

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)**

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных технологий

Направление подготовки: «Программная инженерия»
Профиль подготовки: «Разработка программно-информационных систем»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

на тему
“Разработка программно-аппаратного комплекса для мониторинга показателей сердца человека”

Выполнил: студент группы
382008-1 Булгаков Даниил Эдуардович
Подпись

Научный руководитель:
доцент кафедры МОСТ, к.т.н.,
Борисов Николай Анатольевич
Подпись

Нижний Новгород
2024

Содержание

1. Введение	3
2. Описание предметной области	4
2.1 Электрокардиография	4
2.2 Электроды	4
2.3 Отведения	5
2.3.1 Стандартные отведения	6
2.3.2 Усиленные отведения	7
2.3.3 Грудные отведения	8
2.4 Подходы к мониторингу сердечной активности	8
2.5 Вывод	9
3. Разработка проектного решения	10
3.1 Разработка аппаратной части комплекса (Hardware Level)	13
3.1.1 Выбор технологии передачи данных с микроконтроллера на ПК	13
3.1.2 Выбор микроконтроллера	14
3.1.3 Выбор датчика для снятия ЭКГ	14
3.1.4 Тестирование CJMCU-333	15
3.1.5 Тестирование AD8232	17
3.1.6 Разработка модели	18
3.2 Разработка программной части комплекса (Software level)	22
3.2.1 Разработка требований к программной части комплекса (ПЧК)	22
3.2.2 Разработка программной части для модуля сбора данных на базе ESP32	23
3.2.3 Приложение для передачи данных (Application Layer)	26
3.2.4 Сервер для сбора, анализа и хранения данных (Server Layer)	27
3.2.5 База данных (Data Storage Layer)	33
3.2.6 Web Layer (Веб-приложение)	36
4. Заключение	38
5. Список литературы	39
6. Приложение	41

1. Введение

Последние несколько лет наблюдается значительный рост числа заболеваний сердечно-сосудистой системы, что делает наблюдение за состоянием сердца важной задачей в области медицины. Одним из наиболее распространенных методов диагностики и мониторинга сердечной активности является электрокардиография (ЭКГ). ЭКГ представляет собой графическую запись электрической активности сердца, которая позволяет выявлять различные аномалии, такие как аритмии, ишемия, инфаркты и другие патологии. Данный метод широко применяется благодаря своей информативности, неинвазивности и доступности.

Несмотря на высокую эффективность традиционных стационарных систем ЭКГ, их применение ограничено рамками медицинских учреждений. Пациенты, особенно страдающие хроническими заболеваниями, нуждаются в постоянном мониторинге сердечной активности, что представляет собой значительную проблему в условиях стационара. В связи с этим, актуальной становится разработка портативных систем для непрерывного мониторинга сердечной деятельности в повседневной жизни.

Целью данной дипломной работы является создание программно-аппаратного комплекса для мониторинга показателей сердца человека. Комплекс включает модуль для регистрации ЭКГ, приложение для передачи данных с модуля на сервер, а также системы сбора, анализа и хранения данных.

Данная дипломная работа выполнена в форме группового проекта вместе со студентом группы 382006-2 Юнином Даниилом Дмитриевичем.

2. Описание предметной области

2.1. Электрокардиография

Электрокардиография (ЭКГ) — это метод регистрации и анализа электрических полей, возникающих в процессе работы сердца. Этот метод является сравнительно недорогим, но чрезвычайно ценным инструментом для электрофизиологической диагностики в кардиологии.

2.2. Электроды

Электрод — это устройство, предназначенное для проведения электрического тока между телом пациента и электронным прибором. В контексте электрокардиографии (ЭКГ), электроды используются для регистрации электрической активности сердца. Они преобразуют биопотенциалы, возникающие при работе сердца, в электрические сигналы, которые затем обрабатываются и анализируются для диагностики и мониторинга сердечной деятельности.

Электроды играют ключевую роль в обеспечении точности и надежности измерений ЭКГ. Различные типы электродов обладают специфическими характеристиками, которые могут влиять на качество и стабильность записываемых сигналов. Для выбора подходящих типов электродов необходимо основываться на частоте использования, расположения на теле, а также требований к качеству считывания сигнала.

Виды электродов:

1. Гелевые (мокрые) электроды:

Покрываются электропроводящим гелем, который улучшает контакт с кожей и снижает электрическое сопротивление.

2. Текстильные электроды:

Изготавливаются из проводящих текстильных материалов и могут быть интегрированы в одежду.

3. Сухие электроды:

Изготавливаются из проводящих материалов, таких как металлы или углеродные композиции, и не требуют использования геля.

Для наглядного сравнения была добавлена таблица с преимуществами и недостатками каждого типа электродов (Таблица 1.).

Тип электрода	Преимущества	Недостатки
Гелевые (мокрые) электроды	Высокая точность и качество сигнала	Ограниченный срок использования из-за высыхания геля, возможное раздражение кожи
Текстильные электроды	Удобство и гибкость использования	Менее стабильный контакт с кожей, снижение качества сигнала при движении
Сухие электроды	Отсутствие необходимости в геле, сниженный риск раздражения кожи, устойчивость к высыханию, возможность многократного использования	Возможное повышение сопротивления контакта, что может снижать качество сигнала

Таблица 1: Преимущества и недостатки различных типов электродов

В условиях постоянного использования устройства, как в движении, так и в покое, наиболее оптимальным вариантом являются сухие электроды, которые не требуют нанесения геля при каждом использовании и не теряют качество сигнала при физической активности.

2.3. Отведения

Отведения - это способ размещения электродов на теле пациента для получения различных проекций электрической активности сердца. Каждый тип отведения обеспечивает уникальную перспективу наблюдения за электрическими импульсами сердца, что позволяет более детально анализировать работу сердечной мышцы и выявлять различные патологии.

Существует несколько стандартных систем отведений:

- Стандартные отведения
- Усиленные отведения
- Грудные отведения

2.3.1. Стандартные отведения

Стандартные отведения регистрируют разность потенциалов между конечностями человека. Для получения данного типа отведений требуется три электрода: положительный, отрицательный и заземление. Так, правая и левая пара электродов руки образуют первое стандартное отведение - I, электроды правой руки и левой ноги – второе - II, третье отведение III - левая рука и левая нога. Третий электрод используется как заземление (Рис. 1).

