Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП канд.
физ.-мат. наук, доцент
_____ М.Л. Громов
подпись
«09» 06 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ ИССЛЕДОВАНИЯ МЕТОДА ОПТИЧЕСКОЙ ДИАГНОСТИКИ ГЕМАТОМ МОЗГА

по направлению подготовки 03.03.03 Радиофизика
направленность (профиль) «Радиофизика, электроника и информационные системы»

Юрченко Екатерина Алексеевна

Руководитель ВКР
канд. физ.-мат. наук, доцент
_____С.Н. Торгаев
подпись
«9» 06 2023 г.

Автор работы
студент группы № 071906
_____ Е.А. Юрченко
подпись
«9» 06 2023 г.

Томск – 2023

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет

УТВЕРЖДАЮ
Руководитель ООП
канд. физ.-мат. наук, доцент
_____ М.Л. Громов
«03» 10 2022 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра обучающемуся
Юрченко Екатерина Алексеевна
(Ф.И.О. обучающегося)
по направлению подготовки 03.03.03 Радиофизика, направленность (профиль) «Радиофизика, электроника и информационные системы»

1 Тема выпускной квалификационной работы бакалавра
Разработка программно-аппаратного комплекса для исследования метода оптической диагностики гематом мозга

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:
а) на кафедре –
б) в ГЭК –

3 Исходные данные к работе:
Объект исследования – Программно-аппаратный комплекс
Предмет исследования – Метод оптической диагностики гематом
Цель исследования – Разработка программно-аппаратного комплекса

Задачи:
а) Изучение литературы и составление литературного обзора
б) Разработка системы 3D позиционирования
в) Разработка излучателя и приемника излучения
г) Разработка ПО

Методы исследования
Экспериментальный метод

Организация или отрасль, по тематике которой выполняется работа
Медицина и технологии здоровьесбережения

4. Краткое содержание работы
Выпускная квалификационная работа посвящена разработке макета программно--аппаратного комплекса для возможности проведения детальных экспериментов

Руководитель выпускной квалификационной работы
канд. физ.-мат. наук, доцент кафедры _____ / Торгаев С.Н.
(должность, место работы) (подпись) (С.Н. Торгаев)

Задание принял к исполнению
студент _____ / Юрченко Е.А.
(должность, место работы) (подпись) (Е.А. Юрченко)

АННОТАЦИЯ

Выпускная квалификационная работа бакалавра 62 с., 42 рис., 1 таблицы, 19 источников, 2 приложения.

РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ ИССЛЕДОВАНИЯ МЕТОДА ОПТИЧЕСКОЙ ДИАГНОСТИКИ ГЕМАТОМ МОЗГА

Цель работы – разработка программно-аппаратного комплекса для исследования метода оптической диагностики гематом мозга, позволяющего осуществлять автоматизированный сбор экспериментальных данных.

В процессе выполнения выпускной квалификационной работы было сделано следующее:

1. Изучены характеристики и виды гематом, а также определен диапазон длины волны для их обнаружения.

2. Разработаны 3D модели системы позиционирования программно-аппаратного комплекса и осуществлена сборка.

3. Разработана программа управления шаговыми двигателями 17HS4401.

4. Разработана программа для управления системой 3D позиционирования.

5. Разработаны принципиальные схемы, проведена трассировка печатных плат и изготовление печатных узлов излучателя и фотоприемника ИК спектра.

6. Разработаны и изготовлены конструктивные элементы оптических излучателей и приемников для обеспечения необходимой чувствительности.

7. Разработано программное обеспечение для автоматизации экспериментальных исследований.

Программы написаны на языках C++ и C в среде разработки Visual Studio и платформы STM32CubeIDE.

ОГЛАВЛЕНИЕ

Условные обозначения символов, сокращений, терминов 5

Введение 6

1 Обзор литературы по теме исследования 7

1.1 Виды внутричерепных гематом 7

1.1.1 Эпидуральные гематомы 8

1.1.2 Субдуральные гематомы 8

1.1.3 Субарахноидальные гематомы 8

1.1.4 Внутримозговые гематомы 8

1.1.5 Внутрижелудочковые гематомы 9

1.2 Инструментальные методы диагностики внутричерепных гематом 9

1.2.1 Компьютерная томография (КТ) 9

1.2.2 Магнитно - резонансная томография (МРТ) 10

1.3 Обзор аналогов 11

1.3.1 Infrascanner model 2000 11

1.3.2 Микроволновый шлем 12

2 Спектроскопия в ближней инфракрасной области 14

2.1 Метод обработки сигнала 15

2.2 Оптические свойства биологических тканей 18

3 Разработка экспериментального стенда 20

3.1 Конструкция макета для проведения экспериментальных исследований 20

3.2 Разработка излучателя и фотоприемника ИК спектра 21

3.3 Разработка макета системы 3D позиционирования 25

3.3.1 Разработка 3D конструкции экспериментального макета 25

3.3.2 Разработка управления макетом 28

4 Разработка программного обеспечения 33

Заключение 40

Список использованных источников и литературы 41

Приложение А. Код программы для управления двигателями системы 3D позиционирования 43

Приложение Б. Код программы ПО 45

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ СИМВОЛОВ, СОКРАЩЕНИЙ, ТЕРМИНОВ

ЧМГ – черепно-мозговая гематома;

ЧМТ – черепно-мозговая травма

ИК излучение – инфракрасное излучение;

МРТ – магнитно-резонансная томография;

КТ – компьютерная томография;

КОП – коэффициент оптической плотности;

ПК – персональный компьютер;

ШД – шаговый двигатель;

МК – микроконтроллер;

АЦП – аналогово-цифровой преобразователь;

ПО – программное обеспечение

ВВЕДЕНИЕ

В случае черепно-мозговых травм чрезвычайно важно своевременно определить вид и степень тяжести черепно-мозговой гематомы для дальнейшей диагностики и транспортировки пострадавшего. На данный момент достоверными и точными инструментальными методами выявления внутричерепных повреждений являются компьютерная томография (КТ) и магниторезонансная томография (МРТ), однако возможность проведения экспресс-диагностики сразу после травмы почти невозможна.

Таким образом, создание прибора, которое позволит проводить экспресс-диагностику гематом мозга и оперативно выявлять их характеристики, является актуальной задачей. В данной работе исследуется такой оптический метод диагностики как ближняя инфракрасная спектроскопия. Данный метод основан на спектре поглощения молекул различных веществ, интенсивность же поглощения зависит от наличия и концентрации поглощающего компонента в объекте. То есть, если в облучаемом месте головы имеется внутричерепная травма с разрывами сосудов (гематома), то из-за скопления крови концентрация веществ, таких как оксигемоглобина (HbO2) и дезоксигемоглобина (HbH), которые в большей степени влияют на пропускаемый свет из-за повышенной степени поглощения, будет выше в данном участке головы. При пропускании света в диапазоне ближнего инфракрасного излучения (750 – 1400 нм) в направлении гематомы, большая часть излучения поглотится, а малая – отразится.

Стоит отметить, что используемая область спектра безопасна как для выполняющего сканирование, так и для анализируемого объекта (пациента).

Разрабатываемый комплекс можно будет использовать для проведения детальных экспериментов по изучению зависимости изменения оптического сигнала от характеристик гематом.

1 Обзор литературы по теме исследования

1.1 Виды внутричерепных гематом

Гематома — это ограниченное скопление сгустков или жидкой крови в мягких тканях, при закрытых и открытых повреждениях органов и тканей с разрывом (ранением) сосудов, при этом образуется полость с содержанием жидкой или свернувшейся кровью.

Причины возникновения внутричерепных гематом можно условно поделить на два случая:

1) Травматический генез: черепно-мозговые травмы, сильные ушибы головы.
2) Нетравматический генез: резкое повышение внутрисосудистого давления, разрыв аневризмы сосудов головного мозга, артериовенозная мальформация, осложнение гипертонической болезни сердца (геморрагический инсульт).

Скопление крови (гематома) сдавливает мозговую ткань головного мозга, что может привести к летальному исходу или инсульту, если не будет вовремя диагностировано. Для проведения более точных экспериментов и анализа снятых с макета данных необходимо исследовать виды гематом, их различия по глубине залегания между тканями головного мозга. (см. Рисунок 1), а также определить необходимые характеристики для их диагностики.

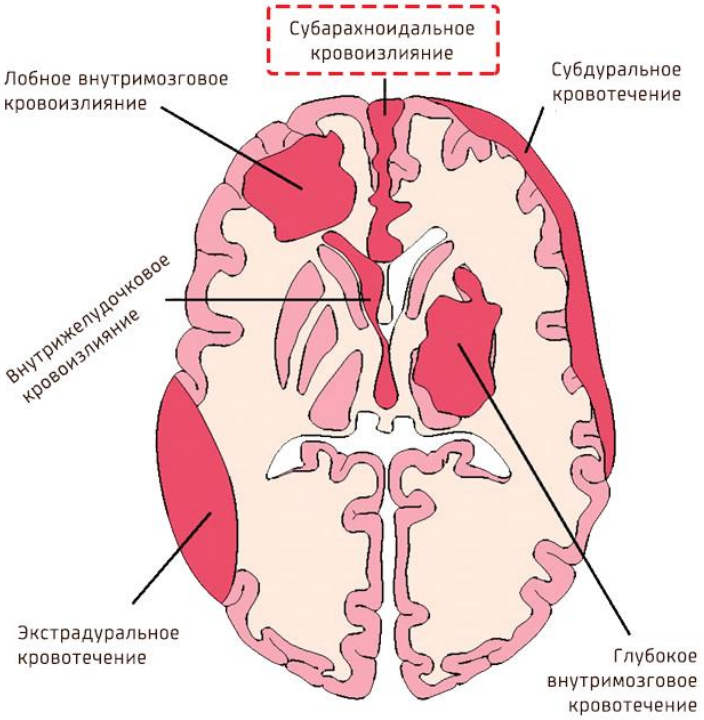


Рисунок 1 – Виды внутричерепных гематом

Характеристиками гематомы является: плотность (количество продуктов распада гемоглобина), размер и местоположение (глубина залегания и виды тканей, между которыми располагается гематома).

1.1.1 Эпидуральные гематомы

Или ее второе название – экстрадуральная гематома, образуется при разрыве менингеальной артерии и ее ветви, менингеальные вены, венозные синусы и диплоические вены, которые располагаются в венозных каналах в толще черепа. Скопление крови заполняет образовавшееся в результате травмы пространство между костями черепа и твердой церебральной оболочкой (костной тканью), что приводи к отслоению последней.

Причиной кровотечений является повреждение сосудов при вдавленном переломе черепа.

1.1.2 Субдуральные гематомы

В большинстве случаев являются следствием ЧМТ. Образуются в следствие разрыва интракраниальных вен, проходящих в субдуральном пространстве и непосредственным травмированием сосудов мозговой коры.. Данная гематома расположена между твердой и паутинной (арахноидальной) церебральными оболочками головного мозга.

Хронические субдуральные гематомы хуже поддаются диагностике ввиду большого интервала между травмой и развитием симптомов, так как по истечению некоторого времени плотность гематомы уменьшается, становясь идентичной плотности мозгового вещества. Постепенное развитие таких симптомов, как ухудшение памяти и сонливость у пожилых людей можно ошибочно принять за деменцию.

1. 1. 3 Субарахноидальные гематомы

Образуются обычно вследствие разрыва артериальной аневризмы, также причиной возникновения такой гематомы могут стать повреждения сосудистой сети мягкой мозговой оболочки. Скопление крови происходит между средней (паутинной) оболочкой и мягкой мозговой оболочкой.

1. 1. 4 Внутримозговые гематомы

Этот тип гематомы, также известной как интрапаренхиматозная гематома, развивается при проникновении крови непосредственно в головной мозг. После травмы головы могут развиться многочисленные внутримозговые гематомы меньшего размера и отличной глубиной залегания.

Данная травма вызывает поражение белового вещества, которое является компонентом проводящего нервного волокна (центральной нервной системы), что приводит к разрыву нейритов в данном белом веществе головного мозга. Такая травма

приводит к поражению головного мозга и временной или постоянной потере функций организма.

Нейриты – это соединения, передающие электрические импульсы или сообщения от нейронов головного мозга к остальным частям организма.

1. 1. 5 Внутрижелудочковые гематомы

Данный тип гематомы является патологией, которая относится к острому нарушению мозгового кровообращения и представляет собой кровотечение в желудочковую систему мозга в результате разрыва сосудов стенки желудочков, геморрагического инсульта или при проникновении крови в желудочки [1].

1.2 Инструментальные методы диагностики внутричерепных гематом

1.2.1 Компьютерная томография (КТ)

Для получения срезов КТ применяется современный рентген-аппарат, подключенный к компьютеру, для выполнения подробных изображений головного мозга.

Принцип компьютерной томографии заключается в том, что рентгеновские лучи томографа проникают в ткани, отображая участки, различающиеся между собой по структуре и плотности. Ввиду того, что излучение проникает в структуру тканей под различными углами, конечное изображение, являющееся непосредственным результатом исследования необходимой области, является многослойным. После завершения процедуры томограф выдает трехмерное изображение, где каждая из структур организма человека, то есть, кости, хрящевые ткани, злокачественные или доброкачественные образования отображаются как отдельный объект. Данные, полученные в ходе КТ, записываются на рентгеновскую пленку или компьютерный диск, который пациент должен предоставить специалисту, назначившему прохождение процедуры.

КТ-изображение представляет собой снимок с информацией об диагностируемых тканях и содержит плотностную характеристику исследованного объекта, что позволяет ориентироваться на объективные плотностные критерии, так как для каждого типа тканей характерен свой интервал отображаемой плотности.

Преимущества КТ:

- является безболезненным методом диагностики, так как не применяются медицинские инструменты и осуществляется без повреждения тканей;
- возможность выявления различий тканей по плотности без теневых наложений, т.к. изображение не проекционное;

- способность получать изображения всех лежащих рядом органов и тканей в одно и то же время, т.е. на одном снимке визуализируются кости, мягкие ткани, сосуды, и в отличие от обычных рентгеновских снимков, их точность очень высока;
- способность диагностирования некоторых заболеваний на ранней стадии, когда еще есть возможность назначить профилактику;
- КТ может быть проведена пациентам с ранее имплантированными в организм медицинскими электронными устройствами, металлическими пластинами и штифтами;

Недостатки КТ:

- приходится небольшая доза диагностического радиационного облучения, если проводятся многочисленные сканирования, то общая доза облучения может быть относительно высокой;
- есть риск развития ряда раковых заболеваний, из-за получения дозы рентгеновского облучения;
- так как в некоторых видах КТ применяют внутривенное рентгеноконтрастное вещество, то есть возможность возникновения аллергических и побочных реакции, такие как химический пневмонит или перфорации кишечника.
- существует ограничение для беременных женщин, из-за потенциального риска для развивающегося плода;
- имеется предел человеческой массы тела, на которую рассчитан движущийся стол.

1.2.2 Магнитно - резонансная томография (МРТ)

Данный метод используется для выявления пораженийи заболеваний практически всех мягких тканей и органов. Изображения при МРТ получают с использованием явления ядерного магнитного резонанса.

Принцип работы МРТ основан на генерации томографом электромагнитных волн, на которые с разной степенью активности откликаются атомы водорода, расположенные в организме человека. Когда делается МРТ, получаемая информация, обрабатывается компьютером и визуализируется на экране монитора. На диагностическую эффективность МРТ влияет степень насыщения водой исследуемой ткани, а именно, мягкотканых структур - экстракраниальных тканей, собственно вещества головного мозга и спинномозговой жидкости.

Преимущества МРТ:

- высокая тканевая дифференцировка мягких тканей в условиях неинвазивного состояния, что позволяет обнаружить патологии;
- МРТ способна четко выявить границы и размеры опухоли, а также степень ее прорастания в мягкие ткани, что не может себе позволить ни одна другая методика;
- при динамическом наблюдении есть возможность многократного повторения;
- не используется рентгеновское излучение, пациент не получает дозы облучения;
- есть возможность выявления на очень ранних стадиях различного вида онкологии в органах и тканях.

Недостатки МРТ:

- на снимках появляются искусственные погрешности, такие как артефакты от дыхания, из-за долгого процесса исследования;
- стоимость и эксплуатация оборудования;
- процедура небезопасна, если ц пациента имеются различного вида артефакты (медицинского назначения или немедицинские объекты из металла);
- при проведении исследования возможен нагрев тканей, что является риском при беременности;
- МРТ небезопасна для больных с аритмией сердца и пароксизмальной активностью головного мозга;
- стоит добавить клаустрофобию и ограничение по весу пацентов;
- специальные требования к помещениям, в которых находятся приборы [2].

1.3 Обзор аналогов

1.3.1 Infrascanner model 2000

Модель представляет портативное сканирующее устройство для первичного обнаружения внутричерепного кровоизлияния и выявления пациентов, нуждающихся в немедленном проведении компьютерной томографии и нейрохирургического вмешательства (см. Рисунок 2).

Рисунок 2 – Infrascanner model 2000

Метод выявления гематомы, используемый в приборе Infrascanner, основан на обработке изображения молекулы гемоглобина, полученного с помощью облучения волнами ИК-спектра ближнего диапазона. То есть имеется оценка уровня поглощения света участка левого и участка правого полушарий головного мозга. При отсутствии гематомы оба полушария в симметричных участках одинаково поглощают свет и отклонений от нормы нет. При наличии внесосудистого сгустка крови, которая поглощает излучение больше, чем внутрисосудистая, так как повышается концентрация (обычно в 10 раз) гемоглобина в остро текущей гематоме по сравнению с нормальной тканью мозга, коэффициент поглощения света возрастает пропорционально уменьшению компонента отраженного света.

Недостатками устройства Infrascanner являются сравнительный анализ симметричных долей левого и правого полушарий мозга при определении наличия гематомы, а также невозможность определения характеристик гематомы (плотность, местоположение, размер) [3].

1.3.2 Микроволновый шлем

Портативное микроволновое устройство или как его ещё называют - «микроволновый шлем», может применяться на догоспитальном этапе для оценки тяжести черепно-мозговой травмы (см. Рисунок 3).

