

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М.А. Бонч-Бруевича»

На правах рукописи

Анваржонов Баходиржон Нодирбек угли

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА МОДЕЛЕЙ И МЕТОДОВ ПЕРЕДАЧИ
ДАННЫХ ДЛЯ КОСТЮМА ТЕЛЕПРИСУТСТВИЯ В СЕТЯХ СВЯЗИ
ПЯТОГО И ПОСЛЕДУЮЩИХ ПОКОЛЕНИЙ**

2.2.15. Системы, сети и устройства телекоммуникаций

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель
доктор технических наук
Мутханна Аммар Салех Али

Санкт-Петербург – 2026

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ.....	2
ВВЕДЕНИЕ.....	5
ГЛАВА 1. Анализ и обзор иммерсивных технологий для предоставления услуг метавселенной и интернета навыков.....	12
1.1. Концепция иммерсивных сетей в метавселенных и в целом.....	12
1.2. Методы предоставления иммерсивных технологий.....	12
1.2.1. Захват движений и состояния пользователя.....	12
Позиционирование и ориентация пользователя в пространстве.....	12
Телеметрия пользователя (Телеметрическое состояние пользователя).....	16
1.2.2. Передача обратной связи пользователю.....	19
Кинестетическая-силовая обратная связь.....	19
Тактильно-рецепторная (кожная) обратная связь.....	20
Болевые и подобные воздействия с применением тактильной-электрической обратной связи на основе ЭМС, ЧЕНЧ и ФЭС.....	21
Вкус и аромат.....	22
1.3. Существующие КТП на момент проведения исследований.....	26
1.3.1. TeslaSuit.....	26
1.3.2. Senso Suit.....	28
1.3.3. Perception Neuron 3.....	28
1.3.4. OWO Skin haptic suit.....	30
1.3.5. HoloSuit.....	31
1.3.6. KOLD.....	33
1.4. Требования и проблемы реализации иммерсивных услуг.....	35
1.5. Проблемы проектирования тактильного интернета и иммерсивных коммуникаций в 6G... 37	
1.6. Анализ методов и подходов к реализации иммерсивных услуг.....	42
1.6.1. Подходы к удовлетворению требований к задержке.....	42
1.6.2. Роль интеллектуальных граничных вычислений и стандартные компоненты.....	43
1.6.3. Пограничные и облачные вычисления.....	45
1.6.4. Применение EI для удовлетворения требований к качеству обслуживания (QoS/QoE).....	46
1.7. Вывод по главе.....	47
ГЛАВА 2. АДАПТИВНЫЙ МЕТОД ПЕРЕДАЧИ ДАННЫХ С ПОВЫШЕНИЕМ ЭФФЕКТИВНОСТИ КАНАЛА ДЛЯ ТЕРМОДАТЧИКОВ КОСТЮМА ТЕЛЕПРИСУТСТВИЯ (КТП).....	49
2.1. Модель костюма телеприсутствия.....	50
2.2. Модель сигналов термодатчиков КТП.....	53
2.3. Представление данных о температуре и их воспроизведение КТП.....	58
2.4. Модель и метод формирования данных.....	64
2.5. Выводы по главе.....	69
ГЛАВА 3. МЕТОД ПЕРЕДАЧИ ТАКТИЛЬНОЙ И КИНЕСТЕТИЧЕСКОЙ ИНФОРМАЦИИ НА БАЗЕ КОСТЮМА ТЕЛЕПРИСУТСТВИЯ.....	71

3.1. Классификация костюмов телеприсутствия.....	71
3.2. Разработка костюма телеприсутствия.....	72
3.2.1. Сетевая схема взаимодействия элементов КТ и соответствующие требования.....	73
3.2.2. Сценарий использования КТП.....	74
3.2.3. Сеть контроллеров и сенсорики костюма телеприсутствия.....	74
3.2.4. Составляющие модули КТП и их характеристики.....	75
3.2.5. Составляющие узлы КТП.....	78
3.2.6. Алгоритмы работы загруженных в МК программ и их описание.....	85
3.3. Структура и назначение используемых выводов управляющих плат.....	91
3.4. Используемые интерфейсы на КТП для контроля и мониторинга применяемых сенсоров и актуаторов.....	97
3.5. Электромагнитная совместимость КТП.....	98
3.5.1. Наиболее уязвимые для помех структурные элементы костюма телеприсутствия..	99
3.5.2. Рекомендации по снижению уровней внутренних помех радиоэлектронным элементам костюма телеприсутствия с целью обеспечения их электромагнитной совместимости.....	100
3.6. Расчет автономности КТП.....	104
3.7. Выводы по главе.....	106
ГЛАВА 4. РАЗРАБОТКА МЕТОДОВ СГЛАЖИВАНИЯ И АДАПТИВНОЙ КОРРЕКТИРОВКИ АНАЛОГОВЫХ СИГНАЛОВ.....	108
4.1. Алгоритм обнаружения помех.....	108
4.2. Алгоритм калибровки.....	109
4.3. Разработанная версия алгоритма скользящего среднего.....	110
4.4. Алгоритм корректировки помех.....	111
4.5. Результаты внедрения разработанных фильтров.....	112
4.6. Примеры применения.....	116
4.7. Выводы по главе.....	116
ГЛАВА 5. МОДЕЛЬНАЯ СЕТЬ ДЛЯ ПЕРЕДАЧИ ИММЕРСИВНЫХ ОЩУЩЕНИЙ В КОСТЮМЕ ТЕЛЕПРИСУТСТВИЯ С ОБРАТНОЙ СВЯЗЬЮ.....	118
5.1. Подход к разработке улучшенной модели передачи иммерсивных данных перчатки КТП.	118
5.2. Модифицированные алгоритмы работы загруженных программ.....	121
5.3. Описание модулей используемых для воспроизведения иммерсивных данных.....	123
5.3.1. Температурная обратная связь с применением элемента Пельтье, специального драйвера.....	124
5.3.2. ЦАП-модуль с буфером для управления несколькими драйверами элемента Пельтье	129
5.3.3. NTC - термисторы для мониторинга температуры составляющих устройства.....	134
5.3.4. Таблеточный вибромоторчик и его драйвер для воспроизведения хаптики.....	137
5.3.5. Проверка температурной обратной связи с применением мини конструкции.....	141
5.4. Итоговая схема подключения составляющих тактильной перчатки.....	145
5.5. Разработка рекомендаций по реализации костюма телеприсутствия, доступного для ежедневного использования.....	150
5.6. Выводы по главе.....	150

ЗАКЛЮЧЕНИЕ.....	151
СПИСОК ЛИТЕРАТУРЫ.....	154
ПРИЛОЖЕНИЕ А - Исполняемый код ESP-32.....	166
Листинг. Скетч-код для плат с ESP32 текущей версии верхних конечностей КТП.....	166
Листинг. Скетч-код для плат с ESP32 текущей версии перчатки КТП.....	176
Листинг. Библиотека с самодельным фильтром скользящего среднего.....	188
Листинг. Библиотека с разработанным адаптивным фильтром корректировки аналоговых сигналов.....	189
Листинг. Библиотека для прошивки используемых плат по воздуху (через локальную сеть)....	193
ПРИЛОЖЕНИЕ В - Акты о внедрении результатов исследования.....	195
ПРИЛОЖЕНИЕ Г - Сведения о полученных патентах, авторских свидетельствах.....	197

ВВЕДЕНИЕ

Актуальность темы исследования

В эпоху пост-интернета, когда его повсеместность настолько глубоко встроена в повседневную жизнь, метавселенная представляет собой новую фазу его развития, где пользователи будут активно взаимодействовать в многомерных онлайн-пространствах. Метавселенная перестала быть просто модным термином и перешла в стадию практических исследований и пилотных проектов. Её актуальность и значимость определяется не сиюминутным интересом, а с долгосрочными трендами в цифровизации общества, экономики и коммуникаций нового уровня.

Развитие метавселенной неразрывно связано с эволюцией телекоммуникационных технологий. В частности, будущая сеть 6G рассматривается как ключевое условие для реализации настоящего иммерсивного опыта. Интеграция метавселенных и 6G-технологий открывает беспрецедентные возможности для создания более реалистичных и интерактивных виртуальных сред. Высокая скорость передачи данных 6G позволит мгновенно передавать большие объёмы информации, что критически важно для обеспечения плавного и непрерывного взаимодействия пользователей с метавселенными. Кроме того, высокая пропускная способность и низкая задержка 6G способствуют более точной и детализированной визуализации, что существенно повышает уровень погружения в виртуальные миры.

С технологической точки зрения, 6G предоставляет новые возможности для развития беспроводных сенсорных сетей, что позволяет интегрировать метавселенные с физическим миром на новом уровне. Это открывает перспективы для создания гибридных сред, где виртуальные и реальные объекты взаимодействуют в реальном времени, создавая эффект присутствия и интерактивности.

В диссертации представлен опыт, полученный в результате работы над прототипом костюма телеприсутствия, который является интерфейсом для взаимодействия физического и виртуального миров и реализует один из методов предоставления иммерсивных услуг. Костюм телеприсутствия в сочетании с шлемом виртуальной реальности позволят пользователю не только видеть, но и ощущать физическое взаимодействие в цифровом пространстве.

Данное исследование в диссертационной работе позволяет расширить имеющиеся знания в области телеприсутствия, которая является ключевым элементом метавселенной. Анализ исследований в мировом научном сообществе по данной тематике, что служит вектором определяющим направление в развитии дальнейших исследований, представлены описания архитектурных аспектов сетевого взаимодействия и модель сети, изложены алгоритмы взаимодействия составных частей костюма телеприсутствия между собой и физическим миром. Таким образом, разработка и исследование методов обработки и передачи данных для костюма телеприсутствия в сетях связи пятого и последующих поколений представляет собой актуальную научную задачу.

Степень разработанности темы исследования. Определяющий вклад в теоретические и экспериментальные исследования рассматриваемых научных проблем внесли российские и зарубежные ученые: А.Е. Кучерявый, Е.А. Кучерявый, К.Е. Самуйлов, А.Н. Волков и О.А. Маркелов, А.С. Аджемов, В.М. Вишневский, Ю.В. Гайдамака, В.С. Елагин, А.И. Парамонов, С.Н. Степанов, В.Г. Карташевский, В.О. Тихвинский, Б.С. Гольдштейн, А.Б. Гольдштейн, а также ведущие зарубежные исследователи M. Z. Chowdhury, M. Latva-aho, H. Pennanen, C. Puliafito, M. Verma и A. A. Ateya.

Работы вышеупомянутых авторов внесли значительный вклад в исследования в области сетей связи пятого поколения, гетерогенного трафика, сокращения задержек, методов передачи данных для представления иммерсивных услуг.

Настоящая диссертация, в отличие от известных подходов к построению и функционированию сетей связи, направлена на решение научной задачи разработки моделей и методов обработки и передачи данных для костюма телеприсутствия в сетях связи пятого и последующих поколений для представления иммерсивных услуг.

Объект и предмет исследования. Объектом исследования являются сети связи пятого и последующих поколений. Предметом исследования являются модели и

методы представления иммерсивных услуг для костюма телеприсутствие в сетях связи пятого и последующих поколений

Цель и задачи исследования. Уменьшение объема трафика, поступающего в сеть связи от костюма телеприсутствия, за счет разработки моделей и методов передачи иммерсивных ощущений, включающих три типа обратной связи: температурную, тактильную и кинестетическую.

Цель работы достигается путём последовательного решения следующих задач:

- провести анализ существующих методов предоставления иммерсивных технологий;
- разработать метод сжатия динамического диапазона сигналов с помощью S-кривой;
- разработать адаптивный метод передачи данных о температуре костюма телеприсутствия;
- разработать модельную сеть для костюма телеприсутствия
- разработать модель передачи иммерсивных ощущений в костюме телеприсутствия, включающая три типа обратной связи

Научная новизна. Научная новизна работы состоит в следующем:

- Исследованы свойства сигналов датчиков температуры костюма телеприсутствия, разработан метод компрессии сигналов, обеспечивающий уменьшение объема передаваемых данных, а также определены допустимые границы изменения температурного воздействия и точность его представления. В отличие от существующих решений данный метод позволяет уменьшить объем трафика, поступающего в сеть связи от подсистемы данных о температуре костюма телеприсутствия, на 20%.
- Разработан метод формирования данных, получаемых от датчиков температуры костюма телеприсутствия, учитывающий случайный характер их значений и специфические особенности, а также обеспечивающий уменьшение объема передаваемых данных. По сравнению с существующими решениями данный метод позволяет получить выигрыш в объеме передаваемых данных до 30%.

- Создана модель передачи иммерсивных ощущений в костюме телеприсутствия. В отличие от существующих решений данная модель включает три взаимосвязанных типа обратной связи: температурную, тактильную и кинестетическую.

Теоретическая значимость работы. Теоретическая значимость диссертационной работы заключается в установлении зависимости между вероятностными свойствами сигналов термодатчиков, разрядностью их кодирования и объёмом передаваемых данных, позволяющие обоснованно выбирать параметры формата передачи в зависимости от числа датчиков и дисперсии измеряемых температур. Также в установлении аналитического выражения для выигрыша в объёме данных как функции количества датчиков и разброса их значений, что позволяет на этапе проектирования системы прогнозировать ожидаемую нагрузку на канал. Теоретическая значимость работы также заключается в развитии научно-методических основ формирования иммерсивных ощущений в системах телеприсутствия. Предложенная модель расширяет представления о передаче сенсорной информации за счет комплексного использования температурной, тактильной и кинестетической обратной связи.

Практическая значимость работы. Практическая ценность работы состоит в том, что данные результаты могут быть использованы при проектировании подсистем обмена тактильными данными в устройствах телеприсутствия и интерфейсах взаимодействия с метавселенными, при проектировании систем телеприсутствия, дистанционного управления роботами и иммерсивных сервисов в сетях 5G/6G, Также для чтения лекций и проведение практических занятий по курсам «Интернет вещей и самоорганизующиеся сети», «Современные проблемы науки в области инфокоммуникаций», «Системы, сети и устройства телекоммуникаций».

Практическая значимость работы заключается в возможности применения разработанной модели при создании и совершенствовании костюмов телеприсутствия и других систем человек–аватара.

Реализация результатов работы. Полученные в диссертационной работе результаты внедрены в Санкт-Петербургском государственном университете телекоммуникаций им. проф. М.А. Бонч-Бруевича при чтении лекций и проведении практических занятий по курсам «Искусственный интеллект в сетях и системах связи», "Современные проблемы науки в условиях перехода к сетям шестого поколения (6G)», «Системы, сети и устройства телекоммуникаций". Акт реализации диссертационных исследований представлен в Приложении к диссертации.

Методы исследования. При проведении исследований были использованы методы теоретического исследования, теории оптимизации, методы имитационного и натурного моделирования на основе модельной сети.

Положения, выносимые на защиту

Основные положения, выносимые на защиту:

1. Метод адаптивного сжатия и кодирования данных о температуре для костюма телеприсутствия с помощью S-кривой, позволяющий на 20% сократить разрядность двоичного представления сигнала, что также дает возможность уменьшить объем трафика, поступающего в сеть связи от подсистемы данных о температуре костюма телеприсутствия, на 20%.
2. Адаптивный метод передачи данных о температуре костюма телеприсутствия, позволяющий получить выигрыш в объеме передаваемых данных до 30% за счет использования вероятностных свойств сигналов, величина выигрыша зависит от числа термодатчиков и степени разброса значений температуры.
3. Модель передачи иммерсивных ощущений в костюме телеприсутствия, включающая три типа обратной связи: температурную, тактильную и кинестетическую и подсистема модельной сети для услуг телеприсутствия, в которых предусмотрена возможность индивидуальной калибровки

интенсивности каждого типа воздействия, что позволяет обеспечить адекватное восприятие виртуальной и физической среды.

Степень достоверности и апробация результатов

Результаты были проверены с помощью моделирования и математического анализа. Итерационные алгоритмы были протестированы в различных сценариях, включая натурные эксперименты на физических устройствах различных аппаратных архитектур. Результаты исследований обсуждались на международных конференциях и получили положительную экспертную оценку в рецензируемых изданиях с признанным научным авторитетом, что еще раз подтвердило их достоверность.

Основные результаты диссертационной работы докладывались и обсуждались на Международной научно-технической и научно-методической конференции и семинарах: IEEE Future Networks World Forum (FNWF), Дубай, Объединенные Арабские Эмираты, 2024 года; DCCN (International conference on Distributed Computer and Communication Networks: Control, Computation, Communications) в 2023 году; Международная конференция по современным системам вычислений и связям следующего поколения (International Conference on Advanced Computing & NextGeneration Communication ICACNGC 2024) – Санкт-Петербург, Российская Федерация – 2022; Международная конференция по технологиям голографического телеприсутствия и голосетевселенной (International Conference on Holographic Telepresence Technologies and Holnetverse) – Санкт-Петербург, Российская Федерация – 2024; 14-ая международная научно-техническая конференция. Владимир, 2021; й XI Международная научно-техническая и научно-методическая конференция (АПИНО 2022, АПИНО 2024); Всероссийская научно-техническая и научно-методическая конференции (ПКМ-2022); Научно-техническая конференция Санкт-Петербургского НТО РЭС им. А.С. Попова, посвященная Дню радио. – 2026.

Публикации по теме диссертации. Всего по теме диссертации опубликовано 10 работ. В числе опубликованных работ – 3 статьи в рецензируемых научных журналах, входящих в перечень изданий, рекомендуемых ВАК Министерства высшего образования и науки Российской Федерации (из них 3 в изданиях, соответствующих искомой специальности), 2 статей в рецензируемых изданиях, входящих в международные базы данных SCOPUS и WoS , 5 работы в журналах и сборниках докладов конференций, включенных в РИНЦ.

Соответствие паспорту специальности. Содержание диссертации соответствует следующим пунктам паспорта специальности 2.2.15. Системы, сети и устройства телекоммуникаций и соответствует следующим пунктам паспорта специальности: 1, 2, 4, 9, 12.

Личный вклад автора. Основные результаты диссертации получены автором самостоятельно. В экспериментальных исследованиях автор принимал непосредственное участие на всех этапах.

ГЛАВА 1. Анализ и обзор иммерсивных технологий для предоставления услуг метавселенной и интернета навыков.

1.1. Концепция иммерсивных сетей в метавселенных и в целом

Иммерсивные сети - это технологии, создающие эффект полного или частичного погружения в виртуальный мир или смешение виртуального и реального мира[33-37].

Это сети и технологии, которые позволяют дистанционно передавать тактильные ощущения между виртуальными и физическими мирами с ультра малыми задержками.

Метавселенная - это цифровая вселенная, в которой пользователи могут взаимодействовать друг с другом, работать, играть и общаться в постоянной, общей 3D-среде, доступ к которой часто осуществляется с помощью устройств Иммерсивных технологий. По сути, это сеть взаимосвязанных виртуальных миров, объединяющих физическую и цифровую среды[49].

На сегодняшний день можно имитировать или воспроизвести почти все внешние физические ощущения и некоторые внутренние ощущения используя специальные нейросимуляторы и микротоки.

1.2. Методы предоставления иммерсивных технологий

1.2.1. Захват движений и состояния пользователя

Существуют множество методов (сенсоров) для захвата движений и физического состояния пользователя.

К основным из них можно отнести следующие:

Позиционирование и ориентация пользователя в пространстве

А. Захват с помощью камер наблюдения (Motion Capture)

Стоит отметить, что можно использовать как одну, так и несколько обычных или же трехмерных камер, а также задействовать или нет специальные нательные метки и хромакей на заднем плане.

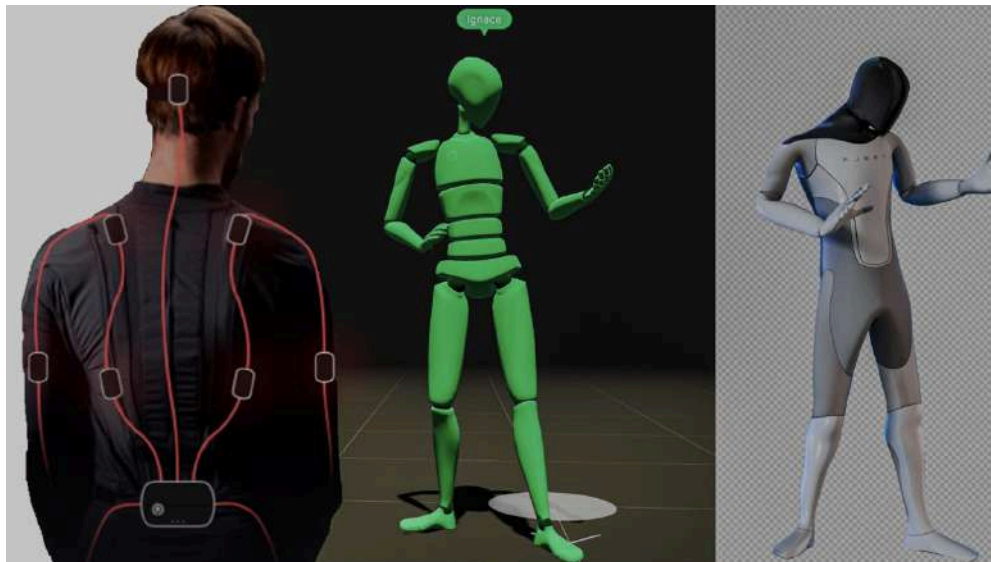


Рисунок 1.1 - Motion Capture

Таблица 1.1 - Достоинства и недостатки метода Motion Capture

Достоинства	Недостатки
Простота реализации	Ограничение в пространстве
Некоторые системы не требуют дополнительных меток, что облегчает использование	
Доступность	Малая точность
Нет ограничения движения частей тела	

В. Применение инерционных IMU сенсоров

Инерционные измерительные устройства (IMU) представляют собой комплексный набор датчиков, предназначенных для измерения и регистрации физических параметров движения объекта в пространстве.

К таким сенсорам можно отнести акселерометр, гироскоп, магнитометр, барометр и другие подобные сенсоры.

- **Акселерометр:** Измеряет ускорение объекта в трех осях пространства. Она определяет, насколько быстро или медленно объект движется вверх/вниз, вперед/назад и влево/вправо.
- **Гироскоп:** Измеряет угловую скорость объекта вокруг трех осей. Этот датчик позволяет определить, как быстро объект вращается вокруг каждой из осей.

- **Магнитометр:** Измеряет магнитное поле вокруг объекта, что помогает определить ориентацию в пространстве относительно магнитного севера.

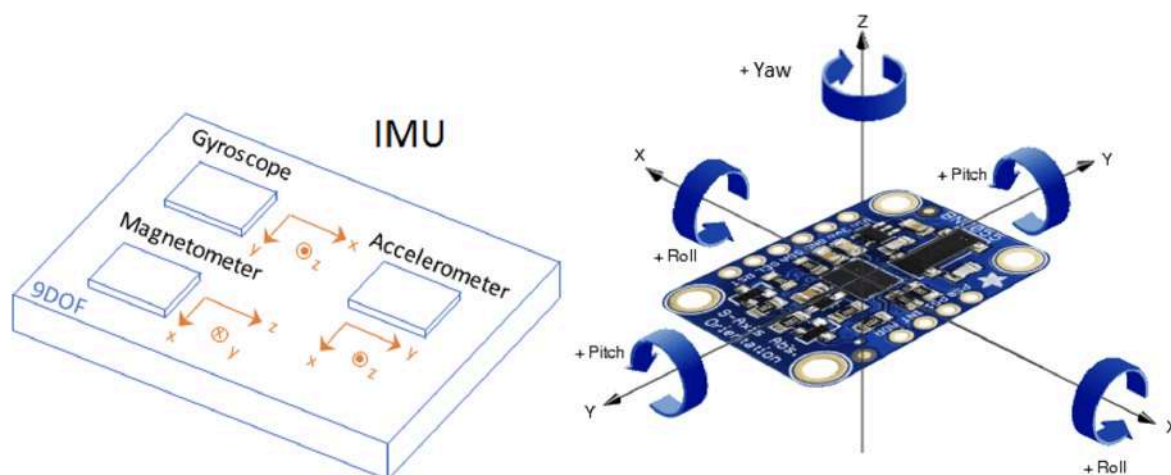


Рисунок 1.2 - Внешний вид IMU датчика

Таблица 1.2 - Достоинства и недостатки использования IMU датчиков

Достоинства	Недостатки
Размер	Отклонение от реальных значений со временем
Независимость от места установки	Неустойчивость к электромагнитным полям
Не изнашивается, т.к. на него не действуют механические силы	Требует постоянной калибровки

С. Использование элементов с переменным сопротивлением

Элементы с переменным сопротивлением представляют собой компоненты электрической цепи, сопротивление которых может изменяться в зависимости от определенных параметров или воздействий. Эти элементы используются для управления током и напряжением в электрических цепях, а также для создания устройств с регулируемыми характеристиками.

К ним можно отнести потенциометры, датчики изгиба или растяжения (токопроводящие текстильные нити).

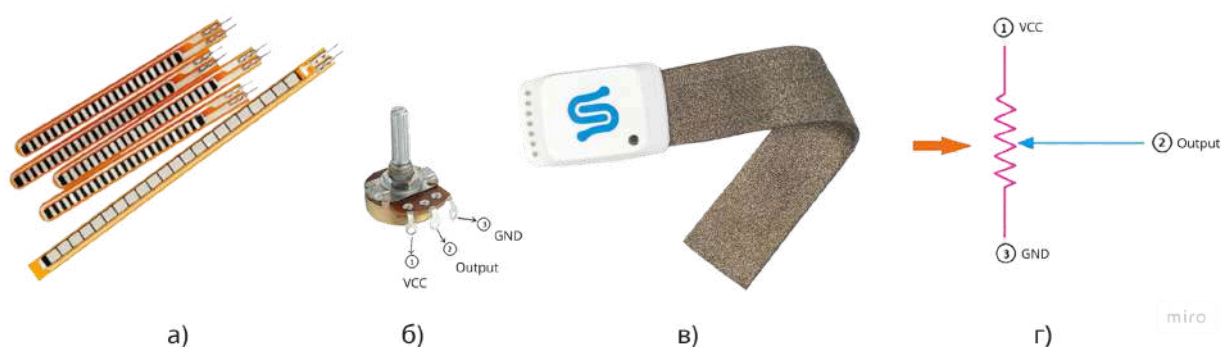


Рисунок 1.3 - Резистивные сенсоры: а) датчик изгиба; б) потенциометр; в) датчик растяжения/деформации; г) типичные выводы этих сенсоров

Таблица 1.3 - Достоинства и недостатки использования элементов с переменным сопротивлением

Достоинства	Недостатки
Скорость считывания	Влияние шумов на передаваемый сигнал
Стоимость	Низкая износостойкость
Не требует постоянной калибровки	Точность и стабильность работы

Д. Применение энкодеров

Эти модули, обеспечивая высокую точность, широко применяются во многих устройствах, и нашли широкое применение в разных сферах, от домашнего быта до промышленности. На сегодняшний день, можно встретить разные виды энкодеров, от простых оптических до электромагнитных (с высокой точностью).



Рисунок 1.4 - Внешний вид энкодера

Таблица 1.4 - Достоинства и недостатки использования энкодеров

Достоинства	Недостатки
Высокая точность	Высокая стоимость
Износостойкость	Обычно, требуют много выводов для считывания значений

Телеметрия пользователя (Телеметрическое состояние пользователя)

Телеметрические данные пользователя необходимы для адаптации тактильных интерфейсов под текущее состояние организма: от температуры кожи и частоты пульса до мышечной и мозговой активности.

Первым и наиболее простым параметром, поддающимся непрерывному мониторингу, является температура частей тела пользователя или элементов используемого тактильного устройства.

А. Температура частей тела пользователя или частей используемого тактильного устройства

Для определения температур, в основном используют **NTC датчики температуры**.

Это уникальные устройства, применяемые для измерения и контроля температуры в различных системах. Название NTC является аббревиатурой от Negative Temperature Coefficient, что означает "отрицательный температурный коэффициент". При повышении температуры датчика его сопротивление уменьшается, а при понижении – возрастает.



Рисунок 1.5 - Внешний вид термистора

Соотношение температуры и сопротивления датчика температуры на 10 кОм показано на следующей таблице.

Таблица 1.5 - Соотношение температуры и сопротивления датчика температуры

Температура, °C	5	10	20	30	40
Сопротивление, Ом	22070	17960	12091	8312	5827

В. ЭКГ (сердцебиение) и ЭЭГ (нейро-активность мозга)

Электрокардиография (ЭКГ) в тактильном интернете позволяет учитывать сердечный ритм пользователя, который влияет на чувствительность к прикосновениям: мы лучше ощущаем касания в момент расслабления сердца (между ударами) и хуже — в момент пульсового толчка. Дыхание тоже связано с ритмом сердца и влияет на восприятие. Интеграция ЭКГ даёт возможность подстраивать подачу тактильных сигналов под естественные циклы организма пользователя, делая ощущения более естественными.

Кроме этого, данные ЭКГ могут использоваться для оценки эмоционального состояния пользователя в реальном времени.

Электроэнцефалография (ЭЭГ) позволяет регистрировать электрическую активность головного мозга и в режиме реального времени отслеживать когнитивные состояния пользователя — уровень внимания, усталость, эмоциональную вовлечённость и моторное намерение. В контексте тактильного интернета ЭЭГ выступает не столько как канал вывода сенсорных данных, сколько как инструмент адаптации: система может подстраивать интенсивность, частоту или тип тактильных стимулов в зависимости от текущего состояния мозга пользователя.

Например, в реабилитационных системах ЭЭГ-интерфейсы позволяют пациентам с двигательными нарушениями управлять экзоскелетами или роботизированными протезами через воображение движения (моторный образ). В игровых и обучающих сценариях данные ЭЭГ помогают регулировать сложность заданий: если уровень внимания падает, система упрощает задачу или меняет характер обратной связи, удерживая пользователя в состоянии оптимальной вовлечённости.

Кроме того, интеграция ЭЭГ с тактильными интерфейсами открывает возможность создания замкнутых нейроадаптивных систем: устройство не только передаёт тактильный сигнал, но и анализирует мозговой ответ на него,

корректируя параметры стимуляции в реальном времени. Это позволяет выстраивать персонализированные сценарии взаимодействия, учитывающие индивидуальные нейрофизиологические особенности пользователя и обеспечивающие максимальную эффективность восприятия.

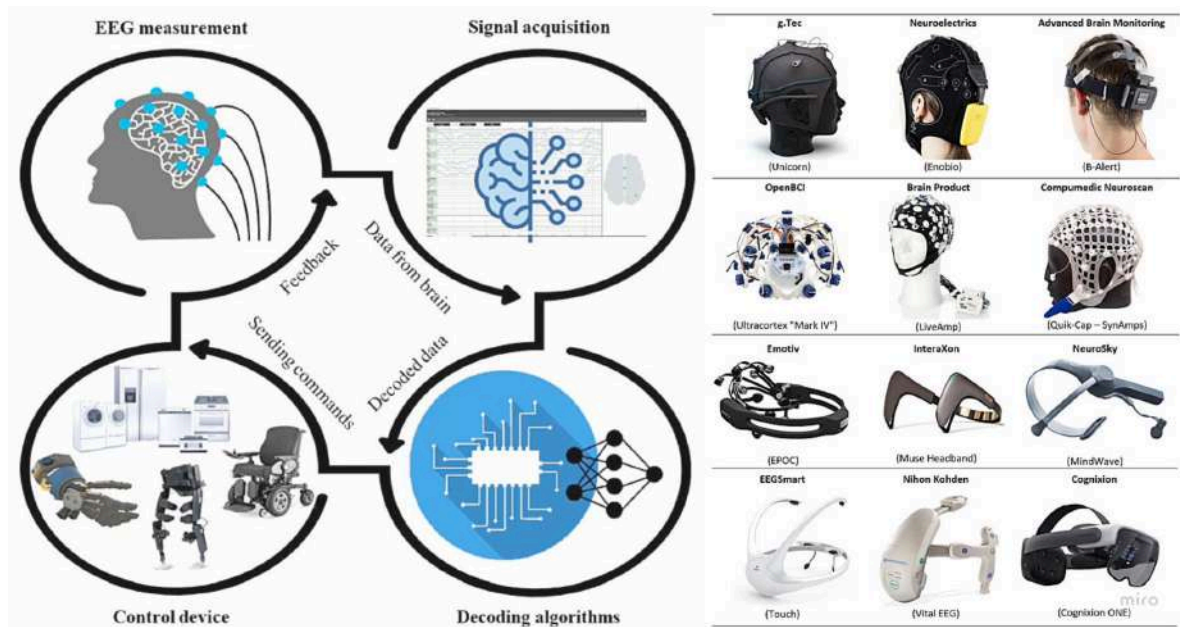


Рисунок 1.6 - Иллюстрация схемы взаимодействия мозг-ПК с применением неинвазивных ЭЭГ датчиков и их примеры

С. ЭМГ активность (активность мускул)

Электромиография (ЭМГ) регистрирует электрическую активность скелетных мышц, возникающую при их сокращении. В контексте тактильного интернета ЭМГ может использоваться как для управления устройствами (например, протезами и экзоскелетами), так и для оценки состояния мышц пользователя в реальном времени. В отличие от ЭЭГ, которая фиксирует активность мозга, ЭМГ даёт информацию о намерениях движения на уровне периферической нервной системы, что делает её более прямой и надёжной для задач моторного управления.

Например, в реабилитационных системах ЭМГ-сигналы с поверхностных электродов, наложенных на мышцы предплечья или руки, позволяют управлять роботизированными протезами с высокой точностью: достаточно лёгкого сокращения мышцы, чтобы устройство распознало намерение и выполнило нужное действие. В сочетании с тактильной обратной связью (вибрацией или давлением) это создаёт замкнутый контур управления, где пользователь не только управляет устройством, но и «чувствует» его работу.

Кроме того, ЭМГ может использоваться для биометрической аутентификации — паттерны мышечной активности уникальны для каждого

человека. Интеграция ЭМГ в системы тактильного интернета позволяет создавать персонализированные интерфейсы, которые адаптируются не только к физиологическому состоянию пользователя, но и к его индивидуальным моторным паттернам, делая взаимодействие более естественным и интуитивным.

1.2.2. Передача обратной связи пользователю

Тактильное взаимодействие - это форма общения или взаимодействия, основанная на ощущениях через касание. Оно включает в себя передачу и восприятие тактильной информации, которая может варьироваться от простых ощущений, таких как давление или текстура, до более сложных форм коммуникации. Тактильное взаимодействие в сетях связи означает возможность передачи ощущений через удаленное взаимодействие с использованием сетевых технологий. С использованием различных устройств и технологий, таких как вибрация или электротактильные устройства, сети связи могут предоставлять обратную связь пользователям при взаимодействии. На сегодняшний день активно используются следующие виды тактильной обратной связи:

Кинестетическая-силовая обратная связь

Кинестетическая-силовая обратная связь представляет собой метод обеспечения взаимодействия пользователя с системой, в которой предоставляются физические силовые ощущения в ответ на действия пользователя. Этот вид обратной связи направлен на предоставление ощущений силы, давления, веса или других физических параметров в реальном времени, чтобы улучшить восприятие и контроль пользователя. Позволяет чувствовать надавливаемые, весовые, импульсные или схожие кинетические силы, направленные на управляемую виртуальную или физическую модель.



Рисунок 1.7 - Пример устройства, реализующего кинестетическую-силовую обратную связь

Для его реализации можно воспользоваться приводами или модулями, которые могут создать механическое сопротивление. К примеру, это сервопривод, шаговый двигатель, электромеханические ткани и т.д.

Тактильно-рецепторная (кожная) обратная связь.

Данный вид обратной связи действует на тактильные рецепторы кожи человека. Такими рецепторами могут быть Тельца Пачини (чувствуют вибрацию), Тельца Мейснера (легкие прикосновения), Клетки Меркеля (давление, грани, углы и точки), Тельца Руффини (растяжение кожи, ощущение тепла) и Колбы Краузе (ощущение холода).

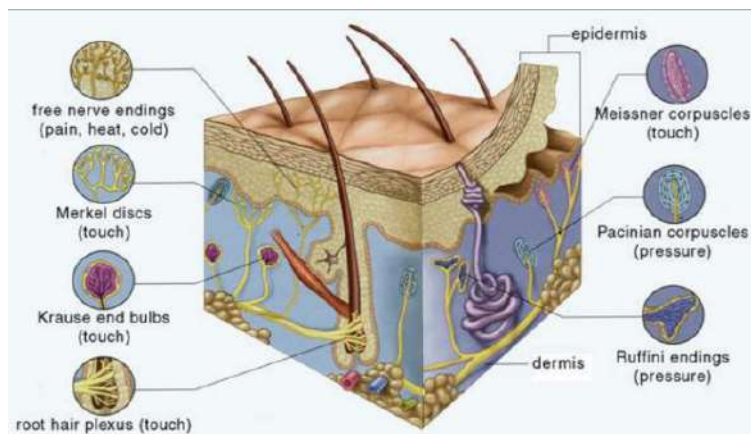


Рисунок 1.8 - Виды кожного воздействия

Для воспроизведения тактильной обратной связи такого вида можно использовать такие модули-актуаторы, как:

- Элементы Пельтье (для генерации тепла и холода - температурной обратной связи)
- Мелкие вибромоторчики (для воспроизведения шероховатости материала - хаптики);
- Специальные мелкие подушки на электропроводимых нитях, при подаче тока которые могут дать эффект давления на кожу.

Болевые и подобные воздействия с применением тактильной-электрической обратной связи на основе ЭМС, ЧЭНЧ и ФЭС

Тактильная-электрическая обратная связь (ТЭОС) представляет собой метод обеспечения обратной связи с пользователем путем применения электрических стимулов для создания тактильных ощущений. В процессе ТЭОС электрические импульсы передаются коже, мышцам или нервам пользователя с целью вызвать определенные тактильные ощущения.

- **ЭМС** (EMS, Electrical muscle stimulation, Миостимулятор или электромиостимулятор)

Устройство для воздействия на мышцы тела с помощью электрических импульсов. К телу прикладываются электроды в непосредственной близости к стимулируемым мышцам. Посылаемые от устройства электрические импульсы похожи на импульсы нервной системы, которые заставляют мышцы сокращаться. Она широко применяется в спортивной медицине для физиотерапевтического воздействия, направленного на восстановление или увеличения функциональных возможностей мышечных и нервных тканей.

- **ЧЭНЧ** (TENS, Transcutaneous electrical nerve stimulation, чрескожная электронейростимуляция).

Высокочастотная ЧЭНЧ активирует механизмы, использующиеся нервной системой для ингибирования боли. Электрические импульсы от электродов, расположенных на коже над очагом боли, стимулируют нервы к блокаде болевых сигналов в направлении мозга, и боль не воспринимается. Низкочастотная ЧЭНЧ стимулирует выброс эндорфинов — естественных ингибиторов боли.

- **ФЭС** (FES, Functional electrical stimulation, Функциональная электростимуляция)

Вид электростимуляции в которой можно стимулировать как двигательные, так и сенсорные нервы. В некоторых случаях нервы стимулируются для генерации локализованной мышечной активности, т.е. стимуляция направлена на создание

прямого мышечного сокращения. В других приложениях стимуляция используется для активации простых или сложных рефлексов. Другими словами, афферентные нервы стимулируются, чтобы вызвать рефлекс, который обычно выражается как скоординированное сокращение одной или нескольких мышц в ответ на стимуляцию сенсорного нерва.

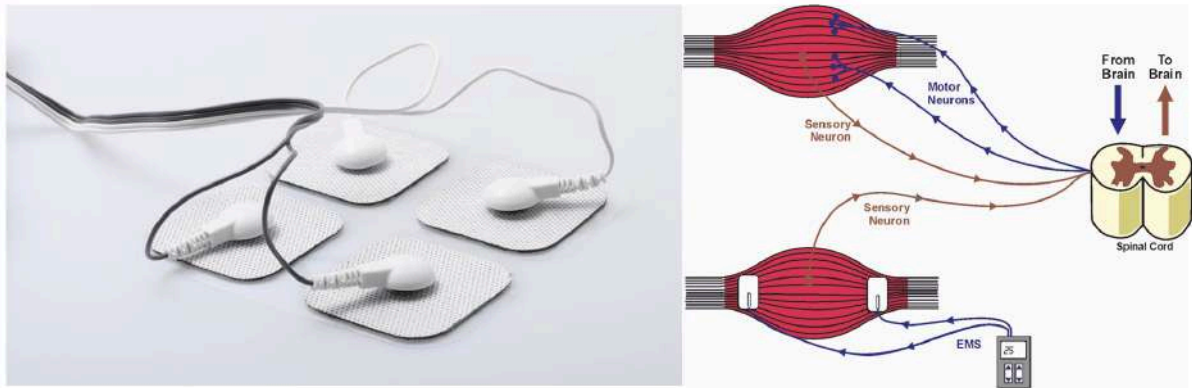


Рисунок 1.9 - Технологии мышечной / кожной стимуляции

Вкус и аромат

- ***Воспроизведение обонятельных данных с применением полушлема с ароматическими картриджами***

Относительно простым способом передачи обонятельной информации является использование устройств, которые распыляют заранее подготовленные ароматические композиции — своего рода «принтеры запахов», встроенные в гарнитуры виртуальной реальности. Такой подход не требует прямого вмешательства в нейронные процессы и опирается на естественные механизмы восприятия: пользователь вдыхает молекулы ароматического вещества, и его обонятельная система сама запускает каскад реакций в мозге.

К примеру, существует полушлем Feelreal — мультисенсорная маска для VR, в которой учтены основные аспекты передачи ароматов. Устройство крепится на любой VR-шлем (Oculus, HTC Vive, PlayStation VR и другие) и синхронизируется с контентом через Bluetooth или Wi-Fi, добавляя к виртуальной реальности не только запахи, но и тактильные ощущения, перепады температур и даже водяной туман.

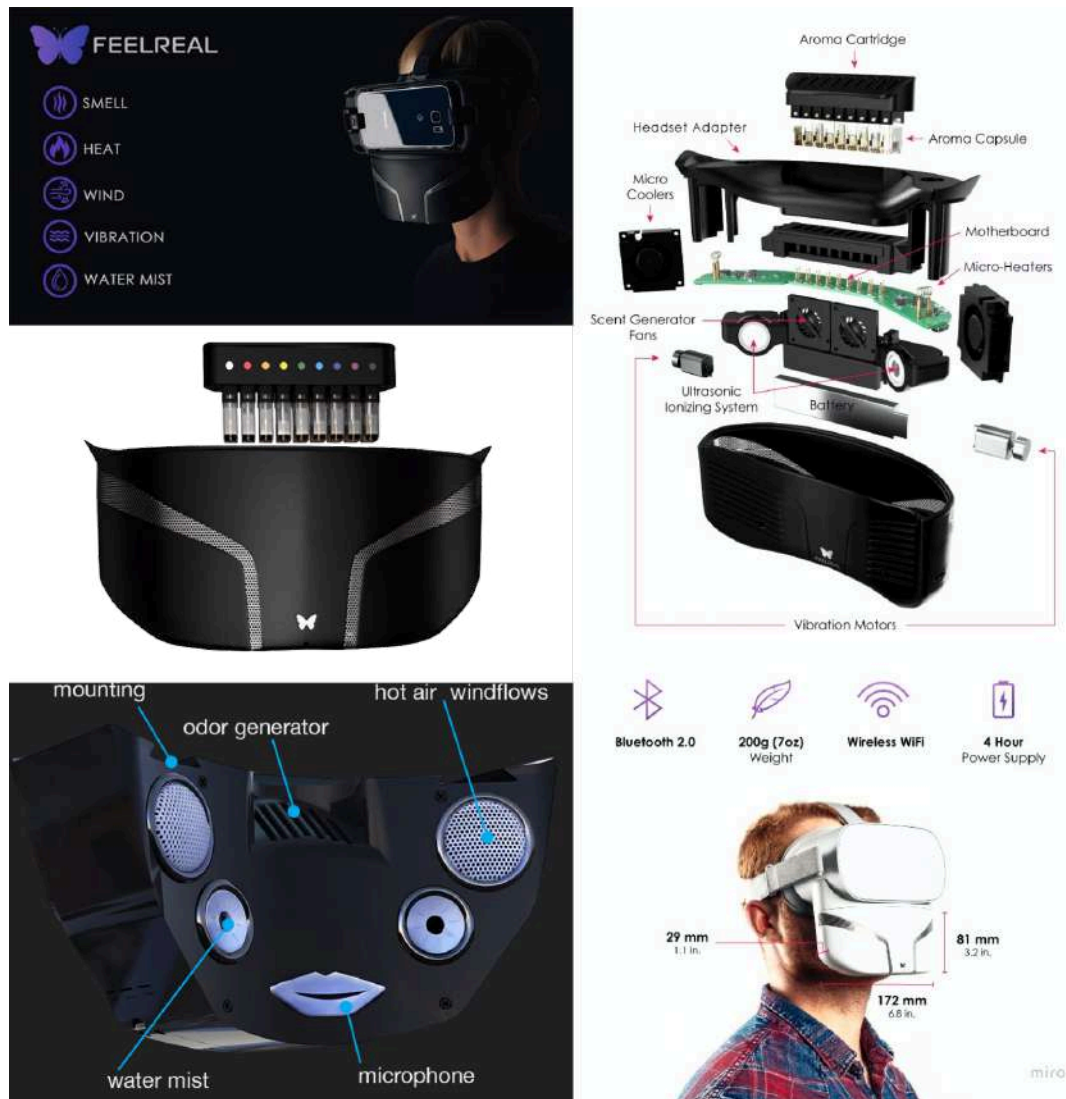


Рисунок 1.10 - Обонятельный полушлем от стартап команды FeelReal

- **Симуляция обонятельных данных с применением спец акустики (ультразвуков) направленной на обонятельную луковицу для активации ароматических рецепторов**

Исследователи впервые применили сфокусированный ультразвук для неинвазивной стимуляции обонятельной луковицы человека. С помощью ультразвукового преобразователя, закреплённого на лбу, они смогли вызвать у испытуемых четыре различных ощущения: свежий воздух, запах мусора, озон и костёр. Эффект был воспроизведён на двух людях и подтверждён в слепом тесте. Авторы предполагают, что такая стимуляция может открыть путь к созданию нового канала передачи информации в мозг — «обонятельного интерфейса» с высокой размерностью (до 400 рецепторных типов), сравнимой с латентными пространствами языковых моделей. В перспективе это позволит «записывать» смысловые векторы напрямую в обонятельную систему, минуя сложную

обработку зрительной или слуховой коры. Работа носит экспериментальный и спекулятивный характер, но демонстрирует принципиальную возможность управления обонятельными ощущениями с помощью ультразвука.

Результаты данного исследования были опубликованы на сайте разработчиков - <https://writetobrain.com/olfactory> (дата обращения: 01.09.2026).

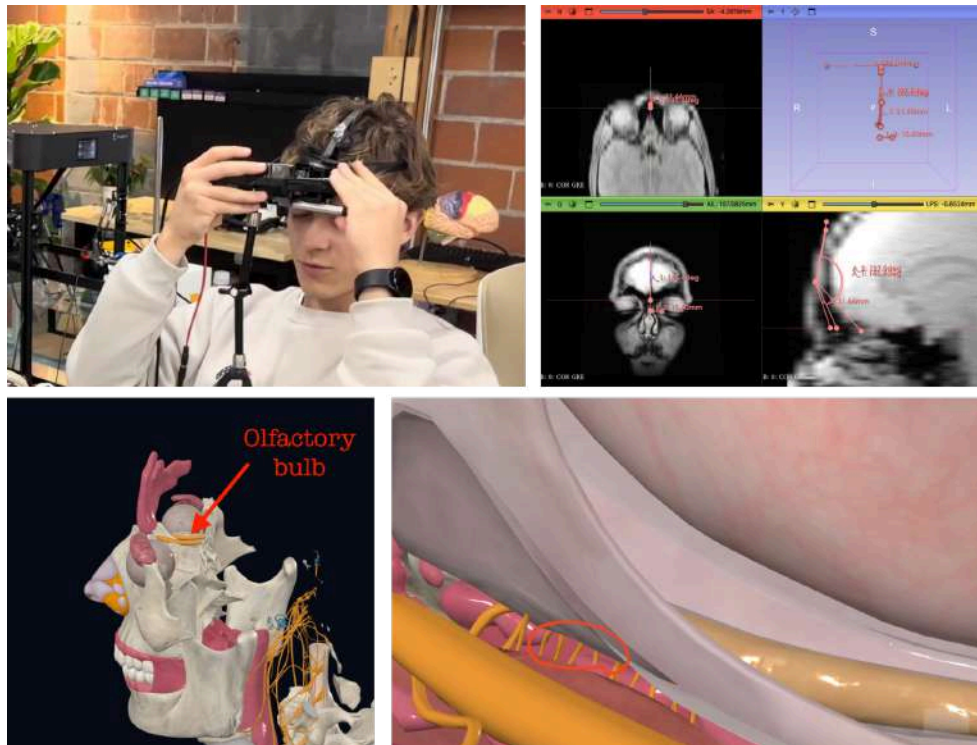


Рисунок 1.11 - Экспериментальная установка с ультразвуковым трансдюсером на голове испытуемого, МРТ и расположение обонятельной луковицы в анатомической 3D-модели головы

- ***Передача вкусовых ощущений: электростимуляция языка***

В отличие от обоняния, где возможны как прямое воздействие на мозг, так и распыление ароматов, передача вкуса требует контакта с рецепторами языка. Наиболее перспективный метод — электростимуляция, имитирующая четыре базовых вкуса (кислый, солёный, горький, сладкий) через изменение частоты и интенсивности сигналов.

К примеру, учёные из Городского университета Гонконга предложили элегантное решение — виртуальный «леденец» размером $8 \times 3 \times 1$ см и весом 15 г, работающий по Bluetooth. Устройство содержит девять каналов с гидрогелями, насыщенными вкусовыми добавками (сахар, соль, лимонная кислота, вишня, маракуйя, зелёный чай, молоко, дуриан, грейпфрут). Принцип ионофореза направляет ионы ароматизаторов через гидрогель на рецепторы языка, а добавление запахов усиливает реалистичность восприятия. Виртуальный «леденец» открывает перспективы для обучения детей, диагностики вкусовых

расстройств и даже дегустации блюд через интернет, хотя пока ограничен часом работы из-за высыхания гидрогелей.

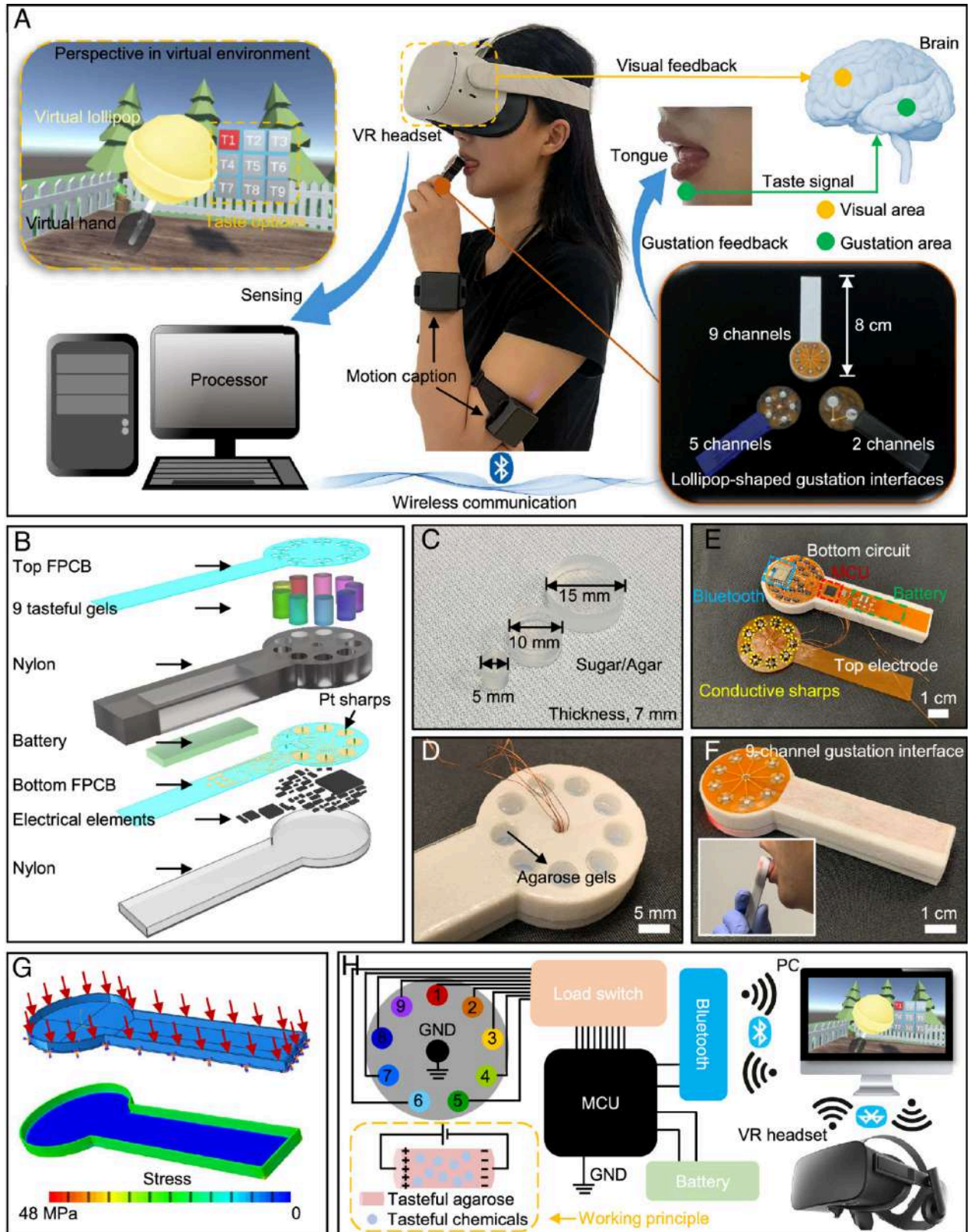


Рисунок 1.12 - Схема передачи вкусовых ощущений в виртуальной среде с использованием ионофорезного вкусового интерфейса

- **Связка «Вкус + Обоняние + Осязание»**

В Тактильном интернете эти органы чувств не работают по отдельности. Чтобы передать, например, «свежеиспеченный круассан или чипсы», система одновременно должна:

- Передать текстуру (хруст) через вибромоторчики или динамики в наушниках (костная проводимость).
- Передать запах масла и теста через диспенсер.
- Сымитировать сладко-соленый вкус через электрическую стимуляцию языка.

Мозг получит все три сигнала с точной синхронизацией по времени (задержка < 1 мс) и «склеит» их в единое ощущение.

1.3. Существующие КТП на момент проведения исследований

Активно развивающаяся область технологии телеприсутствия требует глубокого анализа рынка. Наша цель - не просто превзойти конкурентов, но и предложить более обширный спектр услуг, делая наш костюм телеприсутствия более доступным по цене. В ходе исследования рынка мы стремимся выявить уникальные преимущества, которые позволят нам успешно конкурировать и удовлетворять более широкий спектр потребностей.

1.3.1. TeslaSuit

TeslaSuit представляет собой интегрированный комплексный костюм для виртуальной и дополненной реальности с различными сенсорами и электростимуляторами.

Назначение и применение:

- Виртуальная и дополненная реальность: TeslaSuit разработан для улучшения взаимодействия пользователя с виртуальной и дополненной реальностью. Костюм обеспечивает сенсорную обратную связь и электростимуляцию, чтобы усилить ощущения пользователя в виртуальной среде.
- Обучение и тренинг: Одним из ключевых применений TeslaSuit является использование в сферах обучения и тренинга. Электростимуляция и датчики могут имитировать различные сценарии, что полезно для симуляции реальных ситуаций.

- Медицинское применение: Костюм может использоваться в медицинских областях для виртуального обучения и тренинга медицинского персонала, а также для реабилитации пациентов.



Рисунок 1.13 - Костюм TeslaSuit

Таблица 1.6 - Сравнительная таблица преимуществ и недостатков TeslaSuit

Преимущества	Недостатки
Сенсорная обратная связь: TeslaSuit обеспечивает пользователей физической обратной связью в виртуальной среде.	Сложность в использовании: Некоторые пользователи могут столкнуться с сложностью в использовании и настройке TeslaSuit, особенно если требуется интеграция с другими устройствами и программным обеспечением.
Электростимуляция: Электростимуляция позволяет передавать тактильные ощущения, такие как вибрации и импульсы, что может быть полезным для симуляции различных физических сценариев.	Ограниченность контента: Эффективность TeslaSuit может быть ограничена доступным контентом, и не все виртуальные или дополненные реальности могут поддерживать функциональность костюма.
Сценарии использования: TeslaSuit может использоваться в различных областях, от виртуального обучения до развлекательных приложений и медицинских исследований.	
Интегрированные датчики: Костюм оснащен различными датчиками для отслеживания движений и	Цена: Костюмы виртуальной реальности, включая TeslaSuit, могут

физиологических пользователя, что виртуальный опыт.	параметров дополняет	быть довольно дорогими, что делает их недоступными для широкой аудитории.
---	-------------------------	--

1.3.2. Senso Suit

Senso Suit - это костюм, который позволяет записывать движения, играть в игры в виртуальной реальности и управлять роботами. Senso Device Inc недавно поделились о том, что начали разрабатывать данный костюм. В настоящее время про него мало что известно.



