью, позволяющей увеличить его КПД. Как видно из вышеприведенных формул, количество теплоты, бесполезно отдаваемое на участке изохорного охлаждения (2-3), равно по модулю теплоте, требуемой на участке изохорного нагрева (4-1): $Q_{41} = -Q_{23}$

Охлаждение и нагрев газа на этих участках цикла происходит в процессе физического вытеснения газа из горячей части цилиндра в холодную и обратно. Поэтому на пути газа можно поставить регенератор, то есть устройство, аккумулирующее тепло проходящего через него горячего газа на участке 2-3, и отдающее потом это тепло

холодному газу на обратном переходе на участке 4-1. В этом случае суммарный теплообмен цикла с внешними источниками будет происходить только на изотермических участках. КПД станет равным $\eta = 1 - \frac{T_2}{T_1}$, то есть формально сравняется с циклом Карно. К сожалению, он будет все же существенно меньше, потому как процессы теплообмена в регенераторе неравновесны и необратимы, и таким же будет и весь цикл. Однако при одинаковых интервалах изменения давления и объемов цикл Стирлинга даст большую работу (хоть и с меньшим КПД), чем цикл Карно.

Одной из причин, снижающих кпд двигателя Стирлинга, является внешний теплообмен с нагревателем и холодильником. В двигателе внутреннего сгорания топливо сгорает внутри рабочего газа и полностью и непосредственно передает ему тепло сгорания. В двигателе Стирлинга теплообмен рабочего газа и источников тепла и охлаждения происходит через стенки цилиндров за счет их теплопроводности, и это приводит к бесполезной потере энергии.

Выводы.

Двигатель Стирлинга — тепловая машина, работающая не только от сжигания топлива, но от любого источника тепла, например — солнечных лучей.

Исключительное свойство двигателей Стирлинга, что разрешает применять нетрадиционные топлива, например, биогаз, уголь и даже отходы деревообрабатывающей промышленности, а также использование любых других видов энергии делает их особенно привлекательными в связи с использованием энергии из возобновляемых источников.

Двигатель Стирлинга имеет преимущества:

1. Простота конструкции;
2. Экономичность;

3. Экологичность;
4. Многообразие топлива;
5. Увеличенный ресурс.

Недостатки:

1. Громоздкость и материалоёмкость;
2. Плохая теплопроводность;
3. Принцип перехода двигателя Стирлинга на другие режимы функционирования.

Я проверила работу двигателя на разном горючем веществе: сухой спирт, парафиновая свеча. Выявила на каком горючем веществе двигатель вращается с большей скоростью, большим временем. Это оказался сухой спирт. Он преобладает над парафиновой свечей.

Секция IX. Автоматизация, робототехника и механика

Автоматическая линия производства вязких лекарственных препаратов

Сафронов Пётр Андреевич
г. Москва, ГБОУ Школа №1502 при МЭИ, 11 класс
Научный руководитель: Воротников Сергей Анатольевич, к.т.н., Доцент кафедры РК-9 МГТУ им. Н.Э. Баумана

На данный момент фармацевтическая промышленность в значительной степени автоматизирована, но при этом автоматизация малотоннажных и опытных производств лекарственных препаратов практически не осуществлена. Одной из причин слабого развития автоматизации таких производств является проблема качественного перемешивания вязких жидкостей, наряду с этим существует проблема точной дозировки их по таре малого объёма. В данной работе выполняется проектирование малотоннажной автоматической линии, в ходе которого отдельное внимание уделяется решению обозначенных выше проблем.

Объектом данной проектной работы является автоматическая линия розлива вязкого лекарственного препарата "Люголь" (на основе глицерина). В ходе работы было проведено

исследование, направленное на разработку дозатора, который выполняет розлив препарата в тару малого объема. Для обеспечения качественного изготовления препарата, была разработана мешалка, осуществляющая приемлемое перемешивание составляющих данного лекарства. Дополнительно, было выполнено эскизное проектирование автоматической линии розлива.

Разработанная установка состоит из четырёх основных узлов: транспортера, ёмкости с мешалкой, дозатора и узла укупора флаконов с препаратом. Автоматическая линия включает в себя пневматическую и электрическую системы и управляется при помощи программируемого логического модуля **Siemens LOGO 8**. В работе также предусматривается моделирование отдельных узлов и создание макета данной линии розлива.

СОДЕРЖАНИЕ

Аннотация к проекту

Перечень принятых сокращений

1. Введение

Актуальность

Цели и задачи проекта

Техническое задание

2. Исследовательская часть

Обзор аналогов

Структура автоматической линии

Особенности работы с вязкими жидкостями

Разработка перемешивающего устройства

Разработка дозатора

3. Конструкторская часть

Детализированная функциональная система

Выбор приводных механизмов

Конструирование узлов линии

Разработка электрической и пневматической схем

4. Система управления автоматической линией

Структура системы управления

Алгоритм работы автоматической линии

Управляющий блок

Управляющая программа

Моделирование управления линией

Заключение

Список используемых источников

Приложения

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

- АЛР – автоматическая линия розлива
- ДПТ – двигатель постоянного тока;
- ПЛМ – программируемый логический модуль;
- ПР – пневмораспределитель;
- ПС – пневматическая система;
- ПЦ – пневмоцилиндр;
- ПЦР – пневмоцилиндр узла розлива;
- ПЦУ – пневмоцилиндр узла укупорки;
- САПР – система автоматизированного проектирования;
- САУ – система автоматического управления;
- ТЗ – техническое задание;
- ШВП – шарико-винтовая передача;
- ЭС – электрическая система.

1. ВВЕДЕНИЕ

Актуальность

На данный момент автоматизация процесса розлива развита во многих сферах производства: химической, нефтяной, пищевой и многих других. Однако, существуют области, в которых автоматизация процесса розлива нуждается в доработке, в частности фармацевтическая промышленность.

Одной из причин слабо развитой автоматизации таких производств является проблема качественного перемешивания вязких жидкостей, другой – техническая сложность точной дозировки их по таре малого объёма.

Данная сфера промышленности, больше чем другие, требует высокого качества производимой продукции. С каждым годом требования к ней растут, а также появляются новые лекарственные средства [1]. Для крупных производств автоматизация производства препаратов достаточно широко развита, но полная автоматизация малотоннажных и опытных производств медицинских средств практически не осуществлена [2]. Также автоматизированные линии не имеют широкого распространения в сфере малого бизнеса. Например, в большой части производств, находящихся при частных аптеках, розлив и укупор флаконов с препаратами производится вручную. По всем этим причинам фармацевтическая промышленность нуждается в

доработке автоматизированных линий, а также автоматизации опытных и малотоннажных производств.

В проекте рассматривается разработка автоматической линии малотоннажного розлива и укупора вязких лекарственных препаратов, в частности антисептического средства «Люголь». Данный медицинский препарат состоит из глицерина, как вспомогательного вещества, и действующего начала (действующими веществами в вышеуказанном лекарственном средстве являются ***I2*** и ***KI*** в растворе).

Цели и задачи проекта

Цель работы - создание автоматической линии розлива вязких медицинских препаратов, в частности антисептического средства «Люголь».

Задачи:

1. Определение структуры автоматической линии;
2. Разработка мешалки;
3. Разработка дозатора;
4. Конструирование автоматической линии;
5. Разработка алгоритма управления системой.

Техническое задание

Данная установка производства вязких лекарственных препаратов должна удовлетворять параметрам фармакологических производств. По исходным данным и условиям было составлено ТЗ.

Технические требования:

- Полная автоматизация технологических процессов линии;
- Обеспечение укупора флаконов с препаратом;
- Не допускается протечка лекарственного препарата;
- Обеспечить розлив вязких жидкостей;

Исходные данные:

- Минимальная производительность – 450 шт./ч;
- Обеспечить розлив в диапазоне от 10 до 100 мл;
- Объем стандартного флакона для препарата - 25 мл;
- Диаметр стандартного флакона–37 мм;
- Высота стандартного флакона – 75 мм.

1. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

Обзор аналогов

Автоматические или частично автоматизированные линии часто применяются в разных областях промышленности, поэтому на рынке существует ряд готовых линий. Для определения рациональной структуры и функционала автоматической линии для малотоннажного производства был проведен обзор аналогов.

Обзор аналогов автоматизированных линий

Для определения структуры АЛР был проведен обзор аналогичных готовых линий розлива вязких жидкостей. Анализ проводился по следующим критериям: производительность установки и её цена. Также следует учесть, что установка должна иметь узел укупора, т.к. недопустимо попадание инородных тел во флакон с препаратом.

Таблица 1. Обзор аналогов автоматизированных линий.

Название	Производительность, шт./ч	Цена, руб.	Примечания
Триблок розлива DCGF 18-18-6[3]	2000-5000	3 000 000	Линейный розлив: Есть узел укупора; 12-32 сопла

Полуавтомат розлива ПС-5[4]	450	148 000	Линейный розлив; Нет узла укупора; 2 сопла
Полуавтомат розлива Альтер-04[5]	500	400 000	Линейный розлив; Нет узла укупора; 1 сопло
Автоматическая линия розлива в ПЭТ[6]	2000	8 000 000	Револьверный розлив; Есть узел укупора

На рынке существует большое количество разнообразных автоматических или частично автоматизированных линий розлива вязких жидкостей. Некоторые из них не имеет узла укупора, а та часть, которая имеет данный узел, обладает высокой, недоступной для малого бизнеса, стоимостью (оптимальная стоимость установок для частного малообъемного производства, по экспертной оценке, должны составлять не более 1 500 000 рублей). Также следует учесть производительность обозреваемых установок. Некоторые линии обладают высокой производительностью, однако установки, подходящие для малого бизнеса по стоимости, низкоэффективны.

Структура автоматической линии

На основании данных, полученных при обзоре аналогов, было выявлено, что в производстве чаще всего применяются линейный и револьверный (роторный) типы розлива. Для малотоннажных производств револьверный розлив применяется реже, так как этот узел, дороже и конструктивно более сложен. Его применяют для автоматизации более крупных производств. По этой причине в проектируемой установке будет осуществлен линейный тип розлива вязких жидкостей. Так же следует учесть, что после узла розлива необходим узел укупора флаконов с препаратом.

Таким образом, для осуществления функционала линии розлива вязких жидкостей, как и у аналогичных систем розлива, установка должна включать в себя следующие части:

1. Конвейерную ленту, осуществляющую перемещение флаконов по установке;
2. Перемешивающее устройство, выполняющее приемлемое смешивание составляющих лекарственного препарата;
3. Узел розлива, выполняющий точное дозирование препарата по таре малого объема;
4. Узел укупора флаконов с препаратом, предотвращающий попадание инородных тел во флакон.

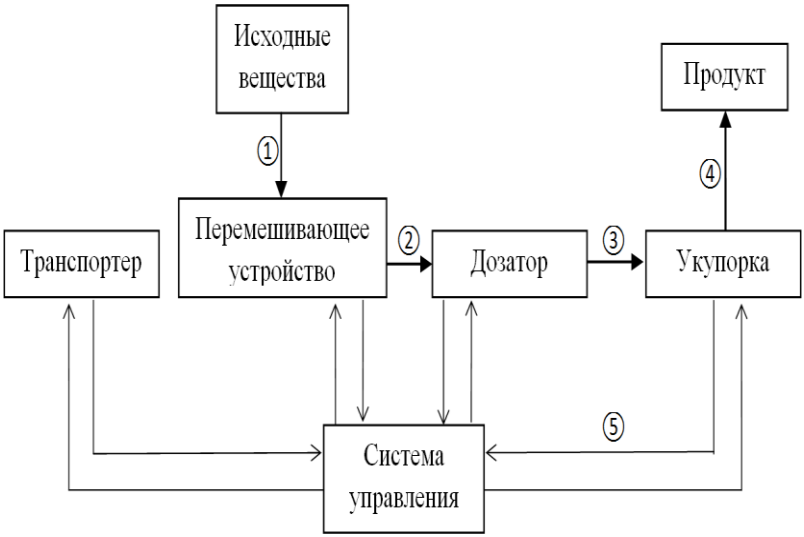


Рисунок 1 - Функциональная схема линии.

На рисунке 1 обозначена функциональная схема АЛР, где

① – поток несмешанных составляющих препарата;

② – поток готового препарата;

③ – поток флаконов с разлитым препаратом;

④ – поток готовой продукции;

⑤ – различные сигналы, необходимые для управления.

Особенности работы с вязкими жидкостями

Некоторые жидкости, такие как глицерин или тяжелые масла, обладают особенно большой вязкостью. Вязкость является свойством жидкостей оказывать сопротивление перемешиванию одной части жидкости относительно другой. Это обусловлено большим трением между слоями жидкости. По этой причине при работе с вязкими жидкостями создается ряд проблем. К примеру, при перекачивании, вязкие жидкости создают большое сопротивление. Это явление описывает закон Пуазейля:

$$Q = \frac{\Delta p \cdot r^4 \cdot \pi}{8 \cdot l \cdot \eta}$$

Q – объемный расход жидкости;

Δp – разница давлений в начале трубы и в конце;

r – радиус трубы;

l – длина трубы;

η −коэффициент динамической вязкости жидкости.

Для наглядности проблемы приведем несколько примеров значений коэффициента динамической вязкости:

Вода при 20°C	Глицерин при 40°C	Глицерин при 20°C
$\eta = 10^{-3}$ Па · с	$\eta = 330 \cdot 10^{-3}$ Па · с	$\eta = 1480 \cdot 10^{-3}$ Па · с

В законе Паузейля видно, что объемный расход обратно пропорционален вязкости, а, значит, гидравлическое сопротивление прямо пропорционально ей. Это означает, что при равных условиях работы с жидкостью, глицерин при 40°С, будет создавать в 330 раз большее сопротивление, чем вода, а глицерин при 20°С – в 1480 раз большее сопротивление, относительно воды. Большое гидравлическое сопротивление вызывает проблемы на производствах, так как создает повышенную нагрузку на компоненты оборудования и снижает производительность систем, и, как следствие, установки, работающие с вязкими жидкостями, имеют значительно большую стоимость.

При проектировании данной установки были выявлены две основные проблемы, связанные с большой вязкостью, используемой в препарате жидкости: сложность качественного перемешивания всех составляющих препарата и сложность их точного дозирования. Таким образом, узлами установки, которые требуют максимальной проработки, станут перемешивающее устройство и узел розлива.

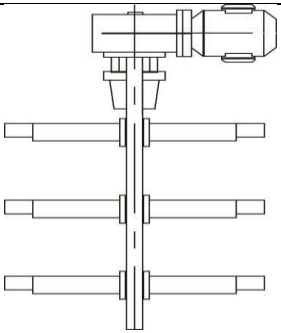
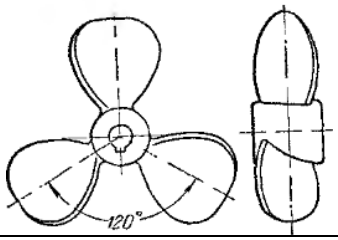
Разработка перемешивающего устройства

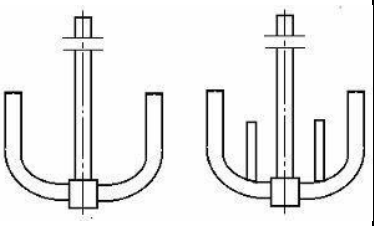
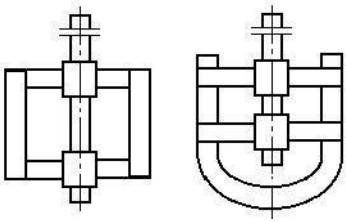
Обзор существующих решений

Для поиска оптимальной реализации перемешивания был проведен обзор существующих конструкций среди четырёх основных видов мешалок, которые рассмотрены в таблице 2 [7].

Критериями для сравнения приведенных типов мешалок стали вязкость перемешиваемой жидкости и технологическое применение каждого из типов.

Таблица 2. Обзор 4-х типов мешалок

Тип мешалки	Применение	Вязкость жидкости	Внешний вид
Лопастная	Смешивание жидкостей с различным химическим составом	Жидкости средней вязкости.	
Пропеллерная	Эмульгирование и растворение	Маловязкие жидкости	

Якорная	Качественное и максимально равномерное перемешивание без застойных зон	Жидкости высокой вязкости	
Рамная	Увеличение интенсивности теплообмена. Исключение образования осадка на стенках	Жидкости высокой вязкости	

Ввиду уникальности каждого технологического процесса, на рынке распространены только стандартные мешалки, но на производствах силами заказчика ведется их доработка. Рассматривая лопастные мешалки, количество лопастей на одном валу увеличивают, лопасти фиксируются в несколько ярусов. Якорные используют в связке с другими типами мешалок, что обеспечивает качественное смешение жидкостей без застойных зон. Для улучшения качества перемешивания жидкости и рамной мешалкой скорость вращения рамки меняют.

Препарат «Люголь» состоит из следующих химических веществ: глицерин (94%), йод (1%), йодид калия (2%), вода очищенная (3%). Основным составляющим данного препарата является глицерин, при этом препарат имеет неоднородный химический состав. Опираясь на описание каждого из видов мешалок и методы их видоизменения, можно сделать вывод, что для данной установки самым оптимальным является использование видоизмененной лопастной мешалки. На рынке существует ряд примеров лопастных мешалок с частичной доработкой [8].

В источнике [8] для усовершенствования лопастных мешалок используется модификация лопастей. Наклон лопастей в направлении центра резервуара необходим для лучшего облегания формы сосуда (Рисунок 2а). Поворот плоскости лопастей

обеспечивает появление осевого потока, благодаря которому происходит интенсивное смешивание вязких жидкостей между верхним и нижним слоями (Рисунок 2б).

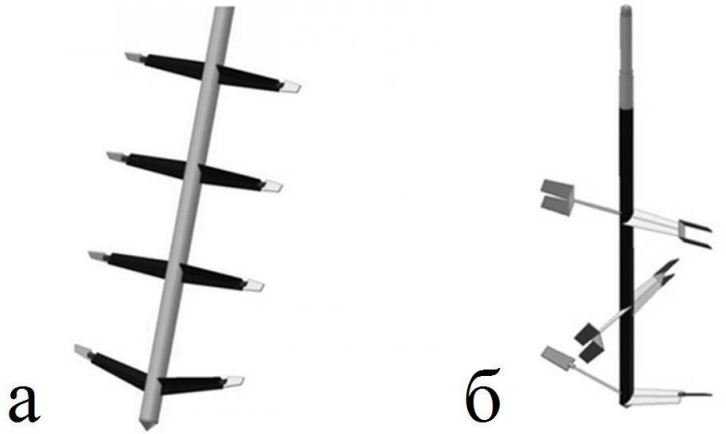


Рисунок 2 – Варианты исполнения лопастных мешалок, а – мешалка с наклоном лопастей в направлении центра оси, б – мешалка с наклоненными плоскостями лопастей относительно оси вращения

Несмотря на частичную доработку перемешивающих устройств, существующие на свободном рынке аналоги, частично не удовлетворяют требованиям фармацевтических производств, поэтому необходима разработка собственного устройства качественного перемешивая жидкостей.

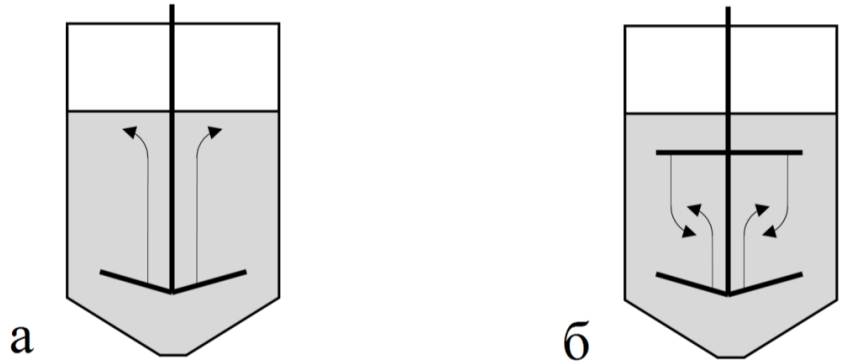
Разработка перемешивающего устройства

Для разработки лопастей мешалки необходимо знать форму резервуара. С целью уменьшения количества застойных зон в перемешивающем устройстве, было решено, что резервуар должен иметь цилиндрическую форму. Для упрощения процесса переналадки установки необходимо осуществление быстрого и полного слития препарата из резервуара. Для решения данной проблемы дно резервуара обычно делают конусоидальной формы.

С целью качественного смешения различных составляющих препарата «Люголь» необходимо осуществить смешивание слоев жидкости. Для решения данной проблемы плоскости лопастей обычно поворачивают на некоторый угол (рисунок 2б). Встает задача определения оптимального угла поворота лопастей. Для перемешивания слоев жидкости необходимо создать максимальный осевой поток. График зависимости осевого потока от угла наклона лопасти очень сложен и зависит от многих факторов, но значительный осевой поток будет создаваться в окрестности угла 45°.

Также стоит обратить внимание на количество лопастей мешалки. Для данной мешалки возможна установка одной или двух пар лопастей (большее количество лопастей будет проблематичнее разместить на относительно небольшой оси). Рассмотрим перемешивание жидкостей в обоих случаях. На рисунке 3а

представлена мешалка, имеющая одну пару лопастей, на рисунке 3б – мешалка с двумя парами лопастей. Лопасти в обоих случаях



расположены так, что создается поток жидкости, направленный к центру устройства.

Рисунок 3 – Мешалки, а – с одной парой лопастей, б – с двумя парами лопастей

В первом случае поток жидкости просто проходит через разные слои жидкости, перемешивая их. Во втором случае создается два встречных потока: от верхней пары лопастей вниз и от нижней пары лопастей вверх. Во втором случае идет образование турбулентных зон, что в особенности характерно для вязких жидкостей, поэтому происходит более качественное перемешивание составляющих препарата. Но стоит учесть, что при этом в нижних и верхних слоях жидкости будут создаваться застойные зоны. Эта проблема решается периодическим

кратковременным включением реверса перемешивающего устройства.

Таким образом, было решено, что проектируемая мешалка будет иметь 2 пары лопастей, каждая из которых будет создавать поток жидкости, направленный к центру перемешивающего устройства, и через некоторые промежутки времени перемешивающее устройство должно включать реверс.

Расчет параметров мешалки

Для разработки перемешивающего устройства очень важным является точное определение параметров конструкции и правильного подбора приводного механизма.

Минимальная производительность аналогичных линий розлива равно 450 бут./ч, а объем каждой бутылки 25 мл, следовательно можно вычислить минимальный объем буферного резервуара:

$V_{min} = V_т \cdot Q \cdot k_з = 25 \cdot 450 \cdot 1{,}1 \approx 12{,}5$ л;

V_{min} – минимальный объем резервуара без учета конуса для слива;

$V_т$ – объем тары;

Q – минимальная производительность линии;

$k_з$ – коэффициент запаса ($k_з$=10%).

Для определения диаметра мешалки необходимо знать диметр резервуара. Были взяты наиболее распространенные пропорции высоты и диаметра резервуара:

H = 2D;

H − высота резервуара (без учета конуса для слива жидкости);

D − диаметр резервуара.

Добавив данную пропорцию к формуле нахождения объема цилиндра, получим:

$$\{V_{min} = \frac{\pi \cdot D^2}{4}_{min} \cdot H;$$

π·D2 ·Dmin·2

Vmin = min 4D = √2·Vmin = √2·12,5 ≈ 2,8 дм = 282 мм;

min π 3.14

Зная минимальный диаметр резервуара, согласно формулам источника, можно получить диаметр мешалки воспользуемся формулами расчета лопастных мешалок [9]:

$$D_{min} = 0{,}7 \cdot d_M$$

$* 282 \approx 200$

мм; $d_м$ – минимальный диаметр мешалки.

Далее необходимо вычислить мощность перемешивания жидкости (мощность, необходимая для вращения мешалки):

$N_n = k_N \cdot \rho_м \cdot \nu^3 \cdot d^5$ – мощность перемешивания. (1)

k_N – критерий мощности;

ρ – плотность жидкости;

ν – частота обращения мешалки.

Критерий мощности – величина, зависящая от мешалки (т.е. включающая в себя тип мешалки, количество лопастей, их наклон и т.д.) и от центробежного числа Рейнольдса. Параметры мешалки известны, поэтому для нахождения критерия мощности необходимо вычислить число Рейнольдса по формуле:

$\rho \cdot \nu \cdot d2$

Re =

м – формула нахождения числа Рейнольдса

η

η − динамическая вязкость жидкости.

По экспертной оценке, частота вращения лопастей в мешалке такого типа не должна превышать 2 Гц. Плотность глицерина: ρ=1260 кг динамическая

м

вязкость: μ=1480мПа · с, поэтому число Рейнольдса:

Re =

ρ · ν · d2

m =

η

1260 · 2 · 0,142

1480 · 10−3

= 0,033 · 103 = 33

Критерий мощности k_N находится при известном типе мешалки и числе Рейнольдса при помощи кривых различных критериев мощности (Рисунок 4).

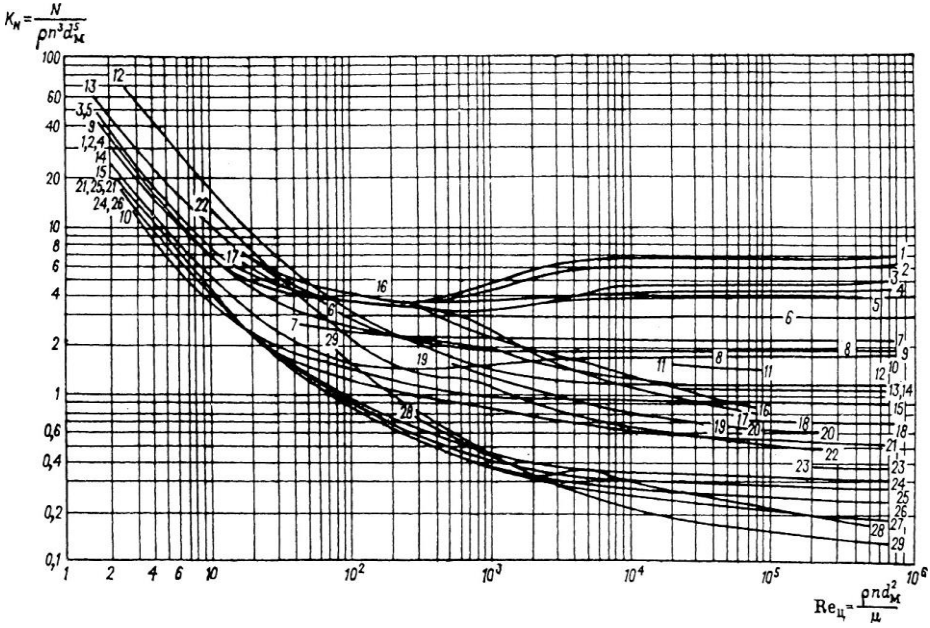


Рисунок 4 – Различные кривые критерия мощности

Лопастная мешалка с двумя парами лопастей, наклоненных под 45° находится на кривой 28. Число Рейнольдса равно 33, поэтому критерий мощности примерно равен 4. Пользуясь формулой (1), рассчитаем мощность перемешивания:

$$N_m = k_N \cdot \rho \cdot \nu^3 \cdot d^5 = 4 \cdot 1260 \cdot 2^3 \cdot 0{,}2^5 = 3{,}22 \text{ Вт}.$$

Далее следует рассчитать мощность, которую необходимо сообщить оси перемешивающего устройства.

$$N_c = \frac{N_n \cdot k_з}{\eta_{меш}}$$

$$= \frac{8{,}47 \cdot 1{,}2}{0{,}72} = 5{,}37 \text{ Вт};$$

$k_з$ – коэффициент запаса (20%. По экспертной оценке); $\eta_{меш}$ – КПД мешалки (72%).

Исходя из расчетов, можно сделать вывод, что двигатель должен сообщать оси мешалки не менее 5,37 Вт и вращаться с частотой 120 об/мин.

Заключение по разработке перемешивающего устройства

Исходя из исследований, приведенных в предыдущих двух пунктах, можно подвести итог по разделу разработки перемешивающего устройства. Итоги подведены в качестве таблицы с техническими характеристиками проектируемого узла перемешивания.

Таблица 3. Технические характеристики перемешивающего устройства

№	Наименование параметра	Ед. измерения	Величина
1	Сообщаемая мощность	Вт	5,37
2	Частота обращения	Об/мин	120
3	Количество пар лопастей	-	2
4	Наклон плоскости лопастей	градусы	45
5	Вместимость резервуара	л	12,5

Разработка дозатора

Обзор различных типов дозаторов

Исходя из данных, приведенных в обзоре аналогов автоматизированных линий, можно сделать вывод, что на рынке существует большое количество различных типов дозирующих устройств. Наиболее распространенные из них – поршневой

дозатор и перистальтический дозатор. Для поиска оптимальной реализации узла розлива был проведен обзор аналогов дозирующих устройств, приведенный в качестве таблицы 4.

Критериями при данном обзоре стали следующие параметры: цена установки и диапазон объема дозирования (дозируемая жидкость – лекарство, поэтому существует необходимость розлива по таре малого объема: от 10 до 100 мл). Обзор был проведен только среди перистальтических и поршневых дозаторов. Узлы, дозирующие по уровню, не вошли в обзор, по причине высокой стоимости и нерентабельности для фармакологических производств.

Таблица 4. Обзор аналогов дозирующих установок

Название установки	Диапазон розлива, мл	Цена установки, руб.	Тип дозатора
Дозатор поршневой PPF-100 [10]	10-100	95 363.	Поршневой дозатор
Дозатор автоматический DF-200 [11]	10-200	125 000	Поршневой дозатор

Помпа дозирующая Jecod DP-4 [12]	1-9999	9 630	Перистальтический дозатор
Kamoer LLS Plus [13]	7-12000	22 990	Перистальтический дозатор

Обзор был проведен среди средних ценовых категорий обоих видов дозатора. Установки, имеющие стоимость около нижней границы ценовой категории (для перистальтического дозатора от 200 руб., для поршневого дозатора от 10 000 руб.), а так же около верхней ее границы (для обоих типов установок около 700 000 руб.) в обзор не вошли. Исходя из данных таблицы 4, можно сделать вывод, что оба типа дозирующих устройств удовлетворяют заданному диапазону розлива. Установки, основанные на перистальтическом дозаторе, имеют более низкую стоимость (так же граница ценовой категории ниже, чем у поршневого дозатора), ввиду простоты конструирования данного узла. Перистальтический дозатор по своей конструкции не подразумевает использование клапанов, поршневой же, нуждается в использовании двух обратных клапанов, вследствие чего его цена выше.

Разработка перистальтического дозатора

На рынке существует большое количество готовых перистальтических дозаторов, имеющих хорошие характеристики, но высокую цену. Было

решено изготовить собственное дозирующее усторйтсво, прототипом, для которого станет существующий на рынке перистальтический дозатор. Наиболее распространенные конструкции таких дозаторов показаны на рисунке 5.

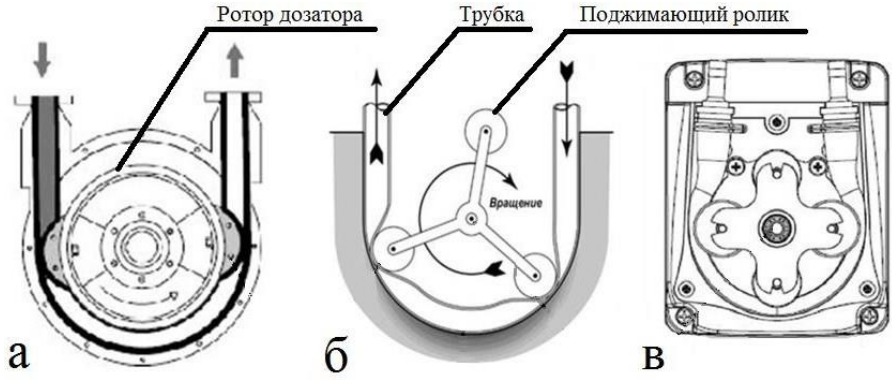


Рисунок 5 - Существующие виды перистальтических дозаторов, а–с двумя роликами, б – с тремя роликами, в –с четырьмя роликами.

С целью увеличения точности проектируемого дозатора, а также уменьшения пульсации подачи жидкости, необходимо увеличение количества роликов, поэтому было решено, что проектируемый дозатор будет иметь 4 ролика на роторе. Для дозаторов такого типа существует ряд технологических проблем, некоторые из которых представлены на рисунке 8.

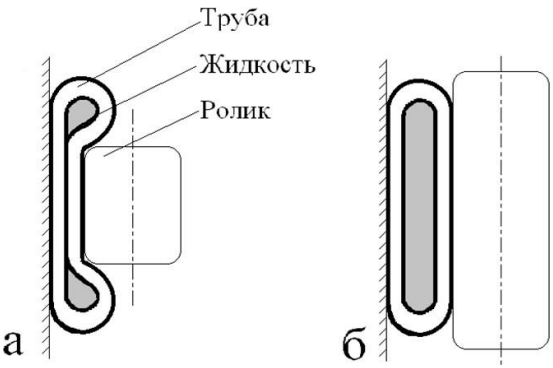


Рисунок 6 – Сложности конструкции дозатора, а – неполное прижатие по края трубы, б – неполное прижатие по всему периметру трубы

Рисунок 6а показывает следующую проблему: ролик ротора не полностью перекрывает доступ дозируемой жидкости в краях трубы, что вызывает потерю точности дозировки. Эту проблему стоит учесть в разработке, хотя она решается не сложно: длина ролика должна быть не менее, чем половина длины внутренней окружности трубки, а если учесть погрешность в изготовлении

детали и установке ее на узел, то длина ролика должны быть не менее половины длины внешней окружности трубы:

$$L \geq \frac{L_{окр.}}{2} = \frac{L_{окр.}}{2} = \frac{\pi \cdot D}{2} \quad (2)$$

L − длина ролика

$L_{окр.}$ − длина внешней окрудности D − внешний диаметр трубы

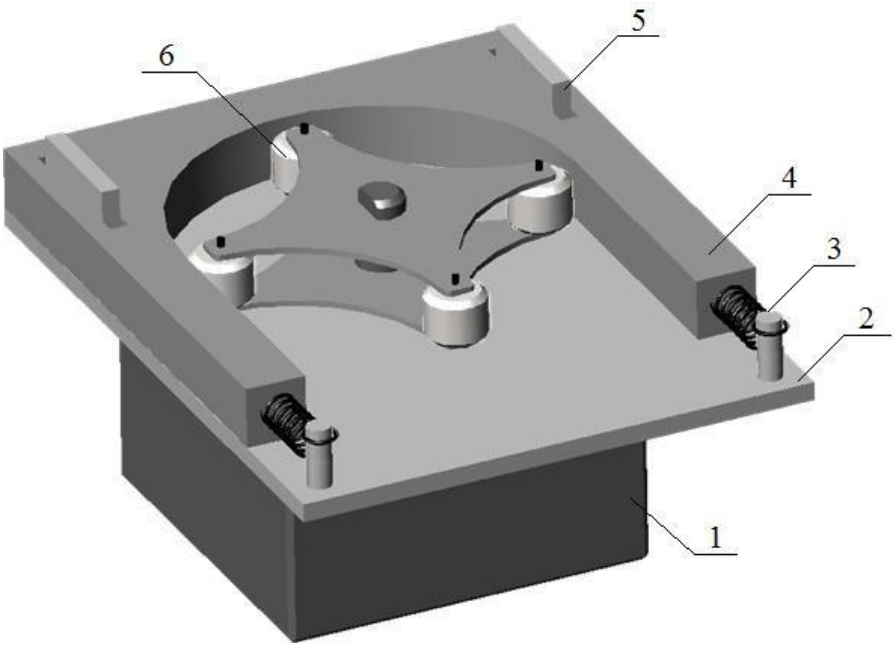


Рисунок 7 – Эскиз перистальтического дозатора Проектируемый узел включает в себя двигатель (1), основание дозатора

(2), две пружины (3), корпус дозатора (4), две направляющие корпуса дозатора

(5) и четыре прижимающих трубку ролика (6).Трубка помещается в «арку» корпуса перистальтического дозатора. Плотное её поджатие производится при помощи пружин, осуществляющих прижатие корпуса дозатора к роликам. Этим же образом решается проблема заклинивания дозатора, в случае некорректной центровки ротора.

Расчет параметров перистальтического дозатора

Для конструирования перистальтического дозатора необходимо рассчитать такие параметры, как длину ролика на роторе и диаметр окружности, входящей в раму крепления трубы. Для расчета длины ролика воспользуемся формулой (2). Для выбранной нами трубки внешний диаметр: D=7 мм, следовательно

$\pi \cdot D$

$L \geq \quad = 2$

$3{,}14 \cdot 7 \quad 2$

$= 11$мм

Далее необходим расчет диаметра окружности, входящей в корпус дозатора (dк). Объем, находящийся между двумя роликами (объема одного шага) занимает одну четверть от объема целого кольца(Vк),значит, выражение для определения объема между двумя роликами имеет следующий вид:

Vк l

V = = S ·

π · d2

= т ·

π · dк

= d

π2 · d2

· т ;

4 4 4 4

к 16

V − объем одного шага дозатора (межджу двумя роликами); Vк − объем всего кольца, входящего в раму дозатора;

dт − внутренний диаметр трубы;

dк − диаметр кольца, входящего в раму дозатора; 3.142 · 0,52

V = dк · 16 = 6,5 · dк.

Для увеличения точности розлива возьмем целый объем одного шага дозатора – 1 мл, тогда для розлива флакона, двигатель должен будет пролить этот объем 24 раза (т.е. сделать 6 полных оборотов). В таком случае диаметр арки в корпусе проектируемого перистальтического дозатора будет равен 65 мм.