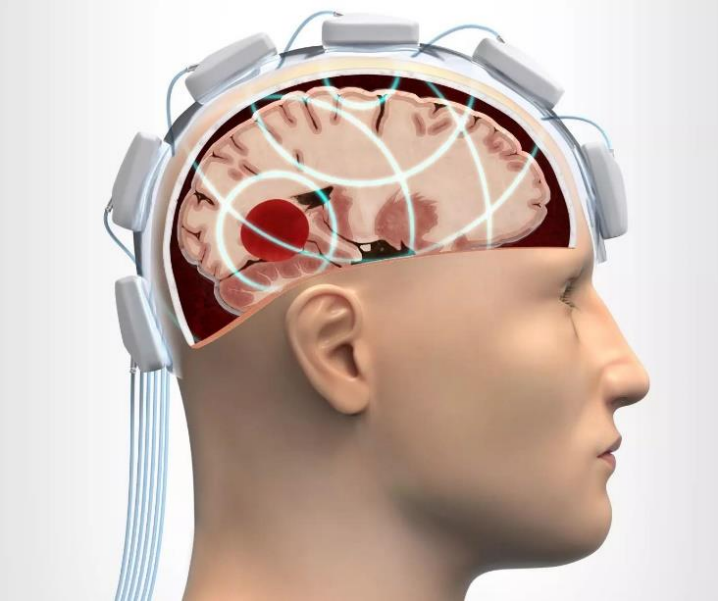


Рисунок 3 – Микроволновый шлем

Принцип работы микроволнового шлема: генерируемые приборов микроволны распространяются по мозговой ткани пациента, по-разному отражаясь от здоровых и пораженных участков. Компьютер собирает и подвергает обработке отраженный сигнал, выдавая на экран информацию о местоположении и размерах кровоизлияния.

Диагностический прибор состоит из трех основных частей: компьютер для сбора и анализа результатов исследования, шлем с микроволновыми антеннами, генератор микроволновых импульсов.

Недостатками устройства является его громоздкость и Санитарно-эпидемиологические требования к помещениям, в которых находится прибор [4].

2 Спектроскопия в ближней инфракрасной области

Спектроскопия ближней инфракрасной области представляет собой метод, основанный на анализе разных объектов и отслеживании изменений биологических тканей, обусловленных способностью поглощать электромагнитное излучение в диапазоне длин волн от 700 до 1400 нм за счет оптических свойств основных хромофоров (окрашенное вещество, способное поглощать кванты света определенной волны).

Данный метод основан на сочетании молекулярной спектроскопии и статистических методов исследования многофакторных зависимостей, который позволяет определить состав объекта без его разложения

Преимущества спектроскопии в ближней инфракрасной области

- возможность быстрой диагностики с созданием биологически безопасных средств;
- существует потенциальная возможность получения изображения исследуемых биологических структур с пространственным разрешением, сравнимых с длиной волны зондирующего излучения;
- возможность многофункциональной диагностики исследуемых объектов;
- уникальные характеристики и особенности оптических свойств биологических тканей;
- наличие терапевтического окна прозрачности биотканей, где поглощение света минимально в диапазоне волн 650–1200 нм.

Метод основан на том, что спектры поглощения молекул являются характеристическими для данного вещества, а интенсивность поглощения связана с содержанием поглощающего компонента в облучаемом объекте. Используемая область спектра безопасна как для человека, проводящего диагностику, так и для анализируемого объекта (пациента). Ближняя инфракрасная область граничит с видимым диапазоном спектра и характеризуется длинами волн от 700 до 1400 нм. Также стоит учесть, что в диапазоне от 600 нм до 1200 нм излучение глубже проникает в биологическую ткань, с минимальными потерями на рассеяние и поглощение, что называется терапевтическим окном прозрачности биотканей. То есть при выборе необходимой длины волны стоит учесть, что в данном диапазоне можно достигнуть глубоко расположенных объектов [5].

В технике инфракрасной спектроскопии ближней области (fNIRS) измеряемое относительное количество поглощенного света с определенной длиной волны может указывать на относительные концентрации оксигемоглобина (HbO2) и

дезоксигемоглобина (HBr) в изучаемой области головного мозга, которые могут быть рассчитаны путем применения модифицированного закона Ламберта-Бера.

В разрабатываемом устройстве диагностика и определение местоположения гематом основаны на том, что гемоглобин является хромофором - веществом, которое поглощает свет на конкретной длине волны. И если в данном месте есть гематома, то из-за скопления крови растет концентрация оксигемоглобина (HbO2) и дезоксигемоглобина (HBr) в данном месте, и при пропускании ИК излучения в диапазоне 700 - 1200 нм в направление гематомы, большая часть света будет поглощена, а малая часть отразится, это и будет означать наличие гематомы в данном месте [5].

2.1 Метод обработки сигнала

Один из методов обработки сигналов является обработка данных, полученных с портативного томографа Infrascanner, с помощью компьютерной томографии.

Infrascanner Model 1000 для обнаружения гематом основывается на сравнении уровней поглощения света участками левого и правым полушариями мозга (см. Рисунок 4). Infrascanner, устройство ближнего инфракрасного спектра, состоит из источника и приемника ИК, и портативного ПК (см. Рисунок 5).

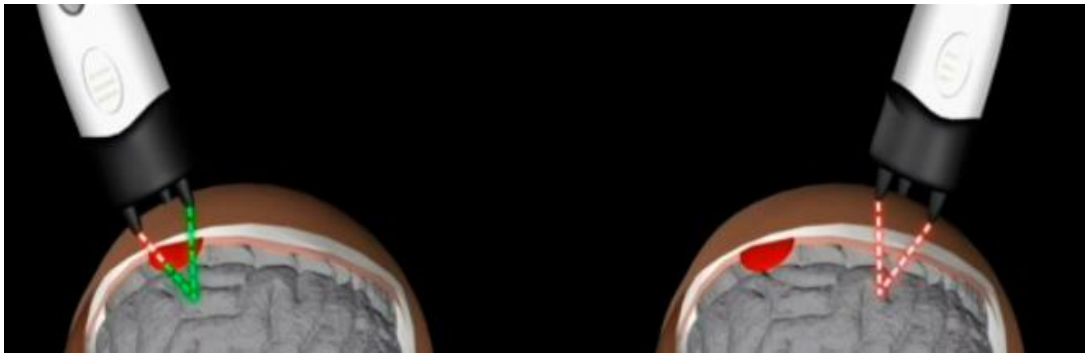


Рисунок 4 – Изображение среза головы

Рисунок 5 – Портативный томограф Infrascanner

Принцип диагностики внутричерепных гематом с использованием инфракрасного сканера основан на обработке изображения молекулы гемоглобина, полученного с помощью облучения волнами ИК-спектра ближнего диапазона. Датчик оснащен источником ИК (NIR, длина волны 808 нм) и кремниевым приемником сигнала (чувствительный элемент), который имеет фильтр полосы пропускания для минимизации помех, вызываемых отраженным сигналом. Датчик, через источник ИК, посылает инфракрасное излучение, которое попадает на область ткани через оптоволокно. Детектор (чувствительный элемент) принимает отраженный сигнал. Далее, проходя через АЦП, сигнал поступает на ПК, который преобразует полученные данные для наглядного представления результата. Оптоволокно легко соприкасается напрямую к коже головы, минуя волосяной покров. Для данной системы работы датчика нет необходимости сбривать волосяной покров головы для проведения исследования.

Сканирования поверхности головы на наличие гематом необходимо в определенных точках для сканирования (см. Рисунок 6). При повреждении рекомендуемых мест локализации датчика можно использовать неповрежденные участки головы. Необходимо, чтобы соблюдалась полная симметрия точек, где происходит сканирование (куда направляется источник ИК).

Рисунок 6 – Расположение точек для проведения БИК-измерения – (F (frontal) – лобный, P (parietal) – теменной, T (temporal)– височный, O (occipital) - затылочный).

О наличии или отсутствии гематомы можно судить по значению коэффициента оптической плотности (КОП). Значение КОП можно найти по следующей формуле:

$$\Delta OD = \log_{10}\left(\frac{I_N}{I_H}\right),$$

где OD- коэффициент оптической плотности, I_N- интенсивность отражения ИК здоровой частью мозга, I_H- интенсивность отражения ИК гематомой.

При КОП>0,2 с помощью пары источник – приемник ИК зафиксирована внутричерепная гематома. Если при измерении КОП отличается от 0,2 или больше 0,2, то

пара источник – приемник ИК производит три последовательных измерения, для подтверждения наличия гематомы. Если КОМ меньше 0,2, то никакой гематомы нет.

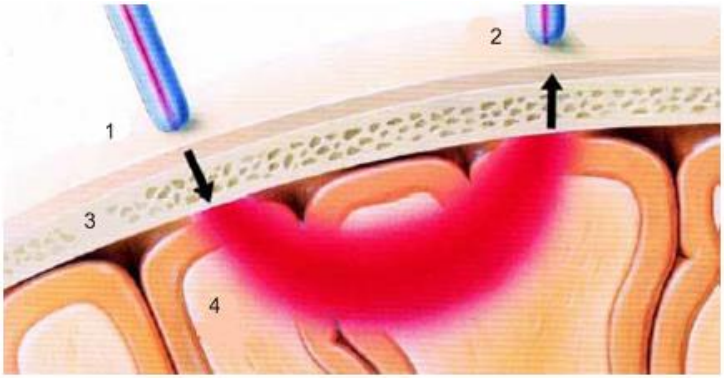


Рисунок 7 – Условное изображение прохождения ближнего инфракрасного излучения через ткань: 1 – источник излучения; 2 – приемник излучения; 3 – костная оболочка; 4 – мозговая оболочка.

Так как источник и приемник располагаются на расстоянии 4 см друг от друга, возможно измерять ИК поглощение на площади глубиной 2–3 см и шириной до 2 см.

На рисунке 6 представлен один из методов оценки результатов. Единственное отличие от вышеописанной системы снятия данных состоит в том, что прибор на данный момент не соединен с ПК и совершает только горизонтальные перемещения, что не позволяет определить глубину гематомы. Для возможности определения глубины залегания внутричерепной гематомы необходимо добавить еще два или пять датчиков, которые будут снимать вертикальный сигнал под углом с нескольких сторон для более полной картины, расположение и количество датчиков нужно определить экспериментально. Далее, используя формулу коэффициента оптической плотности, можно определить наличие гематомы в исследуемом участке головного мозга.

После определения КОП можно составить график зависимости КОП от расстояния, на которое мы перемещаем горизонтальный датчик. Данный график поможет определить площадь распространения кровоизлияния, а данные с вертикальных датчиков помогут отслеживать глубину гематомы. Также нужна реализация графика зависимости КОП от характеристик гематом для оценки влияния характеристик на сигнал.

На основании того, что принимающие датчики будут иметь довольно подвижное крепление, они могут менять угол поглощения ИК источника. За счет такой конструкции мы сможем снимать необходимые для анализа данные в несколько раз быстрее. После исследования будет составляться графики зависимости КОП от угла, направленного ИК, и зависимость перемещения с нулевой точки отсчета от КОП.

На основании построенных графиков можно будет сделать вывод о наличии или отсутствии гематомы, а также ее характеристики [6].

2.2 Оптические свойства биологических тканей

Биологические ткани являются оптически неоднородными поглощающими средами. Средний показатель преломления биотканей больше, чем у воздуха, поэтому на границе раздела биообъект – воздух часть излучения отражается, а остальная часть проникает в биоткань. Развитие молекулярных процессов и биологических реакций под действием лазерного излучения на биологический материал обусловлено взаимодействием фотонов с молекулами и соединениями молекул биоткани.

При падении пучка лазерного излучения на поверхность биологической ткани могут наблюдаться такие процессы, как: отражение, поглощение, рассеяние и пропускание.

Объемное рассеяние является причиной распространения значительной доли излучения в обратном направлении. Основными источниками рассеяния света в биотканях являются такие причины, как: негомогенность характера структуры биоткани, ячеистая структура и различие в значениях показателей преломления различных компонент биотканей (т.е. между митохондриями, ядром, другими компонентами и цитоплазмой клеток; или внутритканевой жидкостью и фиброзной ткани).

С оптической точки зрения, биоткани (включая и биожидкости: кровь, лимфу и пр.) можно разделить на два больших класса:

- Класс 1: сильно рассеивающие (оптически мутные), такие как кожа, мышцы, хрящ, мозг, стенка сосуда, кровь, склера, оптические свойства которых могут быть достаточно хорошо описаны в модели многократного рассеяния скалярных волн в случайно–неоднородной среде с поглощением;
- Класс 2: слабо рассеивающие (прозрачные), такие как ткани переднего отрезка глаза (роговица, хрусталик), оптические свойства которых описываются в модели однократного (или малократного) рассеяния упорядоченной среды с плотной упаковкой рассеивателей, которые содержат поглощающие центры.

Ослабление коллимированного (лазерного) пучка света в биоткани происходит по экспоненциальному закону, интенсивность прошедшего коллимированного света может быть оценена на основе закона Бугера–Бера:

$$I(z) = (1 - R) * I_0 * e^{-\mu_t z},$$

где R – коэффициент френелевского отражения (при нормальном падении пучка), I_0 – интенсивность падающего света, μ_t – коэффициент экстинкции (коэффициент ослабления), z – толщина образца.

Интенсивность коллимированного пучка света, проникающего через слой рассеивающей ткани со средней плотностью рассеивателей ρ и толщиной l, определяется соотношением:

$$I(z) = I_0 * e^{-\rho\mu_s l},$$

где I_0 — интенсивность падающего пучка, а сечение рассеяния определяется по формуле [7]:

$$\mu_s = \frac{1}{\rho I_0} * \int_{4\pi} I(\theta) d\Omega.$$

3 Разработка экспериментального стенда

3.1 Конструкция макета для проведения экспериментальных исследований

Был разработан экспериментальный макет для проведения исследований и снятия пробных экспериментов по изменению сигнала при прохождении через слоистые материалы (см. Рисунок 8). [8]

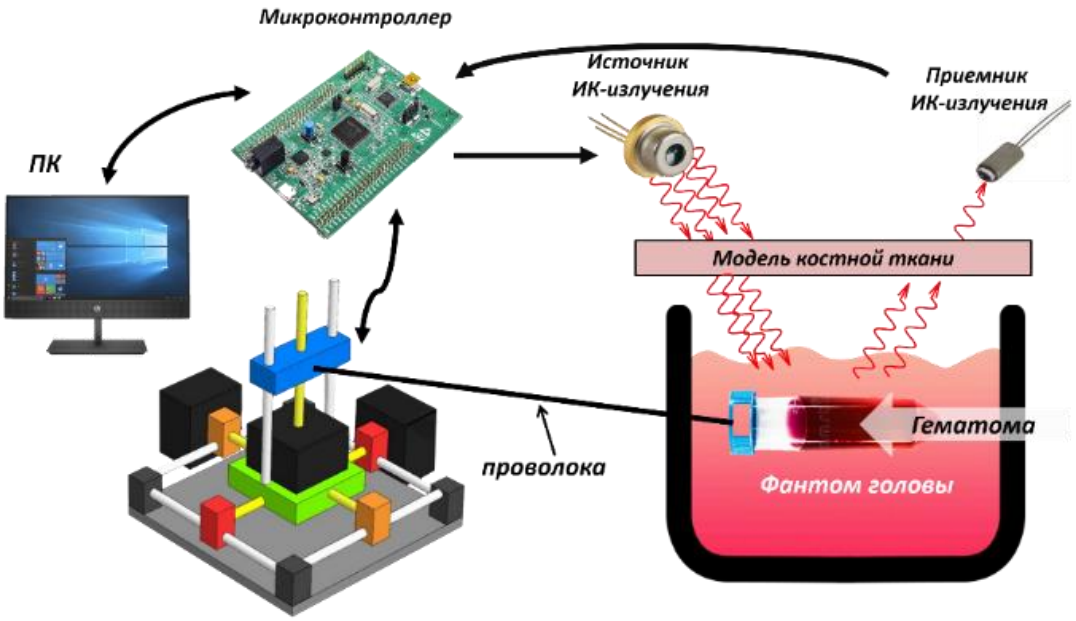


Рисунок 8 – Экспериментальный макет программно-аппаратного комплекса.

Макет содержит систему 3D позиционирования, которая имеет крепление и проволоку для фиксации модельного объекта, имитирующего гематому. Фантом головы представляет собой емкость, наполненную дисперсной густой жидкостью, фторопластом и силиконом, имитирующих оболочки головного мозга человека. Аналоги подобраны по схожести оптических свойств с тканями человека.

Оптический блок данного комплекса состоит из источника и приемника излучения, которые расположены над модельной средой имитирующей структуры головы (фантом головы). Управляющей частью является микроконтроллер, в качестве которого была выбрана микросхема STM32F407 (см. Рисунок 9). Это 32-разрядный микроконтроллер с плавающей точкой. Использование данного микроконтроллера обусловлено тем, что возможно в последующем будет необходимо осуществлять некоторую математическую обработку принимаемых сигналов. Также данный микроконтроллер обладает весьма скоростным АЦП, до 2.4 мегавыборок в секунду, что обеспечит повышение точности последующего анализа [9].

Рисунок 9 – Микросхема STM32F407

Для управления работой данного программно-аппаратного комплекса и сбора данных используется ПК с разработанным ПО.

3.2 Разработка излучателя и фотоприемника ИК спектра

Были разработаны принципиальные схемы источника и приемника излучения (см. Рисунки 10-11).