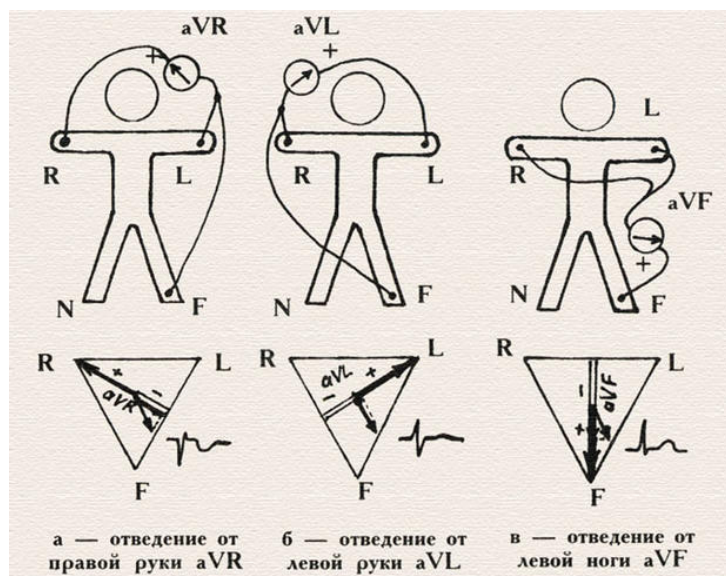


Рис. 1: Стандартные отведения.

Тогда для получения кардиограммы достаточно вычислить разность потенциалов между указанными сигналами.

Отведение	Вычисление
1-ое отведение	LA-RA
2-ое отведение	LL-RA
3-е отведение	LL-LA

LA - левая рука, RA - правая рука, LL - левая нога.

Нетрудно заметить, что в случае, когда нам требуется получить значения сразу по трем отведениям, то аппаратно потребуется считывать только два из них, так как третье можно вычислить путем сложения/вычитания двух других, к примеру:

$$1\text{-ое отведение} + 3\text{-е отведение} = 2\text{-е отведение}$$

Данные отведения позволяют регистрировать следующие типы заболеваний:

- Ишемия миокарда (недостаточное поступление кислорода в сердечную мышцу). Это может проявляться в виде изменений в зубцах ST и T, а также снижения амплитуды зубцов.
- Аритмии, такие как фибрилляция предсердий или желудочковые экстрасистолы. Это может проявляться в виде изменений в ритме и частоте сердечных сокращений, а также в форме зубцов на ЭКГ.
- Блокады проводимости сердца, такие как блокада правой ножки пучка Гиса. Это может проявляться в виде изменений в продолжительности и форме зубцов на ЭКГ.

2.3.2. Усиленные отведения

Усиленные отведения по принципу очень схожи со стандартными отведениями, для них также требуется три электрода. Однако они регистрируют разность потенциалов между одной из конечностей, на которой помещён активный положительный электрод данного отведения и суммарный электродом двух других конечностей. Существуют три таких отведения:

- aVR - усиленное отведение правой руки
- aVL - усиленное отведение левой руки
- aVF - усиленное отведение левой ноги

Для вычисления можно использовать как сигналы с конечностей, так и значения стандартных отведений, используя таблицу.

Отведение	Вычисление	Аналог
aVR	$RA - 0.5 \cdot (LA + LL)$	$-0.5 \cdot (I + II)$
aVL	$LA - 0.5 \cdot (LL + RA)$	$0.5 \cdot (I - III)$
aVF	$LL - 0.5 \cdot (LA + RA)$	$0.5 \cdot (II + III)$

LA - левая рука, RA - правая рука, LL - левая нога

I, II, III - типы стандартных отведений

Данные отведения используются для оценки электрической активности сердца в переднезаднем направлении. Они могут помочь выявить инфаркт миокарда.

2.3.3. Грудные отведения

Грудные отведения регистрируют разницу потенциалов между позитивным электродом, установленным в определённой точке грудной клетки (всего их 6) и единым для остальных пяти электродом Вильсона, потенциал которого равняется нулю (Рис. 2).

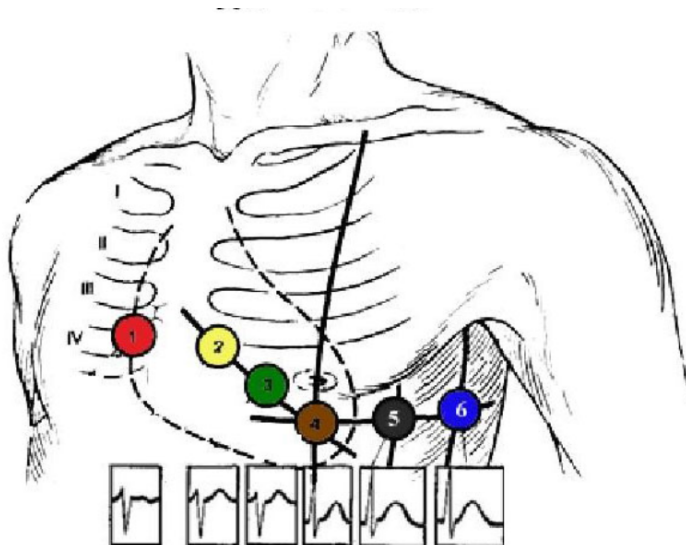


Рис. 2: Грудные отведения.

Данные отведения используются для оценки электрической активности сердца в горизонтальной плоскости. Они могут помочь выявить заболевания миокарда, такие как инфаркт миокарда и аномалии развития сердца.

2.4. Подходы к мониторингу сердечной активности

В настоящее время существуют различные подходы к мониторингу сердечной активности, включая:

1. Традиционные стационарные ЭКГ системы:

Высокая точность и надежность, но требуют нахождения в клинике.

2. Портативные ЭКГ устройства:

Легкие и удобные в использовании, но часто имеют ограниченные возможности по длительности работы и качеству связи.

3. Носимые устройства (например, смарт-часы с функцией ЭКГ):

Удобны для повседневного использования, но часто менее точны и имеют ограничения по функциональности.

Выбор портативных ЭКГ устройств обусловлен стремлением к сочетанию высокой точности измерений с максимальной мобильностью и удобством использования, что в конечном итоге позволит обеспечить непрерывный и эффективный мониторинг сердечной активности пациента в реальном времени для своевременного реагирования на изменения состояния здоровья пациента.

2.5. Вывод

Таким образом, требуется разработать портативный аппаратный комплекс, который требует регистрации двух отведений для покрытия подавляющей области в диагностике сердечных заболеваний. Для снятия электрической активности сердца используется принцип сухих электродов.