Рисунок 1.14 - Костюм Senso Suit

1.3.3. Perception Neuron 3

Perception Neuron - это система моушн-трекинга, которая обычно включает в себя набор сенсоров и аксессуаров, разработанных для отслеживания движений пользователя в реальном времени. Позволяет создавать анимации, симулировать движения и использовать виртуальную реальность (VR) и другие технологии.

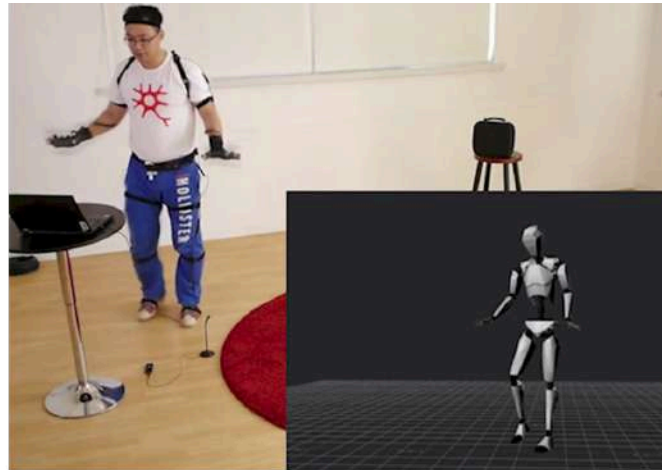


Рисунок 1.15 - Костюм Perception Neuron

Таблица 1.7 - Сравнительная таблица преимуществ и недостатков Perception Neuron

Преимущества	Недостатки
Точность трекинга: Perception Neuron, как правило, известен своей высокой точностью трекинга, что делает его подходящим для создания реалистичных анимаций и симуляций движений.	Стоимость: Технологии моушн-трекинга, включая Perception Neuron, могут быть довольно дорогими, что делает их менее доступными для широкой аудитории.
Легкость и удобство использования: Беспроводные и компактные сенсоры делают Perception Neuron удобным в использовании, не ограничивая движения пользователя.	Ограниченность пространства: В некоторых случаях использование Perception Neuron может быть ограничено размером пространства для трекинга движений.

Реальное время: Perception Neuron предоставляет данные о движениях в реальном времени, что полезно для тех, кто работает с живой анимацией, требующей мгновенного отображения движений.	Не всегда подходит для сложных сценариев: В некоторых сложных сценариях с большим числом дополнительных факторов, таких как объекты, многочисленные движения, система может столкнуться с трудностями.
---	--

1.3.4. OWO Skin haptic suit

OWO Skin haptic suit - футболку с тактильной обратной связью для ощущения пользователем разнообразных полученных повреждений в играх как на приставке, так и в виртуальной реальности.

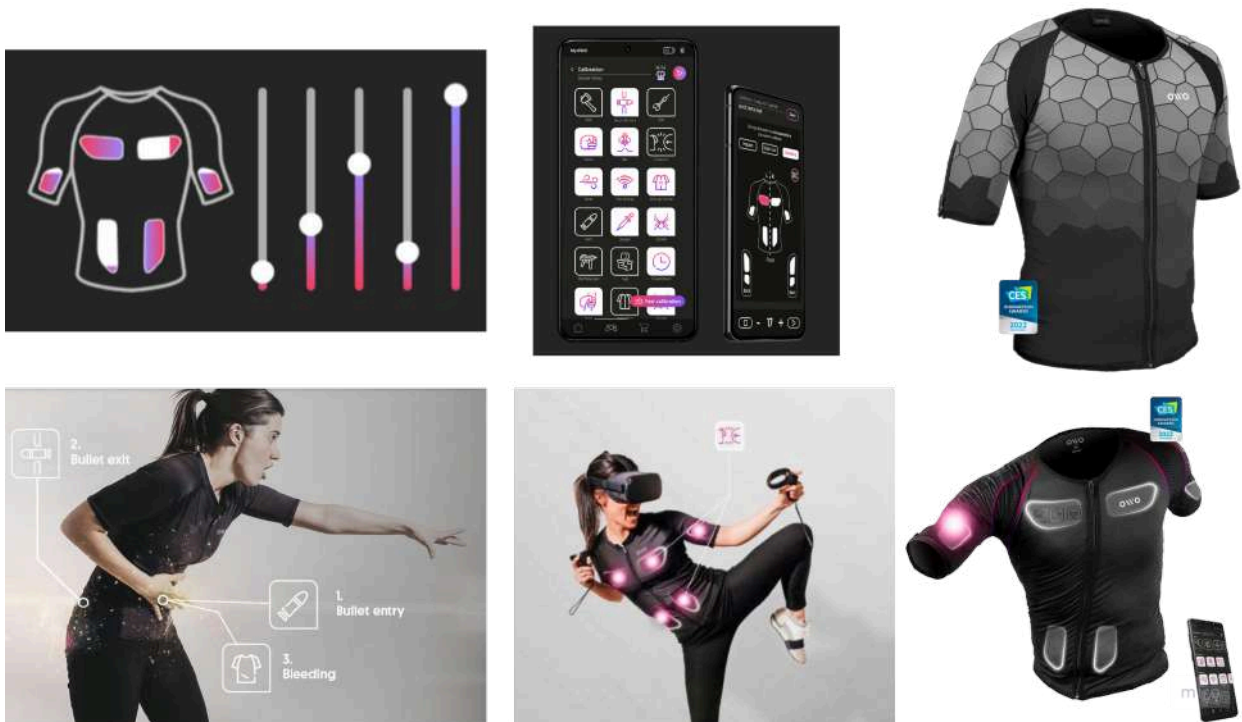


Рисунок 1.16 - OWO Skin haptic suit

Таблица 1.8 - Сравнительная таблица преимуществ и недостатков OWO Skin haptic suit

Преимущества	Недостатки
Точность трекинга: OWO Skin haptic suit обладает высокой точностью трекинга, что делает его идеальным	Стоимость: Себестоимость OWO Skin Haptic Suit высока, что делает

вариантом для полноценного погружения в игры	его недоступным для широкой аудитории
Легкость и удобство использования: Беспроводная футболка делают OWO Skin haptic suit удобным в использовании, не ограничивая движения пользователя.	

1.3.5. HoloSuit

HoloSuit — двунаправленный, беспроводной и простой в использовании костюм для захвата движений всего тела. Благодаря 26 или 36 встроенным датчикам и 9 устройствам тактильной обратной связи, расположенным на обеих руках, ногах и всех десяти пальцах, а также 6 встроенным кнопкам, HoloSuit считывает данные о движении всего тела пользователя и использует тактильную обратную связь для передачи информации обратно. Этот костюм подходит для спорта, здравоохранения, образования, развлечений и промышленных операций.

Костюм XR, который отслеживает и тренирует вас как в реальном, так и в виртуальном мире. Это умный спортивный костюм со встроенными датчиками, которые отслеживают каждое ваше движение с молниеносной скоростью. Можно использовать спортивный костюм как самостоятельно, так и с дисплеем дополненной, виртуальной или смешанной реальности, или с аватаром на внешнем дисплее. Движения пользователя в реальном мире отслеживаются, сохраняются и виртуализируются

Ниже представлены 2D-визуализации трёх аватаров Unity, которые являются примерами аватаров, управляемых с помощью HoloSuit.



Рисунок 1.17 - Костюм HoloSuit.

Таблица 1.9 - Сравнительная таблица преимуществ и недостатков HoloSuit

Преимущества	Недостатки
Точность трекинга: HoloSuit обладает высокой точностью трекинга, что делает его идеальным вариантом для полноценного погружения в игры, медицину, промышленность.	Сложность в использовании: Костюм HoloSuit предполагает единый размер для всего комплекта.
Легкость и удобство использования: Беспроводной костюм HoloSuit удобным в использовании, не ограничивая движения пользователя.	

1.3.6. KOLD

KOLD - созданный в лаборатории MeganetLab 6G «Исследование сетевых технологий с ультра малой задержкой и сверхвысокой плотностью на основе широкого применения искусственного интеллекта для сетей 6G», представляет собой костюм, предназначенный для захвата движений, предоставления тактильных ощущений, передачи кинестетических данных и управления роботами.

Костюм телеприсутствия представляет собой средство для исследования уровней канального, межсетевого, транспортного и приложений, в соответствии с эталонной моделью TCP/IP. Наш костюм обеспечивает эффективный анализ трафика, проведение экспериментов и быструю интеграцию с собственными решениями. Костюм не является конечным продуктом, представляет собой интегрированный аппаратный комплекс, нацеленный на обеспечение взаимодействия и коммуникации в средах виртуальной и дополненной реальности. Данная система включает в себя множество датчиков и сенсоров, формирующих модель концепции интернета навыков, интернета вещей, тактильного интернета. Костюм является интерфейсом взаимодействия с сетевыми вселенными, значительно повышающим уровень иммерсивности в отличие от существующих подходов, использующих только очки/шлемы виртуальной, дополненной и смешанной реальностей



Рисунок 1.18 - Костюм KOLD

Таблица 1.10 - Сравнительная таблица преимуществ и недостатков KOLD

Преимущества	Недостатки
--------------	------------

1. Точность трекинга: KOLD обладает высокой точностью трекинга, что делает его подходящим для создания реалистичных анимаций и симуляций движений.	1. Не всегда подходит для сложных сценариев: В некоторых сложных сценариях с большим числом дополнительных факторов, таких как объекты, многочисленные движения, система может столкнуться с трудностями.
2. Реальное время: KOLD предоставляет данные о движениях в реальном времени, что полезно для тех, кто работает с живой анимацией, требующей мгновенного отображения движений.	2. Ограниченная эргономика: ограниченной свободе движений пользователя, что может оказывать влияние на его комфорт и способность эффективно взаимодействовать с окружающей средой. В некоторых случаях использование KOLD может быть ограничено размером пространства для трекинга движений.
3. Разнообразие применений: Эта технология может использоваться в различных отраслях, включая анимацию, виртуальную и дополненную реальность.	

Таблица 1.11 - Сравнение существующих аналогов костюма телеприсутствия

	Teslasuit	Senso Suit	Perception Neuron 3	OWO Skin haptic suit	KOLD
Интерфейсы	Нет информации	USB, Wi-Fi	USB-передатчик	Bluetooth	WiFi USB
Тип/метод трекинга (Захват движений)	IMU (инерционный) 14 IMU сенсоров для захвата движений (6-осный и 9-осный режим)	15 датчиков IMU (внутреннее) SteamVr 1.0(внешнее)	14 IMU датчиков	Нет информации	3D камера IMU (инерционный) Датчики изгиба/Растяжения/потенциометра
Точность отслеживания частей тела	Roll & Pitch: 0.5°	Нет информации	Roll & Pitch 0.02°	Нет информации	Roll & Pitch 2°
Тактильные датчики	ЭМС (Электромышечная стимуляция) ЧЭНС (Чрескожная электронейростимуляция) ФЭС (Функциональная электростимуляция)	Тактильные датчики отсутствуют	Тактильные датчики отсутствуют	Нет информации	Сервоприводы

Характеристики тактильных актуаторов	Для ЭМС и ФЭС: 90 каналов 1-150 Гц частота 1-60 мА сила тока	Тактильные датчики отсутствуют	Тактильные датчики отсутствуют	Нет информации	ток: 130 мА Крутящий момент: 13 кг·см Скорость вращения: 43 об/мин (360° за 1,4 сек)
Используемые ПО / SDK	Control Center Studio SDK: Unreal Engine и Unity FES Elbow flexion Teslasuit Experience Teslabeat	Senso Glove DK3 Unreal Engine, Unity,	SDK: C/C++ API и плагин Unity3D и Unreal. AXIS Studio.	SDK: C++ и C# Плагины для Unity3D и UnrealEngine OWO app	-
Основная сфера применения	Медицина и расширенная реальность	Игры в виртуальной реальности, захват движений	Трансляция на Twitch или полная анимация тела для фильма/игры	Для игр	Медицина, захват движений, для взаимодействия с сетевыми вселенными
Автономность (время работы)	Нет информации	Нет информации	5 часов	8 часов	1 час
Название и страна компании-производителя	Tesla Suit, Великобритания	Senso Devices Inc. США	Noitom, Китай	OWO, Испания	СПбГУТ, Российская Федерация
Главные возможности	Тактильные ощущения. Биомеханический анализ. Захват движения. Реабилитация пациентов	Захват движений Игры в виртуальной реальности. Управление роботом	Захват движений	Для тактильных ощущений в играх	Тактильные ощущения, Захват движений, Реабилитация пациентов, интерфейс для работы в сетевой вселенной

1.4. Требования и проблемы реализации иммерсивных услуг

В то время как современные сети успешно справляются с передачей данных между людьми (голосовые и видеозвонки) и между машинами (интернет вещей и промышленная автоматизация) [57-62], тактильный интернет (ТИ) знаменует собой появление новой парадигмы — взаимодействия человека и машины (H2M)[50-53]. Эта новая модель позволяет пользователю не просто видеть и слышать удаленную

среду, но и осязать ее, а также манипулировать объектами в реальном времени с высокой точностью[48].

Помимо тактильного взаимодействия, ТІ является ключевым драйвером развития иммерсивных коммуникаций, где мультисенсорный опыт — такой как прикосновение, движение и пространственное восприятие — бесшовно интегрируется с цифровыми средами. Иммерсивные коммуникации станут отличительной чертой эпохи 6G [63,64], расширяя возможности сред виртуальной, дополненной и смешанной реальности (VR/AR/MR), миров метавселенных и систем телеуправления роботами.

Подобные приложения предъявляют требования не только к сверхнизкой задержке и высокой надежности, но и к интеллектуальной адаптивности сети и динамическому распределению ресурсов для обеспечения бесперебойной работы пользовательских интерфейсов [65-76]. Достижение этих целей требует применения передовых сетевых технологий, включая программно-конфигурируемые сети (SDN) [89] и периферийный интеллект (EI), а также решений на основе искусственного интеллекта.

Несмотря на огромный потенциал, ТІ и иммерсивные коммуникации сталкиваются с рядом проблем, сдерживающих их широкое внедрение и масштабирование. К ключевым научно-исследовательским задачам можно отнести следующие:

- **Сверхнизкая задержка и надежность.** Достижение задержки менее одной миллисекунды и сверхвысокой надежности (на уровне 99,99999%) является одной из главных инженерных проблем.
- **Адаптивность и гибкость сети.** Традиционные архитектуры сетей не способны динамически адаптироваться для решения задач распределения ресурсов и принятия решений в реальном времени.
- **Вынос вычислений и интеллект на периферию.** Приближение облачных вычислений к конечным устройствам создает проблему дополнительной задержки. В связи с этим, трендом является развитие периферийного интеллекта (EI), который обеспечивает обработку данных на основе ИИ в реальном времени.
- **Ресурсоемкая оркестрация сети.** Для создания масштабируемой и адаптивной сетевой архитектуры требуется бесшовная интеграция SDN, многоуровневых периферийных вычислений (MEC), виртуализации сетевых функций (NFV) и оптимизаций на основе ИИ.
- **Безопасность и доверие.** Внедрение тактильных коммуникаций и систем удаленного управления порождает новые риски для безопасности, что

потребуется создания инфраструктур защиты с использованием блокчейн-технологий и устойчивых к атакам алгоритмов ИИ.

В связи с этим, данная работа представляет всесторонний обзор ключевых технологий, обеспечивающих работу ТИ и иммерсивных коммуникаций, с особым акцентом на SDN и EI. Основные вклады этой работы включают:

- Глубокий анализ эволюции ТИ и иммерсивных коммуникаций с акцентом на их значимость в сетях 6G.
- Комплексный обзор трудностей, связанных с достижением сверхнизкой задержки, высокой надежности и динамической реакции, особенно для приложений тактильного взаимодействия в реальном времени.
- Обзор поддерживающих технологий, таких как MEC, NFV, SDN и оптимизация на основе ИИ, с особым вниманием к SDN и EI как ключевым факторам.
- Сравнительный анализ ведущих научных работ, оценку современного состояния исследований, их ограничений и нерешенных проблем.
- Обзор будущих направлений исследований, ориентированных на создание интеллектуальных, масштабируемых и безопасных сетевых структур для развития ТИ и иммерсивных коммуникаций.

1.5. Проблемы проектирования тактильного интернета и иммерсивных коммуникаций в 6G

Эволюция тактильного интернета и иммерсивных коммуникаций в 6G представляет собой фундаментальный сдвиг парадигмы в телекоммуникационных сетях, направленный на обеспечение управления в реальном времени, сенсорного взаимодействия и иммерсивного опыта с ультравысокой надежностью и скоростью отклика [40-44]. Однако их реализация и проектирование сталкиваются с огромными трудностями, обусловленными жесткими требованиями этих приложений: субмиллисекундной задержкой, экстремальной надежностью, передовыми механизмами безопасности и бесшовной интеграцией с новыми технологиями, такими как дополненная и виртуальная реальность, цифровые двойники и интеллектуальная автоматизация [38, 39].

Сложность разработки систем, поддерживающих ТИ и иммерсивные коммуникации 6G, заключается в необходимости соблюдения чрезвычайно жесткого набора требований к производительности: круглая задержка менее 1 мс, надежность выше 99,99999%, детерминированный и синхронизированный отклик, масштабируемость, высокая спектральная и энергетическая эффективность, а

также надежные гарантии безопасности в динамических и ресурсно-ограниченных средах. На рисунке 1.19 показана потенциальная уровневая структура систем ТИ с перечислением поддерживающих технологий и основных требований для каждого уровня.

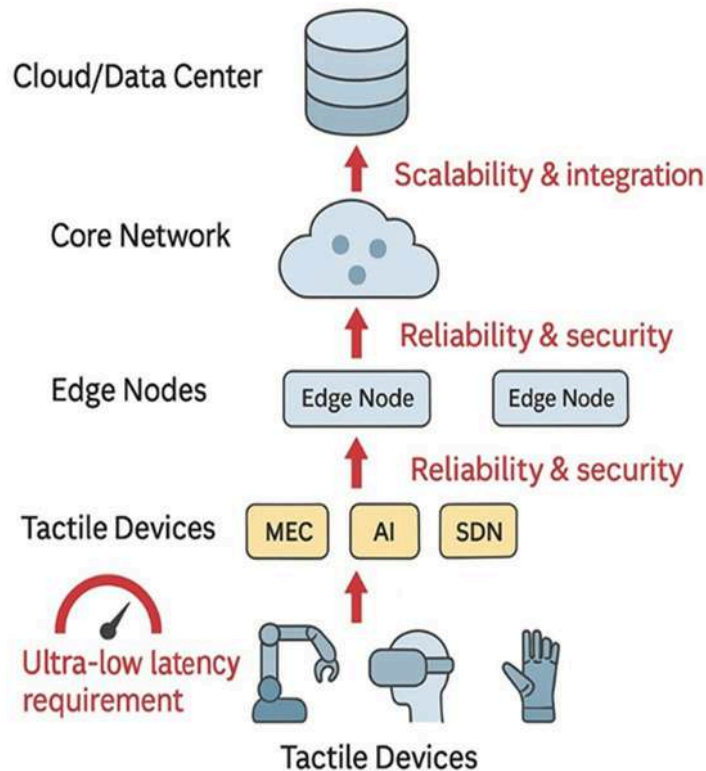


Рисунок 1.19 - Потенциальная уровневая структура систем ТИ

В основе всех этих проблем лежит необходимость полной архитектурной трансформации. ТИ и иммерсивные системы — это не просто усовершенствование существующих широкополосных сетей, а революционный сдвиг, предполагающий двустороннюю тактильную обратную связь, управление с участием человека и удаленное взаимодействие в реальном времени, что требует новых архитектурных и протокольных решений. Одной из самых больших технологических проблем является достижение сквозной задержки менее одной миллисекунды, что значительно ниже, чем у первых реализаций 5G. Этот целевой показатель охватывает время от кодирования сигнала до его передачи, обработки, исполнения и возврата обратной связи. Практически это требует инновационных стеков протоколов, разгрузки вычислений на периферию (МЕС) и архитектуры маршрутизации с минимальным числом промежуточных узлов.

Тесно связанная с задержкой проблема надежности должна достигать сверхвысоких значений, чтобы гарантировать безошибочную передачу сигналов управления и тактильной обратной связи. Приложения, такие как удаленная хирургия или автономный транспорт, требуют вероятности ошибки на пакет менее

10^{-7} . Столь высокая надежность необходима, чтобы отказ даже одного пакета не привел к катастрофическим последствиям, особенно в системах тактильной обратной связи, где потеря данных ведет к неверной или запаздывающей физической реакции. Существующие протоколы передачи данных, такие как TCP (с задержками повторной передачи) и UDP (без гарантий надежности), по своей природе недостаточны. В связи с этим ведутся исследования новых механизмов транспортного уровня, таких как детерминированные сети (DetNet) и сети с чувствительным ко времени (TSN).

Безопасность является еще одной критической проблемой, которая становится все более сложной в гетерогенных, распределенных и ограниченных по задержке средах. В отличие от традиционных интернет-приложений, TI требует сквозного безопасного обмена данными в реальном времени через децентрализованные и гетерогенные сети. В то время как традиционные модели безопасности, такие как IPSec, обеспечивают надежную защиту для стандартных приложений, они неприемлемы для тактильных и иммерсивных систем из-за создаваемой ими вычислительной задержки. Приложения AR, VR и телеуправления требуют облегченных архитектур безопасности, которые не вносят дополнительной задержки в процессы обработки и передачи данных. Кроме того, гетерогенность устройств — от маломощных тактильных актуаторов до пограничных узлов с ИИ — требует многоуровневой облегченной архитектуры безопасности, которая может динамически настраиваться в зависимости от контекста взаимодействия (особенно в сценариях XR). Поэтому необходимо исследовать новые алгоритмы шифрования, архитектуры с нулевым доверием и парадигмы безопасных вычислений, чувствительные к задержке.

Концепция перцепционной прозрачности добавляет еще один уровень сложности. В приложениях TI качество пользовательского опыта определяется не столько сырыми показателями задержки или надежности, сколько плавностью имитации естественного взаимодействия. Это требует сверхточной координации между ведущим и ведомым устройствами, чтобы оператор воспринимал полный и непрерывный цикл взаимодействия. Любое отклонение, задержка или дрожание сигнала разрушают иллюзию прямого контроля и снижают иммерсивность. Беспроблемная системная интеграция также имеет решающее значение для успешного внедрения TI. Необходимо гармонизировать платформы интернета вещей, унаследованные сети, облачные сервисы и сети мобильной связи для удовлетворения уникальных требований TI. При этом системы должны поддерживать как текущие, так и будущие приложения, основанные на тактильной связи и сенсорной интеграции, что требует учета обратной совместимости, стандартизации и архитектурной согласованности.

Проектирование тактильных устройств является еще одним узким местом. Эти устройства, обеспечивающие обратную связь для осязания и управления, характеризуются своими степенями свободы. Для удовлетворения требований иммерсивных коммуникаций будущие тактильные устройства должны обеспечивать более сложные взаимодействия, повышенную точность и более быстрое время отклика. Это стимулирует развитие технологий актуаторов и сенсоров, а также способность таких устройств работать в условиях ограниченных и изменяющихся во времени сетей.

Что касается кодирования и декодирования данных, приложения TI принципиально отличаются от аудиовизуальной коммуникации. Тактильная связь по своей природе является двунаправленной и состоит из нескольких одновременных потоков данных (сила, скорость, положение, момент силы), что порождает большие объемы данных и требует энергоэффективных методов кодирования. Сетевое кодирование, особенно случайное линейное сетевое кодирование (RLNC), является перспективным решением. В отличие от блочных систем, RLNC использует скользящее окно и поддерживает повторное кодирование внутри сети, что повышает гибкость и надежность. Кроме того, рассматриваются алгоритмы сжатия с потерями в реальном времени на основе дискретного косинусного преобразования (DCT) или вейвлет-преобразования (WT) для минимизации размера полезной нагрузки при сохранении перцепционного качества.

Маршрутизация в TI представляет собой особую задачу. По сравнению с традиционными IP-сетями, где допускается вариативность задержки, TI требует постоянного и детерминированного бюджета задержки. Пакеты должны передаваться за время, не превышающее 33 мкс, что является значительно более жестким требованием по сравнению с длительностью символа OFDM в LTE или 5G. Это требует разработки новых схем маршрутизации, способных реагировать сверхбыстро, минимизировать число переходов и эффективно использовать периферийные ресурсы. Размещение узлов MEC рядом с конечным оборудованием значительно сокращает коммуникационный путь и задержку. Однако вопросы оркестрации и управления мобильностью между этими узлами остаются открытыми для исследований.

На следующей таблице приводится сводка по описанным выше вызовам при разработке TI и иммерсивных коммуникаций 6G.

Таблица 1.12 - Основные проблемы тактильного интернета и иммерсивных коммуникаций

Проблема	Описание	Существующее ограничение	Необходимые усовершенствования
Сквозная задержка	Круговая задержка 1 мс или менее	Текущая задержка 5G > 5 мс	MEC, SDN, интеллектуальное планирование
Сверхвысокая надежность	Вероятность ошибки на пакет < 10^{-7}	Протоколы UDP/TCP неадекватны	Новые протоколы, DetNet, TSN
Безопасность	Многоуровневые, чувствительные к задержке и контексту структуры безопасности	IPSec слишком тяжеловесен для тактильной связи в реальном времени	Облегченная криптография, обнаружение угроз на основе ИИ
Перцепционная прозрачность	Бесшовное и реалистичное взаимодействие для пользователя	Высокий джиттер, плохая синхронизация	Контур обратной связи в реальном времени, точная синхронизация
Проектирование тактильных устройств	Увеличение степеней свободы, улучшение разрешения обратной связи	Ограниченная точность, низкая отзывчивость	Усовершенствованные датчики, тактильный рендеринг на основе ИИ
Кодирование и декодирование	Эффективное сжатие и помехоустойчивая передача двунаправленных тактильных сигналов	Большая полезная нагрузка, задержки повторной передачи	RLNC, сжатие в реальном времени (DCT/WT)
Маршрутизация и мобильность	Сверхбыстрые адаптивные пути	IP-маршрутизация слишком медленная	Маршрутизация с учетом периферии,

	маршрутизации, оптимизированные по задержке и надежности		адаптация динамической топологии
Системная интеграция	Интероперабельность с 5G, IoT и облачной инфраструктурой	Фрагментированные стандарты, устаревшие системы	Унифицированная архитектура, оркестрация на основе ИИ

1.6. Анализ методов и подходов к реализации иммерсивных услуг

В данном разделе рассматриваются ключевые технологии, лежащие в основе ТІ и иммерсивных коммуникаций, с особым акцентом на использование программно-конфигурируемых сетей (SDN) и периферийного интеллекта (ЕІ) как основных инструментов. SDN повышает программируемость, адаптивность и динамический контроль сетевой инфраструктуры. В то же время ЕІ использует интеллект на основе искусственного интеллекта для принятия решений на границе сети с целью оптимизации задержки, использования ресурсов и качества предоставления услуг. В работе также описываются другие поддерживающие технологии, включая виртуализацию сетевых функций (NFV), цифровые двойники, квантовые вычисления и блокчейн.

1.6.1. Подходы к удовлетворению требований к задержке.

ТІ требует сквозной задержки в одну миллисекунду, включая время на получение сигнала обратной связи. Развертывание МЕС на границе радиосети сокращает задержку связи, однако само по себе не позволяет достичь требуемого рубежа в 1 мс. Основной причиной являются фундаментальные физические ограничения, связанные со скоростью света. Для задержки в 1 мс сигнал может пройти максимум 300 км, а с учетом двусторонней связи это расстояние сокращается до 150 км. Принимая во внимание другие задержки, с использованием МЕС эффективное расстояние сокращается до 15 км. Для приложений ТІ и иммерсивных коммуникаций, предполагающих связь между крупными городами и странами, этого расстояния недостаточно. Периферийный интеллект (ЕІ) предлагает решение этой проблемы. Внедрение ЕІ позволяет достичь требуемой субмиллисекундной задержки за счет построения модели

удаленной среды, которая может прогнозировать сигнал обратной связи и имитировать реакцию удаленной стороны. Эта модель называется моделью-посредником. Используя алгоритмы ИИ, она предсказывает сигналы обратной связи, избегая времени ожидания реального ответа. Модель периодически обновляется на основе реальных полученных сигналов.

1.6.2. Роль интеллектуальных граничных вычислений и стандартные компоненты.

Граничный интеллект (EI) представляет собой интеграцию ИИ [45-48] и периферийных вычислений (например, MEC), обеспечивая обработку данных, принятие интеллектуальных решений и автономность в сетях 6G в реальном времени. В отличие от облачных конфигураций, EI минимизирует задержку, риски утечки данных и оптимизирует пропускную способность за счет обработки данных непосредственно в точке их генерации. EI может использоваться в сетях ТИ и иммерсивных коммуникаций для динамического распределения ресурсов, прогнозного управления трафиком и автоматизированной оркестрации сервисов. В частности, EI способствует снижению сквозной задержки, обеспечивает рендеринг в реальном времени и персонализацию контента. Локальная обработка данных минимизирует задержку между действиями пользователя и откликом системы, что крайне важно для бесшовного опыта. Кроме того, периферийные узлы могут брать на себя рендеринг сложных сцен AR/VR, снижая нагрузку на устройства пользователей, а алгоритмы ИИ на периферии могут адаптировать контент в реальном времени на основе предпочтений и поведения пользователя.

Традиционные модели глубокого обучения являются слишком ресурсоемкими для эффективного развертывания на периферии. Использование облегченных моделей ИИ, федеративного обучения и сотрудничества между периферией и облаком позволяет EI преодолеть технические ограничения. Облегченные нейронные сети, такие как MobileNet, SqueezeNet и TinyML, разработаны для эффективной работы на периферийных устройствах с ограниченными ресурсами. Однако внедрение EI для сетей с MEC также вносит проблемы, связанные с вычислительной нагрузкой на инференс моделей ИИ на периферии, ограничениями по обработке данных в реальном времени и масштабируемостью моделей в крупных сетях [77-79].

Таблица 1.13 - Проблемы развертывания периферийного интеллекта (EI) для сетей ТИ и иммерсивных коммуникаций

Проблема	Технические соображения
----------	-------------------------

Ограниченность ресурсов	Ограниченная вычислительная мощность периферийных узлов требует методов оптимизации моделей, таких как прореживание и квантизация.
Безопасность и конфиденциальность	Дифференциальная конфиденциальность и гомоморфное шифрование снижают риски, связанные с децентрализованным ИИ.
Масштабируемость	Динамическое распределение ресурсов и SDN обеспечивают эффективную масштабируемость.
Энергоэффективность	Сжатие моделей ИИ и планирование с учетом энергопотребления оптимизируют расход энергии.
Гетерогенность оборудования	Периферийные устройства различаются по вычислительным возможностям, что требует адаптивных моделей ИИ.
Синхронизация данных	Обновления в реальном времени должны синхронизироваться между распределенными периферийными узлами для обеспечения согласованности.

В таблице 1.14 перечислены стандартные ключевые компоненты платформ EI. Оркестрация на периферии является основной частью платформы EI, включающей управление развертыванием, масштабированием и миграцией периферийных сервисов. Она состоит из нескольких ключевых компонентов: модуль размещения сервисов (определяет место развертывания в зависимости от спроса и доступности ресурсов), балансировщик нагрузки (распределяет нагрузку для предотвращения перегрузок) и модуль отказоустойчивости (обеспечивает непрерывность работы благодаря избыточности и механизмам самовосстановления) [80-83].

Таблица 1.14 - Ключевые компоненты платформ периферийного интеллекта

Компонент	Описание
Периферийные узлы	Устройства или мини-центры обработки данных на периферии сети, которые выполняют модели ИИ локально.

Модель ИИ/МО	Гибридная структура ИИ в 6G, объединяющая обучение с подкреплением, глубокое обучение и графовые нейронные сети (GNN) для оптимального управления сетью и принятия решений.
Федеративное обучение	Децентрализованный подход к обучению ИИ, который позволяет обучать модели без централизованной агрегации данных.
Оркестрация периферии	Программное обеспечение, управляющее развертыванием, масштабированием и миграцией периферийных сервисов.
Цифровые двойники	Виртуальные копии сетевых объектов, используемые для прогнозного анализа и оптимизации в реальном времени.
Инфраструктура 5G/6G	Высокоскоростное соединение с низкой задержкой, поддерживающее обработку ИИ на периферии в реальном времени.
Сети IoT и датчиков	Источники данных, предоставляющие входные данные в реальном времени для принятия решений на основе ИИ.
Нейроморфные вычисления	Микросхемы ИИ, вдохновленные мозгом, предназначенные для энергоэффективного вывода на периферии.
Аппаратное ускорение ИИ	Специализированные процессоры, такие как TPU и GPU, для оптимизированных вычислений ИИ.

1.6.3. Пограничные и облачные вычисления

Хотя ЕІ делает акцент на локальной обработке, необходимость сотрудничества с облаком сохраняется для решения задач, требующих глобального контекста или выполнения крайне ресурсоемких операций. Это сотрудничество реализуется через разгрузку задач (offloading), федеративное обучение и периферийное кэширование. Вычислительно сложные задачи могут разгружаться в облако, в то время как задачи, критичные к задержке, остаются на периферии. Федеративное обучение — это распределенная парадигма машинного обучения, в которой периферийные устройства совместно обучают глобальную модель без обмена исходными данными. Периферийное кэширование используется для хранения часто запрашиваемых данных на границе сети, что снижает задержку и потребление пропускной способности.

1.6.4. Применение ЕІ для удовлетворения требований к качеству обслуживания (QoS/QoE)

Граничные вычисления может применяться для динамической оптимизации параметров сети для достижения требуемого уровня QoS и QoE в иммерсивных приложениях. Это включает использование обучения с подкреплением для прогнозирования трафика и динамической балансировки нагрузки, а также графовых нейронных сетей для повышения эффективности маршрутизации в сложных топологиях. Мультиагентные системы ИИ могут применяться для распределенного принятия решений, повышая устойчивость сетей. Все это способствует достижению ключевых показателей эффективности (KPI), необходимых для ТІ и иммерсивных коммуникаций 6G [84-88].

В таблице 1.15 приведены решения на базе граничных вычислений для различных проблем, связанных с достижением требуемых KPI для ТІ и иммерсивных коммуникаций. В таблице 1.16 представлены решения ЕІ для различных гетерогенных проблем, возникающих при разработке ТІ и иммерсивных коммуникаций.

Таблица 1.15 - Решения на базе распределенного интеллекта для достижения требуемых KPI в ТІ и иммерсивных коммуникациях 6G

KPI	Проблема	Решение	Обсуждение
Повышение эффективности и задержки	Достижение субмиллисекундной задержки для тактильной обратной связи.	Граничные кэширование	Предварительная загрузка данных и моделей на периферию для сокращения времени обработки.
		Облегченные модели ИИ	Использование компактных нейронных сетей, которые могут эффективно работать на периферийных устройствах.
		Сетевая нарезка (Network Slicing)	Выделение выделенных сетевых ресурсов для приложений ТІ.
Надежность и отказоустойчивость	Обеспечение непрерывной работы в динамических средах.	Избыточные граничные узлы	Развертывание нескольких периферийных узлов для обеспечения возможности восстановления после сбоев.
		Самовосстанавливающиеся алгоритмы	Алгоритмы ИИ, которые могут автономно обнаруживать и восстанавливаться после сбоев.
Безопасность и конфиденциальность	Защита конфиденциальных данных от несанкционированного доступа.	Шифрование на периферии	Шифрование данных на периферии перед их передачей.
		Хаптические кодеки	Кодеки, разработанные для процедур кодирования и декодирования при передаче и реконструкции тактильной

			и кинестетической обратной связи по сетям связи.
		Федеративное обучение	Обучение моделей ИИ без обмена исходными данными.

Таблица 1.16 - Решения на базе граничного интеллекта для критических проблем
ТІ и иммерсивных коммуникаций 6G

Проблема	Обоснование проблемы	Решение	Обсуждение
Требования к пропускной способности и вычислительным ресурсам	Приложения AR/VR требуют высокой пропускной способности и вычислительных ресурсов.	Периферийный рендеринг	Разгрузка задач рендеринга на периферийные серверы снижает вычислительную нагрузку на устройства пользователей.
		Сети доставки контента (CDN)	Распределение контента по нескольким периферийным узлам для снижения задержки и использования пропускной способности.
Мобильность пользователя	Обеспечение стабильной производительности при перемещении пользователей между различными сетевыми зонами.	Периферийный хэндовер	Бесшовная передача сессий между периферийными узлами при перемещении пользователей.
		Прогнозное кэширование	Предвидение перемещения пользователя и предварительная загрузка контента на ближайший периферийный узел.
Энергоэффективность	Увеличение времени автономной работы устройств AR/VR.	Энергоэффективные модели ИИ	Использование облегченных моделей, потребляющих меньше энергии.
		Периферийная разгрузка	Снижение вычислительной нагрузки на пользовательские устройства путем разгрузки задач на периферийные серверы.

1.7. Вывод по главе

Приведенные примеры в текущей главе служат мотивацией и вдохновением на то, что хоть и с неким трудом, можно реализовать полноценные системы с иммерсивными устройствами и сервисами для метавселенных и тактильного интернета.

Также, рассмотренные методы улучшения сетевого взаимодействия, в том числе, применение ИИ и граничных вычислений, позволяют разработать

тактильные перчатки и КТП, которые смогут соответствовать требованиям и стандартам по тактильному интернету и метавселенных.

Проведённый анализ в данной главе, служит теоретико-методологической базой для последующей разработки собственного КТП, позволяя целенаправленно преодолевать выявленные недостатки существующих аналогов за счёт применения вышеописанных сетевых решений, оптимизации протоколов маршрутизации и интеграции расширенного набора телеметрических датчиков для адаптивной персонализации тактильных интерфейсов.

ГЛАВА 2. АДАПТИВНЫЙ МЕТОД ПЕРЕДАЧИ ДАННЫХ С ПОВЫШЕНИЕМ ЭФФЕКТИВНОСТИ КАНАЛА ДЛЯ ТЕРМОДАТЧИКОВ КОСТЮМА ТЕЛЕПРИСУТСТВИЯ (КТП)

Развитие технологий связи и вычислительной техники, как основных составляющих современной инфокоммуникационной системы, ведет к появлению новых услуг, в том числе услуг виртуальной и дополненной реальности, а также услуг телеприсутствия. В направлениях развития инфокоммуникационной системы появилось создание метавселенной, т. е. виртуальной среды для создания моделей и объектов как реального, так и виртуального мира. Это направление, находящееся пока на начальной стадии развития, может иметь большое значение в будущем, т. к. позволит объединить самые различные области знаний и приложений.

В настоящее время ведутся работы по созданию средств взаимодействия с метавселенной, на данном этапе сама метавселенная может быть заменена некой вычислительной системой, позволяющей реализовать необходимую функциональность. Как правило, такие средства взаимодействия представляют собой пользовательские устройства (терминалы), имеющие бóльшую функциональность, чем традиционная клавиатура, мышь или джойстик. Например, это различные устройства виртуальной и /или дополненной реальности, всевозможные манипуляторы и другие устройства. Основными идеями создания таких устройств являются идеи погружения пользователя в виртуальный мир (услуга виртуальной реальности) или виртуальный перенос пользователя в иное место (услуга телеприсутствия). Степень погружения пользователя (иммерсивность), т. е. степень чувственного ощущения реальности происходящего характеризует качество услуги и во многом зависит от окончного оборудования.

В настоящее время подобные устройства взаимодействуют с человеком через органы его чувств и двигательную систему. В частности, костюм телеприсутствия (КТП) является таким устройством. КТП совмещает в себе множество сенсоров

(датчиков) и актуаторов (воздействующих устройств). Сенсорами могут быть датчики угла, ускорения, положения в пространстве, давления, температуры и многие другие. Актуаторами – различного типа двигатели, манипуляторы, нагреватели (охладители) и другие устройства.

КТП – это устройство, которое позволяет получить данные о состоянии субъекта и передать ее на прикладную платформу. Также КТП позволяет получить информацию от прикладной платформы и предоставить ее субъекту. В частности, КТП оборудован датчиками температуры для получения тактильной информации и информации об окружающей среде, а также актуаторами – элементами Пельтье, позволяющими реализовать представления информации о температуре субъекту. Потенциально, количество таких датчиков может быть значительным, что ведет к необходимости передачи достаточно больших объемов данных, а с учетом требований к иммерсивности передача данных должна осуществляться достаточно быстро.

В данной работе рассматривается задача передачи информации тактильных ощущений, а именно о температуре.

2.1. Модель костюма телеприсутствия

КТП содержит множество сенсоров и актуаторов. Сенсоры служат для получения информации о тактильных ощущениях и о состоянии КТП (угол поворота, нажатие (давление), температура, ускорение, ориентация в пространстве). Актуаторы служат для представления информации субъекту путем создания тактильных ощущений и кинетических воздействий. Информация, получаемая от сенсоров, как правило, представляет собой аналоговый сигнал, величина которого изменяется в зависимости от степени воздействия на сенсор. Получаемый аналоговый сигнал должен пройти операцию аналого-цифрового преобразования (АЦП). После этого данные, содержащие информацию о сигнале, передаются на прикладную платформу, которая может быть включена в сеть связи (рисунок 2.1).

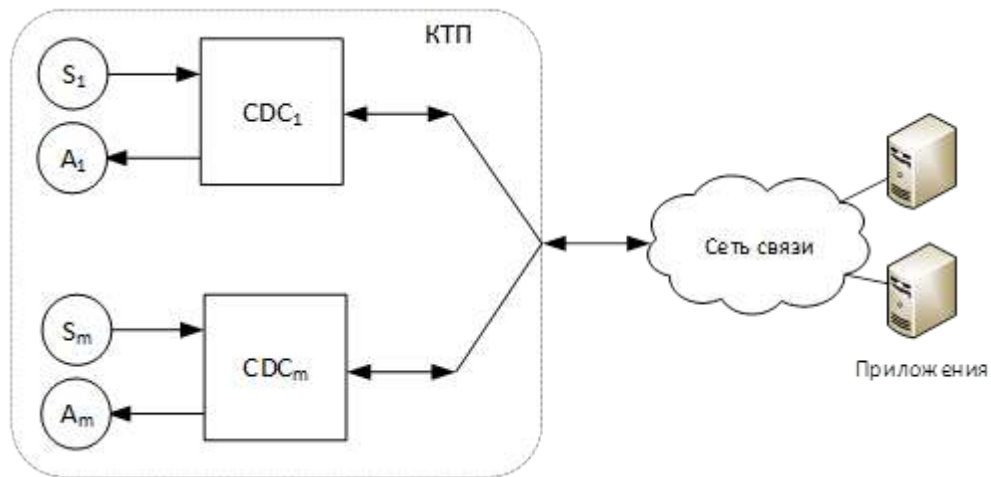


Рисунок 2.1 - Структура взаимодействия сенсоров и актуаторов КТП

Канал связи КТП с сетью связи, в общем случае, является тонким местом, например, при подключении пользователя к сети через низкоскоростной канал. Объем передаваемых данных, интенсивность трафика зависят от количества передаваемой информации. Это количество определяется свойствами получаемых от сенсоров сигналов и особенностями реакции человека на соответствующие воздействия. В большинстве случаев сенсор и актуатор образуют пару устройств, предназначенную для получения и представления информации. Передача и прием информации осуществляется с помощью кодеков $CDC_1 \dots CDC_m$. Кодек содержит в своем составе (рисунок 2.2) устройство предварительной обработки сигнала (ПОС), АЦП, цифроаналоговый преобразователь (ЦАП) и устройство восстановления сигнала (УВС), возможна также и цифровая обработка сигнала (ЦОС) [29].

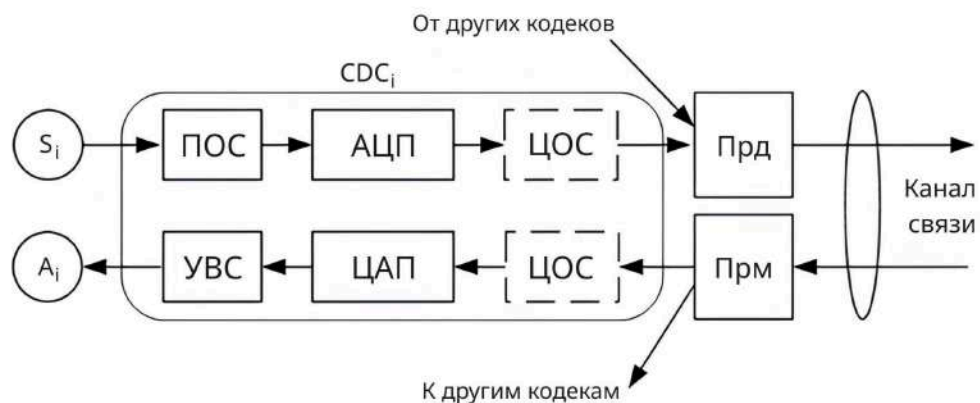


Рисунок 2.2 - Структура кодека в составе КТП

Все кодеки КТП обмениваются данными через его общий канал связи, содержащий передатчик TX и приемник данных RX. Интенсивность трафика в канале связи зависит от способа кодирования информации и передачи данных. Эффективность информационного обмена и использования ресурсов канала связи зависят от количества передаваемой информации и способа ее передачи. Передатчик и приемник содержат буферы, куда помещаются предназначенные для передачи и принятия данные (рисунок 2.3). Таким образом, в канале возникает задержка на ожидание в буфере, которая обусловлена, временем передачи пакета t_{TX} (его размером и скоростью), а при приеме - временем обработки пакета t_{RX} . Эти задержки зависят от времени передачи (обработки) и интенсивности поступления пакетов (интенсивности трафика). При конечном размере буфера вероятность потерь пакетов становится отличной от нуля.

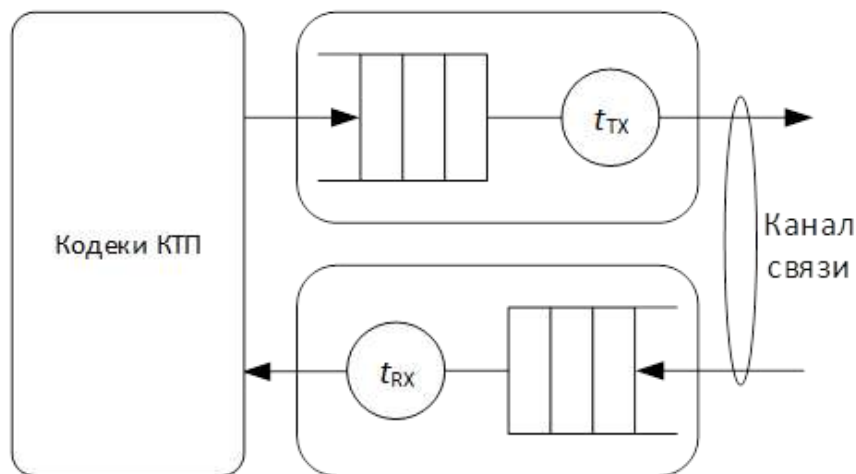


Рисунок 2.3 - Структура взаимодействия с каналом связи

Как задержка в буфере, так и потери пакетов оказывают влияние на качество функционирования системы в целом и на качество восприятия услуги. В общем случае, средняя величина задержки в буфере определяется как:

$$\overline{W} = \frac{\bar{\rho}\tau}{2(1-\rho)} \varepsilon(T, S), \quad (2.1)$$

где $\varepsilon(T, S)$ – величина, зависящая от множества свойств трафика T и множества свойств процесса обслуживания S ; ρ – интенсивность трафика (нагрузка, использование канала):

$$\rho = \lambda \bar{\tau}, \quad (2.2)$$

λ – интенсивность поступления пакетов; $\bar{\tau}$ – время передачи пакета:

$$\bar{\tau} = \frac{\bar{L}}{b}, \quad (2.3)$$

$\bar{L}$ – средняя длина кадра; b – скорость передачи.

Выражение (2.1) демонстрирует зависимость задержки от нагрузки (использования канала). Очевидно, что для обеспечения качества предоставления услуги целесообразно стремиться к уменьшению величины нагрузки. При ограниченном ресурсе канала, т. е. скорости передачи данных, это возможно за счет уменьшения интенсивности кадров λ или за счет уменьшения времени передачи $\bar{\tau}$ (уменьшения среднего размера пакета).

Разумеется, невозможно уменьшить интенсивность передачи пакетов и их размер, не потеряв информацию. В данном случае для решения задачи необходимо найти способ кодирования и передачи информации, обеспечивающий эффективность системы в целом, т. е. своевременную доставку ее достаточного количества. Для решения данной задачи в работе рассматривается одна из подсистем КТП и выбирается способ кодирования данных этой подсистемы с учетом свойств сигналов, производимых ее датчиками.

2.2. Модель сигналов термодатчиков КТП

Сигналы, производимые датчиками температуры КТП, можно разделить на две группы. В первую входят сигналы, производимые датчиками, расположенными на кончиках пальцев, они ориентированы на передачу наиболее информативных тактильных тепловых ощущений, возникающих при выполнении действий руками. Во вторую группу входят датчики, которые могут быть размещены на иных участках тела (на руках, груди, спине), они предназначены, в основном, для передачи информации о температуре окружающей среды, на данном этапе построения КТП.

Информация, получаемая первой группой датчиков, важна при выполнении оператором различных действий руками (манипулятором) или взаимодействия с виртуальной копией при прикосновениях к объектам. Время доставки тактильной информации о температуре объекта прикосновения зависит от множества факторов. В качестве основных можно выделить: теплоемкость и тепловое сопротивление материалов на пути распространения тепла от объекта до чувствительного элемента, время преобразования аналогового сигнала в цифровой (в том числе и периодичность опроса датчика), время передачи и цифроаналогового преобразования данных, время реакции (изменения температуры) актуатора, а также теплоемкость и тепловое сопротивление материалов на пути распространения тепла от актуатора к поверхности пальцев оператора. Теплоемкости и тепловые сопротивления материалов также определяют время, необходимое для передачи тепла.

Таким образом, данные на выходе АЦП появляются уже с задержкой, обусловленной физическими процессами передачи тепла. Для выбора способа их доставки необходимо описать свойства этих сигналов. Будем рассматривать такие свойства как динамический диапазон значений, форму сигнала и энергетический спектр сигнала.

Как было отмечено выше, изменения температуры датчиков зависят от совокупности факторов, в том числе от действий (поведения) субъекта. Поэтому, как и в работе, выделим две группы сигналов: расположенные на пальцах (характерны наиболее быстрые движения и изменения температуры) и на других частях тела (характерны медленные движения и изменения температуры).

Как известно, для точного восстановления сигнала с конечным спектром, частота дискретизации должна не менее, чем вдвое превосходить верхнюю частоту спектра сигнала, согласно теореме Котельникова – Шеннона – Найквиста. В реальных случаях ширина спектра сигнала может быть весьма большой (она ограничена лишь представлением результатов измерений). Высокочастотные составляющие могут быть обусловлены тепловым шумом датчика и других элементов, поэтому выбирать верхнюю частоту, полученную в результате

наблюдений, нецелесообразно. В виду этого, в качестве верхней частоты спектра сигнала следует принять ее значение для наибольшей (заданной) доли энергии. Выберем значения для 80 и 90 % энергии сигнала для определения верхней частоты спектра сигнала.

На рисунке 2.4 приведена форма сигнала, полученного на выходе датчика температуры $x(t)$ первой группы («быстрые» изменения). Сигнал получен при регулярном поочередном воздействии на термодатчик высокой и низкой температурами (высокая температура – 70 °С, низкая – 4 °С). Динамический диапазон сигнала составил 8,7 дБ. Периодичность изменения температуры составляла около 0,5 с, форма сигнала походит на треугольную, однако вернее ее описать экспоненциальным нарастанием и снижением значения, как реакцию на скачек температуры. Этот сигнал имитирует быстрые касания пальцами руки и относится к «быстрым» сигналам термодатчика.

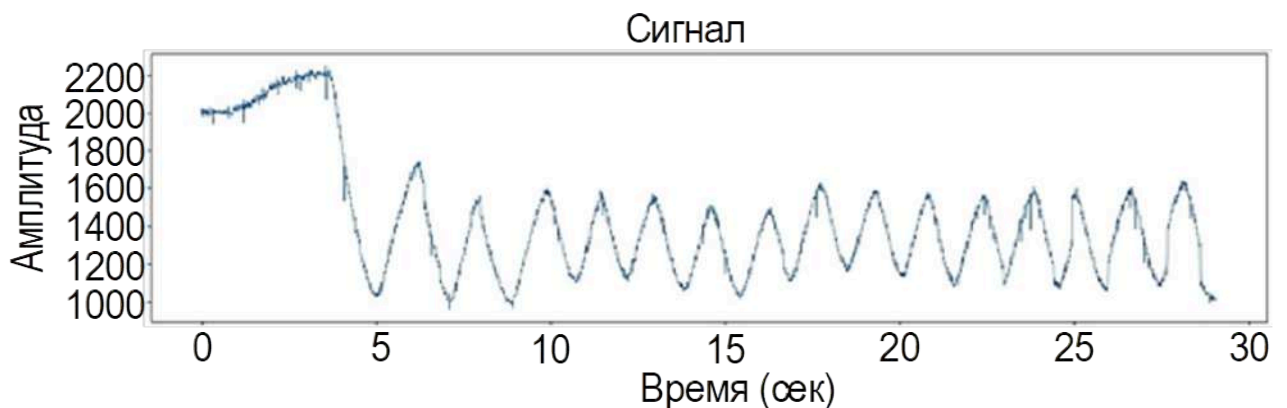


Рисунок 2.4 - Типовая форма сигнала на выходе датчика температуры первой группы

На рисунке 2.5 представлен энергетический спектр приведенного выше сигнала $S(f)$. Показано, что подъем энергетического спектра имеет место в области низкой (близкой к нулю) частоты и в области частоты в 1 Гц. Сигнал достаточно низкочастотный, с постоянной составляющей. Наибольшая доля энергии сосредоточена в сравнительно малом диапазоне частот, приблизительно от 0 до 2Гц.



Рисунок 2.5 - Энергетический спектр сигнала на выходе термодатчика первой группы

На рисунке 2.6 приведена гистограмма зависимости доли энергии «быстрого» сигнала от частоты.

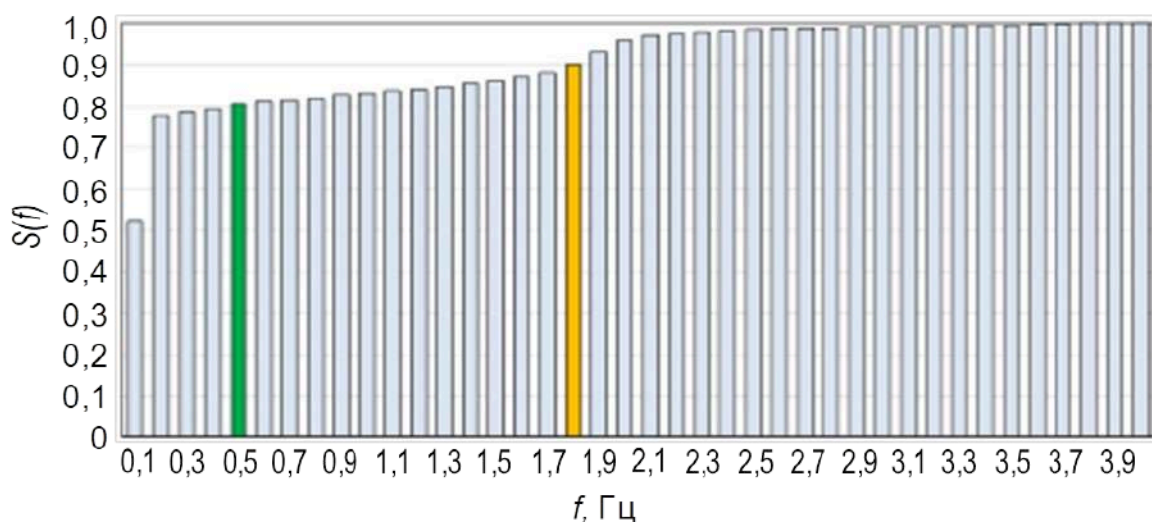


Рисунок 2.6 - Зависимость доли энергии «быстрого» сигнала температуры от частоты

Цветом на рисунке отмечены частотные интервалы, на которых наблюдаются значения 80 и 90 % энергии сигнала. Доля энергии сигнала до 80 % передается частотами до 0,5 Гц, а доля энергии 90 % – частотами до 1,8 Гц.

На рисунке 2.7 приведен сигнал «медленных» изменений температуры $x(t)$ от датчиков второй группы. Периодичность изменения температуры в этом случае составила около 5 с. В этом случае экспоненциальные нарастание и спад более отчетливы. Динамический диапазон сигнала составил 20,3 дБ. На рисунке 2.8 представлен энергетический спектр этого сигнала. Наибольшая мощность имеет

место в области частот от 0 до 1 Гц. Энергетический спектр лежит ниже по частоте, чем сигнала от первой группы датчиков.