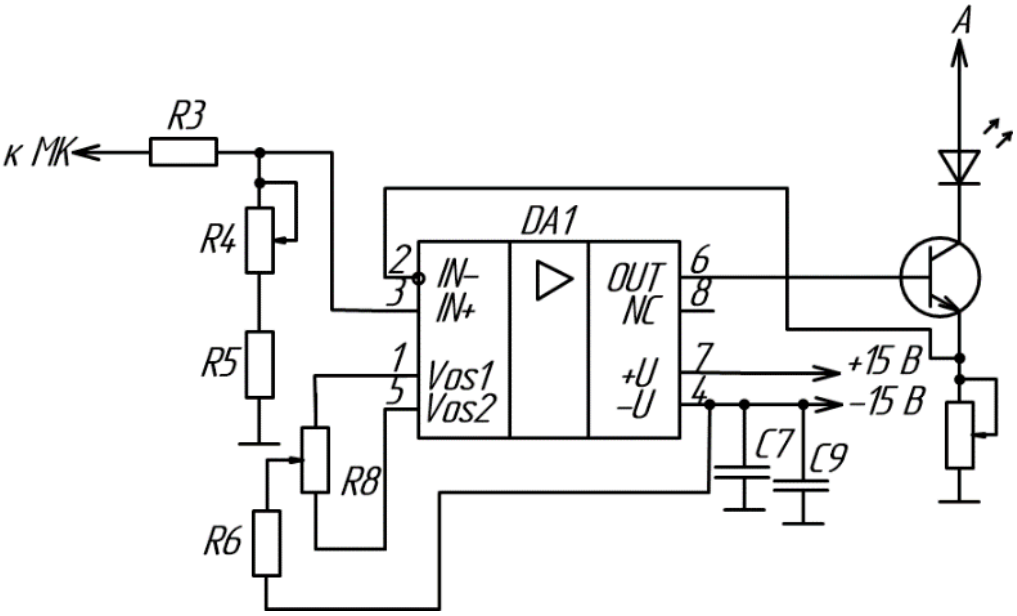


Рисунок 10 – Принципиальная схема источника излучения

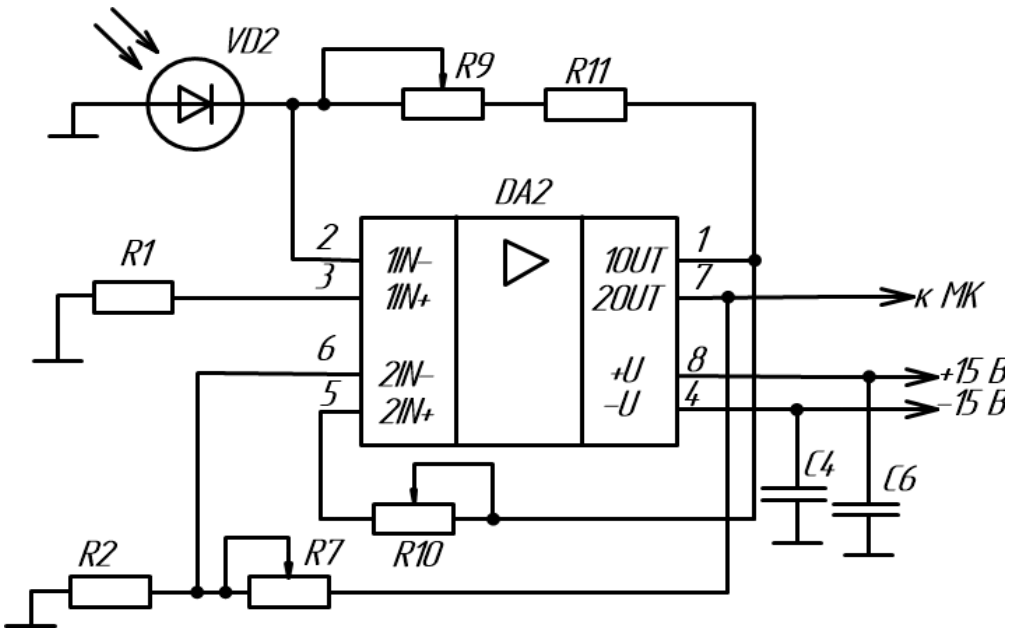


Рисунок 11 – Принципиальная схема приемника излучения

В качестве источника излучения был выбран лазерный диод L808H1 (см. Рисунок 12), который обладает следующими характеристиками:

- рабочая длина волны – от 798 до 818 нм;
- максимальная выходная мощность – 300 мВт;
- максимальный рабочий ток – до 450 мА;
- прямое падение напряжения при токе 400 мА – 2,1 В;
- рабочее напряжение: 3,1 В [10].

Рисунок 12 – Лазерный диод L808H1

При выборе источника излучение, был учтен спектр поглощения крови. Соответственно, при длине волны от 700 до 1400 нм (что соответствует ближней ИК области спектра), интенсивность отражения излучения несвернувшейся крови превосходит в примерно шесть раз интенсивность отражения свернувшейся крови.

Управление источником излучения реализовано на операционном усилителе TL071C со следующими техническими характеристиками:

- напряжение нагрузки: ± 20 В;
- напряжение питания: ± 15 В;
- максимальный ток входного смещения: 200 пА;
- максимальный ток нагрузки: 2,5 мА [11].

В качестве приёмника излучения был выбран кремниевый фотодиод Ф – 256 в металлостеклянном корпусе (см. Рисунок 13), который обладает следующими характеристиками:

- диапазон спектральной чувствительности – от 0,4 до 1,1 мкм;
- рабочее напряжение – 5 В;
- темновой ток при 20 ºС – ≤0,1 мкА;
- рабочая температура – от -60 ºС до +85 ºС;
- интегральная чувствительность: ≥ 5 мА/лм [12].

Рисунок 13 – Фотодиод ФД – 256

Схема приёмника реализована на операционном усилителе TL072C со следующими характеристиками:

- напряжение нагрузки: ± 18 В;
- напряжение питания: ± 15 В;
- ток входного смещения: 200 пА;
- ток нагрузки: 2,5 мА [13].

Были разработаны основные узлы печатных плат для источника (см. Рисунки 14-15) и приемника излучения (см. Рисунки 16-17).

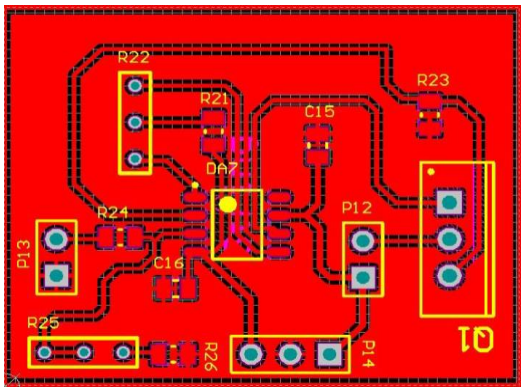


Рисунок 14 – Трассировка платы TL071C

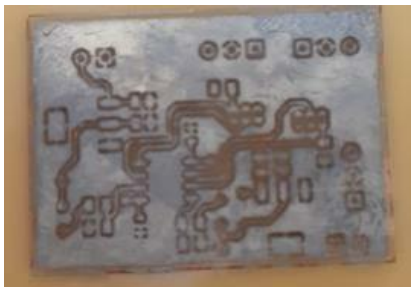


Рисунок 15 – Реализованная печатная плата TL071C (источник излучения)

Размер печатной платы TL071C – 33,27×26,3 мм. Длина дорожек питания – 0,4 мм, длина основных дорожек – 0,254 мм. Односторонняя печатная плата.

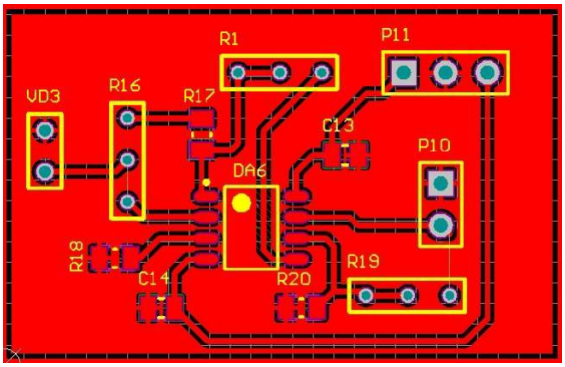


Рисунок 16 – Трассировка платы TL072C

Рисунок 17 – Реализованная печатная плата TL072C (приемник излучения)

Размер печатной платы TL072C – 33,75×21,5 мм. Длина дорожек питания – 0,4 мм, длина основных дорожек – 0,254 мм. Односторонняя печатная плата.

3.3 Разработка макета системы 3D позиционирования

Была смоделирована и разработана система 3D позиционирования (см. Рисунок 18), которая включает в себя опорные детали, движущуюся подставку, деталь для возможности перемещения пробирки с кровь по вертикали, крепление для аналога гематомы и шаговые двигатели [14,15].

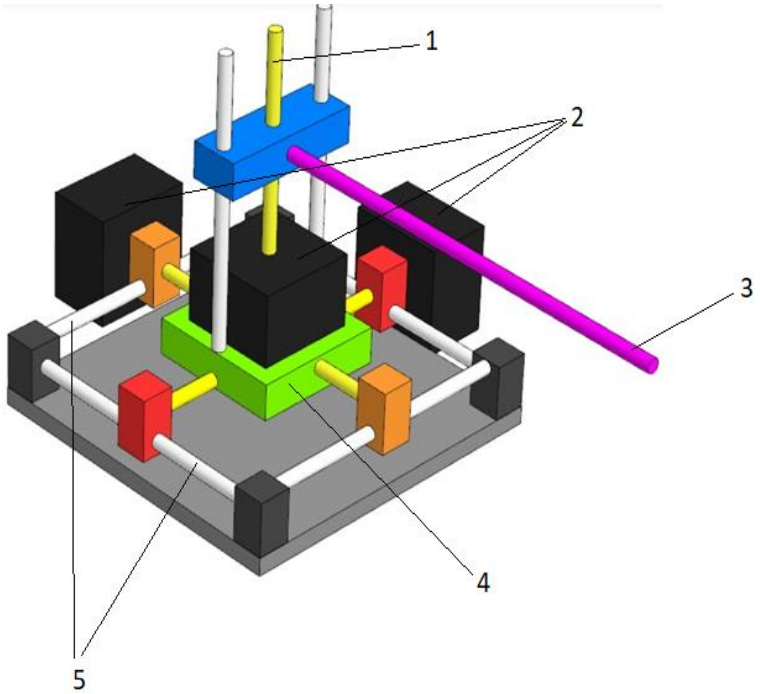
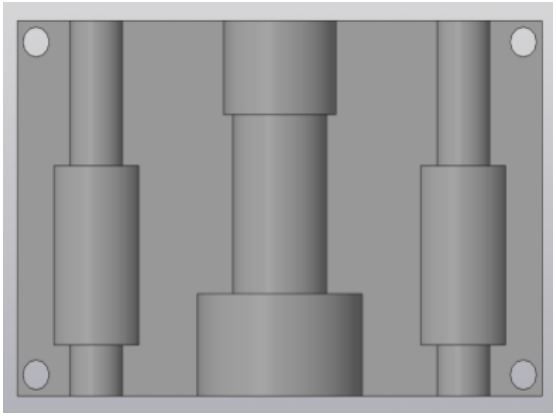


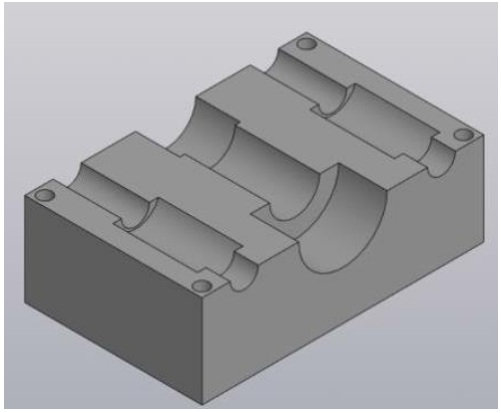
Рисунок 18 – Система 3D позиционирования: 1–шпилька; 2–шаговые двигатели; 3–крепление для перемещения пробирки с кровью; 4–движущиеся подставка; 5–направляющие

Перемещение движущейся подставки (4) осуществляется с помощью шаговых двигателей 17HS4401 (2) и шпилек (1). В данной конструкции присутствуют направляющие (5) для более точного положения аналога гематомы в пространстве (фантоме головы). Аналог гематомы при помощи проволоки (металлической лески) закрепляется на креплении (3) для проведения экспериментальных исследований. Данная система позволяет перемещать аналог гематомы в пространстве, то есть на нужное число заданных шагов по необходимому направлению, что позволит изменять глубину залегания и местоположение аналога гематомы в фантоме головы.

3.3.1 Разработка 3D конструкции экспериментального макета

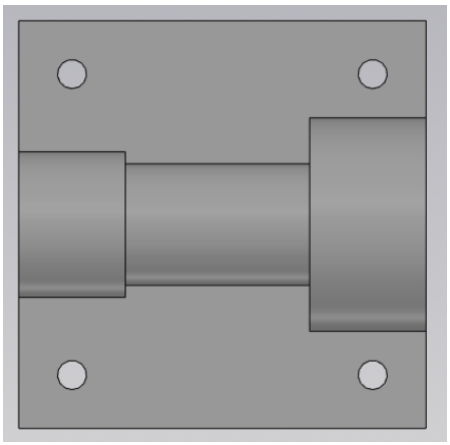
В программе САПР КОМПАС 3D были смоделированы детали экспериментальной установки, такие как: движущаяся подставка, крепление для аналога гематомы (деталь, которая обеспечивает перемещение колбы с кровью по вертикали), детали, на которой закрепляются направляющие и опорные детали (см. Рисунки 19-22).

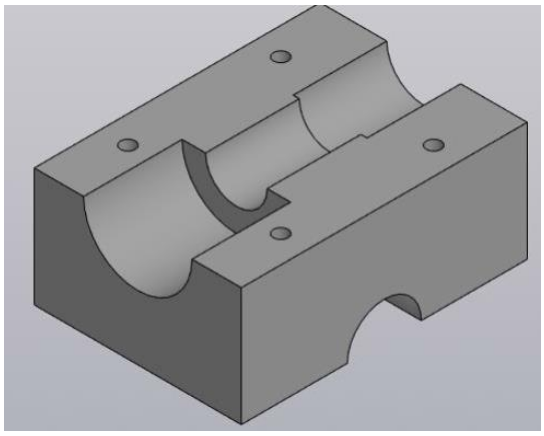




а б

Рисунок 19 – 3D модель детали, обеспечивающая перемещение аналога гематомы по вертикали. а – вид сверху, б – изометрия детали





а б

Рисунок 20 – 3D модель движущейся подставки. а – вид сверху, б – изометрия детали

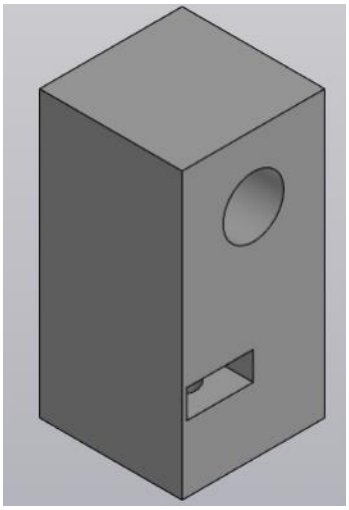


Рисунок 21 – 3D модель опорной детали для закрепления направляющих

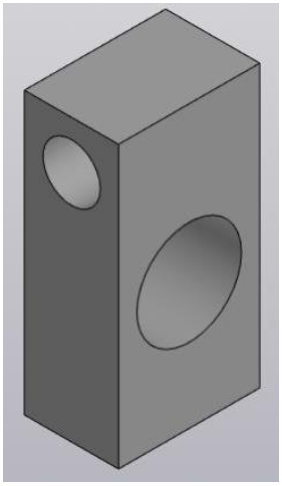


Рисунок 22 – 3D модель детали для закрепления направляющих и резьбовых шпилек

Для перемещения по горизонтали и вертикали используются шесть направляющих, диаметр которых 6 см и длина 20 см, четыре резьбовые шпильки – диаметром 8 см и длиной 15 см и соединительные крепежные конструкции с пружиной. С помощью станка (3D принтера) смоделированные детали были реализованы (см. Рисунки 23-26).

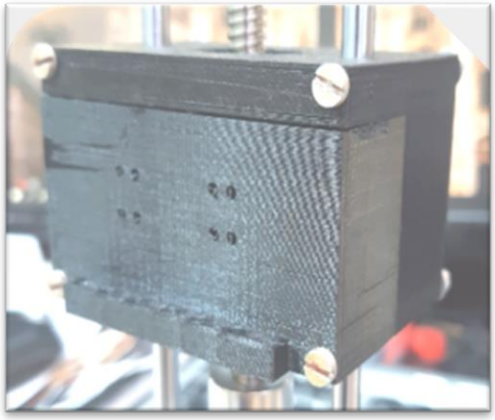


Рисунок 23– Выполненная деталь держателя колбы с кровью (аналог гематомы)

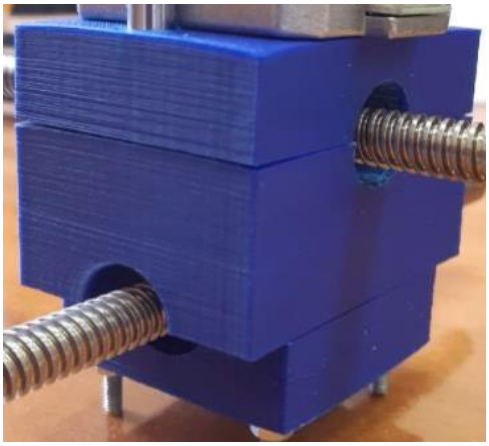


Рисунок 24– Выполненная деталь движущейся подставки

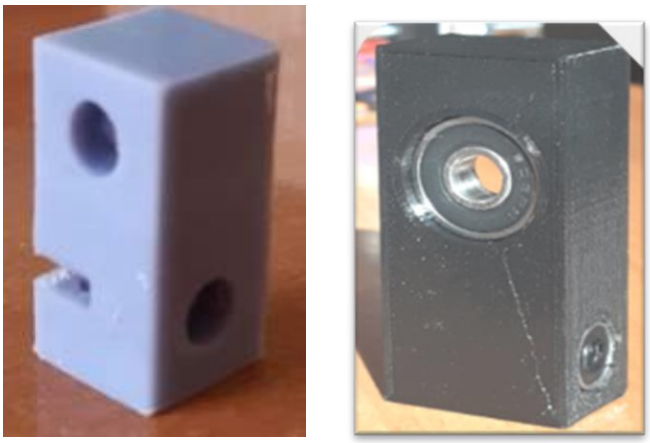


Рисунок 25– Выполненные опорные детали макета

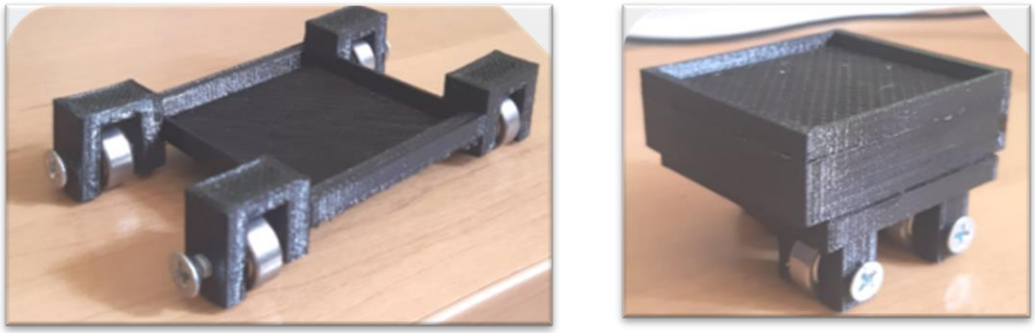


Рисунок 26– Выполненные детали опоры шаговых двигателей

В данной системе 3D позиционирования использовались гибкие муфты для компенсации нарушения соосности вала шагового двигателя и шпильки.