3. Разработка проектного решения

Архитектура системы играет ключевую роль в обеспечении надежного и эффективного функционирования программно-аппаратного комплекса для мониторинга показателей сердца. При её проектировании было принято решение использовать принцип многослойной архитектуры. Это решение было обосновано следующими причинами:

1. **Разделение обязанностей:** В многоуровневой архитектуре четко определены роли компонентов, каждый из которых отвечает за определенный функционал.
2. **Повторное использование кода:** Благодаря модульной структуре многоуровневой архитектуры компоненты системы становятся более автономными и могут быть повторно использованы. Это позволяет избежать дублирования кода и обеспечивает более эффективное развитие и поддержку программного продукта.
3. **Гибкость и масштабируемость:** Многоуровневая архитектура обеспечивает гибкость и возможность модификации системы. Новые функциональные возможности могут быть легко добавлены или изменены без воздействия на другие компоненты. Кроме того, такая архитектура хорошо масштабируется при необходимости увеличения производительности или добавления новых узлов.
4. **Улучшенная поддержка и тестирование:** Четкое разграничение компонентов упрощает поддержку и тестирование системы. Каждый уровень может быть протестирован отдельно, что позволяет выявлять и исправлять ошибки на ранних стадиях разработки.

Архитектура проекта разделена на четыре основных слоя (Рис. 3):

1. Веб-приложение.
2. Сервер для сбора, хранения и анализа данных, а также модуль для работы с базой данных.
3. Приложение для передачи данных с ESP32 на сервер.
4. Модуль сбора данных ЭКГ с датчиков, основанный на ESP32.

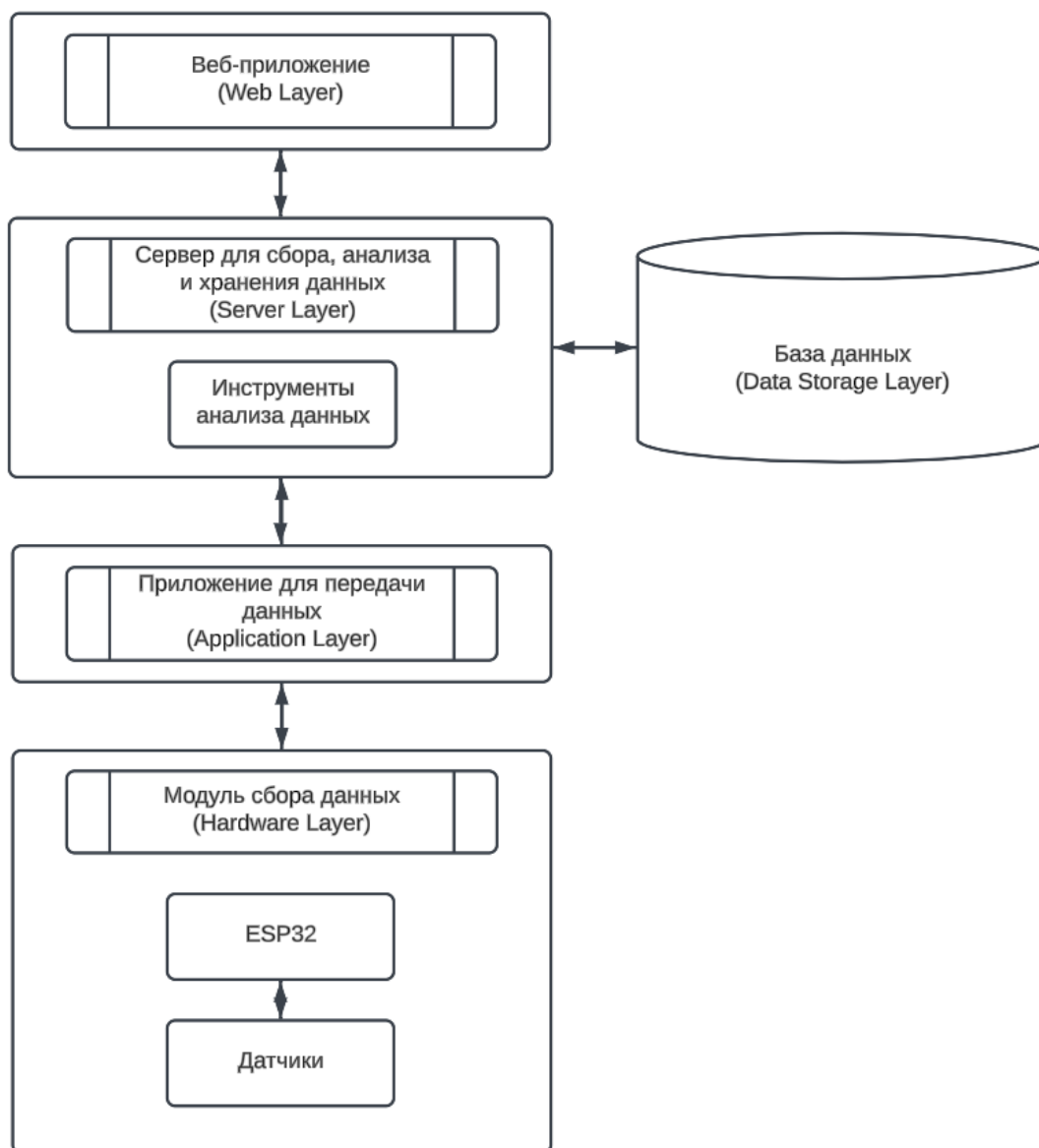


Рис. 3: Архитектура.

Каждый слой, кроме сервера, содержит один модуль, рассмотрим каждый из них:

1. Модуль сбора данных (Hardware Layer):

Он включает в себя аппаратное обеспечение для снятия показаний ЭКГ, такое как микроконтроллер ESP32 и усилитель сигнала AD8232. Микроконтроллер ESP32 отвечает за считывание данных с электродов и преобразование их в цифровой формат, а также передачу в Application Layer. Усилитель сигнала AD8232 усиливает сигнал ЭКГ для более точного считывания.

2. Приложение для передачи данных (Application Layer):

Отвечает за прием, обработку и передачу данных с модуля сбора данных на сервер для

анализа и хранения. В приложении реализована логика работы с данными, включая их форматирование, упаковку и отправку на сервер.

3. Сервер для сбора, анализа и хранения данных (Server Layer):

Этот слой осуществляет прием, анализ и хранение данных, полученных от приложения для передачи данных. В рамках сервера реализованы функциональности по обработке данных, анализу показателей сердечной активности и хранению результатов.

4. База данных (Data Storage Layer):

Обеспечивает долговременное хранение данных и предоставляет интерфейсы для их извлечения и манипуляций. Данные о показателях сердечной активности сохраняются в базе данных для последующего доступа и анализа.