3. КОНСТРУКТОРСКАЯ ЧАСТЬ

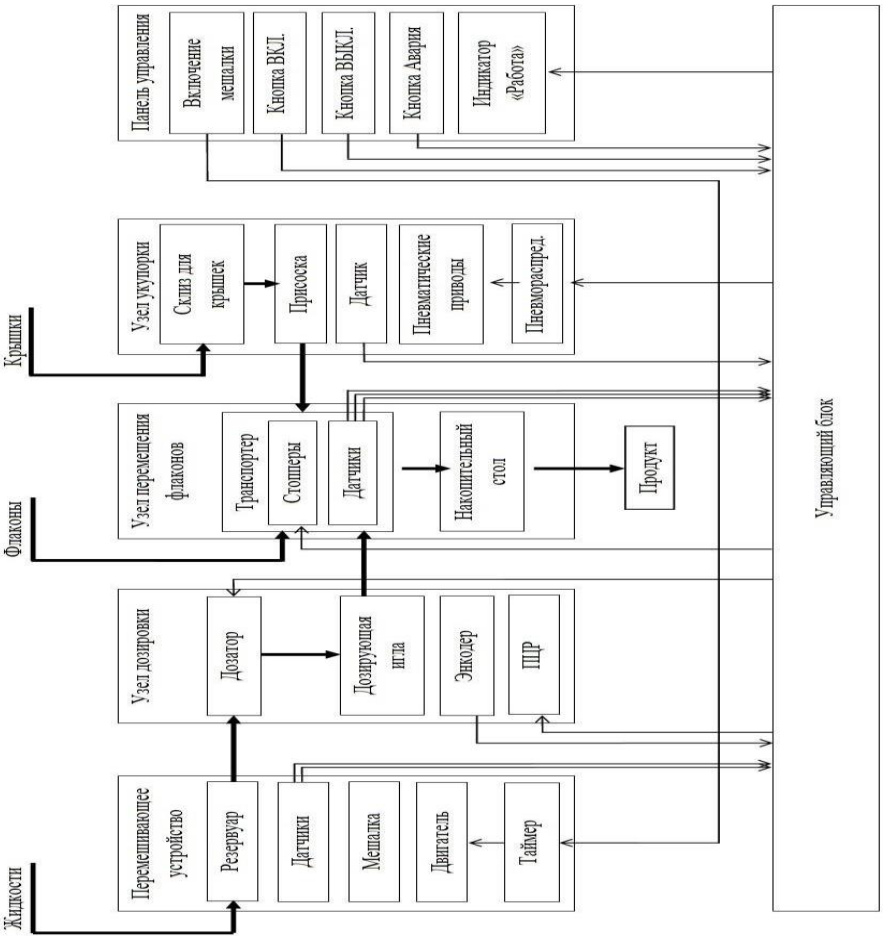
Для отладки таких систем, как пневматическая, электрическая и система автоматизированного управления был разработан и изготовлен макет автоматической линии. На нем проверяется работоспособность вышеуказанных систем, что позволит в дальнейшем применять эти системы для промышленной линии. В данной части описывается разработка и конструирование макета АЛР.

Детализированная функциональная схема

В исследовательской части была приведена краткая функциональная схема автоматической линии. Для выбора приводных механизмов необходимо знать их типы и количество, поэтому появляется необходимость детализации существующей схемы с отображением всех блоков, датчиков, исполнительных устройств, драйверов и панели управления. Детализированная функциональная система АЛР приведена на рисунке 8. Введем расшифровку условных обозначений:

- линия, обозначающая перемещение физических объектов (Жидкостей, готового препарата, флаконов, крышек, готового продукта);

- линия, обозначающая управляющие сигналы.



Выбор приводных механизмов

Выбор двигателя перемешивающего устройства

В пункте 2.4.3. «Расчет параметров мешалки» был определен объем резервуара (12,5 л), но для конструирования макета АЛР целесообразно использование резервуара меньшего объема и соответственно меньшего диаметра лопастей. Это приведет к изменению мощности, которую необходимо сообщать жидкости. Для применяемого резервуара был проведен аналогичный расчет определения мощности. Мощность, которую необходимо сообщить жидкости составила 0,6 Вт.

Для подобных устройств обычно используют два типа двигателей: асинхронный двигатель и двигатель постоянного тока (далее ДПТ). Для удобства управления было решено использовать ДПТ.

Далее необходимо определить мощность двигателя, которая зависит от КПД редуктора (для обеспечения запаса мощности были взяты нижние границы диапазонов КПД редукторов). Мощность мотор-редуктора определялась по следующей формуле:

Nn

N =

ДПТ

· ηред

Nn − сообщаемаямешалке мощность (0,6 Вт); ηДПТ − КДП двигателя (90%);

ηред − КПД редуктора.

Таблица 5. Сравнение трех видов редукторов

№	Тип редуктора	КПД редуктора, %	Мощность мотор-редуктора, Вт
1.	Червячный	42	1,43
2.	Цилиндрический	90	0,7
3.	Планетарный	90	0,7

С целью определения оптимального решения был проведен обзор различных двигателей. В исследовании по разработке мешалки было определено, что вал двигателя должен вращаться с частотой 120 об/мин.

Таблица 6. Обзор мотор-редукторов для перемешивающего устройства

№	Наименование двигателя постоянного тока	Мощность, Вт	Частота, Об/мин	Напряжение, В	Ток, А	Тип редуктора	Передаточное отношение
1	WG3929 [14]	19	94	24	0,8	Червячный	31
2	IG-22CGM	1,7	11	24	0,2	Планетарный	66

	[15]		0				
3	RA-12WGM [16]	0,46	135	6	0,175	Цилиндрический	102
4	IG-12GM [17]	0,59	160	3	0,340	Планетарный	64

На основании данных таблицы 6, можно сделать вывод, что двигатели с червячным редуктором показывают чрезмерно избыточную мощность, и при этом недостаточную частоту вращения. На основе вышеуказанных критериев, был выбран двигатель IG-22CGM с планетарным редуктором, избыточной мощностью 1,7 Вт и частотой 110 об/мин. Фотография данного привода приведена на рисунке 1 в приложении А.

Выбор транспортера

Из данных ТЗ известно, что диаметр стандартного флакона для розлива препарата «Люголь» - 37 мм. Но стоит учесть, что стеклянная тара обладает рядом особенностей, в том числе видоизменением формы сосуда до эллипса, поэтому транспортерная лента должна быть не менее 40 мм по ширине. Длина транспортера должна быть достаточной для расположения на нем некоторого количества флаконов, что обеспечит наличие буферной зоны автоматической линии. Диаметр флакона - 37 мм, значит буферная зона линии $5 \cdot 37 \approx 200$ мм. Так же стоит учесть габариты узлов розлива и укупора (в сумме не менее 150 мм). В этой связи, длина транспортера должна быть не менее 350 мм.

Было решено использовать для макета транспортер фирмы «Дидактические системы» длиной 400 мм, шириной 40 мм (Рисунок 2 см. приложение А).

Выбор механизмов отсекателей и элементов центрирования

Для функционирования линии существует необходимость остановки и центрирования флаконов. Есть несколько способов решения данной проблемы: пневматические отсекатели, электромагнитные отсекатели или линейные актуаторы. В проектируемой установке оптимальным будет использование соленоидов поступательного движения, так как этот приводной механизм наиболее прост и дешев. При использовании соленоидов с небольшим ходом (не более 15 мм), этот приводной механизм будет являться одновременно и отсекателем, и элементом центрирования, как показано на рисунке 9.

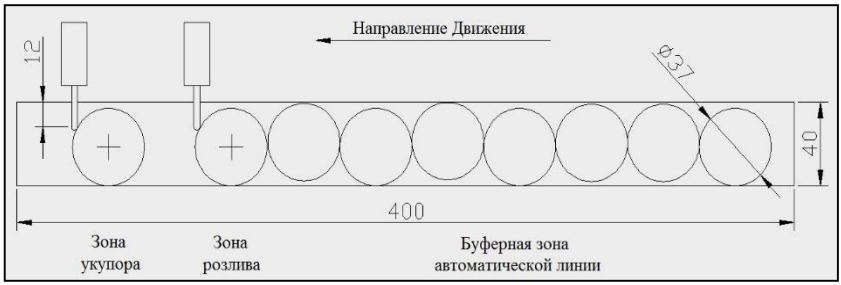


Рисунок 8 – Схема отсекания и центрирования флаконов

Выбор приводных механизмов для узлов розлива и укупора

Для того чтобы не допускать пролитие препарата на ленту во время розлива, необходимо осуществить опускание дозирующей иглы внутрь флакона. Для осуществления движения дозирующей иглы вдоль вертикальной оси наиболее рациональным является использование пневматического привода, так как он обеспечивает опускание иглы в бутылку с высокой скоростью и приемлемой точностью. Кроме того, этот приводной механизм более компактный, что упрощает и упрочняет его крепление над конвейером. Для изготовления макета был выбран возвратный однополостной пневмоцилиндр (этот приводной механизм наиболее прост и дешев) с площадью сечения 16 мм2 и ходом 15 мм (Рисунок 3 см. приложение А).

В макет узла розлива также входит двигатель перистальтического дозатора. Для создания давления в трубе используется двигатель от стеклоподъемника ВАЗ. Этот приводной механизм наиболее распространенный и дешевый, при этом он имеет достаточную мощность (12 Вт) и частоту вращения (70 об/мин), а также червячный редуктор, что обеспечивает большой момент.

Основными задачами узла укупора является перемещение и запрессовка крышек. Наиболее оптимальным для решения данной задачи будет использование присоски крышек, для которой создание пониженного давления будет осуществляться эжектором. Присоске необходимо придать движение вдоль двух осей: вертикальной и горизонтальной. В данном узле существует несколько основных проблем: точность позиционирования присоски, плавность хода и достаточное, но не избыточное усилие для укупора бутылки. Оптимальным решением, удовлетворяющим перечисленным требованиям, является пневматический привод, поэтому для перемещения каретки с присоской было решено использовать ПЦ.

Для изготовления макета данного узла были выбраны однополостные возвратные ПЦ. Для горизонтального перемещения был выбран ПЦ с ходом 100 мм, а для вертикального – 15 мм.

Конструирование узлов линии

Узел транспортировки флаконов

Данный узел включает в себя транспортер, систему склизов, стопперы для флаконов и датчики наличия флакона (концевые выключатели).

Эскиз макета данного узла представлен на рисунке 10. Для изготовления ограничителей и склизов на макете используется

алюминиевые уголки. Стопперы и концевые выключатели крепятся к корпусу транспортера на стальные уголки.

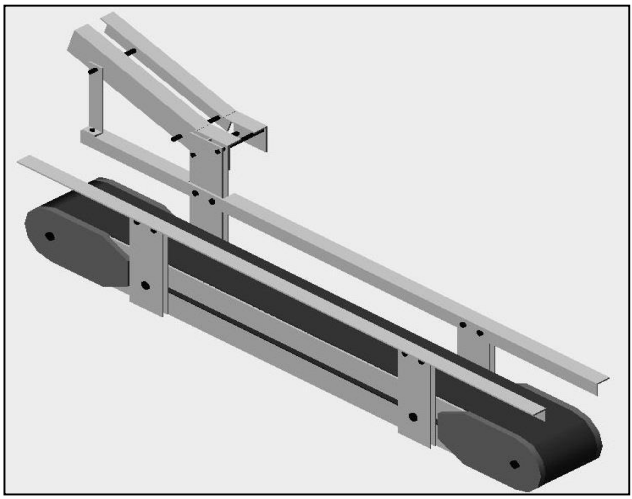


Рисунок 10 – Эскиз узла транспортировки флаконов

Узел перемешивания препарата

Узел перемешивания препарата включает в себя следующие составляющие: резервуар, двигатель, ось перемешивающего устройства, муфта и лопасти.

Ось мешалки изготавливается из нержавеющей стали d=6 мм. Для ее крепления к ротору необходима муфта. Так как частота вращения не превышает 120 об/мин, было решено использовать глухую муфту крепления оси.

Развёртка лопастей мешалки изготавливается из нержавеющей стали толщиной 2 мм (рисунок 11). После вырезания заготовки, плоскости каждой из двух лопастей сгибаются на 45 градусов.

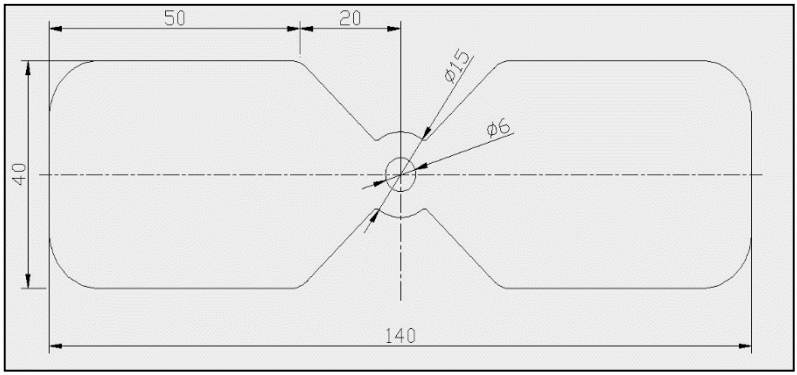


Рисунок 11 – Развёртка лопастей мешалки

Узел дозировки препарата

Узел дозировки препарата состоит из следующих составляющих: перистальтический дозатор, подающая трубка, ПЦ узла розлива и дозирующая игла.

Для дозировки препарата используется перистальтический дозатор. Данный узел конструируется в соответствии с его разработкой в части 2.5.2 (рисунок 12). Дополнительно

производится обрамление дозатора при помощи уголка. Подробное конструирование данного механизма представлено в приложении Б.

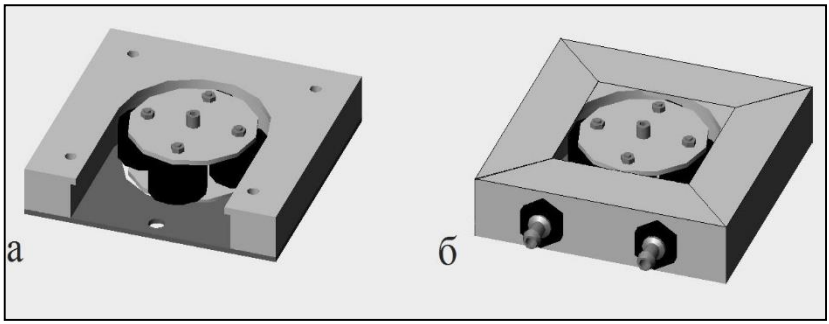


Рисунок 12 – Перистальтический дозатор, а – корпус, б – корпус с обрамлением и креплением штуцеров.

Узел укупорки флаконов

Для осуществления перемещения пробки необходима присоска, эта же деталь будет осуществлять укупор флаконов. На рынке существуют присоски необходимого диаметра, но они не подходят для данного узла, потому что имеют недостаточную жесткость. По этой причине было решено изготовить данную деталь самостоятельно. Присоска изготавливается из стального цилиндра d=20 мм на токарном станке. Эскиз изображен на рисунке 13.

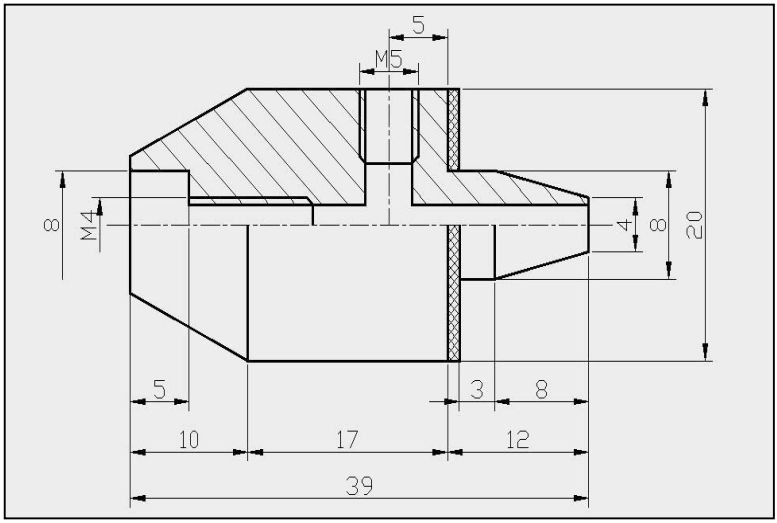


Рисунок 13 – Эскиз присоски

Узел укупорки флаконов укупорки флаконов представлен на рисунке 14. Он включает в себя следующие составляющие: каретка с ПЦ вертикального действия (5), направляющие (2), линейные подшипники (3), крепление направляющих к плоскости (1), ПЦ горизонтального действия (4), присоску (6), а также портал для крепления направляющих

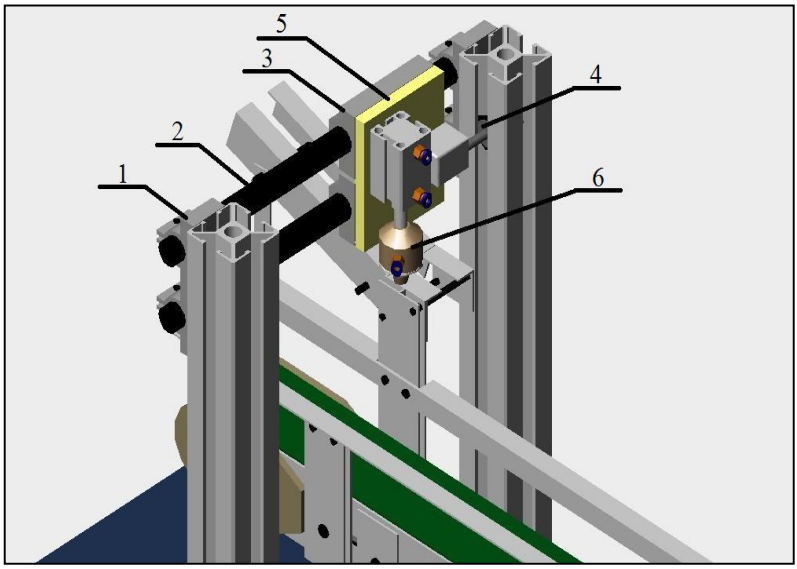


Рисунок 14 – Эскиз узла укупорки

Для изготовления макета данного узла используется станочный профиль 30х30 мм, стальные направляющие d=12 мм и выбранные ранее ПЦ. Крепление ПЦ горизонтального действия (4) осуществляется к профилю. Крепление ПЦ вертикального действия (5) к каретке осуществляется при помощи стального швеллера.

Эскиз крепления присоски изображен на рисунке 14 (б). Используется пневматический тройник Festo с цанговым зажимом 6 мм, поэтому при подводе вакуума к свободному зажиму тройника создается пониженное давление на присоске.

Разработка электрической и пневматической схем

Электрическая и пневматическая схемы были создана в программе Fluid Sim. Данный САПР позволяет симулировать подачу питания на электрические компоненты и сжатого воздуха на пневматические компоненты. В дополнение к этим функциям Fluid Sim позволяет использовать логические элементы для построения программы.

Пневматическая схема установки

Пневматическая схема проектируемой установки приведена на рисунке

16. Она включает в себя 3 ПЦ (один для узла розлива и два для узла укупора), 4 пневмораспределителя (далее ПР) и эжектор.

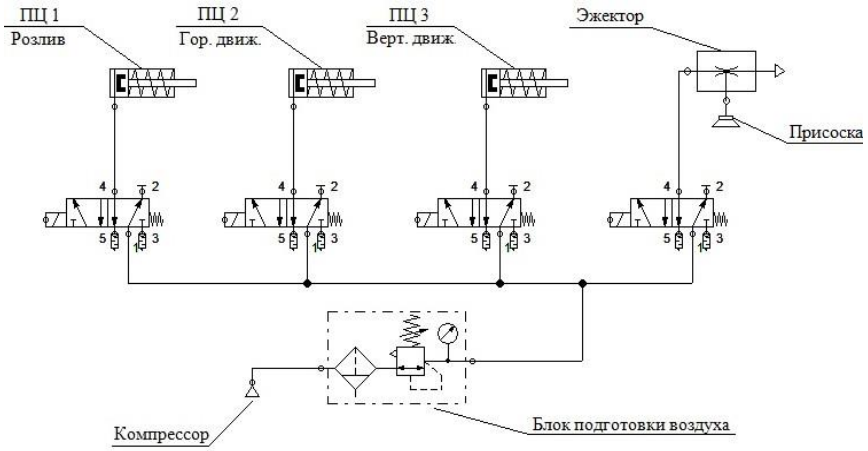


Рисунок 15 – Пневматическая схема установки

С целью уменьшения стоимости установки были выбраны наиболее распространенные на рынке бистабильные ПР типа 5-2.

Электрическая схема установки

В связи с тем, что время розлива препарата в несколько раз больше времени укупора флаконов, было решено передавать сигнал для движения приводных механизмов узла укупора и розлива по времени, а не по готовности элементов, что сокращает количество используемых датчиков. Датчики занимают 10 из 12 входов контроллера. Это означает, что у заказчика всегда остается возможность доработки линии для увеличения ее производительности или небольшого изменения функционала.

Таблица 7 представляет собой таблицу подключений датчиков и исполнительных устройств к контроллеру.

Таблица 7. Таблица подключений

Вход	Назначение	Выход	Назначение
I1	Включение установки	Q1	Стоппер 1 (розлив)
I2	Выключение установки	Q2	Стоппер 2 (укупор)

I3	Аварийное отключение	Q3	Мотор розлива (M1)
I4	Датчик наличия флакона на узле розлива	Q4	ПЦ розлива (ПЦ 1)
I5	Датчик наличия флакона на узле укупора	Q5	ПЦ гор. движения укупора (ПЦ 2)
I6	Датчик наличия флакона в начале транспортера	Q6	ПЦ верт. движения укупора (ПЦ 3)
I7	Датчик верхней границы жидкости в резервуаре	Q7	Эжектор присоски (ПЦ 4)
I8	Датчик нижней границы жидкости в резервуаре	Q8	Индикатор
I9	Датчик наличия крышек на склизе		
I10	Энкодер привода дозатора		
I11	Не используется		
I12	Не используется		

На рисунке 17 приведена электрическая схема подключений датчиков и исполнительных элементов к ПЛМ. На схеме представлена разводка контроллера с подключением всех датчиков. Стоит уточнить, что питание на транспортер и накопительный стол

подается все время работы линии. Двигатель перемешивающего устройства включается через таймер, который автоматически включает реверс через настраиваемые промежутки времени. Питание данного таймера происходит от отдельной кнопки на панели управления.

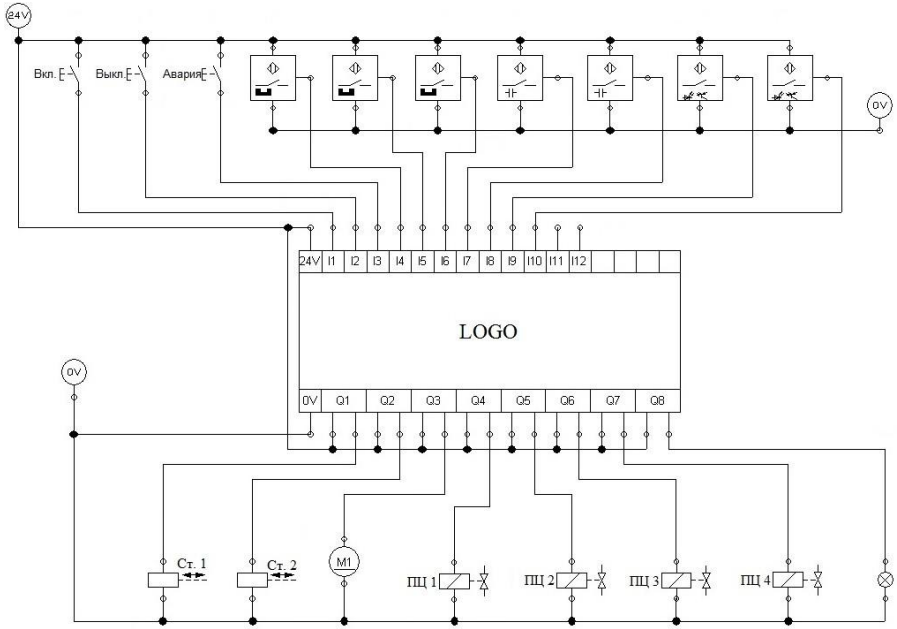


Рисунок 16 – Электрическая схема установки

4. СИСТЕМА УПРАВЛЕНИЯ АВТОМАТИЧЕСКОЙ ЛИНИЕЙ

В данном разделе показана разработка системы управления автоматической линией, разработан алгоритм работы АЛР и проведено моделирование работы системы автоматического управление (САУ).

Структура системы управления автоматической линии

Все управляющие сигналы проектируемой автоматической линии делятся на сигналы электрической системе (ЭС) и сигналы пневматической системе (ПС). Структурная схема САУ представлена на рисунке 17.

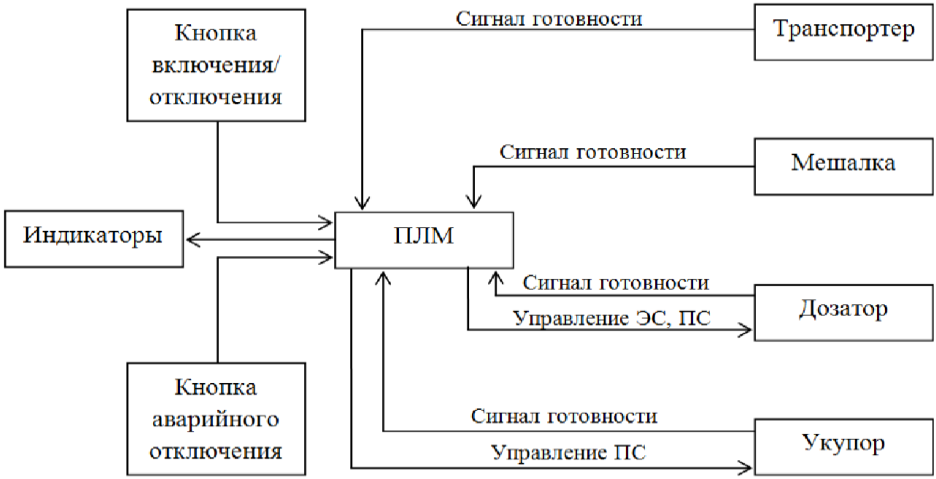


Рисунок 17 – Структура системы управления

Алгоритм работы автоматической линии

Для осуществления управления автоматической линией был составлен алгоритм работы, изложенный поэтапно ниже:

1) Изначальное положение линии:

Все ПЦ втянуты. Стопперы задвинуты. В резервуаре перемешивающего устройства нет жидкости. На склизе нет крышек. На транспортере нет бутылок.

2) Запуск установки:

Включение кнопки питания. Загорается индикатор «Питание», индикатор

«Работа» не горит. Замыкание кнопки «Вкл», индикатор «Работа» подает сигнал об отсутствии жидкости в резервуаре (мигает один раз через небольшой промежуток времени). После добавления жидкости индикатор «Работа» подает сигнал об отсутствии крышек на склизе (мигает дважды через небольшой промежуток времени). После добавления крышек на склиз индикатор «Работа» подает сигнал об отсутствии флаконов на транспортере (мигает трижды через небольшой промежуток времени). После добавления флаконов на транспортер (при выполнении всех трех вышеизложенных условий) запускается работа станции: начинают вращаться лопасти перемешивающего устройства (через некоторые промежутки времени включается

реверс), начинает вращаться конвейерная лента, начинает вращаться накопительный стол. Индикатор

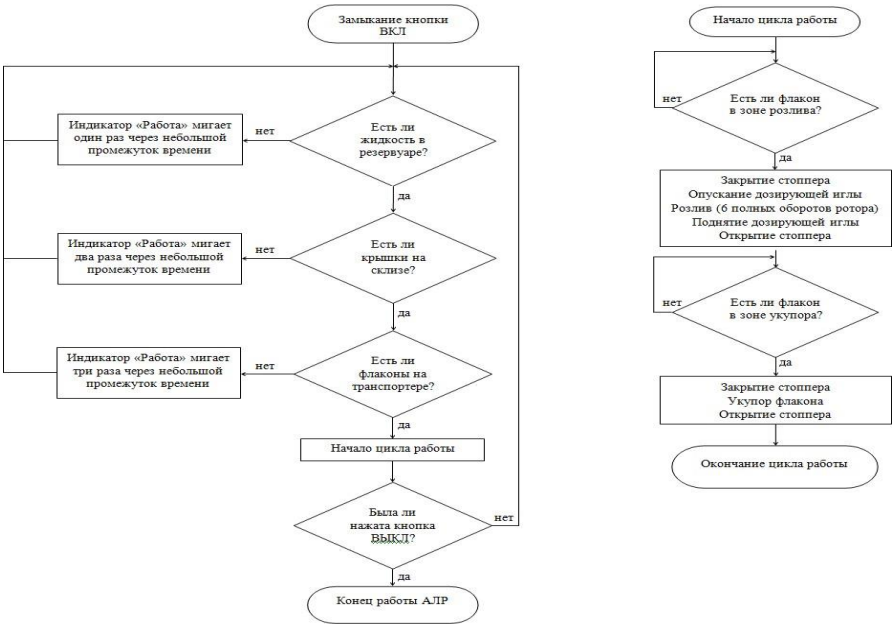
«Работа» подает сигнал работы установки (горит без миганий).

3) Розлив препарата:

Флакон на движущемся транспортере замыкает концевой выключатель узла розлива. Выдвигается стоппер 1. Спустя 0,5 секунды внутрь флакона опускается дозирующая игла. Спустя еще 0,5 секунды начинается розлив препарата. По истечению 6-ти полных оборотов ротора перистальтического дозатора розлив прекращается, дозирующая игла поднимается, и стоппер 1 задвигается. Розлив препарата закончен.

4) Укупорка флаконов с препаратом:

Флакон на движущемся транспортере замыкает концевой выключатель узла укупора. Выдвигается стоппер 2, присоска опускается вниз, начинает работать эжектор. Спустя некоторое время крышка прикрепляется к присоске, присоска поднимается вверх, далее передвигается к флакону и опускается вниз, закупоривая флакон с препаратом. Эжектор перестает создавать пониженное давление. Присоска возвращается на изначальное положение. Стоппер задвигаются. Укупорка флаконов закончена.



Для наглядности и упрощения использования алгоритма при создании управляющей программы была составлена блок-схема алгоритма работы линии, изображенная на рисунке 18.

Рисунок 18 – блок-схема алгоритма работы линии

Для управления АЛР могут быть использованы различные типы управляющих блоков. Оптимальным будет использование программируемого логического модуля Siemens LOGO 8, потому что он обладает рядом преимуществ:

- питание, схожее по напряжению с соленоидами пневмораспределителей и ДПТ (24 В);
- возможность параллельного опроса датчиков;
- возможность дополнения количества входов и выходов при помощи расширяющих модулей.

Существуют две разновидности данного ПЛМ: контроллер с транзисторными и релейными выходами. Данная установка включает в себя большое количество ДПТ, поэтому целесообразно использование ПЛМ с релейными выходами. Такая разновидность контроллера имеет 8 входных и 4 релейных выходных порта, а расширяющий модуль еще 4 входных и выходных порта. Таким образом, используемый блок имеет 12 входных и 8 выходных портов.

Управляющая программа

Управляющая программа была создана в среде LOGO Soft Comfort при помощи логических элементов и таймеров. Одним из преимуществ данной среды является возможность осуществления параллельных действий, поэтому управляющую программу можно разделить на четыре основные части: включение установки; опрос датчиков, разрешающих работу; розлив препарата; укупорка флакона.

Изображение части программы, отвечающей за включение установки, представлено на рисунке 19. Оно включает в себя три входа (вкл., выкл., авария) и выдает разрешающий сигнал всем исполняющим устройствам. Зелеными стрелками обознаются обрезанные линии. С данной части схемы сигналы идут к выходным блокам.

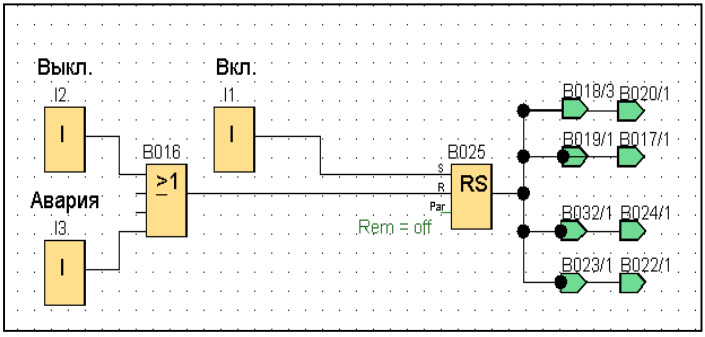
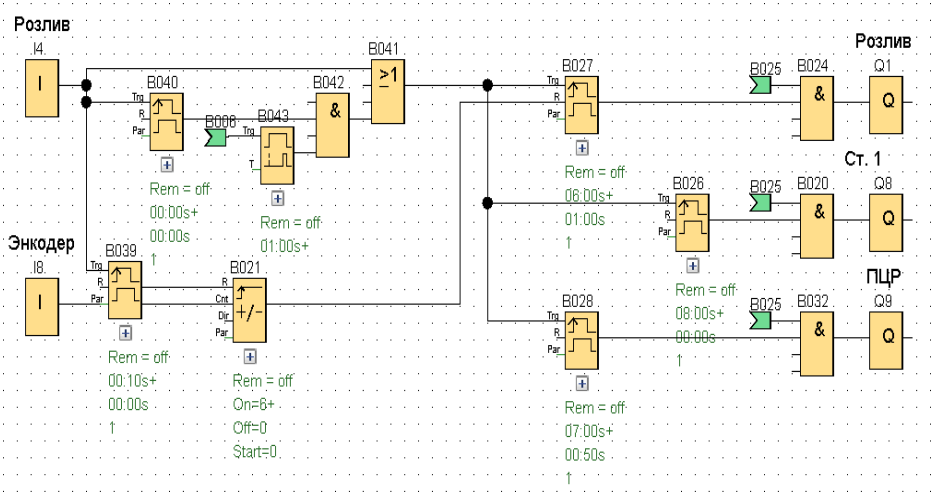


Рисунок 19 – Включение установки

Часть программы, отвечающая за опрос датчиков, показана на рисунке 20. В нее входит 4 датчика: датчик наличия флаконов на транспортере, датчики верхней и нижней границ жидкости в резервуаре и датчик наличия крышек на склизе. Эта часть выдает один из четырех возможных сигналов индикатору. Помимо стандартных логических элементов, используются генераторы импульсов, таймеры с задержкой включения и таймеры, срабатывающие по переднему фронту сигнала.



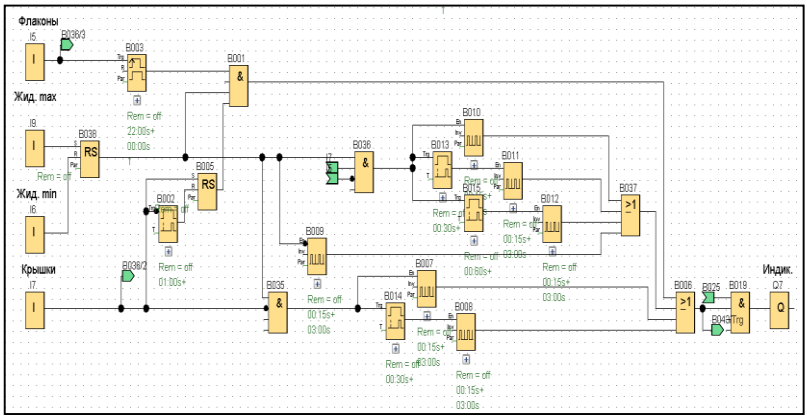


Рисунок 20 – Опрос датчиков автоматической линии

Далее приложена часть, отвечающая за осуществления розлива препарата во флакон. Эта часть включает в себя 2 входных блока: датчик наличия флакона на узле розлива и энкодер мотор-редуктора перистальтического дозатора. Эта часть программы состоит из элементов логики, таймера с задержкой включения, таймеров, срабатывающих по фронту и счетчика сигналов энкодера (розлив прекращается после шести полных оборотов ротора дозатора). Дополнительно в этой части программы выполняется алгоритм возобновления работы после аварийного отключения установки.

Рисунок 21 – Розлив препарата

Четвертая часть программы - управление узлом укупорки флаконов. Эта часть набором своих элементов похожа на предыдущую. Здесь так же выполняется алгоритм возобновления работы после аварийного отключения установки. оборотов ротора

Рисунок 21 – Розлив препарата

Четвертая часть программы - управление узлом укупорки флаконов. Эта часть набором своих элементов похожа на предыдущую. Здесь так же выполняется алгоритм возобновления работы после аварийного отключения установки.

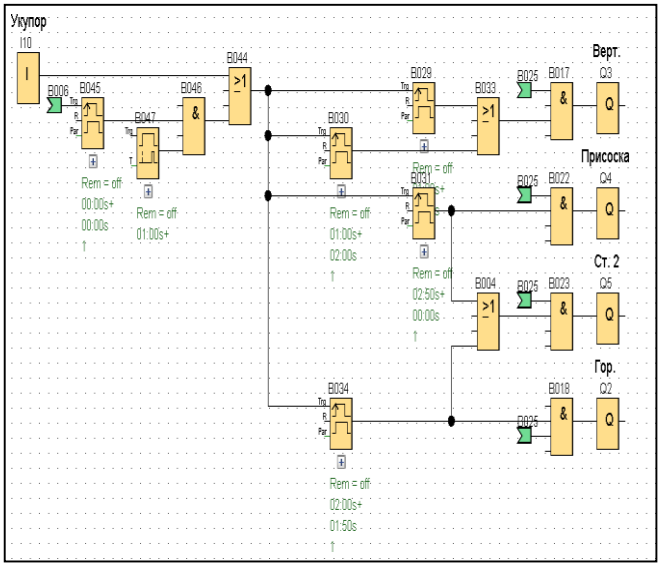


Рисунок 22 – Укупорка флаконов

Моделирование управления линией

В данном разделе производится моделирование и отладка управляющей программы. Для моделирования управления линией в программе Fluid Sim P была создана циклограмма работы САУ.