3.3.2 Разработка управления макетом

Для реализации перемещения конструкции был использован шаговый двигатель 17HS4401 (см. Рисунок 27). Представленный шаговый двигатель обладает следующими характеристиками [16]:

- номинальный ток: 1,7 А;
- крутящий момент: 5,5 кг×см;
- угловой шаг: 1,8±5%, 200 шагов;
- число фаз: 2;
- габариты: 42×42×48 мм;
- длина вала: 24 мм;
- диаметр вала: 5 мм.

Рисунок 27– Шаговый двигатель 17HS4401

Для управления шаговым двигателем был выбран драйвер шагового двигателя A4988 (см. Рисунок 28). Технические характеристики драйвера [17]:

- напряжения питания: 8-35 В
- режим микрошага: 1, 1/2, 1/4, 1/8, 1/16
- напряжение логики: 3-5.5 В
- защита от перегрева
- максимальный ток на фазу: - 1 А без радиатора; - 2 А с радиатором
- размер: 20 х 15 мм.

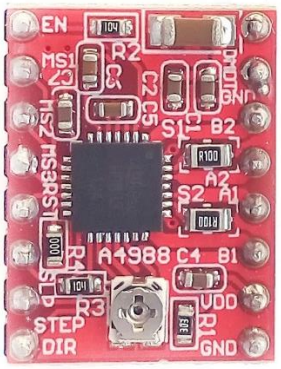


Рисунок 28– Драйвер A4988 для управления шаговыми двигателями

Подключение драйвера к шаговому двигателю и микроконтроллеру осуществлялось в соответствии с рисунком 29, описание выводов драйвера A4988, которые используются в работе, представлены в таблице 1. Для обеспечения грамотной работы шагового двигателя, необходимо подать на VMOT питание 17 В.

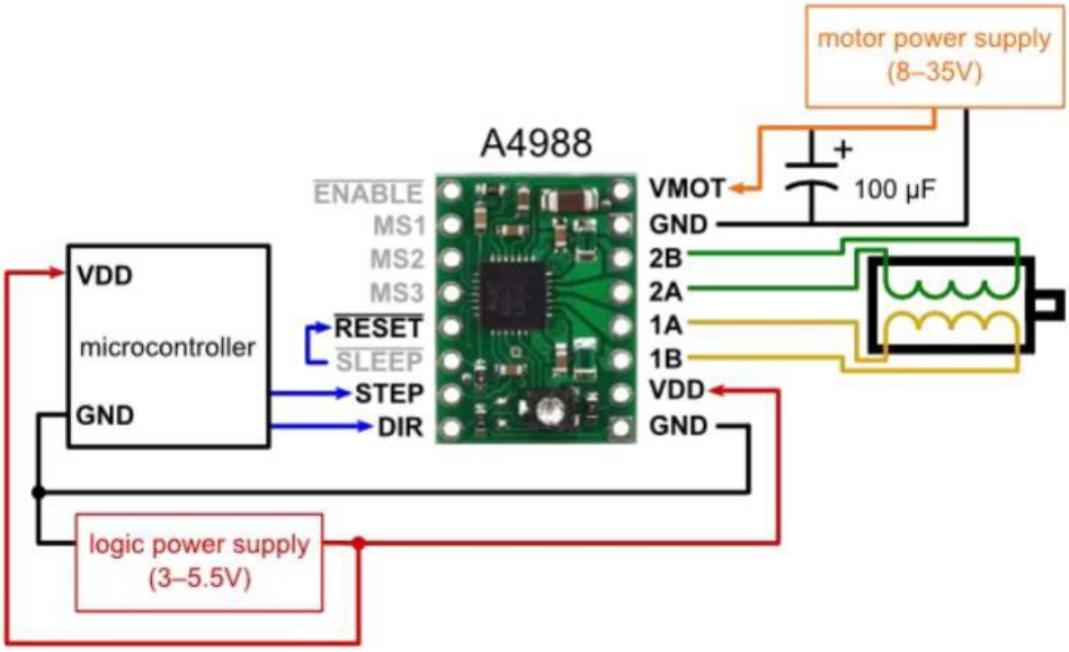


Рисунок 29– Подключение драйвера A4988

Таблица 1 – Назначение выходов драйвера A4988[18]

Контакт	Назначение
ENABLE	включение/выключение драйвера
MS1, MS2, MS3	контакты для установки микрошага
RESET	сброс микросхемы
STEP	генерация импульсов для движения двигателей
DIR (direction)	установка направление вращения
VMOT	питание для двигателя (8 – 35 В)
GND	общий (земля)
2B, 2A, 1A, 1B	для подключения обмоток двигателя
VDD	питание микросхемы (3.5 –5В)

С помощью платформы STM32CubeIDE был реализован программный код для управления шаговыми двигателями системы 3D позиционирования (Приложение А), блок схема которого представлен на рисунке 30.

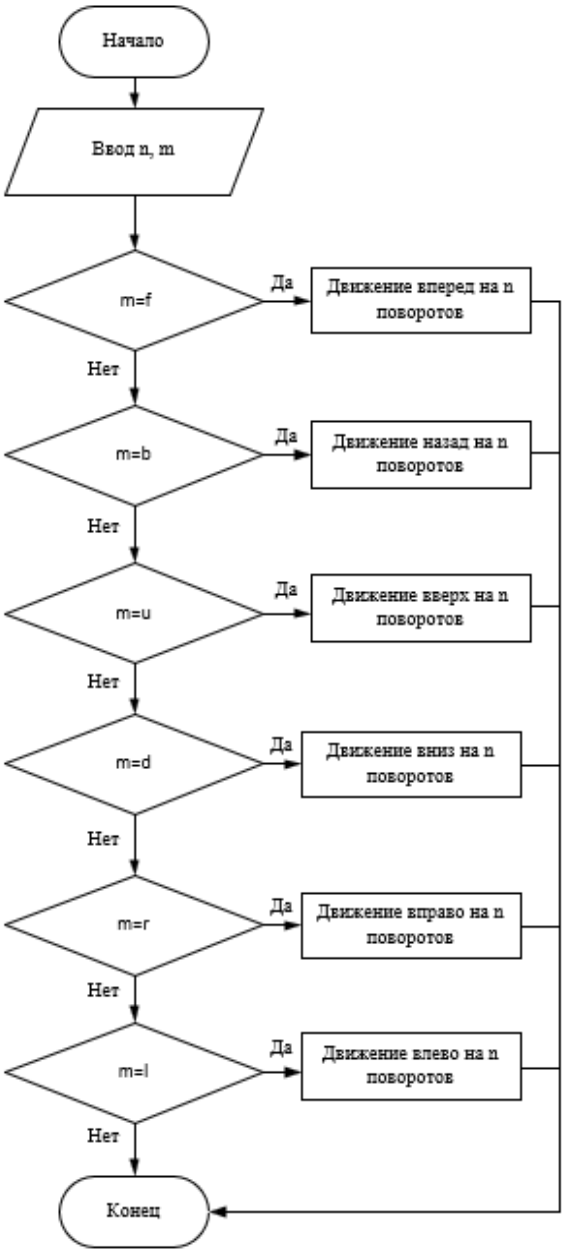


Рисунок 30– Блок-схема программы для управления шаговыми двигателями

При запуске данного программного кода, пользователь должен задать количество шагов, то есть переменную n, и направление движения – переменную m. Данные значения передаются по UART через программу Terminal на микроконтроллер. Переменная m (направление движения) может принимать следующие значения:

- «f» - forward – вперед;
- «b» - back – назад;
- «u» - up – вверх;
- «d» - down – вниз;
- «r» - right – вправо;

- l» - left – влево.

После выбора направления движения на порт ENABLE соответствующих драйверов подается значение 1, что разрешает работу нужного драйвера. Шаговый двигатель начинает движение по заданному направлению на заданное количество шагов n. Когда будет достигнута необходимая позиция аналога гематомы, установка перестает двигаться.

За начальное положение может быть принято либо центральное, относительно оси X и Y, либо один из углов установки. Для ограничения перемещения системы по осям, используются тактовые кнопки, что позволяет предотвратить перемещение аналога гематомы вне ограниченной плоскости (фантоме головы).

Реализованная система 3D позиционирования представлена на рисунке 31. Распечатанные детали были соединены, исполнительные устройства макета (шаговый двигатель и драйвер) подключены по схеме. Данная конструкция позволяет перемещать аналог гематомы в пространстве, что позволит регулировать местоположение и глубину залегания пробирки с кровью в будущих экспериментах.

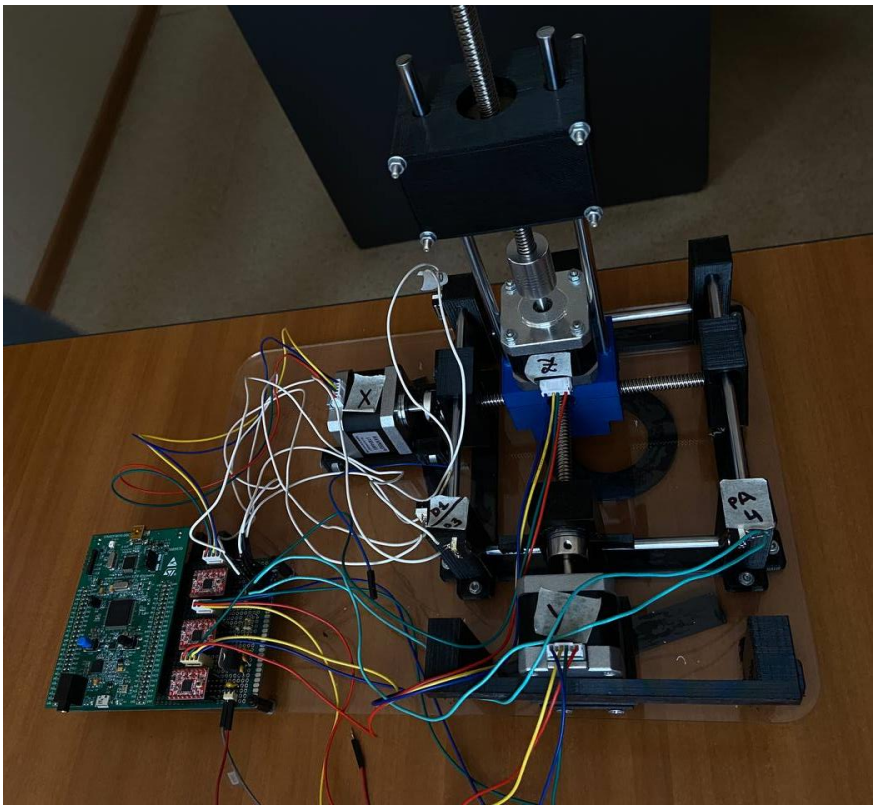


Рисунок 31– Макет системы 3D позиционирования

4 Разработка программного обеспечения

Для разработки программного обеспечения была выбрана среда разработки Visual Studio. Был реализован прототип программного обеспечения для автоматизации экспериментальных исследований. На данном этапе сделаны 3 вкладки, с помощью которых можно осуществлять управление установкой и снятие данных – вкладка «Измерения пошаговые», «График» и «Подключение».

Во вкладке «Измерения пошаговые» (см. Рисунок 32) имеются необходимые инструменты для управления системой 3D позиционирования и источником излучения.

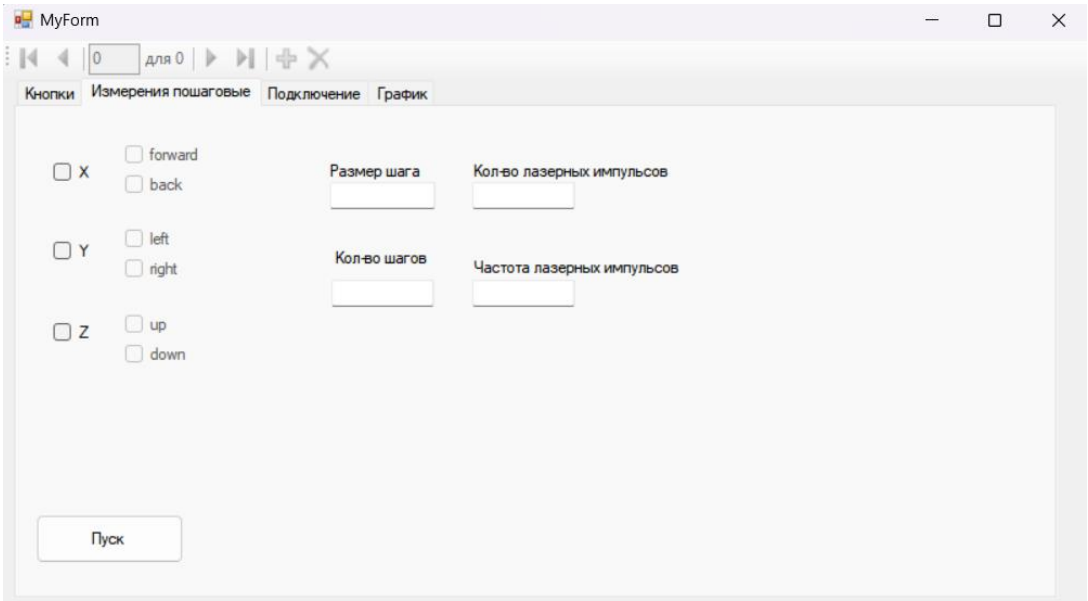


Рисунок 32 – Прототип программного обеспечения для автоматизации экспериментальных исследований, вкладка «Измерения пошаговые»

Для управления системой 3D позиционирования реализована возможность настройки размера шага и задания количества шагов, также можно выбрать необходимые оси для реализации движения конструкции механической части экспериментальной установки в нужном направлении.

В данной вкладке также имеется настройка излучателя ИК излучения, есть возможность изменения частоты подачи лазерных импульсов в заданном количестве. На данном этапе часть данных подается на микроконтроллер через COM-порт (направление движения, размер и количество шагов), а часть записывается в .txt документ при нажатии кнопки «Пуск» (см. Рисунки 33-34).

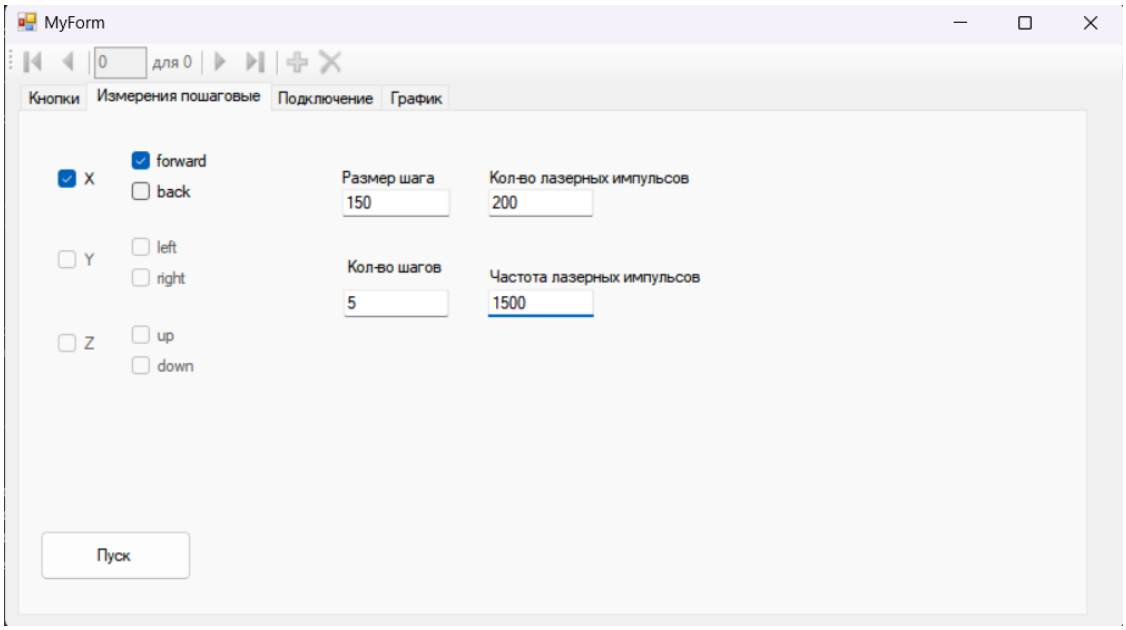


Рисунок 33– Пример работы программы

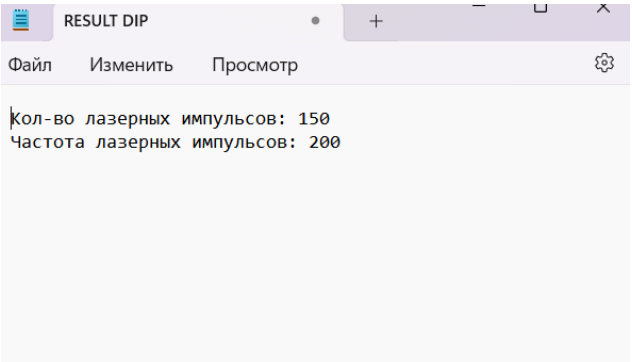


Рисунок 34 – Запись данных в файл

Во вкладке «График» (см. Рисунок 35) имеются необходимые инструменты для построения зависимости изменения сигнала от характеристик гематомы (плотность, размер, местоположение) по снятым данным с приемника. Данная вкладка позволяет пользователю построить график по выбранному .csv файлу или же по введенным данным с клавиатуры (вид графика для данных, введенных с клавиатуры, прописывается в коде, на рисунке 35 в пример представлен график косинуса).