5. Web Layer (Веб-приложение):

Этот слой предоставляет пользовательский интерфейс для взаимодействия с системой через веб-браузер. Пользователи могут просматривать данные, настраивать параметры мониторинга и получать заключения по результатам исследования ЭКГ.

3.1. Разработка аппаратной части комплекса (Hardware Level)

В аппаратной части проекта решающее значение имеет выбор компонентов, которые обеспечивают полноценное функционирование комплекса. Для правильного выбора компонентов требуется учитывать их стоимость, производительность, совместимость и интерфейсы коммуникации.

В этом разделе будет рассмотрен процесс выбора и обоснование использования конкретных компонентов, фокусируясь на микроконтроллере, датчиках для снятия ЭКГ и других важных устройствах для поддержки работоспособности модуля в целом.

3.1.1. Выбор технологии передачи данных с микроконтроллера на ПК

На текущий момент наиболее популярными технологиями являются:

- Wi-Fi
- Bluetooth
- BLE
- ZigBee
- Z-Wave

Так как аппаратный комплекс должен поддерживать мониторинг на протяжении длительного времени, основным критерием отбора является энергопотребление. Высокая скорость передачи данных не требуется, т.к. максимальное количество информации, собранной за день, не может быть выше 35 МБ.

Таким образом, наиболее подходящей технологией является BLE (Bluetooth Low Energy), главным преимуществом которой является крайне низкое потребление энергии. К примеру, средняя потребляемая мощность BLE от 0,01 Вт до 0,5 Вт, что примерно в 10 раз меньше, чем стандартная технология Bluetooth.

3.1.2. Выбор микроконтроллера

При выборе микроконтроллера рассматривались следующие характеристики:

- Размер платы
- Интегрированные модули радиосвязи, а именно BLE
- Наличие I2C и UART интерфейсов
- Энергопотребление
- Вычислительная мощность (кол-во ядер)
- Количество пинов

С учетом всех факторов, были выбраны микроконтроллеры ESP32 и ESP32-3C. Ниже приведены основные сравнительные характеристики.

	ESP32	ESP32-3C
CPU	Xtensa LX6	RISC-V
Кол-во ядер	2	1
Пины GPIO	34	22
Потребление	До 325 мА	До 240 мА
Размеры	31 x 18 x 3.0 мм	24 x 16 x 3.1 мм

По итогу решено использовать микроконтроллер ESP32-3C поскольку он обладает меньшими размерами и более низким энергопотреблением, однако придется столкнуться с проблемами производительности поскольку выбранный микроконтроллер имеет одно ядро и нельзя будет отправлять полученные данные в фоновом режиме на втором ядре.

3.1.3. Выбор датчика для снятия ЭКГ

При выборе модели датчика сердечного ритма были определены следующие критерии, которым он должен соответствовать:

1. Датчик должен поддерживать считывание на частоте в 100 Гц
2. Датчик не должен иметь больших размеров
3. Измерения датчика должны быть разборчивыми
4. Цена датчика должна быть бюджетной

После проведения анализа было решено остановиться на следующих датчиках:

- AD8232
- CJMCU-333

Далее проходил этап их сравнения на тестовом стенде.

Чтобы свести к минимуму влияние внешних факторов на показания датчиков было решено разработать тестирующий стенд. Основным питающим элементом была литиевая батарея, чтобы избежать зашумления сигнала, которое может возникнуть при питании от сети.

Приведем описание для датчиков, которые были главными претендентами на выбор.

3.1.4. Тестирование CJMCU-333

CJMCU333 — маломощный прецизионный инструментальный усилитель. Датчик имеет универсальную конструкцию с тремя операционными усилителями, небольшой размер и малое энергопотребление. Один внешний резистор устанавливает любой коэффициент усиления от 1 до 1000 (Рис. 4).

После калибровки и тестирования, ЭКГ выглядело следующим образом (Рис. 5, 6).

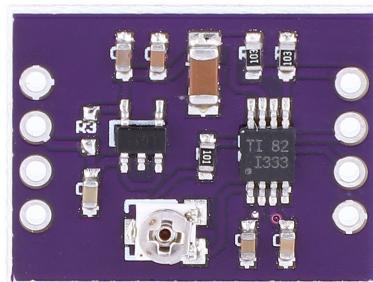


Рис. 4: CJMCU-333



Рис. 5: CJMCU-333 ЭКГ Пример-1.

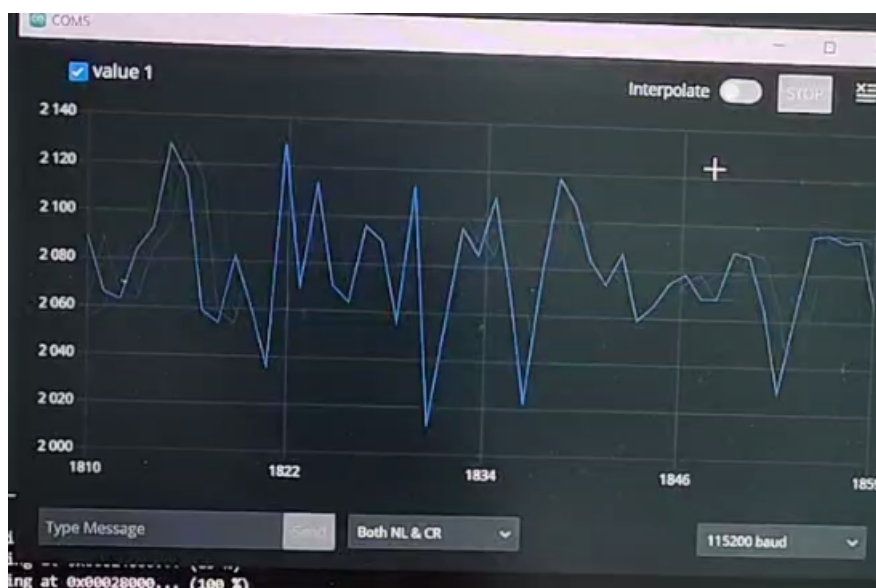


Рис. 6: CJMCU-333 ЭКГ Пример-2.

После проведения немалого количества трудоемких настроек датчика были результаты, который можно видеть в прикрепленных картинках. По графикам невозможно отследить периодичность. Кривая имеет большое количество артефактов и шума. Разобрать данную ЭКГ-кардиограмму не представляется возможным. Результаты тестирования были неудовлетворительными, поэтому данный датчик применяться в проекте не будет.