На циклограмме показан набор графиков, описывающих действие конкретных узлов. Для каждого приводного механизма характерна собственная форма графика. На основании данных зависимостей можно определить время задержки для каждого узла. На рисунке 23 представлены зависимости перемещения штока пневмоцилиндров с нагрузкой и без от времени.

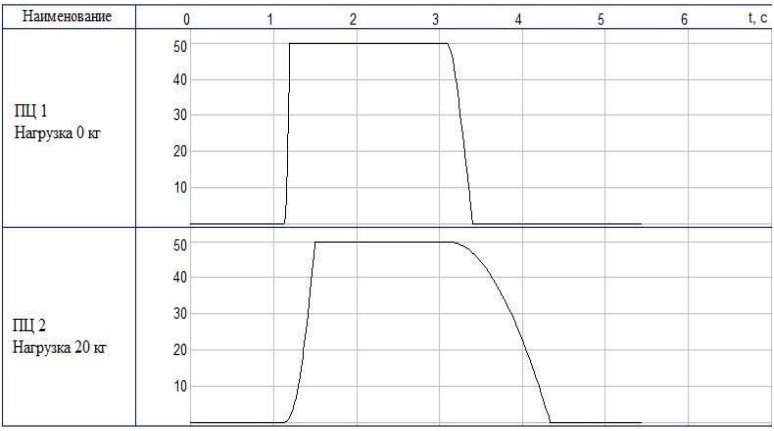


Рисунок 23 – Зависимости хода ПЦ с нагрузкой и без

Наклон и форма кривых получена при моделировании с учетом различных факторов (для пневматики – длина хода штока цилиндра, нагрузка, давление и т.д.; для двигателей – индуктивность, максимальные обороты и т.д.; для соленоидов – индуктивность и длина хода).

На примере данных зависимостей можно сделать вывод, что при действии на ПЦ пренебрежимо малой нагрузки, время задержки составляет не менее 0,5 секунды. Также стоит учесть, что время задержки нелинейно меняется при росте нагрузки на ПЦ.

Для моделирования работы управляющей программы была составлена циклограмма, представленная на рисунке 24.

Циклограмма работы САУ состоит из пяти временных промежутков работы АЛР:

1 – движение флакона до касания с концевым выключателем узла розлива; 2 – розлив препарата;

3 – движение флакона до касания с концевым выключателем узла укупора; 4 – укупорка флакона;

5 – движение флакона к накопительному столу. Окончание цикла работы.

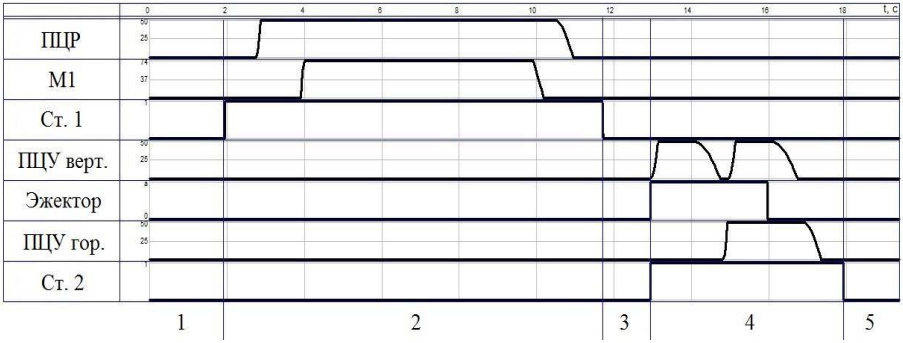


Рисунок 24 – Циклограмма работы САУ

Дополнительно было проведено моделирование работы программы управления САУ в среде LOGO Soft Comfort. С фотографиями моделирования вы можете ознакомиться в приложении В.

ЗАКЛЮЧЕНИЕ

В ходе данной проектной работы была достигнута поставленная цель – создана автоматическая линия розлива вязких медицинских препаратов. Для этого были решены все поставленные задачи:

1. Была определена структуры автоматической линии;
2. Разработано перемешивающее устройство;
3. Разработан перистальтический дозатор;

4. Выполнена компоновка и конструирование автоматической линии;
5. Разработан алгоритм управления системой.

В дополнение к поставленным задачам была проведена сборка макета автоматической линии, на котором были отработаны технические решения и отлажена система автоматизированного управления линией. Результаты данной работы позволят в будущем изготовить и отладить спроектированную автоматическую линию, которая в дальнейшем сможет быть внедрена в любое малотоннажное фармакологическое производство.

Изображения процесса работы, а также результатов приведены в приложении Г.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

[1] Гончарова А. Е. Основные тенденции и перспективы развития фармацевтического рынка Российской Федерации // Вопросы экономики и управления. 2016. № 5. С. 38-41.

[2] QUARCO pharm & medical equipment. URL:https://quarco.ru/projects/sibgmu-project.html

[3] Триблок розлива DCGF 18-18-6. URL:http://promgarant.agroserver.ru/tovari/841113.htm

[4] Полуавтомат розлива ПС-5.

URL: https://tiu.ru/p335614524-poluavtomat-rozliva-dlya.html

[5] Полуавтомат розлива Альтер-04. URL: http://www.alterkmv.ru/o_04.php [6]Автоматическая линия розлива в ПЭТ.

URL:https://www.farpost.ru/lyubertsy/business/equipment/avtomaticheskaja-linija- rozliva-v-pet-taru-ot-3-do-5-litrov-58938735.html

[7] Мешалки для пластиковых емкостей.

URL: http://plast-product.ru/meshalki-dlya-plastikovyx-emkostej

[8] KATES Mieszadła.

URL: http://kates.com.pl/pl/page/mieszadł

[9] Методика расчет мешалок.

[10] Дозатор поршневой PPF-100.

URL: http://hmru.ru/catalog/dozatory-porshnevye/dozator-porshnevoj-ppf-100/

[11] Дозатор автоматический.

URL:DF-200 https://profiteh.com/p26692504-dozator-shprits-dozator.html

[12] Помпа дозирующая JecodDP-4.

URL:https://www.aquashop.ru/sea/morskaya_akvariumistika/pompy/prod_DEL-

DP-4

[13] Перистальтический дозатор KamoerLLSPlus.

URL: http://stan-upakovka.ru/magazin/product/peristalticheskiy-dozator-nasos- kamoer-lls-plus

[14] Червячный мотор-редуктор.

URL: https://electroprivod.ru/wg3929.htm

[15] Планетарный мотор-редуктор.

URL: https://electroprivod.ru/ig-22gm.htm

[16] Цилиндрический мотор-редуктор.

URL: https://electroprivod.ru/ra-12wgm.htm

[17] Планетарный мотор-редуктор.

URL: https://electroprivod.ru/ig-12gm.htm

ПРИЛОЖЕНИЕ А

В данном приложении приведены фотографии некоторых приводных механизмов проектируемой автоматической линии.

Рисунок 1 – Двигатель перемешивающего устройства

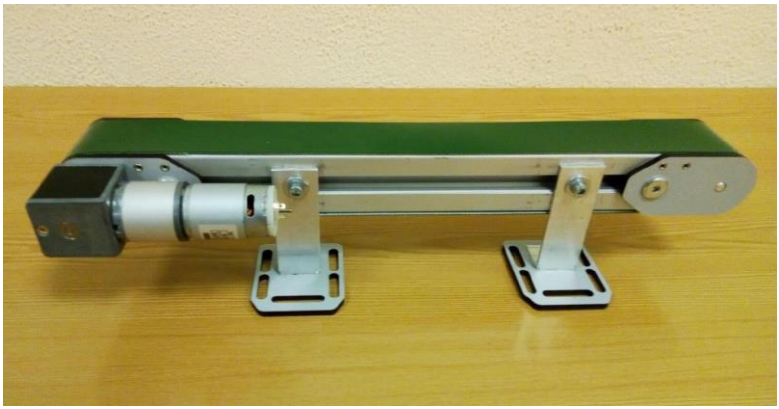


Рисунок 2 – Транспортер

Рисунок 3 – Пневмоцилиндр с ходом штока 15 мм

ПРИЛОЖЕНИЕ Б

В данном приложении приведено подробное описание конструирования перистальтического дозатора.

Выбранный ранее мотор-редуктор имеет квадратный в сечении вал, закрепленный на двух подшипниках, поэтому было решено крепить ротор перистальтического дозатора к нему. При этом валу редуктора нарезается резьба, как показано на рисунке 4.

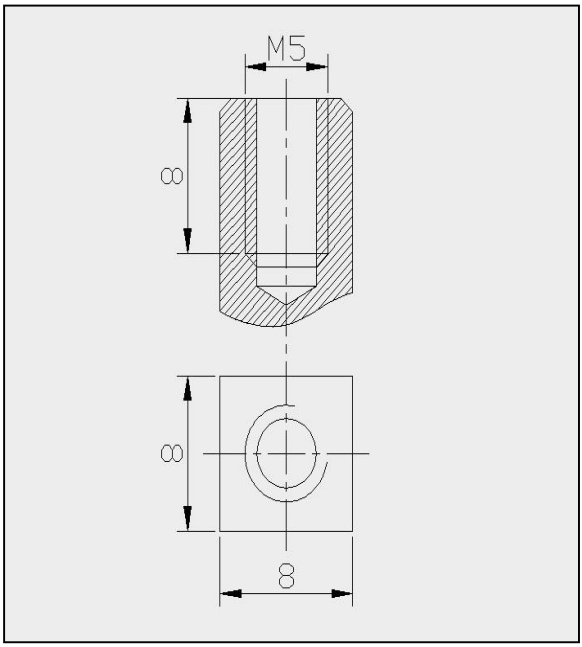


Рисунок 4 – Вал редуктора перистальтического дозатора

Ротор перистальтического дозатора изготавливается из двух алюминиевых пластин 2 мм. Нижняя пластина имеет квадратное отверстие под вал редуктора, верхняя – стандартное отверстие для винта. Таким образом, при скреплении двух пластин нижняя передает момент двигателя, а верхняя закрепляет ротор дозатора на валу. Чертеж пластин приведен на рисунке 5.

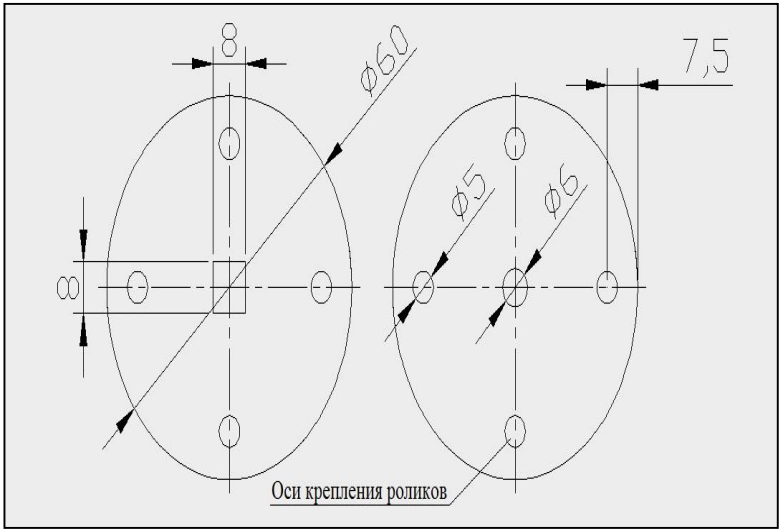


Рисунок 5 – Ротор перистальтического дозатора

Пластины скреплены между собой при помощи четырех стоек, изображенных на рисунке 6. Эти же стойки используются для крепления роликов дозатора.

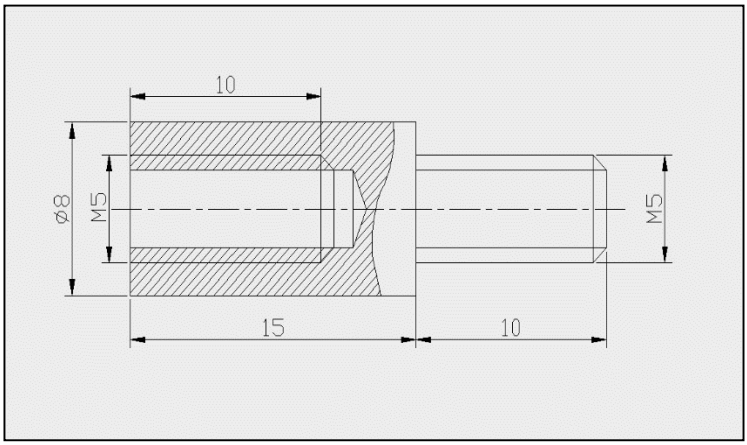


Рисунок 6 – Ось крепления роликов дозатора

Корпус дозатора состоит из двух частей: подвижной части и основания. Подвижная часть выполнена в форме арки с диаметром 69 мм, но верхние 3 мм прорезаны на 67 мм, с целью фиксации трубки внутри корпуса дозатора. Чертеж обеих частей представлен на рисунке 7.

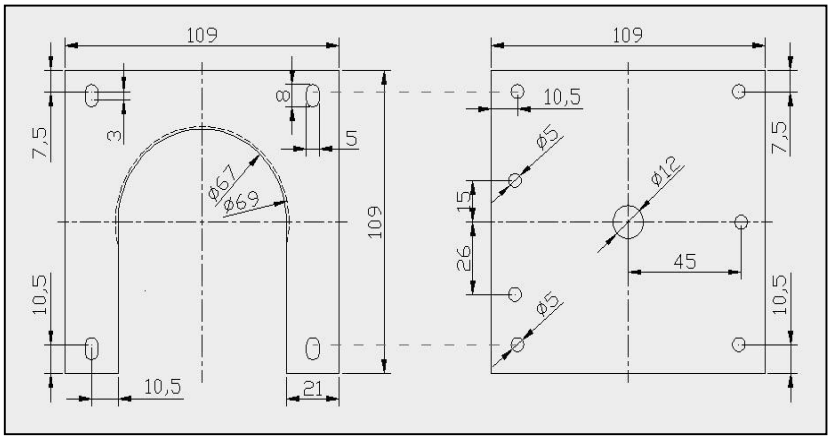


Рисунок 7 – Корпус перистальтического дозатора

Поскольку при работе дозатора трубка испытывает давление со стороны ролика, ее следует закрепить к основанию дозатора. Для жесткого зажатия трубки изготавливаются два штуцера. Они вытачиваются на токарном станке из стального цилиндра d=10 мм и закрепляются на основании дозатора при помощи стальных уголков (рисунок 8).

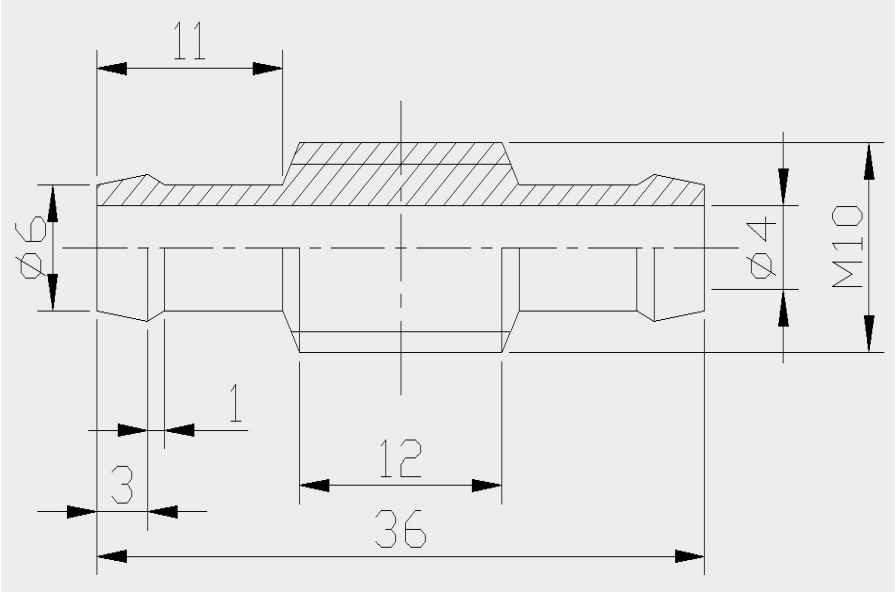


Рисунок 8 – Чертеж штуцеров

ПРИЛОЖЕНИЕ В

В данном приложении приведена программа работы САУ, написанная при помощи логических модулей в среде LOGO soft comfort. Дополнительно здесь размещаются фотографии моделирования процессов.

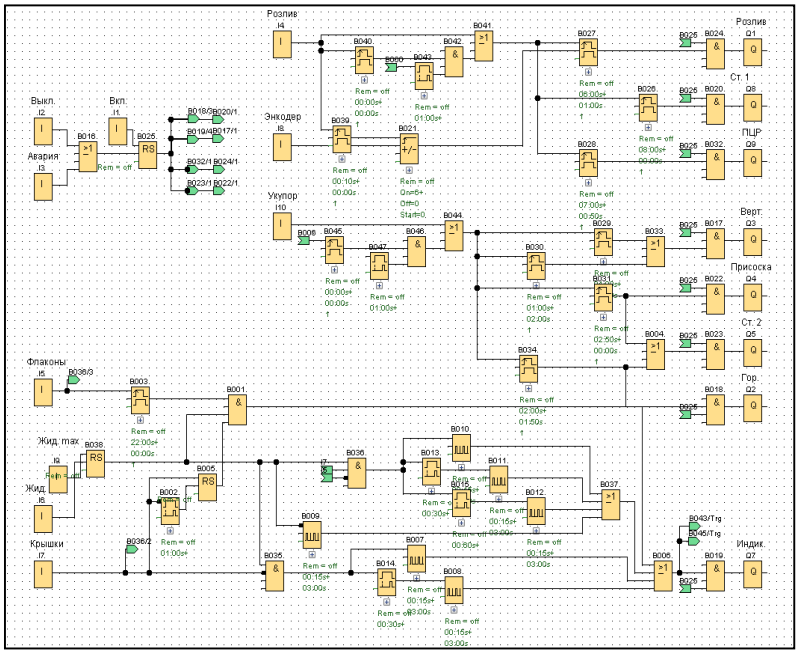
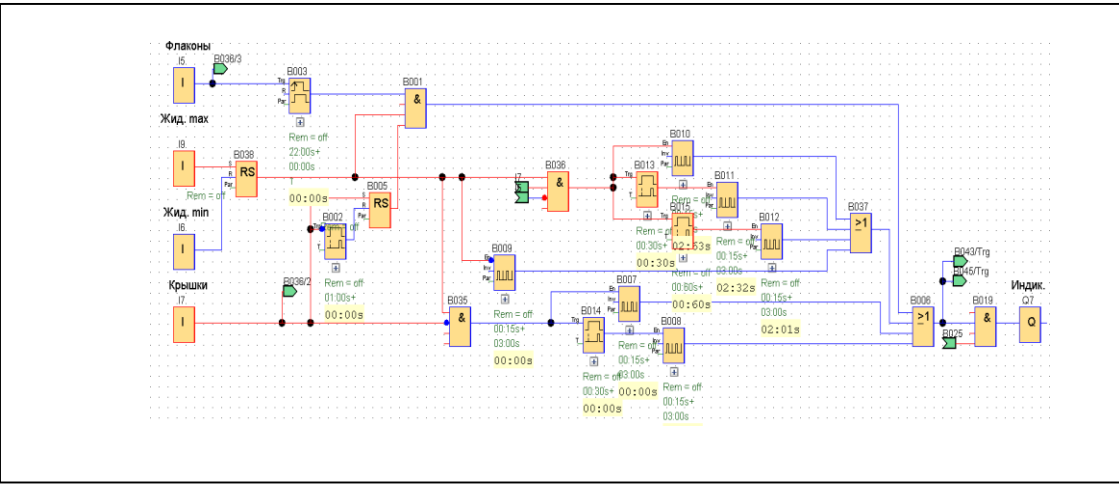


Рисунок 9 – Программа работы АЛР

Рисунок 10 – Моделирование опроса датчиков



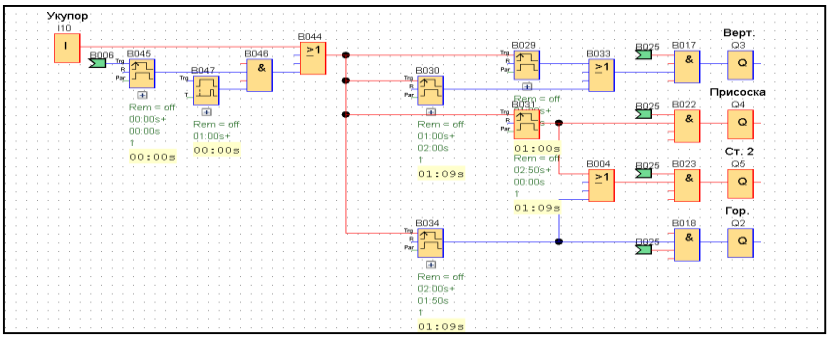


Рисунок 11 – Моделирование программы узла укупора

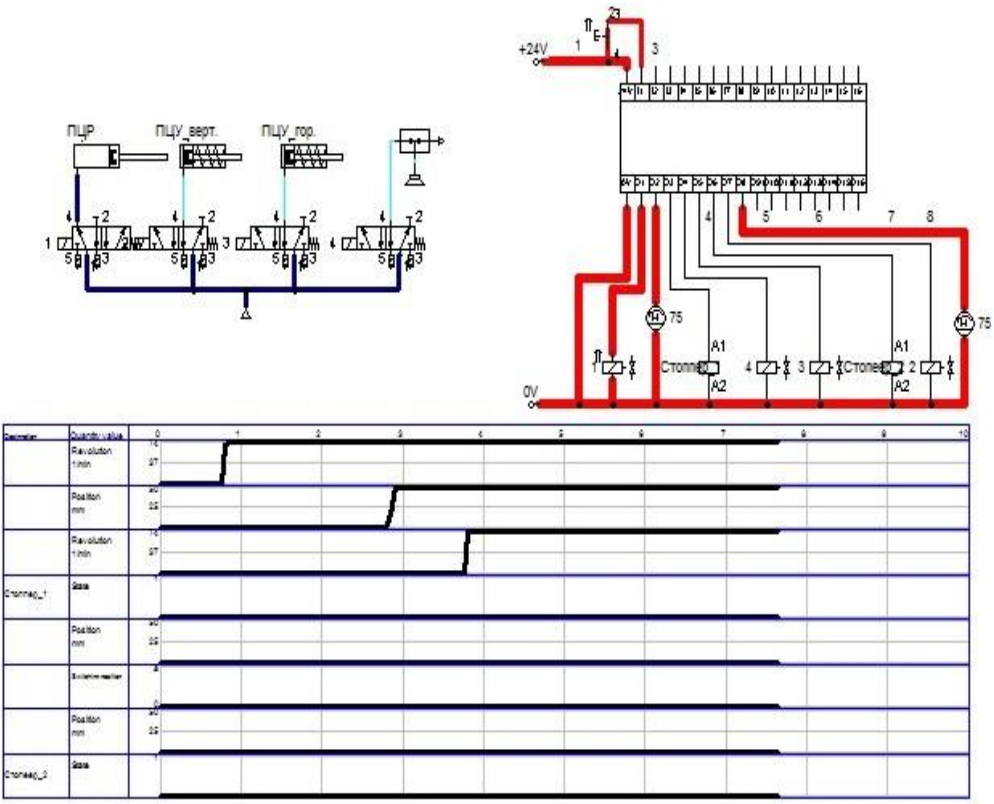


Рисунок 12 – Моделирование пневматической и электрической систем.

Создание циклограммы работы САУ

ПРИЛОЖЕНИЕ Г

В данном приложении приведены изображения макета АЛР и процессов работы над ним.

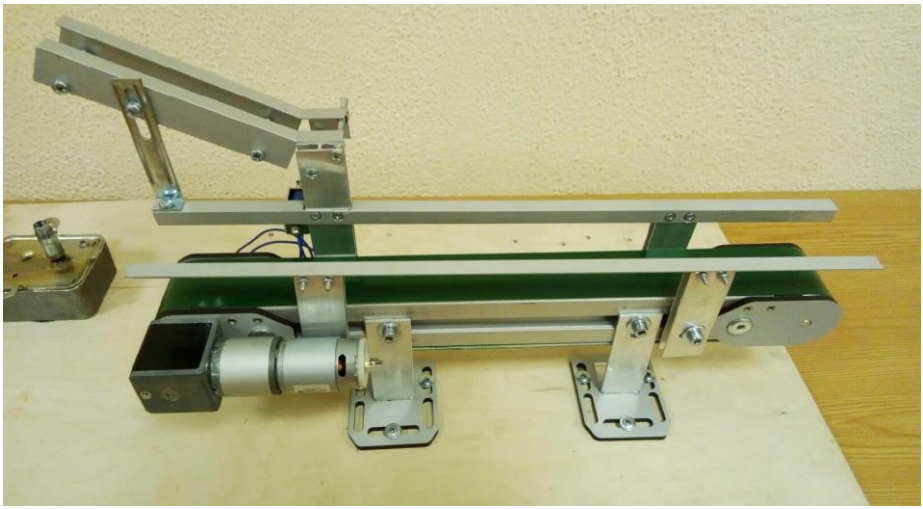


Рисунок 13 – Транспортер

Рисунок 14 – Перемешивающее устройство

Рисунок 15 – Промежуточный этап сборки макета

Рисунок 16 – Макет автоматической линии

Секция XI. Фундаментальные науки

Совершенствование численных методов решения многомерных оптимизационных задач

Симановский Марк Анатольевич

г. Москва, ГБОУ Лицей 1580 при МГТУ им. Н.Э. Баумана, 11 класс

Содержание

1 Введение
2 Цель работы
3 Теоретическая часть
  3.1 О связи производных и оптимумов
    3.1.1 Производная
    3.1.2 Оптимум и алгоритмы его нахождения
4 Основная часть
  4.1 Математическая модель
  4.2 Программная реализация модели
    4.2.1 Язык разработки
    4.2.2 Установка ПО
    4.2.3 Представление числа
    4.2.4 Реализация методов оптимизации
5 Сравнение с классическими методами
  5.1 Тест 1. Производная
  5.2 Тест 2. Оптимизация функции
6 Решение оптимизационных задач
7 Перспективы
8 Заключение
Список литературы
9 Приложения
  9.1 Спуск
  9.2 Характеристики тестируемой машины
  9.3 Логотип работы

Аннотация

В данной работе в качестве способов совершенствования методов оптимизации рассматривается параллелизм вычислительных задач, введение гиперкомплексного двумерного кольца над вещественными числами, расширенного нетривиальным делителем нуля, а также использование таблиц дифференцирования. Уравнение Даламбера-Эйлера, разложение аналитической функции в ряд Тейлора, а также некоторые другие утверждения из дифференциального исчисления и линейной алгебры позволяют определить производную вещественной функции как проекцию функции двумерной алгебры на ось базиса нетривиального делителя нуля. Независимость вычисления некоторых вспомогательных методов дает возможность их одновременного параллельного расчета. Рассмотренные модификации испытываются в задачах поиска производной функции и сравниваются с классическими численными и символьными методами. Они сильно опережают последних по скорости, но практически не отличаются от них по точности. На базе рассмотренных улучшений реализуется численный детерминированный метод первого порядка поиска оптимума функции, совершенствуемый параллелизмом вычислительной задачи. Он тестируется на примере реальных оптимизационных задач.

Рассмотренные модификации позволяют усовершенствовать методы дифференцирования и оптимизации. Они могут быть внедрены для практического применения.

Ключевые слова: Оптимизация, численные методы, символьные методы, детерминированные методы, многомерная оптимизация, математическое программирование, математическое моделирование, python, теория колец, высшая алгебра, математический анализ, теория функций комплексного переменного.

1 Введение

Современное развитие вычислительной техники и интенсивное проникновение математических моделей в различные сферы деятельности человека обеспечивает технический прогресс. В процессе исследования и проектирования моделей обычно ставится задача определения наилучших параметров тех или иных объектов. Задача поиска минимума или максимума целевой функции на заданной области конечномерного пространства, ограниченной набором равенств и/или неравенств, называется задачей оптимизации. Математическое программирование – область математики, которая изучает методы решения задач на оптимизацию с ограничениями. При решении задач оптимизации численными методами возникает ряд трудностей, таких как ошибки вычислительной процедуры. Нередко в процессе вычисления возникают большие погрешности, что снижает эффективность оптимизации. Наиболее часто доставляют неприятности ошибки при округлении, а также при аппроксимации производных разностными выражениями. При программировании компьютерных алгоритмов для минимизации ошибок обычно используются символьные вычисления вместо численных методов, но они слишком медленны, чтобы использовать их, например, для решения задач в реальном времени [7].

2 Цель работы

Цель данной работы заключается в исследовании численных методов решения многомерных оптимизационных задач и разработке способов улучшения рассмотренных методов как по точности, так и по производительности.

3 Теоретическая часть

3.1 О связи производных и оптимумов

3.1.1 Производная

Определение 1. Пусть функция $y = f(x)$ определена на множестве M и существует точка $x_0 : x_0 \in M$. Изменим x_0 на такую величину Δx, не зависящую от x_0, что $x_0 + \Delta x \in M$. Тогда изменение функции $\Delta y = f(x + \Delta x) - f(x)$. Производной функции $f(x)$ в точке x_0 назовем передел отношения изменения функции к изменению его независимого аргумента x, при стремлении изменения к нулю (1).

$$f'_x(x) = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (1)$$

Для обозначения производных принято употреблять несколько различных нотаций [1] (Таблица 1). В данной работе будет использоваться, в основном, нотация Коши.

Таблица 1: Нотации производных

$\frac{df}{dx}$	Нотация Лейбница
f'_x	Нотация Лагранжа
$D_x f$	Нотация Коши

Однако не для каждой точки из области определения функции может существовать производная. Необходимым и достаточным условием ее существования является равенство пределов изменений слева, справа и в точке при существовании каждого предела (2). Условие существования производной называют дифференцируемостью, а процесс нахождения производной - дифференцированием.

$$\exists D_x f(x_0) \Leftrightarrow \lim_{\Delta x \to 0_-} \frac{\Delta y(x_0)}{\Delta x} = \lim_{\Delta x \to 0} \frac{\Delta y(x_0)}{\Delta x} = \lim_{\Delta x \to 0_+} \frac{\Delta y(x_0)}{\Delta x} \quad (2)$$

Определение 2. Назовем функцию элементарной, если ее можно получить с помощью конечного числа арифметических операций и композиций основных (также элементарных) функций.
Основными считаются функции, приведенные в Таблице 2.

Таблица 2: Основные функции

Степенная функция	x^n
Показательная и логарифмическая функции	a^x, lnx
Тригонометрические и обратные им функции	*sinx,cosx,arcsinx...*

Из тригонометрических функций можно рассматривать лишь синус и обратный ему арксинус, а все остальные – выражать через него $\left(tgx = \frac{sinx}{sin(\frac{\pi}{2} - x)}\right)$. Более того, из следствий основного логарифмического тождества и тождества Эйлера все элементарные функции представимы через комбинации экспонент и натуральный логарифм $\left(a^b \equiv e^{b\ln(a)}, \quad sin(x) \equiv \frac{e^{ix} - e^{-ix}}{2i}\right)$, но для удобства косинус также включается в эту таблицу.

Любая элементарная функция непрерывна и дифференцируема на своей области определения [2]. Зная некоторые арифметические правила, связанные с производной и производные основных функций, можно найти производную любой элементарной функции, не прибегая

к определению производной (1) через предел [2]. Перечислим их в Таблице 3.

Таблица 3: Правила дифференцирования

Название	$f(x)$	$D_x f(x)$
Производная константы	C	0
Производная умножения на константу	$C \cdot g(x)$	$C \cdot (D_x g(x))$
Производная суммы	$u(x) + v(x)$	$D_x u(x) + D_x v(x)$
Производная произведения функций	$u(x) \cdot v(x)$	$(D_x u(x)) \cdot v(x) + (D_x v(x)) \cdot u(x)$
Производная композиции функций	$u(v(x))$	$D_v u(v) \cdot D_x v(x)$
Производная обратной функции	$g(x)$	$\frac{1}{D_g x(g)}$
Производная степенной функции	xn	$nxn-1$
Производная показательной функции	ax	$a^x \cdot ln(a)$
Производная логарифмической функции	$log_a(x)$	$\frac{1}{x \cdot ln(a)}$
Производная синуса	$sin(x)$	$cos(x)$
Производная косинуса	$cos(x)$	$-sin(x)$
Производная арксинуса	$arcsin(x)$	$\frac{1}{\sqrt{1-x^2}}$

Рассмотрим основную теорему дифференциального исчисления, связанную с оптимизацией.

Теорема 1. *(Теорема Ферма)*. Пусть функция $f(x)$ на промежутке $(a;b)$ принимает наибольшее (наименьшее) значение в точке $x_0 \in (a;b)$. Тогда $D_x f(x_0) = 0$.

Доказательство. Предположим, что $D_x f(x_0)$ 6= 0. Если функция принимает наибольшее значение в точке x_0 то, справедливо, что $\forall x_1 \in (x_0 - \delta; x_0 + \delta)$ $f(x_0) >$ $f(x_1)$. Из (1) следует, что $D_x f(x_0) = \lim_{x - x_0 \to 0} \frac{f(x) - f(x_0)}{x - x_0}$. Если $D_x f(x_0) > 0$, то $f(x_1) - f(x_0) > 0, x_1 > x_0$, откуда следует, что $f(x_0) < f(x_1)$, что противоречит условию наибольшего значения. Аналогично и с другим знаком.

Теорема 2. *(Теорема Ролля)*. Пусть функция $f(x)$ на промежутке $[a;b]$ непрерывна и дифференцируема, а $f(a) = f(b)$. Тогда существует такая точка $c \in (a;b)$, что $D_x f(c) = 0$.

Доказательство. По второй теореме Вейерштрасса, функция достигает своего наибольшего и наименьшего значения на отрезке. Пусть $f_{наиб} = M$, а $f_{наим} = m$. Если $m = M$, то функция стационарна на всем отрезке, значит производная в любой точке отрезка равна нулю. Тогда рассмотрим $M > m$ ($m > M$ не возможно по определению). Пусть $f(x_0) \in \{m, M\}, x_0 \in (a;b)$. Значит по теореме Ферма $D_x f(x_0) = 0$.

Теорема 3. *(Теорема Лагранжа)*. Пусть функция $f(x)$ непрерывна и дифференцируема на отрезке $[a;b]$. Тогда существует такая точка $c \in (a;b)$, что $D_x f(c) = \frac{f(b) - f(a)}{b - a}$.

Доказательство. Пусть $g(x) = f(x) - f(a) - \frac{f(b) - f(a)}{b - a}(x - a)$. Тогда $g(a) = g(b) = 0$, функция дифференцируема и принимает одинаковое значение на концах отрезка, значит по теореме Ролля существует точка $c \in (a;b)$, что $D_x g(c) = 0$. $D_x g(x) =$

$$D_x\left(f(x) - f(a) - \frac{f(b) - f(a)}{b - a}(x - a)\right) = D_x f(x) - \frac{f(b) - f(a)}{b - a},$$

следовательно, верно (3).

$$D_x f(c) = \frac{f(b) - f(a)}{b - a} \quad (3)$$

Формулу (3) также называют формулой конечных приращений и используют для численных вычислений производных, устремляя разницу между b и a.

Так как производная элементарной функции является элементарной функцией[1], то она тоже может иметь производную. Производная от производной называется производной второго порядка. $D_x(D_x f(x)) = D_x 2f(x)$. До сих пор рассматривалась унарная функция y : D(y) → E(y), но очень часто на практике зависимая переменная может быть связана не с одной, а с несколькими переменными[3]. Такая функция $y = f(x_1,x_2,...,x_n)$ называется функцией многих переменных. Для них работают теоремы дифференциального исчисления функций одной переменной, но добавляются новые понятия, такие как частные приращения.

Определение 3. Частным приращением по независимой переменной x_i функции многих переменных $\phi(x_1,x_2,...,x_i,...,x_n)$ в точке (x_{10}) назовем изменение функции при изменении переменной x_i $\Delta\phi = \phi(x_{10},x_{20},...,x_{i0} +\Delta x_i,...,x_{n0})-\phi(x_{10},x_{20},...,x_{i0},...,x_{n0})$. Исходя из этого определим частную производную как (4).

$$D_{x_i} f(x_{1_0},...,x_{i_0},...,x_{n_0}) = \lim_{\Delta x_i \to 0} \frac{f(x_{1_0},x_{2_0},...,x_{i_0}+\Delta x_i,...,x_{n_0}) - \varphi(x_{1_0},x_{2_0},...,x_{i_0},...,x_{n_0})}{\Delta x_i} \quad (4)$$

Определение 4. Градиентом функции $f(x_1,x_2,...,x_n)$ назовём вектор размерностью n, состоящий из частных производных по соответствующим переменным.

$$\nabla f(x_1,x_2,...,x_n) = (D_{x1}f(x_1,x_2,...,x_n),D_{x2}f(x_1,x_2,...,x_n),...,D_{xn}f(x_1,x_2,...,x_n))$$

Следует учитывать, что x_j относительно x_i считается постоянной, так как не зависит от нее, потому и $D_{xi}(x_j) = 0$ (для $1 \le i,j \le n, i \neq j$). Также возможно брать производную функции по разным переменным. $D_{xy}f(x,y) = D_x(D_y f(x,y))$. Для непрерывных элементарных функций смешанные производные равны: $D_{xy}f = D_{yx}f$ [3].