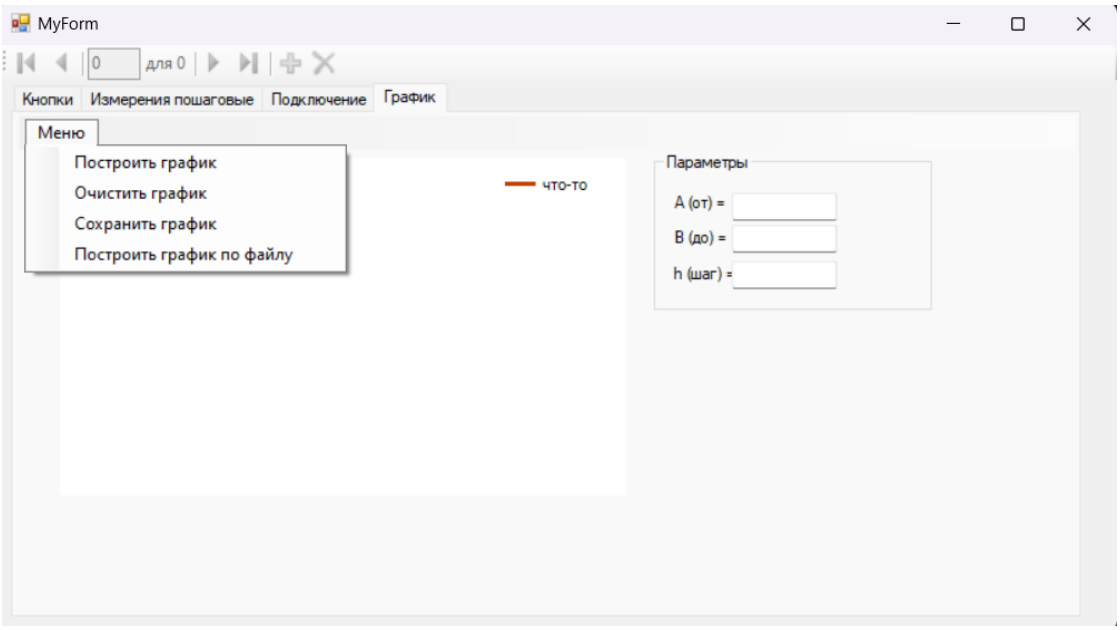


Рисунок 35 – Прототип программного обеспечения для автоматизации экспериментальных исследований, вкладка «График»

После введения необходимых данных (в частности, начала, конца и размер шага) и нажатия в меню «Построить график» (см. Рисунок 36) строится график косинуса (график был выбран для примера и прописан в коде).

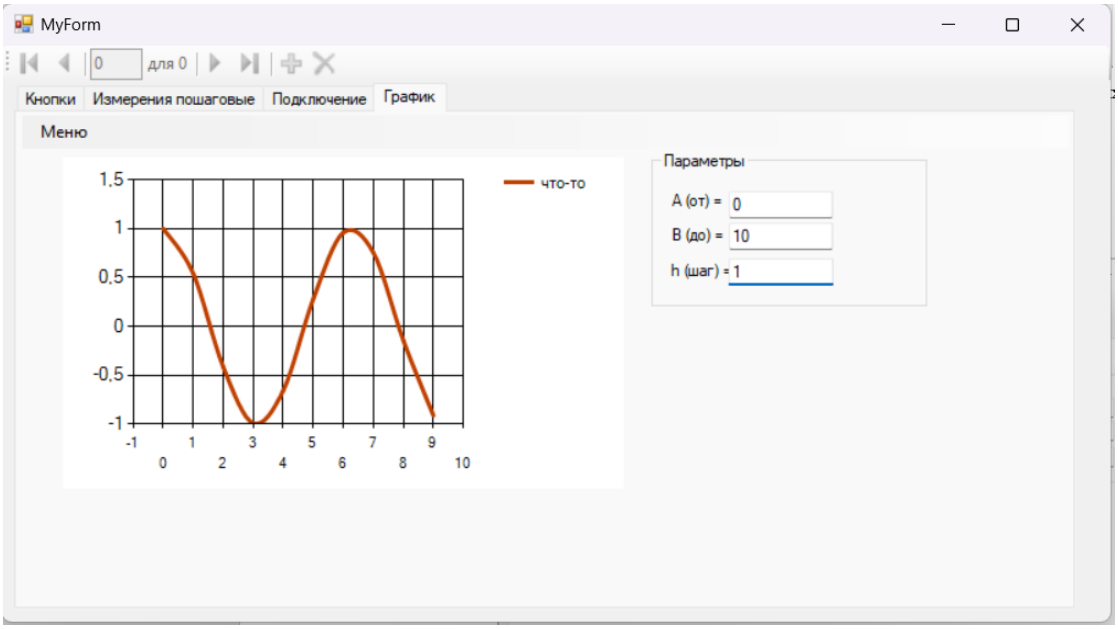


Рисунок 36 – Прототип программного обеспечения для автоматизации экспериментальных исследований, вкладка «График»

Построение графика по выбранному .csv файлу осуществляется путем выбора нужной функции в меню, в частности нужно выбрать «Построить график по файлу», а далее выбрать из имеющихся на ПК .csv файлов нужный (см. Рисунок 37 – 38).

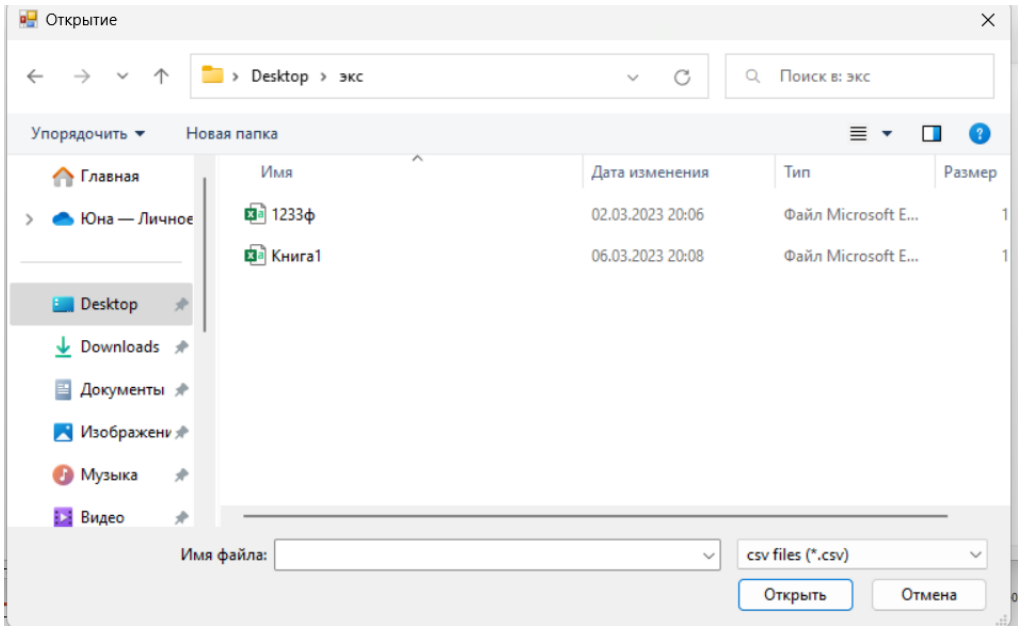


Рисунок 37 – Пример выбора файла

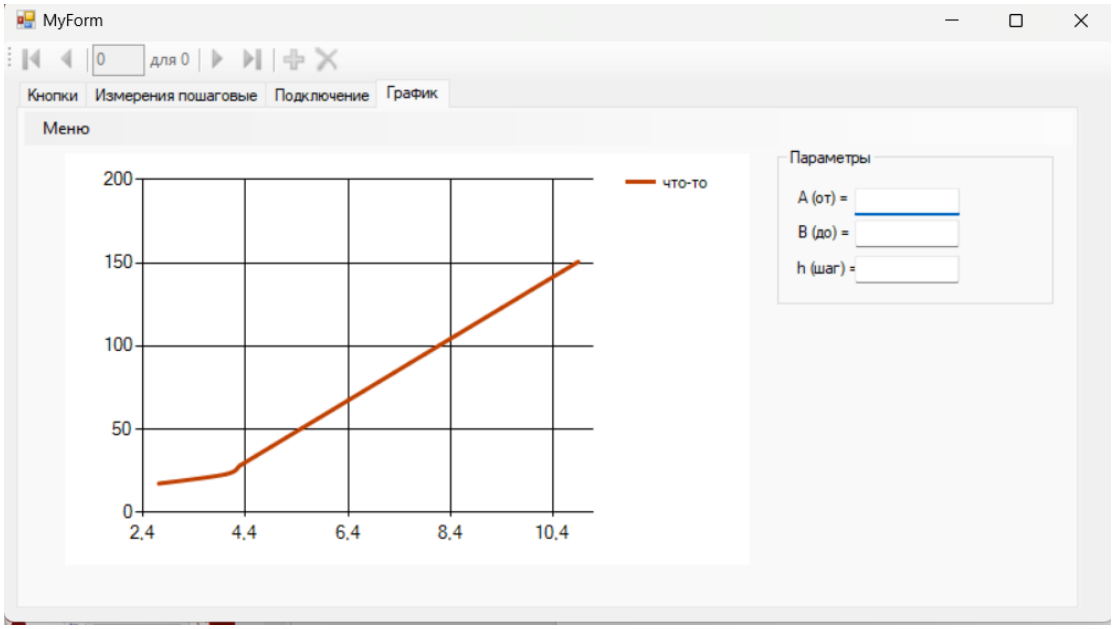


Рисунок 38 – Прототип программного обеспечения для автоматизации экспериментальных исследований, вкладка «График»: построенный график по файлу

Построенный график можно сохранить в формате jpeg для дальнейшей работы (см. Рисунок 39-40), также график можно очистить.

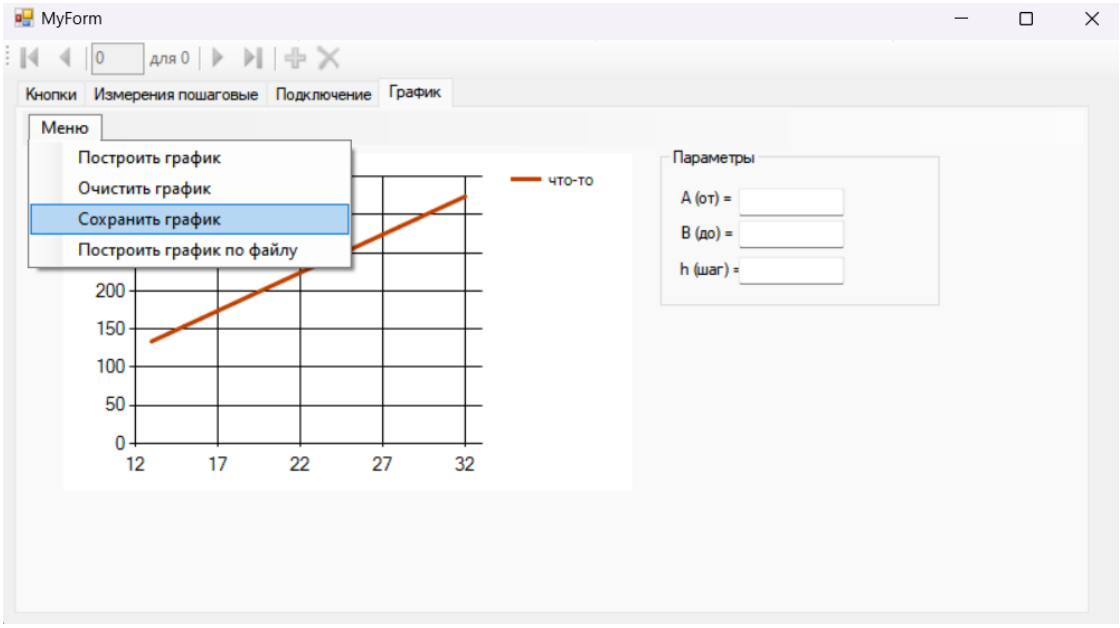


Рисунок 39 – Прототип программного обеспечения для автоматизации экспериментальных исследований, вкладка «График»: сохранение графика

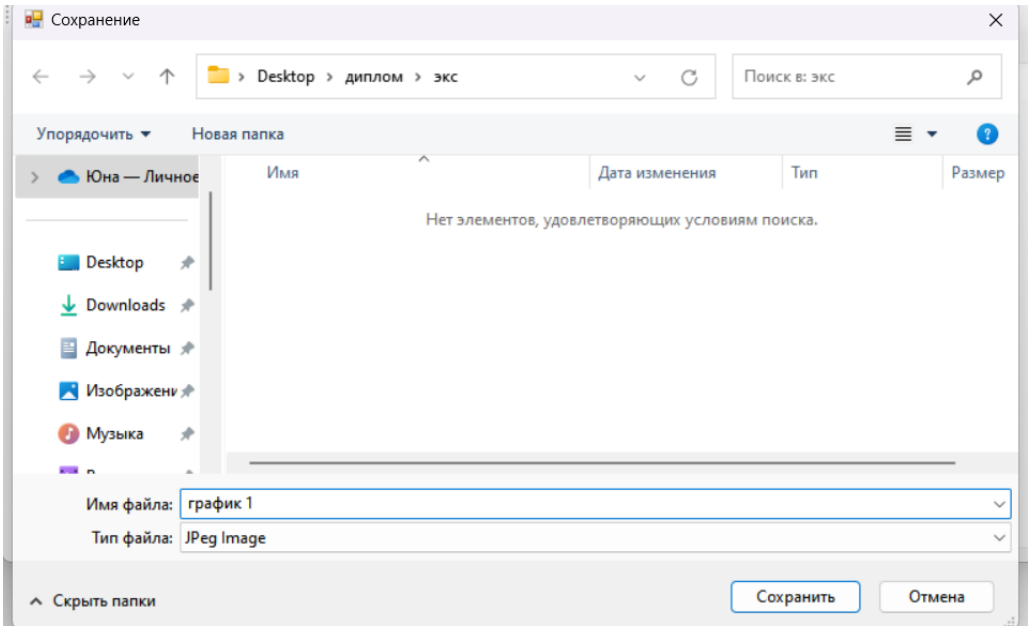


Рисунок 40 – Прототип программного обеспечения для автоматизации экспериментальных исследований, вкладка «График»: сохранение графика

По построенным графикам по результатам экспериментов можно будет сделать вывод о наличии или отсутствии гематомы в определённом месте, а также проанализировать ее характеристики.

Во вкладке «Подключение» реализована возможность выбора необходимого порта из подключенных COM-портов. Для получения списка подключенных портов нужно нажать кнопку «доступные порты» (см. Рисунок 41-42). В примере осуществлялось подключение только одного COM-порта.

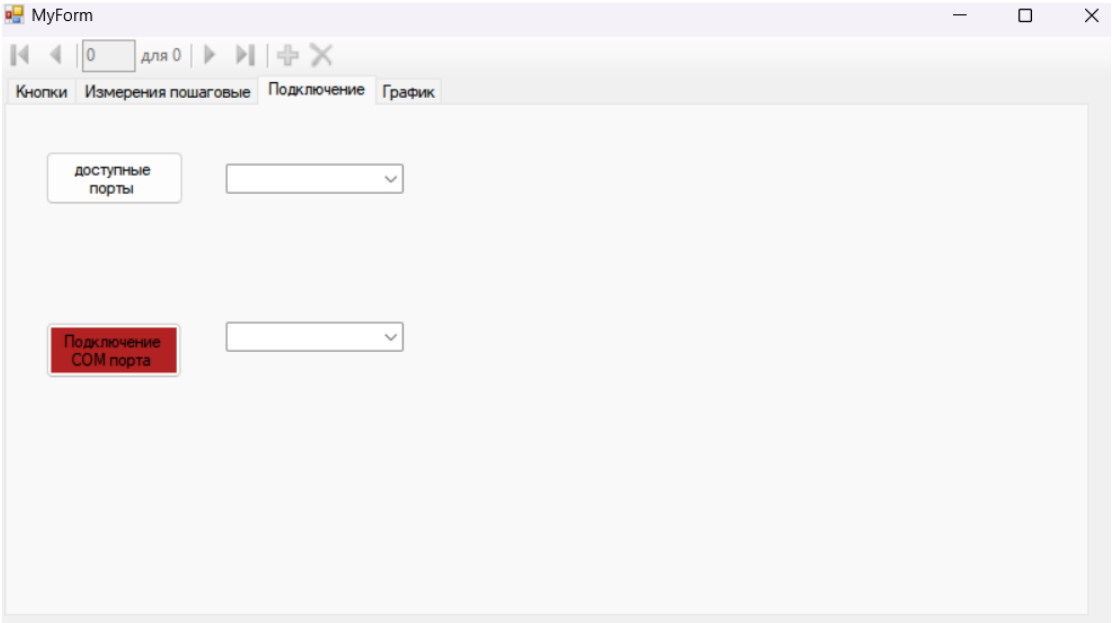


Рисунок 41 – Прототип программного обеспечения для автоматизации экспериментальных исследований, вкладка «Подключение»: без подключения COM-порта

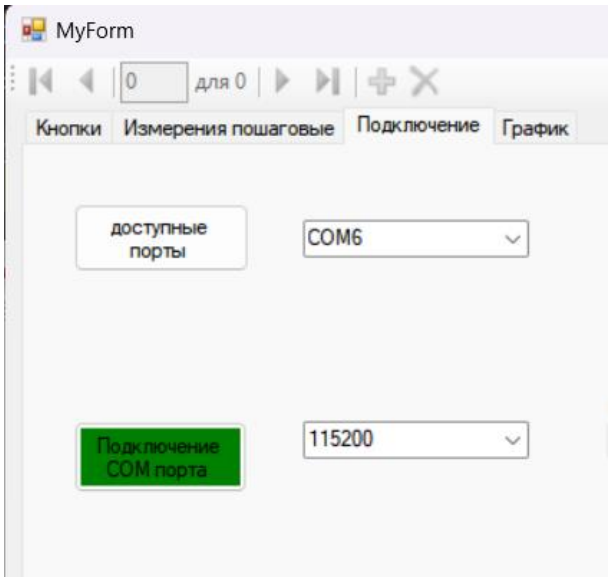


Рисунок 42 – Прототип программного обеспечения для автоматизации экспериментальных исследований, вкладка «Подключение»: с подключением COM-порта

Для настройки работы и подключения предоставлен выбор скорости (BR) передачи данных (бит/с), минимальное значение скорости передачи данных BR=300 бит/с, а максимальное – BR=115200 бит/с.