3.1.5. Тестирование AD8232

AD8232 — это интегрированный блок формирования сигнала для ЭКГ и других приложений измерения биопотенциала. Он предназначен для извлечения, усиления и фильтрации слабых сигналов биопотенциала в условиях шумов, например, создаваемых движением или удаленным размещением электродов (Рис. 7).

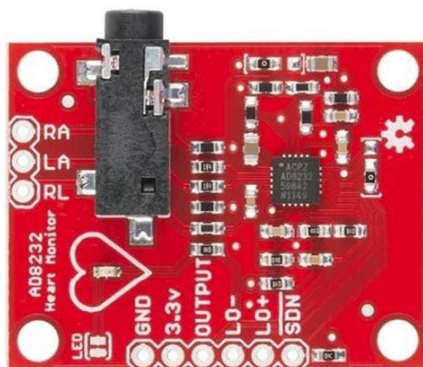


Рис. 7: AD8232

При тестировании график ЭКГ выглядел следующим образом (Рис. 8).



Рис. 8: График AD8232 1-ое отведение.

По результатам первых тестов можно заметить, что показатели получаются довольно

точными и чистыми. Кривая имеет выраженную периодичность, а также точно прослеживаются зубцы. В качестве эксперимента было проведено считывание ЭКГ по второму отведению (Рис. 9).



Рис. 9: График AD8232 2-ое отведение.

А также по значениям первого и второго отведений было вычислено третье отведение с последующим отображением значений в виде графика (Рис. 9).



Рис. 10: График AD8232 3-ое отведение.

Таким образом, из всех протестированных датчиков был выбран именно AD8232, поскольку только с его помощью удалось достичь таких качественных результатов.

3.1.6. Разработка модели

При разработке модуля необходимо учитывать возможность его размещения на спортивной майке, а значит он должен иметь относительно небольшие размеры и маленький вес. Так как модуль необходимо будет заряжать и отправлять с него данные на сервер было принято решение разбить его на два подмодуля. Так, в первом подмодуле будут располагаться электро-

Провода типа SDN отвечают за перевод подключенного к ним датчика в энергосберегающий режим. GND и 3v3 отвечают за питание датчиков.

На выход идет шесть проводов:

- два провода типа LO1
- два провода типа LO2
- OUT1
- OUT2

Провода типа LO передают сигнал о том, что электроды подключены к датчику и с них считывается сигнал. Провод OUT1 отправляет значение ЭКГ по первому типу отведения, OUT2 по второму.

Подмодуль с микроконтроллером

Основная задача данного подмодуля является считывание данных с датчиков ЭКГ, промежуточное хранение и их последующая отправка. Для сохранения большого объема информации на определенное время требуется внешний накопитель. Так как модуль является автономным, для него требуется батарея с возможностью зарядки.

Таким образом, минимальный набор требуемых компонентов выглядит следующим образом:

- ESP32-3C
- Преобразователь напряжение в 3В
- Модуль с SD-картой
- Модуль для зарядного устройства
- Литий-ионная батарея
- Магнитный коннектор

Исходя из этого, была разработана примитивная схема данного подмодуля (Рис. 12).

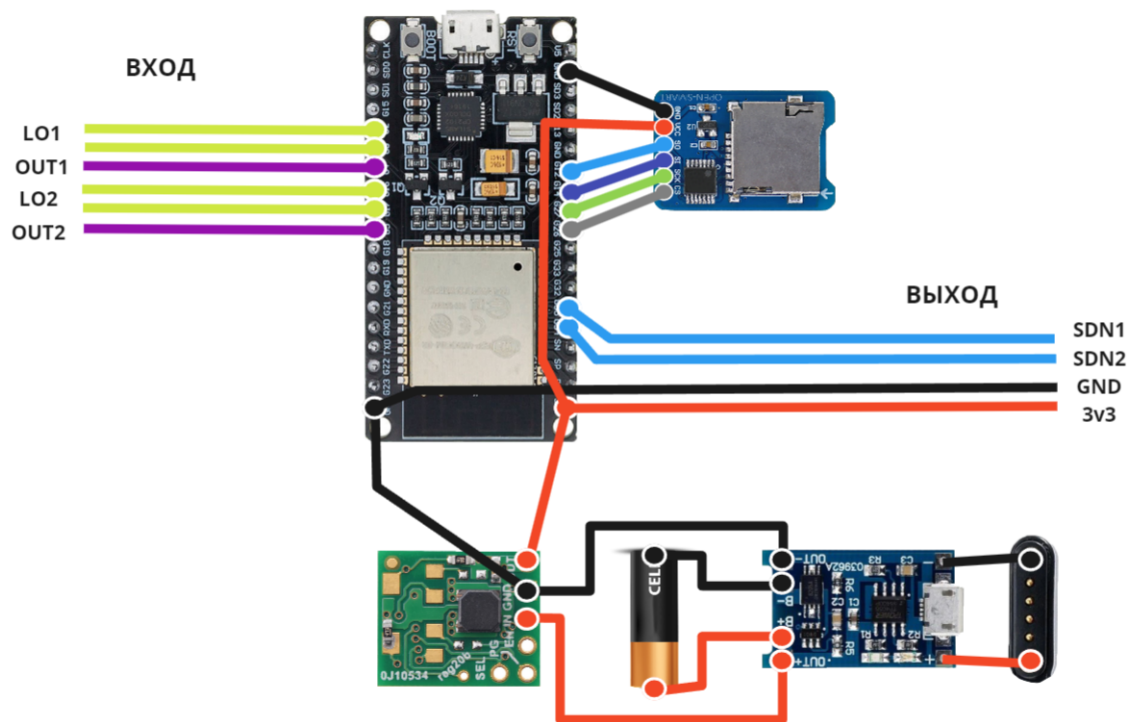


Рис. 12: Подмодуль с микроконтроллером.

Входные и выходные провода соединятся с подмодулем ЭКГ, поэтому ознакомиться с назначением каждого провода можно в вышеупомянутом разделе.

3D модель

Для проектирования первой тестовой модели был использован инструмент Tinkercad из-за простоты использования для создания примитивных объектов. Полученная модель была распечатана на 3D принтере.

3.2. Разработка программной части комплекса (Software level)

В программной части комплекса необходимо было выбрать правильные технологии, инструменты и методы разработки.

В данном разделе будет рассмотрен процесс выбора технологий и методов разработки, обоснование принятых решений и анализ их влияния на функциональность и эффективность всего комплекса мониторинга, а также определение функциональных и нефункциональных требований.