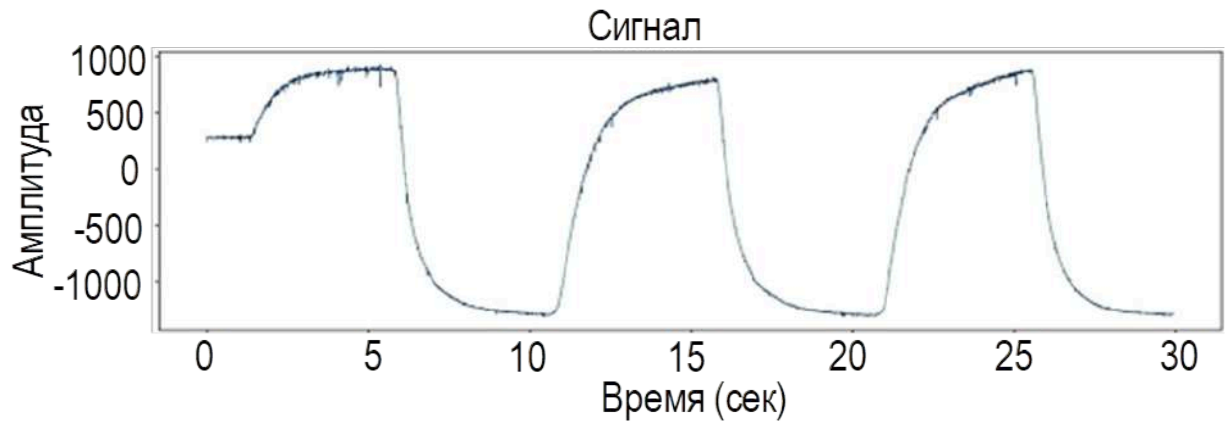


Рисунок 2.7 - Типовая форма сигнала на выходе датчика температуры второй группы



Рисунок 2.8 - Энергетический спектр сигнала на выходе термодатчика второй группы

На рисунке 2.9 приведена гистограмма зависимости доли энергии «медленного» сигнала от частоты.

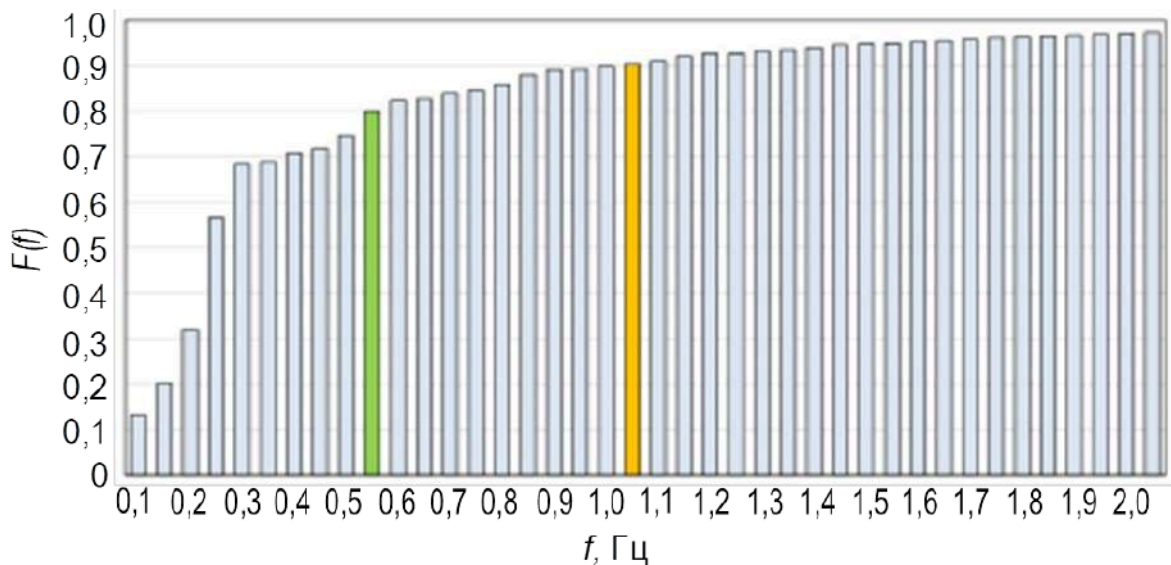


Рисунок 2.9 - Зависимость доли энергии «медленного» сигнала температуры от частоты

Цветом на рисунке отмечены частотные интервалы, на которых наблюдаются значения 80 и 90 % энергии сигнала. Доля энергии сигнала до 80% передается частотами до 0,55 Гц, а доля энергии 90 % – частотами до 1,05 Гц. Таким образом можно заметить, что для дискретизации «медленных» сигналов термодатчиков может быть выбрана частота 2 Гц, а для «быстрых» – 4 Гц. Это обеспечит достаточное точное воспроизведение принимаемого сигнала. В случае необходимости частота дискретизации может быть понижена до 1 Гц, она обеспечит воспроизведение около 80 % энергии как «медленных», так и «быстрых» сигналов.

Для устранения шумовой составляющей целесообразно использовать фильтр низкой частоты с частотой среза, превышающей верхнюю частоту сигнала, но вдвое ниже частоты дискретизации.

2.3. Представление данных о температуре и их воспроизведение КТП

Отметим, что необходимым условием является то, что диапазон температур, воспроизводимых актуаторами КТП, должен соответствовать диапазону воспринимаемых температур с учетом ограничений безопасности для человеческого организма. Также следует отметить, что согласно закону

Вебера – Фехнера, степень восприятия теплового воздействия человеком не линейна. Фактически человек способен различать степень теплового воздействия в достаточно узком диапазоне нормальных для организма температур. За его границами воздействия различимы в меньшей степени.

Ограничения воспроизводимой температуры сверху могут быть определены с помощью ГОСТ Р 51337-99. Согласно данному стандарту, температура поверхности касания, превышающая некоторую величину, способна вызвать ожоговую травму (далее просто ожог), степень которой может быть различна и зависит от времени контакта и температуры кожи. Ввиду того, что температуру кожи при получении ожога определить сложно, в документе приводится температура контактируемой поверхности. При этом опасность и степень ожога зависят от свойств материала контактируемой поверхности. Наиболее «опасны», с точки зрения вероятности получения ожога, материалы с большей теплопроводностью и теплоемкостью, например, металлы.

При реализации передачи тепловых ощущений в КТП используются материалы с высокой теплопроводностью, что необходимо учитывать при выборе пороговой температуры. Например, при контакте с металлом в течение 1 мин, ожоговый порог составляет 51 °С, при времени контакта 1 с – 85 °С, а при времени контакта 10 с – около 55 °С. Таким образом, при построении КТП следует учитывать время теплового воздействия, которое должно быть ограничено безопасной величиной для данной температуры.

Для оценки нижней границы температуры воспользуемся ГОСТ Р ИСО 13732-3-2013. При низкой температуре контактируемой поверхности имеет место риск получения травмы в виде обморожения (далее просто обморожения). Вероятность обморожения зависит от температуры контактируемой поверхности, теплопроводности материала поверхности и продолжительности контакта. Например, согласно данному стандарту, риск обморожения наступает при контакте с алюминиевой поверхностью при температуре 0 °С в течение 40 с или при температуре –20 °С в течение около 1 с. Потеря чувствительности при аналогичном контакте и температуре 0 °С наступает

в течение интервала около 2 с. Болевые ощущения возникают при температуре 0 °С в интервале времени 1 с.

Таким образом, анализируя данные из двух упомянутых документов, можно сделать вывод о том, что целесообразно ограничить диапазон воспроизводимых температур безопасными величинами из расчета на некоторый интервал времени, достаточный для отключения или снятия КТП. Выберем в качестве такого интервала 10 с. Тогда верхняя граница температуры составит 50 °С, а нижняя граница будет равной –5 °С. Тогда полный диапазон температур [–5 °С; 50 °С] составляет 55 ° (при использовании линейной шкалы, динамический диапазон значений будет равен около 21 дБ).

Из сказанного следует, что во всем диапазоне температур имеет место участок, где изменение температуры наиболее информативно для человека. Логично предположить, что этот участок находится вблизи нормального значения температуры человеческого тела (35,5...37,2 °С), при этом температура кожного покрова составляет около 32 °С. Согласно [10], болевые ощущения у человека возникают, когда температура выходит за границы диапазона [15 °С; 43 °С]. Будем полагать, что этот диапазон температур близок к наиболее информативному для человека диапазону. На данном этапе построения КТП выберем за базовый несколько сокращенный интервал температур [20 °С; 40 °С]. Значения в этом диапазоне должны быть представлены достаточно точно.

Вопрос обеспечения точности измерений затрагивает различные аспекты построения КТП и в данной работе в полной мере не рассматривается. Здесь исследуется лишь вопрос выбора дискретности представления значений температуры. Произведем этот выбор на основе требований к точности медицинских термометров, ошибка измерения, согласно этому документу, не должна превышать 0,1 °С. Выберем эту величину как шаг представления данных о температуре в интервале значений [20 °С; 40 °С]. Выбор, сделанный на данном этапе как предварительный, обусловлен отсутствием достаточно полных описаний тактильной чувствительности человека в части восприятия температуры и в дальнейших исследованиях может уточняться. Таким образом, для представления

значений температуры в указанном диапазоне требуется 200 значений (т. е. 8 двоичных разрядов). Для представления с той же дискретностью температуры в диапазоне $[-5\text{ }^{\circ}\text{C}; 50\text{ }^{\circ}\text{C}]$ потребуется 550 значений, т. е. 10 двоичных разрядов.

Применим операцию сжатия динамического диапазона значений (рисунок 2.10). На рисунке 2.10а приведена характеристика в абсолютных значениях из расчета на 8 двоичных разрядов, а на рисунке 2.10b – нормированная характеристика. В диапазоне от 20 до 40 °C значения представляются линейной зависимостью с дискретизацией 0,1 °C. В диапазонах от –5 до 20 °C и от 40 до 50 °C производится сжатие значений. Особенностью в этом случае является то, что сжатие динамического диапазона требуется как «сверху», так и «снизу».

Для аппроксимации данной характеристики непрерывной функцией предлагается использовать S -кривую вида:

$$y(x) = \frac{1}{1 + e^{-\frac{x-x_0}{b_0}}}, \quad (2.4)$$

где b_0 – наклон кривой в диапазоне от 20 до 40 °C (в данном случае $b_0 = 0,1$); x_0 – точка перегиба, вычисляется исходя из минимума суммарного квадратического отклонения S -кривой от заданной кусочно-линейной характеристики в диапазоне от 20 до 40°C (в данном случае $x_0 \approx 0,524$).

Декомпрессия сигнала при его восстановлении производится функцией обратной к (2.4):

$$y^{(-1)}(x) = x_0 - b_0 \ln\left(\frac{1}{x} - 1\right). \quad (2.5)$$

Графики функций (4) и (5) приведены на рисунке 2.11 (красная кривая).

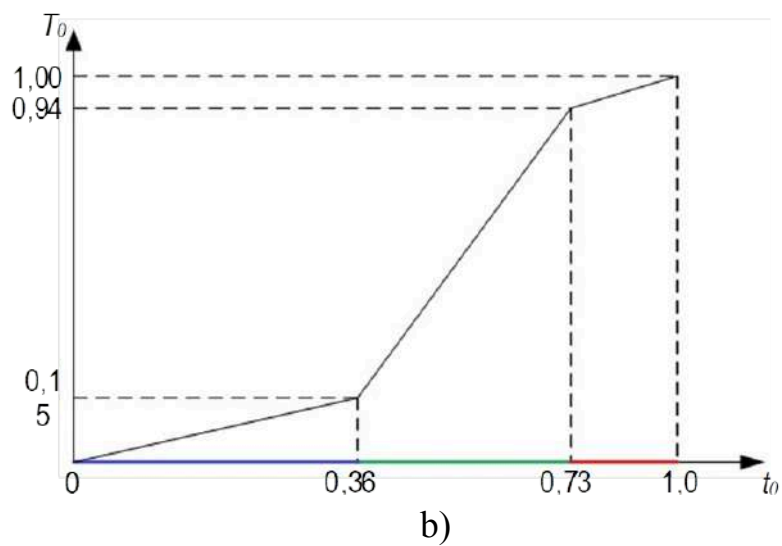
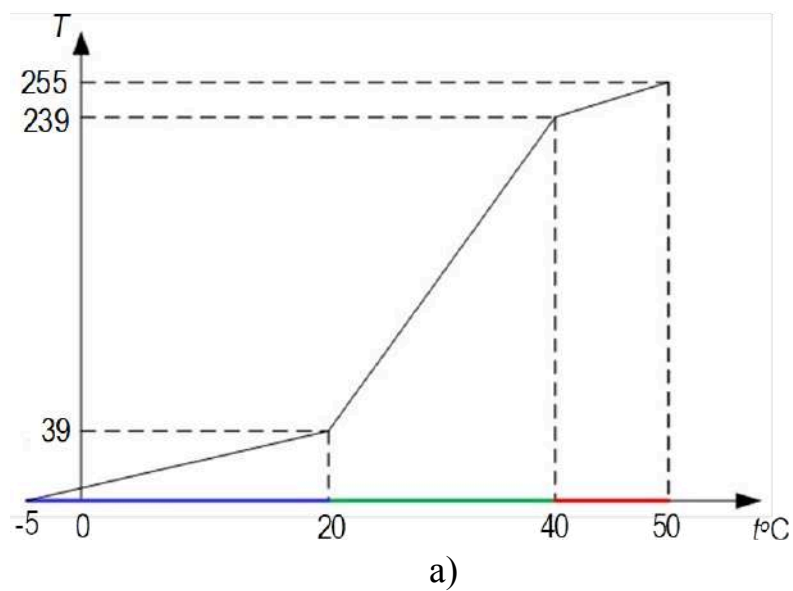


Рисунок 2.10 - Кусочно-линейная характеристика преобразования значений температуры: а) абсолютные значения из расчета на 8 двоичных разрядов; б) нормированная характеристика

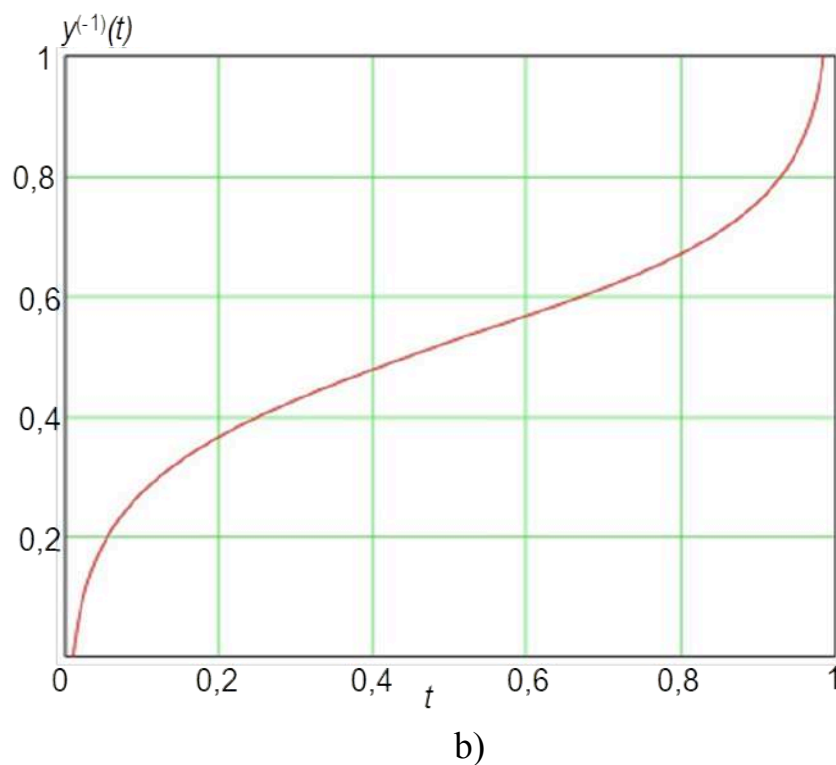
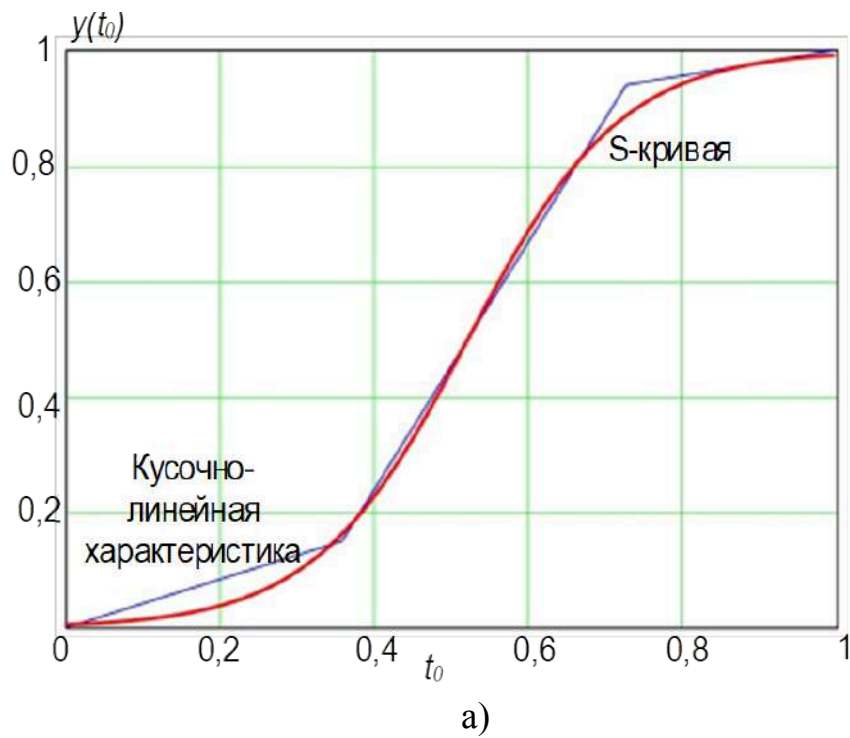


Рисунок 2.11 - Аппроксимация характеристик преобразования (а) и восстановления (б) значений температуры S-кривой

Применение операции сжатия динамического диапазона сигнала температуры позволяет сократить количество разрядов двоичного кода с 10 до 8, т. е. на 20 %. Это позволяет понизить интенсивность трафика подсистемы термодатчиков КТП на близкую к 20 % величину. При этом обеспечивается дискретность

представления значений температуры в диапазоне тактильной чувствительности с шагом 0,1 °C.

2.4. Модель и метод формирования данных

Как было отмечено выше, подсистема термодатчиков КТП может содержать достаточно большое их количество. Следует заметить, что сигналы от термодатчиков случайны, но в большинстве случаев достаточно сильно коррелированы друг с другом. Например, сигналы термодатчиков пальцев руки близки, при захвате рукой некоторого предмета, температура поверхности которого вероятнее всего близка со всех сторон. Большинство окружающих предметов имеют температуру, близкую к комнатной. Это позволяет предположить, что измеренные значения температуры термодатчиками КТП будут отличаться на весьма небольшую величину. В таком случае нерационально передавать полные данные о каждом из датчиков, т. е. все 8 бит из расчета на каждый датчик.

Предположим, что значения температуры, измеряемые датчиками КТП, случайны и имеют распределение вероятности $F(t)$ и плотность вероятности $f(t)$ со средним значением $\bar{t}$, для множества из n датчиков.

Определим отклонения температуры от среднего значения с помощью выражения:

$$\Delta t_i = \hat{y}(\bar{t}) - \hat{y}(t_i),$$

$$i = 1 \dots n,$$
(2.6)

где $\hat{y}()$ – применение функции (4) и преобразование в двоичное 8-разрядное число.

Тогда вероятность того, что значение температуры окажется в диапазоне от $[\bar{t} - \Delta; \bar{t} + \Delta]$, где Δ является некоторым интервалом значений, будет определяться как:

$$p_{\Delta} = \int_{\bar{t}-\Delta}^{\bar{t}+\Delta} f(t)dt. \quad (2.7)$$

Практически было установлено, что вероятность отличия показаний какого-либо из датчиков t_i от среднего значения $\bar{t}$ более, чем на 2° , невысока. На основе этого свойства предлагается сократить объем передаваемых данных о температуре, путем передачи только среднего значения и отклонений от среднего значения для каждого из датчиков. Предлагается следующий формат представления данных о температуре для группы из $0 < n \leq 127$ датчиков (таблица 2.1).

Таблица 2.1 - Формат представления данных о температуре для группы датчиков

№ байта	Значение			Примечание
0	I, n			Число датчиков в группе n , идентификатор протокола I
1	$y(\bar{t})$			Среднее значение температуры
2	B			Число разрядов для отклонения
2	Δt_1	Δt_2	Δt_3	Отклонения температуры для 1-го, 2-го датчиков
$[\frac{n}{2}] + 2$	Δt_{n-1}	Δt_{n-2}	Δt_n	Отклонения температуры для $(n-1)$ -го и n -го датчиков
$[\frac{n}{2}] + 3$	$\bar{t}_k$			Значение температуры k -го датчика, если $ \Delta t_k > \Delta$

$\lceil \frac{n}{2} \rceil + m$	$\bar{t}_r$	Значение температуры k -го датчика, если $ \Delta t_r > \Delta$
---------------------------------	-------------	--

Предложенный формат представляет собой блок данных, который содержит информацию о количестве датчиков, типе протокола, среднем значении температуры и отклонениях температуры каждого из датчиков от среднего значения. Число датчиков определяется целым числом от 0 до 127 (младшие 7 разрядов первого байта). Если старший разряд первого байта соответствует «0», то используется сокращенный формат данных, если – «1», то полный формат данных.

В сокращенном формате следующий байт содержит среднее значение температуры, полученное согласно (4):

$$y(\bar{t}), \quad \bar{t} = \frac{1}{n} \sum_{i=1}^n t_i. \quad (2.8)$$

Следующий байт содержит количество двоичных разрядов, используемых для записи отклонения. Это число определяется по выражению:

$$B = \log_2(2\Delta), \quad (2.9)$$

где Δ – величина одностороннего отклонения значения от среднего, которая может быть записана в B -разрядное двоичное число.

Выбор величины Δ или B влияет на вероятность того, что число может быть записано в этом количестве разрядов (2.7), следовательно, и на общее количество передаваемых данных (в байтах):

$$V_{\Delta}(B) = \lceil nB/8 \rceil + 2 + n \left(1 - \int_{\bar{t}-2^{B-1}}^{\bar{t}+2^{B-1}} f(t) dt \right). \quad (2.10)$$

Выбор B^* можно рассматривать как задачу целочисленной оптимизации:

$$B^* = \arg \min_B \left\{ [nB/8] + 2 + n \left(1 - \int_{\bar{t}-2^{B-1}}^{\bar{t}+2^{B-1}} f(t) dt \right) \right\}, \quad (2.11)$$

$$2 < B \leq 8.$$

Выражение (2.11) минимизирует общий объем передаваемых данных путем выбора оптимально значения разрядности B .

В данном случае в качестве $f(t)$ выбрано нормальное распределение:

$$f(t) = g(t, \bar{t}, \sigma^2), \quad (2.12)$$

где $g(t, \bar{t}, \sigma^2)$ – плотность нормального распределения вероятности; параметры $\bar{t}$ – среднее значение; σ^2 – дисперсия могут быть определены на основе полученных данных от датчиков температуры:

$$\bar{t} = \frac{1}{n} \sum_{i=1}^n t_i, \quad \sigma^2 = \frac{1}{n} \sum_{i=1}^n (t_i - \bar{t})^2. \quad (2.13)$$

На рисунке 2.12 приведен вид зависимости (2.10) от числа разрядов B .

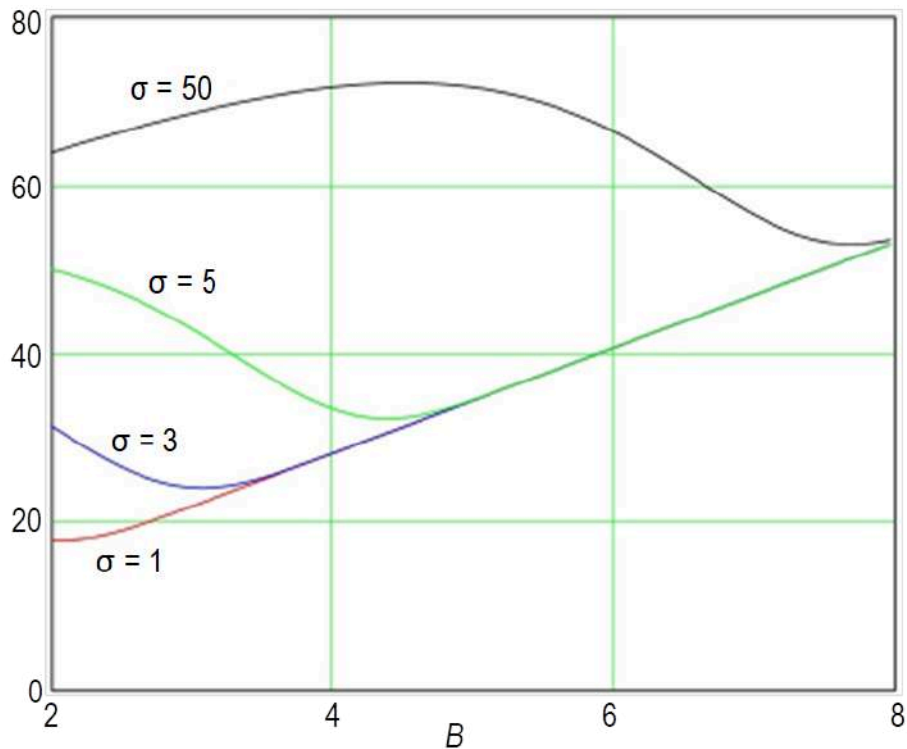


Рисунок 2.12 - Зависимость размера, передаваемых данных от числа разрядов

Использование сокращенного формата целесообразно, если он уменьшает количество передаваемых данных. Эффективность его использования может быть определена как:

$$e = 100 \frac{V - V_{\Delta}}{V} \%, \quad (2.14)$$

где V – размер данных при передаче без сжатия:

$$V = n + 1 \text{ байт.}$$

Выражение (2.14) можно интерпретировать как выигрыш, получаемый при использовании сокращенного формата по сравнению с полным форматом представления данных. Применение сокращенного формата целесообразно, когда выражение (2.14) принимает значение больше нуля. Из рисунка 2.13 видно, что с ростом разброса значений эффективность снижается. Например, при $\sigma = 50$ метод неэффективен. Когда метод неэффективен следует выбирать полный размер данных.

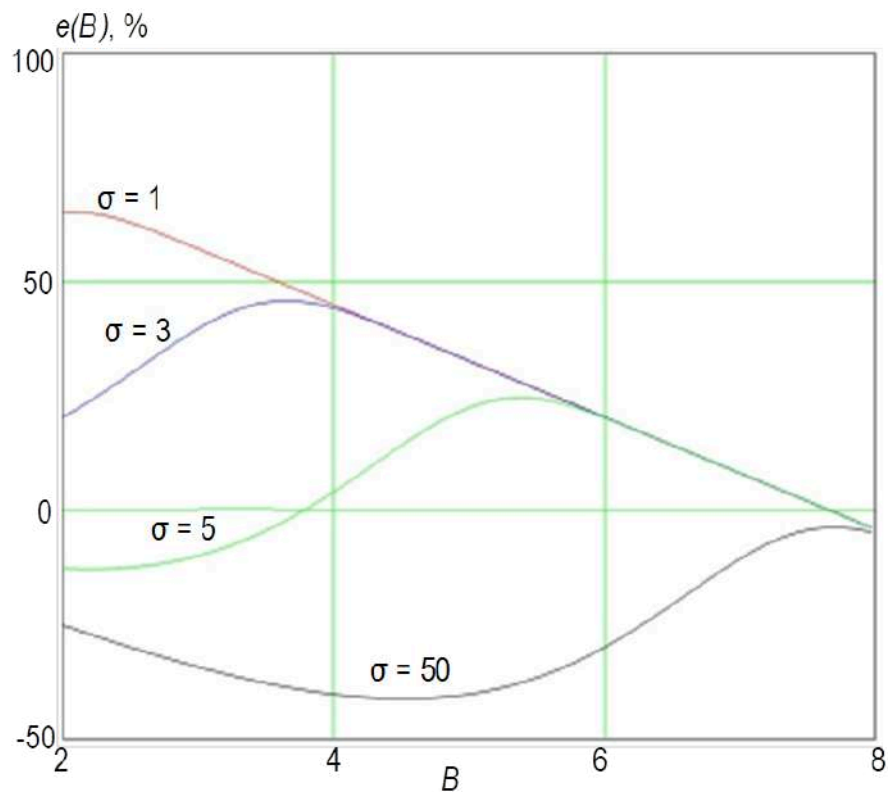


Рисунок 2.13 - Зависимость эффективности от числа разрядов и разброса значений температуры

2.5. Выводы по главе

Модель данных КТП потенциально может содержать большое количество поступающих от тактильных и температурных сенсоров данных, а также тех, которые поступают к актуаторам. Это требует выбора метода их представления и передачи. Исследования данных, получаемых от датчиков температуры, позволили оценить динамический диапазон и энергетический спектр сигнала. Динамический диапазон составляет около 20 дБ. Анализ сигналов показал, что целесообразно выделить две группы сигналов от термодатчиков: «медленные», получаемые от сенсоров, реагирующих, в основном, на температуру окружающей среды, и «быстрые» сигналы от датчиков, реагирующих на прикосновения. Анализ энергетических спектров сигналов показал, что верхняя граница спектра «медленных» сигналов составляет около 1 Гц, а «быстрых» – около 2 Гц. Это позволяет выбрать частоту дискретизации 2 Гц для «медленных» и 4 Гц для «быстрых» сигналов термодатчиков.

Учитывая закон Вебера – Фехнера, разработан метод сжатия динамического диапазона сигналов с помощью *S*-кривой. Данный метод позволяет на 20 % сократить разрядность двоичного представления сигнала (с 10 до 8 бит), что также дает возможность снизить величину трафика (примерно на 20 %, без учета служебных данных), производимого подсистемой термодатчиков КТП.

Разработан адаптивный метод передачи данных подсистемы термодатчиков, позволяющий повысить эффективность путем снижения объема передаваемых данных за счет использования вероятностных свойств сигналов термодатчиков. Метод позволяет получить выигрыш, величина которого зависит от количества термодатчиков и степени разброса значений температуры. Типовое значение выигрыша составляет около 30 %.

ГЛАВА 3. МЕТОД ПЕРЕДАЧИ ТАКТИЛЬНОЙ И КИНЕСТЕТИЧЕСКОЙ ИНФОРМАЦИИ НА БАЗЕ КОСТЮМА ТЕЛЕПРИСУТСТВИЯ

3.1. Классификация костюмов телеприсутствия

Для разработки классификации костюмов телеприсутствия как интерфейса "пользователь - цифровой аватар" согласно рекомендациям МСЭ и проекту совместимости, учитывающая типы костюмов по функционалу, видам обратной связи и характеру взаимодействия с аватаром, можно выделить следующие типы костюмов. Данная классификация представляет собой своего рода этапы развития технологии [22-29].

Таблица 3.1. Классификация костюмов телеприсутствия

Вид костюма	Основные характеристики	Типы сенсоров	Тип цифрового аватара
Стандартный костюм	Базовый захват движений пользователя	Инерционные датчики (IMU), акселерометры, гироскопы, камеры	Управление базовым аватаром и его анимацией
Костюм с кинематикой	Использование тактильной и кинематической обратной связи	Силовые тензодатчики, сервоприводы и другие	Высокая степень иммерсивности, управление аватарами с обратной связью
Мультисенсорный костюм	Учет температуры, влажности, давления, вибрации	Температурные датчики, датчики влажности, таблеточные (хаптические) вибромоторчики	Реалистичное тактильное воспроизведение, управление продвинутыми аватарами
Костюм с расширенными сетевыми функциями	Интеграция с сетевыми вселенными	Многоканальные беспроводные интерфейсы, сложные архитектуры связи. Интеграции с умными городами. Использование ИИ	Управление роботами-гуманоидами, облачная синхронизация аватаров. Погружение в многопользовательские зеркальные вселенные
Костюм с использованием новых технологий	Использование нейроинтерфейсов	Применение датчиков подключаемые к нейронной сети человека, и другие новые технологии	Управление аватарами нового поколения

3.2. Разработка костюма телеприсутствия

Разработанный костюм телеприсутствия в текущей версии состоит из двух основных частей. Это сам костюм с кинестетической обратной связью (КОС) и тактильная перчатка:



Рисунок 3.1 - Костюм телеприсутствия

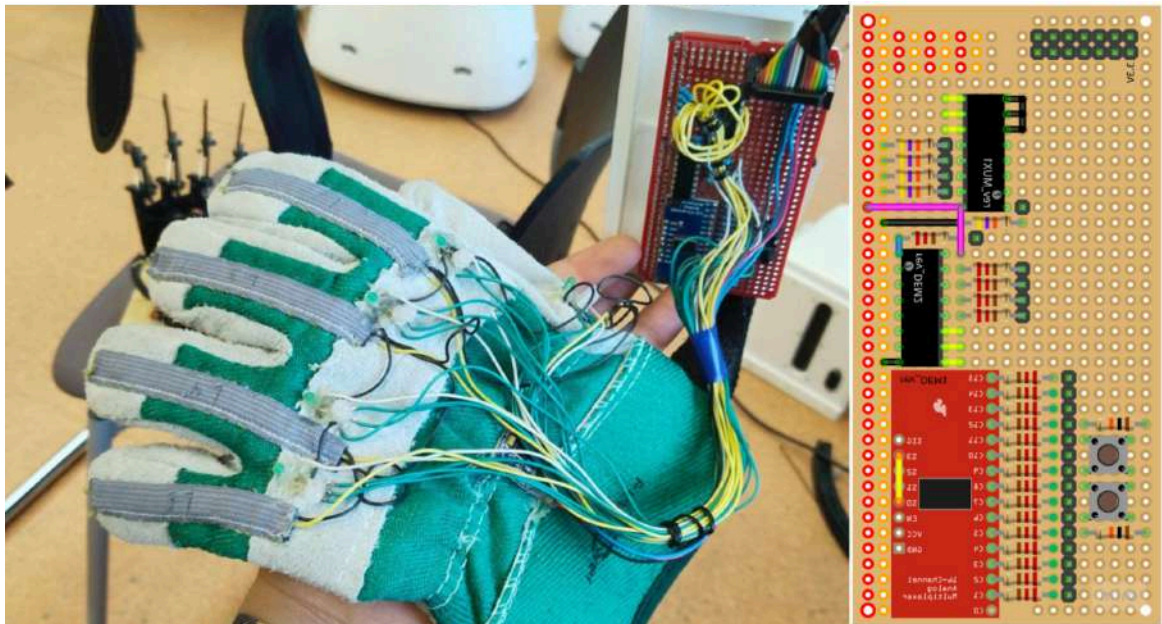


Рисунок 3.2 - Перчатка костюма телеприсутствия и схема дополнительной платы перчатки

3.2.1. Сетевая схема взаимодействия элементов КТ и соответствующие требования

Разработанная система в целом имеет следующий вид:

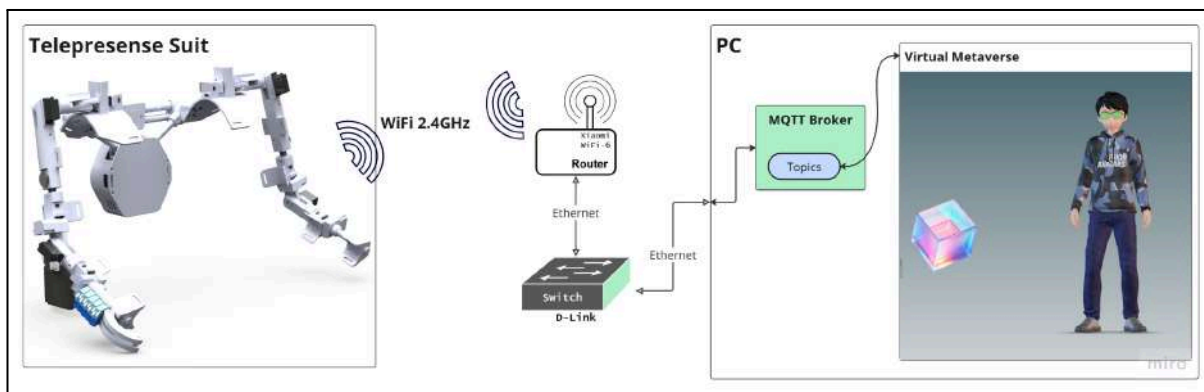


Рисунок 3.3 - Абстрактная сетевая схема взаимодействия КТ с метавселенной.

Если разбить элементы системы на составные части, то получим следующую схему:

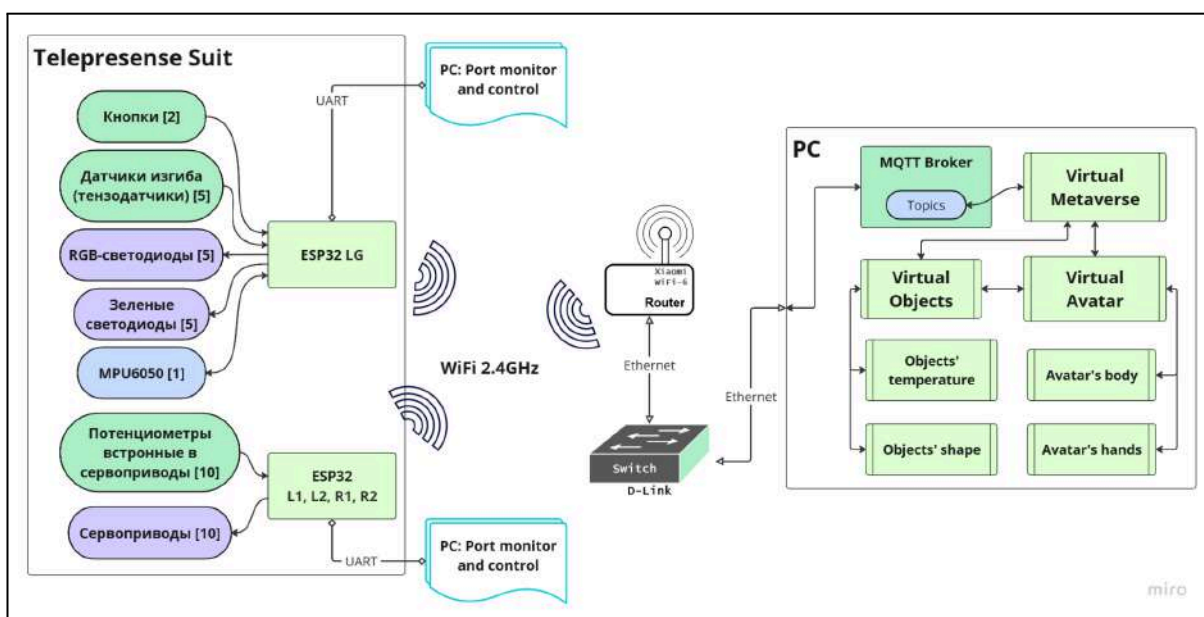


Рисунок 3.4 - Разобранный вид вышеописанной схемы.

Функциональные требования:

- Сигналы управления в системе - двусторонние и асинхронные.
- Передача данных осуществляется по готовности работы обеих сторон.
- КТ собирает данные о состоянии углов поворота конечностей и отправляет их на сервер, на виртуальный аватар.
- Перчатка КТ также отправляет данные изгиба пальцев на сервер, на виртуальный аватар.

- Виртуальный аватар, в свою очередь, если его части тела попадают на препятствия, то отправляет команду остановки соответствующих конечностей КТ.
- Если как-нибудь воздействовать на виртуальный аватар, то эти воздействия также передаются на КТ.
- В случае, если пальцы аватара касаются виртуального объекта с какой-либо температурой, то эти данные также обратно передаются на КТ и на перчатку КТ.

3.2.2. Сценарий использования КТП

В целом, процесс использования КТП пользователем можно обобщенно подразделить на следующие действия:

- Пользователь надевает КТП и подает питание на микроконтроллеры и на сервоприводы.
- Микроконтроллеры, установленные на КТП включаются и выполняется загруженные в них программные алгоритмы (прошивки), которые представлены на следующих подразделах.
- В процессе работы КТ обменивается данными тактильного интернета (иммерсивные данные обратной связи, как положение частей тела пользователя и виртуального аватара, температура виртуального объекта, касание и давление на виртуальный объект, и другие данные при внесении изменений в используемые программы и в конструкцию КТ и его частей)
- При окончании использования КТ, пользователь подает команду завершения работы с помощью кнопок на дополнительной плате перчатки и отключает питание, или просто отключает питание на КТ и его элементы.
- Пользователь снимает КТ.

3.2.3. Сеть контроллеров и сенсорика костюма телеприсутствия

В текущей версии костюма на кейсе костюма (на задней части КТ) имеются пять плат **ESP32**, четыре из которых отвечают за костюм (L1, L2, R1, R2), а один - за перчатку (LG) [31-32].

Расположение управляющих плат показаны на следующих картинках:

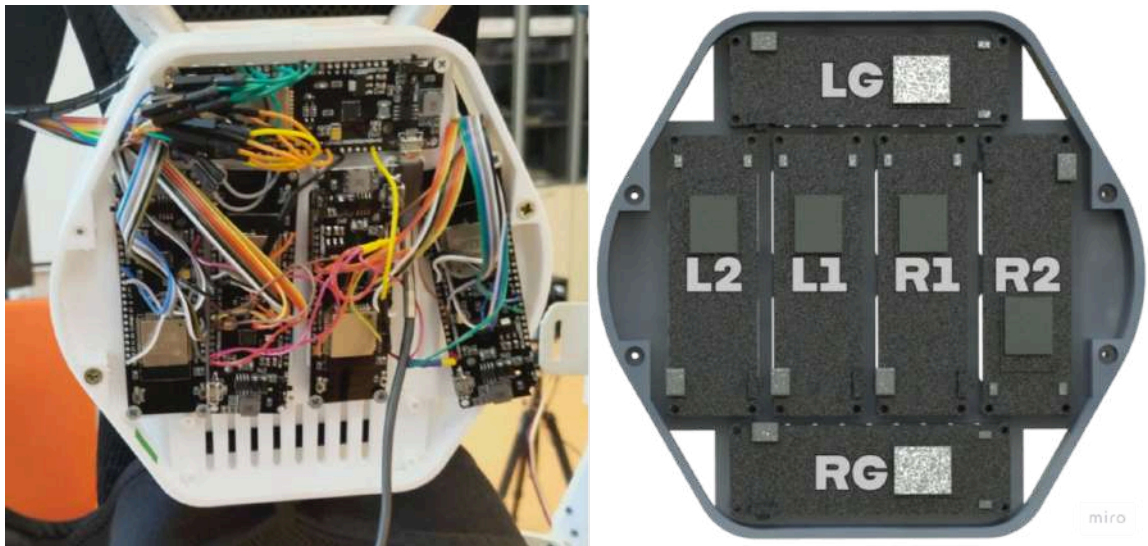


Рисунок 3.5 - Кейс костюма на котором расположены управляющие микроконтроллеры

Назначение плат:

- L1 - работа с **двумя** сервоприводами, которые находятся на **верхней** части **левой** руки
- L2 - работа с **тремя** сервоприводами, которые находятся на **нижней** части **левой** руки
- R1 - работа с **двумя** сервоприводами, которые находятся на **верхней** части **правой** руки
- R2 - работа с **тремя** сервоприводами, которые находятся на **нижней** части **левой** руки
- LG - работа с элементами, которые прикреплены к перчатке:
 - 5 датчиков изгиба подключенных через делитель напряжения, подключенных через восьмиканальный мультиплексор
 - акселерометр и гироскоп
 - 5 RGB светодиодов
 - 5 мелких зеленых светодиодов
 - 2 кнопки

3.2.4. Составляющие модули КТП и их характеристики

Сервоприводы (FB5317M-360) в костюме телеприсутствия необходимы для кинетического/механического воздействия на пользователя. В используемых сервоприводах встроены потенциометры, которые выдают аналоговые сигналы с напряжением (в идеале) от 0 до 3.3В. Аналоговые сигналы с них получаем используя **АЦП** управляющей платы. Сами сервоприводы питаются от 5В.

Управляются эти сервоприводы посредством **ШИМ** сигналов, выдаваемых от цифровых выводов микроконтроллера.



Рисунок 3.6 - Внешний вид сервопривода FB5317M-360

Датчики изгиба (Flex sensor, или тензорезистор) в перчатке необходимы для чтения сгиба пальцев пользователя. Управляющая плата подключена к ним через восьмиканальный мультиплексор с применением делителя напряжения. В качестве опорного напряжения подается 3,3В (с вывода платы - VCC) и в качестве опорных резисторов - используется 10 кОм т.к. датчики изгиба имеют максимальное сопротивление в 10 кОм.

Сигналы с датчиков изгиба также получаем используя АЦП управляющего микроконтроллера.



Рисунок 3.7 - Внешний вид датчики изгиба

Технические характеристики:

- сопротивление сенсора в не согнутом состоянии: $10 \pm 30\%$ кОм
- сопротивление сенсора в согнутом состоянии: 60 К ~ 110 К ом
- При изгибе (изменение сопротивления):
 - Номинальная мощность: 0,5 Вт;
 - Пиковая мощность: 1 Вт;
- Ресурс изгибов: > 1 миллион раз;
- Рабочая температура: -35 ~ + 80 градС;
- Размеры изгибаемой части, мин, Длина x Ширина: 100 x 6.35 мм.

Акселерометр и гироскоп (MPU6050) в КТ может применяться для обнаружения ориентации и положения в пространстве.

Управляющая плата подключается к нему через проводной интерфейс I2C. На схеме, питаем его от 3,3В с вывода VCC платы.

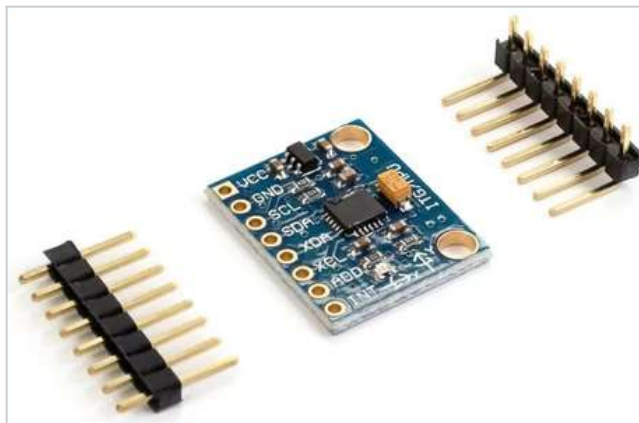


Рисунок 3.8 - Внешний вид MPU6050

Технические характеристики:

- Питание: **3,5 – 6 В**;
- Ток потребления: **500 мкА**;
- Акселерометр диапазон измерений: **$\pm 2 \pm 4 \pm 8 \pm 16g$** ,
- Гироскоп диапазон измерений: **$\pm 250 \ 500 \ 1000 \ 2000^\circ / s$** ,
- Интерфейс: **I2C**.

5 RGB светодиода на плате временно используются для имитации тепла и холода.

Микроконтроллер подключаем к ним через 16-ти канальный демультиплексор и через резисторы (для выводов VCC и GND и для вывода DATA - Ом). К RGB-светодиодам через демультиплексор подаем питание используя ЦАП управляющей платы. Напряжение ЦАПа платы варьируется (в идеале) от 0 до 3,3В.



Рисунок 3.9 - Внешний вид RGB-светодиодов

5 отдельные зеленые светодиоды предназначены для имитации степени надавливания (после касания) физического или виртуального объекта.

Они также подключаются к управляющей плате через отдельный демультимплексор (8-ми канальный) и резисторы (Ом).

Две кнопки на дополнительной плате перчатки являются универсальными и можно их программно настроить на свои усмотрения. Они подключены к микроконтроллеру с применением подтягивающих резисторов.

3.2.5. Составляющие узлы КТП

При реализации костюма телеприсутствия, в текущем начальном этапе, было решено создание костюма с кинестетической обратной связью с использованием приводов, потенциометров. Для это необходимо проанализировать движение частей тела, свободу и направление каждой части. Учитывая анатомию движения частей тела, активные части костюма телеприсутствия можно подразделить на область плечи (для плече-лопаточного сустава), локтя (локтевой сустав), кисти (лучезапястный сустав) и руки. В первых трех частях можно расположить по два привода которые будут сопротивляться движению создавая кинестетическую обратную связь. В качестве приводов можно использовать сервоприводы или шаговые двигатели меньшего размера. При этом, их силовые моменты должны быть достаточными чтобы сопротивляться движению.

Для понимания движения и вращения частей суставов тела необходимо также разобраться с осями тела человека. Существуют три вида осей:

- вертикальная ось
- сагиттальная ось
- фронтальная ось

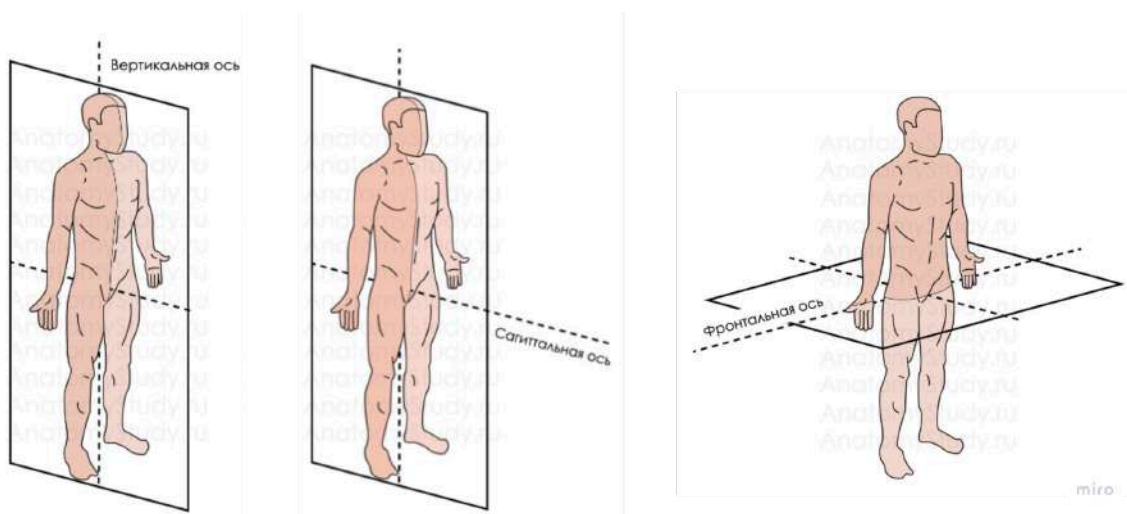


Рисунок 3.10 - Оси тела человека

Рассмотрим направление вращения сервоприводов расположенных на правой части плечи. Первый сервопривод плечи будет отвечать за вращения руки по фронтальной оси, чтобы пользователь смог поднять руку вперед-назад. Вторым, который расположен вслед за ним – вращение по сагиттальной оси, чтобы пользователь смог поднять руку вправо. В анатомии скелета человека движение этих частей называют шаровидной. Для дальнейшего описания и построения схем приводы будут отмечены как - сервопривод для правого плеча X, сервопривод для правого плеча Y. Двух приводов достаточно для базового вращения плечевого сустава человека (Циркумдукции).

Для базовых вращений частей локтевого сустава используются два сервопривода. Один отвечает за вращение предплечья (супинация и пронация предплечья), а второе – за сгибание и разгибание по фронтальной оси. Для дальнейшего описания и построения схем приводы будут отмечены как - сервопривод для правого локтя X, сервопривод для правого локтя Z.

Для движений лучезапястного сустава тоже можно использовать два - три сервопривода. Лучезапястный сустав совершает движение в сагиттальной (отведение и приведение кисти) и фронтальной (сгибание и разгибание) осях. Для дальнейшего описания и построения схем приводы будут отмечены как - сервопривод для правой кисти X, сервопривод для правой кисти Y.

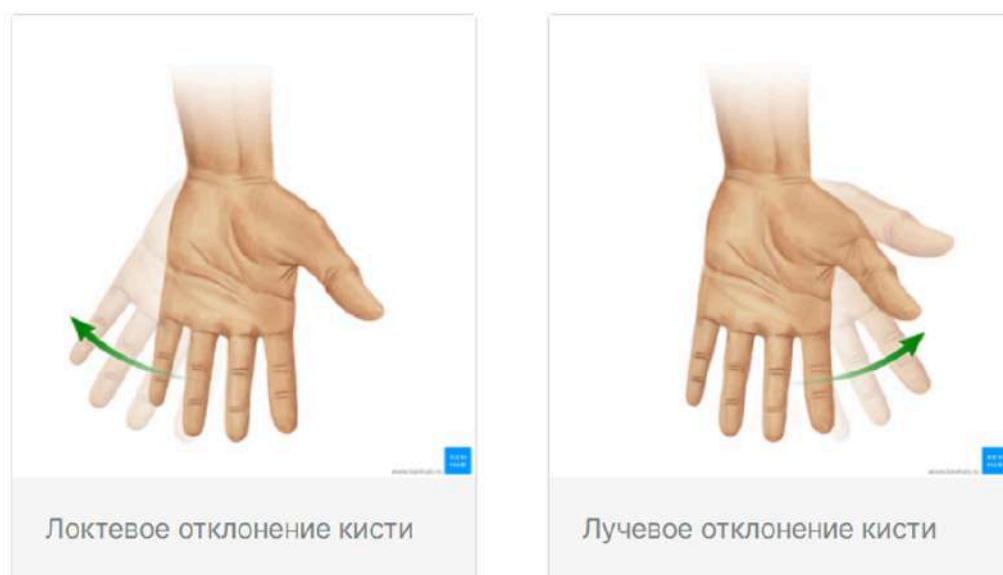


Рисунок 3.11 - Лучевое и локтевое отклонение кисти

Учитывая движения суставов плечи, локтя и кисти, была смоделирована, распечатана и собрана следующая модель костюма телеприсутствия:

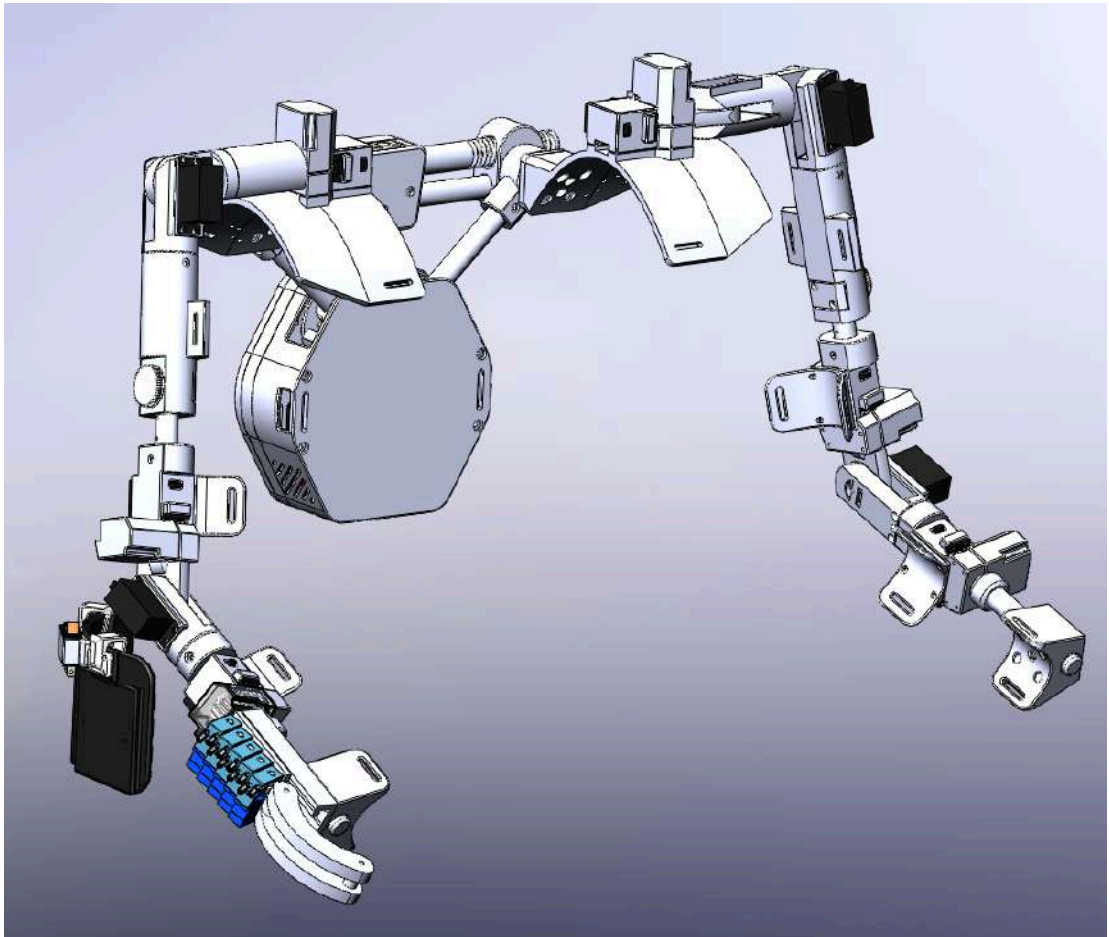


Рисунок 3.12 - Прототип костюма телеприсутствия KOLD

Данная модель состоит из нескольких частей:

- Сборка системы для вращения плечи;
- Сборка системы для вращения локтя;
- Сборка системы для вращения кисти;
- Сборка системы эксцентрика для регулировки положения наплечников;
- Сборка многоуровневого блока управления для крепления бортовой системы и дополнительных плат;
- Сборка с двумя отведенными местами для удобного подключения аккумуляторных блоков к бортовой системе и к сервоприводам через преобразователи.