[1] Считаем, $f(x) = 0$ тоже элементарная функция

3.1.2 Оптимум и алгоритмы его нахождения

Определение 5. Оптимизацией называется процесс поиска экстремального значения заданной функции на данном множестве. Оптимум - набор независимых переменных, значение функции от которых экстремально.

Существует достаточно много методов поиска оптимумов. Классифицируется методы по случайности результата (детерминированные и стохастические), по способу получения результата (аналитические и численные), по необходимости дифференцировать заданную функцию (прямые - производные не используются; методы первого порядка - используется производная первого порядка, градиент; методы второго порядка - используются производные второго порядка, гессиан). В данной работе рассматриваются детерминированные численные прямые методы и методы первого порядка.

Будем считать, что необходимо оптимизировать функцию $F(\rightarrow -x)$ на множестве M^n, где $\rightarrow -x$ - вектор ее независимых переменных $(x_1,x_2,...,x_n) \in M^n$. Тогда пусть задача нахождения оптимума записывается одним из следующих вариантов:

$$\overrightarrow{x_0} = argmin_{x \in \mathbb{M}^n} F(\overrightarrow{x}) - ?$$

$$\overrightarrow{x_0} : F(\overrightarrow{x_0}) \overset{\mathbb{M}^n}{\longmapsto} min = ?$$

Градиентный метод

Рассмотрим алгоритм любого градиентного метода.

− 1. Выберем случайную точку из множества и подведем к ней радиус-вектор $x^0 \in M^n$, точность вычисления ε и множитель шага λ. Определим $i = 1$.

2. Вычислим $\overrightarrow{x^i} = \overrightarrow{x^{i-1}} - \lambda \nabla F(\overrightarrow{x^{i-1}})$.

3. Если $||\overrightarrow{x^i} - \overrightarrow{x^{i-1}}|| < \varepsilon$, то $\overrightarrow{x^i}$ - оптимум, иначе необходимо увеличить *i* на единицу и вернутся к п.2.

Представив идеальную точность ($\varepsilon = 0$) получаем Теорему Ферма: производная функции в точке, где функция принимает экстремальное значение, равна нулю. Но так как иррациональное число возможно представить на ЭВМ как его рациональное приближение, при работе с числом накапливается ошибка и падает точность - для этого условием остановки алгоритма поиска является неравенство.

В то время как выбор начальной точки на сегодняшний день может улучшаться стохастическими методами, алгоритм градиентного спуска также может быть улучшен правильным выбором множителя шага. Когда выбирается наилучший шаг на каждой итерации, градиентный алгоритм называется методом наискорейшего спуска [4].

Для его поиска на каждом шаге необходима следующая одномерная оптимизация:

$\lambda^i = argmin\mathrm{R}0+F(\overrightarrow{x^{i-1}} - \lambda^{i-1}\nabla F(\overrightarrow{x^{i-1}}))$. Проводить ее можно как прямым способом, например, методом дихотомии, так и методом прямого порядка.

4 Основная часть

4.1 Математическая модель

Утверждение 1. Пусть $\Phi < R^2,+,\times >$ - двумерная алгебра над полем действительных чисел, для которой верны следующие утверждения:

$$\exists!\theta \in \Phi : \forall a \in \Phi \; a + \theta = \theta + a = a \quad (5)$$

$$\forall a \in \Phi \; \exists! b \in \Phi : a + b = b + a = \theta \quad (6)$$

$$\exists!\epsilon \in \Phi : \forall a \in \Phi \; a \times \epsilon = \epsilon \times a = a \quad (7)$$

$$\forall a, b \in \Phi \; a + b = b + a = c : c \in \Phi \quad (8)$$

$$(9)$$

$$\forall a, b, c \in \Phi \; (a + b) + c = a + (b + c) \quad (10)$$

$$\forall a, b \in \Phi \; a \times b = b \times a = c : c \in \Phi \quad (11)$$

$$\forall a, b, c \in \Phi a \times (b + c) = a \times b + a \times c \quad (12)$$

$$\exists \delta \in \Phi : \delta \neq \theta \wedge \delta \times \delta = \theta \quad (13)$$

$$\forall a \in \Phi : a = x + \delta y, \; x, y \in \mathbb{R} \quad (14)$$

$$\forall a = x + \delta y \in \Phi, \lambda \in \mathbb{R} \; \lambda a = \lambda x + \delta \lambda y \quad (15)$$

$$\forall n = a + \delta b, m = c + \delta d \in \Phi \; n + m = (a + c) + \delta(b + d) \quad (16)$$

$$\forall n = a + \delta b, m = c + \delta d \in \Phi \; n \times m = (a \cdot c) + \delta(a \cdot d + c \cdot b) \quad (17)$$

$$\theta = 0 + \delta \cdot 0, \epsilon = 1 + \delta \cdot 0$$

Тогда Φ является кольцом над полем вещественных чисел с тремя особыми элементами: θ нейтрален по сложению, нейтрален по умножению, δ является гиперкомплексной единицей - нетривиальным делителем нейтрального по сложению элемента. Каждый элемент кольца - гиперкомплексное число, представимое алгебраически как разложенное по двум базисам, что равнозначно паре (вектору размерности 2):

$$a \in \Phi \equiv x + \delta y \equiv \begin{pmatrix} x \\ y \end{pmatrix}, \; x, y \in \mathbb{R}$$

.

Рассмотрим уравнение Даламбера-Эйлера для комплексного переменного $z = x+\delta y$ (18). Тогда из его следствия[5] о производной функции комплексной переменной верно (21)

$$\omega(z) = u(x,y) + \delta v(x,y) \quad (18)$$

$$\nabla = \frac{\partial}{\partial x} + \delta\frac{\partial}{\partial y} \quad (19)$$

$$\nabla\omega = \left(\frac{\partial}{\partial x} + \delta\frac{\partial}{\partial y}\right)\cdot\left(u(x,y) + \delta v(x,y)\right) = D_x u(x,y) + \delta\left(D_x v(x,y) + D_y u(x,y)\right) \quad (20)$$

$$D_z\omega = \nabla\omega = D_x u(x,y) + \delta D_y u(x,y) \quad (21)$$

Пусть функция $\phi(x), x \in R$ непрерывна и дифференцируема в точке a и ее окрестности. Рассмотрим ее разложение в ряд Тейлора (22). Заменим действительный аргумент x на гиперкомплексный $z = x + \delta$. Воспользуемся биномом Ньютона, чтобы раскрыть сумму в степени k (24). В силу утверждений построения алгебры Φ ($\lambda\delta^n = 0, \forall n >= 2, n.\lambda \in R$) верно (25). Тогда из (21), (22) и (25) следует (26). Раскроем сумму и заметим сходства с (22). Таким образом сумма сумм является суммой функции и ее производной, умноженной на гиперкомплексную единицу (27).

$$\varphi(x) = \sum_{k=0}^{\infty} D_{x^k}\varphi(a)\frac{(x-a)^k}{k!} \quad (22)$$

$$\varphi(z)=\sum_{k=0}^{\infty}D_{z^k}\varphi(z)\frac{((x-a)+\delta)^k}{k!} \tag{23}$$

$$((x-a)+\delta)^k=\sum_{i=0}^{k}C_k^i(x-a)^{k-i}(\delta)^i=(x-a)^k+k(x-a)^{k-1}\delta+C_k^2(x-a)^{k-2}(\delta)^2+\dots \tag{24}$$

$$((x-a)+\delta)^k=(x-a)^k+k(x-a)^{k-1}\delta \tag{25}$$

$$\varphi(z)=\sum_{k=0}^{\infty}\frac{1}{k!}D_{z^k}\varphi(z)\left((x-a)^k+k(x-a)^{k-1}\delta\right)= \tag{26}$$

$$=\sum_{k=0}^{\infty}D_{z^k}\varphi(z)\frac{(x-a)^k}{k!}+\delta\sum_{k=1}^{\infty}D_{z^k}\varphi(z)\frac{(x-a)^{k-1}}{(k-1)!}$$

$$\varphi(z)=\varphi(x+\delta)=\varphi(x)+\delta D_x\varphi(x) \tag{27}$$

Вычтем из (27) значение функции в точке a и поделим на δ. Получается уравнение, схожее с (1):

$$D_x\varphi(x)=\frac{\varphi(x+\delta)-\varphi(x)}{\delta}\equiv\lim_{\Delta x\to 0}\frac{\varphi(x+\Delta x)-\varphi(x)}{\Delta x}$$

«Совпадение» интуитивно объяснимо тем, что Δx при стремлении к нулю и так мало, а в квадрате уже настолько пренебрежимо, что $(\Delta x)^2 \approx 0$. Однако совпадения нет, так как нет и уравнения, где δ стоит в знаменателе. Так как Φ - кольцо, то обратного элемента для делителя нуля не определено. Возможность умножения на элемент, обратный делителю нуля (деление на него) существует в колесах[6]. Расширение кольца до колеса лежит в перспективе развития данной работы. Но решением данной проблемы является равносильность записей гиперкомплексного числа в алгебраическом и векторном виде (28):

$$f(x+\delta)=f(x)+\delta D_x f(x)\equiv f\left(\begin{pmatrix}x\\1\end{pmatrix}\right)=\begin{pmatrix}f(x)\\D_x f(x)\end{pmatrix} \tag{28}$$

при следующем нижеследующем утверждении.

Утверждение 2. Функция от вектора есть вектор со следующей функцией

$$f\left(\begin{pmatrix} x \\ k \end{pmatrix}\right) = \begin{pmatrix} f(x) \\ \alpha(f(x), k) \end{pmatrix},$$

где α - некоторое дифференциальное отображение.

Тогда производную в точке можно определить из выбора второй координаты векторазначения функции от вектора-переменной из точки и гиперкомплексной единицы. Выбор можно осуществить математически, как скалярное произведение вектора-значения функции с вектором $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ или умножением матрицы $\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ на вектор-значение функции.

Очевидно, что использовать скалярное произведение выгодней: сложность вычисления результата произведения квадратных матриц размерностью n составляет $O(n^3)$ по определению или $O(n^{\log_2 7})$ по алгоритму Штрассена и требует $O(n^2)$ памяти, в то время как скалярное произведение двух векторов требует $O(n)$ времени, а это операция, выполняемая для каждой ячейки матрицы (всего n^2 раз), и требует лишь $O(n)$ памяти.

Тогда производная функции с помощью введения кольца Φ определяется по уравнению

$$D_x f(x) = \left\langle f\left(\begin{pmatrix} x \\ 1 \end{pmatrix}\right), \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle_{(29)}$$

Так как скалярное произведение двух векторов представимо, как $h\vec{a}, \vec{b}i = |a| \cdot |b| \cdot \angle\cos(\vec{a}, \vec{b})$, а длина проекции вектора $\vec{b}$ на вектор $\vec{a}$ есть $pr_b a = |a| \angle\cos(\vec{a}, \vec{b}) = \frac{\langle \vec{a}, b \rangle}{|a| \cdot |b|} \cdot |a| = \frac{\langle \vec{a}, b \rangle}{|b|}$, то геометрический смысл производной во $\vec{b}$ веденной алгебре - проекция функции на мнимую ось.

Рассмотрим в трехмерном пространстве XYZ множество всех таких точек P =

$$\begin{pmatrix} t \\ f(t) \\ D_t f(t) \end{pmatrix} \in \mathbb{R} \times \Phi$$

для любого $t \in$ R и их проекции. Для примера положим $f(x) = x^2 sin(x)$. График множества точек P - (рис. 1). Заметим, что множество точек $\begin{pmatrix} t \\ f(t) \end{pmatrix} \forall t \in$ R - параметрическая запись функции $y = f(x)$, тогда проекция всех точек P на плоскость XY - все точки функции $f(x)$ (рис. 2).

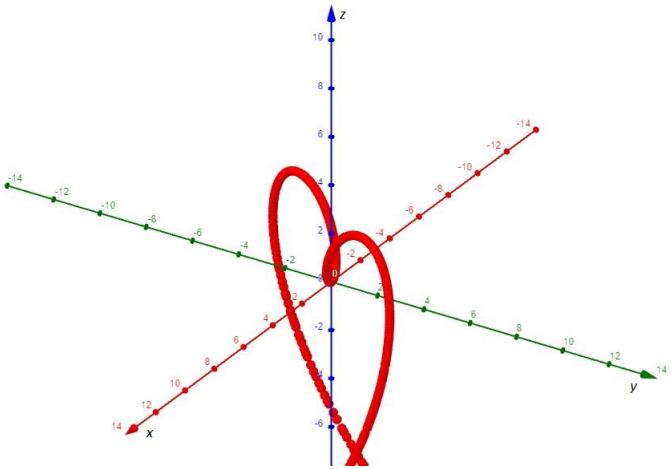


Рис. 1: Множество точек P. Функция в пространстве R × Φ.

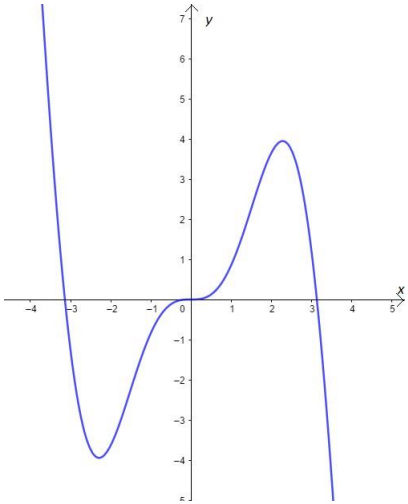


Рис. 2: Множество проекций точек P на плоскость XY. Функция в поле R.

Аналогично $\begin{pmatrix} t \\ D_t f(t) \end{pmatrix} \ \forall t \in \mathbb{R}$ - параметрическая запись функции $y = D_x f(x)$, тогда проекция всех точек P на плоскость *XZ* - все точки производной функции $f(x)$ (Рис. 3).

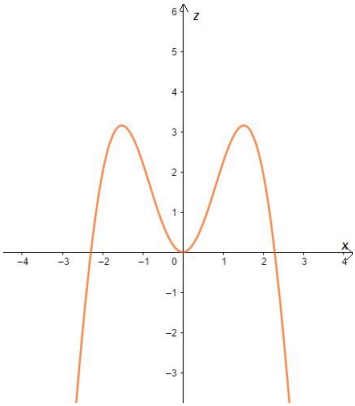


Рис. 3: Множество проекций точек P на плоскость XZ. Производная функции в поле R.

На следующем рисунке (рис. 4) изображены проекции всех точек P на плоскость YZ. Каждая координата задается лишь значением функции для некого аргумента и значением производной от того же аргумента. Учитывая принятые утверждением 2 обозначения и аналогию с предыдущими примерами, множество таких точек - это диаграмма функции в кольце *Ф*. Следует отметить, что ее нельзя назвать графиком функции, так как полученное множество рассчитывается для всех значений аргумента, ось которого перпендикулярна плоскости проекции, а значит, теряется условие одно-многозначного соответствия.

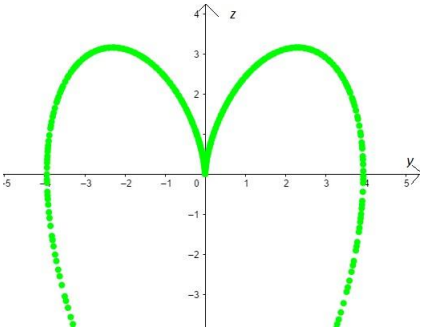


Рис. 4: Множество проекций точек P на плоскость YZ. Функция в кольце Φ.

Тогда в одном пространстве все точки и их проекции выглядят как на рис. 5.

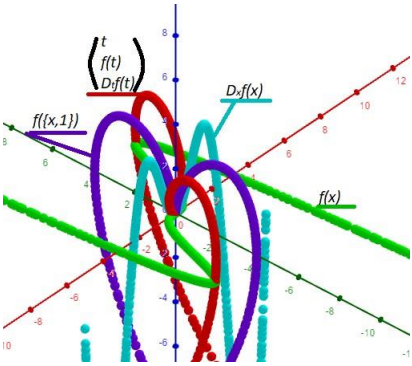


Рис. 5: Множество точек P и их проекций.

Теперь рассмотрим многомерный случай. Пусть задана функция $f(x,y,z)$, требуется найти ее градиент. Из «совпадения», полученного из (27), следует, что приращивается та переменная, вторая координата вектора которой равна 1. Назовем для удобства $snd = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Тогда выражение для градиента выглядит, как

$$\nabla f(x,y,z) = (\left\langle f\left(\begin{pmatrix}x\\1\end{pmatrix},\begin{pmatrix}y\\0\end{pmatrix},\begin{pmatrix}z\\0\end{pmatrix}\right),\overrightarrow{snd}\right\rangle \left\langle f\left(\begin{pmatrix}x\\0\end{pmatrix},\begin{pmatrix}y\\1\end{pmatrix},\begin{pmatrix}z\\0\end{pmatrix}\right),\overrightarrow{snd}\right\rangle, \left\langle f\left(\begin{pmatrix}x\\0\end{pmatrix},\begin{pmatrix}y\\0\end{pmatrix},\begin{pmatrix}z\\1\end{pmatrix}\right),\overrightarrow{snd}\right\rangle),$$

(30)

Пример 1.

$f(\overrightarrow{x}) = x^2 - 2,\ D_x f$ - ?

$D_x f = \langle f(x+\delta), snd\rangle = \langle (x+\delta)^2 - 2, snd\rangle = \langle x^2 + 2x\delta + \delta^2 - 2, snd\rangle = \langle x^2 + 2x\delta - 2, snd\rangle =$

$\left\langle \begin{pmatrix} x^2 - 2 \\ 2x \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle = 2x$

$D_x f = 2x$

Пример 2. $f(x,y) = x^2 - y^3$, ∇f - ?

$$D_x f(x,y) = \langle f(x+\delta, y), snd\rangle = \langle (x+\delta)^2 - y^3, snd\rangle = \langle x^2 + 2x\delta + \delta^2 - y^3, snd\rangle = \langle (x^2 - y^3) + (2x)\delta, \overrightarrow{snd}\rangle = \left\langle \begin{pmatrix} x^2 - y^3 \\ 2x \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle = 2x$$

$$D_y f(x,y) = \langle f(x, y+\delta), \overrightarrow{snd}\rangle = \langle x^2 - (y+\delta)^3, \overrightarrow{snd}\rangle = \langle x^2 - y^3 - 3y\delta^2 - 3y^2\delta + \delta^3, \overrightarrow{snd}\rangle = \langle (x^2 - y^3) + (3y^2)\delta, \overrightarrow{snd}\rangle = \left\langle \begin{pmatrix} x^2 - y^3 \\ 3y^2 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle = 3y^2$$

$\nabla f = (2x, 3y^2)$

Рассмотрим дифференцирование композиции двух функций с помощью введенного кольца. Пусть $y = f(g(x))$, тогда из Таблицы 3: $D_x y = D_g f \cdot D_x g$. Из (28) следует, что $\alpha(f(x),1) = D_x f(x)$, пусть тогда $\alpha(f,g) = Df \cdot g$. Заметим, что $(f(g(x + \delta))) =$

$$f\left(g\left(\begin{pmatrix}x\\1\end{pmatrix}\right)\right) = f\left(\begin{pmatrix}g(x)\\D_x g(x)\end{pmatrix}\right) = \begin{pmatrix}f(g(x))\\D_g f(g)\cdot D_x g(x)\end{pmatrix} = \begin{pmatrix}f(g(x))\\D_x f(g(x))\end{pmatrix} = \begin{pmatrix}y\\D_x y\end{pmatrix}$$

. Тогда для вычисления производной композиции двух функций нет необходимости вычислять отдельно производные этих функций и умножать их, достаточно лишь «правильно» применять функцию к вектору. Производная композиции выражается уравнением (31).

D **−→E**

$D_x f(g(x)) =$ $f(g(x+\delta)),snd$ (31)

Отсюда следует еще одно важное свойство. До этого мы вычисляли производную практически по определению - как разность функции от переменной с приращением и без. Воспользуемся определением элементарной функции. Тогда будем находить производную основной функции по правилу таблицы дифференцирования, а элементарной неосновной - по правилу композиции. Попробуем найти по определению производную синуса: $sin(x+\delta) - sin(x) = sin(x)cos(\delta) + sin(\delta)cos(x) - sin(x) = sin(x)(cos(\delta) - 1) + sin(\delta)cos(x)$. Ничего определенного. Применим тождество Эйлера: $sin(x+\delta)-sin(x) =$

$$\frac{e^{i(x+\delta)} - e^{-i(x+\delta)} - e^{ix} + e^{-ix}}{2i} = \frac{e^{ix}e^{i\delta} - e^{-ix}e^{-i\delta} - e^{ix} + e^{-ix}}{2i} = \frac{e^{ix}(e^{i\delta} - 1) - e^{ix}(e^{-i\delta} - 1)}{2i}$$

.

Учитывая, что экспонента по формуле Тейлора раскрывается как $1 + x + \frac{x}{2} + \frac{x}{6} + \ldots$,

$$e^{i\delta} = 1 + i\delta - \frac{\delta}{2} + \frac{-i\delta}{6} + \ldots = 1 + i\delta$$

$$sin(x) = \frac{e^{ix}(e^{i\delta} - 1) - e^{ix}(e^{-i\delta} - 1)}{2i} = \frac{e^{ix}(1 + i\delta - 1) - e^{ix}(1 - i\delta - 1)}{2i} = \delta\frac{e^{ix}i + e^{-ix}i}{2i} =$$

а $\delta^2 = 0$, подставим:. Тогда $sin(x + \delta) -$

$\delta\frac{e \quad + e}{2} = cos(x)\delta$. Получилось довольно громоздко. А теперь вычислим с использованием предопределенных таблиц:

$$D_x sin(x) = \left\langle sin\left(\begin{pmatrix} x \\ 1 \end{pmatrix}\right), \overrightarrow{snd} \right\rangle = \left\langle \begin{pmatrix} sin(x) \\ \alpha(sin(x), 1) \end{pmatrix}, \overrightarrow{snd} \right\rangle = \left\langle \begin{pmatrix} sin(x) \\ cos(x) \end{pmatrix}, \overrightarrow{snd} \right\rangle = cos(x)$$

Тот же результат, но при использовании таблиц, мы получили за меньшее количество действий, от чего при использовании рациональных приближений ответ, очевидно, получается с меньшей погрешностью.

Пример 3. $u(v) = \sin v, v(x) = x^3$, $D_x u$ - ?

$$D_x u = \quad sin \quad \begin{pmatrix} x \\ 1 \end{pmatrix}^3 \quad , \overrightarrow{snd} \quad = \left\langle sin\left(\begin{pmatrix} x^3 \\ \alpha(x^3, 1) \end{pmatrix}\right), \overrightarrow{snd} \right\rangle = \left\langle sin\left(\begin{pmatrix} x^3 \\ 3x^2 \end{pmatrix}\right), \overrightarrow{snd} \right\rangle =$$

$$\left\langle \begin{pmatrix} sin(x^3) \\ \alpha(sin(x^3), 3x^2) \end{pmatrix}, \overrightarrow{snd} \right\rangle = \left\langle \begin{pmatrix} sin(x^3) \\ cos(x^3) \cdot 3x^2 \end{pmatrix}, \overrightarrow{snd} \right\rangle = 3x^2 cos(x^3)$$

* ! +

4.2 Программная реализация модели

4.2.1 Язык разработки

Языком реализации выбран python постольку поскольку он поддерживает замыкания, длинную арифметику и комплексные числа «из коробки», а также для него есть удобные инструменты тестирования (python magic) и построения графиков (matplotlib). Кроме того существует библиотека символьных вычислений SymPy, с которой удобно будет сравнивать результат.

4.2.2 Установка ПО

Для установки интерпретатора языка python и интерактивной оболочки Jupyter достаточно выполнить следующую команду в консоли на *nix системе с пакетным менеджером APT

```
1        sudo apt install python3
2        sudo apt install ipython3 ipython3−qtconsole
```

После установим необходимые библиотеки: NumPy для работы с линейными пространствами, matplotlib для построения графиков, SymPy для символьных вычислений

```
1        pip3    install numpy
2        pip3    install matplotlib
3        pip3    install sympy
```

4.2.3 Представление числа

Описанная алгебра Φ может быть определена не только над полем вещественных чисел. Возможна практически любая алгебраическая структура, которую возможно описать императивно. Потому создадим класс Numer, описывающий базовое число, над которым строится алгебра. В данном случае алгебру строим над классом float - одномерными рациональными числами с плавающей запятой.

Так как $\Phi < F^2,+,\times >$ - двумерная алгебра, то каждое число задается несколькими значениями. Для универсальности опишем вектор, он также понадобится для работы с многомерными функциями. Определим сложение векторов и их скалярное произведение. Теперь опишем само кольцо в классе HCRA(hyper complex ring algebra) на основе утверждений (5 – 17). В классе DCalculus реализуем таблицу производных, а также публичные методы с замыканиями дифференцирования функции: d(f)(x) и grad(f)(x) с обработкой аргументов в функциональном стиле.

```
1                def d( f ) :
2                def f_(x) :
3                return f ( DCalculus (x ,1) ) . I ()
4                return f_
5
6        def full_derivative ( f ,    args ) :
```

```
7           return f (∗map(lambda p: DCalculus (p,1) , args ) ) . I
            () 8 def partial_derivative ( f , args , i ) :
9               return f (∗map(lambda j :   DCalculus ( j [1] , int ( j [0]
                == i ) ) ,
                  enumerate( args ) ) ) . I ()
10              def grad ( f ) :
11              def grad_( args ) :
12              return tuple( f (∗map(lambda j : DCalculus ( j [1] , int
                ( j [0] == i ) ) , enumerate( args ) ) ) . I () for i ,p in
                enumerate( args ) )
13              return grad_
```

4.2.4 Реализация методов оптимизации

Реализуем метод наискорейшего градиентного спуска. Будем оптимизировать λ одновременно двумя способами: прямой дихотомией и найдя нуль производной функции (с использованием теоремы Ферма). В алгоритме Ньютона-Рафсона используется отношение функции к ее производной. Вспомним формулу 28. Таким образом, не производя скалярного произведения мы получаем и значение функции в точке, и значение ее производной, таким образом экономим вычислительное время не выполняя лишних расчетов. Хоть и в общем случае алгоритм поиска корня Ньютона-Рафсона сходится быстрее дихотомии, он может и не сойтись вовсе, для этого вместе с ним и работает другой алгоритм из другого класса (прямая оптимизация вместо первого порядка). Запустить расчёт двух функций одновременно можно с помощью механизма параллелизма multiprocessing. Для этого замкнем функции линейной оптимизации, чтобы они не вычислялись раньше времени, например, при создании списка их них, передадим функции imap unordered как второй аргумент, а как первый - конструкцию вызова размыкания lambda x: x(). imap является итерирующей, потому взять значение функции, которая вычислилась раньше другой просто:

```
1           res = 0
```

```
2    lo = lambda l :    f (*tuple(map(lambda x , y : x-l *y , point_0 , nabla_0
         ) ) )
3    for k in imap_unordered(lambda x : x() ,
4    [ dichotomy_opt( lo , domain=(0,1000) , eps = eps ) , 5 newtonraphson(d( l0 ) , domain [0] , eps=eps **0.5) ]) :
6        res = k
7        break
```

5 Сравнение с классическими методами

5.1 Тест 1. Производная

Найдем значения производных функций в некоторых точках с помощью метода конечных приращений, символьного вычисления и введения кольца, измерим точность и время работы каждого метода и занесем данные в таблицу. Функцию запишем следующим образом:

```
1                    def f (x) : return x**2 - 2*x + 1
```

Из классических методов можно отметить системы символьных вычислений, отличающихся наилучшей точностью, так как оперируют математическими символами, а не десятичными приближениями, и численный метод, основанный на теореме Лагранжа о конечных приращениях, выделяющийся скоростью вычислений, так как требует лишь вычисления значения функции в двух точках. Следует отметить заранее, что оригинальная теорема подверглась изменению для оптимизации точности (32), однако все равно для быстро растущих функций остается неустойчивой [9], а параметр eps подбирался в ходе эксперимента таким образом, чтобы обеспечить наименьшую разницу со значением, полученным от численной подстановки в функцию, полученную символьными методами в пределах одного класса чисел float.

$$D_x f \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x},\ |\Delta x| << 1 \quad (32)$$

Метод конечных приращений приведен ниже [Листинг 1], символьное вычисление производной - [Листинг 2], вычисление производной в кольце - [Листинг 3]. Все имеют одинаковую сигнатуру, оформленную в виде замыканий: производная в точке x вычисляется как d(f)(x).

Листинг 1: FiniteAdditions.py

```
1
d
e
f
d
(
f
)
:
2
e
p
s
=
1
e
–
7
#
3
d
e
f
f
_
(
x
)
```

```
4        return ( f (x+eps ) − f (x−eps ) ) /(2∗ eps )
5        return f_
```

Листинг 2: SymbDerivative.py

```
1    from sympy import ∗
2    x = symbols ( 'x ' ) 3 f = . . .
     #определение функции
4        def d( f ) :
5        def f_(v) :
6        return f . diff (x) . subs ({x : v}) .n()
7        return f_
```

Листинг 3: Differntial.py

```
1        . . .
2        def d( f ) :
3        def f_(x) :
4        return f ( DCalculus (x ,1) ) . I ()
5        return f_
6        . . .
```

Испытывались два вида функций: алгебраические и трансцендентные в случайных точках, в таблицу занесен средний результат на выборке из 10000 испытаний на каждый аргумент для уменьшения погрешности, испытания проводились на ЭВМ с характеристиками из [Приложения 9.2].

Таблица 4: Фрагмент таблицы "Результаты вычисления производных. Время, μs."

$f(x)$	x	Конечные приращения	Символьные вычисления	Кольцо Φ

$x^2 - 6x + 9$	4	0.870	168	36.8
	127.5	0.733	195	36.7
	95412	1.010	170	36.9
$sin(x) - e^x$	-2	0.764	445	17.7
	−150.94	0.765	198	17.7
	418.21	0.781	241	17.7

Таблица 5: Фрагмент таблицы "Результаты вычисления производных. Значения."

$f(x)$	x	Конечные приращения	Симв. вычисления	Кольцо Φ
$x^2 - 6x + 9$	4	2.0000001299536 05	2.0	2.0
	127.5	248.99996788008139	249.00	249.0
	95412	190830.23071289062	190818.00	190818.0
$sin(x) - e^x$	−2	-0.5514820822227762	-0.551482119783755	-0.5514821197837552
	-150.94	0.9897139521819476	0.989714003690002	0.9897140036900024

	418 .21	- 4.22956200168 18725e+181	- 4.2295608431 3071e+181	- 4.229560843130 7084e+181

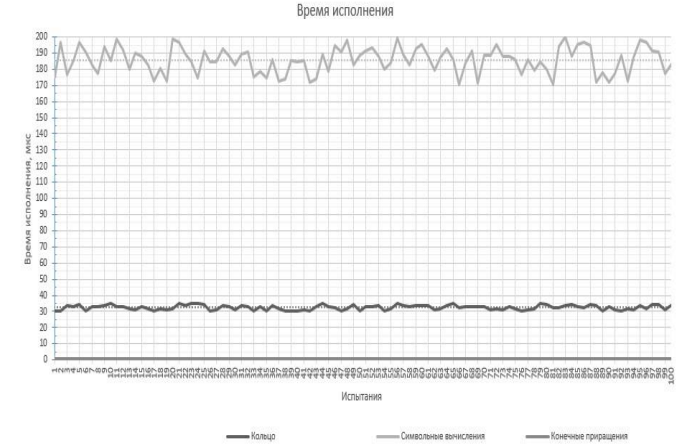


Рис. 6: Время вычисления значения производной алгебраической функции в точке.

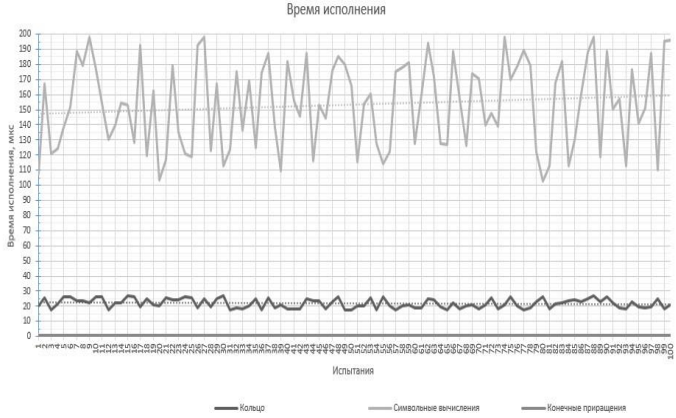


Рис. 7: Время вычисления значения производной трансцендентной функции в точке.

Благодаря использованию таблиц дифференцирования производные элементарных трансцендентных функций вычисляются быстрее. Более низкая скорость работы вычисления производной алгебраической функции объясняется тем, что изначально программа не знает, является ли показатель степени нестационарной функцией, а все числа преобразуются в процессе вызова переопределенного оператора к числам из алгебры *Ф* для поддержания замкнутости, потому в общем виде по свойствам сложной функции $D_x(u(x)^{v(x)}) = u(x)^{v(x)-1} \cdot (v(x) \cdot D_x u(x) + u(x)\ln u(x) \cdot D_x v(x))$. Очевидно, что такое количество операций значительно увеличивает время работы.

Рассмотрим меру роста времени вычисления значения производной функции (33) от n в точке $x = 5$.

$$\sum_{i=0}^{n}(x+1)^i \tag{33}$$

Запишем функцию и вычисление ее производной следующим образом.

Листинг 4: DerivativeOfSum.py

```
1           . . .
2           def S( i ) :
3           def f f (x) :
4           s = 0
5           for j in range( i ) :
6           s += (x+1)** i
7           return s
8           return f f
9
10      i = . .
11      %timeit d(S( i ) ) (5)
12      . . .
```

И проведем схожие с предыдущими измерения времени работы, результаты которых занесем в нижеприведенную таблицу.

Таблица 6: Результат вычисления производной, заданной конечной суммой. Время, μs.

n	Расчёт суммы	Конечные приращения	Симв. вычисления	Кольцо Φ
1	0.86	1.78	113.2	18.4
2	1.24	2.41	133.6	32.3
3	1.62	3	145.6	46.5
5	2.45	4.35	169.1	74.3
10	4.44	7.26	231	143
15	7.2	11.2	283	216
25	12.3	18	394	357
50	26	36.8	861	712
100	59.7	80.2	1512	1450
$R2$	0.9973	0.9948	0.996	0.9999

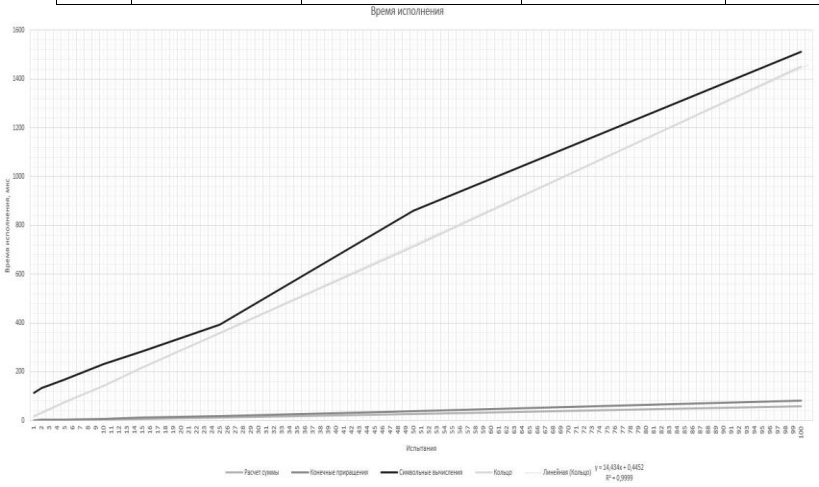


Рис. 8: Время вычисления значения производной функции, заданной суммой, в точке.

Из полученных данных видно, что при росте n, любой из методов растет линейно, так как и сам расчёт функции требует линейного времени. Однако, следует отметить, что вычисление производной с помощью кольца наиболее устойчиво, коэффициент детерминации $R^2 = 1 - \frac{\sigma_n^2}{\sigma_t^2}$ ближе к 1, чем остальные.

5.2 Тест 2. Оптимизация функции

Поставим задачу поиска оптимума функции *f*(*x*,*y*), являющегося вершиной эллиптического параболоида. *f*(*x*,*y*) = $10x^2 + y^2$

Для начала найдем ответ аналитически:

$$f(x,y) = 10x^2 + y^2$$
$$\overrightarrow{n_0} = (x_0, y_0) = argmin_{n \in \mathbb{R}^2} f(*\overrightarrow{n}) = f(x_0, y_0) -$$
$$\frac{\partial f}{\partial x} = 20x$$
$$\frac{\partial f}{\partial y} = 2y$$
$$\nabla f = (20x, 2y)$$
$$\begin{cases} 20x_0 = 0 \\ 2y_0 = 0 \end{cases} \Rightarrow \begin{cases} x_0 = 0 \\ y_0 = 0 \end{cases}$$

?

$$A = \frac{\partial^2 f}{\partial x^2} \mid_{\overrightarrow{n_0}} = 20$$

$$B = \frac{\partial^2 f}{\partial y^2} \mid_{\overrightarrow{n_0}} = 2$$

$$C = \frac{\partial^2 f}{\partial x \partial y} \mid_{\overrightarrow{n_0}} = 0$$

$$AC - B = 20 \cdot 2 - 0 = 40 > 0$$

$$A = 20 > 0$$

$\Rightarrow \overrightarrow{n_0} = (0,0)$ - минимум функции *f*(*x*,*y*)

Таким образом, правильным ответом будет пара (0,0).