3.2.1. Разработка требований к программной части комплекса (ПЧК)

Требования к программной части комплекса можно разделить на функциональные и нефункциональные.

Функциональные требования:

1. Сбор данных ЭКГ:

ПЧК должен уметь собирать данные ЭКГ с помощью модуля ESP32 и AD8232.

2. Передача данных:

Данные ЭКГ должны передаваться от модуля на сервер в реальном времени.

3. Анализ данных:

Сервер должен анализировать поступающие данные, выявлять аномалии и предупреждать пользователя.

4. Хранение данных:

Данные должны сохраняться в базе данных для последующего анализа и отчетности.

5. Пользовательский интерфейс:

ПЧК должен иметь удобный пользовательский интерфейс для просмотра текущих и исторических данных ЭКГ, а также для получения заключений по показаниям ЭКГ.

6. Отображение данных в реальном времени:

Пользователям должна быть предоставлена возможность мониторинга показаний ЭКГ в реальном времени через веб-интерфейс.

Нефункциональные требования:

1. Отказоустойчивость:

Приложение должно быть устойчивым к сбоям и обеспечивать непрерывную работу в течение длительного времени.

2. Безопасность данных:

Все данные, передаваемые между компонентами системы, должны быть защищены с помощью соответствующих механизмов шифрования и аутентификации.

3. Производительность:

Приложение должно обеспечивать высокую производительность при передаче и обработке данных, чтобы минимизировать задержки и обеспечить оперативную реакцию на изменения состояния пациента.

4. Масштабируемость:

Система должна быть способна масштабироваться в зависимости от количества пользователей и объема данных, обрабатываемых ежедневно.

5. Простота использования:

Веб-интерфейс должен быть интуитивно понятным и легким в использовании даже для неопытных пользователей.

3.2.2. Разработка программной части для модуля сбора данных на базе ESP32

Выбор фреймворка

Для разработки программного обеспечения на ESP32 был выбран Arduino Framework в связке с PlatformIO благодаря простоте и удобству использования. Arduino Framework предоставляет все необходимые инструменты для написания, компиляции и загрузки кода на микроконтроллер. Его поддержка широкого спектра библиотек и большая пользовательская база делают решение возникающих проблем более легким.

Архитектура

При проектировании программного обеспечения использовался модульный подход, что позволяет изолировать различные функциональные компоненты системы. Это упрощает разработку, отладку и сопровождение кода, а также способствует повторному использованию и легкости расширения.

В системе используется паттерн Singleton, что позволяет гарантировать, что каждый модуль имеет только одну глобальную точку доступа. Это особенно важно для модулей, которые управляют аппаратными ресурсами, такими как SD-карта или BLE-сервисы. Singleton паттерн обеспечивает уникальность экземпляра и глобальный доступ к нему, что упрощает управление состоянием и ресурсами.

Структура программного кода на ESP32

Работа с модулями организована путем разделения на заголовочные файлы (.h) и файлы реализации (.cpp). Заголовочные файлы содержат объявления функций и классов, обеспечивая интерфейс между модулями, тогда как файлы реализации содержат конкретную реализацию этих функций.

Модули и точка входа

1. BLE Service Handler Module

Обрабатывает взаимодействие через Bluetooth Low Energy (BLE). Включает функции для настройки и обработки BLE-соединений.

2. Configuration Helper Module

Содержит параметры конфигурации, такие как настройки сети и параметры подключения. Обеспечивает централизованное управление конфигурациями.

3. Data Package Module

Осуществляет упаковку и распаковку данных для передачи. Включает структуры данных и функции для их обработки.

4. SPI Flash Module

Управляет взаимодействием с SPI Flash памятью. Содержит функции для чтения и записи данных во флеш-память.

5. Main Module(Основная точка входа в программу)

Отвечает за инициализацию системы и подключение модулей. Также содержит основной

цикл работы, в котором происходит вызов функций других модулей. Также в этом модуле происходит считывание сигнала ЭКГ с модулей AD8232, используя интерфейсы I2C.

Основная логика работы программы

Основная логика работы, инициализация и управление модулями содержится в Main Module программы. Ключевыми элементами данного модуля являются две основные функции Arduino Framework, а именно:

- **setup():** Инициализация системы и подключение модулей.
- **loop():** Основной цикл работы, в котором происходит вызов функций других модулей.

Рассмотрим основную логику работы этих функций:

setup():

1. Инициализация серийного порта для вывода отладочной информации.
2. Настройка BLE-сервиса с помощью вызова соответствующих функций из BLE Service Handler.
3. Инициализация и настройка портов для работы с AD8232.
4. Загрузка конфигурации из SPI Flash памяти.

loop():

1. Циклический опрос состояния системы.
2. Сбор данных с сенсоров и их обработка.
3. Упаковка данных с помощью Data Package Module.
4. Передача данных через BLE на сервер.
5. Управление состояниями системы и обработка событий, таких как потеря соединения или ошибки передачи данных. (В данном случае данные сохраняются на SD карту)

Тестирование

В ходе разработки проекта возникли ограничения в виде отсутствия собранного тестирующего стенда, что затруднило проведение автоматизированного традиционного тестирования.

На данный момент применяется тестирование модуля только в области форматирования и проверки стиля кода, который на него загружается, что в конечном итоге все равно способствует повышению его читаемости и качества.

3.2.3. Приложение для передачи данных (Application Layer)

Electron-приложение служит посредником для передачи данных ЭКГ с устройства ESP32 на сервер. Для обеспечения безопасности и функциональности приложения были выделен набор наиболее важных сценариев использования.

Сценарии использования

- **Аутентификация и авторизация:**

1. Пользователь открывает Electron приложение и вводит свои учетные данные (логин и пароль).
2. Приложение отправляет данные на сервер для аутентификации.
3. В случае успешной аутентификации пользователь получает доступ к функционалу приложения.
4. Пользователь может выйти из системы, закрыв Electron приложение или используя соответствующую функцию в интерфейсе приложения.
5. После выхода из системы пользователь теряет доступ к функционалу приложения до следующего входа.

- **Отправка данных ЭКГ на анализ с ESP32 на сервер:**

1. Пользователь может отправлять данные ЭКГ на сервер для анализа и получения дополнительной информации о своем состоянии здоровья с помощью технологии BLE.

- **Просмотр метаданных:**

1. Пользователь может видеть общее количество отправленных пакетов.
2. Пользователь может просматривать информацию о качестве подключения.

Проектированием и разработкой этого модуля занимался студент группы 382006-2 Юнин Даниил Дмитриевич.