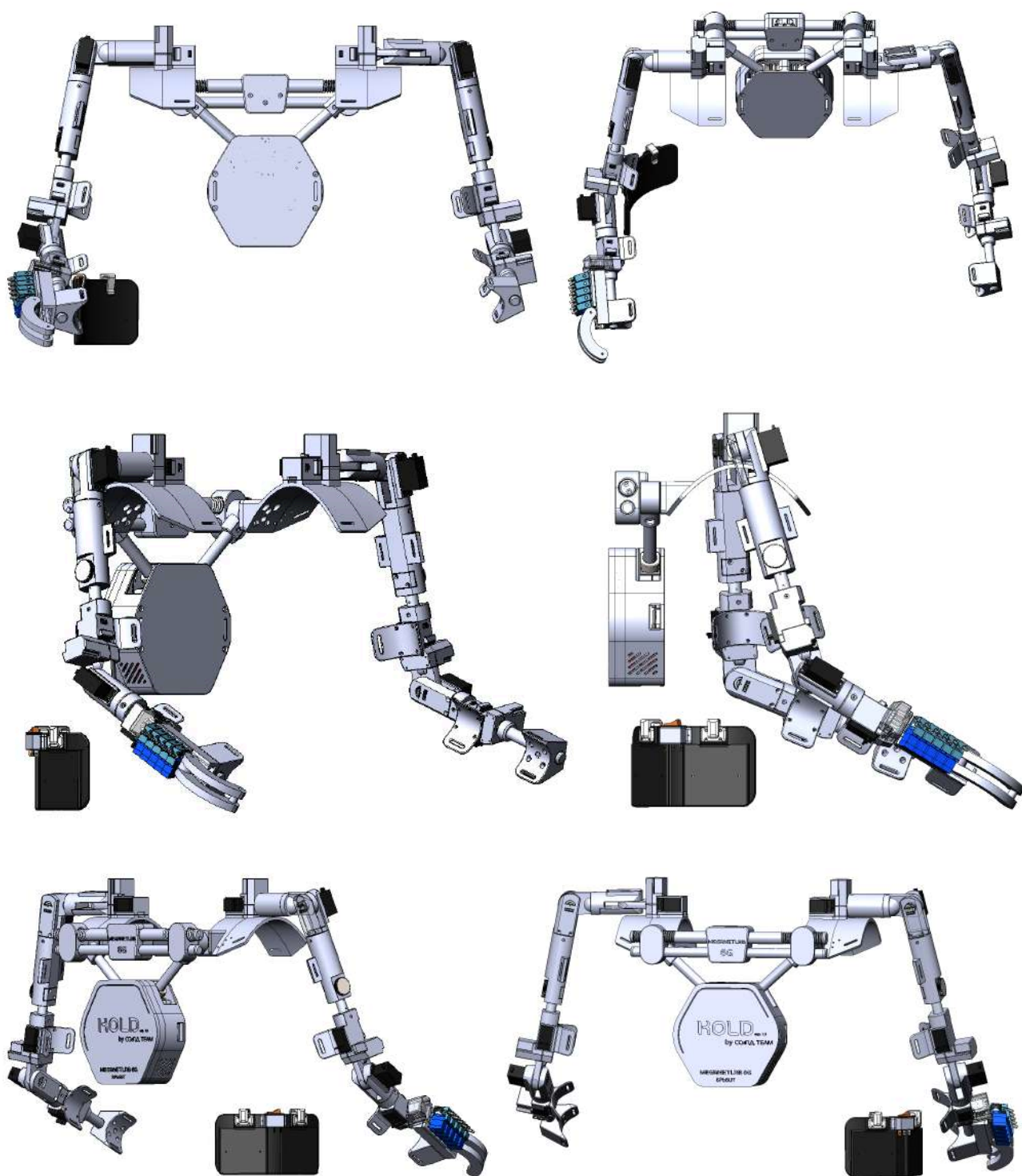


Рисунок 3.13 - Вид на сборку с разных сторон

В данной сборке сервоприводы расположены таким образом, чтобы пользователь смог выполнить базовые циркуляционные движения.

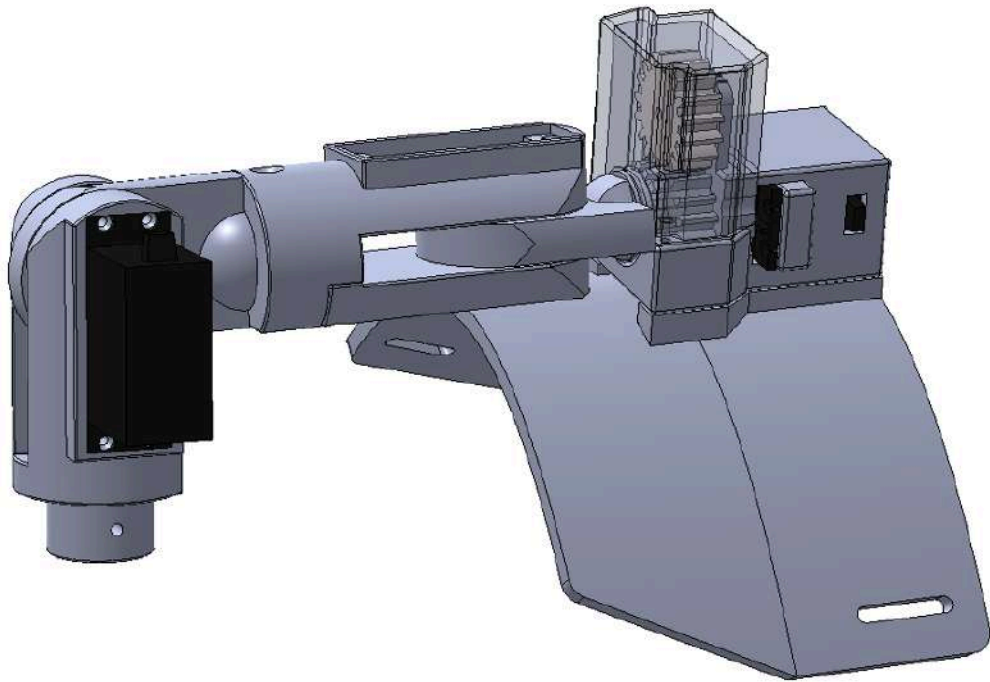


Рисунок 3.14 - Сборка системы для вращения плеч

Потенциометр в конструкции для вращения по фронтальной оси соединен к сервоприводу через шестерню, а тот потенциометр, который измеряет вращение по сагиттальной оси - расположен концентрично напротив вращающейся части сервопривода.

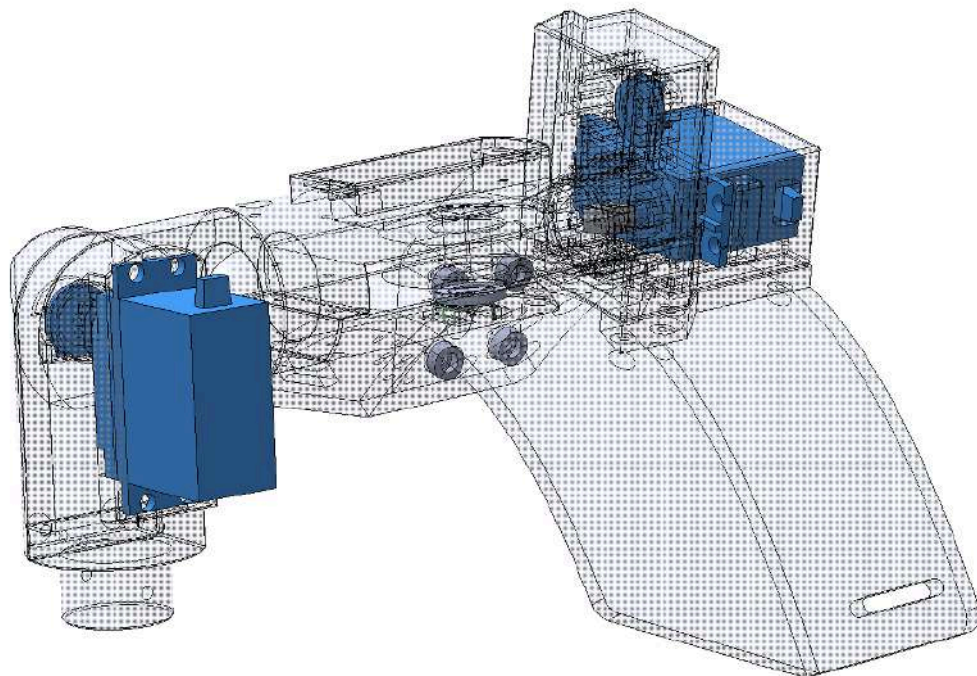


Рисунок 3.15 - Расположение потенциометров

Аналогичным образом была сконструирована система для вращения локтевого сустава и кисти.

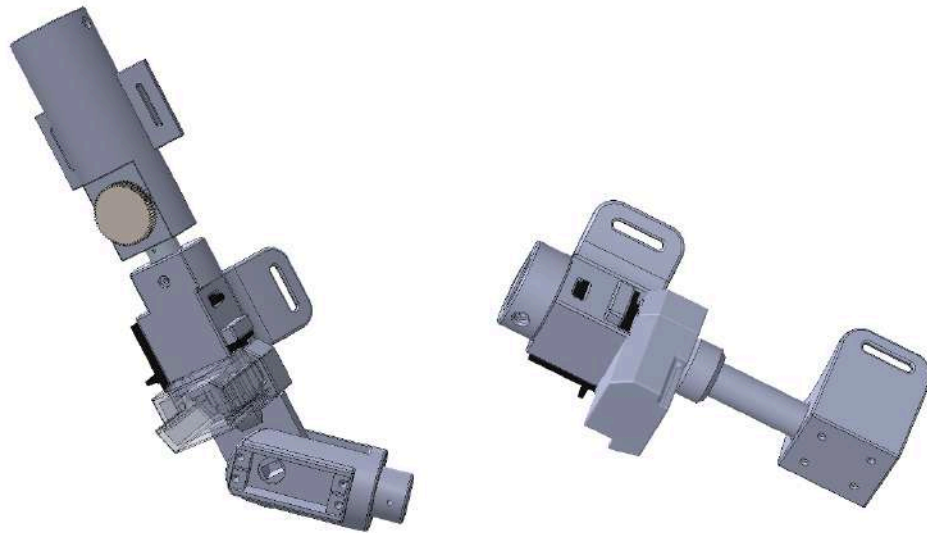


Рисунок 3.16 - Сборка систем для вращения локтя и кисти

В процессе моделирования, было создано специальная система эксцентрика, которую можно с легкостью регулировать.

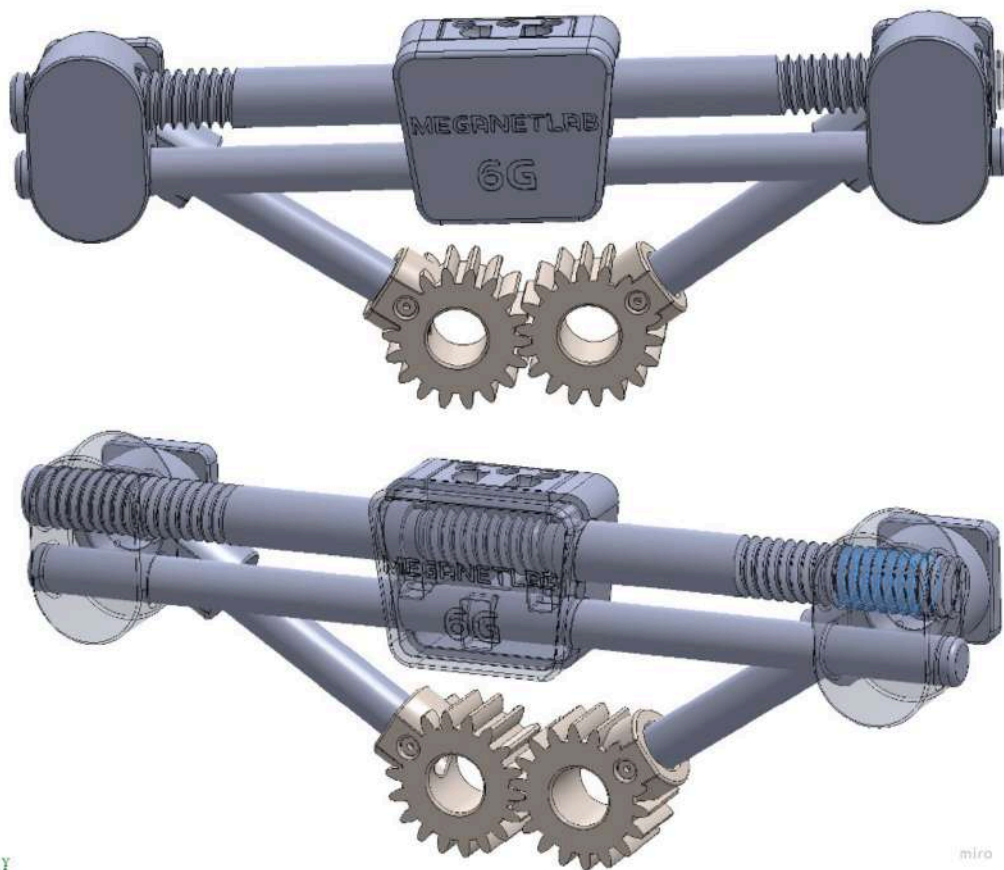


Рисунок 3.17 - Внешний и внутренний виды

Система блока управления состоит из двух уровней. На первый уровень можно крепить бортовой микрокомпьютер Raspberry Pi и одну плату с ESP32, а на второй – 6 плат с ESP32.



Рисунок 3.18 - Конструкция и расположение элементов

Были спроектированы слоты для крепления аккумуляторных блоков, в ней имеются два «крючка» для крепления к ремням одежды человека.



Рисунок 3.19 - Слоты для крепления аккумуляторного блока

3.2.6. Алгоритмы работы загруженных в МК программ и их описание

На используемых микроконтроллерах имеются 2 ядра и они могут работать асинхронно (параллельно).

В алгоритмах работы микроконтроллеров используются оба ядра. Это можно заметить на следующих схемах.

Процессы на микроконтроллерах ESP32 можно поделить на три составляющие:

- Начальная функция Setup. Запускается при запуске МК (микроконтроллера) и в ней заложены алгоритмы для подготовки и калибровки используемых элементов и функций. Она также запускает асинхронную циклическую функцию Low Priority Loop (на ядре 0) и, после выполнения функции Setup, программа переходит в циклическую функцию High Priority Loop (на ядре 1).
- Циклическая функция High Priority Loop (ядро #1). Используется для поддержки связи с MQTT сервером (брокером) [54-57] через беспроводной интерфейс WiFi. Она также применяется для отправки данных с сенсоров и получения данных для актуаторов. А также она, при желании, может отправить измеренные на обоих ядрах - задержки для отладки.
- Циклическая функция Low Priority Loop (ядро #0). Предназначена для работы с подключенными к ней элементами (модулями \ сенсорами и актуаторами) КТ.

Описание прошивки, загруженной в микроконтроллер ESP32 LG:

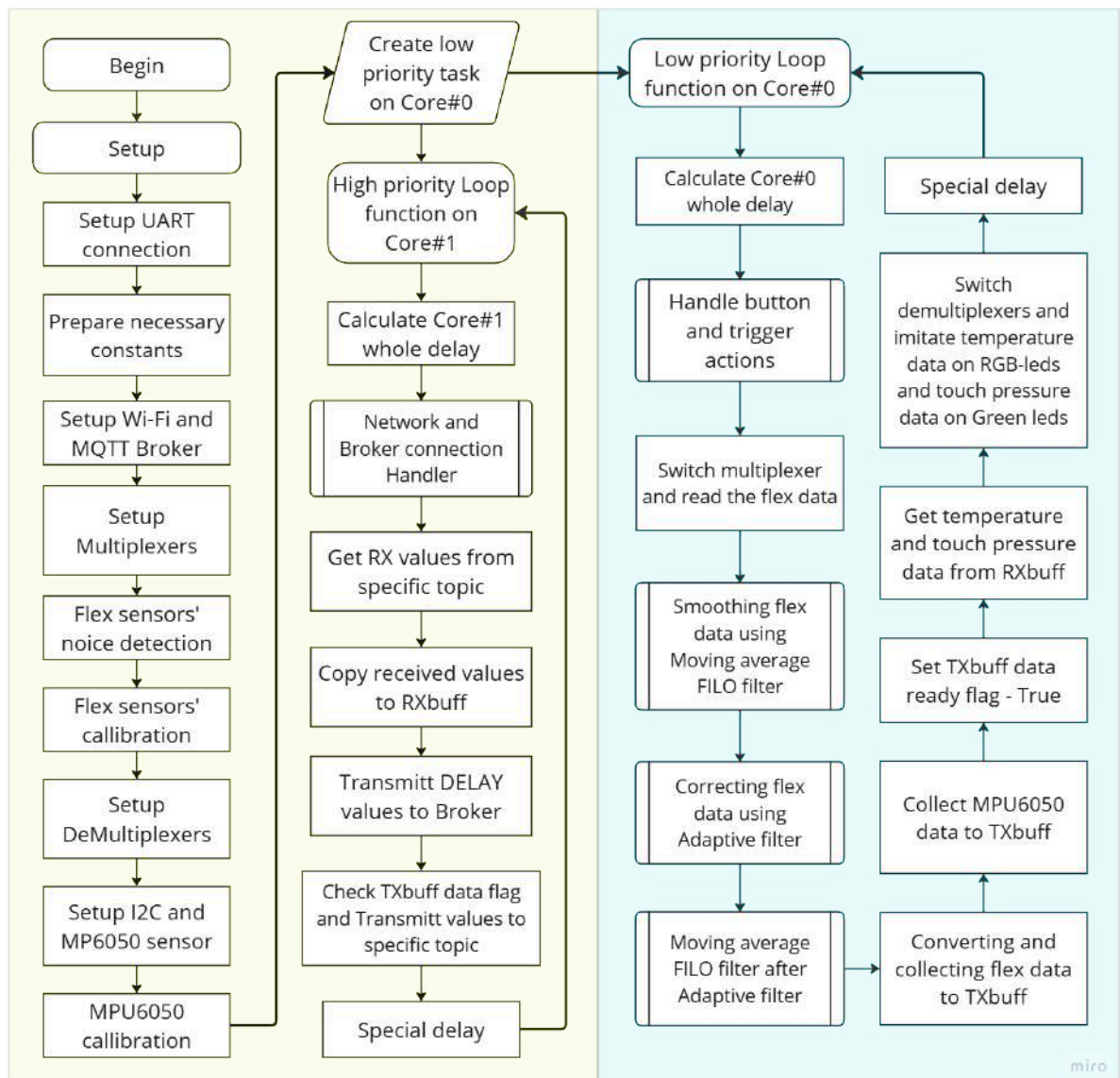


Рисунок 3.20 - Алгоритм работы кистей оператора КТ (микроконтроллер ESP32 LG, подключенной к перчатке)

При запуске устройства выполняется ряд подготовительных процессов (функция Setup):

- Настройка UART интерфейса - на этапе разработки устройства используется для отладки и взаимодействия платы с разработчиком.
- Подготовка необходимых констант и переменных для улучшения быстродействия в работе устройства
- Подключение к WiFi и к MQTT брокеру. А также подписка на необходимые топики для получения данных с них.
- Подготовка мультиплексоров для работы с датчиками изгиба
- Определение границ помех сенсоров, подключенных к АЦП для дальнейшего сглаживания

- Калибровка сенсоров, подключенных к АЦП
- Подготовка демультимплексоров для работы со светодиодами
- Настройка подключения к акселерометру и гироскопу (MPU6050) и его калибровка
- Создание дополнительной задачи (асинхронного цикла) с низким приоритетом на другом ядре микроконтроллера

После создания новой задачи у нас имеются два асинхронного цикла с разными приоритетами выполнения.

Так как необходимо обеспечить бесперебойную связь, задаче для взаимодействия с сетью ставим высокий приоритет. Цикл работы данной задачи состоит из следующих подзадач:

- Начало цикла (HighPriorityLoop Core#1)
- Вычисление задержки выполнения цикла данной задачи (для отладки; можно использовать для создания специальной задержки на данную задачу)
- Проверка соединения к беспроводной сети (WiFi) и к серверу (MQTT брокеру). А также выполнение некоторых задач синхронизации
- Получение принятых на указанный топик данных и их присвоение на переменную RXbuff
- Отправка вычисленной задержки циклов задач HighPriorityLoop и LowPriorityLoop запущенных на обоих ядрах
- Проверка готовности данных на TXbuff для отправки на сервер и отправка на указанный топик сервера
- (опционально) Удержание (задержка) данного цикла для предотвращения перегрузки сервера и для экономии энергии микроконтроллера
- Конец цикла

А цикл работы для взаимодействия с модулями, подключенных на микроконтроллер состоит из следующих подзадач:

- Начало цикла (LowPriorityLoop Core#0)
- Вычисление задержки выполнения цикла данной задачи (для отладки; можно использовать для создания специальной задержки на данную задачу)
- (опционально) Можно выполнить специальные команды клиентского или сетевого уровня при подключении кнопки (кнопок) на микроконтроллер. Нажатия можно персонализировать с применением специальной библиотеки.

Сервисные кнопки на перчатке пользователя

- Подцикл для работы с датчиком изгиба используя мультиплексор:
 - Чтение данных с датчиков изгиба
 - Трехэтапная обработка данных с аналоговых сенсоров:
 - Авторский алгоритм скользящего среднего с низкими коэффициентами

- Адаптивный фильтр с низкими коэффициентами
- (при необходимости) Авторский алгоритм скользящего среднего с низкими коэффициентами (для сглаживания ступенчатости Адаптивного фильтра)
 - Сбор обработанных данных аналоговых сенсоров в переменную TXbuff для отправки на указанный топик сервера.
- Получение данных о трехмерной скорости гироскопа (для вычисления ориентации в пространстве) и о трехмерного ускорения акселерометра (для вычисления положения в пространстве) и их сбор в переменную TXbuff для отправки на указанный топик сервера
- Установка флага готовности отправки данных с сенсоров на указанный топик сервера
- Подцикл для работы со светодиодами используя демультиплексоры:
 - **Температурная обратная связь.** Получение данных температурной обратной связи с переменной RXbuff и их имитация на RGB-светодиодах
 - **Хаптическая обратная связь.** Получение данных степени давления нажатия с переменной RXbuff и их имитация на зеленых светодиодах
- (опционально) Удержание (задержка) данного цикла для предотвращения перегрузки сервера и для экономии энергии микроконтроллера
- Конец цикла

Описание прошивки, загруженной в микроконтроллерах ESP32 L1, L2, R1, R2:

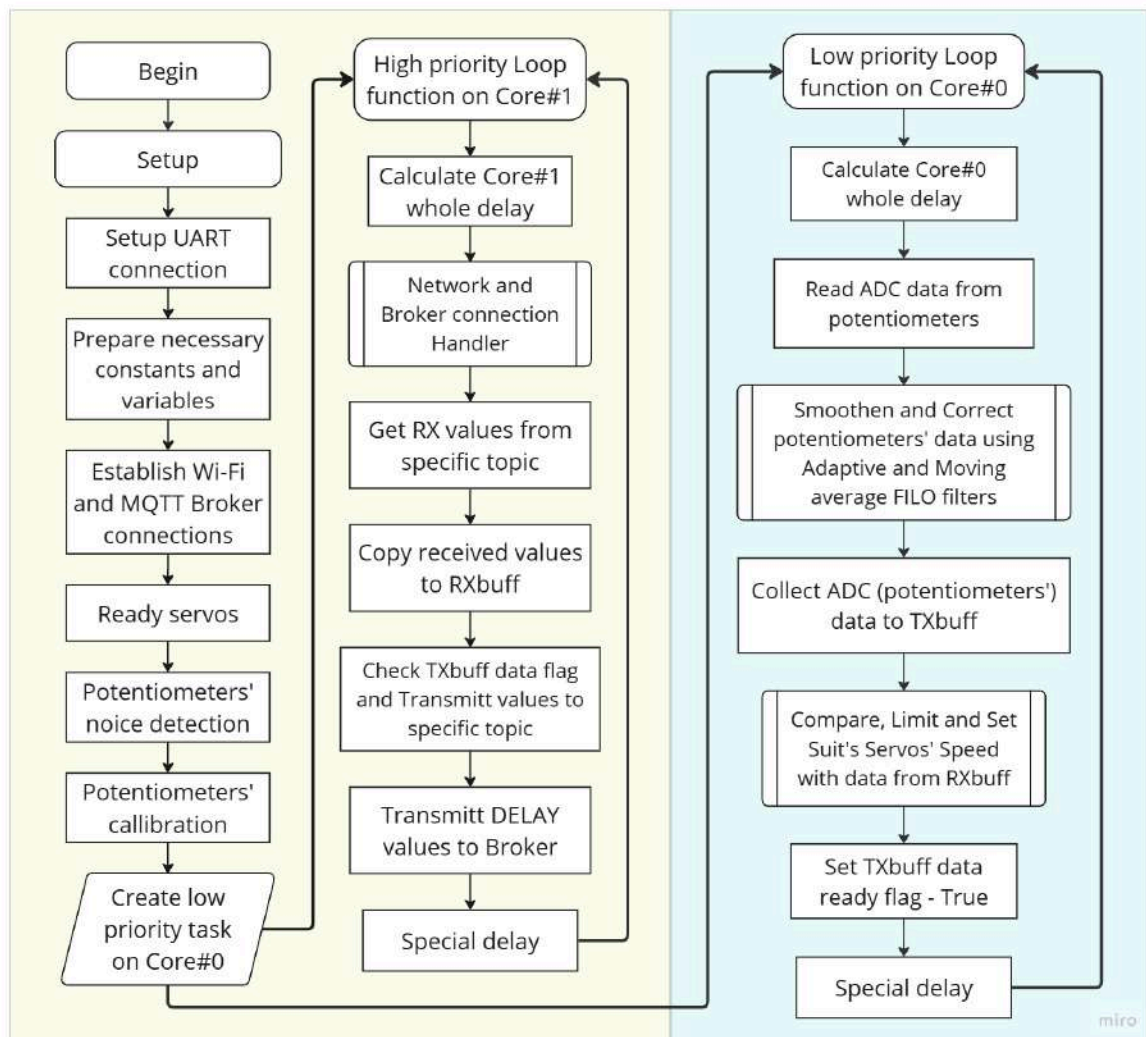


Рисунок 3.21 - Общий алгоритм работы верхних конечностей оператора костюма телеприсутствия

Алгоритм работы верхних конечностей оператора схож с алгоритмом для кистевой части. Но имеются несколько различий:

- В качестве сенсоров используются потенциометры для чтения угла (скорости вращения) сервоприводов
- Цикл задачи для работы с подключенными модулями работает с наименьшей задержкой за счет меньшего количества подключенных модулей
- Цикл для взаимодействия с сетью также выполняется за наименьшее время за счет наименьших данных для обмена между сервером и микроконтроллером (клиентом)
- Также, для обмена данных используются другие топики на сервере

Если подробно, то:

При запуске устройства выполняется ряд подготовительных процессов (функция Setup):

- Настройка UART интерфейса - на этапе разработки устройства используется для отладки и взаимодействия платы с разработчиком.
- Подготовка необходимых констант и переменных для улучшения быстродействия в работе устройства
- Подключение к WiFi и к MQTT брокеру. А также подписка на необходимые топики для получения данных с них.
- Подготовка сервоприводов
- Определение границ помех потенциометров, подключенных к АЦП для дальнейшего сглаживания
- Калибровка потенциометров, подключенных к АЦП
- Создание дополнительной задачи (асинхронного цикла) с низким приоритетом на другом ядре микроконтроллера

После создания новой задачи у нас имеются два асинхронного цикла с разными приоритетами выполнения.

Так как необходимо обеспечить бесперебойную связь, задаче для взаимодействия с сетью ставим высокий приоритет. Цикл работы данной задачи состоит из следующих подзадач:

- Начало цикла (HighPriorityLoop Core#1)
- Вычисление задержки выполнения цикла данной задачи (для отладки; можно использовать для создания специальной задержки на данную задачу)
- Проверка соединения к беспроводной сети (WiFi) и к серверу (MQTT брокеру). А также выполнение некоторых задач синхронизации
- Получение принятых на указанный топик данных и их присвоение на переменную RXbuff
- Проверка готовности данных на TXbuff для отправки на сервер и отправка на указанный топик сервера
- Отправка вычисленной задержки циклов задач HighPriorityLoop и LowPriorityLoop запущенных на обоих ядрах
- (опционально) Удержание (задержка) данного цикла для предотвращения перегрузки сервера и для экономии энергии микроконтроллера
- Конец цикла

А цикл работы для взаимодействия с модулями, подключенных на микроконтроллер состоит из следующих подзадач:

- Начало цикла (LowPriorityLoop Core#0)
- Вычисление задержки выполнения цикла данной задачи (для отладки; можно использовать для создания специальной задержки на данную задачу)
- Подцикл для работы с потенциометрами, встроенных в сервоприводы:

- Считывание аналоговых данных с потенциометров
- Трехэтапная обработка данных с аналоговых сенсоров:
 - Авторский алгоритм скользящего среднего с низкими коэффициентами
 - Адаптивный фильтр с низкими коэффициентами
 - (при необходимости) Авторский алгоритм скользящего среднего с низкими коэффициентами (для сглаживания ступенчатости Адаптивного фильтра)
- Сбор обработанных данных аналоговых сенсоров в переменную TXbuff для отправки на указанный топик сервера.
- Сравнение, ограничение и установка состояния скорости вращения сервоприводов с применением полученных в переменную RXbuff данных (с сервера)
- Установка флага готовности отправки данных с сенсоров на указанный топик сервера
- (опционально) Удержание (задержка) данного цикла для предотвращения перегрузки сервера и для экономии энергии микроконтроллера
- Конец цикла

3.3. Структура и назначение используемых выводов управляющих плат

Описанные в главе 7.6.1, элементы подключены к дополнительным платам. А дополнительные платы подключены к управляющим платам через длинные собранные в одну шину - провода.

Длина проводов, названия подключенных выводов и другие визуальные данные отмечены в схематичной картинке ниже:

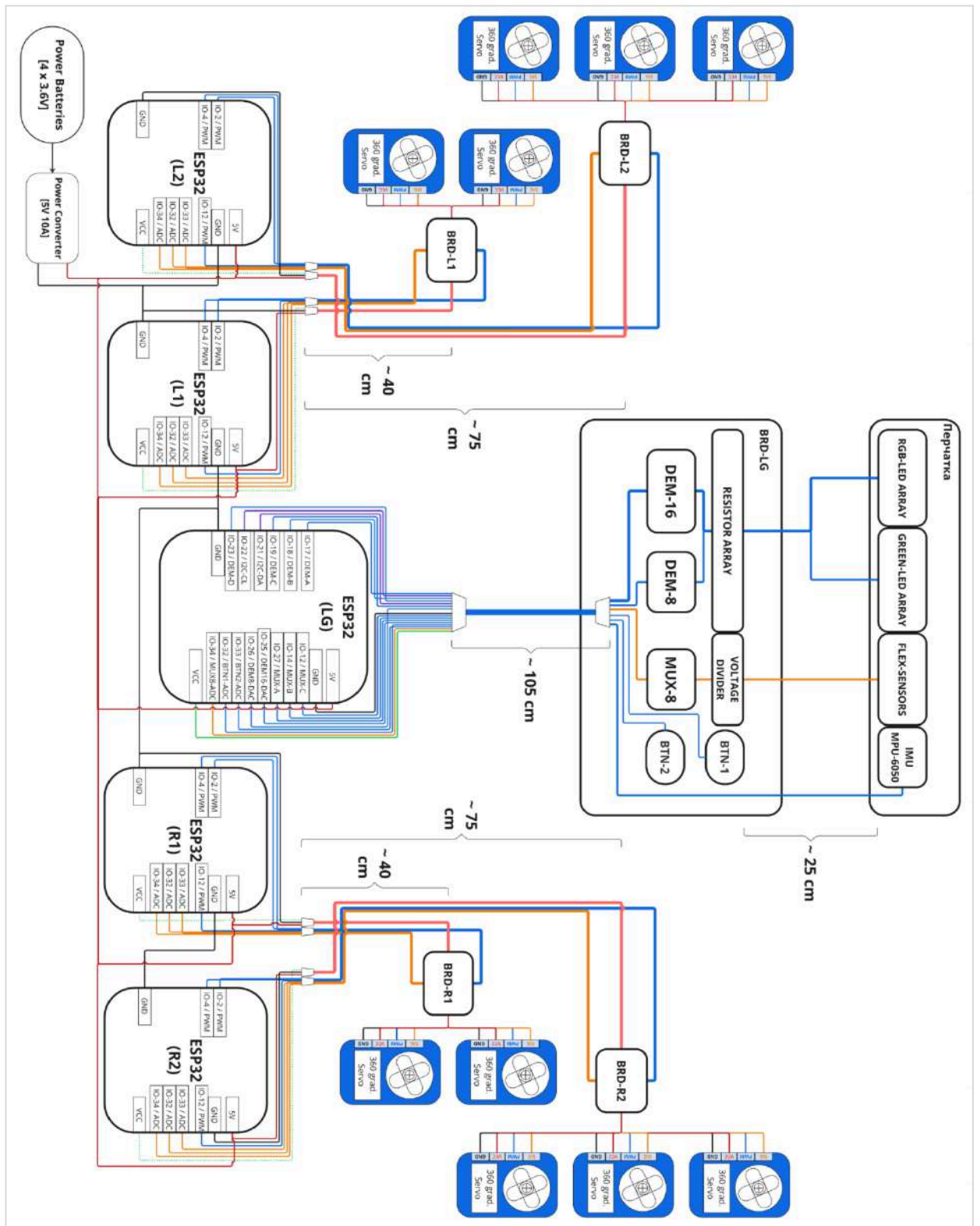


Рисунок 3.22 - Схема соединения элементов костюма телеприсутствия (КТ)

На схеме можно заметить элемент BRD.

Они являются дополнительными платами для облегчения подключения сервоприводов к платам ESP-32 L1, L2, R1, R2.

К одной из них можно подключить 3 сервопривода и 3 потенциометра, и она разветвляет выводы питания и земли на подключенные к нему потенциометры и сервоприводы.

Обозначения выводов на микроконтроллерах L1, L2, R1, R2 (костюм телеприсутствия, КТ):

- PWM - выводы с которых подаем ШИМ сигналы на сервоприводы
- ADC - АЦП выводы с помощью которых получаем аналоговые сигналы с потенциометров, встроенных в сервоприводы
- 5V - вывод питания платы от 5В. Сервоприводы питаются не от этого вывода, а от внешнего источника питания
- GND - вывод земли
- VCC - вывод 3.3В платы. Можно применить для питания некоторых сенсоров. Раньше (на предыдущей версии) на этих платах применялись для питания потенциометров отдельных от сервоприводов (которые не встроены в сервоприводы).

В текущей версии подключенные к ним провода остались на месте, но не используются. Поэтому их линии на схеме начерчены точками

Обозначения выводов на микроконтроллере LG (перчатка):

- 5V - вывод питания платы от 5В.
- GND - вывод земли
- VCC - вывод 3.3В платы. Используется для питания модуля акселерометра и гироскопа
- I2C-DA (I2C data) и I2C-CL (I2C clock) - выводы для общения с модулем акселерометра и гироскопа.
I2C-DA используется для обмена данными, а I2C-CL применяется для синхронизации передаваемых сигналов.
- BTN2-ADC и BTN1-ADC - выводы подключенные к кнопкам на дополнительной плате перчатки
- MUX8-ADC - используется для получения аналоговых сигналов с датчиков изгиба через мультиплексор. Данный вывод является выходным выводом 8 канального мультиплексора (MUX-8)
- MUX-A, MUX-B и MUX-C - три вывода для подключения мультиплексора (MUX-8) подключенной к датчикам изгиба
- DEM16-DAC - используется для подачи аналоговых сигналов/напряжений на RGB-светодиоды для имитации тепла и холода на пальцах перчатки. Данный вывод является входным выводом 16 канального демультимплексора (DEM-16)

- DEM8-DAC - используется для подачи аналоговых сигналов/напряжений на зеленые-светодиоды для имитации степени давления после нажатия на пальцах перчатки. Данный вывод является входным выводом 8 канального демультиплексора (DEM-8)
- DEM-A, DEM-B, DEM-C и DEM-D - три вывода для подключения демультиплексоров (DEM-8 и DEM-16) подключенных к RGB и к зеленым светодиодам

Если рассмотреть данные выводы по подробнее:

- **LG:**
 - **Цифровые выходы** (для управления мультиплексорами и демультиплексорами):
 - GPIO-17 / DEM-A
 - GPIO-18 / DEM-B
 - GPIO-19 / DEM-C
 - GPIO-23 / DEM-D
 - GPIO-12 / MUX-C
 - GPIO-14 / MUX-B
 - GPIO-27 / MUX-A
 - **Цифровые входы** (для двух кнопок на дополнительной плате BRD-LG):
 - GPIO-33 / BTN2
 - GPIO-32 / BTN1
 - **Выводы для интерфейса I2C** (для работы/общения с акселерометром и гироскопом):
 - GPIO-21 / I2C-DA (I2C Data: цифровой вход-выход)
 - GPIO-22 / I2C-CL (I2C Clock: цифровой выход, тактирование)
 - **Аналоговый выход с ЦАП интерфейса** (для подачи сигналов на светодиоды через демультиплексоры):
 - GPIO-25 / DEM16-DAC
 - GPIO-26 / DEM8-DAC
 - **Аналоговый вход на АЦП интерфейс** (для сигналов с датчиков изгиба через мультиплексор):
 - GPIO-34 / MUX8-ADC
 - **Выводы питания:**
 - 5V - питание микроконтроллера
 - GND - земля
 - VCC - вывод напряжения 3.3В с микроконтроллера

- **L1, L2, R1, R2:**

- **ШИМ выходы** (для подачи сигналов на сервоприводы):
 - GPIO-2 / PWM
 - GPIO-4 / PWM
 - GPIO-12 / PWM (для **L1** и **R1**: на жгутах имеются подключенные провода, но к ним не подключены сервоприводы)
- **Аналоговые входы на АЦП интерфейс** (для сигналов с потенциометров):
 - GPIO-33 / ADC
 - GPIO-32 / ADC
 - GPIO-34 / ADC
- **Выводы питания:**
 - 5V - питание микроконтроллера
 - GND - земля
 - VCC - вывод напряжения 3.3В с микроконтроллера

Вышеописанные выводы соединяются с сенсорами и актуаторами КТ с применением жгутов. Их содержание:

К дополнительной плате BRD идут два набора проводов (жгутов).

Один - для питания, в которой содержатся три вывода:

- 3.3В - с вывода питания VCC платы ESP32;
- 5В - с внешнего источника питания;
- земля - общая земля.

Другой - для управления сервоприводами (генерируя ШИМ сигналы на платах ESP32) и получения аналоговых сигналов с потенциометров:

- PWM [1...3] - провода, через которые подаются ШИМ сигналы с ESP-32 на используемые сервоприводы
- ADC [1...3] - провода, через которые проходят аналоговые сигналы с встроенных на сервоприводы потенциометров на АЦП интерфейс плат ESP-32.

На схеме имеется также и другая дополнительная плата BRD-LG для перчатки. В ней находятся вспомогательные для коммутации имитируемых обратной связи элементов и она позволяет уменьшить количество проводов в жгуте направленных с платы перчатки ESP-32 LG на саму перчатку.

Она содержит в себе:

- 8-ми канальный мультиплексор с резисторами (для делителя напряжения) для получения данных с 5-ти датчиков изгиба
- 8-ми канальный демultipлексор с резисторами в которую подключены зеленые светодиоды - для имитации воспроизведения давления нажатия на виртуальный объект

- 16-ти канальный демультиплексор с резисторами в которую подключены RGB-светодиоды - для имитации воспроизведения температуры виртуального объекта
- две кнопки
- контакты проводов акселерометра и гироскопа

Для работы со всеми этими элементами имеется коннектор, в которую можно подключить специальный жгут, в котором имеются необходимые для этих элементов провода.

На этом специальном жгуте имеются 16 проводов. Они нужны для следующих выводов:

- 4 выводов 8-ми канального мультиплексора. Один из них - для аналоговых сигналов с датчиков изгиба на АЦП интерфейс ESP-32 LG, а три - для управления мультиплексора.
 - цифровой выход с ESP32-LG (GPIO-27) для управления первого переключателя (MUX-A)
 - цифровой выход с ESP32-LG (GPIO-14) для управления второго переключателя (MUX-B)
 - цифровой выход с ESP32-LG (GPIO-12) для управления третьего переключателя (MUX-C)
 - аналоговый вход на АЦП интерфейс ESP-32 LG (GPIO-34)
- 4 выводов 8-ми канального демультиплексора. Один из них - для подачи аналогового напряжения с ЦАП интерфейса ESP-32 LG на зеленые светодиоды, а три - для управления демультиплексора.
 - цифровой выход с ESP32-LG (GPIO-17) для управления первого переключателя (DEM-A)
 - цифровой выход с ESP32-LG (GPIO-18) для управления второго переключателя (DEM-B)
 - цифровой выход с ESP32-LG (GPIO-19) для управления третьего переключателя (DEM-C)
 - аналоговый выход с ЦАП интерфейса ESP-32 LG (GPIO-26)
- 5 выводов 16-ми канального демультиплексора. Один из них - для подачи аналогового напряжения с ЦАП интерфейса ESP-32 LG на RGB-светодиоды, а четыре - для управления демультиплексора.
 - цифровой выход с ESP32-LG (GPIO-17) для управления первого переключателя (DEM-A)
 - цифровой выход с ESP32-LG (GPIO-18) для управления второго переключателя (DEM-B)
 - цифровой выход с ESP32-LG (GPIO-19) для управления третьего переключателя (DEM-C)

- цифровой выход с ESP32-LG (GPIO-23) для управления четвертого переключателя (DEM-D)
- аналоговый выход с ЦАП интерфейса ESP-32 LG (GPIO-25)
- 2 вывода для кнопок
 - цифровой вход на ESP32-LG (GPIO-32) для 1-й кнопки
 - цифровой вход на ESP32-LG (GPIO-33) для 2-й кнопки
- 2 выводов для работы с акселерометром и гироскопом
 - I2C Data (GPIO-21)
 - I2C Clock (GPIO-22)
- 2 вывода для питания
 - 3.3В (VCC)
 - Земля (GND)

Следует отметить что выводы управления (их 3) 8-ми канального демультиплексора подключены к выводам управления 16-ти канального демультиплексора (их 4) для параллельного управления этими демультиплексорами и для уменьшения количества выводов с ESP-32 LG на дополнительную плату BRD-LG. По этой причине, количество проводов в жгуте не 19, а 16.

3.4. Используемые интерфейсы на КТП для контроля и мониторинга применяемых сенсоров и актуаторов

Мониторинг и контроль модулей КТП происходит с применением интерфейсов АЦП, ЦАП, ШИМ и I2C.

Интерфейсы АЦП и ЦАП.

ЦАП на микроконтроллере ESP32 LG (перчатка) используется для подачи аналоговых сигналов на:

- RGB-светодиоды на перчатке для имитации воспроизведения температуры виртуального объекта
- зеленые светодиоды на перчатке для имитации степени давления нажатия на виртуальный объект

Подавая значение между 0 и 255 мы меняем напряжение на выходе ЦАП от 0 до 3.3В (примерно от 0.1 до 3.2В).

АЦП на микроконтроллерах используются для получения аналоговых сигналов с:

- датчиков изгиба на перчатке (с применением делителя напряжения)
 - потенциометров (встроенных в сервоприводы) на костюме телеприсутствия
- АЦП и ЦАП сигналы варьируются в диапазоне от 0В до 3.3В (в идеале).

В реальности: примерно от 0.1В до 3.2В.

Параметры ШИМ для сервоприводов: Амплитуда, Частота и Скважность.

ШИМ сигналы формируются на управляющих микроконтроллерах ESP32 L1, L2, R1 и R2, на выводах GPIO-2, GPIO-4 и GPIO-12 этих микроконтроллеров. Параметры ШИМ сигналов для сервоприводов [30]:

- Так как напряжение логики управляющих микроконтроллеров ESP32 равна 3.3В, то **амплитуда** генерируемого ШИМа также равна **3.3В**.
- Из документации используемого сервопривода можно понять что сервопривод принимает ШИМ сигналы с периодом 2000 мкс (микросекунд) или 0,002 в секундах.

Из этого можно вычислить что **частота** генерируемого ШИМ сигнала равняется **500 Гц**

- **Скважность.** Исходя из документации сервопривода, для управления сервоприводом необходимо подать импульсы t_H с длительностью:
 - $t_H = 1500 \dots 2000$ мкс для вращения по часовой стрелке
 - $t_H = 1500 \pm 20$ мкс для остановки сервопривода
 - $t_H = 500 \dots 1500$ мкс для вращения против часовой стрелки

Скважность для данных импульсов можно вычислить по формуле взятого с сайта ChipDip:

$$S = T / t_H$$

Диапазоны скважности ШИМ сигналов:

- $S = 1.33(3)$ [при $t_H = 1500$] ... 1.001 [при $t_H = 1999$] для вращения по часовой стрелке
- $S = 1.351$ [при $t_H = 1480$] ... 1.316 [при $t_H = 1520$] для остановки сервопривода
- $S = 4$ [при $t_H = 500$] ... $1.33(3)$ [при $t_H = 1500$] для вращения против часовой стрелки

3.5. Электромагнитная совместимость КТП

После проведения экспериментов и вычислений над КТП, были получены следующие результаты.

3.5.1. Наиболее уязвимые для помех структурные элементы костюма телеприсутствия

Основные структурные элементы рассматриваемого костюма телеприсутствия включают:

- микроконтроллеры ESP32;
- сервоприводы FB5317M-360;
- решетки светодиодов;
- датчики изгиба;
- мультиплексор;
- демультимплексоры.

Помехи между проводами, соединяющими эти элементы, по уровню относительно небольшие. Однако они могут оказывать существенное влияние на работу чувствительных элементов костюма, поступая на их входы вместе с полезными сигналами, предназначенными для обработки или управления работой этих элементов. Помехи могут также искажать сигналы, передаваемые на аватар и тем самым приводить к ошибочным действиям последнего.

Мультиплексор объединяет на выходе сигналы, поступающие на его вход по каналам пальцев перчатки. Демультимплексор решает обратную задачу: распределяет сигнал, поступающий на его вход по каналам пальцев перчатки. Управляющими сигналами являются адреса каналов, которые представлены 3-х или 4-х битовыми бинарными кодами. Уровни управляющих сигналов составляют единицы вольт и намного превосходят рассматриваемые помехи, а потому влияние помех на переключение каналов практически ничтожно.

То же самое можно сказать о ШИМ сигналах, поступающих с микроконтроллеров на сервоприводы. Амплитуда импульсов ШИМ сигналов составляет примерно 3 В и помехи между проводами практически не влияют на ширину импульсов ШИМ сигнала, которая определяет скорость и направление вращения двигателей сервоприводов.

Что касается решеток светодиодов, то поскольку они только качественно отображают информацию (температуру и давление), поступающую от аватара в костюм телеприсутствия, в отсутствие количественных оценок рассматривать влияние помех на цвет и интенсивность свечения светодиодов не приходится.

Таким образом, исследованию на влияние помех, создаваемых внутренними источниками, на качество полезных сигналов подлежат, в первую очередь, аналоговые сигналы, передаваемые костюмом телеприсутствия аватару. Эти сигналы поступают с выходов сервоприводов и мультиплексора на АЦП микроконтроллеров ESP32. Помеха, превышающая интервал квантования АЦП, как указывалось ранее, неизбежно приведет к передаче ошибочного значения полезного сигнала, что, в свою очередь, вызовет неверные действия руки и/или пальцев аватара. Чем меньше интервал квантования, тем выше вероятность ошибки. Повышение интервала квантования менее точно воспроизводит на приемном конце передаваемый сигнал. Так что с точки зрения точности воспроизведения полезного сигнала выбор числа уровней его квантования является компромиссной задачей. Однако входы АЦП микроконтроллеров ESP32 и провода, подводящие к ним сигналы, являются наиболее уязвимыми для помех точками структурных элементов костюма телеприсутствия. Здесь следует заметить, что окончательное качество сигнала, принимаемого аватаром, будет зависеть не только от искажений, полученных при его формировании, но и от отсутствия или наличия внешних помех и их мощности при передаче сигнала аватару.

3.5.2. Рекомендации по снижению уровней внутренних помех радиоэлектронным элементам костюма телеприсутствия с целью обеспечения их электромагнитной совместимости

Если в результате проведенной оценки уровней внутренних помех, связанных с взаимодействием электрических и магнитных полей проводов, несущих полезные сигналы, уровень помехи превышает допустимый уровень, то

необходимо принять меры по снижению уровня помехи. К числу этих мер относятся:

- уменьшение толщины жгута, в котором имеют место помехи. В случае, если несколько проводов создают в каком-то проводе недопустимую помеху, то часть этих проводов или все они, или провод, в котором эта помеха образуется, следует перенести из данного жгута в другой или образовать новый жгут, содержащий меньшее число проводов, а полученные новые жгуты разнести по расстоянию;
- экранирование проводов. Обычно эффективность экранирования рассматривается по отношению к гармоническим помехам. Рассмотренные выше помехи имеют импульсный характер. В представлении импульсных последовательностей рядом Фурье первая гармоника имеет наибольшую амплитуду и наименьшую частоту. Поэтому подавление помехи на этой частоте рассматривается в первую очередь;
- экранирование может быть использовано как для снижения уровня электрических, так и магнитных помех. При использовании экранированных проводов большое значение имеет правильное заземление экрана. Экран не эффективен, если он не заземлен должным образом;
- для хорошего экранирования электрического поля необходимо [3]:
 - минимизировать длину центрального проводника, который выступает за пределы экрана;
 - обеспечить хорошее заземление экрана. Если длина провода не превышает одной двенадцатой части длины волны, то заземления достаточно выполнить в одной точке. Для более длинного провода может понадобиться множественное заземление;
- немагнитный экран, заземленный на одном конце провода, не дает никакого подавления магнитного поля. Частота среза частотной характеристики экрана $\omega_s = R_s/L_s$, где R_s и L_s сопротивление и индуктивность экрана соответственно. На низких частотах, т.е. частотах ниже ω_s , подавление

магнитного поля отсутствует даже при правильном заземлении экрана. Экран начинает подавлять магнитные поля, частота которых превышает частоту среза экрана. Экран может быть использован как для экранирования источника помехи, так и для экранирования рецептора помехи.

При экранировании магнитного поля источника помехи, чтобы воспрепятствовать распространению этого поля от нагруженного проводника, экран следует заземлять на обоих концах провода. Это обеспечивает экранирование магнитного поля на частотах выше частоты среза. Еще один способ снижения магнитного поля в окружении, состоит в использовании скрученных проводов – прямого и обратного. Или экранированной пары скрученных проводов.

Лучшим способом защиты рецептора от магнитных помех является уменьшение области контура рецептора. Под этой областью понимается область, которую охватывает ток, текущий в цепи рецептора. Заземление экранированного провода с двух концов снижает размеры этой области.

В качестве примера рассмотрим зависимость уровня наведенных помех от значения коэффициента заполнения в проводе, который несет полезный сигнал при следующих условиях.

**** Провод источник помех ****

Радиус провода, мм: 1,0

Сопротивление нагрузки, кОм: 5,0

Напряжение источника помех, В: 3,0

Частота импульсов, имп/сек: 5000

Расстояние до провода-рецептора, мм: 4,0

Длина провода, м: 1,00

Относительная диэлектрическая проницаемость оболочки: 3,0

Число проводов-помех: 2

**** Провод рецептор помех ****

Радиус, мм: 0,5

Сопротивление нагрузки, кОм: 1,0

Диапазон уровня полезного сигнала, В: 3,0

Емкость между землей и проводом, пФ: 10,0

Полоса пропускания, кГц: 10,0

Число уровней квантования: 256

Обозначим U_E и U_H напряжения, наведенные в проводе рецепторе помех соответственно электрическим и магнитным полем проводов источников помех.

Значение коэффициента заполнения τ/T , где τ длительность, а T период импульсов, изменяется в рассматриваемом случае от 0,25 до практически единицы (0,999). Длительность ШИМ сигналов изменяется от 500 мкс до 1999 мкс, а период равен 2000 мкс.

Результат расчетов зависимости U_E и U_H от отношения τ/T по программе, реализующей приведенные в данном отчете соотношения, представлен в приводимой ниже таблице.

Таблица 3.2. Зависимость помех, наведенных в проводе рецепторе, от коэффициента заполнения

τ/T	0,25	0,28	0,30	0,35	0,40	0,47	0,50	0,55	0,60	0,65	0,75	0,80	0,90	0,95	1,0
U_E , мВ	7,145	9,275	8,214	1,323	0,001	5,345	4,185	0,30	0,000	3,375	1,23	0,000	0,663	1,93	0,000
U_H , мкВ	5,036	6,537	5,789	0,932	0,000	3,767	2,950	0,21	0,000	2,379	0,86	0,000	0,487	1,36	0,000

Из таблицы видно, что помехи растут с уменьшением коэффициента заполнения. Помеха, вызванная электрическим полем в жгуте значительно превосходит помеху, связанную с магнитным полем (милливольты против микровольтов), что вообще-то и следовало ожидать. Учитывая диапазон уровня полезного сигнала $V = 3$ В и число уровней квантования $N = 256$, можно найти $\Delta V = V/N = 11,719$ мВ $< (U_E)_{\max} = 9,275$ мВ, т.е. по выбранному критерию помеха, вызванная электрическим полем соседних проводов, является неопасной. Однако очевидно, что увеличение числа уровней квантования, например, в два раза или уменьшение диапазона передаваемого сигнала в два раза при неизменных остальных параметрах рассматриваемого примера приведут к тому, что помеха U_E станет недопустимой. В этом случае использование немагнитного экрана для провода рецептора помех может оказаться полезным.

В заключение отметим еще раз, что окончательное качество передаваемых сигналов определяется всей совокупностью искажений, имеющих место как

внутри костюма телеприсутствия, так и вне его при передаче сигналов удаленному пользователю.

3.6. Расчет автономности КТП

Для расчета времени автономной работы костюма телеприсутствия необходимо знать количество потребляемого тока на каждом его элементе в разных состояниях. А для теоретического расчета минимальной автономной работы устройства можно используются значения максимального потребления тока каждого модуля.

В разработанном костюме телеприсутствия интегрированы два блока питания, каждый из которых содержит четыре последовательно соединенных аккумулятора 18650. Первый источник энергии предназначен для питания четырех микроконтроллеров ESP32. Второй блок предоставляет энергию для работы десяти сервоприводов постоянного вращения модели Feetech FS5113R / 360°.

Характеристики используемых аккумуляторов:

- выходное напряжение: 3,6В (или 3,7В)
- емкость: 3400 мАч

В каждом из них имеются преобразователи напряжения со следующими характеристиками:

- входное напряжение: 10 - 30В
- выходное напряжение: 5В
- выходной ток: 10А
- максимальная выходная мощность: 50 Ватт

Напряжение с каждого блока равняется 14,4В (или 14,8 при напряжении аккумулятора 3,7В). Перед подачей питания на модули, его напряжение необходимо преобразовать/понизить используя преобразователь с вышеуказанными характеристиками.

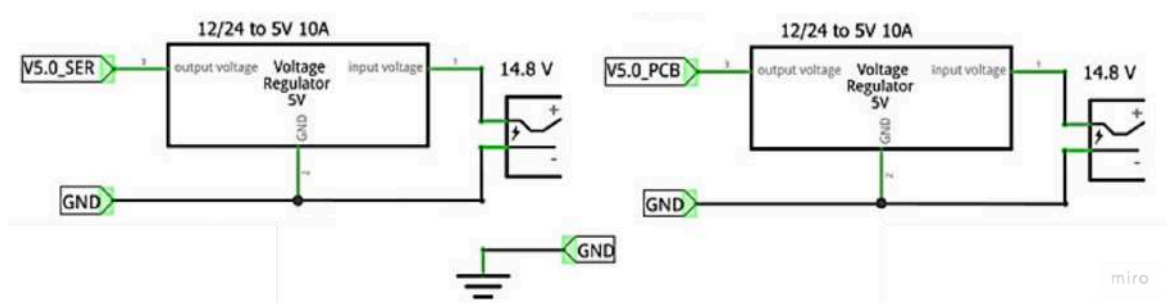


Рисунок 3.23 - Схема понижения (12/24 В - 5 В) напряжения с повышением силы тока (10 А)

Из документации модулей и проводя эксперименты получены следующие данные:

- Плата с ESP32 номинально потребляет 250мА.
- Минимальный ток сервопривода Feetech FS5113R составляет 130мА. А максимальный потребляемый ток (или ток блокировки рычага) составляет 2.2А.