Теперь получим ответ численным методом. Запишем эту функцию и найдем ее оптимум методом градиентного спуска:

```
1  def f (x , y) : return 10*x**2+y**2
2  opt = gradient_descent ( f , start =(10 ,10 ,) , domain=((-100, 100) ,) , eps=1e-6)
3  print (opt )
```

Ответом программы является пара (5.922274725121076e-14 -1.1252020499405204e-10) довольно близкая по значению к правильному, а ответ получен всего за 10 итераций [Приложения: Спуск 7] в среднем за 0.9 ms на тестируемой машине с характеристиками из Приложения.

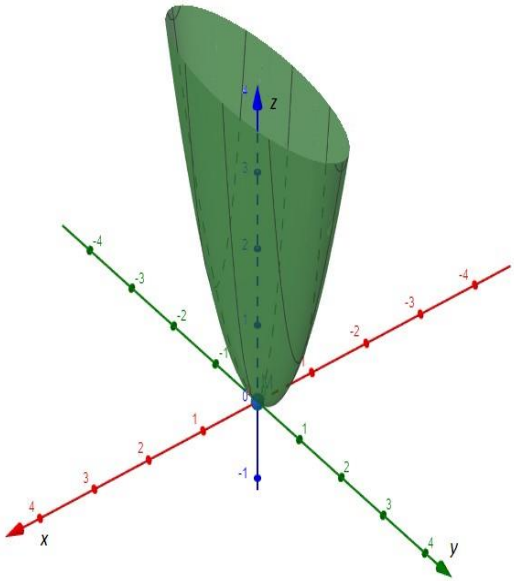


Рис. 9: $f(x,y) = 10x^2 + y^2$

Рассмотрим другой пример. Функция Розенброка - невыпуклая функция двух переменных, используемая для оценки оптимизационных алгоритмов (34).

$$R(x,y) = (1 - x)^2 + 100(y - x^2)^2 \qquad (34)$$

Пара (1,1) является минимумом функции, $R(1,1) = 0$. Градиентный спуск с дифференцированием, реализованным с помощью метода конечных приращений, даже с $\varepsilon = 10^{-12}$ не смог найти и за 50000 итераций оптимума данной функции. Наш градиентный спуск нашел за 503 итерации (7.5 ms) довольно близкую к правильному ответу пару - (0.9999925731400436, 0.9999853369329239)

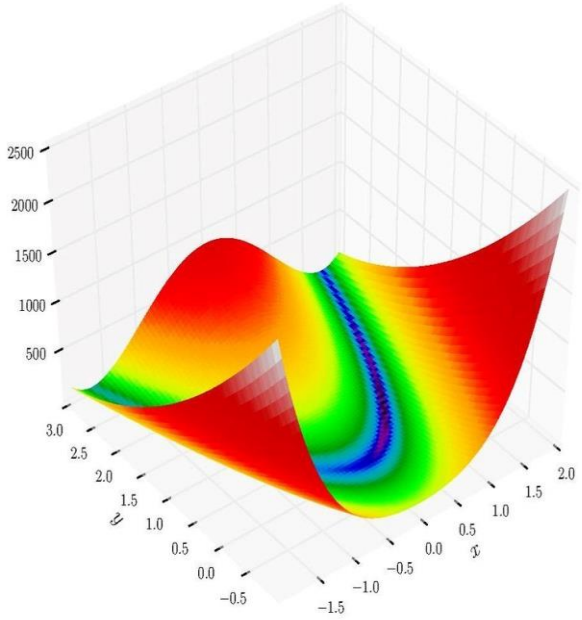


Рис. 10: $f(x,y) = (1 - x)^2 + 100(y - x^2)^2$

6 Решение оптимизационных задач

Рассмотрим решение оптимизационных задач.

Задача 1. Найти угол, при котором средняя скорость полёта за первую секунду тела максимальна, считая, что сила сопротивления воздуху прямо пропорциональна модулю скорости и противоположна по направлению, при запуске с горизонта с начальной скоростью v_0 = 1м/с, m = 1 кг.

Решение. Составим, для начала, математическую модель. Тело летит в близи земли, считаем, что на него действует ускорение свободного падения $\rightarrow -g$. Запишем II

закон Ньютона для него:

$$\vec{m}\vec{a} = m\vec{D}_t\vec{v} = m\vec{g} + \vec{F}_c$$

Пусть $\vec{-F} = -k\vec{v}$. Введём ДСКП так, что прямая $y = 0$ совпадает с горизонтом. Спроецируем уравнение на оси:

$$\begin{cases} D_t v_x = -\frac{k}{m} v_x \\ D_t v_y = -g - \frac{k}{m} v_y \end{cases}$$

Заменим $\frac{k}{m}$ на ω_0 Данные дифференциальные уравнения имеют следующие решения: (v_x $x - \omega 0 t$

(t) = v

$$v_y(t) = v_0 y e^{-\omega_0 t} - \omega_0 g(1 - e^{-\omega_0 t})$$

г
д
е
$v_{0x} = v_0 \cos(\alpha)$,
$v_{0y} = v_0 \sin(\alpha)$.
П
р
о
и
н

т
е
г
р
и
р
у
е
м
о
б
а
у
р
а
в
н
е
н
и
я:

$$\begin{cases} x(t) = \int_0^\tau v_x(t)dt = v_0\cos(\alpha)\frac{m}{k}\left(1 - e^{-\frac{k}{m}t}\right) - gt \\ y(t) = \int_0^\tau v_y(t)dt = \frac{m}{k}\left(-gt + (v_0\sin(\alpha) + \frac{mg}{k})(1 - e^{-\frac{k}{m}t})\right) \end{cases}$$

Средняя скорость рассчитывается по формуле $\overline{\frac{S}{t}}$, где S - путь, пройденный телом за время t. $S_x = x(\tau) - x(0), S_y = y(\tau) - y(0)$, тогда $S = \sqrt{S_x^2 + S_y^2}$. Учитывая, что $x(0) = y(0) = 0$,

$$v_{\text{ср}} = \frac{\sqrt{\left(v_0 \cos(\alpha)\frac{m}{k}\left(1 - e^{-\frac{k}{m}t}\right) - gt\right)^2 + \left(\frac{m}{k}\left(-gt + (v_0 \sin(\alpha) + \frac{mg}{k})(1 - e^{-\frac{k}{m}t})\right)\right)^2}}{\tau}$$

Запишем это как функцию для ЯП Python с учётом того, что τ = 1с. и найдём ее оптимум градиентным спуском. Так как ранее мы находили минимумы «спускаясь», а теперь задача состоит в поиске максимума, домножим функцию на -1. Зададим g = 10, константы пропорциональности k положим 1. Область поиска корня domain = (0, pi/2).

```
1  def v(a) :
2  return −sqrt((−g∗1 + m/k∗v0∗cos (a)∗(1−exp(−k/m∗1) ) )∗∗2 +
       (−g∗1 + (v0∗sin (a) + m∗g/k) ∗(1 − exp(−k/m∗1) ) ) ∗∗2)/1
```

Программа возвращает ответ: 1.5707963267948974, что, практически, совпадает с рациональным приближением $\frac{\pi}{2}$

Ответ: $\alpha \approx 1.5707963267948974$

Задача 2. На какой высоте h над центром круглого стола надо подвесить лампочку, чтобы освещенность края была наибольшей?[8]. Считать радиус стола R = 3м, силу света лампочки I = 100кд.

Решение. Построим к задаче чертеж (Рис.11) Пусть расстояние до края - L, тогда освещенность края находится по формуле (35).

$$E = \frac{I}{L^2}\cos\alpha \qquad (35)$$

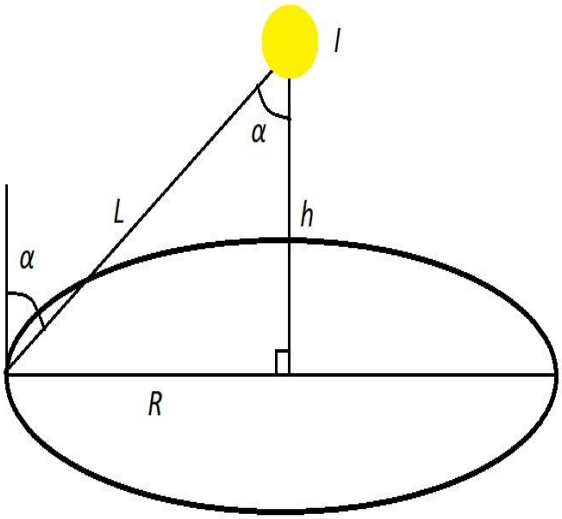


Рис. 11: Чертеж к Задаче 2

Заметим из прямоугольного треугольника, что $\cos\alpha = \frac{h}{L}$, а $L = \sqrt{h^2 + R^2}$. Теперь функцию освещенности возможно задать через высоту:

$$E(h) = \frac{I}{h^2 + R^2} \cdot \frac{h}{\sqrt{h^2 + R^2}}$$

$$E(h) = \frac{Ih}{(h^2 + R^2)^{\frac{3}{2}}}$$

R^+

Необходимо найти такие h_0,E_0, что $h_0 : E(h_0)\ 7{-\!\!\!\rightarrow}\ min = E_0$–? Запишем функцию $E(h)$ в ЯП Python:

```
1          I = 100
```

```
2        R = 3
3        def E(h) : return I*h/((h**2 + R**2) **1.5)
```

И оптимизируем с помощью градиентного спуска:

```
1        h_0 = gradient_descent (E, start =(0 ,0 ,) , domain=((0,
         10000) ,) , eps=1e−6)
2        E_0 = E(h_0)
3        print(h_0, E_0)
```

Результат выполнения программы: 2.121320343559642 4.276668660663894.

Ответ: наибольшая освещенность края в 4.276668660663894 люкса достигается высоте лампочки над центром стола в 2.121320343559642 метров.

7 Перспективы

В дальнейшем планируется расширить алгебраическое кольцо до колеса, и вычислять с его помощью смешанные производные второго порядка, что необходимо для составления матрицы Гессе для метода оптимизации BFGS, а также реализо вать сам метод BFGS.

Области применения

Результат работы может быть применен в сферах деятельности, где необходимо точно и быстро рассчитывать производные и вычислять оптимальные значения некоторых функций. Например, для автономного управления на предприятиях при расчете оптимальных изменений системы в реальном времени, принимая за аргументы эвристических функций показания датчиков данной системы. Также возможно внедрение в сервисы онлайн-дистрибуции, такие как интернет-магазины и кассы для динамического ценообразования при учете многих статистических и реальных параметров.

8 Заключение

Рассмотренные в данной работе модификации, а именно, параллелизм вычислений, введение алгебраического кольца, создание таблицы правил, действительно усовершенствовали как методы дифференцирования, так и алгоритмы поиска оптимумов заданных функций по точности относительно классических численных методов и по скорости при сравнении с системами символьных вычислений. Данные улучшения могут быть внедрены и для практического применения.

Список литературы

[1] Фихтенгольц Г.М. Курс дифференциального и интегрального исчисления. том 1. — М.: Изд. ФИЗМАТЛИТ., 2001. 616 с.

[2] Сканави М.И. Элементарная математика. — М.: Изд. Наука, 1967, 609 с.

[3] Просветов Г.И. Математический анализ. Задачи и решения. — М.: Изд. Альфа-пресс, 2018, 271 с.

[4] Поршнев С.В. Численные методы на базе MathCAD. — СПб.: Изд. БХВ-Птербург, 2005, 464 с.

[5] Яглом И.М. Комплексные числа — М.: Изд. ФИЗМАТЛИТ., 1963.

[6] Setzer, A. Wheels — 1997, 9 p.

[7] Новиков М.Ю. ИСПОЛЬЗОВАНИЕ СИМВОЛЬНЫХ ВЫЧИСЛЕНИЙ В ЗАДАЧАХ КОНЕЧНОМЕРНОЙ ОПТИМИЗАЦИИ // Инновации в науке: сб. ст. по матер. XLIII междунар. науч.-практ. конф. № 3(40). – Новосибирск: СибАК, 2015.

[8] Васильева В.А., Кудрина Т.Д., Методическое пособие по математике для поступающих в вузы. — М.: Изд. МАИ, 1992, 304 с.

[9] Иванова Т.В., Численные методы в оптике. — СПб: Университет ИТМО, 2017, 84 с.

9 Приложения

9.1 Спуск

Таблица 7: Последовательный спуск до оптимума функции

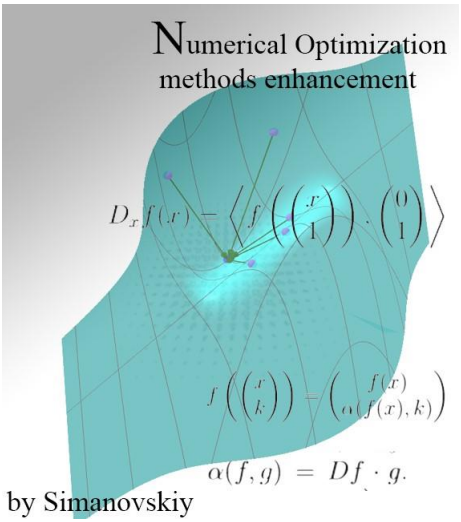
i	(*x*,*y*)
1	(10,10)
2	(−190.00001664153217,−10.000001664153219)
3	(0.004859581337740337,−8.999975920991998)
4	(−0.04372348522719183,−0.002360717875413698)
5	($1.118304285376015 \cdot 10^{-6}$,−0.002124640049925209)
6	$(-1.0061951036467986 \cdot 10^{-5}, -5.295966426081301 \cdot 10^{-7})$
7	($2.573519220807046 \cdot 10^{-10}$,$-4.7663562381166685 \cdot 10^{-7}$)
8	($-2.315492257717275 \cdot 10^{-9}$,$-1.2502280529072341 \cdot 10^{-10}$)
9	($5.922274725121076 \cdot 10^{-14}$,$-1.1252020499405204 \cdot 10^{-10}$)

9.2 Характеристики тестируемой машины

Hard	
CPU	Intel Core i5 8250U
RAM	4GB
Soft	

OS	OpenSuse 42.3
Enviroment	Python 3.7.2

9.3 Логотип работы



by Simanovskiy

Секция № 1. Космические аппараты и ракеты носители

Ракеты-носители сверхлегкого класса

Иванов Никита Павлович
г. Москва, ГБОУ «Школа №1580 при МГТУ им. Н.Э. Баумана», 11 класс
Научный руководитель: Леонов Виктор Витальевич, кандидат технических наук, доцент кафедры СМ-1 МГТУ им. Н.Э. Баумана

Содержание

Введение

Одноступенчатая одноразовая ракета-носитель

Одноступенчатая многоразовая ракета-носитель

Полутораступенчатая ракета-носитель

Заключение

Список литературы

Приложения

Введение

В XXI веке в космонавтике возникла тенденция к уменьшению размеров и массы космических аппаратов. Благодаря высокому уровню развития компьютерных технологий и микроминиатюризации стало возможным создание пико- и наноспутников, так называемых «Кубсатов» (англ. – CubeSat). Это аппараты, собираемые из одного и более типизированных блоков – юнитов – объемом 1 литр и массой

примерно 1 килограмм (рис.1) [1]. На данный момент «Кубсаты» выводятся на орбиты как попутный груз на средних и тяжелых носителях, то есть запускаются основной аппарат и несколько десятков «Кубсатов», причем орбиту обычно выбирает владелец основного аппарата. Желающий отправить «Кубсат» не может самостоятельно заказать запуск в удобные ему сроки, а вынужден подстраиваться под план запуска основной полезной нагрузки. Это, во-первых, невыгодно экономически, а во-вторых, бывает невозможно из-за особенностей задач аппарата. Таким образом, необходимо решить проблему оптимизации вывода пико- и наноспутников в космическое пространство.

В наши дни в некоторых странах, в частности в России, США, Англии, появились проекты ракет-носителей сверхлегкого класса, а также многоразовых носителей, которые с бóльшей выгодой можно использовать для вывода легких грузов в космическое пространство.

Семейство ракет сверхлегкого класса «Таймыр»

«Таймыр» – семейство одноразовых трехступенчатых ракет-носителей (РН) (масса полезной нагрузки (ПН) ~ 12-180 кг), которые различаются количеством стартовых ускорителей и мощностью двигательных установок (рис.2) [2]. Разработка ведется российской компанией ООО «Лин Индастриал» с 2014 года, а с 2015 года в сотрудничестве с космическим кластером «Сколково». В 2015 году испытаны отдельные системы управления «Таймыра». Первый запуск

в космос запланирован на 2020 год. Стоимость запуска 1 кг ПН ориентировочно будет стоить 60 тыс. долларов.

Проект одноступенчатой ракеты Haas 2CA

«Haas 2CA» – одноступенчатая одноразовая РН сверхлегкого класса (масса ПН – 100кг) (рис.3) [3]. Разрабатывается американской компанией «ARCA Space Corporation» с 2013 года. Первый тестовый полет намечен на 2018 год. Стоимость запуска 1 кг ПН ~ 10 тыс. долларов. Особенностью является применение клиновоздушного ракетного двигателя (КВРД) (рис.4).

Проект многоразового космолета Skylon

Еще один вариант вывода грузов на орбиту, с помощью многоразовой системы, предложили в Англии. Проект беспилотного многоразового (до 200 запусков) космолета «Skylon» (рис.5) разрабатывается английской компанией «Reaction Engines Limited», начиная с 1992 года [4]. Предполагается оснастить космолет новейшими турборакетнопрямоточными двигателями (рис.6), которые работали бы и как обычные турбореактивные двигатели на малых высотах, и как ракетные двигатели на больших высотах. Космолет только за счет двух таких двигателей самостоятельно осуществляет весь полет, от руления по взлетно-посадочной полосе до вывода ПН (~12т на низкую экваториальную орбиту) в космос, а после сам производит посадку. Первый тест двигателя намечен на 2020 год.

В настоящее время все большее значение приобретает экономическая сторона вопроса запуска «Кубсатов». Важно осуществить их вывод на орбиту в наиболее короткие сроки, причем по относительно низкой цене. Область сверхлегких ракет-носителей является пока что мало развитой. В связи с этим в данной работе будут рассмотрены следующие варианты ракет-носителей для вывода ПН массой 50 кг на низкую опорную орбиту высотой 200 км:

1) Одноступенчатая одноразовая ракета-носитель;

2) Одноступенчатая многоразовая ракета-носитель;

3) Полутораступенчатая ракета (на трехкомпонентном топливе).

Одноступенчатая одноразовая ракета-носитель

Уход от многоступенчатой схемы позволяет повысить надежность ракеты, так как отсутствуют многие не подлежащие предварительным испытаниям системы (например, элементы разделения ступеней в полете). Существенно упрощается предстартовая подготовка ракеты, в частности, процесс заправки топливом требует меньше времени и дополнительного оборудования. Отсутствие сбрасываемых во время полета ступеней позволяет не арендовать участки земли для их безопасного приземления, а, следовательно, экономит средства.

Рассчитаем необходимую стартовую массу топлива ракеты-носителя.

1) По формуле Циолковского потребная конечная скорость ракеты-носителя:

$\upsilon = \omega_{эф}\, ln(1 + \frac{М_т}{М_к})$, (1)

где $\omega_{эф}$ – эффективная скорость истечения газов, м/с; $М_т$ – стартовая масса топлива, кг; $М_к$ – конечная масса ракеты-носителя (без топлива), кг.

Из закона всемирного тяготения необходимая скорость на орбите высотой H=200 км: $\upsilon_к = \sqrt{G\frac{M_з}{R_з+H}} = 7784$ м/с, где $М_з = \frac{R_з^2 g_0}{G} = 5{,}97 * 10^{24}$ кг – масса Земли, $R_з = 6371$ км – радиус Земли, $g_0 = 9{,}81$ м/с^2 – ускорение свободного падения на Земле.

Благодаря обтекаемой форме носовой части ракеты потери скорости составят ~10%. Тогда потребная скорость РН будет $\upsilon = 8650$ м/с.

2) Преобразуем формулу (1) и обозначим $D = \frac{М_т}{М_к} = e^{\frac{\upsilon_к}{\omega_{эф}}} - 1$ (2).

Конечную массу ракеты-носителя запишем в виде: $М_к = М_{пг} + М_{то} + М_{су} + М_{ду}$ (3), где $М_{пг}$, $М_{то}$, $М_{су}$, $М_{ду}$ – соответственно массы полезной нагрузки, топливного отсека, систем управления, двигательной установки. Запишем уравнение (3) через массовые коэффициенты: $М_к = М_{пг} + М_т k_{то} + М_0 k_{су} + F k_{ду}$, (4) где: $k_{то}$, $k_{су}$, $k_{ду}, \frac{кг}{Н}$ – коэффициенты, соответствующие массам $М_{то}$, $М_{су}$, $М_{ду}$.

Учитывая, что $\mathrm{M}_0 = \mathrm{M}_\text{к} + \mathrm{M}_\text{т}$, формулу (2), а также представив тягу у поверхности Земли как $F = P_0 = n_0 M_0 g_0$ (n_0 – тяговооруженность РН), из выражения (4) получаем формулу для начальной массы топлива $\mathrm{M}_\text{т}$ через известные величины и массовые коэффициенты:

$$\mathrm{M}_\text{т} = \frac{D\mathrm{M}_\text{пг}}{1-k_\text{су}-n_0 g_0 k_\text{ду}-D(k_\text{то}+k_\text{су}+n_0 g_0 k_\text{ду})} \quad (5).$$

3) Рассчитаем по формуле (5) потребную массу топлива для одноступенчатой ракеты на топливной паре ‘керосин + жидкий кислород’. Для нее эффективная скорость истечения газов равна [5]: $\omega_\text{эф} = 3475$ м/с. Тогда $D = e^{\frac{8650}{3475}} - 1 = 11{,}05$. Зададим значения нужных коэффициентов: $n_0 = 1{,}1$; $g_0 = 9{,}81 \frac{\text{м}}{\text{с}^2}$; $k_\text{су} = 0{,}01$; $k_\text{ду} = 0{,}002 \frac{\text{кг}}{\text{Н}}$; $k_\text{то} = 0{,}05$. **Потребная стартовая масса топлива для нашей ракеты-носителя:** $\mathrm{M}_\text{т} = \mathbf{8254}$ кг.

4) Выбор именно таких значений коэффициентов был сделан на основе сравнения результатов формулы (5) для различных существующих ракет-носителей [6,7,8]. Для этого были выполнены расчеты с различными комбинациями значений коэффициентов и посчитаны относительные погрешности: $\Delta = \frac{|\mathrm{M}_\text{т посчитанная} - \mathrm{M}_\text{т истинная}|}{\mathrm{M}_\text{т истинная}}$ (6).

Для двухступенчатой РН «Зенит-2М» обсчитаем только вторую ступень. Учитываем, что 2я ступень запускается на некоторой высоте над поверхностью Земли, поэтому примем ускорение свободного

падения $g_0 \approx 9{,}6\ \text{м/с}^2$. По формуле (5) получаем значение $M_{\text{т посчитанная}} = 74230$ кг, $\Delta = 10\%$. Аналогично, для двухступенчатой РН «Электрон» $\Delta = 16\%$. Для одноступенчатой РН «Корона» $\Delta = 1\%$. Все полученные и используемые известные значения сведены в общую таблицу П.1.

5) Зная массу топлива $\text{М}_\text{т} = 8254$ кг, из формул (2), (3) и (4) получаем значения масс составляющих ракеты. Составим обобщающую таблицу 1:

Таблица 1. Массовые характеристики одноступенчатой одноразовой РН

$\text{М}_\text{пг}$, кг	$\text{М}_\text{т}$, кг	$\text{М}_\text{к}$, кг	$\text{М}_\text{ду}$, кг	$\text{М}_\text{су}$, кг	$\text{М}_\text{то}$, кг	М_0, кг
50	8254	747	194	90	413	9000

6) Найдем объемы окислителя ($V_{\text{ок}}$) и горючего ($V_{\text{гор}}$):

Рассмотрим РН «Зенит-2М» (двухступенчатая), для которой используют аналогичной нашей РН топливо (керосин + жидкий кислород) [9]. Обозначим $\alpha = \frac{\text{М}_\text{гор}}{\text{М}_\text{ок}}$. Для первой ступени РН «Зенит-2М» имеем $\alpha_1 = \frac{90219\ \text{кг}}{236567\ \text{кг}} = 0{,}381$; для второй – $\alpha_2 = \frac{23056\ \text{кг}}{59431\ \text{кг}} = 0{,}388$. Для нашей ракеты примем усредненное значение $\alpha = \frac{\alpha_1 + \alpha_2}{2} = 0{,}385$. Составим систему уравнений:

$$\begin{cases} \mathrm{M_{гор}} = \mathrm{M_т} - \mathrm{M_{ок}}, \\ \mathrm{M_{ок}} + 0{,}385 * \mathrm{M_{ок}} = \mathrm{M_т} \end{cases} \Leftrightarrow \begin{cases} \mathrm{M_{гор}} = 2294 \text{ кг}, \\ \mathrm{M_{ок}} = 5960 \text{ кг} \end{cases}$$

Из [5] берем значения плотностей: $\rho_{\text{ок}} = 1{,}141 \frac{\text{г}}{\text{см}^3}$, $\rho_{\text{гор}} = 0{,}835 \frac{\text{г}}{\text{см}^3}$. Запишем объемы: $V_{\text{ок}} = \frac{\mathrm{M_{ок}}}{\rho_{\text{ок}}} = \frac{5960}{1141} = 5{,}224 \text{ м}^3$, $V_{\text{гор}} = \frac{\mathrm{M_{гор}}}{\rho_{\text{гор}}} = \frac{2294}{835} = 2{,}747 \text{ м}^3$, общий объем баков $V_{\Sigma} = 7{,}97 \text{ м}^3$.

7) Для наддува баков используем гелий, хранящийся в шар-баллоне в баке с окислителем при температуре жидкого кислорода $T_0 = 90$ К под давлением $p_0 = 50$ атм. Пройдя через теплообменный аппарат, гелий при температуре $T = 600$ К и давлении $p = 1{,}5$ атм. поступает в баки РН по трубам.

Запишем уравнения Клапейрона-Менделеева для количества вещества гелия в топливных баках ракеты: $\nu = \frac{pV_{\Sigma}}{RT} = \frac{1{,}5*10^5*7{,}97}{8{,}31*600} = 239{,}77$ моль. Для гелия в шар-баллоне: $V_{He} = \frac{\nu R T_0}{p_0} = \frac{pV_{\Sigma}T_0}{Tp_0} = \frac{1{,}5*10^5*7{,}97*90}{600*50*10^5} = 0{,}0359 \text{ м}^3$. Радиус шар-баллона: $R = \sqrt[3]{\frac{3V_{He}}{4\pi}} = 0{,}205$ м.

Толщину стенки h баллона найдем по формуле [10]: $h = \frac{p_0 f R}{2\sigma c}$, где:

- f – коэффициент безопасности конструкции (давление высокое, примем $f = 3$);
- σ – предел прочности материала (для титанового сплава ВТ6С $\sigma \approx 1400$ Мпа [11]);

- с – коэффициент прочности сварного шва, примем с = 0,85.

Тогда $h = 1{,}3$ мм. $h \ll R$, поэтому примем объем шар-баллона равным $V_{He} = 0{,}0359$ м^3.

8) В баке с окислителем, помимо баллона с гелием, располагаются дополнительные элементы, такие как трубопровод и крепления баллона, датчик уровня жидкости, гасители колебаний. С учетом занимаемого ими объема, а также дополнительного окислителя (резерв), примем **значение объема полости бака окислителя** $V_{\text{ок}} = 5{,}4$ м^3. Аналогично, можно принять **значение объема полости бака с керосином** $V_{\text{гор}} = 2{,}85$ м^3.

9) Чтобы определить оптимальный диаметр РН, посчитаем $\beta = {}^{d}/_{L}$ – отношение диаметра к длине РН для различных существующих проектов ракет-носителей [3,6,7,12]. Объединим результаты в таблицу П.3. Усредним данные значения и для нашей ракеты получим $\beta = 0{,}078$. Суммарный объем баков: $V_0 = V_{\text{гор}} + V_{\text{ок}} = 2{,}85 + 5{,}4 = 8{,}25$ м^3.

10) Баки горючего (нижний) и окислителя (верхний) несущие, имеют совмещенные днища. Для упрощения дальнейших расчетов рассматриваем баки как цилиндры.

$V_0 = Sl_0 = \frac{\pi d^2 l_0}{4}$, где d – диаметр ракеты-носителя, l_0 – общая длина двух баков. Общая длина РН: $L = l_0 + l_1$, где l_1 – суммарная

длина отсека полезной нагрузки и хвостового отсека. $l_1 = \frac{d}{\beta} - \frac{4\,V_0}{\pi d^2}$ (7). Используя эту формулу, составим таблицу 3:

Таблица 3. Зависимость l_1 от диаметра d

d, м	0,95	0,98	1	1,03	1,05	1,07	1,1	1,13	1,15	1,2
l_1, м	0,53	1,6	2,31	3,29	3,92	4,53	5,42	6,26	6,8	8,086

Оптимальное значение $l_1 = 3{,}92$ м из таблицы, что соответствует диаметру $d = 1{,}05$ м. Длина РН: $L = {d}/{\beta} = {1{,}05}/{0{,}078} = 13{,}5$ м. Общая длина двух баков: $l_0 = L - l_1 = 9{,}5$ м.

Объемы баков ракеты относятся как их длины, поэтому получаем длины соответственно баков окислителя и горючего: $l_{\text{ок}} = \frac{5{,}4}{8{,}25} * 9{,}5 \approx 6{,}2$ м, $l_{\text{гор}} = \frac{2{,}85}{8{,}25} * 9{,}5 \approx 3{,}3$ м.

Параметры данной ракеты-носителя занесены в общую таблицу П.2.

11) Для снижения массы конструкции используем вафельное выполнение стенок баков. Это позволяет избежать необходимости усиления обечаек баков промежуточными шпангоутами. Шпангоуты ставятся только в местах крепления днищ. Управление вектором скорости ракеты осуществляется путем качания сопла двигателя.

Одноступенчатая многоразовая ракета-носитель

Данный вариант ракеты-носителя может использоваться многократно для запуска грузов в космос (ориентировочно около 100 пусков) и позволяет существенно экономить время и средства на строительстве носителей. Выполним в первом приближении расчет параметров многоразовой ракеты-носителя.

1) Воспользуемся расчетом одноразовой ракеты и всеми полученными значениями коэффициентов, обозначим M_1 – масса необходимого для вывода ракеты на орбиту топлива, $M_{\text{констр}}$ – масса конструкции РН, $M_{\text{к1}}$ – конечная масса РН (перед началом возврата на Землю). В качестве полезной нагрузки выступают сам груз, посадочное оборудование (массой $M_{\text{по}}$) и необходимое для посадки дополнительное топливо (массой $M_{\text{тд}}$):

$$\text{M}_1 = \frac{D(\text{M}_{\text{пг}}+M_{\text{по}}+M_{\text{тд}})}{1-k_{\text{су}}-n_0 g_0 k_{\text{ду}}-D(k_{\text{то}}+k_{\text{су}}+n_0 g_0 k_{\text{ду}})} \quad (8); \quad M_{\text{к1}} = M_{\text{констр}} + M_{\text{пг}} + M_{\text{тд}} + M_{\text{по}} \ (9).$$

2) Рассмотрим тяжелую ракету-носитель компании «SpaceX» Falcon 9 v1.1(R) [13]. У нее возвращаемой является первая ступень, при посадке выдвигаются 4 посадочные опоры (рис.7). Применим эту систему для нашей ракеты, масса опор – 70 кг. Еще 30 килограмм будут весить остальные посадочные системы: решетчатые рули (выпускаемых при посадке), газодинамическая система

стабилизации и навигационная система (для определения точного курса к посадочной площадке). Таким образом, $M_{\text{по}} = 100$ кг.

Массу необходимого для посадки топлива примем $M_{\text{тд}} = 130$ кг. Из формул (10) и (12) получаем соответственно $M_{\text{к1}} = 4200$ кг, $M_{01} = 50{,}6$ т, тогда $M_1 = M_{01} - M_{\text{к1}} = 46{,}4$ т.

Параметры данной ракеты-носителя занесены в общую таблицу П.2.

3) Оценим полученное значение стартовой массы ракеты-носителя в сравнении с подобными существующими проектами – сверхтяжелыми многоразовыми РН «Корона» [8] и «Россиянка» [12]. Обозначим $\delta = \frac{M_{\text{пг}}}{M_0}$; для «Короны» $\delta_1 = \frac{11000}{280000} \approx \frac{1}{25}$, для «Россиянки» $\delta_2 = \frac{21500}{750000} \approx \frac{1}{35}$. В нашем случае получаем $\delta_0 = \frac{50}{50600} \approx \frac{1}{1000}$. Коэффициенты δ_1 и δ_0 различаются в 40 раз, а δ_2 и δ_0 - примерно в 30 раз. При больших значениях массы полезного груза суммарная масса остальных посадочных систем и агрегатов и необходимого для посадки топлива составляет незначительную часть от всей массы, выводимой ракетой на орбиту. В нашем же случае $M_{\text{пг}} = 50$ кг, а масса «дополнительных элементов» $M_{\text{тд}} + M_{\text{по}} = 130 + 100 = 230$ кг, что почти в 5 раз больше массы полезной нагрузки. Поэтому данный вариант РН крайне неэффективен для вывода сверхлегких грузов на орбиту.

Полутораступенчатая ракета-носитель

Полутораступенчатая ракета представляет собой носитель, использующий трехкомпонентное топливо: два разных вида горючего сжигаются в одной и той же двигательной установке (с одним окислителем). Взлет происходит на более плотном горючем, например, на керосине. В менее плотных слоях атмосферы в камеру сгорания начинает поступать менее плотное горючее, но с бóльшим значением удельного импульса, например, жидкий водород. Чем бóльшую часть всего горючего составляет керосин, тем меньше масса конструкции, но тем больше масса топлива, и наоборот для водорода. Таким образом, можно подобрать рациональное соотношение масс двух видов горючего, чтобы максимально снизить стартовую массу ракеты. Выполним расчет основных технических характеристик ракеты-носителя на топливе метан – жидкий водород – кислород. Выбор метана вместо керосина обусловлен несколькими факторами:

- ✓ эффективная скорость истечения продуктов горения топлива метан – кислород выше, чем у пары керосин – кислород на 3,4% [14];
- ✓ метан – основной компонент природного газа, поэтому он дешевле керосина;
- ✓ наличие криогенных компонентов топлива позволяет без особых усложнений конструкции оснастить бак с жидким метаном криогенным оборудованием.

1) Проведем расчет полутораступенчатой РН аналогично расчету первого варианта сверхлегкой ракеты, используем те же обозначения и дополнительно введем: $\mathrm{M}_{\text{констр}}$ – масса конструкции ракеты (без полезной нагрузки), $k = \frac{\mathrm{M}_{\text{т}}}{\mathrm{M}_{\text{констр}}}$, тогда получим формулу для начальной массы топлива: $\mathrm{M}_{\text{т}} = \frac{k\mathrm{M}_{\text{пг}}(e^{v_{\text{хар}}/\omega_{\text{эф}}}-1)}{k+1-e^{v_{\text{хар}}/\omega_{\text{эф}}}}$ (17).

2) Рассмотрим разрабатываемый российской компанией ООО «Лин Индастриал» проект многоразовой ракеты-носителя «Зея» [15] на трехкомпонентном топливе (метан – жидкий водород – жидкий кислород). Воспользуемся известными параметрами ракеты $\mathrm{M}_{\text{пг(Зея)}} = 200$ кг, $\mathrm{M}_{0\text{(Зея)}} = 19{,}2$ т, $\mathrm{M}_{\text{констр(Зея)}} = 2{,}1$ т, высота орбиты – 200 км, $v_{\text{хар}} = 8650\ \frac{\text{м}}{\text{с}}$. Тогда $k = \frac{\mathrm{M}_{\text{т(Зея)}}}{\mathrm{M}_{\text{констр(Зея)}}} = \frac{\mathrm{M}_{0\text{(Зея)}}-\mathrm{M}_{\text{констр(Зея)}}}{\mathrm{M}_{\text{констр(Зея)}}} = 8{,}14$.

Из формулы (17) получим значение $\omega_{\text{эф}}$ для «Зеи» (будет являться приведенным значением $\omega_{\text{эф}}$ за весь полет): $\omega_{\text{эф}} = v_{\text{хар}} \Big/ \ln\left(1 + \frac{k\mathrm{M}_{\text{т}}}{\mathrm{M}_{\text{т}}+k\mathrm{M}_{\text{пг}}}\right) = 4057\ \frac{\text{м}}{\text{с}}$ (18).

3) По формуле (17), подставив известные значения, определяем массу топлива для нашей ракеты $\mathrm{M}_{\text{т}} = 4288$ кг, с учетом резерва округляем до $\mathrm{M}_{\text{т}} = 4500$ кг. Тогда масса конструкции $\mathrm{M}_{\text{констр}} = 560$ кг. $\mathrm{M}_0 = \mathrm{M}_{\text{констр}} + \mathrm{M}_{\text{т}} + \mathrm{M}_{\text{пг}} = 5110$ кг, $\mathrm{M}_{\text{су}} = \mathrm{M}_0\mathrm{k}_{\text{су}} = 50$ кг, $\mathrm{M}_{\text{ду}} = \mathrm{g}_0\mathrm{n}_0\mathrm{M}_0\mathrm{k}_{\text{ду}} = 110$ кг. Масса топливного отсека: $\mathrm{M}_{\text{то}} = \mathrm{M}_{\text{констр}} - \mathrm{M}_{\text{су}} - \mathrm{M}_{\text{ду}} = 400$ кг.