После выбора нужно скорости и нажатия кнопки «Подключение COM порта» осуществляется подключение к системе. Если подключение успешно – кнопка станет зеленого цвета, иначе – останется красной.

Для настройки и управления макетом необходимо в начале осуществлять подключение в данной вкладке.

Данное программное обеспечение разрабатывалось в среде разработки Visual Studio на языке C++ [19]. Полная программа представлена в Приложении Б.

ЗАКЛЮЧЕНИЕ

В результате выполненной работы были рассмотрены виды ЧМГ и методы их диагностики. Были изучены существующие методы диагностики выявления внутричерепных гематом, такие как КТ и МРТ, их преимущества и недостатки. Также были рассмотрены аналоги, анализ которых показал, что тип приборов для экспресс-диагностики внутричерепных гематом, работающих на методе ближнего ИК спектра, не распространён, а имеющиеся аналоги имеют ряд существенных недостатков. Также были описаны оптические свойства тканей и метод ближней инфракрасной спектроскопии, по принципу которой работает разрабатываемое устройство.

Были выбраны электронные компоненты для оптического блока. На основе выбранных элементов были спроектированы и рассчитаны принципиальные схемы источника и приемника инфракрасного излучения. Проведена трассировка и реализация печатных плат.

Также была разработана и собрана система 3D позиционирования, которая позволяет регулировать местоположение аналога гематомы в пространстве, что поможет в будущем проверить гипотезу зависимости степени поглощения ИК спектра дезоксигемоглобином от местонахождения и вида внутричерепной гематомы. На первом этапе данной работы были спроектированы и реализованы основные детали макета. Далее проведен выбор исполнительных устройств данной системы и осуществлено их подключение. Разработан программный код для управления шаговыми двигателями с помощью драйверов.

Было разработано ПО для управления макетом программно-аппаратного комплекса для проведения экспериментальных исследований, в частности реализовано: управление системой 3D позиционирования с возможностью настройки количества и размера шагов; настройка излучателя, а именно регулирование количества и частоты лазерных импульсов; построение графика (как по файлу так и вручную), а также его сохранение в качестве картинки; настройка подключения COM-порта с выбором скорости передачи данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Внутричерепные гематомы [Электронный ресурс] / Справочник MSD. – URL: https://www.msdmanuals.com/ru/дома/травмы-и-отравления/травмы-головы/внутричерепные-гематомы (дата обращения 13.11.2021).
2. Morgan Clinical Evaluation of a Portable Near-Infrared Device for Detection of Traumatic Intracranial Hematomas / 2. C.S. Robertson, E.L. Zager, R.K. Narayan [et al.] // Journal of Neurotrauma. – 2010. – Vol. 27, is. 9. – P. – 1597-1604.
3. Near Infrared Technology [Электронный ресурс] / Infrascan. – URL: https://infrascanner.com/models/ (дата обращения 23. 03. 2022).
4. Микроволновый шлем для оценки черепно-мозговой травмы [Электронный ресурс] / MedBe. – URL: https://medbe.ru/news/golovnoy-i-spinnoy-mozg/mikrovolnovyy-shlem-dlya-otsenki-cherepno-mozgovoy-travmy/ (дата обращения 23.03.2022).
5. The Infrascanner, a handheld device for screening in situ for the presence of brain haematomas [Электронный ресурс] / Brain Injury. – URL: www.tandfonline.com/doi/full/10.3109/02699052.2010.506636 (дата обращения 23. 03. 2022).
6. Оптические свойства ткани [Электронный ресурс] / StudFiles. – URL: https://studfile.net/preview/9661365/page:6/ (дата обращения 23. 03. 2022).
7. Оптический практикум по физике живых систем [Электронный ресурс] / Elibrary. – URL: http://www.elibrary.sgu.ru/uch_lit/940.pdf (дата обращения 18.02.2022)
8. Юрченко Е.А. Разработка системы для диагностики гематом мозга / Юрченко Е.А. // Двадцать седьмая Всероссийская научная конференция студентов-физиков и молодых учёных – 2023. – Т. 1. – С. 178-179.
9. Микроконтроллер STM32F407VGT6 [Электронный ресурс] / chipdip. – URL: https://www.chipdip.ru/product/stm32f407vgt6 (дата обращения 18.02.2022)
10. Лазерный диод L808H1 datasheet [Электронный ресурс] / THORLABS. – URL: https://azimp.ru/thorlabs/nir-laser-diodes/82836/ (дата обращения 23. 03. 2022).
11. Операционный усилитель TL071C datasheet [Электронный ресурс] / chipdip. – URL: https://static.chipdip.ru/lib/513/DOC010513647.pdf (дата обращения 23. 03. 2022).
12. Фотодиод кремниевый – ФД – 256 [Электронный ресурс] / chipdip. – URL: www.static.chipdip.ru/lib/287/DOC000287862.pdf (дата обращения 23. 03. 2022).

13. Операционный усилитель TL072C datasheet [Электронный ресурс] / chipdip. – URL: https://static.chipdip.ru/lib/513/DOC010513647.pdf (дата обращения 23. 03. 2022).
14. Юрченко Е.А. РАЗРАБОТКА ЭКСПЕРИМЕНТАЛЬНОЙ УСТАНОВКИ ДЛЯ ИССЛЕДОВАНИЯ ИК МЕТОДА ДИАГНОСТИКИ ГЕМАТОМ МОЗГА/ Юрченко Е.А. // СБОРНИК МАТЕРИАЛОВ XVIII Международной школы конференции студентов, аспирантов и молодых ученых – 2022. – С. 113-114.
15. Юрченко Е.А. УСТРОЙСТВО АВТОМАТИЗАЦИИ ПРОЦЕССА ИССЛЕДОВАНИЯ МЕТОДА ИК-СПЕКТРОСКОПИИ ДЛЯ ДИАГНОСТИКИ ВНКТРЕЧЕРЕПНЫХ ГЕМАТОМ/ Юрченко Е.А. // НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ИССЛЕДОВАНИИ СЛОЖНЫХ СТРУКТУР МАТЕРИАЛЫ ЧЕТЫРНАДЦАТОЙ МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ – 2022. – С. 34-35.
16. Шаговый двигатель 17HS4401 [Электронный ресурс] / Gistroy. – URL: https://gistroy.ru/wiki-parts/17hs4401/ (дата обращения 02.02.2022)
17. Драйвер шагового двигателя A4988 [Электронный ресурс] / 3Diy. – URL: https://3d-diy.ru/wiki/arduino-moduli/drajver-shagovogo-dvigatelya-a4988/ (дата обращения 12.03.2022)
18. Обзор драйвера шагового двигателя A4988 [Электронный ресурс] / robotchip. – URL: https://robotchip.ru/obzor-drayvera-shagovogo-dvigatelya-a4988/ (дата обращения 12.03.2022)
19. System.Windows.Forms [Электронный ресурс] / learn.microsoft. – URL: https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.forms?view=windowsdesktop-7.0 (дата обращения 12.10.2022)

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ ДЛЯ УПРАВЛЕНИЯ ДВИГАТЕЛЯМИ МАКЕТА

```
#include "stm32f4xx.h"
#include "stm32f4xx_rcc.h"
#include "stm32f4xx_gpio.h"
#include "stm32f4xx_adc.h"
#include "stm32f4xx_tim.h"
int v;
GPIO_InitTypeDef GPIO_InitStructure; //объявление структуры настройки портов
//основная программа
void DELAY(int c) {
 int x=0;
 while (x<c-1) {
 x++;
 }
}
void Step_And_Directiont(int n, char m) {
  if (m=='f') {
   GPIO_ResetBits(GPIOA, GPIO_Pin_1);
   GPIO_SetBits(GPIOB, GPIO_Pin_2);
   GPIO_SetBits(GPIOB, GPIO_Pin_1);
   for(int a=0; a<n-1;a++)  {
    GPIO_SetBits(GPIOB, GPIO_Pin_1);
    DELAY(1000);
    GPIO_ResetBits(GPIOB, GPIO_Pin_1);
    DELAY(2000);
   }
   GPIO_SetBits(GPIOA, GPIO_Pin_1);
  }
  else if (m=='b') {
   GPIO_ResetBits(GPIOA, GPIO_Pin_1);
   GPIO_ResetBits(GPIOB, GPIO_Pin_2);
   GPIO_SetBits(GPIOB, GPIO_Pin_1);
   for(int a=0; a<n-1;a++)  {
    GPIO_SetBits(GPIOB, GPIO_Pin_1);
    DELAY(1000);
    GPIO_ResetBits(GPIOB, GPIO_Pin_1);
    DELAY(2000);
   }
   GPIO_SetBits(GPIOA, GPIO_Pin_1);
  }
 }
int main(void) {
 RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA|RCC_AHB1Periph_GPIOB, ENABLE);
 GPIO_InitTypeDef GPIO_InitStructure;
 RCC_AHB1PeriphClockCmd(RCC_AHB1ENR_GPIOHEN, ENABLE);//включение тактирования
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1; //выбор настраиваемых выводов
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT; //режим – аналоговый(для АЦП)
 GPIO_InitStructure.GPIO_OType = GPIO_OType_PP; //тип – push-pull
 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz; //выбор максимальной частоты
 GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL; // без подтягивающего резистора
 GPIO_ResetBits(GPIOA, GPIO_Pin_1);
```

```
 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN; // выбор режима - вход
 GPIO_InitStructure.GPIO_OType = GPIO_OType_PP; // тип - push-pull
 GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN; //подтяжка к земле
 GPIO_Init(GPIOA, &GPIO_InitStructure); //заполнение объявленной структуры
 GPIO_ResetBits(GPIOA, GPIO_Pin_0);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 | GPIO_Pin_2; //выбор настраиваемых выводов
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT; //режим – вывод
 GPIO_InitStructure.GPIO_OType = GPIO_OType_PP; //тип – push-pull
 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz; //выбор максимальной частоты
 GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL; // без подтягивающего резистора
 GPIO_Init(GPIOB, &GPIO_InitStructure); //заполнение объявленной структуры
 GPIO_ResetBits(GPIOB, GPIO_Pin_1);
 GPIO_ResetBits(GPIOB, GPIO_Pin_2);
 while(1) {
  v=GPIO_ReadInputDataBit(GPIOA,GPIO_Pin_0);
  if (v==1) {
   Step_And_Directiont(300, 'f');
  }
  else  {
   GPIO_ResetBits(GPIOA, GPIO_Pin_0);
  }
 }
}
```