3.2.4. Сервер для сбора, анализа и хранения данных (Server Layer)

Выбор фреймворка и языка

Python вместе с Flask был выбран в качестве основного фреймворка для разработки серверной части веб-приложения по нескольким причинам:

- 1. Легковесность и Гибкость:**

Flask отличается простотой и легковесностью в использовании, что делает его идеальным выбором для разработки небольших и средних проектов. При этом, он обладает достаточной гибкостью, чтобы удовлетворить потребности различных задач.

- 2. Простота Создания RESTful API:**

Flask предоставляет все необходимые инструменты для создания RESTful API, что позволяет легко интегрировать сервер с различными компонентами системы. Это критически важно для проекта, где требуется эффективное взаимодействие с базой данных и клиентскими приложениями.

- 3. Широкая Поддержка Библиотек:**

Язык программирования Python обладает богатой экосистемой библиотек, что делает его идеальным выбором для разработки веб-приложений. Python предоставляет широкие возможности для анализа данных, что идеально подходит для обработки медицинских данных.

Архитектура

В разработке серверной части веб-приложения использован принцип архитектуры, основанный на модели MVC (Model-View-Controller), дополненный использованием REST API для взаимодействия с клиентской стороной, написанной на Vue.js.

На серверной стороне, основные компоненты архитектуры включают в себя:

- **Модель (Model):**

Представлена моделями данных и логикой взаимодействия с базой данных, реализованными с использованием ORM (объектно-реляционного отображения). Данные хранятся и обрабатываются на сервере, что обеспечивает централизованное хранение и управление информацией.

- **Представление (View):**

Фронтенд, написанный на Vue.js, действует как "представление" и отображает данные, полученные через REST API от сервера. Фронтенд обеспечивает пользовательский интерфейс для взаимодействия с приложением и представляет пользовательский опыт.

- **Контроллер (Controller):**

Контроллеры, представленные в виде маршрутов в API, обрабатывают HTTP-запросы от клиента и возвращают соответствующие данные. Они являются посредниками между моделью и представлением, обеспечивая передачу данных между ними и управление бизнес-логикой приложения.

Помимо этого, в архитектуре также будет интегрирован модуль взаимодействия с приложением на Electron. Этот модуль будет отвечать за получение данных ЭКГ для дальнейшей обработки на сервере, а также за отображение ЭКГ и заключений на клиентской стороне приложения.

Данный подход обеспечивает четкое разделение бизнес-логики, представления данных и управления запросами на отдельные компоненты, что делает приложение более модульным, легко сопровождаемым и масштабируемым.

Структура сервера

Модуль ORM и Работа с Базой Данных (ORM and Database Handling)

- **Описание:** Модуль отвечает за взаимодействие с базой данных с помощью объектно-реляционного отображения и выполнения SQL-запросов.

- **Назначение:**

- Определение моделей данных.
- Управление транзакциями базы данных.
- Обеспечение абстракции для работы с реляционными данными.
- Подключение к базе данных.
- Управление схемой базы данных и инициализация данных.

- **Компоненты:**

- Модели данных.
- Конфигурация базы данных.
- Скрипты инициализации базы данных.
- Сессии для взаимодействия с базой данных.

Модуль Роутинга (Routing)

- **Описание:** Модуль управляет маршрутами и эндпоинтами API.

- **Назначение:**

- Определение URL маршрутов.
- Обработка входящих HTTP-запросов.
- Связывание запросов с соответствующими функциями контроллеров.

- **Компоненты:**

- Маршруты для аутентификации.
- Маршруты для работы с веб-приложением.
- Маршруты для поддержания взаимодействия с модулем Electron.

Модуль ИИ для Анализа Данных (AI Data Analysis)

- **Описание:** Модуль использует алгоритмы машинного обучения и другие методы искусственного интеллекта для анализа данных ЭКГ.

- **Назначение:**
 - Предобработка данных ЭКГ.
 - Анализ и классификация данных ЭКГ.
 - Генерация заключений на основе анализа.
- **Компоненты:**
 - Алгоритмы машинного обучения.
 - Модели для анализа данных.
 - Методы предсказания и оценки.

Модуль Взаимодействия с Electron (Electron Integration)

- **Описание:** Модуль обеспечивает интеграцию с десктопным приложением, написанным на Electron.
- **Назначение:**
 - Получение данных ЭКГ от десктопного приложения.
- **Компоненты:**
 - Методы для работы с API Electron
 - Логика обработки входящих данных.
 - Отправка пакетов ЭКГ записей в базу данных

Модуль Аутентификации и Авторизации (Authentication and Authorization)

- **Описание:** Модуль управляет регистрацией, входом и выходом пользователей, а также контролем доступа.
- **Назначение:**
 - Обработка регистрации и аутентификации пользователей.
 - Управление сессиями и токенами.
 - Обеспечение безопасности доступа к API.

- **Компоненты:**

- Методы регистрации и входа пользователей.
- Управление JWT токенами.
- Политики доступа и проверки прав пользователей.

Модуль Логирования и Мониторинга (Logging and Monitoring)

- **Описание:** Модуль отвечает за запись логов и мониторинг состояния системы.

- **Назначение:**

- Логирование ошибок и событий.
- Мониторинг производительности и состояния системы.

- **Компоненты:**

- Конфигурация логирования.
- Методы мониторинга.

Модуль Тестирования (Testing)

- **Описание:** Модуль обеспечивает тестирование всех компонентов сервера.

- **Назначение:**

- Написание и выполнение тестов для проверки функциональности.
- Обеспечение качества кода и предотвращение ошибок.

- **Компоненты:**

- Тесты для маршрутов и API.
- Тесты для взаимодействия с базой данных.

Основная логика работы программы

Инициализация и настройка приложения

1. При запуске приложения инициализируется объект Flask.

2. Устанавливается соединение с базой данных.
3. Настраивается JWT для безопасной аутентификации пользователей.

Обработка запросов

1. При поступлении запроса на сервер, Flask маршрутизирует его к соответствующему обработчику.
2. Обработчики маршрутов выполняют необходимые операции, включая обработку запросов аутентификации, доступа к данным, выполнение бизнес-логики и т.д.
3. Данные запросов приходят в формате JSON.

Аутентификация и авторизация

1. При получении запроса, требующего аутентификации, сервер проверяет наличие и валидность JWT в заголовке запроса.
2. Если токен действителен, пользователь получает доступ к запрашиваемым ресурсам или операциям.

Взаимодействие с данными

1. При выполнении запросов на получение, создание, обновление или удаление данных, сервер взаимодействует с базой данных через модели и ORM SQLAlchemy.
2. Данные могут быть переданы обратно клиенту в виде JSON.