4) Для ракеты «Зея» известны значения коэффициентов топливных отсеков (отношение конечной массы топливного отсека к начальной массе топлива) отдельно для метана с необходимым количеством окислителя для его сгорания: $\mu_м = 0{,}05$, и для водорода: $\mu_в = 0{,}1$. $М_м$ и $М_в$ – соответственно массы метана и водорода со «своими» частями окислителя:

$$\begin{cases} М_м + М_в = М_т, \\ М_в\mu_в + М_м\mu_м = М_{то} \end{cases} \Leftrightarrow \begin{cases} М_в = \frac{М_{то} - М_т\mu_м}{\mu_в - \mu_м}, \\ М_м = М_т - М_в \end{cases} \Leftrightarrow \begin{cases} М_в = 3500 \text{ кг}, \\ М_м = 1000 \text{ кг} \end{cases}$$

(19)

Массы топливных отсеков метана и водорода: $М_{то\ м} = 50$ кг, $М_{то\ в} = 350$ кг.

5) Запишем стехиометрические уравнения реакций горения метана и водорода:

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O; \qquad 2H_2 + O_2 \rightarrow 2H_2O$$

Найдем отношения масс метана и водорода к массам жидкого кислорода, требуемым для полного сгорания каждого из компонентов горючих:

$$\frac{М_{CH_4}}{m1_{O_2}} = \frac{\nu_{CH_4}\mu_{CH_4}}{\nu_{O_2}\mu_{O_2}} = \frac{1*16}{2*32} = \frac{1}{4}; \qquad \frac{М_{H_2}}{m2_{O_2}} = \frac{\nu_{H_2}\mu_{H_2}}{\nu_{O_2}\mu_{O_2}} = \frac{2*2}{1*32} = \frac{1}{8}.$$

Используя значения из системы (19), получаем: $М_{CH_4} = 0{,}2М_м = 200$ кг, $m1_{O_2} = 800$ кг, $М_{H_2} = 390$ кг, $m2_{O_2} = 3110$ кг. Суммарная масса окислителя: $М_{O_2} = m1_{O_2} + m2_{O_2} = 3910$ кг.

6) Такие же массы были получены при расчете данной РН как двухступенчатой. Момент перехода на водород можно считать началом работы «второй» ступени, причем, в отличие от классической двухступенчатой РН, не происходит отделения ступени. Расчет выполнялся по формуле (5) с использованием значений коэффициентов из таблицы П.1 для каждой ступени отдельно, начиная со второй, полезной нагрузкой для которой являлись сам груз и топливный отсек «первой» ступени. В качестве ПН «первой» ступени рассматривалась «вторая» ступень целиком (топливный отсек, топливо и сам полезный груз). Чтобы убедиться в правильности выбранных значений коэффициентов была обсчитана двухступенчатая РН «Зенит-2М» [6]. Погрешность по формуле (6) составила 1% для первой ступени.

7) Найдем объемы, занимаемые каждым из компонентов: $V_{CH_4} = \frac{\mathrm{M}_{CH_4}}{\rho 1} = \frac{200}{415} = 0{,}5\ \text{м}^3$, $V_{H_2} = \frac{\mathrm{M}_{H_2}}{\rho 2} = \frac{390}{70} = 5{,}5\ \text{м}^3$, $V_{O_2} = \frac{\mathrm{M}_{O_2}}{\rho 3} = \frac{3910}{1141} = 3{,}5\ \text{м}^3$, суммарный объем $V_{\Sigma} = 9{,}5\ \text{м}^3$.

8) Воспользуемся формулами из расчета одноразовой одноступенчатой ракеты (пункт 7) и получим объем гелия в шар-баллоне: $V_{He} = 0{,}04275\ \text{м}^3$, а радиус шар-баллона: $R = 0{,}22$ м. Аналогично, толщина стенки шар-баллона: $h = 1{,}4$ мм. Примем с запасом следующие значения объемов полостей баков: $V_{\text{бака } H_2} = 5{,}7\ \text{м}^3$, $V_{\text{бака } O_2} = 3{,}7\ \text{м}^3$, $V_{\text{бака } CH_4} = 0{,}6\ \text{м}^3$. Суммарный объем: $V_{\Sigma\text{баков}} = 10\ \text{м}^3$.

7) Воспользуемся формулой (7) и составим таблицу 4:

Таблица 4. Зависимость l_1 от диаметра d полутораступенчатой РН

d, м	1	1,05	1,07	1,1	1,11	1,12	1,1
l_1, м	0,082	1,9	2,6	3,57	3,9	4,2	5,1

$l_1 = 3{,}9$ м – оптимальное значение суммарной длины хвостового отсека и отсека полезной нагрузки, тогда диаметр $d = 1{,}11$ м. Общая длина носителя $L = {}^{d}/_{\beta} = 14{,}2$ м. Длина всех топливных отсеков: $l_0 = L - l_1 = 10{,}3$ м.

Выберем для нашей ракеты компоновочную схему с несущими баками с совмещенными днищами для экономии массы конструкции. Нижний бак – бак с жидким кислородом, выше располагается бак с метаном, а самый верхний – бак с жидким водородом. Обозначим:

$$\xi_1 = \frac{l_{б\,CH_4}}{l_{б\,H_2}} = \frac{V_{б\,CH_4}}{V_{б\,H_2}} = \frac{0{,}6}{5{,}7} = 0{,}1, \quad \xi_2 = \frac{l_{б\,O_2}}{l_{б\,H_2}} = \frac{V_{б\,O_2}}{V_{б\,H_2}} = \frac{3{,}7}{5{,}7} = 0{,}65.$$

$l_{б\,H_2} + \xi_1 l_{б\,H_2} + \xi_2 l_{б\,H_2} = l_0$, откуда $l_{б\,H_2} = 5{,}9$ м, $l_{б\,CH_4} = 0{,}6$ м, $l_{б\,O_2} = 3{,}8$ м.

Параметры данной ракеты-носителя занесены в общую таблицу П.2.

Заключение

В работе были рассмотрены три варианта носителя сверхлегкого класса для выведения полезной нагрузки, например, «Кубсатов» суммарной массой 50 кг. Были посчитаны основные

параметры ракет, для наглядности представлены в общей таблице П.2. Также были созданы 3D-модели одноступенчатой одноразовой и полутораступенчатой ракет (рис.8, рис.9). Вариант многоразовой ракеты из-за слишком большой стартовой массы нецелесообразно использовать, даже несмотря на то, что не нужно строить постоянно новый носитель, а достаточно заново подготовить старый. Наиболее экономически выгодным вариантом среди оставшихся двух является полутораступенчатая ракета-носитель. Несмотря на сложности, связанные с переходом в полете с одного вида горючего на другой, этот носитель существенно выигрывает у одноступенчатой ракеты по стартовой массе и массе топлива (~ в 1,8 раза).

При сравнении сверхлегких носителей с тяжелыми, основными ракетами для запуска «Кубсатов» на данный момент, можно сделать следующие выводы:

- ✓ уровень надежности одноступенчатой (или полутораступенчатой) ракеты выше;
- ✓ одноступенчатая РН гораздо дешевле и проще как в изготовлении, так и в обслуживании;
- ✓ пуск «Кубсатов» на сверхлегкой ракете возможно осуществить в более короткий срок;
- ✓ учитывая предыдущие факторы, даже при равной стоимости запуска одного килограмма полезной нагрузки велика вероятность, что

заказчик выберет сверхлегкую РН за более короткие сроки запуска и более высокие гарантии безопасности пуска.

Таким образом, можно сделать вывод о целесообразности применения одноразовых одноступенчатых (или полутораступенчатых) носителей сверхлегкого класса для вывода легких грузов в космическое пространство. В эпоху нано-технологий создание такой ракеты является перспективным направлением развития космонавтики.

Список литературы

1. Спутник «Кубсат» // Свободная энциклопедия «Википедия» [Сайт]. Режим доступа: https://ru.wikipedia.org/wiki/Кубсат (дата обращения 26.01.2019).
2. Сверхлегкая ракета «Таймыр» [Электронный ресурс] // Официальный сайт ООО «Лин Индастриал», 2018. Режим доступа http://spacelin.ru/proekty/sverkhlegkaya-raketa-taymyr (дата обращения 14.11.2018).
3. From Earth to orbit using a single-stage rocket [Электронный ресурс] // Новостной портал New Technology & Science News, 2017. Режим доступа: https://newatlas.com/single-stage-rocket/48710 (Статья об одноступенчатой РН «Haas 2CA», дата обращения 14.11.2018).
4. Космолет «Skylon» // Свободная энциклопедия «Википедия» [Сайт]. Режим доступа: https://en.wikipedia.org/wiki/Skylon_(spacecraft) (дата обращения 15.11.2018).
5. Егорычев, В.С. Топлива химических ракетных двигателей: учебное пособие / В.С. Егорычев, В.С. Кондрусев.– Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2007.-72с.

6. Ракета "Зенит-2М": характеристики, интересные факты [Электронный ресурс] // Информационный портал «Syl.ru», 2018. Режим доступа: https://www.syl.ru/article/342859 (дата обращения 22.11.2018).
7. РН «Electron» [Электронный ресурс] // Журнал «Все о космосе», 2018. Режим доступа: https://aboutspacejornal.net/%D1%80%D0%BD-electron (дата обращения 20.11.2018).
8. Русский космос: проект «Корона» и другие разработки ГРЦ Макеева [электронный ресурс] // Популярная механика. Режим доступа: https://www.popmech.ru/technologies/363532 (дата обращения 14.11.2018).
9. РН «Зенит 2М» // Свободная энциклопедия «Википедия» [Сайт]. Режим доступа: https://en.wikipedia.org/wiki/Zenit-2M (дата обращения 22.11.2018).
10. Энциклопедия по машиностроению XXL [Сайт]. Режим доступа: http://mash-xxl.info/info/207328/ (Расчет шар-баллона с гелием, с. 361, дата обращения 10.02.2019).
11. Промышленные титановые сплавы [Сайт] // Промышленный магазин «Тиком-М». Режим доступа: http://ticom-m.ru/O-titane-i-titanovyh-splavah/Promyshlennye-titanovye-splavy (Информация о титановом сплаве ВТ6С, дата обращения 10.02.2019).
12. Свободная энциклопедия «Википедия» [Сайт]. Режим доступа:

- https://ru.wikipedia.org/wiki/Россиянка_(ракета-носитель) (РН «Россиянка», дата обращения 13.11.2018).
- https://en.wikipedia.org/wiki/Vector-R , https://en.wikipedia.org/wiki/Vector-H (Ракеты «Vector R» и «Vector H», дата обращения 15.11.2018).

13. РН «Falcon 9» // Свободная энциклопедия «Википедия» [Сайт]. Режим доступа: https://ru.wikipedia.org/wiki/Falcon_9 (дата обращения 22.11.2018).

14. Метан - последняя надежда? [Электронный ресурс] // Интернет-статья «РАКЕТЫ-НОСИТЕЛИ. РАКЕТНЫЕ ДВИГАТЕЛИ», И. Афанасьев. Режим доступа: http://epizodsspace.airbase.ru/bibl/nk/1998/17-18/17-18-1998-3.html (Информация о метане как ракетном топливе, дата обращения 20.11.2018).

15. Одноступенчатая многоразовая ракета «Зея» [Электронный ресурс] // Официальный сайт ООО «Лин Индастриал», 2019. Режим доступа: http://spacelin.ru/proekty/odnostupenchataya-mnogorazovaya-raketa-zeya/ (Дата обращения 07.02.2019).

Приложения

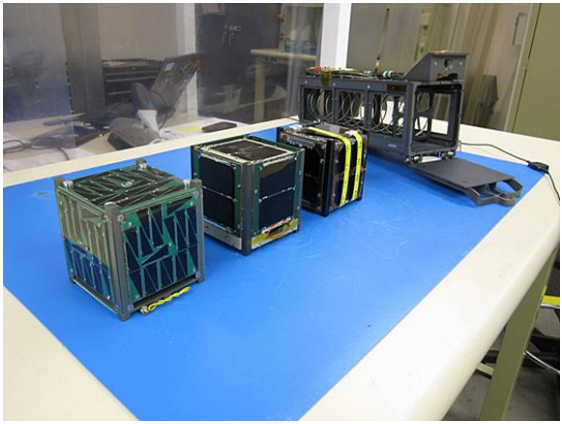


Рис.1. Три «Кубсата» размера 1U и пусковой контейнер

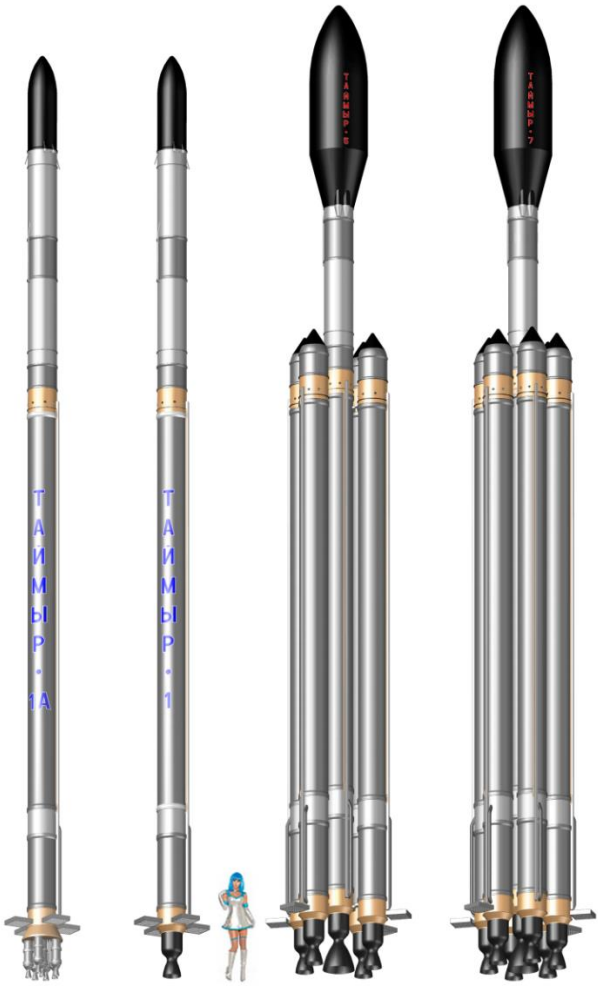


Рис.2. Сверхлегкие ракеты семейства «Таймыр»

Рис.3. Ракета-носитель «Haas 2CA» с КВРД

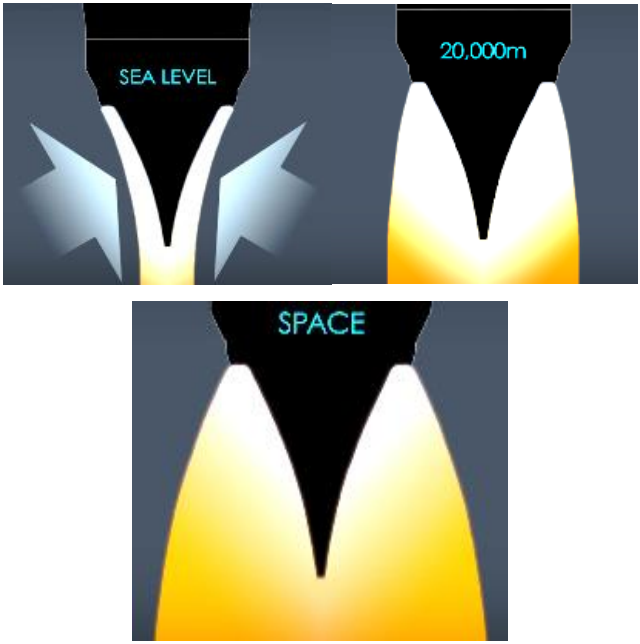


Рис.4. Принцип работы КВРД на разных высотах (слева направо по мере убывания внешнего давления)

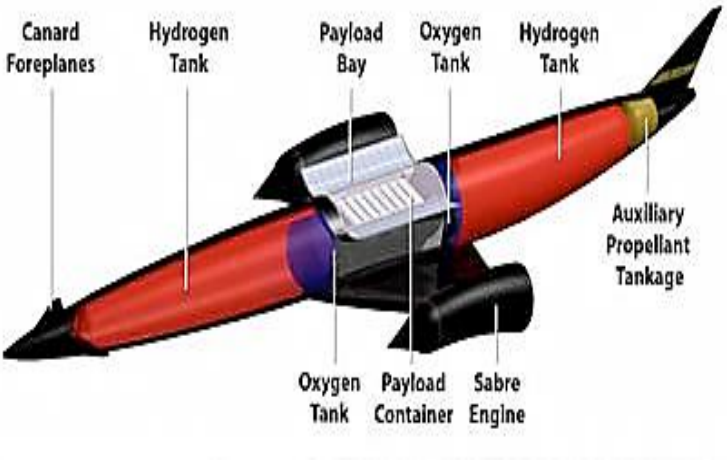


Рис.5. Космолет «Skylon»

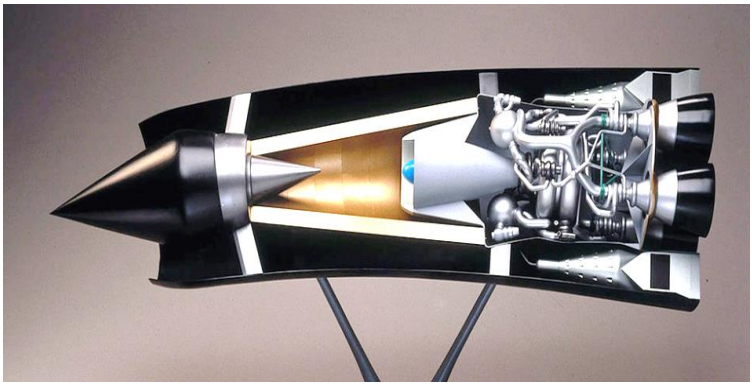


Рис.6. Макет турборакетнопрямоточного двигателя для космолета «Skylon»

Рис.7. Первая ступень РН «Falcon 9», посадка

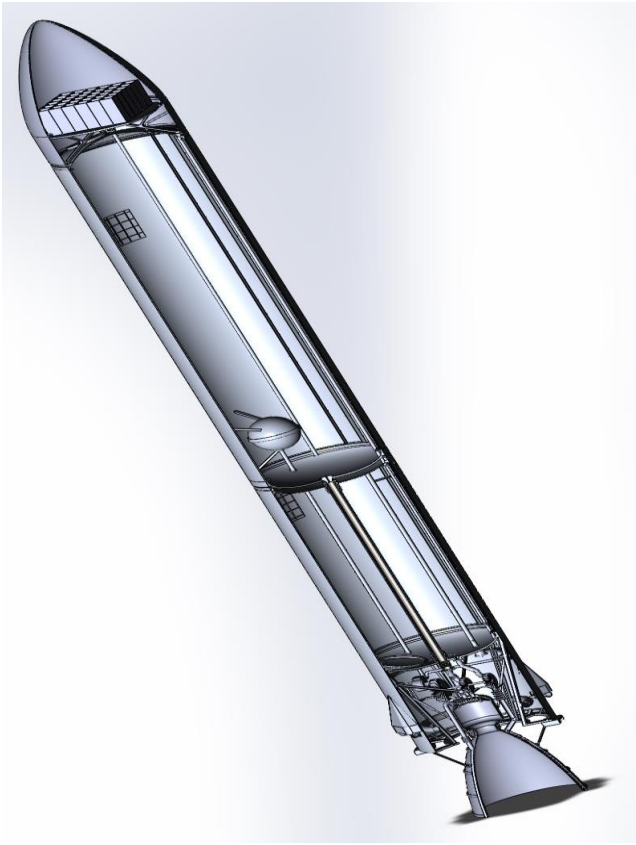


Рис.8. Модель одноступенчатой одноразовой ракеты-носителя

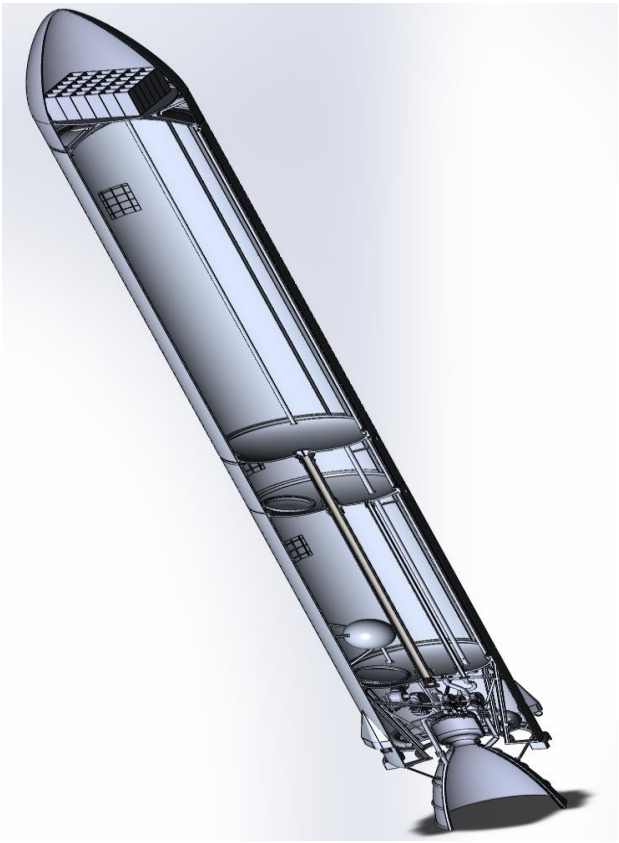


Рис.9. Модель полутораступенчатой одноразовой ракеты-носителя

№	$k_{су}$	$k_{ду}$, $кг/Н$	$k_{то}$	Название ракеты-носителя	$n_0 * g_0$, $м/с^2$	P_0, кН	$\omega_е$, $м/с$	$v_к$, $м/с$	D	$M_{пг}$, кг	M_0, кг	$M_к$, кг	Известное значение $M_т$, кг	$M_т$, кг	Δ
1	0,005	0,001	0,05	(проект)	1,3 * 9,81 = 12,753		3475	9160 (15% потерь)	12,957	50				**6207**	-
				Зенит-2М	$P_0/M_0 = 8,832$	912			4,056	12030	102824	20337	82487	67092	18,7%
				Электрон	$P_0/M_0 = 8,3$	22			4,5263	250	2650	500	2150	1616	25%
				Корона	$P_0/M_0 = 14,89$	4170			7	11500	280000	35000	245000	163991	33%
2	0,005	0,002	0,05	(проект)	1,1 * 9,81 = 10,791		3475	8845 (12% потерь)	11,75	50				**8000**	-
				Зенит-2М	8,832	912			4,056	12030	102824	20337	82487	71480	13,3%
				Электрон	8,3	22			4,5263	250	2650	500	2150	1729	19,6%
				Корона	14,89	4170			7	11500	280000	35000	245000	216540	11,6%
3	0,006	0,002	0,05	(проект)	1,1 * 9,81 = 10,791		3475	8845 (12% потерь)	11,75	50				**9658**	-
				Зенит-2М	8,832	912			4,056	12030	102824	20337	82487	72014	12,7%
				Электрон	8,3	22			4,5263	250	2650	500	2150	1744	19%
				Корона	14,89	4170			7	11500	280000	35000	245000	221300	9,7%
4	0,01	0,002	0,05	(проект)	1,1 * 9,81 = 10,791		3475	8650 (10% потерь)	11,05	50				**8254**	-
				Зенит-2М	8,832	912			4,056	12030	102824	20337	82487	74230	10%
				Электрон	8,3	22			4,5263	250	2650	500	2150	1805	16%
				Корона	14,89	4170			7	11500	280000	35000	245000	242654	1%

Таблица 1. Определение значений массовых коэффициентов

№	**Вариант РН**	$M_{пг}$, кг	M_0, кг	$M_т$, кг	$M_к$, кг	$M_{горючее}$, кг	$M_{горючее}$, кг	$M_{окислит}$ (O_2 ж.), кг
1	Одноступенчатая одноразовая	50	9000	8254	747	2294 (керосин)		5960
2	Одноступенчатая многоразовая	50	50600	46400	4200			
3	Полуторасту пенчатая	50	5110	4500	610	200 (CH_4 ж.)	390 (H_2 жидкий)	3910

№	**Вариант РН**	$M_{конст}$, кг	$M_{ду}$, кг	$M_{су}$, кг	$M_{то}$, кг	$V_{горючего}$, л	$V_{горючего}$, л	$V_{окислите}$, л
1	Одноступенчатая одноразовая	697	194	90	413	2850 (керосин)		5400
2	Одноступенчатая многоразовая	4020						
3	Полуторасту пенчатая	560	110	50	400	600 (CH_4 ж.)	5700 (H_2 жидкий)	3700

№	**Вариант РН**	V_Σ, л	d, м	L, м	l_0, м	$l_{горючего}$, м	$l_{горючего}$, м	$l_{окислите}$, м

1	Одноступенчатая одноразовая	8250	1,05	13,5	9,5	3,3 (керосин)		6,2
2	Одноступенчатая многоразовая							
3	Полуторасту пенчатая	10000	1,11	14,2	10,3	0,6 (CH_4 ж.)	5,9 (H_2 жидкий)	3,8

Таблица 2. Технические характеристики рассматриваемых вариантов РН

Название РН	**Значение β**	**Название РН**	**Значение β**
«Haas 2CA»	$\beta = 1{,}5/16 = 0{,}09375$;	«Vector-H»	$\beta = 1{,}2/19{,}5 = 0{,}0615$;
«Россиянка»	$\beta = 4{,}1/47{,}6 = 0{,}08613$;	«Зенит-2М»	$\beta = 3{,}9/57{,}35 = 0{,}068$;
«Vector-R»	$\beta = 1{,}2/12 = 0{,}1$;	«Электрон»	$\beta = 1{,}2/17 = 0{,}07$;

Таблица 3. Значения отношений диаметра РН к ее длине для различных существующих ракет-носителей

Секция № 2. Аэрокосмические системы

Транспортно-пусковое устройство для запуска малых космических спутников при помощи пушки Гаусса

Баринова Екатерина Александровна
г. Москва, ГБОУ «Школа 641 им. С. Есенина», 11 класс
Научный руководитель: Попов Александр Сергеевич, старший преподаватель кафедры СМ-2, МГТУ им. Н.Э. Баумана

Содержание

Введение
Описание модели
Принцип действия
Вычисление характеристик модели
Вычисление траектории и скорости полета
Сравнение с другими конструкциями
Заключение
Список литературы
Приложения

Введение

В настоящее время существует не мало способов выведения искусственных спутников Земли (ИСЗ) на орбиту, в частности и с международной космической станции (МКС). В данной работе рассматривается возможность использования Пушки Гаусса для запуска малых космических спутников на орбиту с МКС. Современные способы запуска спутников осуществляется при помощи таких концепций как SPL, ISIPOD, P-POD и другие. Однако, все они работают за счет пружины, с

помощью которой осуществляется вывод спутника на орбиту и не способны придавать разгоняемому объекту импульс. Одним из таких способов запуска может быть рельсотрон. Особенности рельсотрона заключаются в том, что требуется высокая начальная скорость снаряда, т.е. сначала нужно выстреливать из обычной пушки, чтобы разогнать снаряд. На ряду с собратьями изобретения Карла Гаусса пушка Гаусса имеет ряд преимуществ: высокая скорострельность и регулируемая скорость пули (в нашем случае спутника), бесшумность, повышенная износостойкость и возможность использования в открытом космосе - пушка Гаусса способна придавать запускаемому аппарату импульс, что в дальнейшем может быть полезно для построения траектории и для изменения параметров орбиты. В данной работе мы проведем исследование транспортно-пускового устройства для запуска малых космических аппаратов при помощи пушки Гаусса и выявим ее особенности и преимущества для запуска малых космических аппаратов.

Цель работы:

Разработать транспортно-пусковое устройство для запуска малых космических аппаратов при помощи пушки Гаусса и определение его параметров.

Задачи:

1. Разработать транспортно-пусковое устройство для запуска малых космических спутников с МКС

2. Определить параметры получившегося устройства
3. Выявить преимущество перед другими способами запуска

Описание модели

Модель состоит из (см. рис. 1):

1- Корпус;

2- Крышка;

3- Магнитный ускоритель (катушка);

4- Блок конденсаторов в изоляционном корпусе;

5- Блок питания (аккумулятор);

6- Крышка для быстрого доступа к блоку конденсаторов и аккумуляторов;

7- Направляющие;

8- Спутник.

Параметры Спутника:

Для запуска используем спутник CubeSat 1U размером [1]:

100х100х113мм; $m_{спутн}$ = 1000 г.

Параметры контейнера:

Размер: 126х126х437 мм ; n = 3 мм – ширина стенки контейнера.

Параметры конденсатора [2]:

C =10000 мкФ ,U =50 В; Линейные размеры (35х45) мм; $m_к =$ 56 г.

Параметры аккумулятора типа "крона" [3]:

Тип: 6f22. Напряжение: $U_{ak} =$ 9 В. Емкость: $E_{ак}$= 270 mAh

48,5х26,5х17,5 мм; $m_{ист}$ =15 г.

Для подтверждения актуальности использования выбранного аккумулятора для данной системы, сравним энергии батареи конденсаторов и энергию аккумулятора:

Энергия, запасаемая в аккумуляторе в соответствии с [4]:

(1) $W = E_{ak} \cdot 3600 \cdot U_{ak} = 8748$ Вт · с (Дж).

Энергия, запасаемая в конденсаторе в соответствии с [5]:

(2) $E = \frac{C \cdot U^2}{2}$, U-напряжение конденсатора; C-ёмкость конденсатора.

Вычислим максимальную энергию, запасаемую в конденсаторах, рассматриваемую в данной работе (энергия для системы с 42 конденсаторами):

(3) $E = \frac{C_{42} \cdot U^2}{2} = 525$ Дж, $C_{42} = C_1 + C_2 + \cdots + C_{42}$, так как конденсаторы соединены параллельно.

Из выше приведенных расчетов, делаем вывод, что выбранный аккумулятор удовлетворяет требованиям системы.

Параметры преобразователя [6]:

70х46х13 мм; $m_{преоб}$ = 65 г.

Принцип действия

Блок питания заряжает батареи конденсаторов. При их разрядке электрический ток поступает на электромагнитный ускоритель. При протекании этого тока в электромагнитном ускорителе возникает магнитное поле, которое разгоняет спутник. На концах спутника при этом образуются полюса, симметричные полюсам катушки, из-за чего после прохода центра катушки спутник притягивается в обратном направлении, то есть тормозится.

Как правило, для получения такого импульса используются электрические конденсаторы с высоким рабочим напряжением.

Параметры ускоряющих катушек, снаряда и конденсаторов должны быть согласованы таким образом, чтобы при выстреле к моменту подлета снаряда к соленоиду индукция магнитного поля в соленоиде была максимальна, но при дальнейшем приближении снаряда резко падала. Стоит заметить, что возможны разные алгоритмы работы ускоряющих катушек. Для наибольшего эффекта импульс тока в соленоиде должен быть кратковременным и мощным.

Кинетическая энергия спутника в соответствии с [5]:

(4) $E = \frac{m \cdot v_{спутн}^2}{2}$, m - Масса спутника; $v_{спутн}$ - его скорость.

Вычисление характеристик модели

Одним из основных элементов пушки Гаусса является электрический конденсатор. Большинство конденсаторов большой емкости, используемых в магнитных ускорителях, электролитические и полярные. Для таких конденсаторов очень важно правильное подключение – положительный заряд подаем к выводу “+”, а отрицательный к “-”. Зная емкость конденсатора и его максимальное напряжение можно найти энергию, которую может накапливать этот конденсатор. В нашем устройстве будет 20 конденсаторов, номинальной емкостью 10 000 мкФ и напряжением равным 50 В

Зная энергию, можно найти ориентировочную кинетическую энергию снаряда – или попросту мощность будущего магнитного ускорителя. Как правило, КПД η Электромагнитного ускорителя масс примерно равен 1-7%. Возьмем среднее значение КПД оптимизированного магнитного ускорителя масс – 5.

(5) $\eta = \frac{E_k}{E_c} \cdot 100\%$.

Подставляя в формулу (5) формулы (2) и (4) получаем, что формула скорости вылета этого спутника:

(6) $v_{спутн} = \sqrt{\frac{C_{общ} \cdot U^2}{100 m_{спутн}}} = 5$ м/с, где $C_{общ} = C_1 + C_2 + C_3 + \cdots + C_{20}$, так как конденсаторы соединены параллельно.

Масса всей конструкции (M) будет складываться из масс конденсаторов($m_{конденсаторов}$), преобразователя($m_{преоб}$), источника тока ($m_{ист}$),самого контейнера($m_{конт}$) и массы спутника, помещенного в контейнер ($m_{спутн}$) .

(7) $M = m_{конденсаторов} + m_{конт} + m_{спутн} + m_{преоб} + m_{ист}$.

Учитывая, что конденсаторы подключены параллельно, то масса всех конденсаторов:

(8) $m_{конденсаторов} = m_{к1} + m_{к2} + m_{к3} + \cdots + m_{к20}$.

Для того, чтобы узнать массу контейнера следует провести расчет объема всего контейнера и вычесть из него объем его полой части:

(9) $m_{конт} = (m_{тела}) - (m_{полой\ части})$, где

(10) $m_{полой\ части} = V' \cdot \rho$,

(11) $m_{тела} = V \cdot \rho$, где V – объем всего тела; V' - объем полой части, а ρ – его плотность (согласно [7] $\rho = 2{,}7$ г/см^3).

Объем контейнера:

(12) $V = x \cdot y \cdot z$, где x, y, z – длина, ширина и высота контейнера.

Объем полой части контейнера:

(13) $V' = (x - n) \cdot (y - n) \cdot (z - n)$, Где n – ширина стенок контейнера.

Подставляя в (9) формулы (10), (11), (12), (13) следует, что

(14) $(m_{конт} = (x \cdot y \cdot z \cdot \rho) - \big((x - n)(y - n)(z - n)\rho\big) =$ 3204 г.

Подставив (14) и (8) в формулу (7), мы можем найти массу всей конструкции (M):

$M = 3204$ г $\approx 3{,}2$ кг, что почти в 1,5 раза меньше, чем масса того же P-PODa (согласно [1] чем масса P-PODa примерно равна 5 кг), несмотря на то, что моя конструкция по размерам больше.

Вычисление траектории и скорости полета

Согласно первому закону Кеплера, тело, обращающееся вокруг Земли, движется по эллипсу, гиперболе или параболе в одном из фокусов которого находится центр Земли. Нас интересует только периодические орбиты, а, из перечисленных такой является эллипс.

Основными параметрами эллиптической орбиты являются большая полуось a, определяющая среднее расстояние до притягивающего центра, и малая полуось (см. рис. 2).

Начальная скорость спутника задается отношением [8]:

(15) $v_0 = \sqrt{\frac{\mu}{R_з + H}} = 7{,}78 \cdot 10^3 м/с$

Связь между большой полуосью a и скоростью после сбрасывания спутника v_p определяется выражением [8]:

(16) $v_p = \sqrt{\mu \cdot \left(\frac{2}{r} - \frac{1}{a}\right)}$, где

(17) $v_p = v_0 + v_{спутн}$. Отсюда следует, что

(18) $a = \frac{\mu \cdot r}{2 \cdot \mu - v_p^2 \cdot r} = 6579 \cdot 10^3$ м, где μ – Гравитационный параметр Земли (согласно [8] $\mu = Gm_з$=398600,5 км^3/с^2); $R_з$ – Средний радиус Земли (согласно [8] $R_з$ =6378 $\cdot 10^3$ м); $r = R_з + H$, $H = 200 \cdot 10^3$ м.

Из первого закона Кеплера мы знаем, что Земля находится в фокусе эллиптической орбиты, вокруг которой вращается ИСЗ. Двигаясь по эллиптической орбите, ИСЗ ближе всего подлетает к центру Земли в точке "П" - в перигее, а дальше всего находится в точке "А" - в апогее (см.рис.1).

Расстояние от притягивающего центра до спутника в перицентре задается выражением [8]:

(19) $r_p = a \cdot (1 - e)$, и в апоцентре

(20) $r_a = a \cdot (1 + e)$,

(21) $e = \frac{r_a - r_p}{r_a + r_P}$ – эксцентриситет эллипса. При этом $r_a + r_p = 2 \cdot a$, отсюда следует, что

(22) $r_a = 2 \cdot a - r_p$, где $r_p = r = R_з + H$, Следовательно $r_a = = 6580 \cdot 10^3$ м.

Тогда Из 19 следует, что $e = 0{,}00015$.

Малая полуось b орбиты связана с большой полуосью a соотношением [9]:

(23) $b = a \cdot \sqrt{1 - e^2} = 6578{,}9 \cdot 10^3$м.

Аналогично рассчитываем параметры для конструкции с 30,36,42 конденсаторами. Результаты расчета приведены в таблице (табл. 1), из которой мы можем увидеть, что, регулируя параметры конденсаторов и массы спутника, мы можем регулировать расстояние апогея (орбиту) и скорость спутника.

По уравнениям (22), (7) и сравнительной таблице (табл. 1) построена зависимость (см. приложения график 1). Из нее видно, что необходимость увеличения апогея при запуске, приводит к нелинейному увеличению массы пускового устройства.