Суммарный потребляемый ток (I_{pcb}) и суммарную потребляемую мощность (P_{pcb}) 4 плат с ESP32 и Raspberry Pi 4 Model B:

$$I_{pcb} = 4 * I_{esp} = 4 * 250mA = 1A$$

$$P_{pcb} = U_{pcb} * I_{pcb} = 5V * 1A = 5W$$

Суммарный потребляемый ток (I_{servos}) и суммарную потребляемую мощность (P_{servos}) используемых 10 сервоприводов:

$$I_{servos} = 10 * I_{ser} = 10 * 2,2A = 22A$$

$$P_{servos} = U_{servos} * I_{servos} = 5V * 22A = 110W$$

Находим максимальную мощность блока с четырьмя последовательно соединенными аккумуляторами:

$$E_{total} = (4 * U_{bat}) * Q_{batt} = 4 * 3,6 * 3,4 = 48,96Wh$$

Из этого находим потребление энергии в Ah при напряжении 5В:

$$Q_{total} = \frac{E_{total}}{U_{pcb}} = \frac{48,96}{5} [Ah] = 9,792[Ah]$$

Расчет времени автономной работы платы (T_{pcb}) и сервоприводов (T_{servos}) вычисляются по следующим формулам:

$$T_{pcb} = 10 * \frac{Q_{total}}{P_{pcb}} = 10 * \frac{9,792}{5} = 19,584[h]$$

$$T_{servos} = 10 * \frac{Q_{total}}{P_{servos}} = 10 * \frac{9,792}{110} = 0,890(18)[h]$$

3.7. Выводы по главе

В результате разработки и практической реализации костюма телеприсутствия (КТП) были получены следующие основные результаты:

- Разработана классификация и создан действующий прототип КТП. Предложена классификация костюмов телеприсутствия по функционалу и типу взаимодействия с аватаром, выделяющая этапы развития технологии от базового захвата движений до систем с использованием нейроинтерфейсов. На основе этого разработан, сконструирован и собран аппаратный прототип костюма и тактильной перчатки (KOLD), реализующий двунаправленный обмен данными между пользователем и виртуальной средой.
- Спроектирована архитектура и сетевая схема взаимодействия. Разработана многоуровневая сетевая схема, включающая модули захвата движения (гироскопы, акселерометры, датчики изгиба) и модули обратной связи (сервоприводы, светодиоды, в будущем — элементы Пельтье). Определены функциональные требования к системе, включая двусторонний и асинхронный обмен данными между микроконтроллерами (ESP32) и сервером через Wi-Fi и протокол MQTT, что обеспечивает интеграцию КТП с метавселенной.
- Разработаны и реализованы алгоритмы управления. Созданы и отлажены программные алгоритмы (прошивки) для микроконтроллеров, использующие двухъядерную архитектуру ESP32 для параллельной обработки: одно ядро обеспечивает высокую приоритетную связь с сетью (обмен данными по MQTT), а другое — обработку сигналов с сенсоров (потенциометров и датчиков изгиба) и управление актуаторами (сервоприводами и светодиодами). Реализованы схемы подключения и

обоснован выбор интерфейсов (АЦП, ЦАП, ШИМ, I2C) для мониторинга и управления всеми элементами системы.

- Обеспечена электромагнитная совместимость и оценена автономность. Проведен анализ электромагнитной совместимости (ЭМС) элементов КТП, в результате которого выявлены наиболее уязвимые для помех узлы (входы АЦП микроконтроллеров, подключенные к аналоговым датчикам). Разработаны практические рекомендации по снижению уровня внутренних помех (экранирование, скрутка проводов, уменьшение толщины жгутов). Выполнен расчет времени автономной работы, который показал, что ограничивающим фактором является энергопотребление сервоприводов (теоретическое время работы около 0,9 часа при максимальной нагрузке), что определяет необходимость оптимизации питания в будущих версиях.

ГЛАВА 4. РАЗРАБОТКА МЕТОДОВ СГЛАЖИВАНИЯ И АДАПТИВНОЙ КОРРЕКТИРОВКИ АНАЛОГОВЫХ СИГНАЛОВ

Целью разработки данных методов сглаживания является разработка высокопроизводительного и реактивного алгоритма, которую можно использовать в иммерсивных технологиях, где каждая микросекунда имеет значительное влияние на общее качество обслуживания пользователя.

4.1. Алгоритм обнаружения помех

При сглаживании данных с сенсоров, одним из важных процессов является определение помех. Его целью является определение действительных экстремумов входящих данных с сенсоров.

В нашем случае, используется простой алгоритм определения помех, которая собирает данные за более чем 2 секунд и находит их среднеарифметические минимумов и максимумов. Но в дальнейших улучшениях планируются ускорить процесс выполнения алгоритма (до 1й секунды) и улучшить точность определения границ.

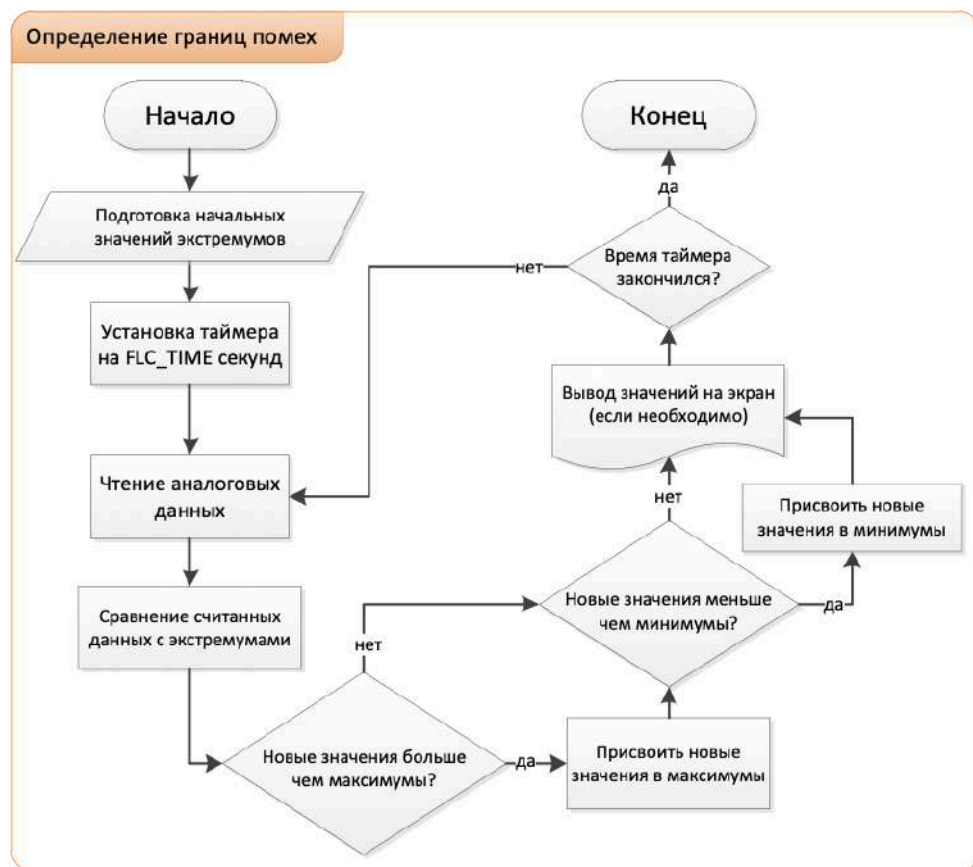


Рисунок 4.1 - Алгоритм определения границ помех

- Начальные значения – это минимально и максимально возможное значение, которая присваивается к переменным которая хранит максимумы и минимумы границ помехов соответственно;
- FLC_TIME – продолжительность калибровки (в миллисекундах);
- Экстремумы – минимумы и максимумы.

Ниже приведен процесс работы алгоритма:



Рисунок 4.2 - Процесс определения границ помех

После определения помех результаты будут уменьшены на 50%.

4.2. Алгоритм калибровки

Для более точного получения экстремумов каждого датчика изгиба или давления используется метод среднего скользящего и корректор помех в алгоритме

калибровки. В начале процесса программа оповещает пользователя о том что он должен сделать для правильной калибровки. Оповещение производится на мониторе сервера или порта.

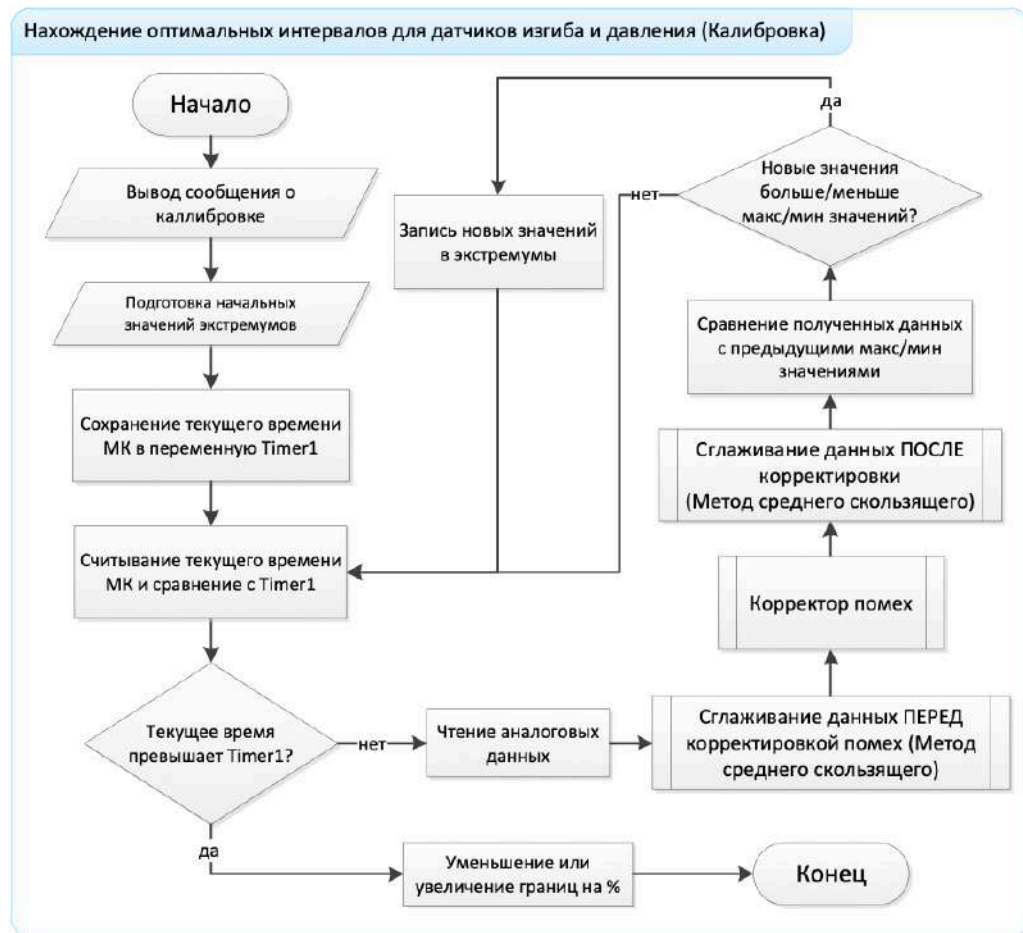


Рисунок 4.3 - Алгоритм калибровки датчика изгиба и давления

4.3. Разработанная версия алгоритма скользящего среднего

Этот алгоритм использует итерацию значений чтобы обработать данные сенсоров. В нем имеются трехмерные переменные которые действуют как буферы, которые хранят значения перед и после фильтрации каждого сенсора в двумерных массивах.

В каждой итерации двумерных массивов вычисляется среднее значение действующей/выбранной строки и вставляется в первую ячейку следующей строки после перемещения ее значений в правую сторону.

$$C_{[n+1; j+1]} = C_{[n+1; j]}$$

$$C_{[n+1; 0]} = \sum_{i=0}^k C_{[n; i]},$$

где:

- C – значение ячейки;
- n и j – индексы ячеек;
- k – количество строк двумерного массива;
- Результирующее значение алгоритма - это среднее значение последней строки.



Рисунок 4.4 - Алгоритм разработанной версии среднего скользящего

4.4. Алгоритм корректировки помех.

Главной целью разработки данного алгоритма является создание фильтра аналоговых сигналов, способной вычислить минимальные смещения, пропуская ложные скачки значений сенсора.

Алгоритм использует значения экстремумов помех, чтобы вычислить исходное положение входного аналогового сигнала.

Результирующее значение равняется середине между нижней и верхней границ разрешенной области. Эти границы и есть экстремумы помех входящего сигнала. Границы смещаются исходя от количества точек (входящих данных, N)

которые вышли из границ за К-входящих общих точек или за ΔT -прошедшего времени.

Как описано выше, алгоритм включает в себя два коэффициента. Первое – это количество ожидаемых входящих точек (N) для вычисления среднего. А второе – это предел ожидания входящих точек. Она может быть определена количеством входящих данных или временем прошедшей после определения точки, которая вышла за пределы границ. Исходя из задержки цикла системы, можно выбрать коэффициент К или ΔT .

В данном проекте, в качестве второго коэффициента, был выбран К (количественное ожидание) равный 4. А коэффициент N равен 3.

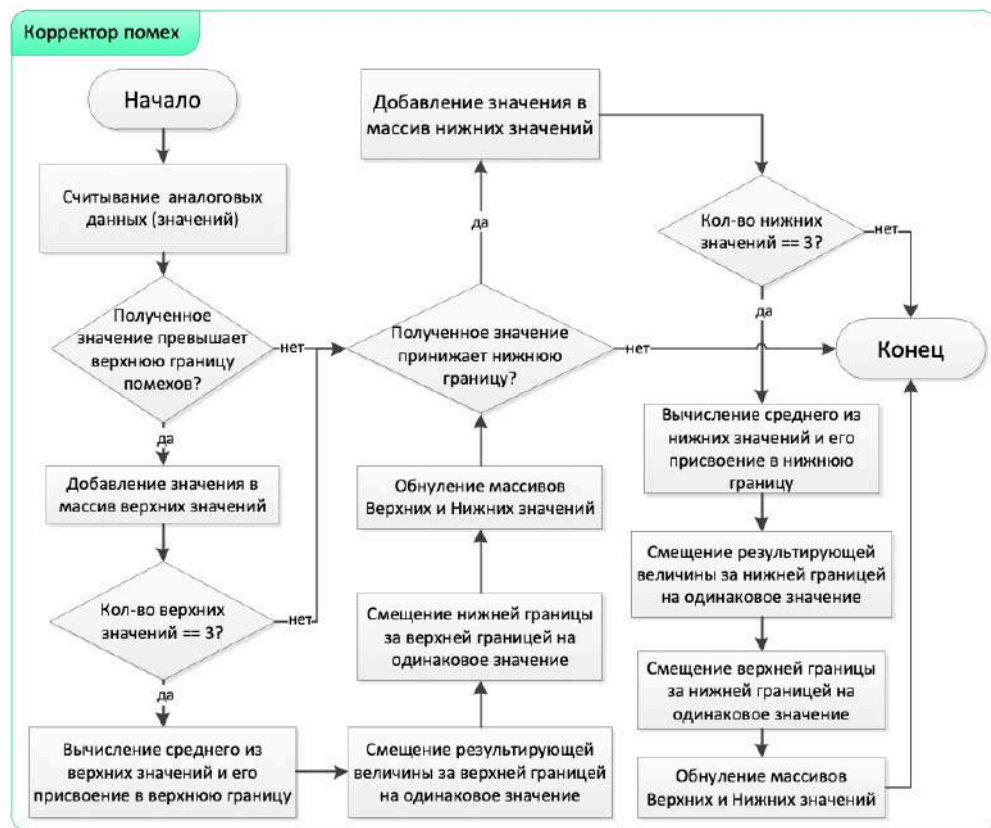


Рисунок 4.5 - Принцип работы корректора помех

4.5. Результаты внедрения разработанных фильтров

На следующих графиках можно увидеть поведение результирующего сигнала при использовании различных комбинаций фильтров.

При считывании данных, специально были добавлены задержки в 10 миллисекунд для проверки своевременной реакции фильтра, а также, чтобы наглядно зафиксировать результаты.

Для начала рассмотрим чистое применение фильтра корректора помех с коэффициентом 1.1:



Рисунок 4.6 - Применение фильтра корректировки помех с коэффициентом 1.1

Далее, можно увидеть более спокойное поведение результирующего сигнала при добавлении скользящего среднего (с коэф. 3x3) перед фильтром коррективки помех:

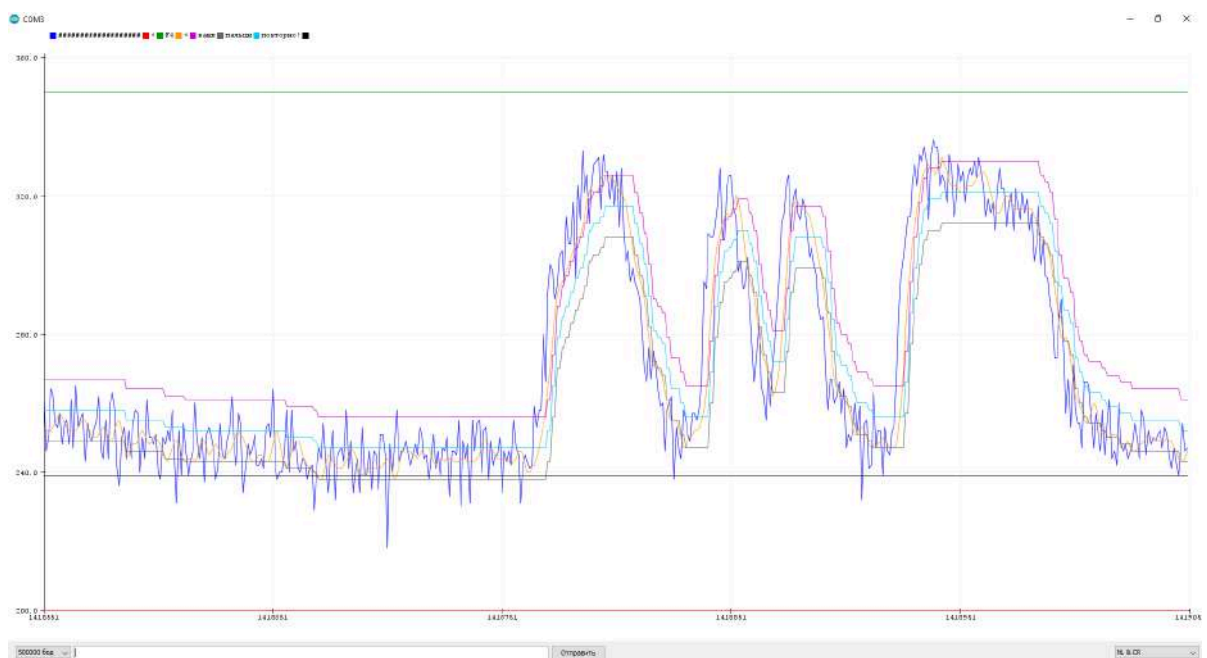


Рисунок 4.7 - Результат незначительного сглаживания входящих сигналов перед их корректировкой

Далее приведены результаты используемых фильтров при указанных коэффициентах

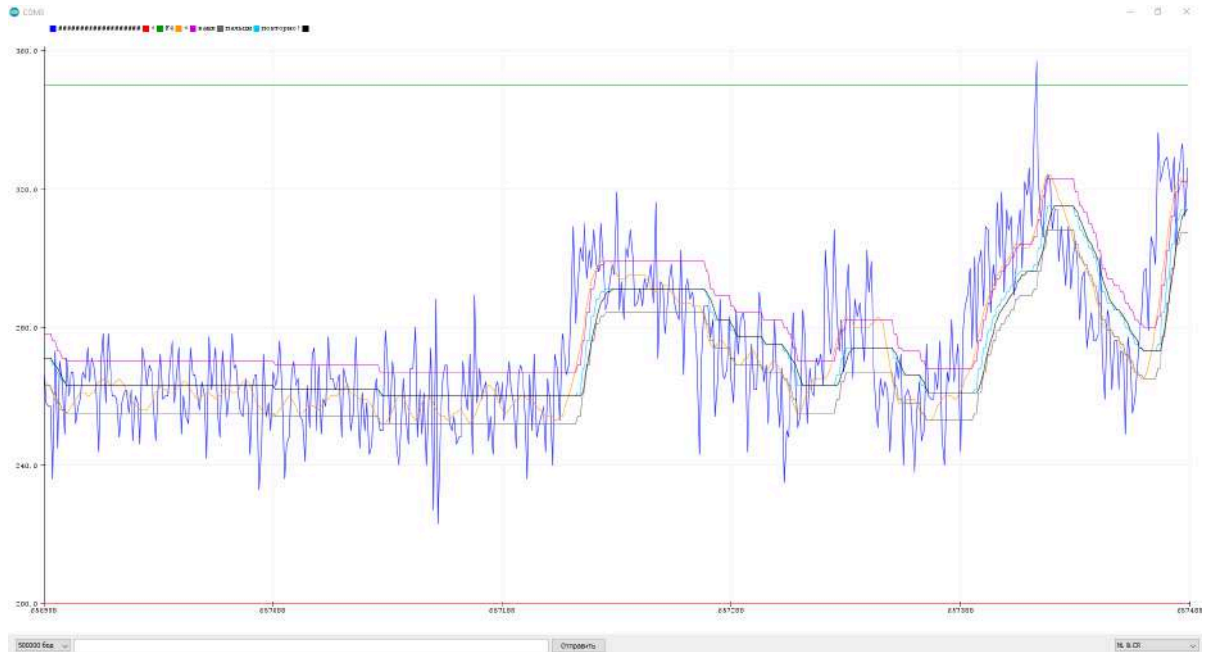


Рисунок 4.8 - Результаты фильтров при использовании вышеприведенных коэффициентов

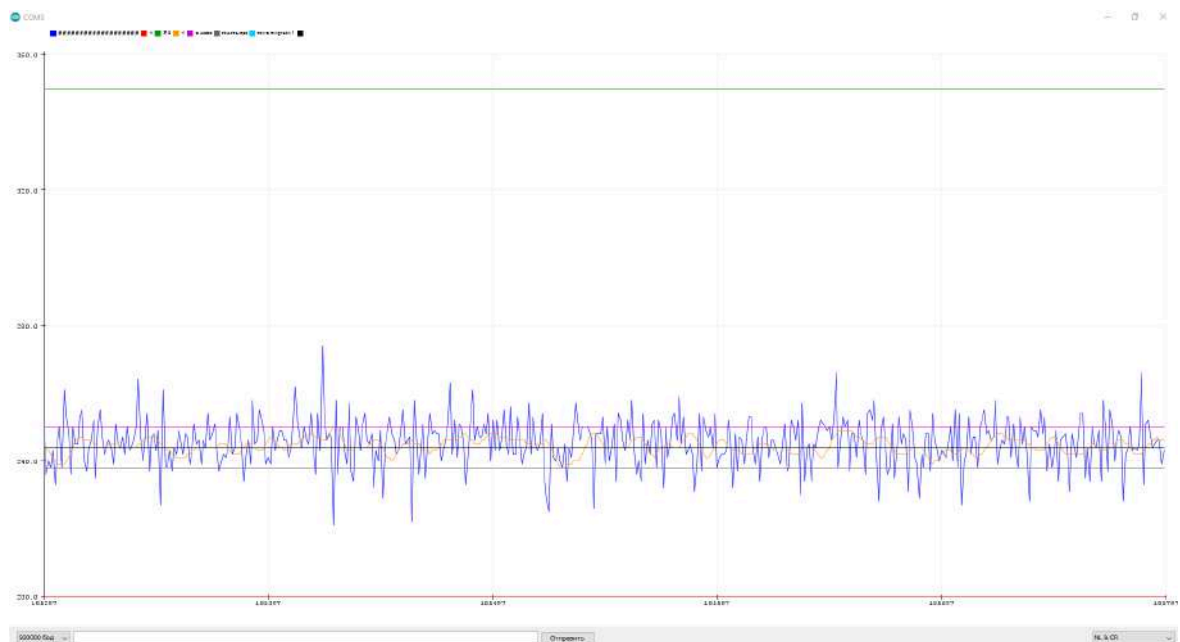


Рисунок 4.9 - Удержание датчика изгиба на одном положении чтобы для проверки постоянства результата в состоянии неподвижности руки при тех же коэффициентах

Для ясности, можно оставить только входные и результирующие данные на графике:

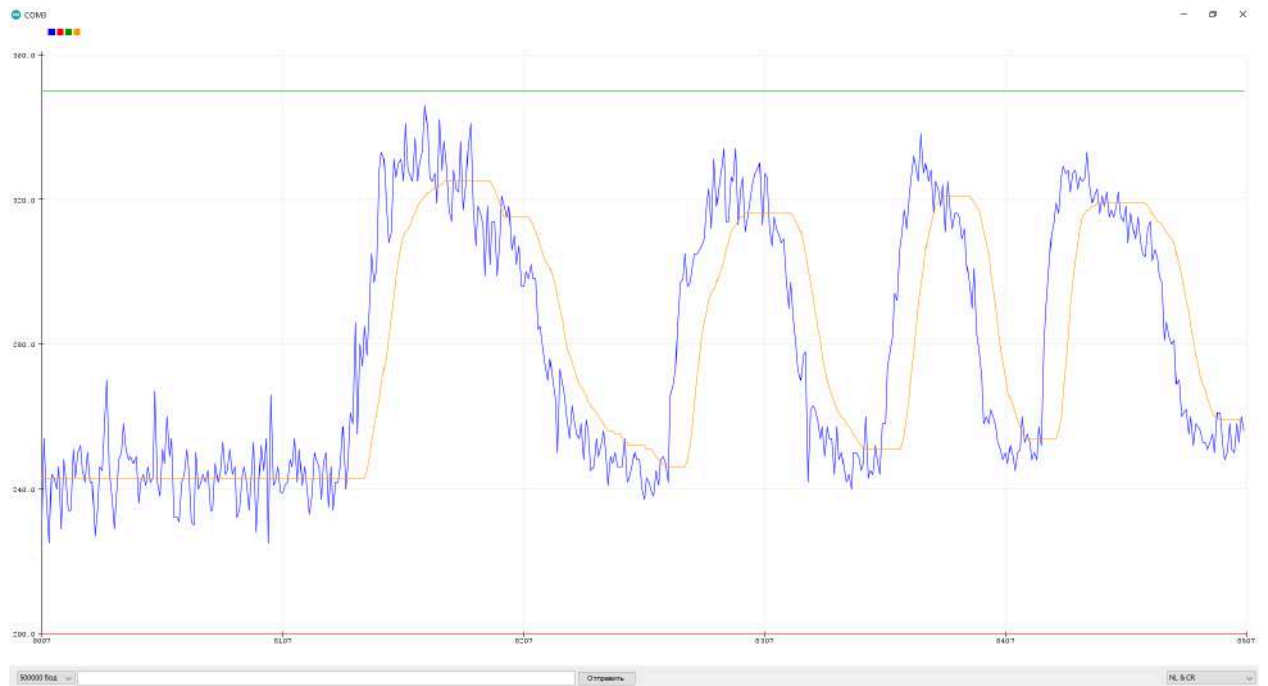


Рисунок 4.10 - Обычное движение датчика изгиба для проверки реакции фильтра

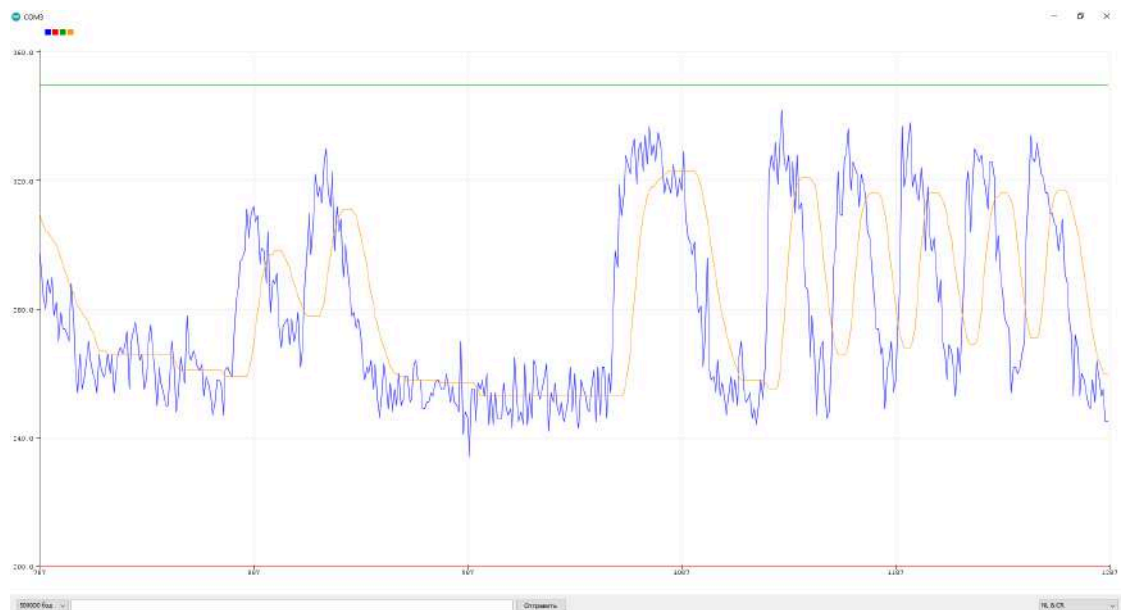


Рисунок 4.11 - Быстрое движение датчика изгиба для проверки реакции фильтра



Рисунок 4.12 - Удержание датчика изгиба на одном положении для проверки постоянства результата.

4.6. Примеры применения

Данный алгоритм можно использовать во всех сферах, где необходимо сглаживать помехи разного характера в аналоговых сигналах.

К примеру, в сфере медицины, в направлении телемедицины, где хирург дистанционно проводит операцию с применением тактильной перчатки, в которой используются датчики изгиба.

Существующие на сегодняшний день датчики, имеют ощутимые помехи, которые могут заставить дрожать роботизированные руки на исполняющей стороне системы при проведении дистанционной операции.

На практике, при успешной калибровке данного фильтра, можно использовать его в типичных случаях.

4.7. Выводы по главе

В данной главе рассмотрен авторский метод сглаживания сигналов с аналоговых сенсоров, и обоснована критическая необходимость фильтрации аналоговых сигналов датчиков изгиба и потенциометров КТП в связи с высокими помехами в интерфейсе АЦП управляющего устройства. В том числе:

- Разработан высокопроизводительный алгоритм калибровки и автоматического определения границ помех, вычисляющий среднеарифметические экстремумы шума в статическом состоянии КТП.

- Создана модифицированная версия фильтра скользящего среднего на основе двумерных массивов, работающая по принципу FILO (First In, Last Out), обеспечивающая достаточное первичное сглаживание аналоговых сигналов.
- Разработан и внедрен оригинальный алгоритм адаптивного корректора помех, который вычисляет положение сигнала, ближе к истинной на основе динамического смещения разрешенной области пропускания. Метод успешно устраняет ложные скачки и дрожание исполнительных механизмов (роботизированных рук), что является очень важным для аналоговых сенсоров систем телемедицины.

ГЛАВА 5. МОДЕЛЬНАЯ СЕТЬ ДЛЯ ПЕРЕДАЧИ ИММЕРСИВНЫХ ОЩУЩЕНИЙ В КОСТЮМЕ ТЕЛЕПРИСУТСТВИЯ С ОБРАТНОЙ СВЯЗЬЮ

5.1. Подход к разработке улучшенной модели передачи иммерсивных данных перчатки КТП

Разработанное устройство состоит из сенсоров и актуаторов (далее - модули) и нескольких управляющих микроконтроллеров (рис.), соединенных используя разные интерфейсы. В составе каждого актуатора имеется специальный чип-драйвер, которая позволяет управлять этим актуатором по полному, и у которого есть свой интерфейс для подключения и управления актуатором.

Используемые драйверы поддерживают такие интерфейсы, как SPI, I2C и PWM.

Для подключения аналоговых сенсоров, используется АЦП интерфейс.

А для проверки работоспособности микроконтроллера, можно подключиться к нему через интерфейс UART.

Для этих целей были выбраны платы с микроконтроллером ESP32, которая поддерживает вышеприведенные интерфейсы, а также другие интерфейсы, которые могут понадобиться для расширения возможностей разработанной системы.

Абстрактную схему подключения используемых элементов можно увидеть на следующем рисунке:

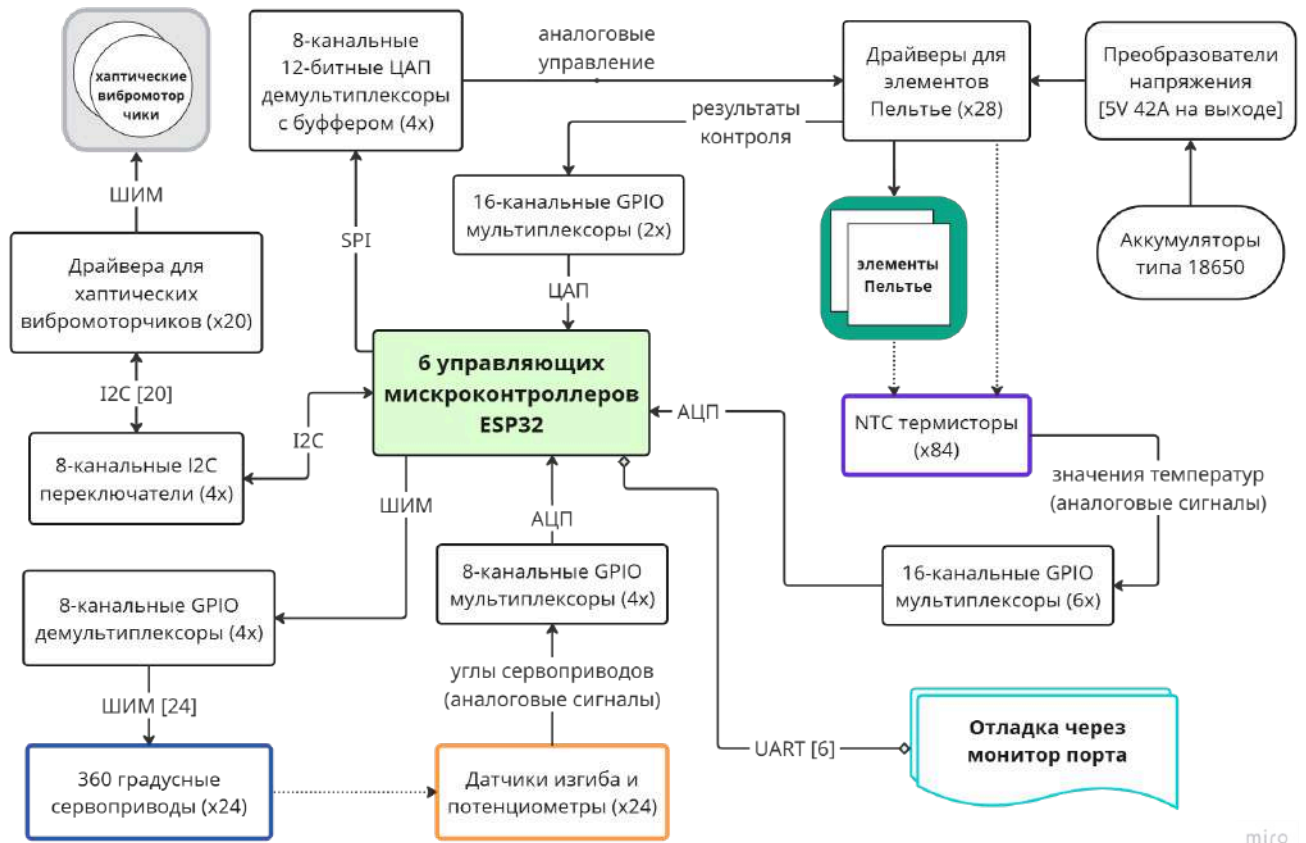


Рисунок 5.1 - Обобщённая схема соединения модулей устройства к управляющим микроконтроллерам.

Некоторые из вышеприведенных модулей, будут расположены на перчатке следующим образом:

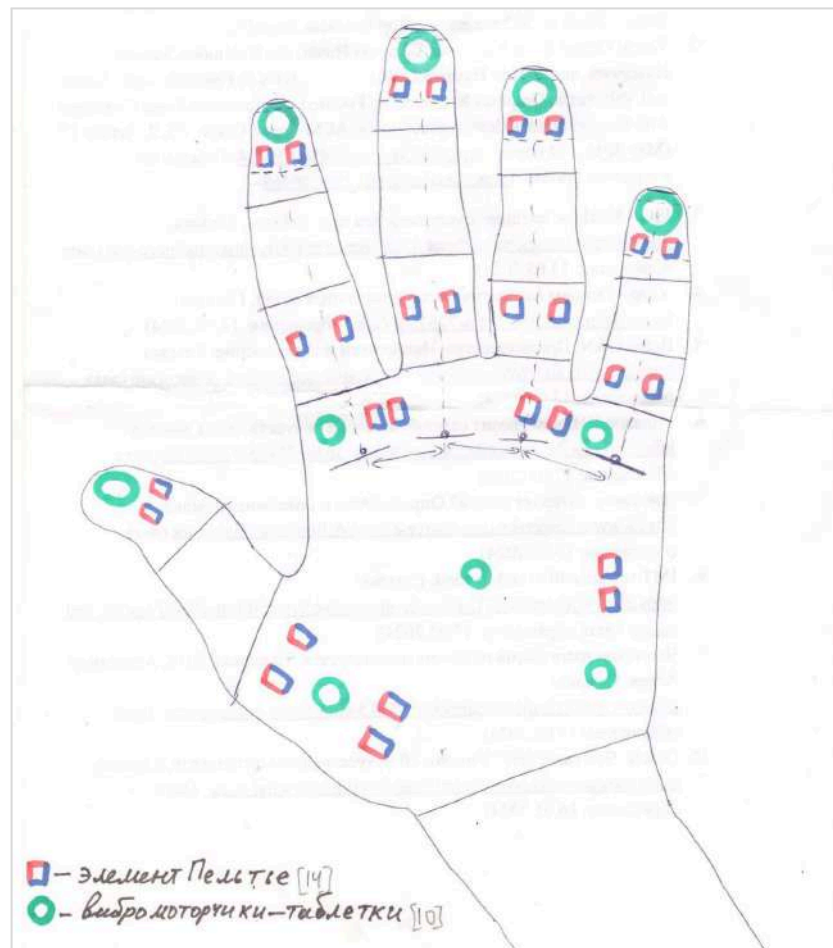


Рисунок 5.2 - Концепция иммерсивной перчатки

Также, схема связи составляющих КТП и перчатки, и с составляющими метавселенной будут выглядеть следующим образом:

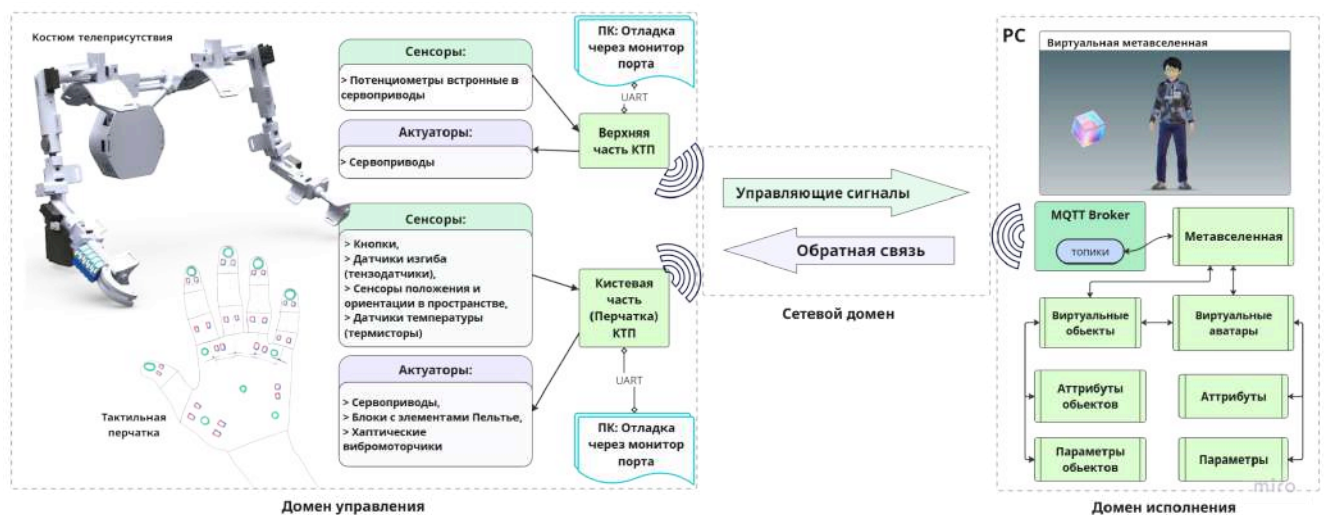


Рисунок 5.3 - Новая схема подключения сенсорной сети для КТП

5.2. Модифицированные алгоритмы работы загруженных программ

В используемом микроконтроллере (на управляющей плате) имеются два ядра, одна из которых используется для обмена данными между управляющим микроконтроллером и сервером, а другая - для управления подключенными модулями (рис.).

Алгоритм работы общей системы описано на следующей схеме:

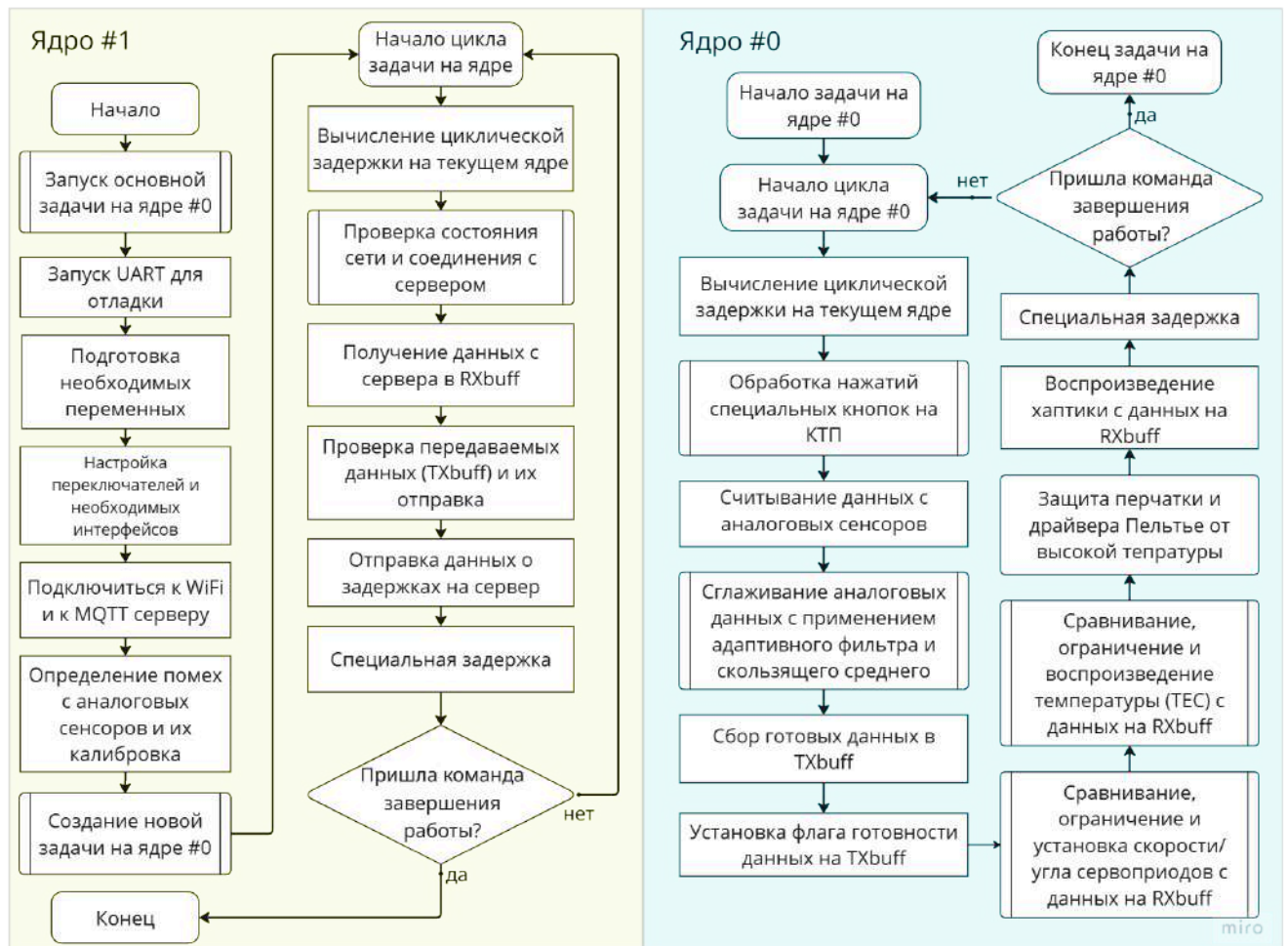


Рисунок 5.4 - Общий алгоритм работы клиентского устройства.

Общий алгоритм можно подразделить на три части: Начальная ф-я для настройки и подготовки платы при запуске, и две циклических функций - для связи и для управления подключенными и настроенными модулями.

В нашем скетче функция `setup()` срабатывает один раз в начале запуска устройства, а функции `loop()` и `loop2()` - работают циклично.

При запуске устройства выполняется ряд подготовительных процессов (функция `Setup`):

- Настройка UART интерфейса - на этапе разработки устройства используется для отладки и взаимодействия платы с разработчиком.
- Подготовка необходимых констант и переменных для улучшения быстродействия в работе устройства
- Подключение к WiFi и к MQTT брокеру. А также подписка на необходимые топики для получения данных с них.
- Определение границ помех сенсоров, подключенных к АЦП для дальнейшего сглаживания
- Калибровка сенсоров, подключенных к АЦП
- Создание дополнительной задачи (асинхронного цикла) с низким приоритетом на другом ядре микроконтроллера

После создания новой задачи у нас имеются два асинхронного цикла с разными приоритетами выполнения.

Так как необходимо обеспечить бесперебойную связь, задаче для взаимодействия с сетью ставим высокий приоритет. Цикл работы данной задачи состоит из следующих подзадач:

- Начало цикла (HighPriorityLoop Core#1)
- Вычисление задержки выполнения цикла данной задачи (для отладки; можно использовать для создания специальной задержки на данную задачу)
- Проверка соединения к беспроводной сети (WiFi) и к серверу (MQTT брокеру). А также выполнение некоторых задач синхронизации
- Получение принятых на указанный топик данных и их присвоение на переменную RXbuff
- Проверка готовности данных на TXbuff для отправки на сервер и отправка на указанный топик сервера
- Отправка вычисленной задержки цикла данной задачи
- (опционально) Удержание (задержка) данного цикла для предотвращения перегрузки сервера и для экономии энергии микроконтроллера
- Конец цикла

А цикл работы для взаимодействия с модулями, подключенных на микроконтроллер состоит из следующих подзадач:

- Начало цикла (LowPriorityLoop Core#0)
- Вычисление задержки выполнения цикла данной задачи (для отладки; можно использовать для создания специальной задержки на данную задачу)
- (опционально) Можно выполнить специальные команды клиентского или сетевого уровня при подключении кнопки (кнопок) на микроконтроллер. Нажатия можно персонализировать с применением специальной библиотеки.

Сервисные кнопки на перчатке пользователя

- Подцикл для работы с модулями используя мультиплексоры, буферные демультимплексоры и I2C-переключатели:
 - Чтение данных с термисторов и датчиков изгиба (АЦП сенсоров)
 - Трехэтапная обработка данных с АЦП сенсоров:
 - Авторский алгоритм скользящего среднего с низкими коэффициентами
 - Адаптивный фильтр с низкими коэффициентами
 - (при необходимости) Авторский алгоритм скользящего среднего с низкими коэффициентами (для сглаживания ступенчатости Адаптивного фильтра)
 - Сбор данных АЦП сенсоров в переменную TXbuff для отправки на указанный топик сервера. Собранные данные также используются для работы с актуаторами клиентского устройства.
 - Применение вычисленных с датчика изгиба данных и данных, полученных с сервера (RXbuff) для установки новой скорости (угла) сервоприводов, подключенных к пальцам пользователя.
- **Кинестетическая обратная связь**
 - Применение вычисленных с термистора данных и данных, полученных с сервера (RXbuff) для установки нового значения температуры, подключенных к пальцам пользователя. **Температурная обратная связь**
 - Защита руки пользователя и драйвера элемента Пельтье от повышенной температуры используя данные с термисторов на перчатке и на поверхности драйвера
 - Воспроизведение хаптики с применением вибромоторчиков.
- **Хаптическая обратная связь**
 - Установка флага готовности отправки данных с сенсоров на указанный топик сервера
 - (опционально) Удержание (задержка) данного цикла для предотвращения перегрузки сервера и для экономии энергии микроконтроллера

5.3. Описание модулей используемых для воспроизведения иммерсивных данных

Для разработки новых методов передачи иммерсивных ощущений были использованы нижеперечисленные модули.

5.3.1. Температурная обратная связь с применением элемента Пельтье, специального драйвера

Целью выбора элемента Пельтье для воспроизведения температуры виртуального или физического объекта (для ТОС) являются следующие:

- Компактность и размеры
- Не требует дополнительных расходных материалов при использовании. Работает за счет тока.
- Отсутствуют механически движущиеся части актуатора
- Бесшумная работа

Несмотря на свои возможности, имеются и недостатки применения данного модуля:

- Потребляет много тока
- Теплая сторона выделяет относительно больше тепла чем выделение холода холодной стороны
- Требуется стабильного напряжения\тока, чтобы ресурсы компонентов элемента не деградировали быстро
- Низкий КПД
- Высокая стоимость как актуатора так и используемого для него драйвера-чипа
- Требуется хорошего теплоотвода на теплой стороне актуатора и на управляющем драйвере

В данном проекте, был выбран микромодуль с обозначением ТВ-7-0.6-0.8, размеры которой позволяют с легкостью расположить на поверхности пальцев разрабатываемой перчатки для тактильного. В документации данного элемента можно увидеть следующие нужные нам для экспериментов характеристики:

Размеры: 4.3 x 4.3 x 2.55 мм

Питание: $U_{max} = 0.9V$; $I_{max} = 2.1A$

Сам микромодуль имеет следующий вид:

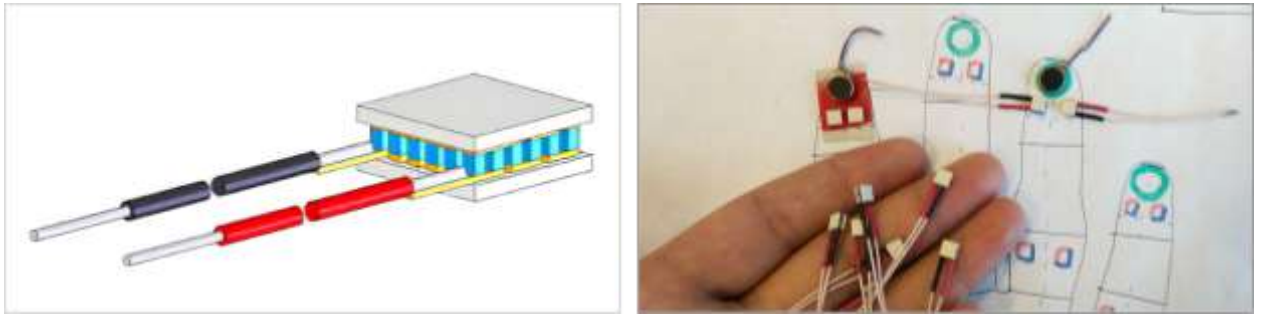


Рисунок 5.5 - Внешний вид микро-элемента Пельтье и его вид с пальцами (для понимания его размеров)

При использовании данного модуля необходимо расширить область обратной (температурной) связи используя теплопроводящий материал.

Данный либо, другой с более высокой теплопроводностью, материал также должен отводить тепло с внешней стороны элемента Пельтье.

Также, при необходимости нужно использовать материал с низкой теплопроводностью для ограничения влияния горячей стороны на холодную сторону элемента.

Примерная конструкция модульного блока изображен на следующем рисунке:

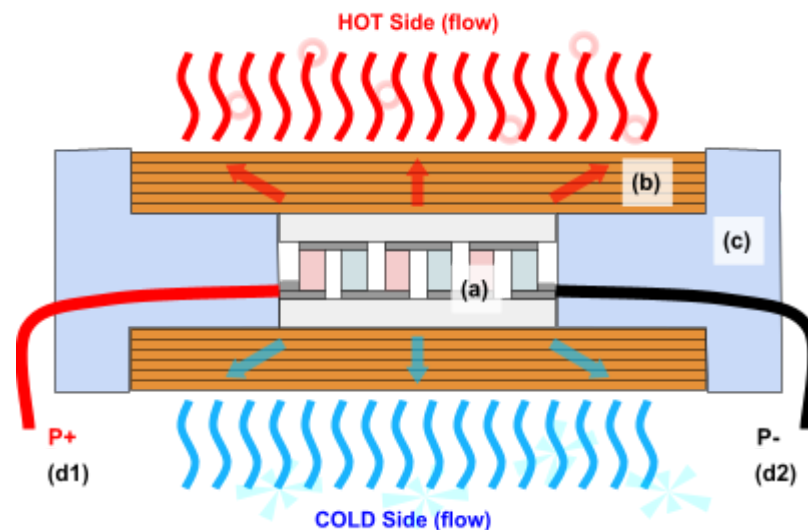


Рисунок 5.6 - Расположение элемента Пельтье в модульной конструкции.

Где:

(a) - элемент Пельтье

(b) - слой, которая максимально отводит температуру (тепло/холод). В нашем случае, это - медной брусок или слои медной фольги (скотча).

(c) - слой, которая максимально отводит температуру

(d) - провода элемента Пельтье.

Причиной выбора меди является высокий коэффициент теплопроводности, которая составляет 394 Вт/(м*К) при температуре 20–100 °С.

Но, также можно применить другие материалы-теплоотводы, которые легче прикрепить или нанести на модульную конструкцию.

Для управления данным микромодулем был выбран специальный чип-драйвер **MAX1968** и разработана специальная тестовая плата на его основе (рис.) которая позволяет подать необходимые токи для управления и мониторинга температуры стороны микроэлемента. прилагаемых к пальцу/руке перчатки.