ПРИЛОЖЕНИЕ Б

ПРОГРАММА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ ПРОГРАММНО-АППАРАТНЫМ КОМПЛЕКСОМ

```
#pragma once
#include <fstream>
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <string>
#include <vector>
#include <sstream>
#include "string"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <msclr\marshal_cppstd.h>
#include <msclr/marshal.h>
#include<map>
#include <algorithm>
#using <System.DLL>
#using <System.Drawing.DLL>
#using <System.Windows.Forms.DLL>
namespace Project1 {
	using namespace std;
	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace msclr::interop;
	using namespace System::Drawing::Imaging;
	using namespace System::IO::Ports;
	int set_flag;
	public ref class MyForm : public System::Windows::Forms::Form{
	public:
		MyForm(void){
			InitializeComponent();
			if (set_flag == 1){
				bool flag = true;
				String^ recive_data;
				while (flag){
					try{
						recive_data = serialPort1->ReadLine();
						flag = false;
					}
					catch (TimeoutException^) {}
				}
				textBox8->Text = recive_data;
			}
		}
	protected:
		~MyForm(){
```

```
                if (components){
                        delete components;
                        serialPort1->Close();
                }
        }
        private: System::Windows::Forms::Button^ coutport;
        protected:
        private: System::Windows::Forms::BindingNavigator^ bindingNavigator1;
        private: System::Windows::Forms::ToolStripButton^ bindingNavigatorAddNewItem;
        private: System::Windows::Forms::ToolStripLabel^ bindingNavigatorCountItem;
        private: System::Windows::Forms::ToolStripButton^ bindingNavigatorDeleteItem;
        private: System::Windows::Forms::ToolStripButton^ bindingNavigatorMoveFirstItem;
        private:                                        System::Windows::Forms::ToolStripButton^
bindingNavigatorMovePreviousItem;
        private: System::Windows::Forms::ToolStripSeparator^ bindingNavigatorSeparator;
        private: System::Windows::Forms::ToolStripTextBox^ bindingNavigatorPositionItem;
        private: System::Windows::Forms::ToolStripSeparator^ bindingNavigatorSeparator1;
        private: System::Windows::Forms::ToolStripButton^ bindingNavigatorMoveNextItem;
        private: System::Windows::Forms::ToolStripButton^ bindingNavigatorMoveLastItem;
        private: System::Windows::Forms::ToolStripSeparator^ bindingNavigatorSeparator2;
        private: System::Windows::Forms::TabPage^ tabPage4;
        private: System::Windows::Forms::GroupBox^ groupBox1;
        private: System::Windows::Forms::TextBox^ textBox_h;
        private: System::Windows::Forms::TextBox^ textBox_b;
        private: System::Windows::Forms::TextBox^ textBox_a;
        private: System::Windows::Forms::Label^ label6;
        private: System::Windows::Forms::Label^ label5;
        private: System::Windows::Forms::Label^ label4;
        private: System::Windows::Forms::DataVisualization::Charting::Chart^ chart;
        private: System::Windows::Forms::MenuStrip^ menuStrip1;
        private: System::Windows::Forms::ToolStripMenuItem^ меню ToolStripMenuItem;
        private:                                System::Windows::Forms::ToolStripMenuItem^
построитьГрафикToolStripMenuItem;
        private:                                System::Windows::Forms::ToolStripMenuItem^
очиститьГрафикToolStripMenuItem;
        private:                                System::Windows::Forms::ToolStripMenuItem^
сохранитьГрафикToolStripMenuItem;
        private: System::Windows::Forms::TabPage^ tabPage3;
        private: System::Windows::Forms::TabPage^ tabPage2;
        private: System::Windows::Forms::CheckBox^ checkBox3;
        private: System::Windows::Forms::CheckBox^ checkBox2;
        private: System::Windows::Forms::CheckBox^ checkBox1;
        private: System::Windows::Forms::Label^ label3;
        private: System::Windows::Forms::Label^ label2;
        private: System::Windows::Forms::Label^ label1;
        private: System::Windows::Forms::TextBox^ textBox5;
        private: System::Windows::Forms::TextBox^ textBox4;
        private: System::Windows::Forms::TextBox^ textBox3;
        private: System::Windows::Forms::Button^ button5;
        private: System::Windows::Forms::TabPage^ tabPage1;
        private: System::Windows::Forms::TextBox^ textBox1;
        private: System::Windows::Forms::TextBox^ textBox2;
        private: System::Windows::Forms::Button^ button1;
        public protected: System::Windows::Forms::TabControl^ tabPage;
        private: System::Windows::Forms::Label^ label7;
        public protected:
```

```
private: System::Windows::Forms::TextBox^ textBox6;
private: System::Windows::Forms::OpenFileDialog^ openFileDialog1;
private: System::Windows::Forms::SaveFileDialog^ saveFileDialog1;
private: System::Windows::Forms::ToolStripMenuItem^
построитьГрафикПоФайлуToolStripMenuItem;
private: System::Windows::Forms::FolderBrowserDialog^ folderBrowserDialog1;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::IO::Ports::SerialPort^ serialPort1;
private: System::Windows::Forms::Button^ applycomport;
private: System::Windows::Forms::ComboBox^ comboBox2;
private: System::Windows::Forms::TextBox^ textBox7;
private: System::Windows::Forms::Label^ label8;
private: System::Windows::Forms::TextBox^ textBox8;
private: System::Windows::Forms::TextBox^ textBox9;
private: System::Windows::Forms::Button^ button4;
private:
private:
public protected:
private:
private:
public protected:
private:
private:
private: System::ComponentModel::IContainer^ components;
protected:
private:
#pragma region Windows Form Designer generated code
void InitializeComponent(void){
this->components = (gcnew System::ComponentModel::Container());
System::ComponentModel::ComponentResourceManager^ resources =
(gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));
System::Windows::Forms::DataVisualization::Charting::ChartArea^
chartArea1 = (gcnew System::Windows::Forms::DataVisualization::Charting::ChartArea());
System::Windows::Forms::DataVisualization::Charting::Legend^
legend1 = (gcnew System::Windows::Forms::DataVisualization::Charting::Legend());
System::Windows::Forms::DataVisualization::Charting::Series^ series1
= (gcnew System::Windows::Forms::DataVisualization::Charting::Series());
this->bindingNavigator1 = (gcnew
System::Windows::Forms::BindingNavigator(this->components));
this->bindingNavigatorAddNewItem = (gcnew
System::Windows::Forms::ToolStripButton());
this->bindingNavigatorCountItem = (gcnew
System::Windows::Forms::ToolStripLabel());
this->bindingNavigatorDeleteItem = (gcnew
System::Windows::Forms::ToolStripButton());
this->bindingNavigatorMoveFirstItem = (gcnew
System::Windows::Forms::ToolStripButton());
this->bindingNavigatorMovePreviousItem = (gcnew
System::Windows::Forms::ToolStripButton());
this->bindingNavigatorSeparator = (gcnew
System::Windows::Forms::ToolStripSeparator());
this->bindingNavigatorPositionItem = (gcnew
System::Windows::Forms::ToolStripTextBox());
this->bindingNavigatorSeparator1 = (gcnew
System::Windows::Forms::ToolStripSeparator());
```

```
this->bindingNavigatorMoveNextItem = (gcnew System::Windows::Forms::ToolStripButton());
this->bindingNavigatorMoveLastItem = (gcnew System::Windows::Forms::ToolStripButton());
this->bindingNavigatorSeparator2 = (gcnew System::Windows::Forms::ToolStripSeparator());
this->tabPage4 = (gcnew System::Windows::Forms::TabPage());
this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
this->textBox_h = (gcnew System::Windows::Forms::TextBox());
this->textBox_b = (gcnew System::Windows::Forms::TextBox());
this->textBox_a = (gcnew System::Windows::Forms::TextBox());
this->label6 = (gcnew System::Windows::Forms::Label());
this->label5 = (gcnew System::Windows::Forms::Label());
this->label4 = (gcnew System::Windows::Forms::Label());
this->chart = (gcnew System::Windows::Forms::DataVisualization::Charting::Chart());
this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
this->меню ToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
this->построитьГрафикToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
this->очиститьГрафикToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
this->сохранитьГрафикToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
this->построитьГрафикПоФайлуToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
this->tabPage3 = (gcnew System::Windows::Forms::TabPage());
this->textBox9 = (gcnew System::Windows::Forms::TextBox());
this->button4 = (gcnew System::Windows::Forms::Button());
this->textBox8 = (gcnew System::Windows::Forms::TextBox());
this->label8 = (gcnew System::Windows::Forms::Label());
this->textBox7 = (gcnew System::Windows::Forms::TextBox());
this->comboBox2 = (gcnew System::Windows::Forms::ComboBox());
this->applycomport = (gcnew System::Windows::Forms::Button());
this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
this->coutport = (gcnew System::Windows::Forms::Button());
this->tabPage2 = (gcnew System::Windows::Forms::TabPage());
this->label7 = (gcnew System::Windows::Forms::Label());
this->textBox6 = (gcnew System::Windows::Forms::TextBox());
this->checkBox3 = (gcnew System::Windows::Forms::CheckBox());
this->checkBox2 = (gcnew System::Windows::Forms::CheckBox());
this->checkBox1 = (gcnew System::Windows::Forms::CheckBox());
this->label3 = (gcnew System::Windows::Forms::Label());
this->label2 = (gcnew System::Windows::Forms::Label());
this->label1 = (gcnew System::Windows::Forms::Label());
this->textBox5 = (gcnew System::Windows::Forms::TextBox());
this->textBox4 = (gcnew System::Windows::Forms::TextBox());
this->textBox3 = (gcnew System::Windows::Forms::TextBox());
this->button5 = (gcnew System::Windows::Forms::Button());
this->tabPage1 = (gcnew System::Windows::Forms::TabPage());
this->textBox1 = (gcnew System::Windows::Forms::TextBox());
this->textBox2 = (gcnew System::Windows::Forms::TextBox());
this->button1 = (gcnew System::Windows::Forms::Button());
this->tabPage = (gcnew System::Windows::Forms::TabControl());
```

```
this->openFileDialog1 = (gcnew System::Windows::Forms::OpenFileDialog());
this->saveFileDialog1 = (gcnew System::Windows::Forms::SaveFileDialog());
this->folderBrowserDialog1 = (gcnew System::Windows::Forms::FolderBrowserDialog());
this->serialPort1 = (gcnew System::IO::Ports::SerialPort(this->components));
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this->bindingNavigator1))->BeginInit();
this->bindingNavigator1->SuspendLayout();
this->tabPage4->SuspendLayout();
this->groupBox1->SuspendLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this->chart))->BeginInit();
this->menuStrip1->SuspendLayout();
this->tabPage3->SuspendLayout();
this->tabPage2->SuspendLayout();
this->tabPage1->SuspendLayout();
this->tabPage->SuspendLayout();
this->SuspendLayout();
this->bindingNavigator1->AddNewItem = this->bindingNavigatorAddNewItem;
this->bindingNavigator1->CountItem = this->bindingNavigatorCountItem;
this->bindingNavigator1->DeleteItem = this->bindingNavigatorDeleteItem;
this->bindingNavigator1->ImageScalingSize = System::Drawing::Size(20, 20);
this->bindingNavigator1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(11) {
    this->bindingNavigatorMoveFirstItem,
    this->bindingNavigatorMovePreviousItem, this->bindingNavigatorSeparator, this->bindingNavigatorPositionItem, this->bindingNavigatorCountItem,
    this->bindingNavigatorSeparator1, this->bindingNavigatorMoveNextItem, this->bindingNavigatorMoveLastItem, this->bindingNavigatorSeparator2,
    this->bindingNavigatorAddNewItem, this->bindingNavigatorDeleteItem
});
this->bindingNavigator1->Location = System::Drawing::Point(0, 0);
this->bindingNavigator1->MoveFirstItem = this->bindingNavigatorMoveFirstItem;
this->bindingNavigator1->MoveLastItem = this->bindingNavigatorMoveLastItem;
this->bindingNavigator1->MoveNextItem = this->bindingNavigatorMoveNextItem;
this->bindingNavigator1->MovePreviousItem = this->bindingNavigatorMovePreviousItem;
this->bindingNavigator1->Name = L"bindingNavigator1";
this->bindingNavigator1->PositionItem = this->bindingNavigatorPositionItem;
this->bindingNavigator1->Size = System::Drawing::Size(1072, 27);
this->bindingNavigator1->TabIndex = 3;
this->bindingNavigator1->Text = L"bindingNavigator1";
```

```
this->bindingNavigatorAddNewItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;
this->bindingNavigatorAddNewItem->Image = (cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorAddNewItem.Image")));
this->bindingNavigatorAddNewItem->Name = L"bindingNavigatorAddNewItem";
this->bindingNavigatorAddNewItem->RightToLeftAutoMirrorImage = true;
this->bindingNavigatorAddNewItem->Size = System::Drawing::Size(29, 24);
this->bindingNavigatorAddNewItem->Text = L"Добавить";
this->bindingNavigatorCountItem->Name = L"bindingNavigatorCountItem";
this->bindingNavigatorCountItem->Size = System::Drawing::Size(55, 24);
this->bindingNavigatorCountItem->Text = L"для {0}";
this->bindingNavigatorCountItem->ToolTipText = L"Общее число элементов";
this->bindingNavigatorDeleteItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;
this->bindingNavigatorDeleteItem->Image = (cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorDeleteItem.Image")));
this->bindingNavigatorDeleteItem->Name = L"bindingNavigatorDeleteItem";
this->bindingNavigatorDeleteItem->RightToLeftAutoMirrorImage = true;
this->bindingNavigatorDeleteItem->Size = System::Drawing::Size(29, 24);
this->bindingNavigatorDeleteItem->Text = L"Удалить";
this->bindingNavigatorMoveFirstItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;
this->bindingNavigatorMoveFirstItem->Image = (cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorMoveFirstItem.Image")));
this->bindingNavigatorMoveFirstItem->Name = L"bindingNavigatorMoveFirstItem";
this->bindingNavigatorMoveFirstItem->RightToLeftAutoMirrorImage = true;
this->bindingNavigatorMoveFirstItem->Size = System::Drawing::Size(29, 24);
this->bindingNavigatorMoveFirstItem->Text = L"Переместить в начало";
this->bindingNavigatorMovePreviousItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;
this->bindingNavigatorMovePreviousItem->Image = (cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorMovePreviousItem.Image")));
this->bindingNavigatorMovePreviousItem->Name = L"bindingNavigatorMovePreviousItem";
this->bindingNavigatorMovePreviousItem->RightToLeftAutoMirrorImage = true;
this->bindingNavigatorMovePreviousItem->Size = System::Drawing::Size(29, 24);
```

```
this->bindingNavigatorMovePreviousItem->Text = L"Переместить назад";
this->bindingNavigatorSeparator->Name = L"bindingNavigatorSeparator";
this->bindingNavigatorSeparator->Size = System::Drawing::Size(6, 27);
this->bindingNavigatorPositionItem->AccessibleName = L"Положение";
this->bindingNavigatorPositionItem->AutoSize = false;
this->bindingNavigatorPositionItem->Font = (gcnew System::Drawing::Font(L"Segoe UI", 9));
this->bindingNavigatorPositionItem->Name = L"bindingNavigatorPositionItem";
this->bindingNavigatorPositionItem->Size = System::Drawing::Size(50, 27);
this->bindingNavigatorPositionItem->Text = L"0";
this->bindingNavigatorPositionItem->ToolTipText = L"Текущее положение";
this->bindingNavigatorSeparator1->Name = L"bindingNavigatorSeparator1";
this->bindingNavigatorSeparator1->Size = System::Drawing::Size(6, 27);
this->bindingNavigatorMoveNextItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;
this->bindingNavigatorMoveNextItem->Image = (cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorMoveNextItem.Image")));
this->bindingNavigatorMoveNextItem->Name = L"bindingNavigatorMoveNextItem";
this->bindingNavigatorMoveNextItem->RightToLeftAutoMirrorImage = true;
this->bindingNavigatorMoveNextItem->Size = System::Drawing::Size(29, 24);
this->bindingNavigatorMoveNextItem->Text = L"Переместить вперед";
this->bindingNavigatorMoveLastItem->DisplayStyle = System::Windows::Forms::ToolStripItemDisplayStyle::Image;
this->bindingNavigatorMoveLastItem->Image = (cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"bindingNavigatorMoveLastItem.Image")));
this->bindingNavigatorMoveLastItem->Name = L"bindingNavigatorMoveLastItem";
this->bindingNavigatorMoveLastItem->RightToLeftAutoMirrorImage = true;
this->bindingNavigatorMoveLastItem->Size = System::Drawing::Size(29, 24);
this->bindingNavigatorMoveLastItem->Text = L"Переместить в конец";
this->bindingNavigatorSeparator2->Name = L"bindingNavigatorSeparator2";
this->bindingNavigatorSeparator2->Size = System::Drawing::Size(6, 27);
this->tabPage4->Controls->Add(this->groupBox1);
this->tabPage4->Controls->Add(this->chart);
this->tabPage4->Controls->Add(this->menuStrip1);
this->tabPage4->Location = System::Drawing::Point(4, 25);
this->tabPage4->Name = L"tabPage4";
```

```
this->tabPage4->Padding = System::Windows::Forms::Padding(3);
this->tabPage4->Size = System::Drawing::Size(1020, 443);
this->tabPage4->TabIndex = 3;
this->tabPage4->Text = L"График";
this->tabPage4->UseVisualStyleBackColor = true;
this->groupBox1->Anchor =
static_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Top |
System::Windows::Forms::AnchorStyles::Right));
this->groupBox1->Controls->Add(this->textBox_h);
this->groupBox1->Controls->Add(this->textBox_b);
this->groupBox1->Controls->Add(this->textBox_a);
this->groupBox1->Controls->Add(this->label6);
this->groupBox1->Controls->Add(this->label5);
this->groupBox1->Controls->Add(this->label4);
this->groupBox1->Location = System::Drawing::Point(614, 34);
this->groupBox1->Name = L"groupBox1";
this->groupBox1->Size = System::Drawing::Size(266, 140);
this->groupBox1->TabIndex = 2;
this->groupBox1->TabStop = false;
this->groupBox1->Text = L"Параметры";
this->textBox_h->Location = System::Drawing::Point(75, 96);
this->textBox_h->Name = L"textBox_h";
this->textBox_h->Size = System::Drawing::Size(100, 22);
this->textBox_h->TabIndex = 5;
this->textBox_b->Location = System::Drawing::Point(75, 64);
this->textBox_b->Name = L"textBox_b";
this->textBox_b->Size = System::Drawing::Size(100, 22);
this->textBox_b->TabIndex = 4;
this->textBox_a->Location = System::Drawing::Point(75, 36);
this->textBox_a->Name = L"textBox_a";
this->textBox_a->Size = System::Drawing::Size(100, 22);
this->textBox_a->TabIndex = 3;
this->label6->AutoSize = true;
this->label6->Location = System::Drawing::Point(16, 99);
this->label6->Name = L"label6";
this->label6->Size = System::Drawing::Size(58, 16);
this->label6->TabIndex = 2;
this->label6->Text = L"h (шаг) =";
this->label5->AutoSize = true;
this->label5->Location = System::Drawing::Point(16, 67);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(53, 16);
this->label5->TabIndex = 1;
this->label5->Text = L"B (до) =";
this->label4->AutoSize = true;
this->label4->Location = System::Drawing::Point(16, 36);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(52, 16);
this->label4->TabIndex = 0;
this->label4->Text = L"A (от) =";
this->chart->Anchor =
static_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::AnchorStyles::Top |
System::Windows::Forms::AnchorStyles::Bottom)
| System::Windows::Forms::AnchorStyles::Left)
| System::Windows::Forms::AnchorStyles::Right));
chartArea1->Name = L"ChartArea1";
```

```
this->chart->ChartAreas->Add(chartArea1);
legend1->Name = L"Legend1";
this->chart->Legends->Add(legend1);
this->chart->Location = System::Drawing::Point(41, 40);
this->chart->Name = L"chart";
series1->BorderWidth = 3;
series1->ChartArea = L"ChartArea1";
series1->ChartType =
System::Windows::Forms::DataVisualization::Charting::SeriesChartType::Spline;
series1->Color =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(192)),
static_cast<System::Int32>(static_cast<System::Byte>(64)),
static_cast<System::Int32>(static_cast<System::Byte>(0)));
series1->LabelBorderColor = System::Drawing::Color::Transparent;
series1->Legend = L"Legend1";
series1->LegendText = L"что-то";
series1->Name = L"Series1";
series1->ShadowColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(192)),
static_cast<System::Int32>(static_cast<System::Byte>(0)),
static_cast<System::Int32>(static_cast<System::Byte>(0)));
this->chart->Series->Add(series1);
this->chart->Size = System::Drawing::Size(545, 300);
this->chart->TabIndex = 0;
this->chart->Text = L"chart1";
this->menuStrip1->ImageScalingSize = System::Drawing::Size(20, 20);
this->menuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) { this->меню ToolStripMenuItem });
this->menuStrip1->Location = System::Drawing::Point(3, 3);
this->menuStrip1->Name = L"menuStrip1";
this->menuStrip1->Size = System::Drawing::Size(1014, 28);
this->menuStrip1->TabIndex = 1;
this->menuStrip1->Text = L"menuStrip1";
this->меню ToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^ >(4) {
this->построитьГрафикToolStripMenuItem,
this->очиститьГрафикToolStripMenuItem, this-
>сохранитьГрафикToolStripMenuItem, this->построитьГрафикПоФайлуToolStripMenuItem
});
this->меню ToolStripMenuItem->Name = L"меню ToolStripMenuItem";
this->меню ToolStripMenuItem->Size = System::Drawing::Size(65, 24);
this->меню ToolStripMenuItem->Text = L"Меню";
this->построитьГрафикToolStripMenuItem->Name =
L"построитьГрафикToolStripMenuItem";
this->построитьГрафикToolStripMenuItem->Size =
System::Drawing::Size(287, 26);
this->построитьГрафикToolStripMenuItem->Text = L"Построить
график";
this->построитьГрафикToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm::построитьГрафикToolStripMenuItem_Click);
this->очиститьГрафикToolStripMenuItem->Name =
L"очиститьГрафикToolStripMenuItem";
this->очиститьГрафикToolStripMenuItem->Size =
System::Drawing::Size(287, 26);
this->очиститьГрафикToolStripMenuItem->Text = L"Очистить
график";
```

```
this->очиститьГрафикToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::очиститьГрафикToolStripMenuItem_Click);
this->сохранитьГрафикToolStripMenuItem->Name = L"сохранитьГрафикToolStripMenuItem";
this->сохранитьГрафикToolStripMenuItem->Size = System::Drawing::Size(287, 26);
this->сохранитьГрафикToolStripMenuItem->Text = L"Сохранить график";
this->сохранитьГрафикToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::сохранитьГрафикToolStripMenuItem_Click);
this->построитьГрафикПоФайлуToolStripMenuItem->Name = L"построитьГрафикПоФайлуToolStripMenuItem";
this->построитьГрафикПоФайлуToolStripMenuItem->Size = System::Drawing::Size(287, 26);
this->построитьГрафикПоФайлуToolStripMenuItem->Text = L"Построить график по файлу";
this->построитьГрафикПоФайлуToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::построитьГрафикПоФайлуToolStripMenuItem_Click);
this->tabPage3->Controls->Add(this->textBox9);
this->tabPage3->Controls->Add(this->button4);
this->tabPage3->Controls->Add(this->textBox8);
this->tabPage3->Controls->Add(this->label8);
this->tabPage3->Controls->Add(this->textBox7);
this->tabPage3->Controls->Add(this->comboBox2);
this->tabPage3->Controls->Add(this->applycomport);
this->tabPage3->Controls->Add(this->comboBox1);
this->tabPage3->Controls->Add(this->coutport);
this->tabPage3->Location = System::Drawing::Point(4, 25);
this->tabPage3->Name = L"tabPage3";
this->tabPage3->Size = System::Drawing::Size(1020, 443);
this->tabPage3->TabIndex = 2;
this->tabPage3->Text = L"Подключение";
this->tabPage3->UseVisualStyleBackColor = true;
this->textBox9->Location = System::Drawing::Point(236, 318);
this->textBox9->Name = L"textBox9";
this->textBox9->Size = System::Drawing::Size(100, 22);
this->textBox9->TabIndex = 8;
this->button4->Location = System::Drawing::Point(33, 331);
this->button4->Name = L"button4";
this->button4->Size = System::Drawing::Size(149, 57);
this->button4->TabIndex = 7;
this->button4->Text = L"Подключение";
this->button4->UseVisualStyleBackColor = true;
this->button4->Click += gcnew System::EventHandler(this, &MyForm::button4_Click);
this->textBox8->Location = System::Drawing::Point(236, 366);
this->textBox8->Name = L"textBox8";
this->textBox8->Size = System::Drawing::Size(100, 22);
this->textBox8->TabIndex = 6;
this->label8->AutoSize = true;
this->label8->Location = System::Drawing::Point(427, 171);
this->label8->Name = L"label8";
this->label8->Size = System::Drawing::Size(93, 16);
this->label8->TabIndex = 5;
this->label8->Text = L"BR значение";
this->textBox7->Enabled = false;
```

```
this->textBox7->Location = System::Drawing::Point(430, 190);
this->textBox7->Name = L"textBox7";
this->textBox7->Size = System::Drawing::Size(100, 22);
this->textBox7->TabIndex = 4;
this->comboBox2->FormattingEnabled = true;
this->comboBox2->Location = System::Drawing::Point(204, 188);
this->comboBox2->Name = L"comboBox2";
this->comboBox2->Size = System::Drawing::Size(166, 24);
this->comboBox2->TabIndex = 3;
this->applycomport->Location = System::Drawing::Point(33, 188);
this->applycomport->Name = L"applycomport";
this->applycomport->Size = System::Drawing::Size(129, 49);
this->applycomport->TabIndex = 2;
this->applycomport->Text = L"Подключение COM порта";
this->applycomport->UseVisualStyleBackColor = true;
this->applycomport->Click += gcnew System::EventHandler(this,
&MyForm::applycomport_Click);
this->comboBox1->FormattingEnabled = true;
this->comboBox1->Location = System::Drawing::Point(204, 51);
this->comboBox1->Name = L"comboBox1";
this->comboBox1->Size = System::Drawing::Size(166, 24);
this->comboBox1->TabIndex = 1;
this->coutport->Location = System::Drawing::Point(33, 39);
this->coutport->Name = L"coutport";
this->coutport->Size = System::Drawing::Size(130, 47);
this->coutport->TabIndex = 0;
this->coutport->Text = L"доступные порты";
this->coutport->UseVisualStyleBackColor = true;
this->coutport->Click += gcnew System::EventHandler(this,
&MyForm::coutport_Click);
this->tabPage2->Controls->Add(this->label7);
this->tabPage2->Controls->Add(this->textBox6);
this->tabPage2->Controls->Add(this->checkBox3);
this->tabPage2->Controls->Add(this->checkBox2);
this->tabPage2->Controls->Add(this->checkBox1);
this->tabPage2->Controls->Add(this->label3);
this->tabPage2->Controls->Add(this->label2);
this->tabPage2->Controls->Add(this->label1);
this->tabPage2->Controls->Add(this->textBox5);
this->tabPage2->Controls->Add(this->textBox4);
this->tabPage2->Controls->Add(this->textBox3);
this->tabPage2->Controls->Add(this->button5);
this->tabPage2->Location = System::Drawing::Point(4, 25);
this->tabPage2->Name = L"tabPage2";
this->tabPage2->Padding = System::Windows::Forms::Padding(3);
this->tabPage2->Size = System::Drawing::Size(1020, 443);
this->tabPage2->TabIndex = 1;
this->tabPage2->Text = L"Измерения пошаговые";
this->tabPage2->UseVisualStyleBackColor = true;
this->label7->AutoSize = true;
this->label7->Location = System::Drawing::Point(414, 160);
this->label7->Name = L"label7";
this->label7->Size = System::Drawing::Size(93, 16);
this->label7->TabIndex = 14;
this->label7->Text = L"Кол-во шагов";
```

```
this->label7->Click += gcnew System::EventHandler(this, &MyForm::label7_Click);
this->textBox6->Location = System::Drawing::Point(414, 188);
this->textBox6->Name = L"textBox6";
this->textBox6->Size = System::Drawing::Size(100, 22);
this->textBox6->TabIndex = 13;
this->checkBox3->AutoSize = true;
this->checkBox3->Location = System::Drawing::Point(198, 161);
this->checkBox3->Name = L"checkBox3";
this->checkBox3->Size = System::Drawing::Size(37, 20);
this->checkBox3->TabIndex = 12;
this->checkBox3->Text = L"Z";
this->checkBox3->UseVisualStyleBackColor = true;
this->checkBox2->AutoSize = true;
this->checkBox2->Location = System::Drawing::Point(198, 123);
this->checkBox2->Name = L"checkBox2";
this->checkBox2->Size = System::Drawing::Size(38, 20);
this->checkBox2->TabIndex = 11;
this->checkBox2->Text = L"Y";
this->checkBox2->UseVisualStyleBackColor = true;
this->checkBox1->AutoSize = true;
this->checkBox1->Location = System::Drawing::Point(198, 86);
this->checkBox1->Name = L"checkBox1";
this->checkBox1->Size = System::Drawing::Size(37, 20);
this->checkBox1->TabIndex = 10;
this->checkBox1->Text = L"X";
this->checkBox1->UseVisualStyleBackColor = true;
this->checkBox1->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox1_CheckedChanged);
this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(638, 169);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(201, 16);
this->label3->TabIndex = 9;
this->label3->Text = L"Частота лазерных импульсов";
this->label2->AutoSize = true;
this->label2->Location = System::Drawing::Point(638, 90);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(191, 16);
this->label2->TabIndex = 8;
this->label2->Text = L"Кол-во лазерных импульсов";
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(409, 81);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(91, 16);
this->label1->TabIndex = 7;
this->label1->Text = L"Размер шага";
this->label1->Click += gcnew System::EventHandler(this, &MyForm::label1_Click);
this->textBox5->Location = System::Drawing::Point(640, 188);
this->textBox5->Name = L"textBox5";
this->textBox5->Size = System::Drawing::Size(100, 22);
this->textBox5->TabIndex = 6;
this->textBox4->Location = System::Drawing::Point(638, 109);
this->textBox4->Name = L"textBox4";
this->textBox4->Size = System::Drawing::Size(100, 22);
```

```
this->textBox4->TabIndex = 5;
this->textBox3->Location = System::Drawing::Point(412, 100);
this->textBox3->Name = L"textBox3";
this->textBox3->Size = System::Drawing::Size(102, 22);
this->textBox3->TabIndex = 4;
this->button5->Location = System::Drawing::Point(16, 369);
this->button5->Name = L"button5";
this->button5->Size = System::Drawing::Size(145, 44);
this->button5->TabIndex = 3;
this->button5->Text = L"Пуск";
this->button5->UseVisualStyleBackColor = true;
this->button5->Click += gcnew System::EventHandler(this,
&MyForm::button5_Click);
this->tabPage1->BackColor = System::Drawing::Color::Honeydew;
this->tabPage1->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
this->tabPage1->Controls->Add(this->textBox1);
this->tabPage1->Controls->Add(this->textBox2);
this->tabPage1->Controls->Add(this->button1);
this->tabPage1->Location = System::Drawing::Point(4, 25);
this->tabPage1->Name = L"tabPage1";
this->tabPage1->Padding = System::Windows::Forms::Padding(3);
this->tabPage1->Size = System::Drawing::Size(1020, 443);
this->tabPage1->TabIndex = 0;
this->tabPage1->Text = L"Кнопки";
this->textBox1->Location = System::Drawing::Point(36, 82);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(100, 22);
this->textBox1->TabIndex = 0;
this->textBox1->TextChanged += gcnew System::EventHandler(this,
&MyForm::textBox1_TextChanged);
this->textBox2->Enabled = false;
this->textBox2->Location = System::Drawing::Point(203, 82);
this->textBox2->Name = L"textBox2";
this->textBox2->Size = System::Drawing::Size(100, 22);
this->textBox2->TabIndex = 1;
this->textBox2->TextChanged += gcnew System::EventHandler(this,
&MyForm::textBox2_TextChanged);
this->button1->Location = System::Drawing::Point(132, 133);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(75, 23);
this->button1->TabIndex = 2;
this->button1->Text = L"button1";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click_1);
this->tabPage->Anchor =
static_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::AnchorStyles::Top |
System::Windows::Forms::AnchorStyles::Bottom)
| System::Windows::Forms::AnchorStyles::Left)
| System::Windows::Forms::AnchorStyles::Right));
this->tabPage->Controls->Add(this->tabPage1);
this->tabPage->Controls->Add(this->tabPage2);
this->tabPage->Controls->Add(this->tabPage3);
this->tabPage->Controls->Add(this->tabPage4);
this->tabPage->Location = System::Drawing::Point(12, 33);
```

```
this->tabPage->Multiline = true;
this->tabPage->Name = L"tabPage";
this->tabPage->SelectedIndex = 0;
this->tabPage->Size = System::Drawing::Size(1028, 472);
this->tabPage->TabIndex = 4;
this->openFileDialog1->FileName = L"openFileDialog1";
this->saveFileDialog1->FileOk += gcnew
System::ComponentModel::CancelEventHandler(this, &MyForm::saveFileDialog1_FileOk);
this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);
this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(1072, 517);
this->Controls->Add(this->tabPage);
this->Controls->Add(this->bindingNavigator1);
this->MainMenuStrip = this->menuStrip1;
this->Name = L"MyForm";
this->Text = L"MyForm";
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>bindingNavigator1))->EndInit();
this->bindingNavigator1->ResumeLayout(false);
this->bindingNavigator1->PerformLayout();
this->tabPage4->ResumeLayout(false);
this->tabPage4->PerformLayout();
this->groupBox1->ResumeLayout(false);
this->groupBox1->PerformLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>chart))->EndInit();
this->menuStrip1->ResumeLayout(false);
this->menuStrip1->PerformLayout();
this->tabPage3->ResumeLayout(false);
this->tabPage3->PerformLayout();
this->tabPage2->ResumeLayout(false);
this->tabPage2->PerformLayout();
this->tabPage1->ResumeLayout(false);
this->tabPage1->PerformLayout();
this->tabPage->ResumeLayout(false);
this->ResumeLayout(false);
this->PerformLayout();

}
#pragma endregion
private: double a, b, h;
private: double x, y;
private: unsigned int index_comboBox2=0, index=300;
private: String^ fileName;
private: System::Void textBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e) {}
private: System::Void textBox2_TextChanged(System::Object^ sender,
System::EventArgs^ e) {}
private: System::Void button1_Click_1(System::Object^ sender, System::EventArgs^ e) {
textBox2->Text = System::Convert::ToString(textBox1->Text);
}


private: System::Void label1_Click(System::Object^ sender, System::EventArgs^ e) {}
private: System::Void button5_Click(System::Object^ sender, System::EventArgs^ e) {
```

```
		ofstream out("C:\\RESULT DIP.txt",ios::app);
		if (out.is_open()){
			if (checkBox1->Checked == true)//Запись координаты
				out << "X" <<"\n" << endl;
			if (checkBox2->Checked == true)
				out << "Y" << "\n" << endl;
			if (checkBox3->Checked == true)
				out << "Z" << "\n" << endl;
			if (textBox3->Text->Length != 0) //Запись задавемых значений{
				int step = Convert::ToInt16(this->textBox3->Text);
				out << "Размер шага: " << step << "\n";
			}
			if (textBox4->Text->Length != 0){
				int number_of_pulses = Convert::ToInt16(this->textBox4->Text);
				out << "Кол-во лазерных импульсов: " << number_of_pulses << "\n";
			}
			if (textBox5->Text->Length != 0){
				int frequency = Convert::ToInt16(this->textBox5->Text);
				out << "Частота лазерных импульсов: " << frequency << "\n";
			}
			if (textBox6->Text->Length != 0){
				int frequency = Convert::ToInt16(this->textBox6->Text);
				out << "Кол-во шагов: " << frequency << "\n";
			}
		}
		out.close();
		checkBox1->Checked = false;
		checkBox2->Checked = false;
		checkBox3->Checked = false;
	}
	private: System::Void checkBox1_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {}
	private: System::Void chart1_Click(System::Object^ sender, System::EventArgs^ e) {}
	private: System::Void построитьГрафикToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
		a = Convert::ToDouble(this->textBox_a->Text);
		b = Convert::ToDouble(this->textBox_b->Text);
		h = Convert::ToDouble(this->textBox_h->Text);
		x = a;
		this->chart->Series[0]->Points->Clear();
		while (x < b){
			y = Math::Cos(x);
			this->chart->Series[0]->Points->AddXY(x, y);
			x += h;
		}
	}
	private: System::Void очиститьГрафикToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
		this->chart->Series[0]->Points->Clear();
	}
	private: System::Void сохранитьГрафикToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
```

```
		SaveFileDialog^ openDlgsave = gcnew SaveFileDialog();
		openDlgsave->Filter = "JPeg Image|*.jpg|Bitmap Image|*.bmp|Gif Image|*.gif|All files
(*.*)|*.*";
		if (openDlgsave->ShowDialog() == System::Windows::Forms::DialogResult::OK){
			chart->SaveImage(openDlgsave->FileName, ImageFormat::Jpeg);
		}}
	private: System::Void label7_Click(System::Object^ sender, System::EventArgs^ e) {}
	private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {}
	private: System::Void saveFileDialog1_FileOk(System::Object^ sender,
System::ComponentModel::CancelEventArgs^ e) {}
	private: System::Void построитьГрафикПоФайлуToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
		ofstream file_graph;
		char delimiter = ',';
		OpenFileDialog^ openDlg = gcnew OpenFileDialog();
		openDlg->Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.* | csv files (*.csv)|*csv";
		if (openDlg->ShowDialog() == System::Windows::Forms::DialogResult::OK){
			/*fileName = gcnew String(openDlg->FileName->ToString());*/
			fileName = openDlg->FileName;
		}
		msclr::interop::marshal_context context;
		string unused, data1, data2, strfileName = context.marshal_as<std::string>(fileName);
		ifstream in{ strfileName };
		string line, field;
		string x1, y1;
		double x_gr, y_gr,p1,p2;
		char comma = ',', point ='.';
		map<string, string> store;
		map <double, double> store2;
		getline(in, line);
		while (getline(in, line)){
			stringstream ss(line);
			getline(ss, data1, ';');
			getline(ss, data2, ';');
			replace(data1.begin(), data1.end(), ',', '.');
			replace(data2.begin(), data2.end(), ',', '.');
			store2[stod(data1)] = stod(data2);
		}
		this->chart->Series["Series1"]->Points->Clear();
		for (auto i : store2){
			x_gr = i.first;
			y_gr = i.second;
			this->chart->Series["Series1"]->Points->AddXY(x_gr, y_gr);
		}}
	private: System::Void coutport_Click(System::Object^ sender, System::EventArgs^ e) {
		cli::array<String^>^ serialPorts = nullptr;
		try{
			serialPorts = SerialPort::GetPortNames(); //Получение списка доступных
COM-портов.
		}
		catch (Win32Exception^ ex){
			MessageBox::Show("Error Win32Exception  " + ex->ToString());
		}
		comboBox1->Items->Clear();
		for each (String ^ port in serialPorts) {
			comboBox1->Items->Add(port);
```

```
		}
		if (comboBox1->Items->Count > 0) comboBox1->SelectedIndex = 0;
		if (index_comboBox2 == 0){
			comboBox2->Items->Add(index);
			index = 2 * index;
			for (index_comboBox2 = 1; index_comboBox2 < 8; index_comboBox2++){
				comboBox2->Items->Add(index);
				index = index * 2;
		}
			comboBox2->Items->Add(57600);
			comboBox2->Items->Add(115200);
		}
	}
	private: System::Void applycomport_Click(System::Object^ sender, System::EventArgs^ e) {
		if (comboBox1->SelectedIndex == -1) {
						MessageBox::Show("Ошибка! Не выбран Com-port");
						return;
		}
		int BR;
		switch (comboBox2->SelectedIndex){
				case 0: BR = 300;
					break;
				case 1: BR = 600;
					break;
				case 2: BR = 1200;
					break;
				case 3: BR = 2400;
					break;
				case 4: BR = 4800;
					break;
				case 5: BR = 9600;
					break;
				case 6: BR = 19200;
					break;
				case 7: BR = 38400;
					break;
				case 8: BR = 57600;
					break;
				case 9: BR = 115200;
					break;
				}
		textBox7->Text = System::Convert::ToString(BR);
		serialPort1->PortName = comboBox1->SelectedItem->ToString();
		serialPort1->BaudRate = BR;
		serialPort1->Parity = Parity::None;
		serialPort1->DataBits = 8;
		serialPort1->StopBits = StopBits::One;
		serialPort1->ReadTimeout = 1000;
		serialPort1->WriteTimeout = 1000;
		try {
				serialPort1->Open();
				set_flag = 1;
			}
		catch (System::Exception^ ex) {
```

```
                MessageBox::Show("Не удаётся подключиться к serial port: " + ex->Message);
                set_flag = 0;
                return;
            }}
        private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e) {
            String^ dataa = textBox9->Text;
            serialPort1->WriteLine(dataa + 0x0D);
            bool flag = true;
            String^ recive_data;
            while (flag){
                try{
                    recive_data = serialPort1->ReadLine();
                    flag = false;
                }
                catch (TimeoutException^) {}
            }
            textBox8->Text = recive_data;
        }};}
```