Для примерного расчёта скорости ИСЗ на круговой орбите можно воспользоваться формулой из [8]

(24) $$v_{kp} = \sqrt{\frac{\mu}{R_з}}.$$

Максимальная скорость спутника достигается в перицентре орбиты и определяется выражением [8]:

(25) $v_p = \sqrt{\frac{\mu}{p}} \cdot (1 + e)$, и в апогее:

(26) $v_a = \sqrt{\frac{\mu}{p}} \cdot (1 - e)$, p — фокальный параметр, который вычисляется по формуле [9]:

(27) $$p = \frac{b^2}{a} = a \cdot (1 - e)^2.$$

Спутник движущийся по эллиптической орбите, встречает максимальное сопротивление в своем перигее, где плотность среды максимальна, и наименьшее — в апогее. Проходя каждый оборот, через более плотные слои атмосферы он, естественно, выходит из них с меньшей скоростью. Поэтому его апогей снижается поскольку в более высоких слоях атмосферы спутник также встречает некоторое сопротивление, его перигей опускается, но в значительно меньшей степени, чем апогей. Таким образом, с каждым новым оборотом орбита спутника все более приближается к круговой.

При движении по круговой орбите спутник, теряя вследствие сопротивления свою энергию, с каждым витком будет спускаться все ниже и ниже по скручивающейся спирали (см. рис. 3). Вследствие

уменьшения размеров орбиты период обращения будет также уменьшаться. Ниже 110—120 км быстрота увеличения плотности атмосферы резко возрастает и спутник не сможет завершить очередной виток. Тогда спутник, войдя в плотные слои атмосферы, сгорает и разрушается, не становясь космическим мусором.

Сравнение с другими конструкциями

ISIPOD (по данным [10]).

Наноспутник устанавливается внутрь контейнера на платформу и прижимается пружинами к крышке. Электрический импульс от ракеты-носителя активирует механизм выпуска крышки, которая поворачивается на угол 170°. Наноспутники отделяются по направляющим рельсам пружинным механизмом, который сообщает ему заданную скорость. Скорость отделения наноспутника определяется используемой пружиной и массой спутника. Наноспутники размещаются в отдельных адаптерах. Возможно многократное использование адаптера при тестовых испытаниях и для повторного запуска наноспутника.

К недостаткам адаптера ISIPOD можно отнести:

- наличие вибрации наноспутника на стадии вывода ракеты-носителя на заданную орбиту, обусловленной собственными частотами системы прижимных и выпускных пружин контейнера и наноспутника;

- относительно большая масса адаптера по отношению к наноспутнику, до 150% на каждое место установки наноспутника;

- относительно малая скорость отделения наноспутника.

SPL. (по данным [11]).

Наноспутник размещается внутри конструкции корпуса и удерживается на месте «продольной пластины» с помощью четырех противоположных «установочных штифтов» и четырех подпружиненных фиксаторов в двери.

Благодаря фиксирующему механизму дверь защелкивается. Он является резервным и содержит два постоянных магнита (с их помощью магнитное поле может быть компенсировано электрическими катушками) в качестве исполнительных механизмов. Для инициирования необходим короткий электрический импульс одному (обоим) постоянным магнитам, поэтому системы не требуют никакой электрической энергии, чтобы оставаться закрытыми. На этом этапе будет компенсирован постоянный магнитный поток, позволяющий пружине открыть защелку двери. Эта непиросистема позволяет многократно запускать и активировать систему развертывания.

Когда дверь достигает 90º, начинается процесс развертывания. Стандартная скорость развертывания пикосателлита составляет 1 м/с.

Эта скорость также может быть адаптирована к потребностям клиента. Система способна развернуть стандартные CubeSats, то есть с массой 1 кг, или развернуть пикоспутник с массой, скажем, 1,5 кг.

Несмотря на то, что данные способы запуска используются уже давно и имеют свои преимущества, моя конструкция позволяет быстро настраивать систему под нужные параметры орбиты, дает возможность задать начальную скорость и запускать спутники большей массой, такие как CubeSat 2U, 3U и т.д., с помощью регулирования параметров конденсаторов и напряжения, подаваемого на катушку.

Заключение

. В данной работе был проведен расчет характеристик модели пусковой установки, использующей принципы работы пушки Гаусса. Так же была приведена сравнительная оценка плюсов и минусов данной системы по сравнению с эксплуатируемыми в наше время. Как итог, использование данного принципа позволяет расширить возможности использования наноспутников. Так, при необходимости достичь больших значений апогея, потребуется существенное увеличение массы пускового устройства, что мы видим из графика (график 1), представляющего собой нелинейную функцию, которая стремится к насыщению.

Также возможна дальнейшая модернизация данной конструкции, которая позволит выведение на орбиту сразу нескольких спутников

подряд, без потери скорости. Полет группировки малых космических спутников на орбите высотой менее 330 км (орбита с МКС поддерживается в пределах 335—400 км) гарантирует предотвращение столкновения спутников с МКС и с другими космическими аппаратами. Кроме того, модульность данной конструкции позволяет быструю замену поврежденных элементов, что упрощает ее длительное использование.

Список литературы

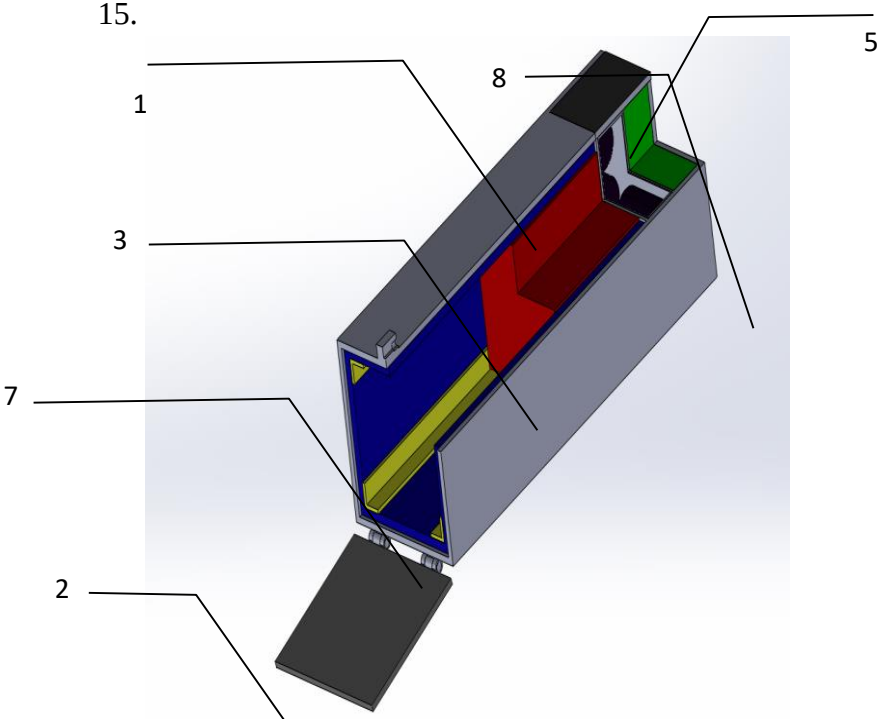
1. CubeSat P-Pod Deployer Requirements, Isaac Nason, Michelle Creedon, Nick Johansen, 2002.
2. http://www.platan.ru/cgi-bin/qwery.pl/id=2010052252
3. https://www.batterika.ru/catalog/show/5862/?utm_content=ch_google_adwords|trg_pla-294682000766|crt_63879731712|kwmt_|ps_1o1|srct_g|trgt_src_devt_m|devm_|cid_242257752|lcl_9047024|fdi_|mrlid_12889|dop_&utm_term=&gclid=CjwKCAiAkrTjBRAoEiwAXpf9CdvDxxLck-2ZXPOW_E05IS24rVT7p5lj76bkPHdUPKiRfcT4RMbStBoCSz0QAvD_BwE
4. http://at-systems.ru/quest/new-quest/what-is-capacity-y.shtml
5. https://ru.wikipedia.org/wiki/пушка_Гаусса
6. https://www.zapas-m.ru/shop/UID_u003c-moshchnyy-povyshayushchiy-dc-dc-preobrazovatel-9v-50v-v-11v-60v-400vt-maks-so-stabilizatorom-t.html
7. Бобошина С. Б., Физика. Справочник, 2014.

8. Д. Е. Охоцимский, Ю. Г. Сихарулидзе "Основы механики космического полета", 1990.
9. https://ru.wikipedia.org/wiki/Эллипс.
10. https://edrid.ru/en/rid/216.012.1c11.html.
11. https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cubesat-concept.
12. Левантовский В.И., "Механика космического полёта в элементарном изложении", М., 1980.

13.Приложения

14.

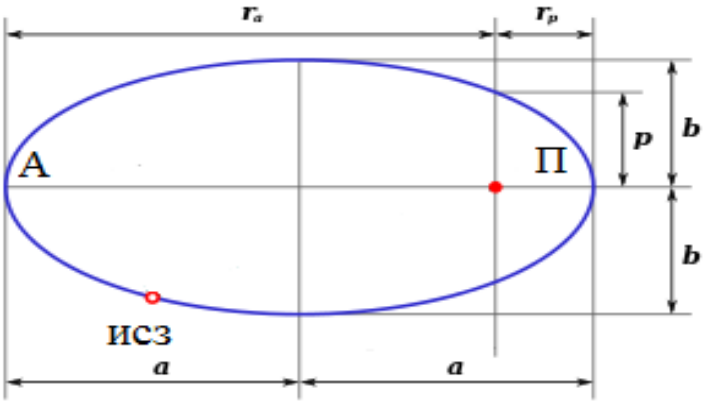
15.



16.

17. Рис. 1 Модель транспортно-пускового устройства для запуска малых космических спутников.

18.



19.

20. Рис. 2 Части эллипсиса.

21. N – количество 22. конденсаторов.	r_a, км.	v_p, км/с	M, г.
30	6584	6,12	3916
36	6586	6,7	4324
42	6588	7,2	4734

Табл. 1 Таблица параметров запуска.

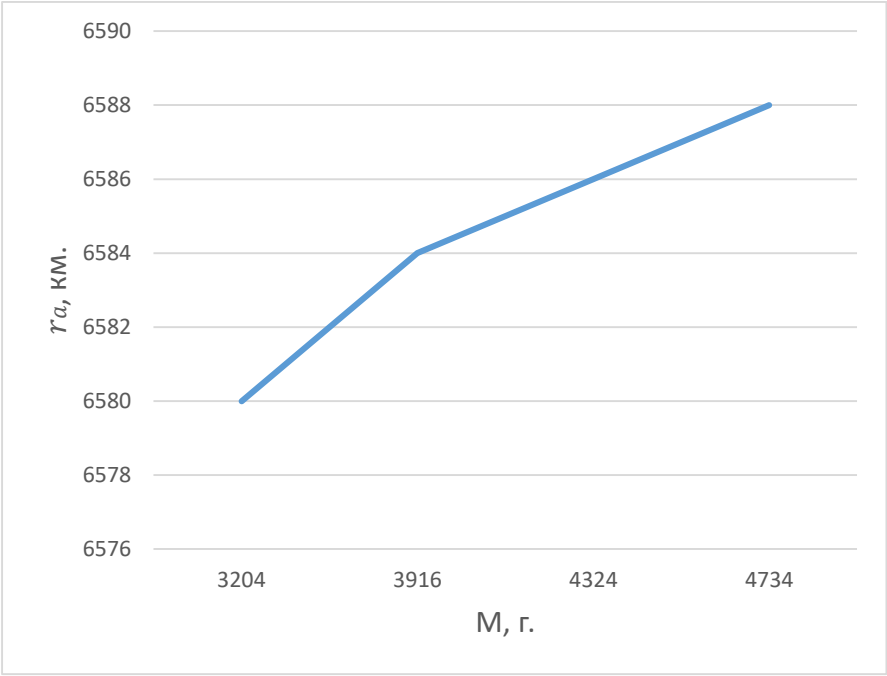


График 1. График зависимости апогея от массы конструкции.

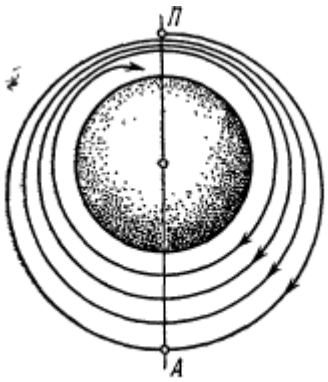


Рис. 3 Согласно [12] снижение спутника в атмосфере.

Секция № 4. Автономные радиоэлектронные устройства управления, автоматические системы и робототехника

СИСТЕМА ПОЗИЦИОНИРОВАНИЯ МОБИЛЬНЫХ РОБОТОВ, ВОСПРОИЗВОДИМАЯ В УЧЕБНЫХ ЛАБОРАТОРИЯХ ЦИФРОВОГО ПРОИЗВОДСТВА

Затекин Дмитрий Владимирович
г. Москва, ГБПОУ «Воробьевы горы»,
11 класс
Научные руководители: Сухоцкий Владимир Андреевич, педагог дополнительного образования, ГБПОУ «Воробьевы горы»
Юдин Антон Владимирович, педагог дополнительного образования, ГБПОУ «Воробьевы горы», инженер, кафедра ИУ4, МГТУ им. Н.Э. Баумана

Проект посвящен разработке системы позиционирования мобильных роботов на основе замера времени прохождения ультразвуковых волн от активных маяков до робота. Система служит элементом навигации мобильных роботов в ограниченном пространстве и может быть использована в робототехнических соревнованиях, для роботов удаленного присутствия, роботов помощников и т.п. Основные требования при проектировании – безопасность, невысокая стоимость, возможность тиражирования в условиях учебных лабораторий. Система апробирована на робототехнических соревнований EUROBOT [1].

Оглавление

Используемые сокращения
Введение

1. Система позиционирования мобильных роботов по методу трилатерации
    1.1. Расчет координат
    1.2 Программа
2. Практическая часть
3. Разработка и отладка системы позиционирования
    3.1. Компоненты системы позиционирования
    3.2. Разработка элементов и тестирование системы позиционирования
4. Усовершенствование системы
5. Изготовление деталей и электронных компонентов
6. Направление дальнейших исследований
Заключение
Информационные источники
Приложения

Используемые сокращения

МР – мобильные робот

УЗ – ультразвук

ИК – инфракрасный

ЧПУ – числовое программное управление

ГОИ – Государственный оптический институт

Введение

Существует множество методов определения координат мобильных роботов (МР). Данный проект разрабатывается для позиционирования роботов, разрабатываемых школьниками и студентами в учебных лабораториях. Поэтому одно из важнейших условий- это недопустимость отрицательных воздействий на здоровье участников. В связи с этим метод на основе лазерного излучения не рассматривался изначально. Второе условие, это невысокая стоимость компонентов и возможность тиражирования в условиях учебных лабораторий. Используя опыт участия в различных робототехнических мероприятиях (Eurobot Open и Junior, Инженерный старт, Балтийский научно-инженерный конкурс, International Exhibition for Young Inventors, Курчатовский проект, РОБОТОН, Техностарт, TECHNOLOGY&SYSTEMS-2018 и подобных), в проект были включены методы определения расстояния [2] на основе ультразвуковых (УЗ) и инфракрасных (ИК) датчиков [3], которые не дороги и наиболее популярны среди начинающих разработчиков робототехнических устройств.

В литературе немного статей с примерами определения координат МР по ИК и УЗ датчикам [4]. В интернете [5] были найдены несколько статей по позиционированию с помощью активных маяков, в которых показана возможность определения координат робота методами триангуляции и трилатерации [6].

1. Система позиционирования мобильных роботов по методу трилатерации

Для проекта была выбрана система позиционирования с помощью стационарных маяков [7], которая не очень удобна при монтаже оборудования, но дает более стабильные результаты при определении

координат и помехоустойчива при отражении УЗ от подвижных и неподвижных объектов.

Схема управления маяками представлена на рис.1. Маяки включают в себя радиоприемник и УЗ излучатель. На роботе установлен центральный узел управления. Робот активирует радиосигналом УЗ передатчик на маяке, фиксирует время прохождения УЗ сигнала, и определяет расстояние до маяка (рис. 1).

Для определения координат используется метод трилатерации. Для работы системы устанавливаются не менее трех стационарных активных маяков в рабочем пространстве МР. Один из маяков определяется, как нулевая точка абсолютной системы координат. Координаты маяков вносятся в программу расчета координат МР. Робот рассчитывает расстояние до каждого из маяков и определяет свои координаты.

На рис.2 приведена схема расположения маяков и МР в относительной системе координат. За нулевую точку отсчета принят первый маяк [Маяк 1(0;0)], маяк 2 смещен на i мм [Маяк 2(i;0)], маяк 3 смещен на i2 мм по оси x и на j2 по оси y [Маяк 3(i2;j2)]. Величина погрешности при замере расстояния до робота в соответствии с техническими характеристиками УЗ датчика может составлять ±3мм. По результатам замеров от трех маяком можно рассчитать зону в которой находится робот.

1.1. Расчет координат.

Для вычисления координат МР, необходима систему из трех уравнений:

$$\begin{cases} L1 = x^2 + y^2 + z^2 \\ L2 = (x - i)^2 + y^2 + z^2 \\ L3 = (x - i_2)^2 + (y - j_2)^2 + z^2 \end{cases} \quad (1)$$

где L1, L2, L3 – расстояния от МР до маяков 1, 2 и 3 соответственно; x, y, z-координаты робота; i - смещения маяка 2 по оси x; i2, j2 - смещение по маяка 3 по оси x, y соответственно.

Выразим три неизвестные координаты:

$$\begin{cases} x = \frac{l_1^2 - l_2^2 + i^2}{2*i} \\ y = \frac{l_1^2 - l_3^2 - 2*x*i_2 + i_2^2 + j_2^2}{2*j_2} \\ z = \pm\sqrt{l_1^2 - x^2 - y^2} \end{cases} (2)$$

В результате вычисления определяются относительные координаты МР. Для получения расстояния от маяка до МР, маяк излучает УЗ сигнал длительностью не менее 8 мкс. При расчете также необходимо учитывать молекулярно-температурные свойства воздуха.

Расстояние высчитывается по следующей формуле:

$$\begin{cases} s = v * \delta + r_{\text{м}} + r_{\text{пр}} \\ v = \sqrt{x * R * T} \end{cases} = s = \sqrt{x * R * T} * \delta + r_{\text{м}} + r_{\text{пр}} (3)$$

где s-расстояние от маяка до МР; ν-скорость звука; δ-время прохождения УЗ сигнала между излучателем и приемником; x-показатель адиабаты; R- газовая постоянная; T- температура воздуха; $r_{\text{м}}$-расстояние от УЗ излучателя до центра маяка; $r_{\text{пр}}$ – расстояние от УЗ приемника до центра МР. Для учета температуры воздуха в центральном узле использовался цифровой датчик температуры (ds1820). Учет температуры в формуле расчета расстояния позволил незначительно увеличить точность системы, поэтому в конечное версии системы датчик температуры не использовался.

1.2 Программа.

Запись формул на языке программирования СИ:

x= (sq (duration1)-sq (duration2) +sq (itwo))/ (2*itwo);

y= (sq (duration1)-sq (duration3)-2*x*itri+ sq (itri) +sq (j))/ (2*j);

z=-sqrt (sq (duration1)-sq(x)-sq(y));

s=sqrt(x*R*T)*t+dr1+dr2;

Состояния системы:

состояние 1 – ожидание включения

состояние 2 – проверка конфигурации системы

состояние 3 – синхронизация маяков

состояние 4 – отправка сигнала МР о готовности системы

состояние 5 – ожидание о запросе координат

состояние 6 – ожидание запроса на УЗ сигнал

состояние 8 – поочерёдное опрашивание маяков

состояние 7 – излучение УЗ сигнала

состояние 9 – отправка координат МР

На рис. 3 представлен алгоритм работы узлов системы: слева – последовательность действий приемника, справа – последовательность действий маяка.

Для ускорения работы платформы Arduino, были использованы нестандартный функции для чтения порта и вывода на него сигнала.

PORTD |= _BV(PD3);

PORTD &= ~ _BV(PD3);

PIND & (1 << 2))

Основные части листинга кода представлен в приложении.

2. Практическая часть

На рис. 4 представлен тестовый вариант проекта, собранной из доступных компонентов. На нем была проверена работоспособность системы и подтверждена целесообразность продолжения работы. Модуль Arduino Nano поочередно активировал УЗ передатчик маяка, принимал полученный сигнал с УЗ приемника и определял расстояние. Расчетные координаты отличались от реальных примерно на 15 см при повторяемости результатов порядка 80%. Работоспособность системы была подтверждена. Были определены эффективные углы расположения датчиков и приемников, что дало возможность приступить к дальнейшему усовершенствованию системы и конструированию корпусных деталей.

3. Разработка и отладка системы позиционирования

3.1. Компоненты системы позиционирования

Для разработки и изготовления компонентов системы позиционирования были подобраны следующие элементы.

Центральный узел. Центральный узел (рис. 16) находится в МР и включает в себя:

- модуль радиопередатчика nrf24l01 частотой 2.4 ГГц, который обеспечивает связь с маяками;

- модуль УЗ приемника hc-sr04;

- модуль Arduino Nano;

- OLED дисплей 128X64 точек 0,96 дюймов I2C, с контроллером SSD1306

- линейный dc-dc преобразовать LM1117MPX-3.3 (3.3v, SOT-223) для питания радиопередатчика nrf24l01;

Стационарный маяк. Стационарный маяк (рис. 8) состоит из следующих элементов:

- модуль радиоприемника nrf24l01 частотой 2.4 ГГц для обеспечения связи с центральным узлом;

- модуль УЗ передатчика hc-sr04;

- управляющей платы модуль Arduino Pro Micro и Arduino Nano;

- линейный dc-dc преобразовать AMS1117-5.0 (5v, SOT-223) для питания всего узла;

- линейный dc-dc преобразовать LM1117MPX-3.3 (3.3v, SOT-223) для питания радиопередатчика nrf24l01;

- аккумулятор 9v.

3.2. Разработка элементов и тестирование системы позиционирования

В ходе работы были смоделированы элементы конструкции приемника центрального узла и маяка, запрограммирована система управления с использованием микроконтроллеров. Детали конструкции и печатные платы для электронных модулей, были изготовлены на станках с ЧПУ.

Второй тест системы был проведен на полигоне для робототехнических соревнований (рис. 5).

На рисунке показано расположение маяков и робота. Три маяка были закреплены над полигоном. Два из них по углам и третий на середине противоположной стороны.

УЗ излучатель, установленный на маяке, имеет эффективный угол покрытия в 60 градусов (рис. 6). Чтобы расширить зону работы, были вмонтированы в каждый маяк по 2 излучателя (рис. 7).

Система работала стабильно. Точность определения координат была в пределах ±10см. Эффективный угол работы УЗ приемников составляет примерно 30 градусов. Для покрытия всего рабочего пространства между маяками потребовалось бы установить на центральный узел 12 приемников. При анализе аналогичных систем было принято решение использовать коническое зеркало (рис. 20). Тестовые испытания показали точность позиционирования в тех же приделах, но при этом зона покрытия расширилась до необходимых размеров.

Электронные модули. На основе схемотехнического моделирования работы электронных компонентов был разработан и изготовлен вычислительный модуль центрального узла системы управления (рис. 7, 8, 16).

4. Усовершенствование системы

В системе использовались УЗ приемники и передатчик на 40кГЦ. УЗ передатчики имеют не большую мощность излучения, поэтому в приёмнике происходит усиление и фильтрации сигнала. В данной работе был изготовлен и протестирован аналог стандартного усилителя. (рис. 8).

Форма сигнала УЗ приемника показана на графике (рис.9). После фильтра сигнал смещен во времени.

В дальнейшем планируется усовершенствование усилителя для работы с более мощными герметичными УЗ излучателями.

5. Изготовление деталей и электронных компонентов

Плата для усилителя и платы для остальных электронных элементов разработаны в среде Eagle Cad (рис. 10). Платы изготовлены на фрезерных станках с ЧПУ (рис. 10, 11, 12).

Элементы конструкции разработаны в среде SolidWorks и изготовлены на 3D принтерах: Ultimaker 2 и Makerbot Replicator (рис. 13,14,15, 16, 17) и лазерном резаке. УЗ зеркало было выточено из дюралюминия на токарном станке и обработано пастой ГОИ до зеркального блеска (Рис. 18).

6. Направление дальнейших исследований

По результатам первых тестов определены следующие направления дальнейшей работы:

- усовершенствование усилителя УЗ с целью увеличения зоны покрытия;

-тестирование усовершенствованной системы под водой;

- увеличение точности позиционирования за счет оптимизации алгоритмов, и увеличения количества датчиков.

Заключение

Достигнута точность определения координат в пределах 10см. Разработку можно считать успешной при сравнении с более точными системами доступными на рынке [8,9], но значительно более дорогостоящими, использующими небезопасное для глаз лазерное излучение. По сравнению с более дешевыми аналогами [10], разработку отличает лучшая точность и отсутствие необходимости дополнительной программной разработки (автоматически определяются координаты МР).

Разработанная система позволяет обновлять координаты положения МР с частотой 4 Гц на дальности до каждого маяка около 3 м.

Испытания показали необходимость установки на робота дополнительных дальномеров для обнаружения препятствий по ходу движения МР.

В развитие проекта целесообразно провести работы для улучшения стабильного приема и увеличения рабочей зоны системы за счет усовершенствования геометрической формы приемного зеркала для УЗ.

Результаты проекта помимо соревнований могут быть полезны при разработке проектов, в которых необходимо обнаружение положения мобильных объектов, например, при разработке роботов-помощников, гидов, в ограниченных пространствах.

Информационные источники

1. Правила молодежных соревнований роботов Евробот 2017: «Город на Луне», основной текст правил: Пер. с англ. с дополнениями НОК Евробот России. / Общ. ред. А.Ю.Вождаев. – М.: Национальный

организационный комитет Евробот России, 2016. – 33с., ил. Режим доступа: http://eurobot-russia.org/main/eurobot-2017 – Проверено 10.02.2018г.

2. Колесников М.А. Аппаратно-программый комплекс по определению местоположения объекта на плоскости методом ультразвуковой триангуляции // сборник научных трудов. 19-ая Молодежная международная научно-техническая конференция "Наукоемкие технологии и интеллектуальные системы 2017". – М.: изд-во МГТУ им.Н.Э.Баумана, 19 апреля 2017 г. – с.234-239.

3. Лапшинов С., Затекин Д., Кириленко В., Гормакова М. Разработка макета изучения передачи ультразвука для применения в образовании // Многопрофильная конференция исследовательских и проектных работ, обучающихся ГБПОУ «Воробьевы горы», 13 февраля 2016 г.

4. Vlasov A., Yudin A. DISTRIBUTED CONTROL SYSTEM IN MOBILE ROBOT APPLICA TION: GENERAL APPROACH, REALIZA TION AND USAGE // Communications in Computer and Information Science. 2011. Т. 156 CCIS. С. 180-192.

5. Сайт Объединенного Студенческого Конструкторского Бюро. Режим доступа: http://class.skycluster.net – Проверено 10.02.2018г

6. Колесников М.А. Аппаратно-программный комплекс по воспроизведению виртуальных изображений на материальной плоскости // Сборник трудов 17-ой международной конференции «Наукоемкие технологии и интеллектуальные системы – 2015» - М.: МГТУ им.Н.Э.Баумана, 22-23 апреля 2015. - С.355-361.

7. Затекин Д.В. Система позиционирования мобильных роботов воспроизводимая
в учебных лабораториях цифрового производства// сборник научных трудов. 20-ая Молодежная международная научно-техническая конференция "Наукоемкие технологии и интеллектуальные системы 2018". – М.: изд-во МГТУ им.Н.Э.Баумана, 25 апреля 2018 г. – с.221-231.

8. Сайт интернет-магазина электроники и робототехники «Robot shop» (ссылка на лидары). Режим доступа: https://www.robotshop.com/en/lidar.html – Проверено 10.02.2018г

9. Сайт гидроакустической системы позиционирования “Track Link”. Режим доступа: http://www.link-quest.com/html/models2.htm – Проверено 10.02.2018г

10. Форум радиолюбителей ссылка на статью «Технологии локального позиционирования». Режим доступа: https://habr.com/ru/company/rtl-service/blog/281837/ – Проверено 10.02.2018г.

Приложения

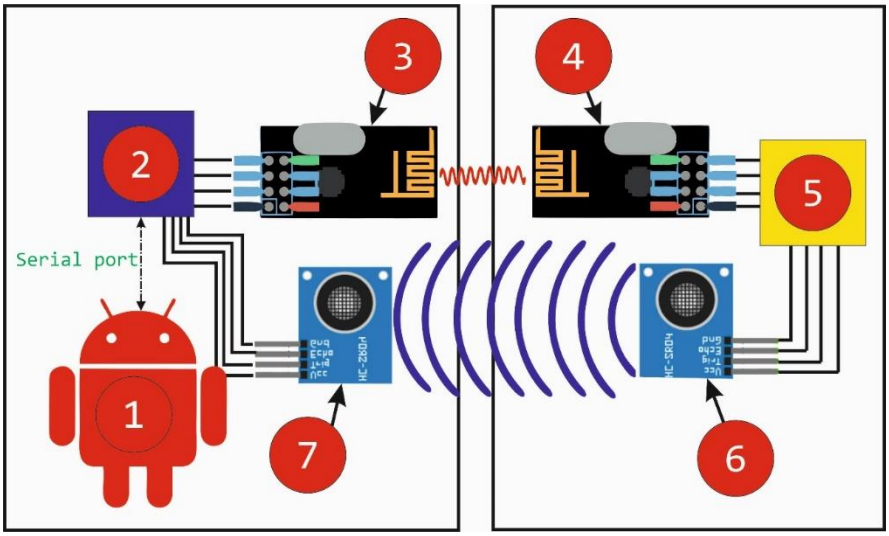


Рис. 1 – Определение положения МР по методу трилатерации. 1 - мобильный робот; 2 - центральный узел; 3 – радиопередатчик; 4 – радиоприемник; 5 – электронный модуль маяка; 6 – УЗ излучатель; 7 - УЗ приемник.

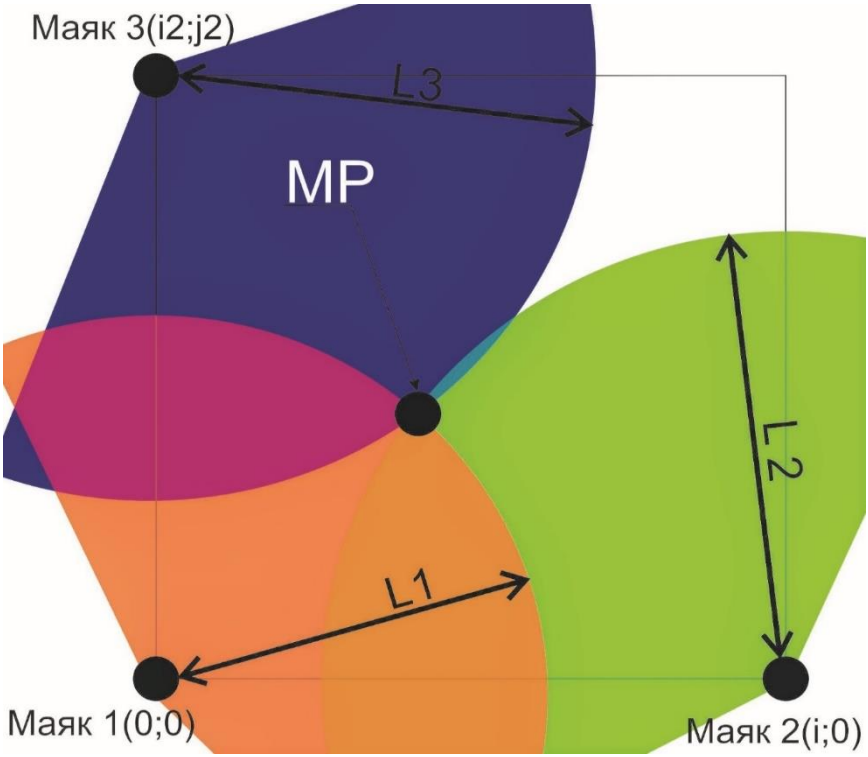


Рис. 2 – Определение координат по трем маякам

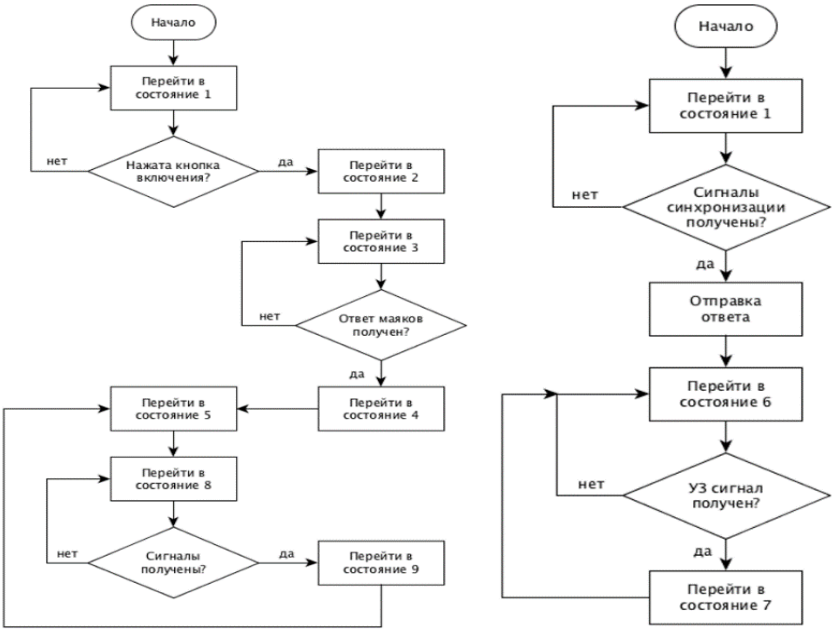


Рис. 3 – Алгоритм работы программы

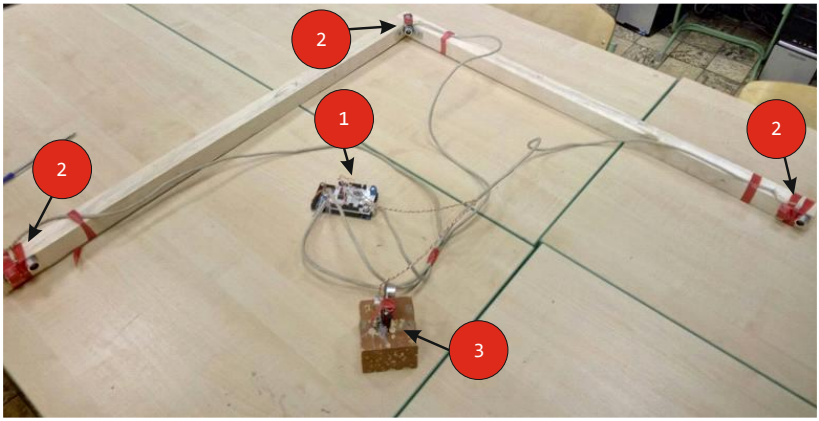


Рис. 4 – Тестовый вариант: 1- модуль Arduino Mega; 2- УЗ передатчики; 3- УЗ приемник

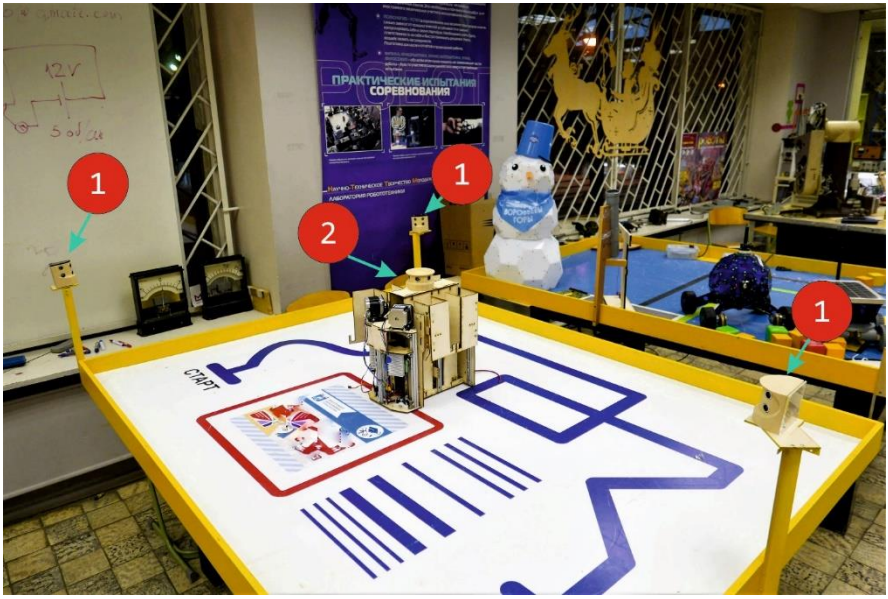


Рис. 5 – Полигон: 1 - маяк; 2 - мобильный робот.