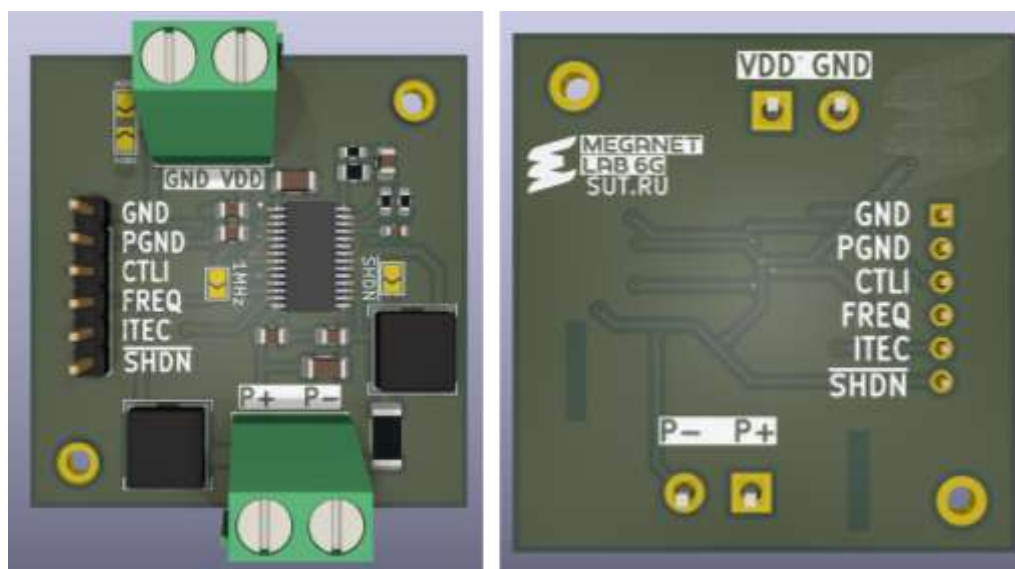


Рисунок 5.7 - Внешний вид разработанной платы с драйвером для управления элементом Пельтье

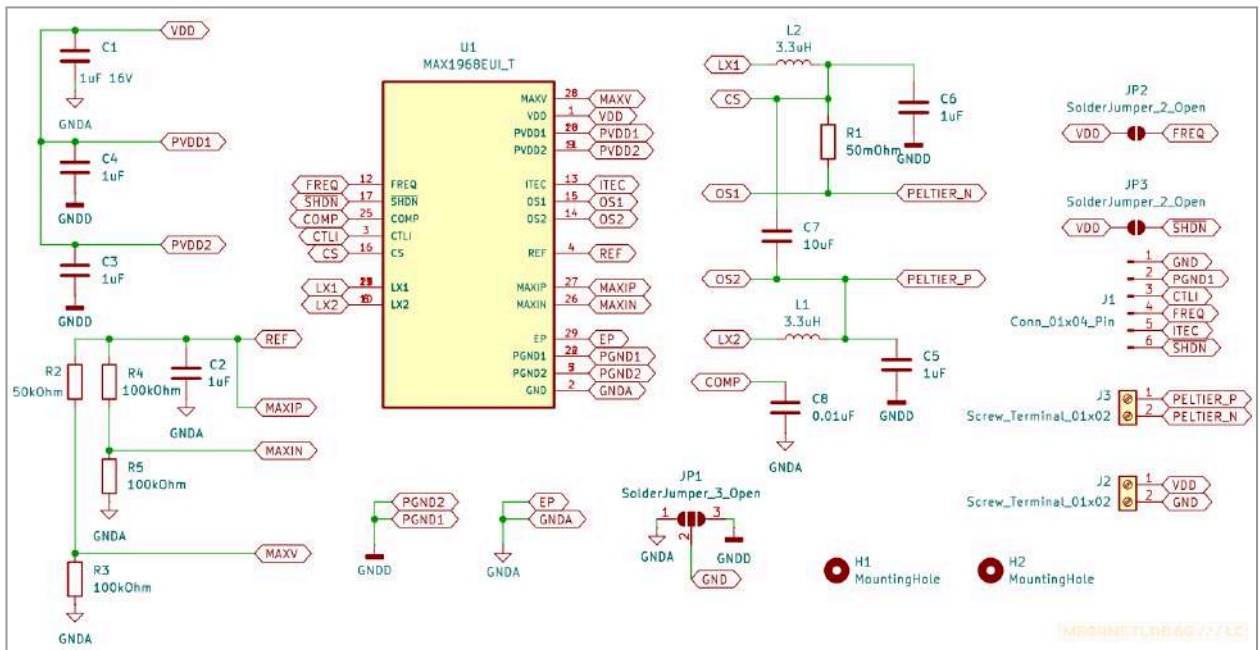


Рисунок 5.8 - Состав (схема) разработанной платы с драйвером для управления элементом Пельтье

Также и для него смотрим необходимые для нас характеристики и возможности из его документации:

Питание драйвера: $U = 3.0 \dots 5.5 \text{ V}$; $I_{\max} = 4.5 \text{ A}$

Питание на выходе до: $U = \pm 4.3 \text{ V}$; $I_{\max} = \pm 3 \text{ A}$

Интерфейс управления: ЦАП

Назначение основных выводов платы:

- CTLI - вывод управления выходным током на выводах [P+; P-]. Диапазон значений напряжения на выходах зависит от питания драйвера, а диапазон тока: от -3 до +3 Ампер. Входное значение напряжения: от 0 до 3 Вольт. При входном напряжении 1,5 Вольт - ток на выходе [P+; P-] будет равняться нулю.
- ITEC - мониторинг подачи тока на выводах [P+; P-]. Выходное значение напряжения: от 0,3 до 2,7 Вольт.
- FREQ - установка частоты работы драйвера (HIGH - 1МГц; LOW - 500кГц). Вывод заранее замкнут к положительному питанию.
- SHDN - вывод для включения / отключения драйвера. Вывод заранее замкнут к положительному питанию.

Подробные сведения можно посмотреть на официальной документации.

Настройка выходного напряжения и тока:

Данный модуль был собран по шаблону MAX1968 evaluation kit (EV kit) и по рекомендациям в официальной документации, и выдаст напряжение на выходе 4В (при подаче 5В на входе) и нуждается доработке для нашего проекта.

Для этого необходимо поменять значения входных напряжений на выводах MAXIP (), MAXIN и MAXV драйвера используя следующие уравнения и делители напряжений опираясь на напряжение на выводе REF:

- максимальный положительный ток через элемент Пельтье:

$$I_{TECP(MAX)} = + \frac{V_{MAXIP}}{10 \cdot R_{SENSE}}$$

$$V_{MAXIP}(V) = REF \cdot \frac{R_7}{R_6 + R_7}$$

- максимальный негативный ток через элемент Пельтье:

$$I_{TECN(MAX)} = - \frac{V_{MAXIN}}{10 \cdot R_{SENSE}}$$

$$V_{MAXIN}(V) = REF \cdot \frac{R_5}{R_4 + R_5}$$

- максимальное напряжение через элемент Пельтье:

$$V_{TEC(MAX)} = 4 \cdot V_{MAXV}$$

$$V_{MAXV} = REF \cdot \frac{R_3}{R_2 + R_3}$$

где:

- REF - опорное напряжение, значение которого равняется 1,5В

$$V_{MAXIN}(V) = REF \cdot \frac{R_{GND}}{R_{REF} + R_{GND}}$$

$$\frac{R_5}{R_4 + R_5} = \frac{V_{MAXI[P,N]}(V)}{REF} = \frac{10 \cdot R_{SENSE} \cdot I_{TEC[P,N](MAX)}}{REF}$$

$$R_2 = 51 \text{ кОм}$$

$$R_3 = 20 \text{ кОм}$$

Для мониторинга подачи тока вычислим формулу для получения тока используя другие формулы в документации:

$$I_{ITEC} = \frac{V_{OS1} - V_{CS}}{R_{sense}} = \frac{V_{CTLI} - 1.5}{10 \cdot R_{sense}}$$

$$V_{ITEC} = 1.5 + 8 \cdot (V_{OS1} - V_{CS})$$

Отсюда находим:

$$\frac{V_{ITEC} - 1.5}{8} = V_{OS1} - V_{CS}$$

Правую часть формулы ставим в вышеописанную формулу (I_{ITEC}) и находим:

$$I_{TEC} = \frac{V_{ITEC} - 1.5}{8 \cdot R_{sense}}$$

, где $R_{sense} = 50 \cdot 10^{-3} = 0.05 \Omega$.

Данную формулу можно использовать в прошивке для мониторинга течения тока через элементы Пельтье.

Также, из формул выше можно вычислить планируемый ток исходя из напряжения управления током V_{CTLI} :

$$I_{TEC} = \frac{V_{CTLI} - 1.5}{10 \cdot R_{sense}}$$

Формула напряжения управления и напряжения тока для сравнения:

$$V_{CTLI} = 10 \cdot R_{sense} \cdot I_{TEC} + 1.5$$

$$V_{ITEC} = 8 \cdot R_{sense} \cdot I_{TEC} + 1.5$$

Данный драйвер может управлять только одним микромодулем (или сборкой микромодулей), и для управления несколькими микромодулями по разному - необходимы отдельные драйвера каждому микромодулю (или их сборке).

5.3.2. ЦАП-модуль с буфером для управления несколькими драйверами элемента Пельтье

Для стабильного управления и мониторинга каждым драйвером - выводы на микроконтроллере могут не хватить и можно использовать отдельный модуль-расширитель.

Для этого был выбран 8-ми канальный ЦАП-модуль с буфером **AD5328BRUZ**.

По некоторым причинам, было принято решение собрать свою тестовую плату для экспериментов (Рис. 5.9)



Рисунок 5.9 - Внешний вид тестовой платы с AD5328BRUZ (8-ми канальный ЦАП-модуль с буфером)

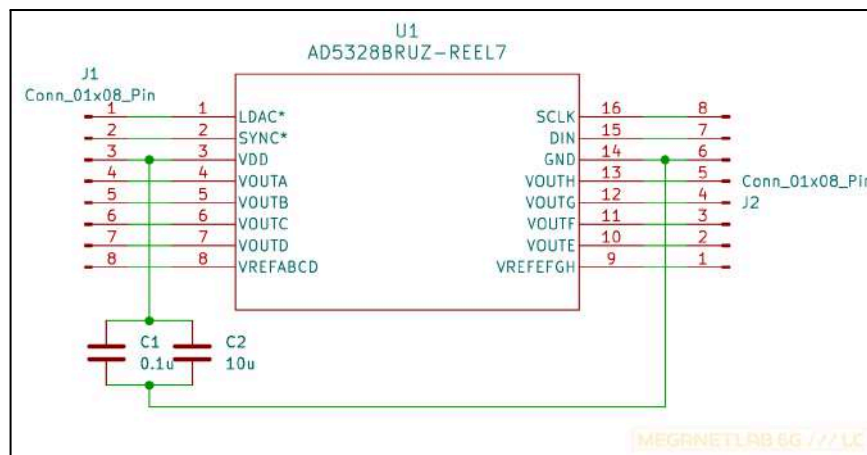


Рисунок 5.10 - Состав (схема) тестовой платы с AD5328BRUZ (8-ми канальный ЦАП-модуль с буфером)

Питание ЦАП-модуля: $U = 3.0 \dots 5.5 \text{ V}$; $I_{\max} = 4.5 \text{ A}$

Интерфейс управления: SPI (Серийный периферийный интерфейс)

Назначение выводов платы:

- VDD - питание (+). Входное напряжение от до Вольт.
- GND - земля (-).
- LDAC - вывод разрешения изменения значений ЦАП выхода платы.
- SYNC - вывод **состояния** SPI интерфейса. Подключается к **SS** выводу SPI интерфейса управляющего микроконтроллера.
- SCLK - вывод **синхронизации** SPI интерфейса. Подключается к **SCK** выводу SPI интерфейса управляющего микроконтроллера.
- DIN - вывод **данных** SPI интерфейса. Подключается к **MOSI** выводу SPI интерфейса управляющего микроконтроллера.
- Vref_[ABCD и EFGH] - опорное напряжение для выводов Vo_[A...H].
- Vo_[A...H] - выходной ЦАП сигнал от 0 до Vref Вольт.

Данный модуль подключается следующим образом используя интерфейс SPI. Стоит отметить, что подключение стоит делать без подачи питания на управляющую плату во избежание короткого замыкания:

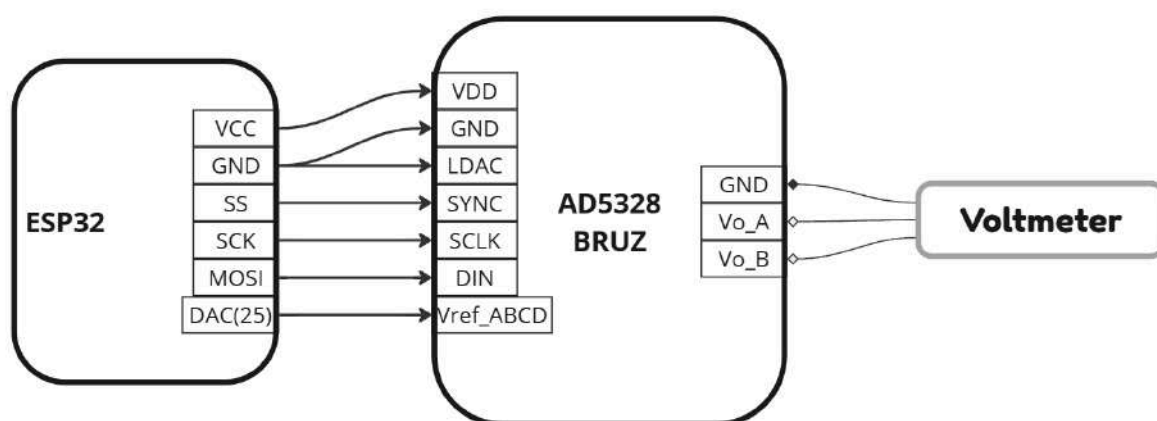


Рисунок 5.11 - Схема подключения AD5328BRUZ к ESP32.

В документации данного модуля можно увидеть структуру двухбайтного пакета передаваемых данных на ЦАП-модуль по проводному интерфейсу SPI (Рис. 5.12) и адресацию выходного сигнала (Рис. 5.13).

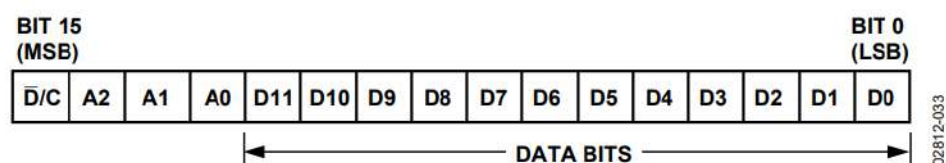


Рисунок 5.12 - Структура двухбайтного пакета передаваемых данных на ЦАП-модуль

A2 (Bit 14)	A1 (Bit 13)	A0 (Bit 12)	DAC Addressed
0	0	0	DAC A
0	0	1	DAC B
0	1	0	DAC C
0	1	1	DAC D
1	0	0	DAC E
1	0	1	DAC F
1	1	0	DAC G
1	1	1	DAC H

Рисунок 5.13 - Адресация выходных ЦАП-сигналов на модуле AD5328BRUZ.

После подключения, используя схему подключения и вышеописанную таблицу значений полей SPI-пакетов, можно управлять данным модулем по SPI, а также измерить задержку управления на стороне управляющего микроконтроллера. Алгоритм работы программы показан на следующей схеме (рис.5.14):

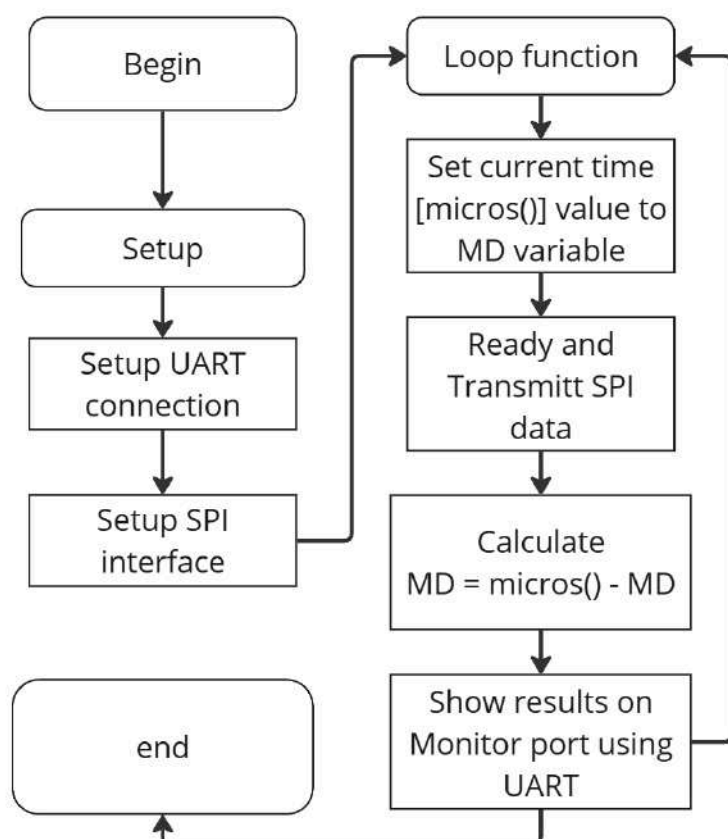
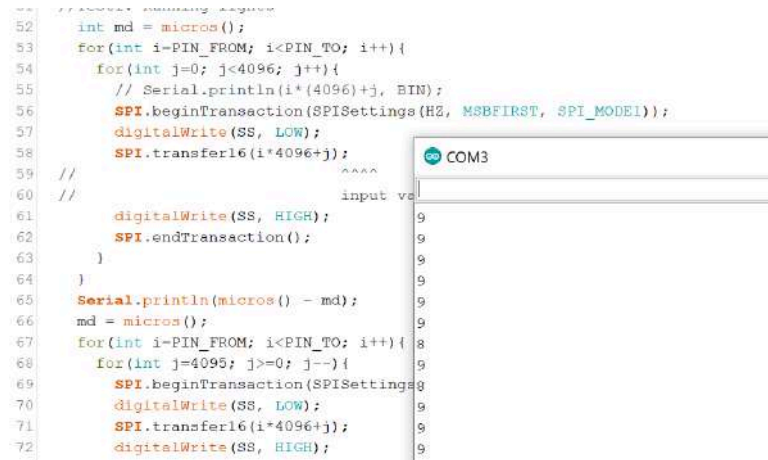


Рисунок 5.14 - Рис.

Результаты проверки показывают, что задержка отправки двухбайтного пакета SPI составляет примерно **9 микросекунд** (рис.5.15). При отправке 8 двухбайтных пакетов по очереди, задержка составляет примерно **65 мкс**.



```

52 // Serial.println(micros() - md);
53 for(int i=PIN_FROM; i<PIN_TO; i++){
54   for(int j=0; j<4096; j++){
55     // Serial.println(i*(4096)+j, BIN);
56     SPI.beginTransaction(SPI_Settings(HZ, MSBFIRST, SPI_MODE1));
57     digitalWrite(SS, LOW);
58     SPI.transfer16(i*4096+j);
59     //      ^^^^^ input value
60     //
61     digitalWrite(SS, HIGH);
62     SPI.endTransaction();
63   }
64 }
65 Serial.println(micros() - md);
66 md = micros();
67 for(int i=PIN_FROM; i<PIN_TO; i++){
68   for(int j=4095; j>=0; j--){
69     SPI.beginTransaction(SPI_Settings(HZ, MSBFIRST, SPI_MODE1));
70     digitalWrite(SS, LOW);
71     SPI.transfer16(i*4096+j);
72     digitalWrite(SS, HIGH);
  
```

Serial Monitor (COM3) output:

```

9
9
9
9
9
9
9
9
9
9
  
```

Рисунок 5.15 - Результат измерения задержки отправки двухбайтного SPI-пакета управления ЦАП модуля

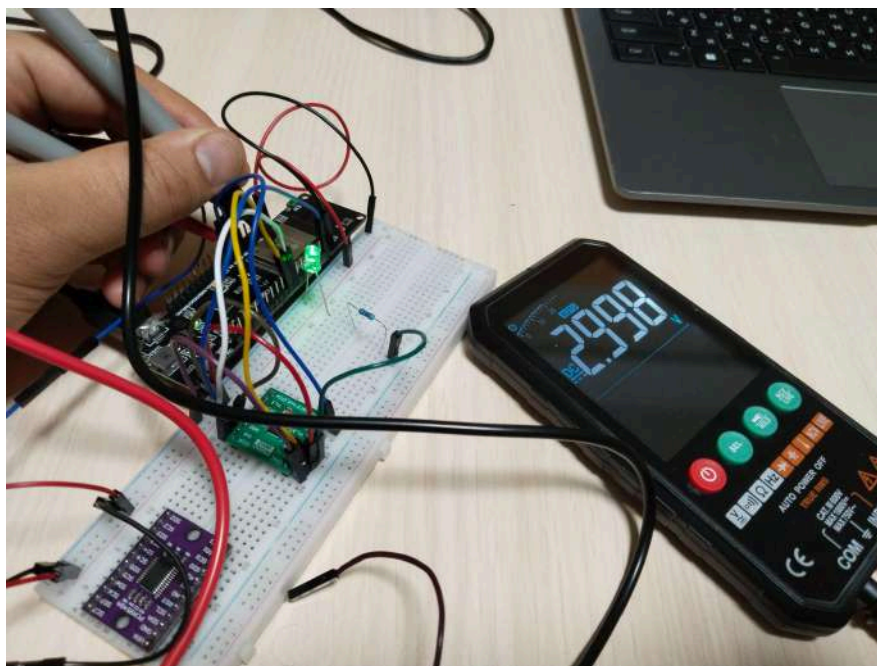


Рисунок 5.16 - Процесс получения опорного напряжения почти в 3В с микроконтроллера для ЦАП модуля.

5.3.3. NTC - термисторы для мониторинга температуры составляющих устройства

Термисторы в разработанной модели можно использовать не только для мониторинга температуры элементов Пельтье, но и также для реализации триггеров регулировки подачи тока на эти элементы.

В данном исследовании были применены NTC термисторы **B57861-S 103-F40** с сопротивлением 10 кОм (при 25 грд. Цельсия) с точностью 1%. График зависимости сопротивления от температуры для, используемого в нашем проекте, термистора - не линейная.



Рисунок 5.17 - Внешний вид используемого термистора

Схема подключения и состав модуля с NTC термистором описан в следующей схеме (рис.5.18):

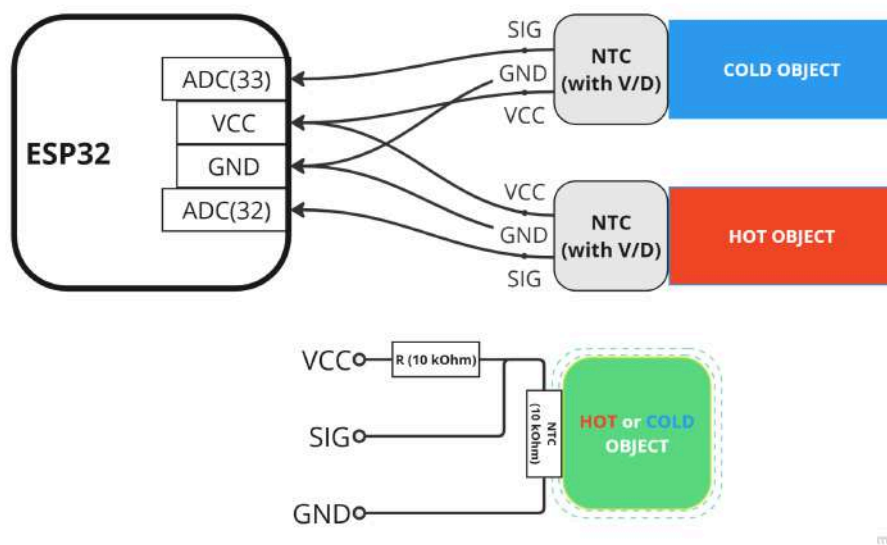


Рисунок 5.18 - Схема подключения и состав модуля с NTC термистором.

Где, V/D (Voltage Divider) - термистор, в составе делителя напряжения.

При вычислении температуры NTC термистора можно использовать специальную функцию:

$$B_{\frac{T_1}{T_2}} = \frac{T_2 \times T_1}{T_2 - T_1} \times \ln \frac{R_{T_1}}{R_{T_2}} \quad ()$$

Значения температур (T_1 и T_2) - в Кельвинах.

Отсюда, находим T_2 :

$$T_2 = \frac{B_{\frac{T_1}{T_2}} \times T_1}{B_{\frac{T_1}{T_2}} - \ln \frac{R_{T_1}}{R_{T_2}} \times T_1} [K] \quad ()$$

Где:

T_1 - базовая точка температуры, которая, обычно равняется 25°C

R_{T_2} - сопротивление термистора при температуре T_2 , которое получаем с термистора (которую находим с помощью описанных ниже формул).

R_{T_1} - сопротивление термистора при температуре T_1

$B_{\frac{T_1}{T_2}}$ - константа материала термистора, которая определяется керамическим материалом, из которого он изготовлен

Так как термистор подключен в составе делителя напряжения, в котором один из его выводов подключен к земле, его значение (R_{T_2}) можно найти с помощью следующих формул:

$$U_{SIG} = U_{VCC} \times \frac{R_2}{R_1 + R_2}$$

и

$$U_{SIG} = U_{VCC} \times \frac{ADC_{in}}{ADC_{bitRes}}$$

Где:

U_{SIG} - выходной сигнал (напряжение), которую получаем с делителя напряжения;

U_{VCC} - питание, которую подаем на делитель напряжения;

В нашем случае, U_{VCC} равняется напряжению вывода VCC используемого нами управляющего микроконтроллера. Его значение можем измерить с помощью мультиметра (относительно земли, GND);

R_2 - сопротивление используемого термистора (R_{T2}). Это то, что мы должны найти:

R_1 - опорный резистор, сопротивление которого равно сопротивлению используемого термистора при его комнатной температуре (R_{T1});

ADC_{in} - Значение, которую получаем с АЦП вывода управляющего микроконтроллера;

$ADC_{bitResVal}$ - значение битности АЦП управляющего микроконтроллера.

Приравнявая формулы () и () находим R_2 :

$$R_2 = R_{T2} = R_1 \times \frac{ADC_{in}}{ADC_{bitResVal} - ADC_{in}}$$

Найденные формулы () и () используются для вычисления температуры с термистора.

При использовании вышеописанных формул в ПО управляющей платы, можно получить следующие результаты на мониторе порта:

COM3	
Therm#1 (TI): 48.496244 °C,	Therm#2 (NS): 40.325408 °C
Therm#1 (TI): 48.208636 °C,	Therm#2 (NS): 40.136704 °C
Therm#1 (TI): 48.338285 °C,	Therm#2 (NS): 40.164903 °C
Therm#1 (TI): 48.338285 °C,	Therm#2 (NS): 40.216699 °C
Therm#1 (TI): 48.269955 °C,	Therm#2 (NS): 40.164903 °C
Therm#1 (TI): 48.269955 °C,	Therm#2 (NS): 40.136704 °C
Therm#1 (TI): 48.147485 °C,	Therm#2 (NS): 40.113233 °C
Therm#1 (TI): 48.399962 °C,	Therm#2 (NS): 40.245005 °C
Therm#1 (TI): 48.780622 °C,	Therm#2 (NS): 40.674164 °C
Therm#1 (TI): 48.976929 °C,	Therm#2 (NS): 40.809461 °C
Therm#1 (TI): 48.717887 °C,	Therm#2 (NS): 40.568457 °C
Therm#1 (TI): 48.461810 °C,	Therm#2 (NS): 40.377597 °C
Therm#1 (TI): 48.717887 °C,	Therm#2 (NS): 40.539719 °C
Therm#1 (TI): 48.592942 °C,	Therm#2 (NS): 40.511019 °C
Therm#1 (TI): 48.780622 °C,	Therm#2 (NS): 40.621245 °C

Рис. Результаты на мониторе порта.

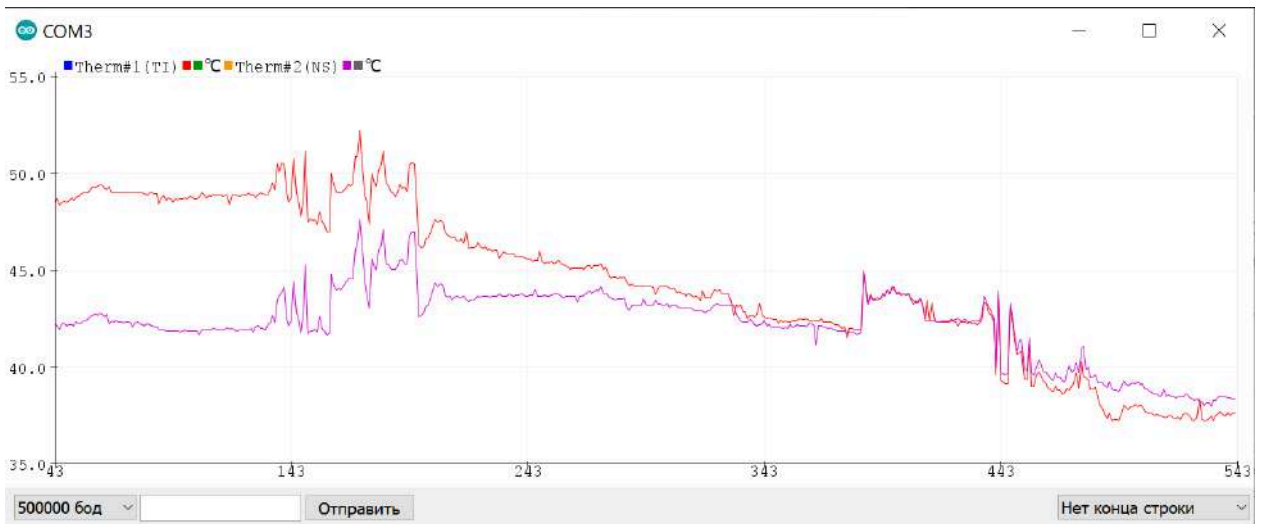


Рисунок 5.19 - Результаты на плоттере порта после удерживания термисторов на горячем металле.

При этом, задержка считывания данных с АЦП и вычисление температуры с него составляет примерно **70 мкс** (рис.5.20):

```

17  pinMode(NTC_TI_PIN, INPUT);
18  }
19
20  double calcTempNTC(int PIN) {
21      int md = micros();
22
23      double R_T2, T2;
24      int ADC_in;
25      ADC_in = analogRead(PIN);
26      R_T2 = R1 * ADC_in / (ADC_bitResVal - ADC_in);
27      T2 = B_T12 * T1 / (B_T12 - log(R_T1/R_T2) * T1);
28      T2 -= 273.15;
29
30      Serial.println(micros() - md);
31      return T2;
32  }
33
34  void loop() {
35      #if 0 //debug
36      Serial.printf("Ain#1: %i, \tAin#2: %i, \t",
37                  analogRead(NTC_TI_PIN),

```

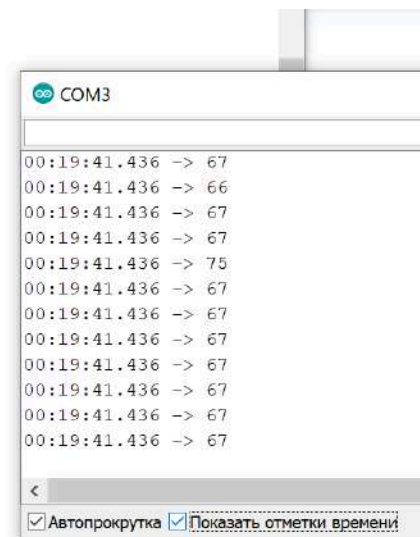


Рисунок 5.20 - Задержка считывания данных с АЦП и вычисление температуры с него.

5.3.4. Таблеточный вибромоторчик и его драйвер для воспроизведения хаптики

Существуют множество методов воспроизведения тактической обратной связи одним из них является использование тактильного вибромоторчика.

В данной работе были выбраны цилиндрические вибромоторчики с размерами 8х3,4мм или 10х2,7мм высоты и диаметра соответственно.

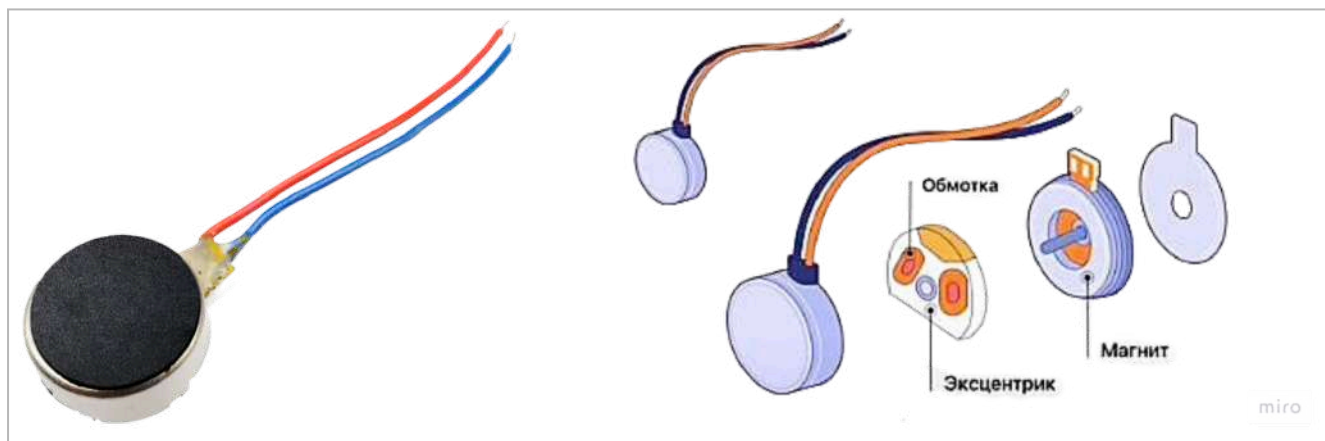


Рисунок 5.21 - Внешний вид используемого таблеточного вибромоторчика

В качестве драйвера был выбран чип DA7280 и сконструирована специальная тестовая плата (рис.), в которую можно с легкостью подключить и управляющий микроконтроллер и хаптический вибромоторчик.

С данным драйвером можно взаимодействовать как с помощью интерфейса I2C так и помощью GPI выводов для передачи ШИМ пульсаций.

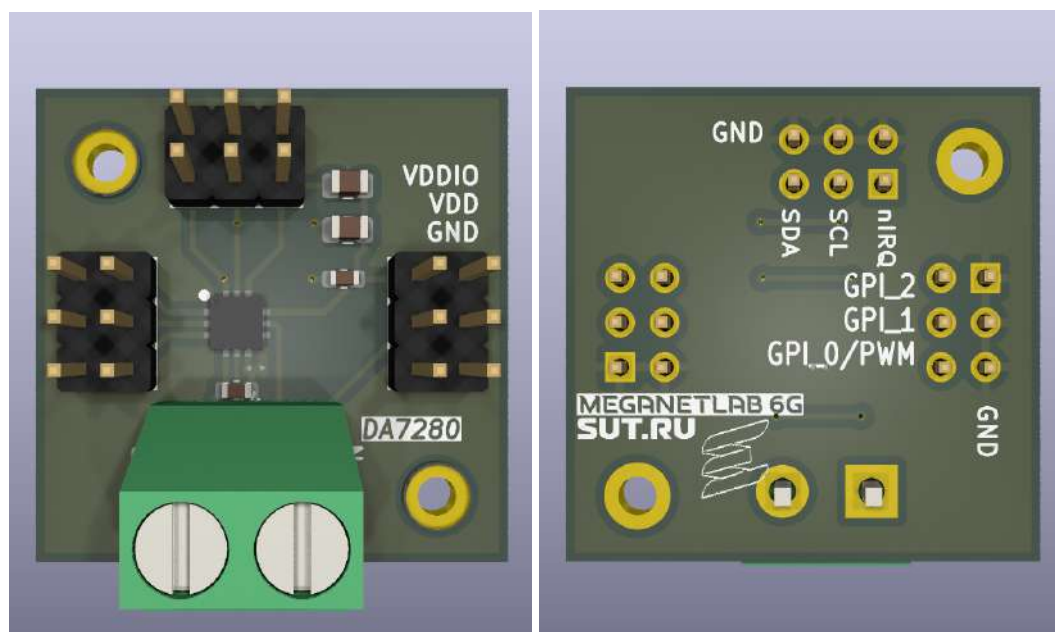


Рисунок 5.22 - Внешний вид тестовой платы с DA7280 (для управления LRA - вибромоторчиками)

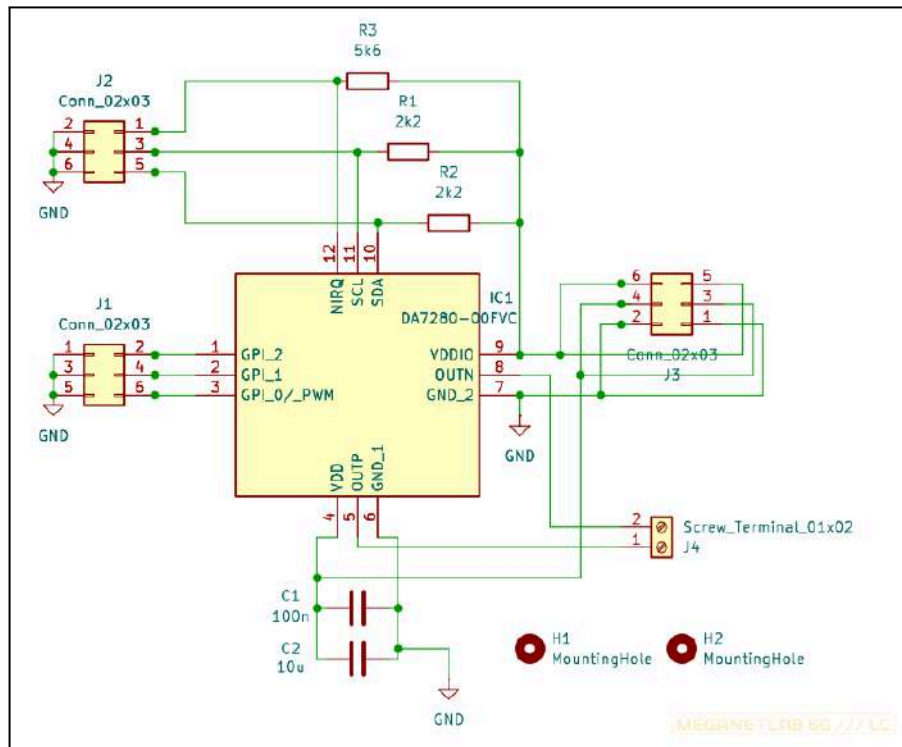


Рисунок 5.23 - Состав (схема) тестовой платы с DA7280 (для управления LRA - вибромоторчиками)

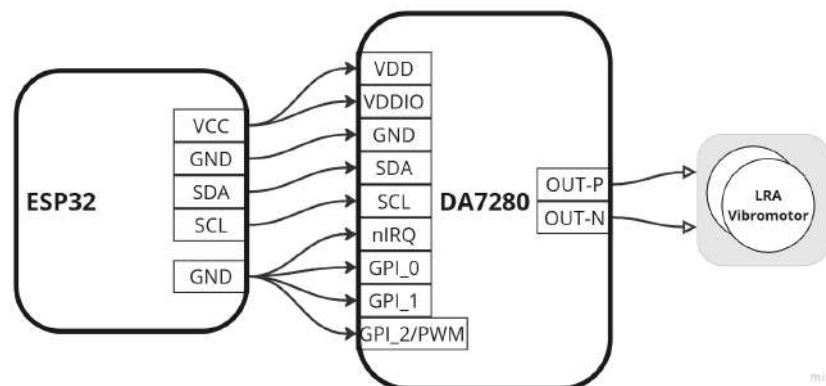


Рисунок 5.24 - Схема подключения вибромоторчика к управляющей плате через драйвер

Распиновка выводов платы:

- VDD - питание для вибромоторчика
- GND - общая земля
- VDDIO - питание для цифровых интерфейсов. Можно подключить к VDD при соблюдении условий описанных в официальной документации или в описании плат реализованных с применением этого драйвера
- SDA - вывод для обмена данных по I2C

- SCL - вывод для получения тактовых сигналов для I2C
- nIRQ - линия запроса прерывания к хосту. Сообщает управляющему микроконтроллеру о завершении какого либо события в драйвере DA7280
- GPI_0/PWM, GPI_1 и GPI_2 - выводы для управления через ШИМ или для воспроизведения заранее записанных на драйвер сигналов хаптики.

Для синхронного управления несколькими вибромоторчиками используются I2C переключатели PCA9548A.

Внешний вид и схема подключения приведены ниже:

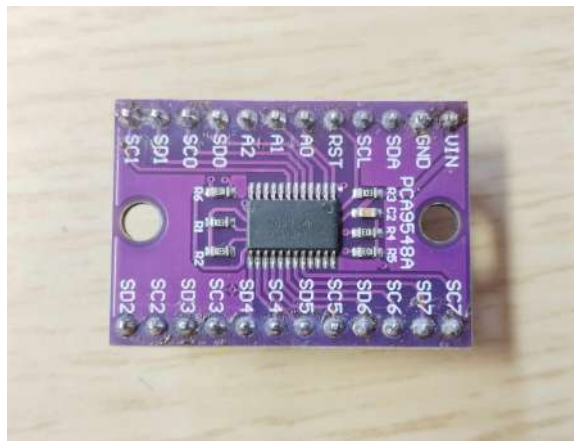


Рисунок 5.25 - Внешний вид переключателя.

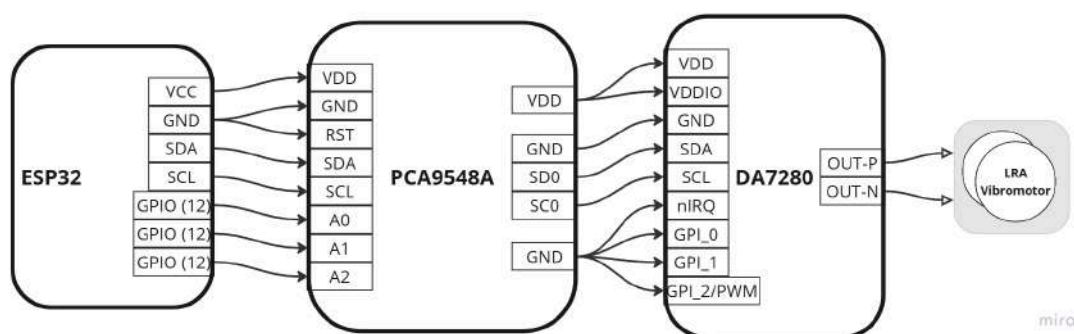


Рисунок 5.26 - Схема подключения вибромоторчика с применением I2C переключателя

5.3.5. Проверка температурной обратной связи с применением мини конструкции

Для проверки работоспособности элементов Пельтье, была сконструирована и напечатана специальная модель, в которую можно расположить элементы воспроизведения и мониторинга температуры, а также для воспроизведения хaptики (касания) и шероховатости объекта:

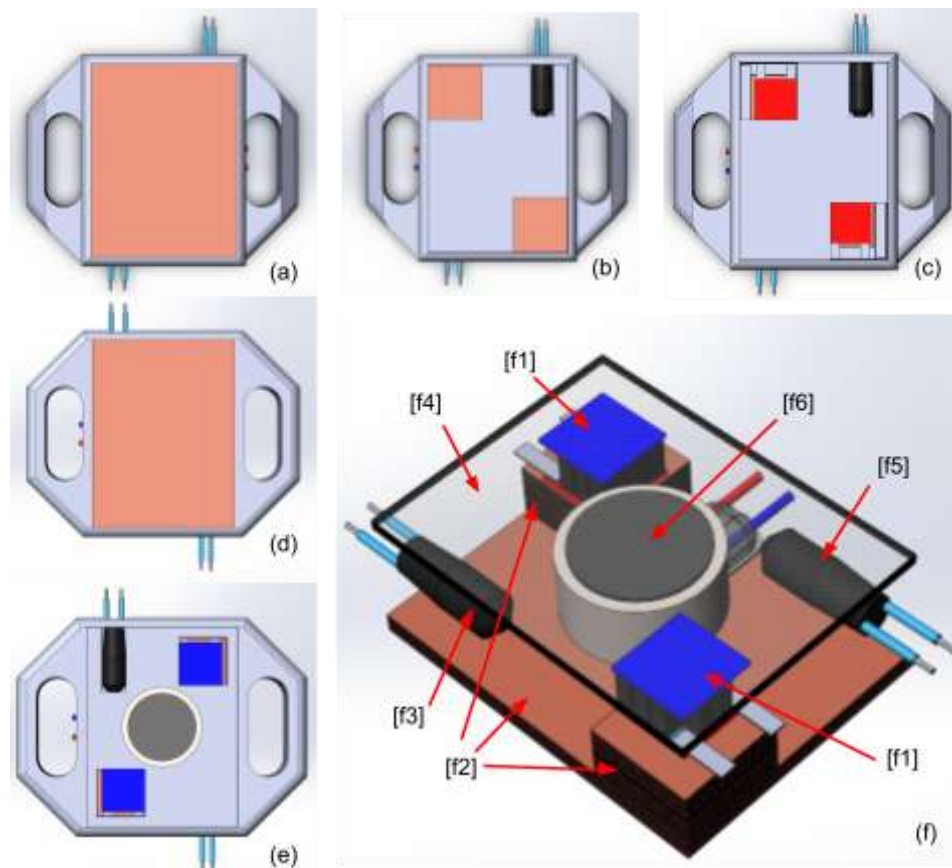


Рисунок 5.27 - Расположение элементов ТОС на тестовом каскаде для пальца.

Где:

(a) - (c) - расположение элементов верхней части модуля

(d) и (e) - расположение элементов нижней части модуля

(f) - состав элементов:

[f1] - элементы Пельтье;

[f2] - Слои медного скотча для теплоотвода для тепловыделяющей стороны элементов Пельтье;

[f3] - NTC термистор для измерения температуры теплоотводящей стороны элемента Пельтье;

[f4] - Слои медного скотча для передачи тепла и холода от элементов Пельтье;

[f5] - NTC термистор для измерения температуры тепло и холодо выделяющей стороны элемента Пельтье;

[f6] - Мини вибромоторчик в виде таблетки для тактической обратной связи.

Данный каскад был напечатан с использованием TPU75 - резины.

Процесс и результат подготовки сборки после печати показан на следующих рисунках (рис.):

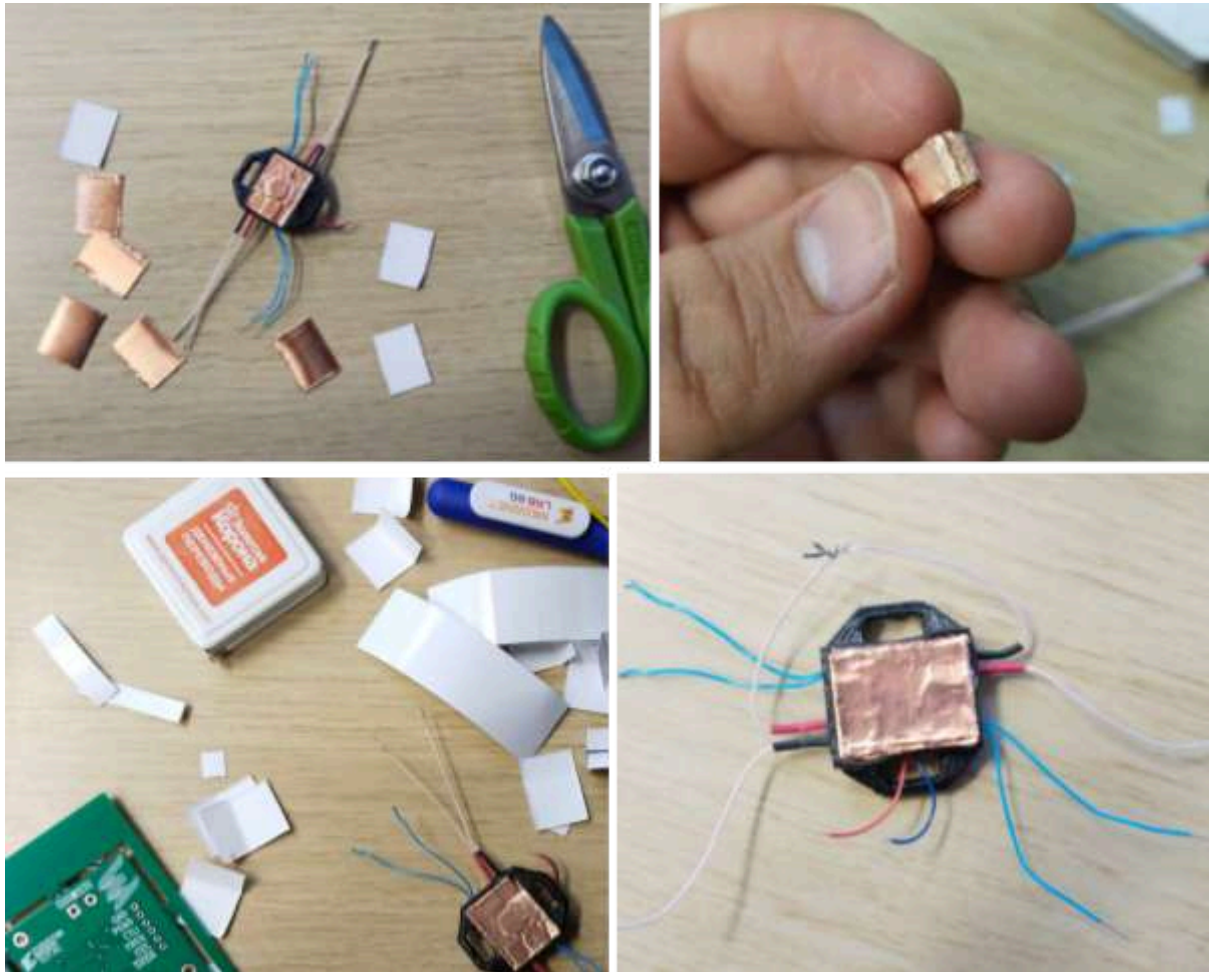


Рисунок 5.28 - Процесс подготовки тестового блока

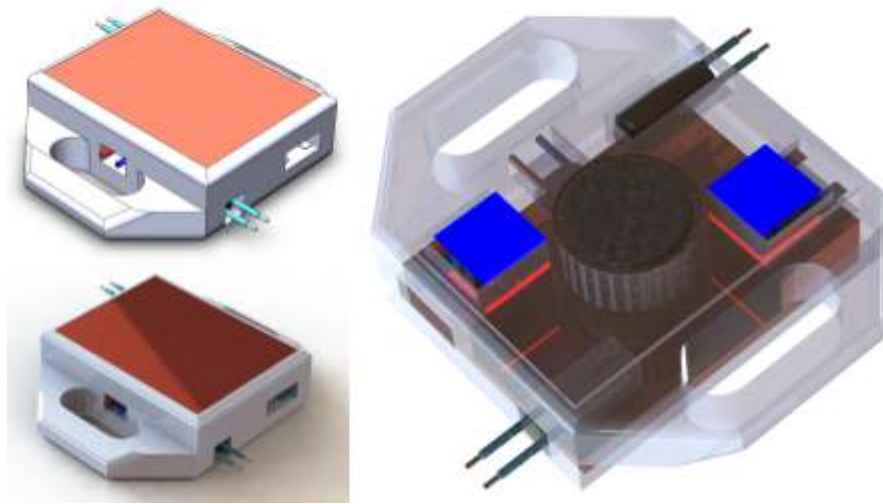


Рисунок 5.29 - Изометрический вид модели.

После подготовки сборки, подключим к NTC термистору делитель напряжения используя резистор 10кОм, а также подключим два элемента Пельтье последовательно к управляющему драйверу на подготовленной нами плате.

Схема подключений элементов сборки, более подробно выглядит следующим образом (рис.):

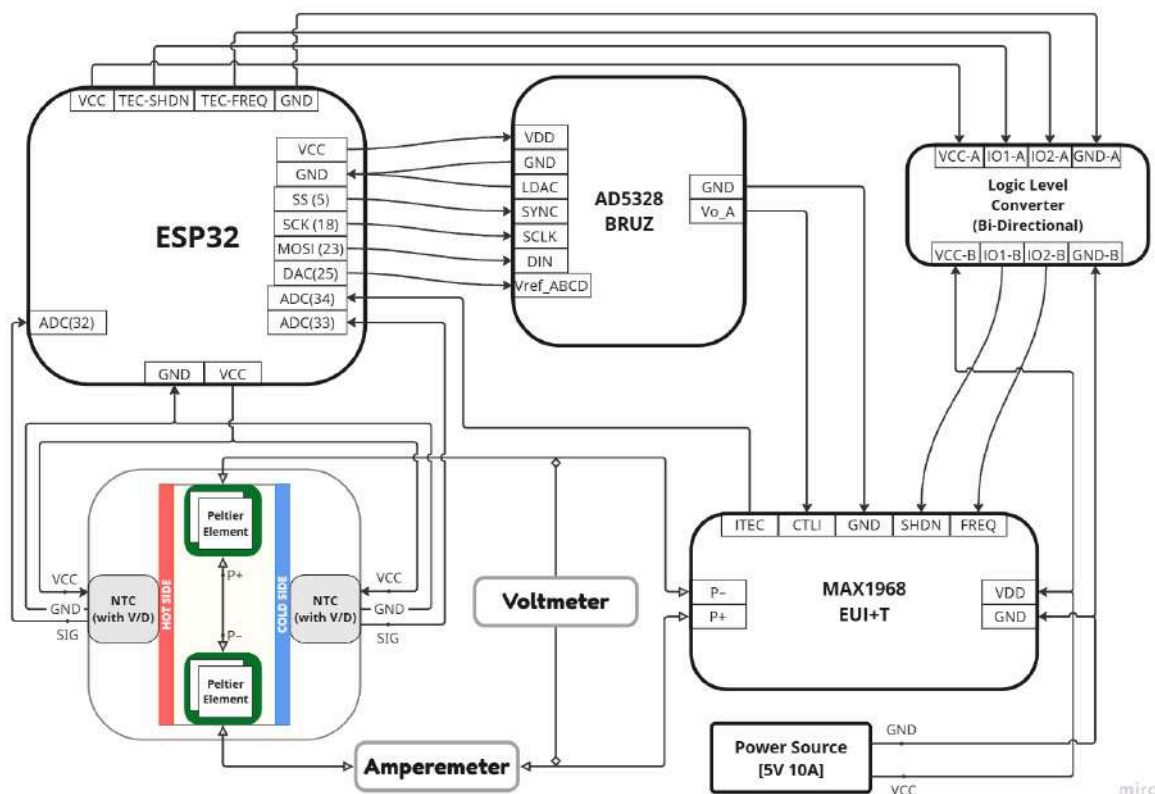


Рисунок 5.30 - Подробная схема подключения выводов для тестирования элементов температурной обратной связи.

Логику работы прошивки и его результаты можно видеть на следующих рисунках:

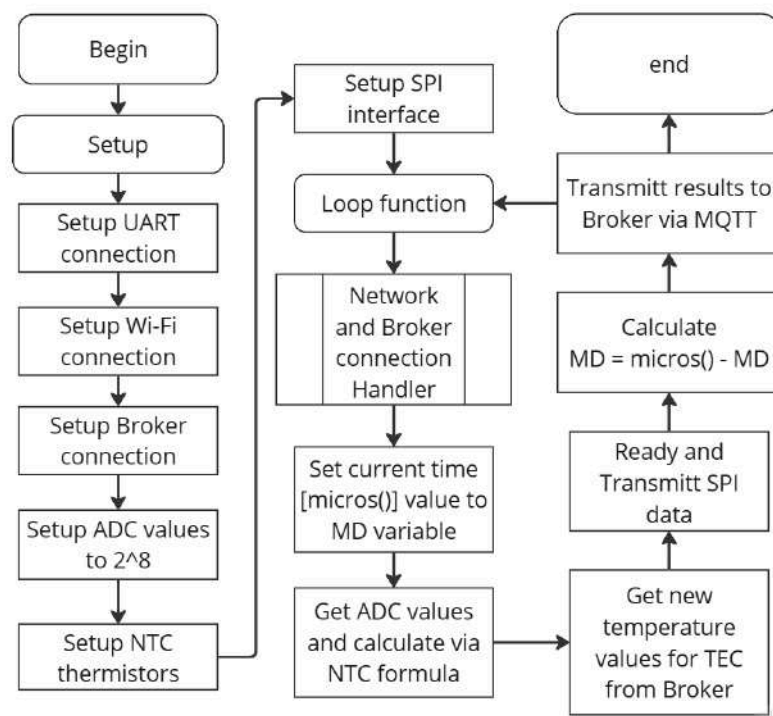


Рисунок 5.31 - Алгоритм работы программы для управления и мониторинга элемента Пельтье

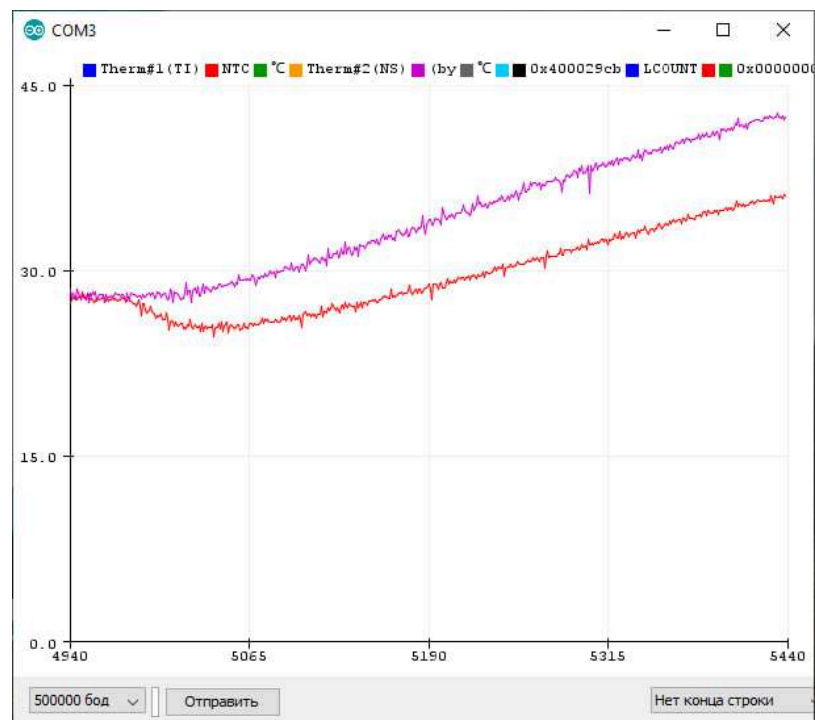


Рисунок 5.32 - Результаты термисторов после подачи тока элементам Пельтье

После проведения эксперимента, была выявлена, что нежелательно использовать многослойные медные листы, т.к. такие слои не смогут отвести тепло от элементов Пельтье.

Вместо этого, рекомендуется использовать более толстые однослойные формы для теплоотвода.

На графике ниже можно увидеть результаты другого эксперимента, в котором используется алюминиевый брусок для теплоотвода.