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

СПРАВКА

Томский Государственный Университет

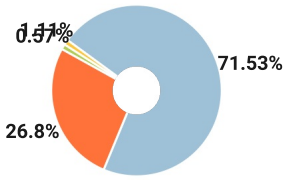
о результатах проверки текстового документа
на наличие заимствований

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Юрченко Екатерина Алексеевна
Самоцитирование рассчитано для: Юрченко Екатерина Алексеевна
Название работы: ВКР Юрченко
Тип работы: Выпускная квалификационная работа
Подразделение:

РЕЗУЛЬТАТЫ

СОВПАДЕНИЯ	26.8%
ОРИГИНАЛЬНОСТЬ	71.53%
ЦИТИРОВАНИЯ	0.57%
САМОЦИТИРОВАНИЯ	1.11%



ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 07.06.2023

Структура документа: Проверенные разделы: основная часть с.2-3, 5-40

Модули поиска: ИПС Адилет; Библиография; Сводная коллекция ЭБС; Интернет Плюс*; Сводная коллекция РГБ; Цитирование; Переводные заимствования (RuEn); Переводные заимствования по eLIBRARY.RU (EnRu); Переводные заимствования по коллекции Гарант: аналитика; Переводные заимствования по коллекции Интернет в английском сегменте; Переводные заимствования по Интернету (EnRu); Переводные заимствования по коллекции Интернет в русском сегменте; Переводные заимствования издательства Wiley ; eLIBRARY.RU; СПС ГАРАНТ: аналитика; СПС ГАРАНТ: нормативно-правовая документация; Медицина; Диссертации НББ; Коллекция НБУ; Перефразирования по eLIBRARY.RU; Перефразирования по СПС ГАРАНТ: аналитика; Перефразирования по Интернету; Перефразирования по Интернету (EN); Перефразированные заимствования по коллекции Интернет в английском сегменте; Перефразированные заимствования по коллекции Интернет в русском сегменте; Перефразирования по коллекции

Работу проверил: Лапутенко Андрей Владимирович

ФИО проверяющего

Дата подписи: 07-06-2023

Подпись проверяющего

Чтобы убедиться
в подлинности справки, используйте QR-код,
который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование корректным, система оставляет на усмотрение проверяющего. Предоставленная информация не подлежит использованию в коммерческих целях.