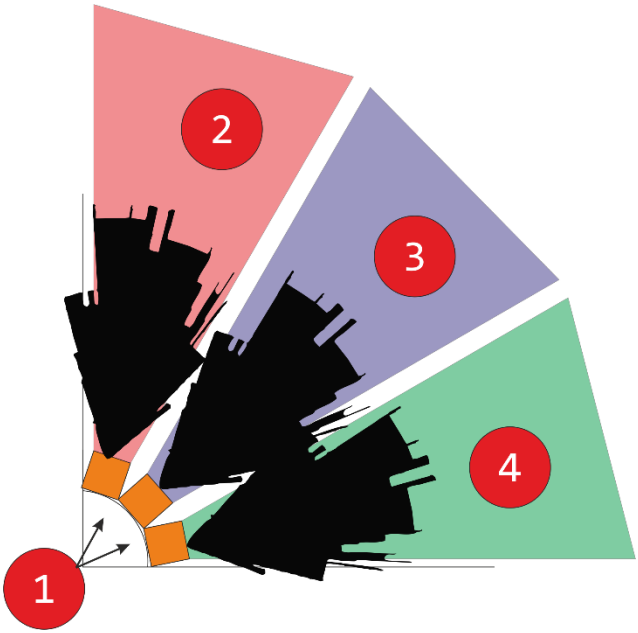


Рис. 6 – Зона покрытия:

1 – УЗ излучатели; 2,3,4 – рабочие зоны излучателя, и диаграмма направленности

Рис. 7 – Маяк

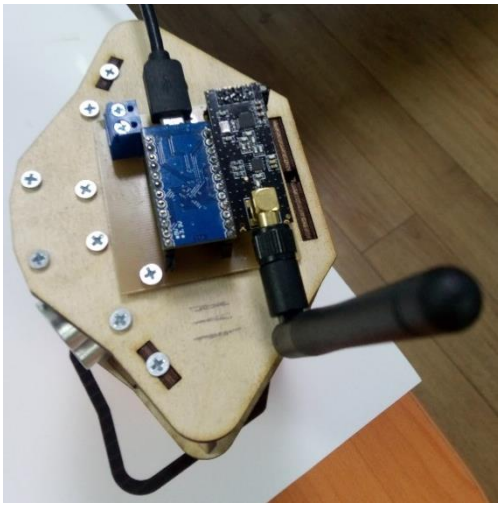


Рис. 8 – модуль центрального узла системы управления

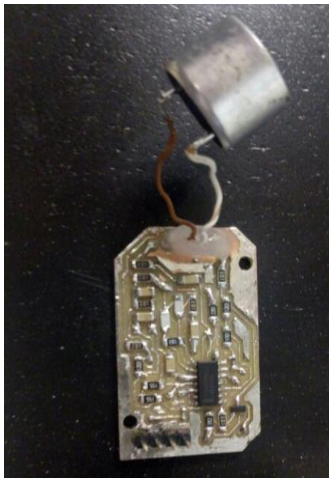


Рис. 8 – Самодельный аналог стандартного усилителя

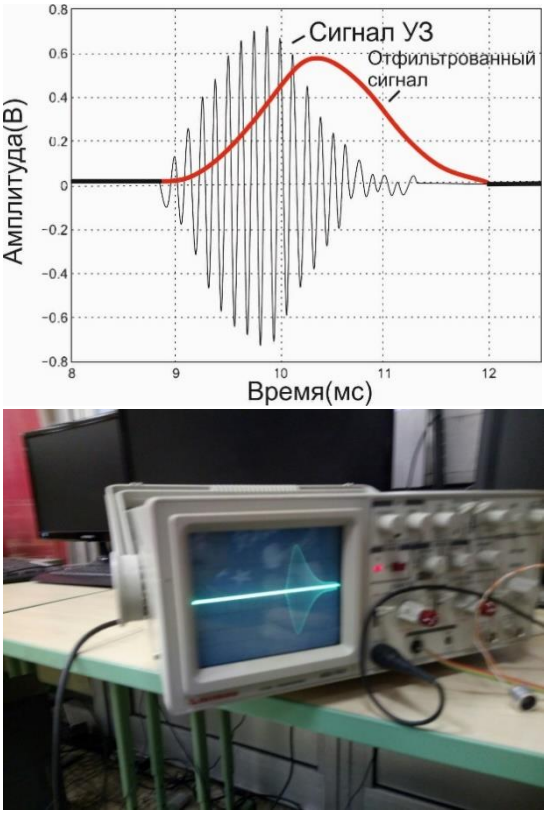


Рис. 9 – График преобразования УЗ сигнала

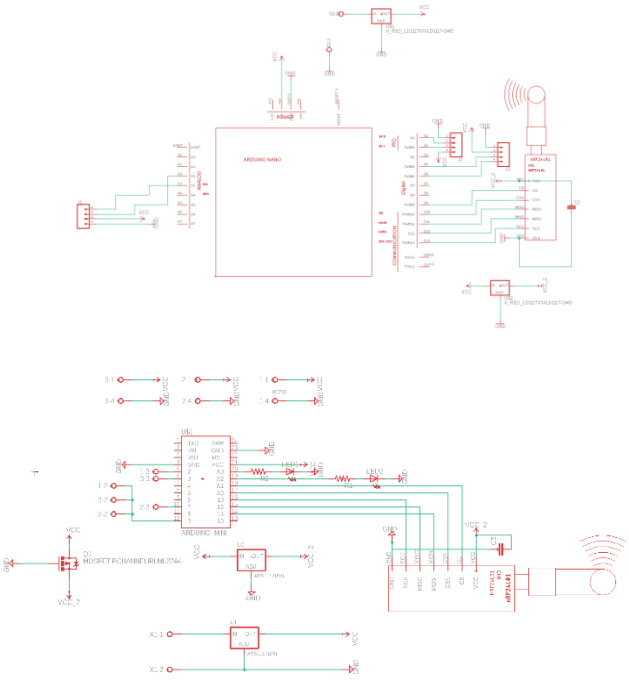


Рис. 10 – Моделирование схемы в Eagle Cad.

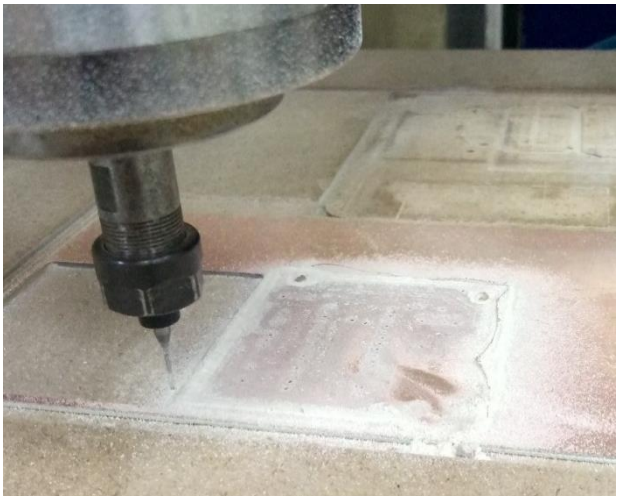


Рис. 11 – Фрезерование печатной платы с помощью концевой фрезы.

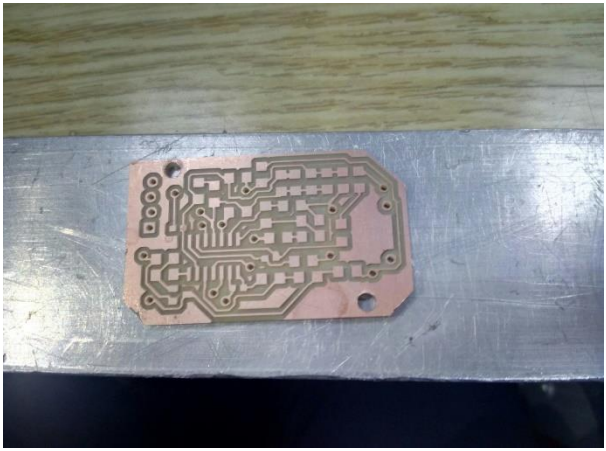


Рис. 12 – Готовая печатная плата.

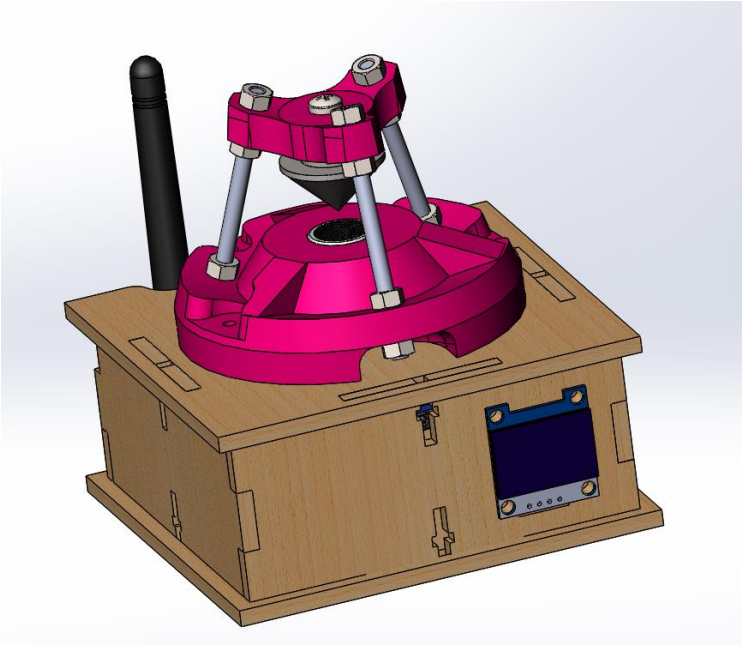


Рис. 13 – Модель центрального приемника, разработанного в среде SolidWorks 2018.

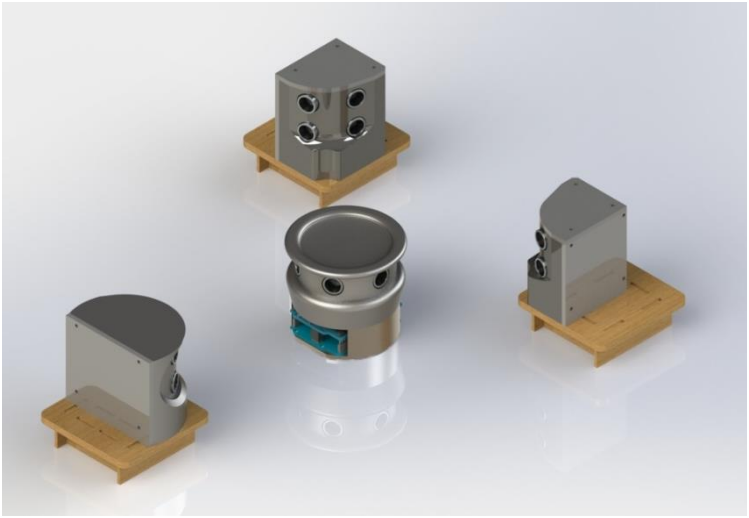


Рис. 14 – Вторая версия системы в среде SolidWorks.

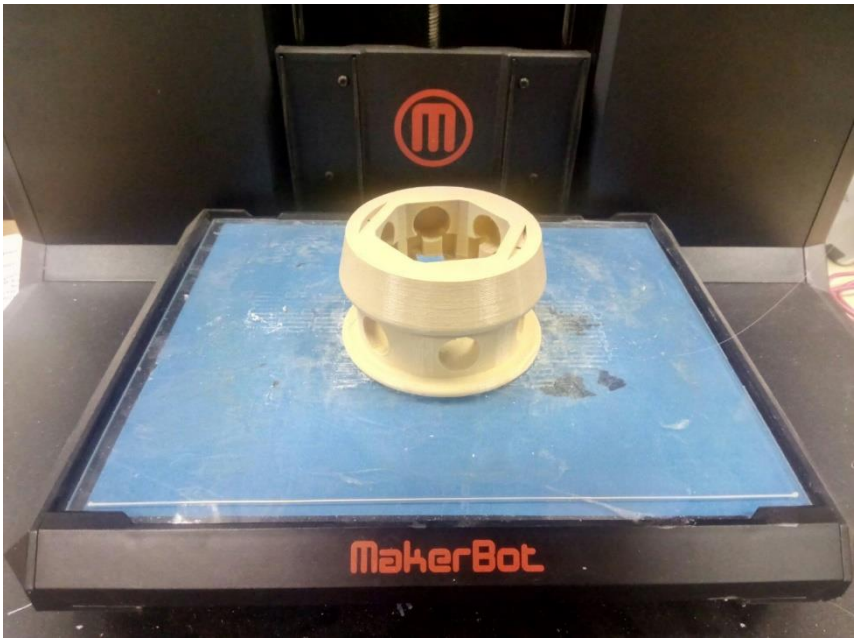


Рис. 15 – Печать корпуса центрального приемника второй версии на 3D принтере.

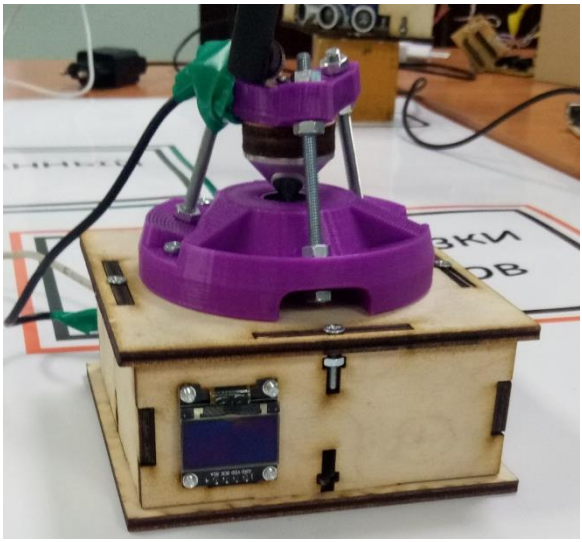


Рис. 16 – Готовая конструкция центрального приемника с УЗ зеркалом.

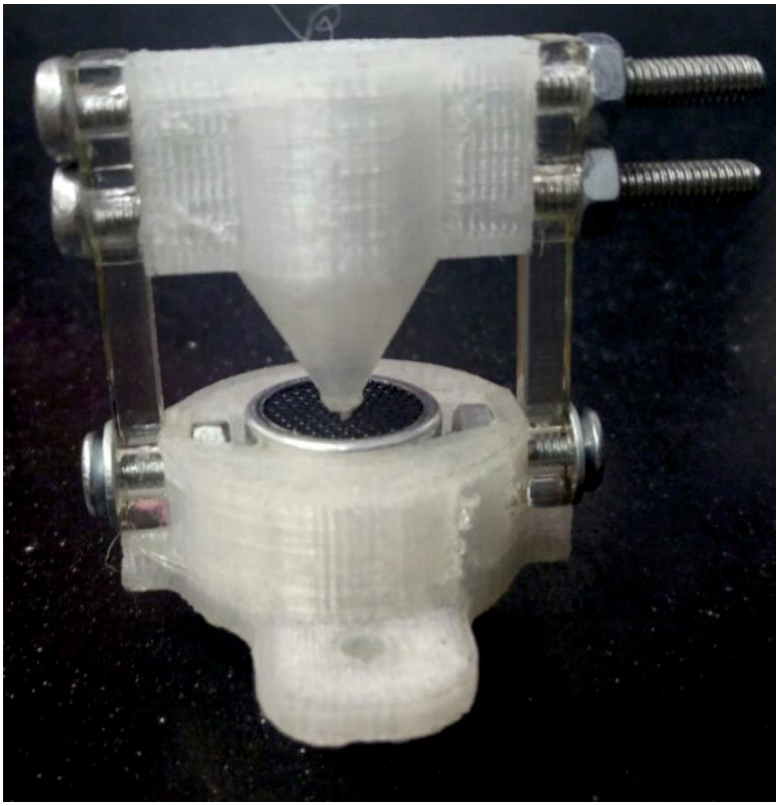


Рис. 17 - Тестовый вариант зеркала УЗ приемника

Рис. 18 – Обработка пастой Гоя УЗ зеркала, выточенного на токарном станке.

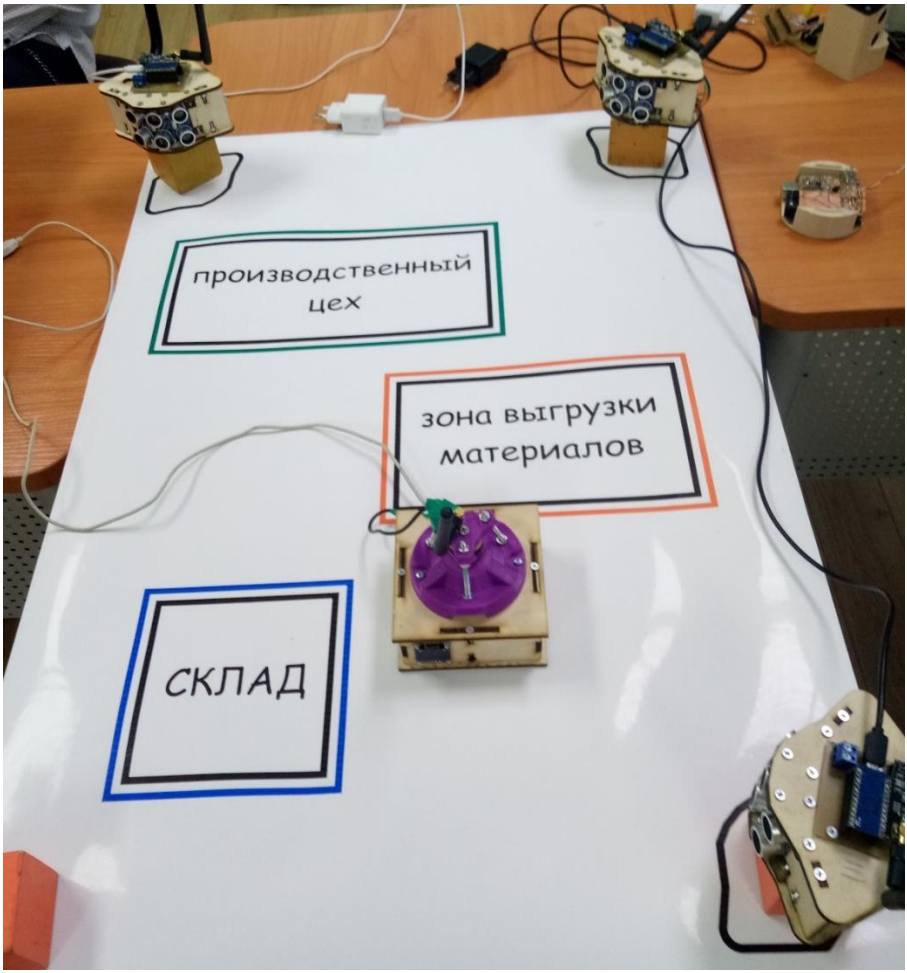


Рис. 19 Демонстрация работы системы на испытательном полигоне.

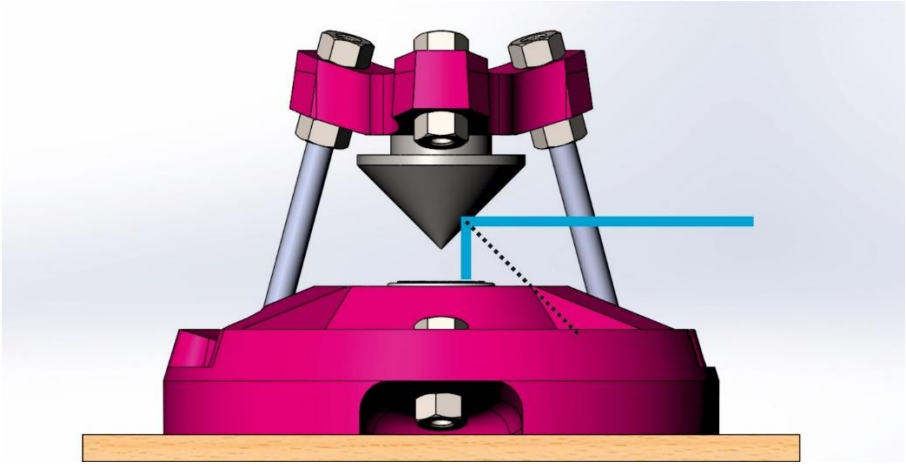


Рис. 20 Отражение УЗ волны от конуса

Листинг кода центрального приемника (основные части кода)

```
void loop() {
  float first22 = obrab("start1", 0);
  float two22 = obrab("start2", 2);
  float tri22 = obrab("s3", 3);
  calculation(first22, two22, tri22);
  Serial.print(x);
  Serial.print(" <- x : y -> ");
  Serial.println(y);
  vivod();
}
```

```
float obrab(String message, int canal)
{
  int io = 0;
  float mas[5];
  String command;
  while (io < 4)
  {
flag:
    transfer_msg(message, canal);
    trigerr();
    mas[io] = obrabotka1() / 29.4;
    delay(70);
    if (mas[io] == 0)
      goto flag;
    io++;
  }
  for (int i = 1; i < 4; ++i)
  {
    for (int r = 0; r < 4 - i; r++)
    {
```

```
      if (mas[r] < mas[r + 1])
      {
        int temp = mas[r];
        mas[r] = mas[r + 1];
        mas[r + 1] = temp;
      }
    }
  }
  return mas[2];
}
void calculation(float dir1, float dir2, float dir3)
{
  x = (sq (dir1) - sq (dir2) + sq (itwo)) / (2 * itwo);
  y = (sq (dir1) - sq (dir3) - 2 * x * itri + sq (itri) + sq (j)) / (2 * j);
}
void trigerr()
{  PORTD |= _BV(PD3);
  delayMicroseconds(10);
  PORTD &= ~ _BV(PD3);}
int obrabotka1()
```

```
{  timing = micros();
  while (1)
  {    if (PIND & (1 << 2))
    {
      time_first = micros();
      while (!PIND & (1 << 2))
      {
        if ((micros() - timing) > 15000)
          return 0;}
      while (PIND & (1 << 2))
      {
        if ((micros() - timing) > 15000)
          return 0;}
      result1 = micros() - time_first;
      if (result1 > 200)
        return int(result1);
      else
        return 0;}}}
```

Секция № 8. Ракетно-космические композиционные конструкции

Эффективность использования электрического паруса

Крутских Анна Андреевна
г. Новомосковск, МБОУ «СОШ №25», 10 класс
Научный руководитель: Николаева Наталья Викторовна, МБУ ДО «Дворец детского (юношеского) творчества»

Оглавление

Аннотация
Введение
Что такое электрический парус?
Возможности паруса. Его достоинства и недостатки
Существующие аналоги
Перемещение существующих космических аппаратов
Вывод
Источники информации

Аннотация

Тема полета к другим планетам достаточно актуальна в настоящее время. Ученые прилагают множество усилий, чтобы создать аппарат, способный за небольшой отрезок времени преодолеть расстояние до ближайших космических тел.

Данная работа посвящена изучению возможности использования электрического паруса для подобных целей. Аппарат на движителе, оснащенном электрическим парусом, является новинкой в области космонавтики. В работе проведено исследование концепции

паруса, выявлены его недостатки и преимущества, рассчитана эффективность паруса для перемещения в межпланетном пространстве космических аппаратов различной массы. Рассчитана скорость перемещения для малых спутников, пилотируемых космических кораблей и автоматических станций.

Работа включает в себя 11 страниц, 2 изображения.

Цель работы: выяснить, насколько эффективным является электрический парус в полете на дальние расстояния.

Задачи:

- изучить концепцию электрического паруса;
- выявить положительные и отрицательные стороны использования данного вида движителя;
- изучить имеющиеся аналоги электрического паруса;
- рассчитать скорость двигателя с электрическим парусом при перемещении различных космических аппаратов.

Введение

С незапамятных времен человеческий взор всегда был обращен к небесам, в Космос. Именно там поколения людей старались найти ответы на многие вопросы, предсказывали будущее или искали разумные цивилизации. С течением тысячелетий интерес человека к космосу не угас, а еще больше усилился, благодаря развитию науки и техники. Многие считают, что в будущем космос является для

человечества единственным спасением, когда на планете не останется никаких условий для существования.

С целью изучения космоса в XX веке началась разработка космических аппаратов. Появились первые двигатели, способные преодолеть притяжение земли и выйти за пределы околоземной орбиты.

В наше время идет активное развитие имеющихся технологий. Возможности человека в космосе возросли в несколько раз.

К сожалению, в настоящий момент двигатели, используемые для освоения космоса, не могут развить достаточную скорость, чтобы в кратчайшие сроки достичь других планет. В связи с этим человечество работает над созданием новых, более мощных способов передвижения в космическом пространстве. Одной из таких новинок стал электрический парус.

Что такое электрический парус?

Электрический парус – вид движителя для космического аппарата, использующий в качестве источника тяги импульс ионов солнечного ветра. Как и солнечный парус, электрический парус использует давление солнечного ветра, но не фотонов, а протонов.

Электрический парус состоит из ряда длинных и тонких тросов, которые имеют высокий положительный заряд, и бортовой электронной пушки. Электронная пушка создает луч электронов, направленный против движения космического

корабля, из-за чего тросы приобретают положительный заряд. Положительно заряженные тросы отталкивают ионы солнечного ветра, что, в свою очередь, приводит к передаче импульса от ионов к парусу иразгону корабля. (Рис.1)

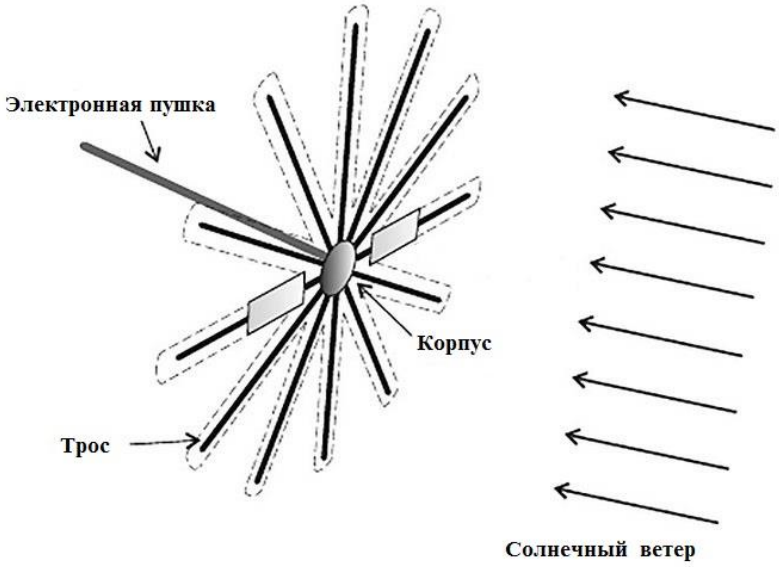


Рисунок 1. Устройство электрического паруса

Электрическое поле достаточной напряженности будет простираться от лучей паруса в пространство на расстояние 100 метров. Заряженные частицы плазмы солнечного ветра, летящие с огромной скоростью, будут сталкиваться с электрическим полем паруса, отталкиваясь от него, как два магнита, развернутые друг к другу одинаковыми полюсами. Это взаимодействие создаст импульс силы,

который будет разгонять космический корабль. На космическом корабле будет установлен источник электрической энергии с высоким выходным напряжением и электронная пушка, с помощью которой на лучах паруса будет создаваться и поддерживаться положительный электрический потенциал. Развернутые лучи паруса будут удерживаться расправленными в космосе за счет центробежных сил, создаваемых вращением всей конструкции вокруг своей оси.

Возможности паруса. Его достоинства и недостатки

Если верить расчёта мученых, электрический парус будет продолжать разгоняться на расстоянии 16-20 или даже 30 а.е. от Солнца. К этому моменту он достигнет очень высокой скорости. К примеру, расстояние примерно в 100 а.е. до гелиопаузы он преодолеет менее чем за 9,9 года с разгоном от Юпитера или за 11 лет без разгона, втрое быстрее «Вояджера-1» и вдвое быстрее аппарата на солнечном парусе.

Главное преимущество подобного паруса в возможности двигаться по направлению к источнику направленных частиц (а не только от него). Кроме того, такой парус проще в производстве и удобнее в эксплуатации: длинный тонкий ус развернуть куда легче, чем натягивать сплошное полотно. Очевидно также, что он куда меньше боится постороннего космического мусора. Но вот сила разгона будет раз в 200 меньше чем у солнечного паруса аналогичной площади.

Важнейшим недостатком паруса является то, что при посадке на другую планету он будет одноразовым — после вхождения в

атмосферу на высокой скорости, аппарат на электрическом парусе будет уничтожен.

Тяга такого движителя невысока: около 10 милиньютонов, но за год она способна разогнать корабль массой 100о кг до скорости 30 км/с. Кроме того, электрический парус может перемещать и более тяжелые грузы при обычной скорости 5-10 км/с. При этом не понадобится выводить в космос тяжелые баки с горючим.

Корабль с электрическим парусом потребует лишь минимального источника энергии на борту. Она будет нужна лишь для того, чтобы поддерживать в парусах-проводах постоянный ток, а также для работы бортовых систем и средств связи.

Для электрического паруса постоянного значения ΔV не существует, но типичные значения для осуществленных миссий составляют 2-4 а.е./год. Для ряда задач, например, для достижения целей на границах Солнечной системы, такие значения ΔV недостаточны. Хотя за счет снижения относительной массы полезного груза до минимума значения ΔV могут быть несколько увеличены.

Существующие аналоги

Одним из аналогов паруса является HERTS. Космический аппарат с электрическим парусом, по некоторым расчетам, может двигаться в 5-10 раз быстрее, чем самый передовой транспорт, построенный нами до сих пор, «Вояджер-1» со своими 17,1 км/с.

Если аппарату "Вояджер-1" потребовалось три десятка лет, чтобы долететь до границы Солнечной системы, то зонд с электронным

парусом должен пройти путь от Солнца до начала гелиопаузы менее чем за десять лет. К слову, это расстояние составляет около 18 млрд км.

HERTS представляет собой массив из одного или двух десятков тонких заряженных алюминиевых проводов суммарной длиной около 20 километров. Его работа основана на электростатическом отталкивании космического аппарата от заряженных частиц. При развёртывании такого паруса в космосе с помощью центробежной силы провода будут охватывать площадь в 219 футбольных полей. Ускорение, которое HERTS получит от солнечного ветра, связано с отталкиванием заряженного паруса от протонов, летящих от Солнца со скоростью от 400 до 750 километров в секунду.

Уменьшенную копию аппарата начали испытывать внутри регулируемой плазменной камеры HISET, где моделируется вакуумно-плазменная среда, приближенная к условиям межпланетного пространства. Результаты испытаний пригодятся для совершенствования всей механики и электроники при конструировании полноразмерной модели. Измеряться будет количество столкновений положительно заряженных электронов и протонов, процент отброшенных протонов, а также объем электронов, которые поглощаются проволокой. Последний показатель особенно важен, так как E-Sail представляет собой систему закрытого типа, которая будет продвигаться в вакуумном пространстве. И если в ней накопится слишком большой заряд статического электричества, парус просто потеряет свою функциональность. Контролировать уровень

заряда должна электронно-лучевая пушка, которая будет выбрасывать "лишние" электроны в пространство.

Второй аналог – парус ESTCube-1, который более полутора лет находился на околоземной орбиту (Рис. 2)

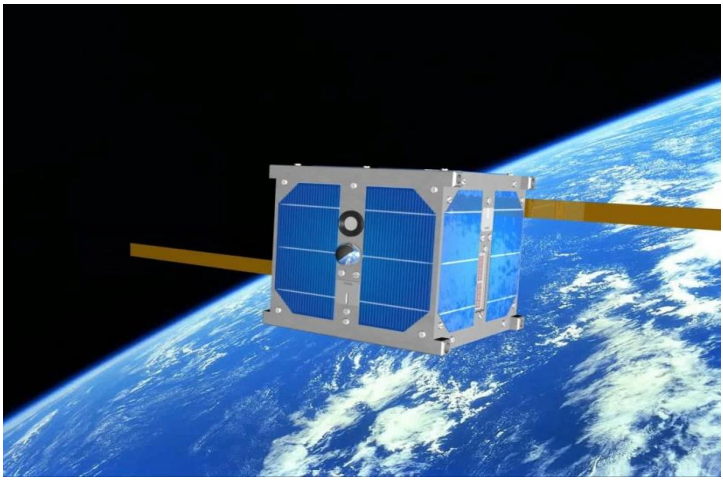


Рисунок 2. ESTCube-1

Эстонский мини-спутник весом 1,3 килограмма был разработан и изготовлен эстонскими студентами и преподавателями Тартуского университета. Он был запущен на орбиту 7 мая 2013 года. Все это время ESTCube-1 летал на высоте 660 километров от поверхности Земли со скоростью 7,46 километра в секунду.

Причиной завершения работы спутника стал износ аккумуляторов. «После отключения электросистем спутник останется в космосе еще примерно на 23 года, пока не опустится в атмосферу Земли, где и сгорит", — сообщил руководитель проекта.

Тросы, выпускаемые спутником, сделаны из алюминия и имеют длину 15 метров.

Перемещение существующих космических аппаратов

Для выяснения эффективности использования электрического паруса, необходимо сравнить его с уже имеющимися двигателями.

Первым аппаратом, данные которого сравнивались с расчетами электрического паруса, стала межпланетная станция «Луна-17», которая достигла поверхности Луны спустя 5 суток. Масса спускаемого аппарата равна 100 кг.

Известно, что за год аппарат с электрическим парусом массой 1000 кг набирает скорость 30 км/с, откуда:

$$F_{\text{тяги}} = a \times m$$

$$a = \frac{\Delta V}{\Delta t} = 0{,}00095057 \text{ м/с}^2$$

$$F_{\text{тяги}} = 0{,}95057 \text{ Н}$$

В данном случае $F_{\text{тяги}}$ – постоянное значение.

Рассчитаем, какое расстояние пролетит аппарат той же массы с использованием электрического паруса.

Ускорение электрического паруса, переносящего данный аппарат, составило бы

$$a = \frac{F_{\text{тяги}}}{m_2} = 0{,}0095057 \text{ м/с}^2$$

Скорость аппарата была бы равна

$$V = at = 4106{,}4624 \text{ м/с}$$

Следовательно, расстояние, которое преодолел бы аппарат за данное время, составило бы

$$S = \frac{V^2}{2a} = 8{,}869958784 \times 10^8 \text{м}$$

Путь, который преодолела «Луна-17», равен $3{,}84399 \times 10^8$м, что в 2,3 раза меньше расчетного. Следовательно, использование паруса в данном случае было бы более эффективным.

Сравним аппарат с электрическим парусом с космическим кораблем «Аполлон-12», который преодолел расстояние до Луны за 4 дня 14 часов 32 минуты 42 секунды.

Ускорение электрического паруса, переносящего «Аполлон-12» массой 43900 кг, составило бы

$$a = \frac{F_{\text{тяги}}}{m} = 2{,}16531 \times 10^{-5} \text{ м/с}^2$$

Скорость, которую космический корабль на электрическом парусе набрал бы за время, понадобившееся «Аполлону-12», чтобы достичь Луны, равна

$$V = at = 8{,}61712 \text{ м/с}$$

Следовательно, расстояние, которое преодолеет аппарат на электрическом парусе за данный отрезок времени, равно

$$S = \frac{V^2}{2a} = 1{,}71465 \times 10^6 \text{м}$$

Путь, проделанный космическим кораблем «Аполлон-12» составил$3{,}84399 \times 10^8$ м, что значительно больше расчетного расстояния для электрического паруса.

Следовательно, электрический парус менее эффективен для перемещения аппаратов большой массы.

Третьим аппаратом, с которым было проведено сравнение, стал «Вояджер-2». За 12860 дней он преодолел расстояние 100 а.е.

Можно вычислить ускорение для аппарата «Вояджер-2» массой 720 кг.

$$a = \frac{F_{\text{тяги}}}{m_2} = 0{,}002705 \text{ м/с}^2$$

Ускорение действует на аппарат только до расстояния в 30 а.е.$=4{,}448 \times 10^{12}$м. Время, необходимое для преодоления данного пути с известным ускорение равно

$$t = \sqrt{\frac{2S}{a}} = 57604682{,}496 \text{ с}$$

Скорость, которую аппарат на электрическом парусе наберет спустя это время, равна

$$V = at = 155820{,}666 \text{ м/с}$$

Оставшееся время аппарат будет идти с постоянной скоростью V. Следовательно, расстояние, которое преодолеет аппарат на космическом парусе с этой скоростью, равно

$$S = Vt = 1{,}641569 \times 10^{14}\text{м}$$

Таким образом, общий путь составит

$$S = 4{,}448 \times 10^{12}\text{м} + 1{,}641569 \times 10^{14}\text{м} = 1{,}686449 \times 10^{14}$$

Путь «Вояджера-2»составил 100 а.е. или $1{,}496\times 10^{13}$ м. Данное расстояние гораздо меньше расчетного, которое мог бы проделать аппарат на электрическом парусе.

Таким образом, электрический парус может быть успешно использован для перемещения в космическом пространстве автоматических межпланетных станций и спутников.

Вывод: использование электрического паруса может быть эффективным только для транспортировки малых спутников и автоматических станций, масса которых невысока. Для перемещения космических кораблей данный вид движителя не рационален.

Источники информации:

1. https://ria.ru/science/20150217/1048198994.html
2. https://hi-news.ru/technology/koncept-elektricheskogo-parusa-poluchil-razvitie.html
3. https://news.rambler.ru/science/33337517-kosmicheskiy-parus-nasa-rabotayuschiy-na-solnechnom-vetre/
4. https://www.innoros.ru/publications/interesting/16/solnechnyi-parus-finskogo-izobretatelya-prinyat-na-ispytaniya-v-nasa
5. https://habr.com/post/393001/
6. https://www.dailytechinfo.org/space/4430-razrabotana-tehnologiya-pozvolyayuschaya-razvernut-kosmicheskiy-solnechnyy-elektricheskiy-parus-radiusom-1-kilometr.html
7. http://www.freepatent.ru/patents/2451629
8. https://ru-universe.livejournal.com/959451.html
9. https://pikabu.ru/story/prostyim_yazyikom_o_kosmicheskikh_dvigatelyakh_5301568