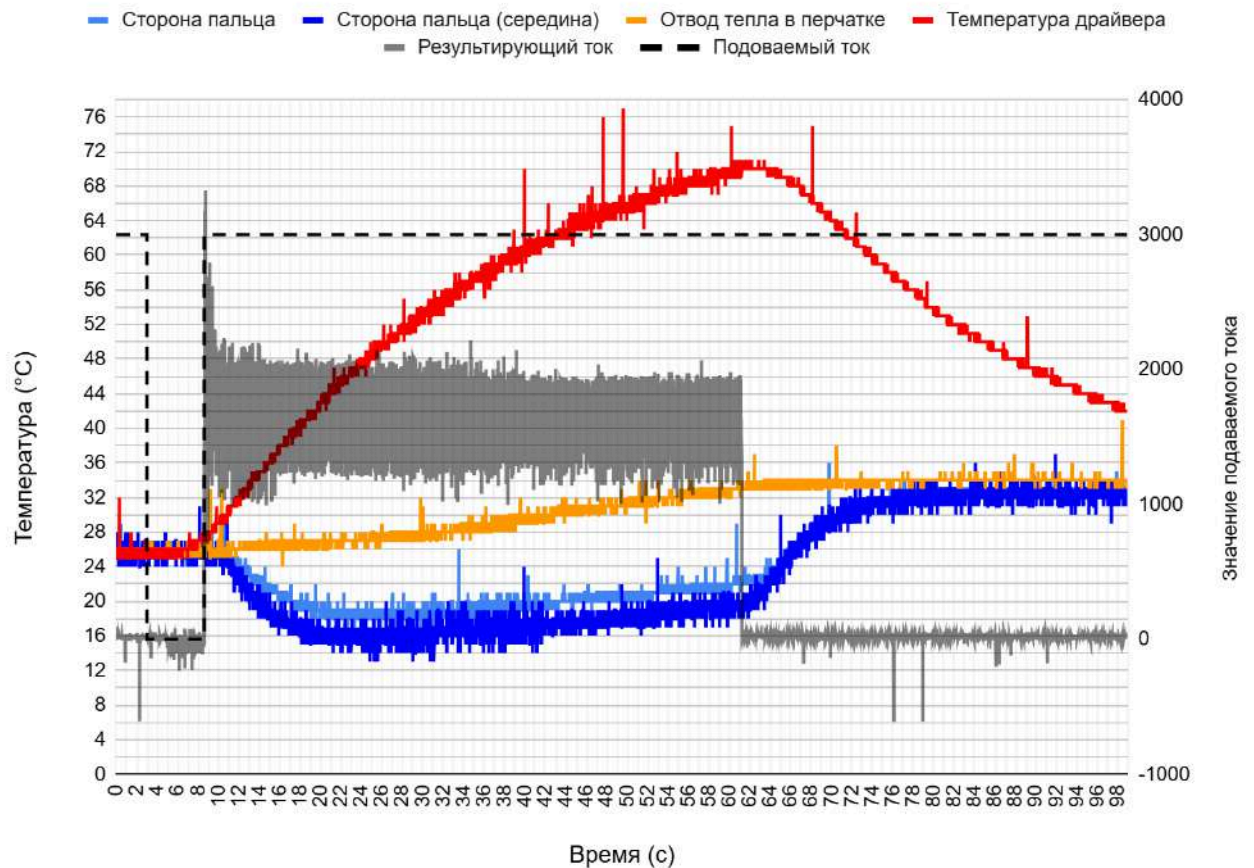


Рисунок 5.33 - Полученные данные с экспериментального блока с элементами Пельтье при передаче программного холода

Обозначения на легенде графика выше:

- ИТЕС - уровень подаваемого тока относительно настроенного на драйвере.
- темп DR - температура поверхности драйвера
- темп HS - температура теплоотвода (обратная сторона элемента Пельтье)
- темп TI и темп TIS - температура, подаваемая на палец пользователя.

5.4. Итоговая схема подключения составляющих тактильной перчатки

По итогам проведенных экспериментов, было решено подключить составляющие элементы иммерсивной перчатки по нижеприведенным схемам.

Схема подключения элементов температурной обратной связи для тактильной перчатки и исходящие от него выводы к управляющим платам можно увидеть на следующих схемах:

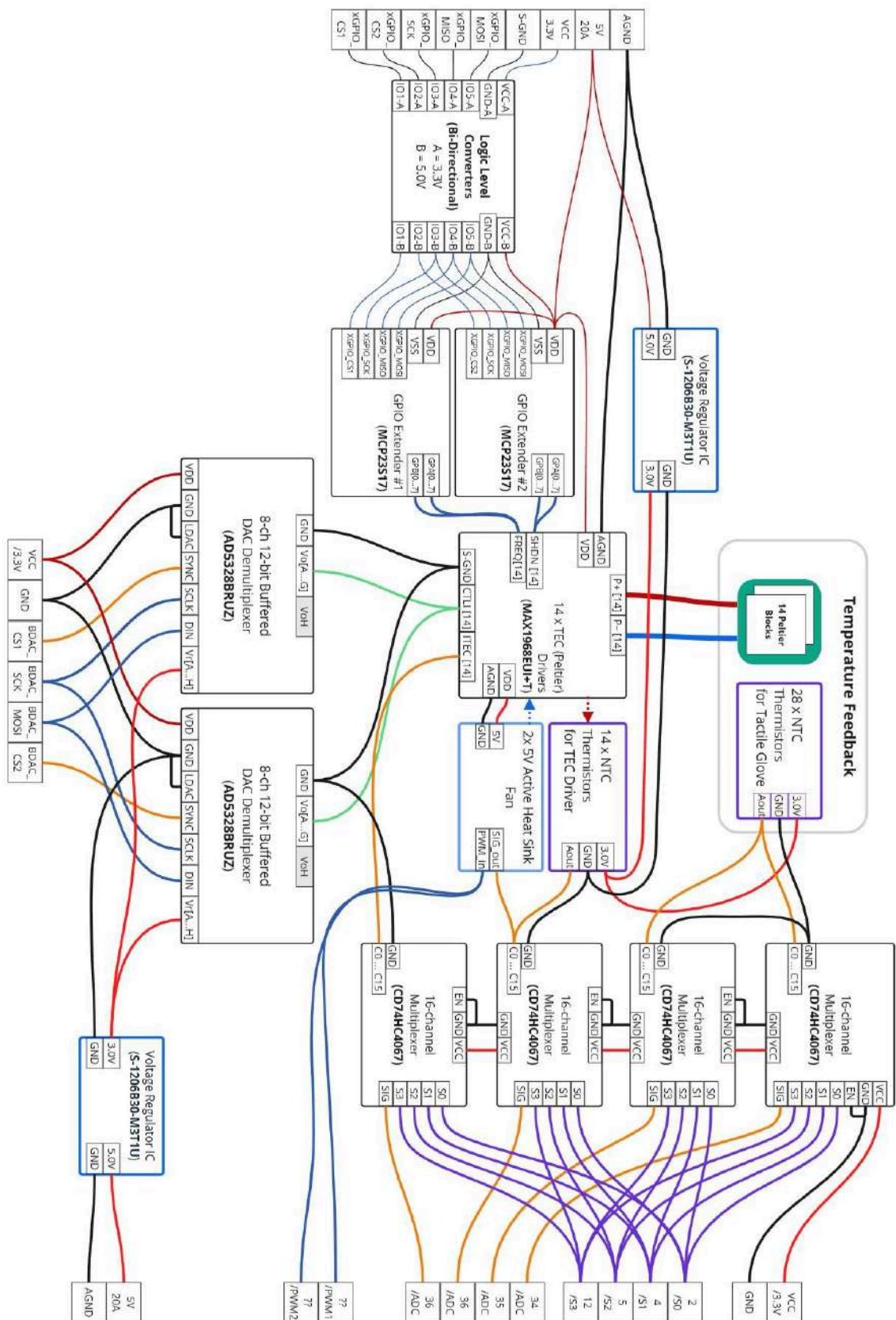
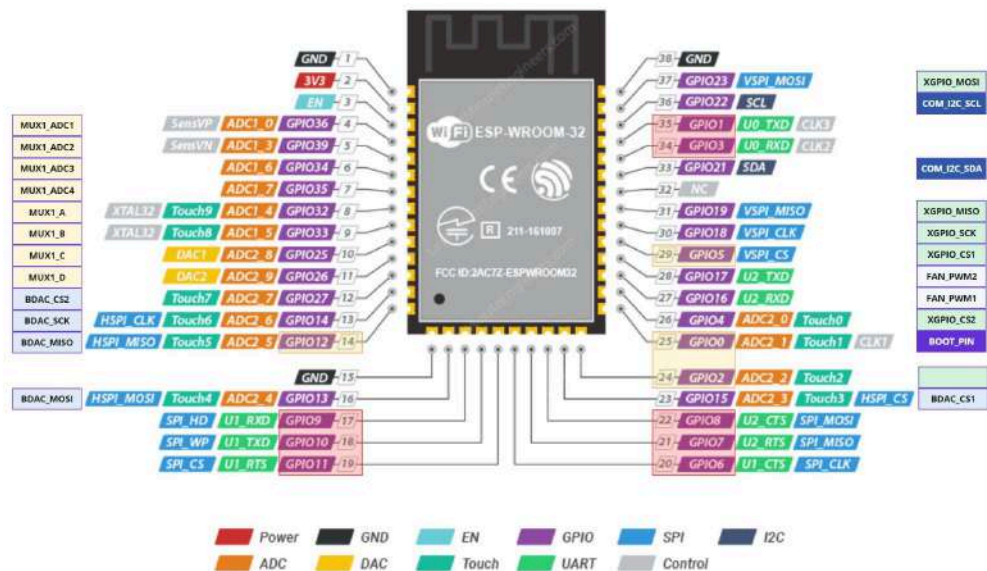


Рисунок 5.34 - Схема соединения элементов температурной обратной связи



ESP32-WROOM-32 Pinout

Last Minute
ENGINEERS.com

Рисунок 5.35 - Подключение выводов температурных актуаторов к первой плате

Для управления кинестетической обратной связью и хаптикой применяются следующие схемы подключений:

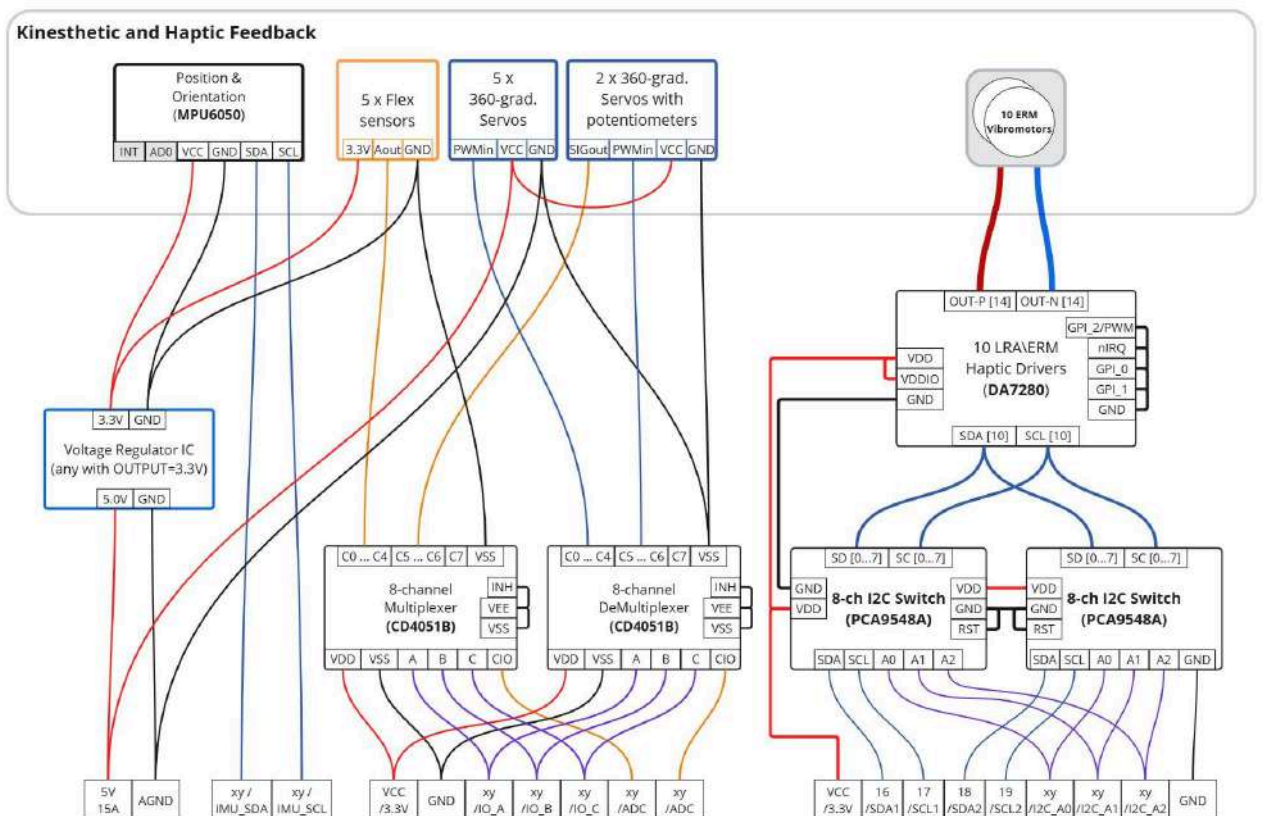


Рисунок 5.36 - Схема соединения элементов кинестетической и хаптической обратной связи

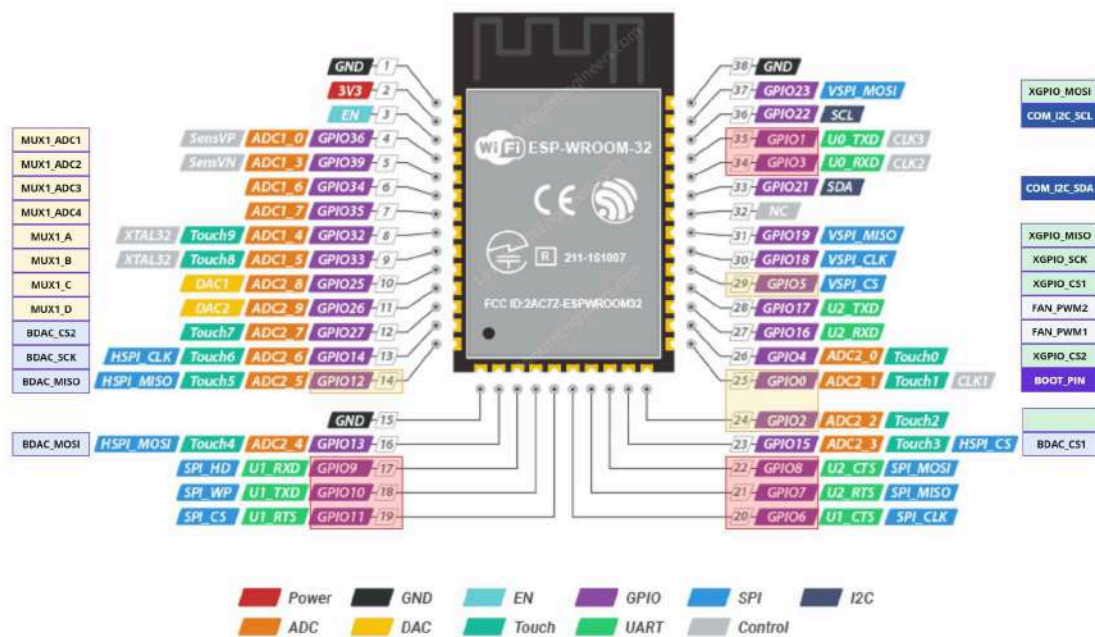


Рисунок 5.37 - Подключение выводов кинестетических и хаптических актуаторов ко второй плате

Для того чтобы вышеприведенные схемы работали корректно, необходимо стабильное питание.

Исходя из характеристик модулей и экспериментальных выходных значений, была составлена следующая схема питания:

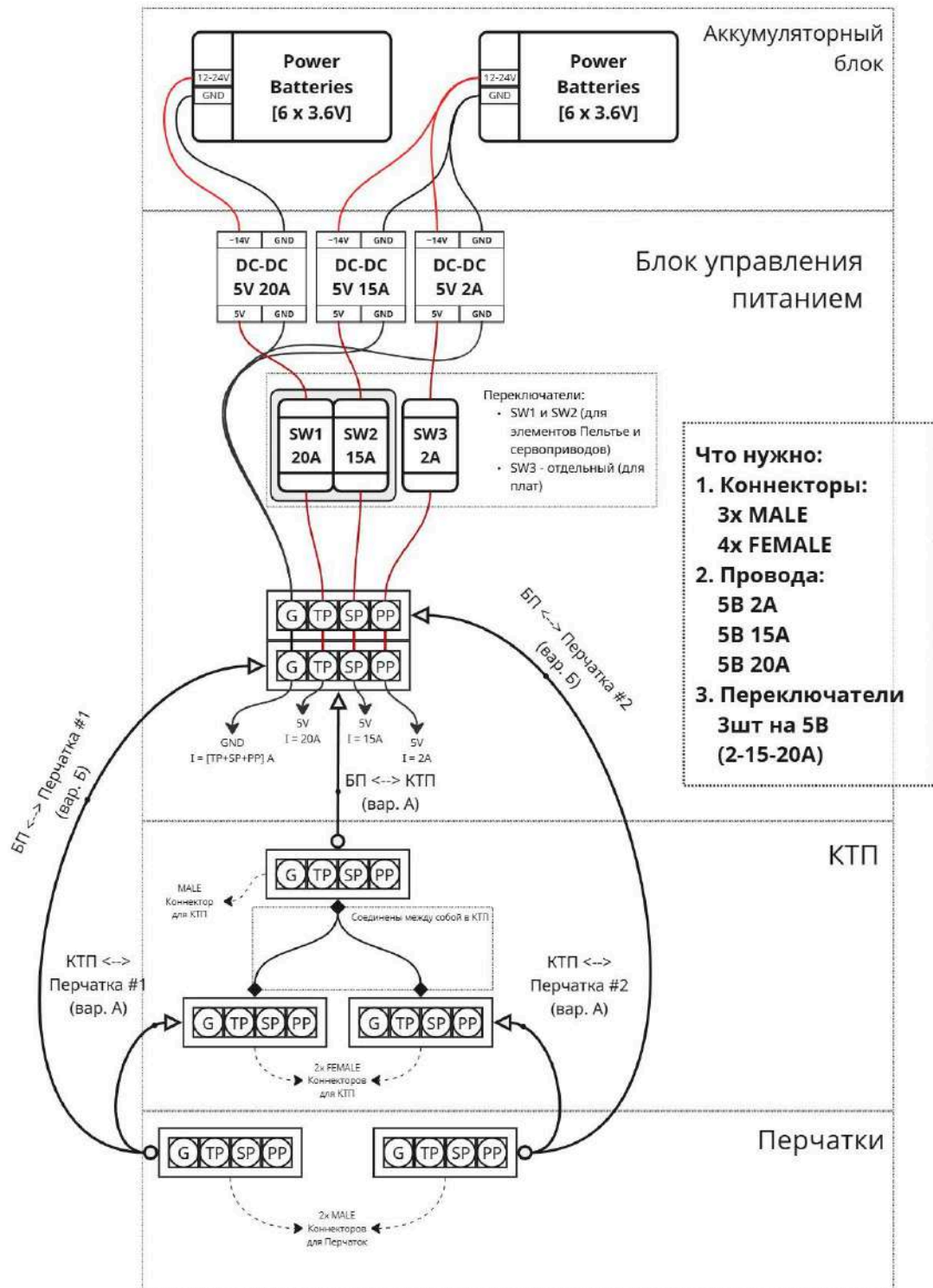


Рисунок 5.38 - Схема питания новой версии составляющих КТП

5.5. Разработка рекомендаций по реализации костюма телеприсутствия, доступного для ежедневного использования.

К основным рекомендациям по улучшению эргономики КТ для пользователя можно отнести следующие:

- Поиск замены используемых сервоприводов на шаговые двигатели, миниатюрных с наименьшими размерами и достаточными, для воспроизведения кинестетической обратной связи пользователю
- Реализация отдельных источников питания для сервоприводов (1), для управляющих плат (2) и для новых встраиваемых элементов в разрабатываемому новому методу воспроизведения тактильных ощущений
- улучшение дизайна конструкции КТ и его дополнительных составляющих для повышения привлекательности
- Применение более мягкой резины (например, TPU-D75) для печати наплечников для устранения дискомфорта пользователю
- и другие.

Кроме общих рекомендаций, также имеются и те, которые касаются и перчатки КТ.

Используемые на данной версии перчатки КТ, элементы и разработанные алгоритмы для них могут только имитировать тактильные обратные связи.

Но, для реального воспроизведения иммерсивных ощущений необходимы применить другие действующие актуаторы и специальные методы для их контроля.

По окончании разработки данной версии, были проанализированы и разработаны авторские методы воспроизведения иммерсивных данных.

К ним можно отнести воспроизведение температуры (тепла и холода), касания (с помощью хаптических вибраций) и кинестетической обратной связи (с применением мини сервоприводов и миниатюрных шаговых двигателей).

5.6. Выводы по главе

Данная глава посвящена разработке модельной сети для передачи иммерсивных ощущений в костюме телеприсутствия. Научная новизна работы заключается в создании комплексной архитектуры мультимодальной обратной связи, в которой предложено:

- совместное использование элемента Пельтье и NTC-термистора в замкнутом контуре терморегуляции с вычислением температуры на основе АЦП-данных (задержка 70 мкс);
- масштабирование управления температурными актуаторами через 8-канальный ЦАП AD5328BRUZ с SPI-интерфейсом (задержка 9 мкс на канал);
- синхронизация трёх типов обратной связи (температурной, хаптической, кинестетической) на единой аппаратной платформе с двухъядерной обработкой.

Практическая значимость подтверждена созданием действующих прототипов плат (на основе микроконтроллеров MAX1968, AD5328BRUZ, DA7280), разработкой модульной конструкции теплового блока и интеграцией всех подсистем в единую сеть с измеримыми временными параметрами, удовлетворяющими KPI тактильного интернета 6G.

Разработанные рекомендации по эргономике (переход на миниатюрные шаговые двигатели, отдельное питание, применение TPU-D75) создают основу для коммерциализации КТП и его внедрения в повседневную практику в сферах телемедицины, дистанционного обучения и виртуальной реальности.

ЗАКЛЮЧЕНИЕ

В диссертационной работе получены следующие основные результаты:

1. Решена актуальная научно-техническая задача разработки комплекса аппаратных, алгоритмических и сетевых решений для создания КТП как иммерсивного интерфейса взаимодействия пользователя с метавселенной и системами тактильного интернета;
2. Проанализированы существующие методы предоставления иммерсивных технологий;

3. Разработан метод сжатия динамического диапазона сигналов с помощью S-кривой. Данный метод позволяет на 20 % сократить разрядность двоичного представления сигнала (с 10 до 8 бит), что также дает возможность снизить величину трафика (примерно на 20 %, без учета служебных данных), производимого подсистемой термодатчиков КТП.

4. Разработан адаптивный метод передачи данных подсистемы термодатчиков, позволяющий повысить эффективность путем снижения объема передаваемых данных за счет использования вероятностных свойств сигналов термодатчиков. Метод позволяет получить выигрыш, величина которого зависит от количества термодатчиков и степени разброса значений температуры. Типовое значение выигрыша составляет около 30 %.

5. Анализ сигналов показал, что целесообразно выделить две группы сигналов от термодатчиков: «медленные», получаемые от сенсоров, реагирующих, в основном, на температуру окружающей среды, и «быстрые» сигналы от датчиков, реагирующих на прикосновения.

6. Анализ энергетических спектров сигналов показал, что верхняя граница спектра «медленных» сигналов составляет около 1 Гц, а «быстрых» – около 2 Гц. Это позволяет выбрать частоту дискретизации 2 Гц для «медленных» и 4 Гц для «быстрых» сигналов термодатчиков.

7. Разработана многоуровневая сетевая схема, включающая модули захвата движения (гироскопы, акселерометры, датчики изгиба) и модули обратной связи (сервоприводы, светодиоды, в будущем — элементы Пельтье). Определены функциональные требования к системе, включая двусторонний и асинхронный обмен данными между микроконтроллерами (ESP32) и сервером через Wi-Fi и протокол MQTT, что обеспечивает интеграцию КТП с метавселенной.

8. Разработаны и реализованы алгоритмы управления. Созданы и отлажены программные алгоритмы (прошивки) для микроконтроллеров, использующие двухъядерную архитектуру ESP32 для параллельной обработки: одно ядро обеспечивает высокую приоритетную связь с сетью (обмен данными по

MQTT), а другое — обработку сигналов с сенсоров (потенциометров и датчиков изгиба) и управление актуаторами (сервоприводами и светодиодами). Реализованы схемы подключения и обоснован выбор интерфейсов (АЦП, ЦАП, ШИМ, I2C) для мониторинга и управления всеми элементами системы.

9. Разработана модель передачи иммерсивных ощущений в костюме телеприсутствия, включающая три типа обратной связи

СПИСОК ЛИТЕРАТУРЫ

1. Анваржонов, Б. Н. У. Обработка данных подсистемы термодатчиков костюма телеприсутствия / Б. Н. У. Анваржонов, А. И. Парамонов // Труды учебных заведений связи. — 2026. — Т. 12, № 4. — С. 54–64. — DOI: 10.31854/1813-324X-2026-12-4-54-64. — EDN: NIFXTG.
2. Анваржонов, Б. Н. У. Анализ задач проектирования тактильного интернета и иммерсивных коммуникаций 6G / Б. Н. У. Анваржонов, А. С. А. Мутханна // Научно-техническая конференция Санкт-Петербургского НТО РЭС им. А. С. Попова, посвященная Дню радио. — 2026. — № 1 (81). — С. 200–202. — EDN: JPLIFY.
3. Анваржонов, Б. Н. У. Перспективные исследования сетей и услуг 2030 в лаборатории 6G MeganetLab СПбГУТ / Б. Н. У. Анваржонов, А. Н. Волков, А. С. А. Мутханна, А. Е. Кучерявый, А. С. Бородин, А. И. Парамонов, С. С. Владимиров, Г. А. Фокин, Р. А. Дунайцев, М. В. Захаров, Л. С. Горбачева, Б. О. Паньков // Электросвязь. — 2023. — № 6. — С. 5–14. — DOI: 10.34832/ELSV.2023.43.6.001. — EDN: CJSYLS.
4. Анваржонов, Б. Н. У. Development of a Telepresence Suit for Providing Holnetverse Services / Б. Н. У. Анваржонов, I. A. Elgandy, A. Volkov, A. Muthanna, A. Koucheryavy // 2024 IEEE Future Networks World Forum (FNWF), Dubai, United Arab Emirates, 2024. — IEEE, 2024. — P. 661–668. — DOI: 10.1109/FNWF63303.2024.11028141.
5. Анваржонов, Б. Н. У. Исследование применения сетевых протоколов в области IoT для разработки роботизированной инфраструктуры / Б. Н. У. Анваржонов, А. Н. Волков, Г. К. Инкин, А. П. Морачевский, Н. М. Саитов // Актуальные проблемы инфотелекоммуникаций в науке и образовании : сб. науч. ст. XIII Междунар. науч.-техн. и науч.-метод. конф. в 4 т. — СПб., 2024. — С. 41–45.
6. Анваржонов, Б. Н. У. Модельная сеть для исследований и обучения в области услуг телеприсутствия / Б. Н. У. Анваржонов, А. Е. Кучерявый, М. А.

- Маколкина, А. И. Парамонов, А. И. Выборнова, А. С. А. Мутханна, А. Ю. Матюхин, Р. А. Дунайцев, С. С. Владимиров, О. И. Ворожейкина, М. В. Захаров, В. Д. Фам, А. В. Марочкина, Л. С. Горбачева, Б. О. Паньков // *Электросвязь*. — 2022. — № 1. — С. 14–20.
7. Анваржонов, Б. Н. У. A First-Priority Set of Telepresence Services and a Model Network for Research and Education / Б. Н. У. Анваржонов, А. Е. Koucheryavy, М. А. Makolkina, А. I. Paramonov, А. I. Vybornova, А. S. A. Muthanna, R. A. Dunaytsev, S. S. Vladimirov, V. S. Elagin, O. A. Markelov, O. I. Vorozheykina, A. V. Marochkina, L. S. Gorbacheva, B. O. Pankov // Springer Nature Switzerland AG 2023 V. M. Vishnevskiy et al. (Eds.): DCCN 2022, CCIS 1748. — 2023. — P. 208–219.
 8. Анваржонов, Б. Н. У. Методика обработки кинестетической информации для системы сбора, передачи и воспроизведения данных тактильного интернета / Б. Н. У. Анваржонов, А. И. Выборнова // Подготовка профессиональных кадров в магистратуре для цифровой экономики (ПКМ-2022) : сб. лучших докладов Всерос. науч.-техн. и науч.-метод. конф. магистрантов и их руководителей / сост. Н. Н. Иванов. — СПб. : СПбГУТ, 2023. — С. 27–32.
 9. Анваржонов, Б. Н. У. Свидетельство о государственной регистрации программы для ЭВМ № 2026687694 Российская Федерация. Модуль управления элементами Пельтье для воспроизведения данных температуры для костюма телеприсутствия : заявл. 10.09.2026 : опубл. 10.09.2026 / заявитель ФГБОУ ВО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича».
 10. Анваржонов, Б. Н. У. Свидетельство о регистрации программы для ЭВМ № 2025697583 Российская Федерация. Модуль передачи аудиосигнала с минимальной задержкой на базе ESP32 для систем распознавания речи : заявл. 25.12.2025 : опубл. 25.12.2025 / Б. Н. У. Анваржонов, Л. С. Горбачева, Д. В. Назаров, А. А. Чуева, З. В. Ким.

11. Анваржонов, Б. Н. У. Свидетельство о регистрации программы для ЭВМ № 2025697612 Российская Федерация. Сервер низкой задержки для передачи и распознавания аудиопотока ESP32 : заявл. 25.12.2025 : опубл. 25.12.2025 / Б. Н. У. Анваржонов, Л. С. Горбачева, Д. В. Назаров, А. А. Чуева, З. В. Ким.
12. Анваржонов, Б. Н. У. Свидетельство о регистрации программы для ЭВМ № 2025610100 Российская Федерация. Программное обеспечение модуля обработки кинестетической информации кистей оператора костюма телеприсутствия : заявл. 10.12.2024 : опубл. 09.01.2025 / Б. Н. У. Анваржонов, А. Н. Волков, Л. С. Горбачева, В. Е. Каялайнен, В. В. Могилатов.
13. Анваржонов, Б. Н. У. Свидетельство о регистрации программы для ЭВМ № 2024691808 Российская Федерация. Программное обеспечение модуля обработки кинестетической информации верхних конечностей оператора костюма телеприсутствия : заявл. 10.12.2024 : опубл. 24.12.2024 / Б. Н. У. Анваржонов, А. Н. Волков, Л. С. Горбачева, В. Е. Каялайнен, В. В. Могилатов.
14. Анваржонов, Б. Н. У. Свидетельство о регистрации программы для ЭВМ № 2023682183 Российская Федерация. Программный модуль исследования зависимости точности позиционирования и ориентации устройств в пространстве от числа элементов реконфигурируемой интеллектуальной поверхности и режима диаграммообразования в сверхплотных сетях миллиметрового и терагерцового диапазона : заявл. 12.10.2023 : опубл. 23.10.2023 / Б. Н. У. Анваржонов, Г. А. Фокин, А. Е. Кучерявый.
15. Анваржонов, Б. Н. У. Сети связи шестого поколения: фрактальные архитектуры, голографические взаимодействия, телеприсутствие, сетевые роботы : монография / Б. Н. У. Анваржонов, А. Е. Кучерявый, А. Н. Волков, А. А. А. Давуд, А. С. А. Мутханна, А. И. Парамонов, В. С. Елагин, Р. А. Дунайцев, С. С. Владимиров, Л. С. Горбачева, М. В. Захаров, А. В. Марочкина, Б. О. Паньков, Д. С. Свечников. — СПб. : Питер, 2024. — 320 с. — ISBN 978-5-4461-4289-7. — (Вклад соискателя: С. 171–216).

16. Анваржонов, Б. Н. У. Прикладные научные исследования в области создания сетей связи 2030, включая услуги телеприсутствия с сетевой поддержкой, и экспериментальная проверка решений при подготовке отраслевых кадров. Вторая очередь : отчет о НИР / Б. Н. У. Анваржонов, А. В. Шестаков, Н. Н. Громова, А. Е. Кучерявый, М. А. Маколкина, А. И. Парамонов, А. И. Выборнова, А. С. А. Мутханна, А. Ю. Матюхин, Р. А. Дунайцев, С. С. Владимиров, Л. С. Горбачева, О. И. Ворожейкина, Б. О. Паньков, К. Э. Есалов, А. В. Помогалова, А. Е. Селиванов, Е. Ю. Куликов, А. А. Сербин, А. С. Попонин [и др.]. — М. : Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации, 2022. — 187 с.
17. Анваржонов, Б. Н. У. Разработка программно-аппаратного комплекса для сбора, передачи и воспроизведения кинестетической информации в рамках концепции тактильного интернета / Б. Н. У. Анваржонов, А. И. Выборнова, Е. С. Сапунова // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022) : сб. науч. ст. XI Междунар. науч.-техн. и науч.-метод. конф. в 4 т. / под ред. А. В. Шестакова ; сост. В. С. Елагин, Е. А. Аникевич. — СПб., 2022. — С. 373–378.
18. Анваржонов, Б. Н. У. Будущие роботизированные системы: особенности, проблемы и прогнозирование / Б. Н. У. Анваржонов, К. Х. Киличева, А. С. А. Мутханна // Перспективные технологии в средствах передачи информации : материалы 14-й Междунар. науч.-техн. конф. — Владимир, 2021. — С. 75–77.
19. ГОСТ Р 51337-99. Безопасность машин. Температуры касаемых поверхностей. Эргономические данные для установления предельных величин горячих поверхностей. — М. : Госстандарт России, 1999.
20. ГОСТ Р ИСО 13732-3-2013. Эргономика термальной среды. Методы оценки реакции человека при контакте с поверхностями. Ч. 3. Контакт с холодными поверхностями. — М. : Стандартиформ, 2014.
21. ГОСТ 302-79. Термометр медицинский максимальный стеклянный. Технические условия. — М. : ИПК Издательство стандартов, 1979.

22. IETF RFC 8406. Taxonomy of Coding Techniques for Efficient Network Communications. — 2018.
23. Rec. ITU-T Y.4238. Requirements for integrating virtual and physical worlds through digital twins in the metaverse. — Geneva : ITU, 2025.
24. FG-NET2030. Network 2030 — Architecture Framework. — Geneva : ITU-T Focus Group on Technologies for Network 2030, 2020.
25. ITU-T Recommendation Y.4238 (11/2025). Requirements for integrating virtual and physical worlds through digital twins in the metaverse. — Geneva : ITU, 2025.
26. ITU Focus Group on Metaverse. Technical Specification ITU FGMV-20. Definition of metaverse. — Geneva : International Telecommunication Union, 2023.
27. ITU-T Technical Report. Networks in 2030 — Gap Analysis of New Services, Capabilities and Use Cases. — Geneva : ITU-T, 2020. — 43 p.
28. Network 2030. A Blueprint of Technology, Applications and Market Drivers Towards the Year 2030 and Beyond / FGNET-2030. — Geneva : ITU-T, 2019. — 19 p.
29. Iversen, V. B. Teletraffic engineering and network planning / V. B. Iversen. — Lyngby : DTU Fotonik, 2015.
30. Басараб, М. А. Цифровая обработка сигналов на основе теоремы Уиттекера-Котельникова-Шеннона / М. А. Басараб, Е. Г. Зелкин, В. Ф. Кравченко, В. П. Яковлев. — М. : Радиотехника, 2004. — 72 с. — EDN: OVRCIY.
31. Matthew, R. Perceptual and Conceptual Distortions of Implicit Hand Maps / R. Matthew, S. Mattioni, N. Ganea // *Frontiers in Human Neuroscience*. — 2015. — Vol. 9. — DOI: 10.3389/fnhum.2015.00656.
32. Palkar, R. The molecular and cellular basis of thermosensation in mammals / R. Palkar, E. K. Lippoldt, D. D. McKemy // *Current Opinion in Neurobiology*. — 2015. — Vol. 34. — P. 14–19. — DOI: 10.1016/j.conb.2015.01.010.

33. Атея, А. А. А. Тактильный интернет как телекоммуникационные системы будущего / А. А. А. Атея, А. С. А. Мутханна, А. Е. Кучерявый // Информационные технологии и телекоммуникации. — 2018. — Т. 6, № 4. — С. 1–9. — EDN: XNECEZ.
34. Ateya, A. A. Enabling Metaverse and Telepresence Services in 6G Networks / A. A. Ateya, A. A. Abd El-Latif, A. Muthanna, A. Volkov, A. Koucheryavy. — New York : River Publishers, 2025. — DOI: 10.1201/978877004674914.
35. Кучерявый, А. Е. Сети связи шестого поколения: фрактальные архитектуры, голографические взаимодействия, телеприсутствие, сетевые роботы / А. Е. Кучерявый, А. А. А. А. Давуд, А. Н. Волков, А. С. А. Мутханна, А. И. Парамонов, В. С. Елагин [и др.]. — СПб. : Питер, 2024. — 318 с.
36. Volkov, A. Optimized Data Transmission and Signal Processing for Telepresence Suits in Multiverse Interactions / A. Volkov, A. Muthanna, A. Paramonov, A. Koucheryavy, I. A. Elgendy // Journal of Sensor and Actuator Networks. — 2024. — Vol. 13, Iss. 6. — P. 82. — DOI: 10.3390/jsan13060082. — EDN: VBEMLB.
37. Anvarjonov, B. Development of a Telepresence Suit for Providing Holnetverse Services / B. Anvarjonov, I. A. Elgendy, A. Volkov, A. Muthanna, A. Koucheryavy // Proceedings of the Future Networks World Forum (FNWF), Dubai, United Arab Emirates, 15–17 October 2024. — IEEE, 2024. — P. 661–668. — DOI: 10.1109/FNWF63303.2024.11028141.
38. Волков, А. Н. Сети связи пятого поколения: на пути к сетям 2030 / А. Н. Волков, А. С. А. Мутханна, А. Е. Кучерявый // Информационные технологии и телекоммуникации. — 2020. — Т. 8, № 2. — С. 32–43. — DOI: 10.31854/2307-1303-2020-8-2-32-43. — EDN: ZWNTDB.
39. Аль-Кереа, З. А. Х. Архитектурная эволюция сетей: распределённый интеллект и конвергенция космических, воздушных, наземных и подводных сетей связи / З. А. Х. Аль-Кереа, А. С. А. Мутханна, А. Е. Кучерявый // Электросвязь. — 2025. — № 10. — С. 63–71.

40. Кучерявый, А. Е. Искусственный интеллект в сетях связи / А. Е. Кучерявый, А. С. Бородин, А. С. А. Мутханна, А. Р. Абделлах, А. Н. Волков // Актуальные проблемы инфотелекоммуникаций в науке и образовании : сб. науч. ст. — СПб. : СПбГУТ, 2021. — С. 8–18.
41. Кучерявый, А. Е. Новые перспективы научных исследований в области сетей связи на 2021–2024 годы / А. Е. Кучерявый, Р. В. Киричек, М. А. Маколкина [и др.]. — СПб. : СПбГУТ, 2020.
42. Кучерявый, А. Е. Сети связи ожидает блестящее наукоёмкое будущее / А. Е. Кучерявый // Электросвязь. — 2022. — № 1. — С. 3–5. — EDN: DPIVMC.
43. Abdellah, A. R. Deep Learning for Predicting Traffic in V2X Networks / A. R. Abdellah, A. Muthanna, M. H. Essai, A. Koucheryavy // Applied Sciences. — 2022. — Vol. 12, Iss. 19. — DOI: 10.3390/app121910030. — EDN: NPFGMK.
44. Iftikhar, S. Deep Learning-Based Pedestrian Detection in Autonomous Vehicles: Substantial Issues and Challenges / S. Iftikhar, Z. Zhang, M. Asim [et al.] // Electronics. — 2022. — Vol. 11, Iss. 21. — DOI: 10.3390/electronics11213551. — EDN: AJUYZX.
45. Бородин, А. С. Искусственный интеллект в сетях связи пятого и последующих поколений / А. С. Бородин, А. Н. Волков, А. С. Мутханна, А. Е. Кучерявый // Электросвязь. — 2021. — № 1. — С. 17–22. — EDN: JDHISS.
46. Singh, D. A Generalized Approach on Outage Performance Analysis of Dual-Hop Decode and Forward Relaying for 5G and Beyond Scenarios / D. Singh, M. A. Ouamri, M. S. A. Muthanna [et al.] // Sustainability. — 2022. — Vol. 14, Iss. 19. — 12870. — EDN: KJBAVU.
47. Mahmood, O. A. Distributed Edge Computing for Resource Allocation in Smart Cities Based on the IoT / O. A. Mahmood, A. R. Abdellah, A. Muthanna, A. Koucheryavy // Information. — 2022. — Vol. 13, Iss. 7. — DOI: 10.3390/info13070328. — EDN: AWGNYK.

48. Ateya, A. A. Energy efficient offloading scheme for MEC-based augmented reality system / A. A. Ateya, A. Muthanna, A. Koucheryavy [et al.] // Cluster Computing. — 2023. — Vol. 26, Iss. 1. — P. 1–18. — EDN: TIOVAV.
49. Кашкаров, Д. В. Метод повышения эффективности URLLC в перспективных сетях связи / Д. В. Кашкаров, А. И. Парамонов, А. Е. Кучерявый // Электросвязь. — 2022. — № 2. — С. 32–37. — EDN: UQLXES.
50. Чистова, Н. А. Метод выбора размеров цифровых кластеров сетей с ультрамалыми задержками / Н. А. Чистова, А. И. Парамонов, А. И. Выборнова, А. Е. Кучерявый // Электросвязь. — 2021. — № 4. — С. 43–48. — EDN: XDBSPA.
51. Akyildiz, I. F. Holographic-type communication: A new challenge for the next decade / I. F. Akyildiz, H. Gu // ITU Journal on Future and Evolving Technologies. — 2022.
52. Кучерявый, А. Е. Модельная сеть для исследований и обучения в области услуг телеприсутствия / А. Е. Кучерявый, М. А. Маколкина, А. И. Парамонов [и др.] // Электросвязь. — 2022. — № 1. — С. 14–20. — EDN: GBQWCV.
53. Ahmad, A. Determination of Critical Edges in Air Route Network Using Modified Weighted Sum Method and Grey Relational Analysis / A. Ahmad, T. Ahmad, M. Ahmad [et al.] // IEEE Transactions on Intelligent Transportation Systems. — 2022. — DOI: 10.1109/TITS.2022.3200140. — EDN: KVPBIM.
54. Ullah, S. Position-Monitoring-Based Hybrid Routing Protocol for 3D UAV-Based Networks / S. Ullah, K. H. Mohammadani, M. A. Khan [et al.] // Drones. — 2022. — Vol. 6, Iss. 11. — DOI: 10.3390/drones6110327. — EDN: KEKZNJ.
55. Alwan, M. H. High Density Sensor Networks Intrusion Detection System for Anomaly Intruders Using the Slime Mould Algorithm / M. H. Alwan, Y. I. Hammadi, O. A. Mahmood [et al.] // Electronics. — 2022. — Vol. 11, Iss. 20. — DOI: 10.3390/electronics11203332. — EDN: MYKNTC.
56. Vorobyova, D. IoT Network Model with Multimodal Node Distribution and Data-Collecting Mechanism Using Mobile Clustering Nodes / D. Vorobyova, A.

- Muthanna, A. Paramonov [et al.] // *Electronics*. — 2023. — Vol. 12, Iss. 6. — DOI: 10.3390/electronics12061410. — EDN: BHNVEU.
57. Bushelenkov, S. Multi-Story Building Model for Efficient IoT Network Design / S. Bushelenkov, A. Paramonov, A. Muthanna [et al.] // *Mathematics*. — 2023. — Vol. 11, Iss. 6. — DOI: 10.3390/math11061403. — EDN: VMJVEM.
 58. Paramonov, A. Fractals for Internet of Things Network Structure Planning / A. Paramonov, E. Tonkikh, A. Muthanna [et al.] // *International Journal of Information Security and Privacy*. — 2022. — Vol. 16, Iss. 1. — P. 1–12.
 59. Тонких, Е. В. Планирование структуры сети интернета вещей с использованием фракталов / Е. В. Тонких, А. И. Парамонов, А. Е. Кучерявый // *Электросвязь*. — 2021. — № 4. — С. 55–62. — EDN: RTHYZH.
 60. Ateya, A. A. Drone-based AI/IoT Framework for Monitoring, Tracking and Fighting Pandemics / A. A. Ateya, A. D. Algarni, A. Koucheryavy, N. F. Soliman // *Computers, Materials & Continua*. — 2022. — Vol. 71, Iss. 2. — P. 4677–4699. — EDN: YSDZSR.
 61. Nahli, F. Novel Path Counting-Based Method for Fractal Dimension Estimation of the Ultra-Dense Networks / F. Nahli, A. Paramonov, N. F. Soliman [et al.] // *Intelligent Automation & Soft Computing*. — 2023. — Vol. 36, Iss. 1. — P. 561–572. — EDN: ZAHRIY.
 62. Дунайцев, Р. А. Интегрированная сеть космос-воздух-земля-море как основа сетей связи шестого поколения / Р. А. Дунайцев, А. С. Бородин, А. Е. Кучерявый // *Электросвязь*. — 2022. — № 10. — С. 5–8. — EDN: QCGIPI.
 63. Vladimirov, S. S. Network Coding Datagram Protocol for TCP/IP Networks / S. S. Vladimirov, A. Vybornova, A. Muthanna [et al.] // *IEEE Access*. — 2023. — P. 43485–43498. — EDN: TNEZMC.
 64. Fokin, G. Algorithm for Topology Search Using Dilution of Precision Criterion in Ultra-Dense Network Positioning Service Area / G. Fokin, A. Koucheryavy // *Mathematics*. — 2023. — Vol. 11, Iss. 10. — DOI: 10.3390/math11102227. — EDN: PYZIJQ.

65. Бутенко, В. В. Сети 5G/IMT-2020&IoT — основа цифровой трансформации / В. В. Бутенко, В. В. Веерпалу, Е. Е. Девяткин, Д. Н. Федоров // Электросвязь. — 2018. — № 12. — С. 4–9. — EDN: VOHOSA.
66. Бородин, А. С. Сети связи пятого поколения как основа цифровой экономики / А. С. Бородин, А. Е. Кучерявый // Электросвязь. — 2017. — № 5. — С. 45–49. — EDN: YNGMKT.
67. Leppanen, K. Key Drivers and Research Challenges for 6G / K. Leppanen, M. Latvaaho // Fifth ITU Workshop on Network 2030, Geneva, Switzerland, 14–16 October 2019. — URL: <https://www.itu.int/en/ITU-T/Workshops-and-Seminars/2019101416/Pages/programme.aspx>.
68. Кучерявый, А. Е. Сети связи 2030 / А. Е. Кучерявый, А. С. Бородин, Р. В. Киричэк // Электросвязь. — 2018. — № 11. — С. 52–56. — EDN: YMVOPB.
69. Andreev, S. Intelligent Access Network Selection in Converged Multi-Radio Heterogeneous Networks / S. Andreev, M. Gerasimenko, O. Galinina [et al.] // IEEE Wireless Communications. — 2014. — Vol. 21, № 6. — P. 86–96.
70. Yastrebova, A. Future Networks 2030: Architecture and Requirements / A. Yastrebova, R. Kirichek, Y. Koucheryavy [et al.] // The 10th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT), November 2018, Moscow : Proceedings. — 2018. — P. 31–36. — EDN: ZACQPZ.
71. Гегель, Г. В. Ф. Наука логики / Г. В. Ф. Гегель. — СПб. : Наука, 1997.
72. Galinina, O. 5G Multi-RAT LTE-WiFi Ultra-Dense Small Cells: Performance, Dynamics, Architecture and Trends / O. Galinina, A. Pyattaev, S. Andreev [et al.] // IEEE Journal on Selected Areas in Communications. — 2015. — P. 1224–1240.
73. Тонких, Е. В. Анализ беспроводной сети интернета вещей высокой плотности / Е. В. Тонких, А. И. Парамонов, А. Е. Кучерявый // Электросвязь. — 2020. — № 1. — С. 51–55. — EDN: IWAHZO.

74. Li, Z. 5G URLLC: Design Challenges and System Concepts / Z. Li, M. Uusitalo, H. Shariatmadari, B. Singh // 15th International Symposium on Wireless Communication Systems (ISWCS), Tokyo, Japan, 8–9 October 2018. — P. 6.
75. Кучерявый, А. Е. Сети связи с ультрамалыми задержками / А. Е. Кучерявый // Труды НИИР. — 2019. — № 1. — С. 69–74. — EDN: XTLZDZ.
76. Кучерявый, А. Е. Самоорганизующиеся сети / А. Е. Кучерявый, А. В. Прокопьев, Е. А. Кучерявый. — СПб. : Любавич, 2011. — 212 с. — EDN: RICRYV.
77. Кучерявый, А. Е. Интернет Вещей / А. Е. Кучерявый // Электросвязь. — 2013. — № 1. — С. 21–24. — EDN: PVTCFH.
78. Кучерявый, А. Е. Тактильный Интернет. Сети связи со сверхмалыми задержками / А. Е. Кучерявый, М. А. Маколкина, Р. В. Киричек // Электросвязь. — 2016. — № 1. — С. 44–46. — EDN: VLPHZZ.
79. Maier, M. The tactile internet: vision, recent progress, and open challenges / M. Maier, M. Chowdhury, B. P. Rimal, D. P. Van // IEEE Communications Magazine. — 2016. — № 54 (5). — P. 138–145.
80. Кучерявый, А. Е. Государственный парк систем коммутации / А. Е. Кучерявый, В. Н. Осипов // Вестник связи. — 1999. — № 3. — С. 35–38.
81. Васильев, А. Б. Тестирование сетей связи следующего поколения / А. Б. Васильев, Д. В. Тарасов, Д. В. Андреев, А. Е. Кучерявый. — М. : ЦНИИС, 2008. — 142 с. — EDN: QMSTDV.
82. Kirichek, R. Internet of Things Laboratory Test Bed / R. Kirichek, A. Koucheryavy // International Conference on Wireless Communication, Networking and Application (WCNA 2016). — LNEE, Vol. 348. — P. 485–494. — EDN: UZPPZT.
83. Kirichek, R. Model networks for internet of things and SDN / R. Kirichek, A. Vladyko, Z. Makharov, A. Koucheryavy // 18th International Conference on Advanced Communication Technology (ICACT). — 2016. — P. 76–79. — EDN: VQHLZJ.

84. Бородин, А. С. Сети связи и пандемия / А. С. Бородин, А. Е. Кучерявый // Электросвязь. — 2020. — № 5. — С. 8–10. — EDN: UYTWTR.
85. Li, R. Network 2030: Market Drivers and Prospects / R. Li // First ITU Workshop on Network 2030, New York City, US, October 2018. — URL: <https://www.itu.int/en/ITU-T/Workshops-and-Seminars/201810/Pages/Programme.aspx>.
86. Vega, M. T. Towards Truly Immersive Holographic-Type Communication: Challenges and Solutions / M. T. Vega // 4th ITU Workshop on Network 2030, S.-Petersburg, Russia, 21–23 May 2019. — P. 45–52. — URL: <https://www.itu.int/en/ITU-T/Workshops-and-Seminars/201905/Pages/programme.aspx>.
87. Листвин, В. Н. DWDM системы / В. Н. Листвин, В. Н. Трещиков. — М. : Техносфера, 2015. — 338 с.
88. Muthanna, A. Secure and reliable IoT networks using fog computing with software-defined networking and blockchain / A. Muthanna, A. A. Ateya, A. Khakimov [et al.] // Journal of Sensor and Actuator Networks. — 2019. — Т. 8, № 1. — С. 15. — EDN: VXLAEA.
89. Volkov, A. SDN Load Prediction Algorithm Based on Artificial Intelligence / A. Volkov, K. Proshutinskiy, A. B. Adam [et al.] // Communications in Computer and Information Science. — Springer, 2019. — Vol. 1141. — P. 27–40. — DOI: 10.1007/978-3-030-36625-43. — EDN: SSKXAK.

ПРИЛОЖЕНИЕ А - Исполняемый код ESP-32

Листинг. Скетч-код для плат с ESP32 текущей версии верхних конечностей КТП

```

/*
© Copyright
Баходиржон Анваржонов (anvarjonovb@gmail.com)
*/

#define APP_VER "exoTI v2.1b"

//Параметры подключения:
#define FW_TYPE 'L2' // Выберите блок (L1 - левый ближний, L2 - левый дальний, R1 - правый
ближний, R2 - правый дальний)
#define USE_STATIC_IP true // true = Статическое IP; false = Динамическое IP
#define TEST_MODE true

#if TEST_MODE
const char* WLAN_SSID = "space_lab";
const char* WLAN_PASSWD = "space_508";
const char* MQTT_BROKER = "192.168.31.137";
#else
const char* WLAN_SSID = "MEGANET_6G_LAB";
const char* WLAN_PASSWD = "space_508";
const char* MQTT_BROKER = "192.168.31.205";
#endif

const int MQTT_PORT = 1883;
String MY_MQTT_ID = "ESP32-CLI_";
#if USE_STATIC_IP
  #if (FW_TYPE == 'L1')
    int my_ip[4] = {192, 168, 31, 140}; // статичный IP адрес для платы L1
  #elif (FW_TYPE == 'L2')
    int my_ip[4] = {192, 168, 31, 141}; // статичный IP адрес для платы L2
  #elif (FW_TYPE == 'R1')
    int my_ip[4] = {192, 168, 31, 142}; // статичный IP адрес для платы R1
  #elif (FW_TYPE == 'R2')
    int my_ip[4] = {192, 168, 31, 143}; // статичный IP адрес для платы R2
  #endif
#endif
int rtr_gateway[4] = {192, 168, 31, 0};
int rtr_subnet[4] = {255, 255, 255, 0};
int rtr_primaryDNS[4] = {8, 8, 8, 8}; //optional
int rtr_secondaryDNS[4] = {8, 8, 4, 4}; //optional

//Параметры для модулей (сенсоров и актуаторов) :
#define USE_SERVOS true
#define FORCE_MOVE_TO_LIMIT false
#define PON_MA_BEFORE 0
#define MA_FILO_BEFORE_CNT 2
#define MA_FILO_BEFORE_LEN 3

```

```

#define PON_NC 1
#define NC_AMMOUNT_OF_VALUES 3
#define NC_MAX_AMMOUNT_OF_VALUES 3
#define NC_CALC_NOICE_INT_TIME 200 // Время для определения границ помех
#define NC_NOICE_INTERVALS_COEF 1
#define PON_MA_AFTER 0
#define MA_FILO_AFTER_CNT 2
#define MA_FILO_AFTER_LEN 3
#define FILLER 100

// #####
// Исполняемый код:

// Подключение библиотек:
#include <PubSubClient.h> // Библиотека для работы с MQTT
#include <esp_random.h> // Нужна для рандомизации ID MQTT клиента

#include <ESP32Servo.h> // Библиотека для работы с сервоприводами

#include <WiFi.h> // Библиотека для работы с WiFi
#include <AsyncTCP.h> // Нужна для работы ElegantOTA
#include <ESPAsyncWebServer.h> // Нужна для работы ElegantOTA
#include <ElegantOTA.h> // Библиотека для беспроводной прошивки

#include "NoiceCorrector.h" // Самодельный фильтр - скользящая средняя
#include "MovingAverageFILO.h" // Самодельный фильтр - скользящая средняя

// Создание объекта для новой задачи (FreeRTOS / Многозадачность / Многопоточность):
TaskHandle_t SA_taskHandler;

// Создание объекта для работы с WiFi как клиент:
WiFiClient client;

// Конфигурации WiFi для статического IP:
IPAddress myIP(my_ip[0], my_ip[1], my_ip[2], my_ip[3]);
IPAddress gateway(rtr_gateway[0], rtr_gateway[1], rtr_gateway[2], rtr_gateway[3]);
IPAddress subnet(rtr_subnet[0], rtr_subnet[1], rtr_subnet[2], rtr_subnet[3]);
IPAddress primaryDNS(rtr_primaryDNS[0], rtr_primaryDNS[1], rtr_primaryDNS[2], rtr_primaryDNS[3]);
IPAddress secondaryDNS(rtr_secondaryDNS[0], rtr_secondaryDNS[1], rtr_secondaryDNS[2],
rtr_secondaryDNS[3]);

// Для беспроводной прошивки:
AsyncWebServer server(80);

// Создание объектов и переменных для работы с MQTT:
PubSubClient mqtt(client);

bool MQTT_reconnect();
void MQTT_ReceiverCallback(char*, uint8_t*, size_t);
uint8_t *received;
String TXbuff, TXdata, STRbuff;

```

```

// Создание объектов самодельного фильтра "Корректора помех" для работы с аналоговыми сигналами:
#if (FW_TYPE == 'L1' || FW_TYPE == 'R1')
NoiseCorrector NC[2] = {NoiseCorrector(NC_AMMOUNT_OF_VALUES,
NC_MAX_AMMOUNT_OF_VALUES)};
#elif (FW_TYPE == 'L2' || FW_TYPE == 'R2')
NoiseCorrector NC[3] = {NoiseCorrector(NC_AMMOUNT_OF_VALUES,
NC_MAX_AMMOUNT_OF_VALUES)};
#endif

// Создание объектов самодельного фильтра "Скользящего среднего" для работы с аналоговыми
сигналами:
#if (FW_TYPE == 'L1' || FW_TYPE == 'R1')
MovingAverageFILO MA_BEFORE[2] = {MovingAverageFILO(MA_FILO_BEFORE_CNT,
MA_FILO_BEFORE_LEN)};
MovingAverageFILO MA_AFTER[2] = {MovingAverageFILO(MA_FILO_AFTER_CNT,
MA_FILO_AFTER_LEN)};
#elif (FW_TYPE == 'L2' || FW_TYPE == 'R2')
MovingAverageFILO MA_BEFORE[3] = {MovingAverageFILO(MA_FILO_BEFORE_CNT,
MA_FILO_BEFORE_LEN)};
MovingAverageFILO MA_AFTER[3] = {MovingAverageFILO(MA_FILO_AFTER_CNT,
MA_FILO_AFTER_LEN)};
#endif

// Создание объектов для работы с подключенными сервоприводами:
#if (FW_TYPE == 'L1' || FW_TYPE == 'R1')
Servo servo[2];
#elif (FW_TYPE == 'L2' || FW_TYPE == 'R2')
Servo servo[3];
#endif

//Лимиты потенциометров, встроенных в сервоприводы:
/*int angle_limits[10][2] = {
    {592, 3725}, //L1-1
    {0, 2326},   //L1-2

    {675, 2730}, //L2-1
    {560, 3329}, //L2-2
    {1078, 3574}, //L2-3

    {0, 2700},   //R1-1
    {547, 2517}, //R1-2

    {844, 3377}, //R2-1
    {528, 3350}, //R2-1
    {17, 3885}   //R2-3
};*/
int angle_limits[10][2] = {
    {74, 466},   //L1-1
    {0, 290},    //L1-2

    {84, 341},   //L2-1
    {70, 404},   //L2-2
    {135, 447},  //L2-3

    {0, 337},    //R1-1
    {68, 315},   //R1-2

```

```

{105, 422}, //R2-1
{66, 419}, //R2-1
{2, 486} //R2-3
};

// Середины для нулевой скорости сервоприводов (0~90 -> вращение в левую сторону; ~90 ->
остановка вращения; ~90-180 -> вращение в правую сторону)
// Середины определяются экспериментально и варьируются в диапазоне 90:
int servo_null[10] = {
    90, //L1-1
    90, //L1-2

    90, //L2-1
    90, //L2-2
    90, //L2-3

    90, //R1-1
    90, //R1-2

    90, //R2-1
    90, //R2-1
    90 //R2-3
};

#include "OTA_Handlers.h"

// Настройка выводов платы, подготовка подключений, калибровка сенсоров и т.д.:
void setup() {
    // Запуск монитора порта (для отладки/Debug):
    Serial.begin(500000);
    Serial.println();
    Serial.println(APP_VER);

    // Конфигурация и Подключение к WiFi:

    #if USE_STATIC_IP // Конфигурация статического IP:
        if (!WiFi.config(myIP, gateway, subnet, primaryDNS, secondaryDNS)) {
            Serial.println("STA Failed to configure");
        }
    #endif
    WiFi.mode(WIFI_STA);
    WiFi.begin(WLAN_SSID, WLAN_PASSWD);
    Serial.print("\n\nПодключение к WLAN ..");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        vTaskDelay(500);
    }
    Serial.println("\nПодключено к " + String(WLAN_SSID));
    Serial.print("\nМой IP адрес: ");
    Serial.println(WiFi.localIP());

    // Подготовка Сервера для OTA прошивки платы:
    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
        request->send(200, "text/html",

```

```

"<!DOCTYPE html>"\
"<head><meta name='color-scheme' content='light dark'></head>"\
"Hi! I am ESP32 on KOLD suit.<br>"\
"My IP: " + WiFi.localIP().toString() + "<br>"\
"Go to <a href='http://' + WiFi.localIP().toString() + '/update'>http://' + WiFi.localIP().toString() +
"/update</a> for OTA update"
);
});
server.begin();
ElegantOTA.setAutoReboot(true);
ElegantOTA.begin(&server); // Start ElegantOTA
// ElegantOTA callbacks
ElegantOTA.onStart(onOTAStart);
ElegantOTA.onProgress(onOTAProgress);
ElegantOTA.onEnd(onOTAEnd);
Serial.println("Запустил HTTP сервер для прошивки. Ссылка: ");
Serial.print(WiFi.localIP());
Serial.println("/update");

// Подключение к Брокеру:
mqtt.setServer(MQTT_BROKER, MQTT_PORT);
mqtt.setCallback(MQTT_ReceiverCallback); // Сохраняем полученные данные в received (callback)
randomSeed(esp_random());
MY_MQTT_ID += String(random(0xffff), HEX);
Serial.print("\nПодключение к брокеру (серверу) под ID = " + MY_MQTT_ID);
mqtt.connect(MY_MQTT_ID.c_str());
mqtt.connected() ? true : MQTT_reconnect(); // 1 = connected <> 0 = Not Connected
#if (FW_TYPE == 'L1')
  mqtt.subscribe("L1_servo");
#elif (FW_TYPE == 'L2')
  mqtt.subscribe("L2_servo");
#elif (FW_TYPE == 'R1')
  mqtt.subscribe("R1_servo");
#elif (FW_TYPE == 'R2')
  mqtt.subscribe("R2_servo");
#endif
Serial.println("\n>> Подключено!");
mqtt.publish("status", "Устройство #1\n>> Подключено!");
vTaskDelay(2000);

//Подготовка Сервоприводов:
#if USE_SERVOS
  servo[0].attach(2);
  servo[1].attach(4);
  #if (FW_TYPE == 'L2' || FW_TYPE == 'R2')
    servo[2].attach(12);
  #endif
#endif

//Подготовка потенциометров:
analogReadResolution(9); // Установка разрешения аналоговых значений: 2^9 -> 512
pinMode(32, INPUT);
NC[0].calcNoiseIntervals(32, NC_CALC_NOISE_INT_TIME);
NC[0].setNoiseIntervalsCoef(NC_NOISE_INTERVALS_COEF);

```

```

pinMode(33, INPUT);
NC[1].calcNoiseIntervals(33, NC_CALC_NOISE_INT_TIME);
NC[1].setNoiseIntervalsCoef(NC_NOISE_INTERVALS_COEF);
#if (FW_TYPE == 'L2' || FW_TYPE == 'R2')
pinMode(34, INPUT);
NC[2].calcNoiseIntervals(34, NC_CALC_NOISE_INT_TIME);
NC[2].setNoiseIntervalsCoef(NC_NOISE_INTERVALS_COEF);
#endif
Serial.println("Calibration successful!!");

// Создание задачи на ядре 0:
xTaskCreatePinnedToCore(
  SA_task, // Function of task
  "Задача на ядре #0 [Работа с сенсорами и актуаторами]", // Task name
  70000, // Heap for task
  NULL,
  1, // Priority
  &SA_taskHandler, // Task handler object
  0 // Core where task runs
);
Serial.println("Created task on Core#0");
}

// Задача на ядре Core#1, которая отвечает за передачу и получение данных по протоколу MQTT:
void loop() {

  // Проверка подключения к Брокеру:
  mqtt.loop();
  mqtt.connected() ? true : MQTT_reconnect(); // 1 = connected <> 0 = Not Connected

  // Обработчик для OTA:
  ElegantOTA.loop();

  // Измерение задержки текущей задачи:
  #if 0 // 1 - включено; 0 - выключено
    static int loop_micros;
    Serial.printf("NETWORKING Task delay: %i\n", micros() - loop_micros); //point B2
    loop_micros = micros(); //point B1
  #endif

  // Отправка данных с TXdata
  #if (FW_TYPE == 'L1')
    mqtt.publish("L1_angles", TXdata.c_str());
  #elif (FW_TYPE == 'L2')
    mqtt.publish("L2_angles", TXdata.c_str());
  #elif (FW_TYPE == 'R1')
    mqtt.publish("R1_angles", TXdata.c_str());
  #elif (FW_TYPE == 'R2')
    mqtt.publish("R2_angles", TXdata.c_str());
  #endif

  // Special functions to give way to other funs:
  vTaskDelay(1);
  // taskYIELD(); //DANGER, use on ur own resp-ty

```

```

}

// Задача на ядре Core#0, которая отвечает за работу с сенсорами и актуаторами:
void SA_task(void * parameter){
    //vTaskDelay(2000);
    #if (FW_TYPE == 'L2' || FW_TYPE == 'R2')
        static int analogRead_data[3] = {0};
    #else
        static int analogRead_data[2] = {0};
    #endif
    for(;;){
        // Serial.println("Core#0 Task in progress"); // DEBUG CODE. Leave it commented

        // Измерение задержки текущей задачи:
        #if 0 // 1 - включено; 0 - выключено
            static int core0_task_micros;
            Serial.printf("SENS_ACTUATOR Task delay: %i\n", micros() - core0_task_micros); //point B2
            core0_task_micros = micros(); //point B1
        #endif

        // Переменная/буфер для сбора данных, для отправки
        TXbuff = "";

        // Считывание аналоговых данных с потенциометров на сервоприводах:
        analogRead_data[0] = analogRead(32);
        analogRead_data[1] = analogRead(33);
        #if (FW_TYPE == 'L2' || FW_TYPE == 'R2')
            analogRead_data[2] = analogRead(34);
        #endif

        // Сглаживание перед ... (Метод среднего скользящего):
        #if PON_MA_BEFORE == 1
            analogRead_data[0] = MA_BEFORE[0].getMovAvgFILO(analogRead_data[0]);
            analogRead_data[1] = MA_BEFORE[1].getMovAvgFILO(analogRead_data[1]);
            #if (FW_TYPE == 'L2' || FW_TYPE == 'R2')
                analogRead_data[2] = MA_BEFORE[2].getMovAvgFILO(analogRead_data[2]);
            #endif
        #endif

        // Адаптивный фильтр / корректор помех:
        #if PON_NC == 1
            analogRead_data[0] = NC[0].calcValue(analogRead_data[0]);
            analogRead_data[1] = NC[1].calcValue(analogRead_data[1]);
            #if (FW_TYPE == 'L2' || FW_TYPE == 'R2')
                analogRead_data[2] = NC[2].calcValue(analogRead_data[2]);
            #endif
        #endif

        // Сглаживание после ... (Метод среднего скользящего):
        #if PON_MA_AFTER == 1
            analogRead_data[0] = MA_AFTER[0].getMovAvgFILO(analogRead_data[0]);
            analogRead_data[1] = MA_AFTER[1].getMovAvgFILO(analogRead_data[1]);
            #if (FW_TYPE == 'L2' || FW_TYPE == 'R2')
                analogRead_data[2] = MA_AFTER[2].getMovAvgFILO(analogRead_data[2]);
            #endif
        #endif
    }
}

```

```

#endif

// Присвоение обработанных данных в буфер для отправки:
TXbuff += analogRead_data[0] + FILLER;
TXbuff += analogRead_data[1] + FILLER;
#if (FW_TYPE == 'L2' || FW_TYPE == 'R2')
    TXbuff += analogRead_data[2] + FILLER;
#endif
TXbuff += "#"; // u 1-2 mcs
TXdata = TXbuff;

// //Ограничение по скорости:
// if (received != 0){
//     if (received[0] < 80){
//         received[0] = 80;
//     }
//     else if (100 < received[0]){
//         received[0] = 100;
//     }
//     if (angle_limits[0][0] < analogRead_data[i][0] && analogRead_data[i][0] < angle_limits[0][1]){
//         servo[0].write(received[0]);
//     }
// }

// }

// //Ограничение по углу:
// if (received != 0){
//     #if (FW_TYPE == 'L1')
//     #define N 2 //
//     #define R 0 //R - позиция данных в массиве; R=[0; 2; 5; 7]
//     #elif (FW_TYPE == 'L2')
//     #define N 3
//     #define R 2
//     #elif (FW_TYPE == 'R1')
//     #define N 2
//     #define R 5
//     #elif (FW_TYPE == 'R2')
//     #define N 3
//     #define R 7
//     #endif
//     for(int n=0; n<N; n++){
//         if (send[n] < angle_limits[R+n][0]){
//             #if (FORCE_MOVE_TO_LIMIT == false)
//                 servo[n].write(servo_null[R+n]); //остановка сервопривода если потенциометр вышел за
нижнюю границу
//             #elif (FORCE_MOVE_TO_LIMIT == true)
//                 servo[n].write(servo_null[R+n]+10); //
//             #endif
//         }
//         else if (angle_limits[R+n][1] < send[n]){
//             #if (FORCE_MOVE_TO_LIMIT == false)
//                 servo[n].write(servo_null[R+n]); //так же как описано выше
//             #elif (FORCE_MOVE_TO_LIMIT == true)
//                 servo[n].write(servo_null[R+n]-10); //так же как описано выше

```

```

// #endif
// }
// else {
//   servo[n].write(received[n]);
// }
// }
// }

// Special functions to give way to other funs:
vTaskDelay(1);
// taskYIELD(); //DANGER, use on ur own resp-ty
}
}

// Callback ф-я для перевода полученных данных в переменную received:
void MQTT_ReceiverCallback(char* topic, uint8_t* payload, size_t plength) {
//   Serial.print("Message arrived [");
//   Serial.print(topic);
//   Serial.print("] ");
//   for (size_t i = 0; i < plength; i++) {
//     Serial.print((char)payload[i]);
//   }
//   Serial.println();

  received = payload;

  // Switch on the LED if an 1 was received as first character
  // if ((char)payload[0] == '1') {
  //   digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level
  //                                   // but actually the LED is on; this is because
  //                                   // it is active low on the ESP-01)
  // } else {
  //   digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH
  // }
}

// Функция для проверки подключения к сети. В ней имеются вызовы для переподключения:
bool MQTT_reconnect() {

  int8_t ret;
  uint8_t retries = 25;
  while ( mqtt.connected() != true ) { // successful connection will return 1
    // Seems [while check] is not working correctly ...

    // Disconnect:
    if (!mqtt.connected()){
      Serial.println("\nDisconnecting from Broker");
      mqtt.disconnect();
    }

    // Connect again:
    mqtt.connect(MY_MQTT_ID.c_str());
    retries--;
    Serial.print("\n[MQTT Error]\tRetries = " + String(retries) + "/25.");

    int cli_state = mqtt.state();

```

```

String cli_state_str;
switch(cli_state){
case -4:
cli_state_str = "The network connection timed out or server didn't respond within the keepalive time.";
break;
case -3:
cli_state_str = "The network connection was lost/broken.";
break;
case -2:
cli_state_str = "The network connection failed.";
break;
case -1:
cli_state_str = "The client is disconnected cleanly.";
break;
case 0:
cli_state_str = "The client is connected.";
Serial.print("mqtt.connected() = ");
Serial.println(mqtt.connected());
return true;
// The problem with the connected state of mqtt... mqtt.loop() is useless. Helps only return true
// Problem with [while] ...
break;
case 1:
cli_state_str = "The server does not support the requested MQTT version.";
break;
case 2:
cli_state_str = "The server rejected the client identifier.";
break;
case 3:
cli_state_str = "The server was unable to accept the connection.";
break;
case 4:
cli_state_str = "The username or password is not valid.";
break;
case 5:
cli_state_str = "The client is not authorized to connect to the server.";
break;
default:
cli_state_str = "UNKNOWN ERROR.";
};
Serial.print("\tClient state: " +String(cli_state)+ ". " +cli_state_str); //WARN!!!! is mqtt cli id necessary?
vTaskDelay(1000); // restore retries interval: 1 sec
if (retries == 0) {
// Когда не удастся подключиться к MQTT серверу,
// индикатор модуля будет БЕСКОНЕЧНО мигать:
Serial.println("\n Out of retries ... ");
while (1) {
Serial.println("[ MQTT Client :: Out of retries... Fall in while(1)]\tClient state: " +String(cli_state)+ ". "
+cli_state_str);
digitalWrite(LED_BUILTIN, HIGH);
vTaskDelay(1000);
digitalWrite(LED_BUILTIN, LOW);
vTaskDelay(1000);
}
}
}
}

```

```
mqtt.publish("status", "I'm back!!");
}
```

Листинг. Скетч-код для плат с ESP32 текущей версии перчатки КТП

```
/*
© Copyright
Баходиржон Анваржонов (anvarjonovb@gmail.com)
*/

#define APP_VER "exoTI-Glove v3.0b"

// Pinout:
// [k finger] -> [GPIO]
// 1st finger -> 17
// 2nd finger -> 18
// 3rd finger -> 21
// 4th finger -> 22
// 5th finger -> 23

//Параметры подключения:
#define FW_TYPE 'Glove' // Выберите блок (L1 - левый ближний, L2 - левый дальний, R1 -
правый ближний, R2 - правый дальний)
#define USE_STATIC_IP true // true = Статическое IP; false = Динамическое IP
#define TEST_MODE true

#if TEST_MODE
const char* WLAN_SSID = "space_lab";
const char* WLAN_PASSWD = "space_508";
const char* MQTT_BROKER = "192.168.31.137";
#else
const char* WLAN_SSID = "MEGANET_6G_LAB";
const char* WLAN_PASSWD = "space_508";
const char* MQTT_BROKER = "192.168.31.205";
#endif

const int MQTT_PORT = 1883;
String MY_MQTT_ID = "ESP32-CLI_";
#if USE_STATIC_IP
int my_ip[4] = {192, 168, 31, 144};
#else
int rtr_gateway[4] = {192, 168, 31, 0};
int rtr_subnet[4] = {255, 255, 255, 0};
int rtr_primaryDNS[4] = {8, 8, 8, 8}; //optional
int rtr_secondaryDNS[4] = {8, 8, 4, 4}; //optional
```

```

#define TX_DATA "glove"
#define TX_ON_REQUIRED 0
#define TX_timer_VAL 1000
#define RX_DATA "hand"
#define CORE0_SPECIAL_DELAY_MCS 0 //Sensors and actuators
#define CORE1_SPECIAL_DELAY_MCS 0 //Data exchange
#define PUBLISH_DELAYS 0
#define DELAYS_DATA "espDelay_glove" //delay of exec loop
#define DELAYS_timer_VAL 20
#define PUBLISH_MPS 0
#define MPS_DATA "MPS_glove" //messages per second
#define MPS_timer_VAL 1000
#define STATUS_DATA "status_glove"
#define CALLIB_DATA "callibData_glove"
#define FILLER 100

// Other params:
#define MUX_A 27
#define MUX_B 14
#define MUX_C 12
#define MUX1_IO 34

#define NC_NOICE_CALLIBRATION 0
#define FLEX_CALLIBRATION 0
#define FLEX_CALLIBRATION_TIME 5000 //0 ... 10000
#define FLEX_PRE_CALLIBRATION_TIME 1000 //do not change this
#define FLEX_CORRECTION 1
#define FLEX_NC 1
#define NC_AMMOUNT_OF_VALUES 3
#define NC_MAX_AMMOUNT_OF_VALUES 3
#define CALC_JITT_INT_TIME 200 // Время для определения границ помех
#define NC_JITTER_INTERVALS_COEF 1.1
#define FLEX_MA_BEFORE 0
#define MA_FILO_BEFORE_CNT 6
#define MA_FILO_BEFORE_LEN 2
#define FLEX_MA_AFTER 0
#define MA_FILO_AFTER_CNT 2
#define MA_FILO_AFTER_LEN 3

// Debug:
#define SHOW_FLEX_LIMITS 1

// #####
// Исполняемый код:

// Подключение библиотек:
#include <PubSubClient.h> // Библиотека для работы с MQTT
#include <esp_random.h> // Нужна для рандомизации ID MQTT клиента

```

```

#include <WiFi.h>           // Библиотека для работы с WiFi
#include <AsyncTCP.h>        // Нужна для работы ElegantOTA
#include <ESPAsyncWebServer.h> // Нужна для работы ElegantOTA
#include <ElegantOTA.h>      // Библиотека для беспроводной прошивки

#include "NoiseCorrector.h"  // Самодельный фильтр - скользящая средняя
#include "MovingAverageFILO.h" // Самодельный фильтр - скользящая средняя

// Создание объекта для новой задачи (FreeRTOS / Многозадачность / Многопоточность):
TaskHandle_t SA_taskHandler;

// Создание объекта для работы с WiFi как клиент:
WiFiClient client;

// Конфигурации WiFi для статического IP:
IPAddress myIP(my_ip[0], my_ip[1], my_ip[2], my_ip[3]);
IPAddress gateway(rtr_gateway[0], rtr_gateway[1], rtr_gateway[2], rtr_gateway[3]);
IPAddress subnet(rtr_subnet[0], rtr_subnet[1], rtr_subnet[2], rtr_subnet[3]);
IPAddress primaryDNS(rtr_primaryDNS[0], rtr_primaryDNS[1], rtr_primaryDNS[2],
rtr_primaryDNS[3]);
IPAddress secondaryDNS(rtr_secondaryDNS[0], rtr_secondaryDNS[1], rtr_secondaryDNS[2],
rtr_secondaryDNS[3]);

// Для беспроводной прошивки:
AsyncWebServer server(80);

// Создание объектов и переменных для работы с MQTT:
PubSubClient mqtt(client);

bool MQTT_reconnect();
void MQTT_ReceiverCallback(char*, uint8_t*, size_t);
String TXbuff, TXdata, STRbuff;

uint32_t BITMAP[10]; //fast mux register move

// Создание объектов самодельного фильтра "Корректора помех" для работы с аналоговыми
сигналами:
NoiseCorrector NC[5] = {NoiseCorrector(NC_AMMOUNT_OF_VALUES,
NC_MAX_AMMOUNT_OF_VALUES)};

// Создание объектов самодельного фильтра "Скользящего среднего" для работы с
аналоговыми сигналами:
MovingAverageFILO MA_BEFORE[5] = {MovingAverageFILO(MA_FILO_BEFORE_CNT,
MA_FILO_BEFORE_LEN)};
MovingAverageFILO MA_AFTER[5] = {MovingAverageFILO(MA_FILO_AFTER_CNT,
MA_FILO_AFTER_LEN)};

```

```

int flexNoiceMM[5][2] = {{180, 187}, {171, 184}, {165, 178}, {177, 190}, {169, 182}};
//int delta_flexNoice[5] = {5, 5, 5, 5, 5}; //delta_flexNoice = flex_noice_max - flex_noice_min;
USED instead of flexNoiceMM[5][2] in the next versions

// f_min f_max s_min s_max
int FS_MM[5][4] = {
  {168, 292, 10, 120}, // 1st finger
  {187, 307, 50, 160}, // 2nd finger
  {187, 311, 50, 160}, // 3rd finger
  {199, 303, 45, 140}, // 4th finger
  {201, 290, 25, 130} // 5th finger
};

// f_min f_max s_min s_max
// int FS_MM[5][4] = {
//   {165, 301, 10, 120}, //17
//   {178, 307, 50, 160}, //18
//   {180, 311, 50, 160}, //21
//   {184, 303, 45, 140}, //22
//   {220, 287, 25, 130} //23
// };

int RXbuff[13] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

#include "OTA_Handlers.h"

// Настройка выводов платы, подготовка подключений, калибровка сенсоров и т.д.:
void setup() {
  // Запуск монитора порта (для отладки/Debug):
  Serial.begin(500000);
  Serial.println();
  Serial.println(APP_VER);

  // Конфигурация и Подключение к WiFi:
  #if USE_STATIC_IP // Конфигурация статического IP:
    if (!WiFi.config(myIP, gateway, subnet, primaryDNS, secondaryDNS)) {
      Serial.println("STA Failed to configure");
    }
  #endif
  WiFi.mode(WIFI_STA);
  WiFi.begin(WLAN_SSID, WLAN_PASSWD);
  Serial.print("\n\nПодключение к WLAN ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    vTaskDelay(500);
  }
  Serial.println("\nПодключено к " + String(WLAN_SSID));
  Serial.print("\nМой IP адрес: ");

```

```

Serial.println(WiFi.localIP());

// Подготовка Сервера для OTA прошивки платы:
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
  request->send(200, "text/html",
    "<!DOCTYPE html>\n"
    "<head><meta name='color-scheme' content='light dark'></head>\n"
    "Hi! I am ESP32 on KOLD suit.\n"
    "<br>My IP: " + WiFi.localIP().toString() +\n"
    "<br>Go to <a href='http://\" + WiFi.localIP().toString() + \"/update' >http://\" +
    WiFi.localIP().toString() + \"/update</a> for OTA update"
    "<br>My FW_name and FW_ver: " APP_VER
  );
});
server.begin();
ElegantOTA.setAutoReboot(true);
ElegantOTA.begin(&server); // Start ElegantOTA
// ElegantOTA callbacks
ElegantOTA.onStart(onOTAStart);
ElegantOTA.onProgress(onOTAProgress);
ElegantOTA.onEnd(onOTAEnd);
Serial.println("Запустил HTTP сервер для прошивки. Ссылка: ");
Serial.print(WiFi.localIP());
Serial.println("/update");

// Подключение к Брокеру:
mqtt.setServer(MQTT_BROKER, MQTT_PORT);
mqtt.setCallback(MQTT_ReceiverCallback); // Сохраняем полученные данные в received
(callback)
randomSeed(esp_random());
MY_MQTT_ID += String(random(0xffff), HEX);
Serial.print("\nПодключение к брокеру (серверу) под ID = " + MY_MQTT_ID);
mqtt.connect(MY_MQTT_ID.c_str());
mqtt.connected() ? true : MQTT_reconnect(); // 1 = connected <> 0 = Not Connected
mqtt.subscribe(RX_DATA);
Serial.println("\n>> Подключено!");
mqtt.publish(STATUS_DATA, "Устройство #glove\n>> Подключено!");
vTaskDelay(2000);

// Подготовка пинов:
analogReadResolution(9); // Установка разрешения аналоговых значений: 2^8 -> 256
pinMode(MUX_A, OUTPUT);
pinMode(MUX_B, OUTPUT);
pinMode(MUX_C, OUTPUT);
pinMode(MUX1_IO, INPUT);
for(int i=0; i<5; i++){
  BITMAP[i]=
    ( i&B0001? ((uint32_t)1 << MUX_A) : 0) |
    ((i&B0010)>>1? ((uint32_t)1 << MUX_B) : 0) |
    ((i&B0100)>>2? ((uint32_t)1 << MUX_C) : 0)
  ;
}

```

```

    BITMAP[i+5]=
      ( i&B0001? 0 : ((uint32_t)1 << MUX_A)) |
      ((i&B0010)>>1? 0 : ((uint32_t)1 << MUX_B)) |
      ((i&B0100)>>2? 0 : ((uint32_t)1 << MUX_C))
    ;
    Serial.println(BITMAP[i], BIN);
    Serial.println(BITMAP[i+5], BIN);
    Serial.println();
  }

  // Определение границ помех:
  String temp_str;
  #if NC_NOISE_CALIBRATION
  String temp = "\n>> Определение помех.\nНе двигайте ваши пальцы
"+String(CALC_JITT_INT_TIME*5/1000+3)+" секунд";
  Serial.println(temp);
  vTaskDelay(2000);
  mqtt.publish(STATUS_DATA, temp.c_str());
  for(int i=0; i<5; i++){
    // digitalWrite(MUX_A, i&B0001);
    // digitalWrite(MUX_B, (i&B0010)>>1);
    // digitalWrite(MUX_C, (i&B0100)>>2);
    REG_WRITE(GPIO_OUT_W1TS_REG, BITMAP[i]);
    REG_WRITE(GPIO_OUT_W1TC_REG, BITMAP[i+5]);
    temp_str += String(
      NC[i].calcNoiseIntervals(MUX1_IO, CALC_JITT_INT_TIME)
    ) + "\t" + String(NC[i].d_point) + "\t" + String(NC[i].u_point) + "\n";
  }
  vTaskDelay(1000);
  Serial.println("\n>> Определение помех завершена!");
  mqtt.publish(STATUS_DATA, "\n>> Определение помех завершена!");
  #else
  mqtt.publish(STATUS_DATA, "Using user defined calibration data");
  for (int i=0; i<5; i++) NC[i].setNoiseIntervals(flexNoiseMM[i][1], flexNoiseMM[i][0]);
  // for (int i=0; i<5; i++) NC[i].setNoiseIntervals(delta_flexNoise[i]); //for next ver

  for(int i=0; i<5; i++) temp_str += String(NC[i].u_point) + "\t" + String(NC[i].d_point) + "\n";
  #endif

  //debug:
  Serial.println("\nКол-во\t| мин.\t| макс. :");
  mqtt.publish(STATUS_DATA, "\nКол-во\t| мин.\t| макс. :");
  Serial.print(temp_str);
  mqtt.publish(STATUS_DATA, temp_str.c_str());

  // Уменьшение\увеличение границ помех (при необходимости):
  for (int i=0; i<5; i++) NC[i].setNoiseIntervalsCoef(NC_JITTER_INTERVALS_COEF);

  // Нахождение оптимальных границ значений изгиба:
  #if FLEX_CALIBRATION

```

```

Serial.println("\n>> Калибровка датчиков изгиба через 3 секунд.");
mqtt.publish(STATUS_DATA, "\n>> Калибровка датчиков изгиба через 3 секунд.");
vTaskDelay(3000);
bool callibrationStartPromt = true;
for (int i = 0; i < 5; i++) {
    FS_MM[i][1] = 0;
    FS_MM[i][0] = 5000;
}
static int FC_timer = millis();
int val;
while (millis() - FC_timer < FLEX_CALIBRATION_TIME) {
    for(int i=0; i<5; i++){
        // digitalWrite(MUX_A, i&B0001);
        // digitalWrite(MUX_B, (i&B0010)>>1);
        // digitalWrite(MUX_C, (i&B0100)>>2);
        REG_WRITE (GPIO_OUT_W1TS_REG, BITMAP[i]);
        REG_WRITE (GPIO_OUT_W1TC_REG, BITMAP[i+5]);
        val = NC[i].calcValue(analogRead(MUX1_IO));
        if (millis() - FC_timer > FLEX_PRE_CALIBRATION_TIME) {
            if (callibrationStartPromt == true){
                Serial.println("\n>> Сжимайте и отпускайте ваши пальцы повторно!\n");
                mqtt.publish(STATUS_DATA, "\n>> Сжимайте и отпускайте ваши пальцы повторно!");
                callibrationStartPromt = false;
            }
            if (val > FS_MM[i][1]) FS_MM[i][1] = val;
            if (val < FS_MM[i][0]) FS_MM[i][0] = val;
        }
        Serial.print(String(val)+"\t");
    }
    Serial.println();
}
Serial.println("\n>> Калибровка завершена!");
mqtt.publish(STATUS_DATA, "\n>> Калибровка завершена!");
#endif

// flexMM[3][0] = 260;
// flexMM[3][1] = 370;

// Показ оптимальных границ значений изгиба:
#if SHOW_FLEX_LIMITS == 1
    vTaskDelay(3000);
    Serial.println("\n#####");
    mqtt.publish(CALLIB_DATA, "\n#####");
    String limits = ">> Границы датчиков изгиба:\n";
    for(int i=0; i<5; i++){
        limits += String(FS_MM[i][0]);
        limits += "\t< F" + String(i) + "< ";
        limits += String(FS_MM[i][1]) + "\n";
    }
    Serial.println(limits);

```

```

mqtt.publish(CALLIB_DATA, limits.c_str());

vTaskDelay(100);
limits = ">> Границы_сервоприводов:\n";
for(int i=0; i<5; i++){
    limits += String(FS_MM[i][2]);
    limits += "\t< S" + String(i) + " < ";
    limits += String(FS_MM[i][3]) + "\n";
}
Serial.println(limits);
mqtt.publish(CALLIB_DATA, limits.c_str());
Serial.println("#####\n");
mqtt.publish(CALLIB_DATA, "#####\n");
#endif

// Создание задачи на ядре 0:
xTaskCreatePinnedToCore(
    SA_task, // Function of task
    "Задача на ядре #0 [Работа с сенсорами и актуаторами]", // Task name
    70000, // Heap for task
    NULL,
    1, // Priority
    &SA_taskHandler, // Task handler object
    0 // Core where task runs
);
Serial.println("Created task on Core#0");
vTaskDelay(10);
Serial.end();
}

void loop() {
    // Проверка подключения к Брокеру:
    mqtt.loop();
    mqtt.connected() ? true : MQTT_reconnect(); // 1 = connected <> 0 = Not Connected

    // Обработчик для OTA:
    ElegantOTA.loop();

    // Измерение задержки текущей задачи:
    #if 0 // 1 - включено; 0 - выключено
        static int loop_micros;
        Serial.printf("NETWORKING Task delay: %i\n", micros() - loop_micros); //point B2
        loop_micros = micros(); //point B1
    #endif

    static int MPS=0;

    //Публикация данных сенсоров на сервер:
    static int TX_timer = micros();

```

```

if(TX_timer_VAL <= micros() - TX_timer){
    TX_timer = micros();

    #if TX_ON_REQUIRED == 1
    static String TXdata_temp;
    if (TXdata != TXdata_temp){
        mqtt.publish(TX_DATA, TXdata.c_str());
        MPS++;
        TXdata_temp = TXdata;
    }
    #else
    if (mqtt.publish(TX_DATA, TXdata.c_str())) {
        MPS++;
    }
    #endif
}

//Дополнительные индикаторы:
#if PUBLISH_MPS
static int MPS_timer = millis();
if(MPS_timer_VAL <= millis() - MPS_timer){
    MPS_timer = millis();
    mqtt.publish(MPS_DATA, MPS);
    MPS=0;
}
#endif

// Special functions to give way to other funs:
vTaskDelay(1);
}

void SA_task(void * parameter){
    //vTaskDelay(2000);
    static int analogRead_data[5] = {0};
    for(;;){
        //Получение данных с сенсоров:

        //Датчик изгиба:
        // Переменная/буфер для сбора данных, для отправки
        TXbuff = "";
        for(int i=0; i<5; i++){

            // Измерение задержки текущей задачи:
            #if 0 // 1 - включено; 0 - выключено
            static int core0_task_micros;
            Serial.printf("SENS_ACTUATOR Task delay: %i\n", micros() - core0_task_micros); //point

```

```

core0_task_micros = micros(); //point B1
#endif

// // Чтение аналоговых данных:

// Смещение и чтение:
// digitalWrite(MUX_A, i&B0001);
// digitalWrite(MUX_B, (i&B0010)>>1);
// digitalWrite(MUX_C, (i&B0100)>>2);
REG_WRITE (GPIO_OUT_W1TS_REG, BITMAP[i]);
REG_WRITE (GPIO_OUT_W1TC_REG, BITMAP[i+5]);
analogRead_data[i] = analogRead(MUX1_IO);

//Предобработка значений для ф-ии map() + Ограничитель по мин\макс значений:
//
#if FLEX_CORRECTION
    if (analogRead_data[i] < FS_MM[i][0]) analogRead_data[i] = FS_MM[i][0];
    if (analogRead_data[i] > FS_MM[i][1]) analogRead_data[i] = FS_MM[i][1];
    // analogRead_data[i]<flexMM[i][0]?
    // (analogRead_data[i] = flexMM[i][0]) :(analogRead_data[i] > flexMM[i][1])?
    // (analogRead_data[i] = flexMM[i][1]) :true;
#endif

// Сглаживание перед вычислением (Метод среднего скользящего):
#if FLEX_MA_BEFORE == 1
    analogRead_data[i] = MA_BEFORE[i].getMovAvgFILO(analogRead_data[i]);
#endif

// Вычислитель (удаление помех):
#if FLEX_NC == 1
    analogRead_data[i] = NC[i].calcValue(analogRead_data[i]);
#endif

// Сглаживание после вычислением (Метод среднего скользящего):
#if FLEX_MA_AFTER == 1
    analogRead_data[i] =MA_AFTER[i].getMovAvgFILO(analogRead_data[i]);
#endif

    TXbuff += map(analogRead_data[i], FS_MM[i][0], FS_MM[i][1], FS_MM[i][2],
FS_MM[i][3]) + FILLER;
}

TXbuff += "#"; // u 1-2 mcs
TXdata = TXbuff;

// Special functions to give way to other funs:

```

```

    vTaskDelay(1);
}
}

// Callback ф-я для перевода полученных данных в переменную received:
void MQTT_ReceiverCallback(char* topic, uint8_t* payload, size_t plength) {
//   Serial.print("Message arrived [");
//   Serial.print(topic);
//   Serial.print("] ");
//   for (size_t i = 0; i < plength; i++) {
//       Serial.print((char)payload[i]);
//   }
//   Serial.println();

    STRbuff = (char *) payload;
    // Serial.println(STRbuff);
    for(int i=0; i<13; i++){
        RXbuff[i] = STRbuff.substring(3*i, 3*i+3).toInt()-FILLER;
    }

    // Switch on the LED if an 1 was received as first character
    // if ((char)payload[0] == '1') {
    //     digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level
    //                                     // but actually the LED is on; this is because
    //                                     // it is active low on the ESP-01)
    // } else {
    //     digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH
    // }
}

// Функция для проверки подключения к сети. В ней имеются вызовы для переподключения:
bool MQTT_reconnect() {

    int8_t ret;
    uint8_t retries = 25;
    while ( mqtt.connected() != true ) { // successful connection will return 1
        // Seems [while check] is not working correctly ...

        // Disconnect:
        if (!mqtt.connected()){
            Serial.println("\nDisconnecting from Broker");
            mqtt.disconnect();
        }

        // Connect again:
        mqtt.connect(MY_MQTT_ID.c_str());
        retries--;
        Serial.print("\n[MQTT Error]\tRetries = " +String(retries)+ "/25.");
    }
}

```

```

int cli_state = mqtt.state();
String cli_state_str;
switch(cli_state){
    case -4:
        cli_state_str = "The network connection timed out or server didn't respond within the keepalive
time.";
        break;
    case -3:
        cli_state_str = "The network connection was lost/broken.";
        break;
    case -2:
        cli_state_str = "The network connection failed.";
        break;
    case -1:
        cli_state_str = "The client is disconnected cleanly.";
        break;
    case 0:
        cli_state_str = "The client is connected.";
        Serial.print("mqtt.connected() = ");
        Serial.println(mqtt.connected());
        return true;
        // The problem with the connected state of mqtt... mqtt.loop() is useless. Helps only return true
        // Problem with [while] ...
        break;
    case 1:
        cli_state_str = "The server does not support the requested MQTT version.";
        break;
    case 2:
        cli_state_str = "The server rejected the client identifier.";
        break;
    case 3:
        cli_state_str = "The server was unable to accept the connection.";
        break;
    case 4:
        cli_state_str = "The username or password is not valid.";
        break;
    case 5:
        cli_state_str = "The client is not authorized to connect to the server.";
        break;
    default:
        cli_state_str = "UNKNOWN ERROR.";
};
Serial.print("\tClient state: " +String(cli_state)+ ". " +cli_state_str); //WARN!!!! is mqtt cli id
necessary?
vTaskDelay(1000); // restore retries interval: 1 sec
if (retries == 0) {
    // Когда не удастся подключиться к MQTT серверу,
    // индикатор модуля будет БЕСКОНЕЧНО мигать:
    Serial.println("\n Out of retries ... ");
    while (1) {
        Serial.println("[ MQTT Client :: Out of retries... Fall in while(1)]\tClient state: "

```

```

+String(cli_state)+ ". " +cli_state_str);
    digitalWrite(LED_BUILTIN, HIGH);
    vTaskDelay(1000);
    digitalWrite(LED_BUILTIN, LOW);
    vTaskDelay(1000);
}
}
}
}
mqtt.publish("status", "I'm back!!");
}

```

Листинг. Библиотека с самодельным фильтром скользящего среднего.

```

/// Moving Average FILO

/*
© Copyright
Баходиржон Анваржонов (anvarjonovb@gmail.com)
*/

//FILO defs:
#ifndef MAX_STACK_COUNT
#define MAX_STACK_COUNT 6
#endif
#ifndef MAX_STACK_LEN
#define MAX_STACK_LEN 6
#endif

class MovingAverageFILO {

    //Базовые элементы класса и переменные:
public:
    //FILO vars:
    int stack_count, stack_len;
    uint16_t FILO[MAX_STACK_COUNT][MAX_STACK_LEN];

    //Constructor:
    MovingAverageFILO (){} //Needed to initialize
    MovingAverageFILO (int FILO_CNT, int FILO_LEN){
        stack_count = FILO_CNT;
        stack_len = FILO_LEN;
    }

    //Main:

    int getMovAvgFILO (uint16_t input){
        //Base:
        for(int j=stack_len-1; j>0; j--){
            FILO[0][j] = FILO[0][j-1];
        }
        FILO[0][0] = input;
    }
}

```

```

//Middle:
for(int k=1; k<stack_count; k++){
    for(int j=stack_len-1; j>0; j--){
        FILO[k][j] = FILO[k][j-1];

        for(int m=0; m<stack_len-1; m++){
            FILO[k-1][stack_len-1] += FILO[k-1][m];
            FILO[k][0] = FILO[k-1][stack_len-1]/stack_len;
        }

//End:
for(int n=0; n<stack_len-1; n++){
    FILO[stack_count-1][stack_len-1] += FILO[stack_count-1][n];

//Result:
return FILO[stack_count-1][stack_len-1]/stack_len;
}

};

/*
Usage example:

#define MAX_STACK_COUNT 6
#define MAX_STACK_LEN 6
#include "MovingAverageFILO.h"
MovingAverageFILO MA(stack_count, stack_len);
//or:
//MovingAverageFILO MA(MAX_STACK_COUNT, MAX_STACK_LEN);

int raw, out;
void setup(){
    Serial.begin(500000);
    pinMode(PIN, INPUT);
}
void loop(){
    raw = analogRead(PIN);
    out = MA.getMovAvgFILO(raw);
    Serial.printf("In: %i\tOut: %i", raw, out);
}
*/

```

Листинг. Библиотека с разработанным адаптивным фильтром корректировки аналоговых сигналов

```

///Адаптивный фильтр

/*
© Copyright
Баходиржон Анваржонов (anvarjonovb@gmail.com)
*/

```

```

//CALC defs:
#ifndef BUFF_MAX_LEN
#define BUFF_MAX_LEN 6
#endif
//Общий класс:
class NoiseCorrector {

public:
    // NoiseCorrector (); //Needed to initialize
    NoiseCorrector () {}; //Needed to initialize
    //Базовые элементы класса и переменные:
    int amountOfValues = 3,
        MaxAmountOfValues = 3,
        u_c = 0,
        d_c = 0,
        u_count = 0,
        d_count = 0,
        u_point = 0,
        d_point = 5000,
        d_set[BUFF_MAX_LEN],
        u_set[BUFF_MAX_LEN],
        m_point;
    bool u_interval_counter = false,
        d_interval_counter = false;

    //Constructor:
    NoiseCorrector (int amountOfValues, int MaxAmountOfValues){
        this-> amountOfValues = amountOfValues;
        this-> MaxAmountOfValues = MaxAmountOfValues;
    }

    //Main:

    // Получение данных / проверка параметров для фильтра:
    int* getParams(){
        int i[2] = {this-> amountOfValues, this-> MaxAmountOfValues};
        return i;
    }

    // Установка новых параметров для фильтра:
    int setNewParams(int amountOfValues, int MaxAmountOfValues){
        this-> amountOfValues = amountOfValues;
        this-> MaxAmountOfValues = MaxAmountOfValues;
    }

    // Определение границ помех:
    int calcNoiseIntervals(int pin, unsigned int duration){
        int FLC_timer = millis();
        int count = 0;
        int input;
        while(millis()-FLC_timer <= duration){
            input = analogRead(pin);

```

```

        if(u_point < input) u_point = input;
        if(input < d_point) d_point = input;
        Serial.println(String(input) + "\t" + String(u_point) + "\t" + String(d_point) + "\t");
        count++;
    }
    return count;
}

// При необходимости: Увеличение/уменьшение границ помех
void setNoiseIntervalsCoef(float coef){ //ещё не тестировался!!!
    //float one_percent = (float)/100;
    m_point = d_point + (u_point-d_point)/2;
    u_point = m_point + coef/2 * (u_point-d_point);
    d_point = m_point - coef/2 * (u_point-d_point);
}

// При необходимости: Пользовательская установка границ
void setNoiseIntervals(int new_u_point, int new_d_point){
    u_point = new_u_point;
    d_point = new_d_point;
}

// Вычисление примерного значения сигнала
// и удержание его в некоем интервале
// используя введенные параметры фильтра:
int calcValue(int input){ //логика требует оптимизации!!!) Необходимо оптимизировать эту ф-ю.
    if (input > u_point) {
        u_interval_counter = true;
        u_set[u_count++] = input;
        if (u_count == amountOfValues){ //speed and resource test!!!
            u_count = 0;
            u_c = 0;
            d_count = 0;
            d_c = 0;
            u_interval_counter = false;
            int temp = u_point;
            u_point = u_set[0];
            for (int j=1; j<amountOfValues; j++) u_point += u_set[j];
            u_point = u_point/amountOfValues; //вместо этого попробуем следующие
            // u_point += m_point;
            // u_point = u_point/(amountOfValues+1);
            // Не заработал...

            d_point = d_point + u_point - temp;
            m_point = d_point + (u_point-d_point)/2;
        }
    }
    else if(u_interval_counter == true){
        u_c++;
        if (u_c == MaxAmountOfValues){
            u_interval_counter = false;
            u_c = 0;
            u_count = 0;
        }
    }
}

```

две отдельные коды:

```

    }

    if (input < d_point) {
        d_interval_counter = true;
        d_set[d_count++] = input;
        if (d_count == amountOfValues) { //speed and resource test!!!
            d_count = 0;
            d_c = 0;
            u_count = 0;
            u_c = 0;
            d_interval_counter = false;
            int temp = d_point;
            d_point = d_set[0];
            for (int j=1; j<amountOfValues; j++) d_point += d_set[j];
            d_point = d_point/amountOfValues; //вместо этого попробуем следующие

```

две отдельные коды:

```

                                // d_point += m_point;
                                // d_point = d_point/(amountOfValues+1);
                                // Не заработал...

                                u_point = u_point + d_point - temp;
                                m_point = u_point + (d_point-u_point)/2;
                                }
                                }
                                else if(d_interval_counter == true){
                                    d_c++;
                                    if (d_c == MaxAmountOfValues){
                                        d_interval_counter = false;
                                        d_c = 0;
                                        d_count = 0;
                                    }
                                }
                                return m_point;
                            }
    };

    /*
    ///Usage example:

    #define BUFF_SET_LEN 6
    #include "JitterCorrector.h"
    JitterCorrector JC(value_amount, max_amount_of_values);
    //or:
    ///JitterCorrector JC(___, ___);

    int raw, out;
    void setup(){
        Serial.begin(500000);
        pinMode(PIN, INPUT);
        JC.calcJitterIntervals(PIN, 100); //Calculate signal noise in 100 milliseconds
    }
    void loop(){
        raw = analogRead(PIN);
        out = JC.calcValue(raw);
        Serial.printf("In: %i\tOut: %i\n", raw, out);

```

```

}

///Share [u_point d_point] btw objects example:

int raw, out;
void setup(){
  Serial.begin(500000);
  pinMode(PIN, INPUT);
  JC.calcJitterIntervals(PIN, 100); //Calculate signal noise in 100 milliseconds
}
void loop(){
  raw = analogRead(PIN);
  out = JC.calcValue(raw);
  Serial.printf("In: %i\tOut: %i\n", raw, out);
}
*/

```

**Листинг. Библиотека для прошивки используемых плат по воздуху
(через локальную сеть)**

```

#include <Arduino.h>

void onOTAStart() {
  // Log when OTA has started
  Serial.println("[Elegant OTA]\tOTA update started!");
  // <Add your own code here>

  //list necc handlers to susp them || find solution to solve problem after suspending them))
  // u may just stop them just calling necc funs
  vTaskSuspend(SA_taskHandler);
  vTaskSuspend(xTaskGetHandle("loopTask"));
}

void onOTAProgress(size_t current, size_t final) {
  // Log every 1 second
  static unsigned long ota_progress_millis;
  ota_progress_millis = 0;
  if (millis() - ota_progress_millis > 1000) {
    ota_progress_millis = millis();
    Serial.printf("[Elegant OTA]\tOTA Progress Current: %u bytes, Final: %u bytes\n", current, final);
  }
}

TaskHandle_t RESTART_taskHandler;
void RESTART_task(void * parameter){
  vTaskDelay(10);
  ESP.restart();
}

```

```

void onOTAEnd(bool success) {
  // Log when OTA has finished
  if (success) {
    Serial.println("[Elegant OTA]\tOTA update finished successfully!");
  } else {
    Serial.println("[Elegant OTA]\tThere was an error during OTA update!");
  }
  // <Add your own code here>
  Serial.println("[Elegant OTA]\tRestarting ...");
  for (int i=3; i>=0; i--){
    Serial.printf(" %i", i);
    vTaskDelay(1000);
  }

  xTaskCreatePinnedToCore(
    RESTART_task, // Function of task
    "RESTART_task", // Task name
    10000, // Heap for task
    NULL,
    1, // Priority
    &RESTART_taskHandler, // Task handler object
    0 // Core where task runs
  );
}

```

ПРИЛОЖЕНИЕ В - Акты о внедрении результатов исследования

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Юридический адрес: набережная реки Мойки,
д. 61, литера А, Санкт-Петербург, 191186

Почтовый адрес: пр. Большевиков, д. 22, корп. 1,
Санкт-Петербург, 193232

Тел.(812) 3263156, Факс: (812) 3263159

<http://sut.ru>

E-mail: rector@sut.ru

ОКПО 01179934 ОГРН 1027809197635

ИНН 7808004760 КПП 784001001

ОКТМО 40909000

14.09.2026 № 129/54-99
на № _____ от _____

УТВЕРЖДАЮ

Проректор по научной работе,
доктор техн.наук, доцент



Рабин
Алексей Владимирович

Акт

о внедрении научных результатов, полученных Анваржоновым Баходиржоном Нодирбек угли в диссертационной работе «Исследование и разработка моделей и методов передачи данных для костюма телеприсутствия в сетях связи пятого и последующих поколений»

Комиссия в составе и.о. декана факультета Инфокоммуникационных сетей и систем Г.А. Фокина, доцента кафедры сетей связи и передачи данных Р.А. Дунайцева и заведующей лабораторией кафедры сетей связи и передачи данных О.И. Ворожейкиной составила настоящий акт в том, что научные результаты, полученные Анваржоновым Баходиржоном Нодирбек угли, использованы:

1. При чтении лекций и проведении практических занятий для бакалавров по дисциплине «Интернет вещей и самоорганизующиеся сети» (Рабочая Программа регистрационный номер №_23.05/105-Д), разделы Программы:
 - Концепции Интернета Вещей, Тактильного Интернета и Интернета Навыков;
 - Сети связи 2030;
2. При чтении лекций и проведении практических занятий для магистров по дисциплине «Масштабные M2M коммуникации» (Рабочая Программа регистрационный номер № 25.05/116-Д), раздел Программы:
 - Архитектура сенсорных узлов и сетей;
 - Прикладные протоколы и способы применения M2M технологий.
3. При чтении лекций и проведении практических занятий для аспирантов по дисциплине «Системы, сети и устройства телекоммуникаций» (Рабочая Программа регистрационный номер №_21.05/780-Д), раздел Программы:
 - Современные тенденции развития инфокоммуникационной системы, основные направления научных исследований.

В указанных дисциплинах используются следующие новые научные результаты, полученные Анваржоновым Баходиржоном Нодирбек угли в диссертационной работе:

- Метод адаптивного сжатия и кодирования данных о температуре для костюма телеприсутствия с помощью S-кривой, позволяющий на 20 % сократить разрядность двоичного представления сигнала, что также дает возможность уменьшить объем трафика, поступающего в сеть связи от подсистемы данных о температуре костюма телеприсутствия, на 20%.

- Адаптивный метод передачи данных о температуре костюма телеприсутствия, позволяющий получить выигрыш в объеме передаваемых данных до 30% за счет использования вероятностных свойств сигналов, величина выигрыша зависит от числа термодатчиков и степени разброса значений температуры.

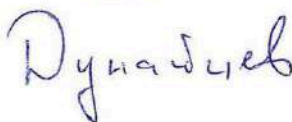
- Модель передачи иммерсивных ощущений в костюме телеприсутствия, включающая три типа обратной связи: температурную тактильную и кинестетическую и подсистема модельной сети для услуг телеприсутствия, в которых предусмотрена возможность индивидуальной калибровки интенсивности каждого типа воздействия, что позволяет обеспечить адекватное восприятие виртуальной и физической среды.

И.о. декана факультета ИКСС
д.т.н., профессор



Г.А. Фокин

Доцент кафедры ССиПД
к.т.н., PhD



Р.А. Дунайцев

Зав. лабораторией кафедры ССиПД



О.И. Ворожейкина

ПРИЛОЖЕНИЕ Г - Сведения о полученных патентах, авторских свидетельствах

РОССИЙСКАЯ ФЕДЕРАЦИЯ

**RU2025610100**

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2025610100

Дата регистрации: 09.01.2025

Номер и дата поступления заявки:
2024691306 10.12.2024

Дата публикации и номер бюллетеня:
10.01.2025 Бюл. № 1

Контактные реквизиты:
нет

Автор(ы):

Волков Артём Николаевич (RU),
Горбачева Любовь Сергеевна (RU),
Каялайнен Валерия Евгеньевна (RU),
Могилатов Владислав Викторович (RU),
Анваржонов Баходиржон Нордибек угли (RU)

Правообладатель(и):

Федеральное государственное бюджетное
образовательное учреждение высшего
образования "Санкт-Петербургский
государственный университет
телекоммуникаций им. проф. М.А.
Бонч-Бруевича" (RU)

Название программы для ЭВМ:

Программное обеспечение модуля обработки кинестетической информации кистей оператора костюма телеприсутствия

Реферат:

Программа предназначена для информационного взаимодействия части костюма телеприсутствия (кистей) с цифровым аватаром с точки зрения эффективности используемых сетевых вычислительных ресурсов бортовой сети костюма телеприсутствия, а также соответствия требованиям услуг телеприсутствия интегрированных сетей 2030 (тактильный интернет). Программа может использоваться для предоставления услуг телеприсутствия в сетях связи пятого и последующих поколений, в том числе приложений метавселенной, например, такой как CityVerse. Функциональные возможности программы: обработка кинестетической информации кистей оператора костюма телеприсутствия, передача полученной кинестетической информации на конечные узлы в сети для информационного взаимодействия с цифровыми аватарами.

Язык программирования: C++/C#

Объем программы для ЭВМ: 23 Кб

РОССИЙСКАЯ ФЕДЕРАЦИЯ

**RU2024691808**

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2024691808

Дата регистрации: 24.12.2024

Номер и дата поступления заявки:
2024691417 10.12.2024

Дата публикации и номер бюллетеня:
24.12.2024 Бюл. № 1

Автор(ы):

Волков Артём Николаевич (RU),
Горбачева Любовь Сергеевна (RU),
Каялайнен Валерия Евгеньевна (RU),
Могилатов Владислав Викторович (RU),
Анваржонов Баходиржон Нордибек угли (UZ)

Правообладатель(и):

Федеральное государственное бюджетное
образовательное учреждение высшего
образования "Санкт-Петербургский
государственный университет
телекоммуникаций им. проф. М.А.
Бонч-Бруевича" (RU)

Название программы для ЭВМ:

Программное обеспечение модуля обработки кинестетической информации верхних конечностей оператора костюма телеприсутствия

Реферат:

Программа предназначена для процесса информационного взаимодействия части костюма телеприсутствия (верхних конечностей) с цифровым аватаром с точки зрения эффективности используемых сетевых вычислительных ресурсов бортовой сети костюма телеприсутствия, а также соответствия требованиям услуг телеприсутствия интегрированных сетей 2030 (тактильный интернет). Программа может использоваться для предоставления услуг телеприсутствия в сетях связи пятого и последующих поколений в том числе приложений метавселенной, например, такой как CityVerse. Функциональные возможности программы: обработка кинестетической информации верхних конечностей оператора костюма телеприсутствия, передача полученной кинестетической информации на конечные узлы в сети для информационного взаимодействия с цифровыми аватарами.

Язык программирования: C++, C#

Объем программы для ЭВМ: 16 КБ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2026687694

**Модуль управления элементами Пельтье для
воспроизведения данных температуры для костюма
телеприсутствия**

Правообладатель: *Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М.А. Бонч-Бруевича» (RU)*

Автор(ы): *Анваржонов Баходиржон Нодирбек угли (UZ)*



Заявка № **2026687564**

Дата поступления **10 сентября 2026 г.**

Дата государственной регистрации

в Ресстре программ для ЭВМ **10 сентября 2026 г.**

*Руководитель Федеральной службы
по интеллектуальной собственности*

документ подписан электронной подписью
Сертификат 00a570e4f7a0c8d531b4b8818675f29506
Владелец **Зубов Юрий Сергеевич**
Действителен с 04.07.2025 по 28.11.2026

Ю.С. Зубов