Работа с ИИ

1. При получении данных ЭКГ от клиента или других источников, сервер обрабатывает их с использованием алгоритмов машинного обучения и анализа данных.
2. Результаты анализа, включая диагностику и заключения, могут быть отправлены обратно клиенту для отображения и дальнейшего использования.

Проектированием и разработкой этой части модуля занимался студент группы 382006-2 Юнин Даниил Дмитриевич.

Тестирование

1. При разработке нового функционала, сервер проходит тестирование вручную с помощью разработанных модульных тестов.
2. Тесты включают в себя проверку основных сценариев использования, а также тестирование работы с данными и взаимодействия с клиентами.

Использование Docker для контейнеризации приложения

Для развертывания серверной части проекта был использован Docker, инструмент, предоставляющий среду контейнеризации для приложений. В этом контексте были созданы два основных файла: Dockerfile и docker-compose.yml.

Dockerfile представляет собой инструкции для создания Docker-образа, включающего необходимые компоненты для запуска приложения. В нем определены шаги по установке Python, установке зависимостей из файла requirements.txt, а также копирование всех файлов проекта внутрь контейнера.

docker-compose.yml, в свою очередь, используется для описания многоконтейнерного приложения и его окружения. В этом файле определен сервис (web), который запускает Flask-приложение в контейнере. Также указана зависимость от контейнера с PostgreSQL, определены порты для доступа к приложению, переменные окружения и настройки томов для монтирования файлов проекта в контейнер.

Использование Docker значительно упрощает процесс развертывания и управления приложением, обеспечивая изолированную среду выполнения и облегчая его перенос между различными окружениями разработки и производства.

3.2.5. База данных (Data Storage Layer)

Выбор СУБД

При принятии решения о выборе базы данных для проекта важно провести тщатель-

ный анализ доступных вариантов, учитывая их особенности, преимущества и недостатки. На рассмотрении находились следующие варианты:

- **PostgreSQL:** надежная и расширяемая реляционная база данных с открытым исходным кодом, известная своей стабильностью и производительностью.
- **SQLite:** легковесная база данных, хранящаяся в виде одного файла, что обеспечивает простоту использования и удобство встроенного хранения данных для приложений с небольшим объемом информации.
- **MySQL:** широко известная реляционная база данных с открытым исходным кодом, которая обладает высокой производительностью и надежностью, что делает ее предпочтительным выбором для развертывания крупных и высоконагруженных систем.
- **MongoDB:** NoSQL база данных, ориентированная на хранение документов, что обеспечивает гибкость в работе с неструктурированными данными и удобство масштабирования.
- **Redis:** in-memory база данных, специализирующаяся на хранении данных в оперативной памяти, обеспечивающая высокую скорость доступа к данным и широкое применение для кэширования и хранения временных данных.

Было решено использовать PostgreSQL, что было обусловлено несколькими факторами. Первоначально, база данных SQLite, хотя и представляет собой легковесное решение, может оказаться недостаточно мощной для проекта, требующего сложных запросов и высокой масштабируемости. MongoDB, с другой стороны, хотя и обладает гибкостью и способностью работать с неструктурированными данными, может не соответствовать требованиям проекта, связанным с реляционной структурой данных, характерной для передачи ЭКГ. В то время как MySQL, популярная реляционная база данных, может оказаться менее гибкой и расширяемой, чем PostgreSQL, и не обеспечить необходимой поддержки для проекта в долгосрочной перспективе. Наконец, Redis, хотя и предлагает высокую производительность в работе с данными в оперативной памяти, может быть менее универсальным в отношении структуры данных и операций с ними. Таким образом, в результате тщательного сопоставления и анализа, PostgreSQL был выбран как наиболее подходящий вариант для проекта системы передачи данных ЭКГ, учитывая его надежность, гибкость, расширяемость и возможности работы с реляционными данными.

Проектирование структуры БД

Проектирование БД для приложения состоит из следующих этапов:

1. Создание схемы базы данных:

- В ходе проектирования было определено три основных сущности: пользователи данные ЭКГ и аутентификационные данные.
- Для каждой сущности была разработана соответствующая таблица в базе данных, представляющая собой логическую структуру для хранения данных.

2. Определение полей и их типов:

- Каждая таблица содержит набор полей, определяющих характеристики сущности.
- Типы данных для полей выбирались с учетом требований к хранению и обработке информации.

3. Определение отношений между таблицами:

- Для связывания данных между таблицами были определены внешние ключи и отношения.
- Например, таблица аутентификационных данных имеет внешний ключ, связывающий ее с таблицей пользователей.

По итогу было создано 3 таблицы, которые являются требуемым минимумом для полноценного функционирования проекта.

1. Таблица пользователей (*users*):

В этой таблице хранится информация о пользователях системы. Включает в себя идентификатор пользователя (*id*), полное имя (*full_name*), пол (*gender*), возраст (*age*), вес (*weight*) и дату рождения (*date_of_birth*). Такие атрибуты позволяют идентифицировать пользователей и хранить их основные характеристики.

2. Таблица аутентификации (*auth*):

Эта таблица содержит информацию о пользователях, необходимую для аутентификации при входе в систему. Включает в себя идентификатор (*id*), логин (*login*), хэшированный пароль (*password*), адрес электронной почты (*email*) и ссылку на пользователя (*user_id*), устанавливающую связь с таблицей "users".

3. Таблица записей данных ЭКГ (*records*):

В этой таблице хранятся записи данных ЭКГ пользователей. Включает в себя идентификатор записи (*id*), дату чтения (*reading_date*), временной интервал (*time_interval*) и сами данные ЭКГ в формате JSON (*readings_data*). Такая структура позволяет хранить и организовывать данные ЭКГ для каждого пользователя.

Использование Docker для контейнеризации приложения

PostgreSQL был развернут с использованием Docker, что позволило создать изолированную среду выполнения и обеспечить удобство управления и масштабируемость приложения. С помощью инструмента `docker-compose` было описано многоконтейнерное окружение для приложения, включая настройки контейнера PostgreSQL, его зависимости, порты и переменные окружения.

3.2.6. Web Layer (Веб-приложение)

Web-приложение является основным источником для изучения и просмотра пользователем своих данных ЭКГ. Оно реализовано как одностраничное приложение (SPA), предоставляющее пользователю удобный интерфейс для взаимодействия с данными.

Сценарии использования

1. Авторизация и аутентификация:

- Пользователи могут войти в систему, предоставив свои учетные данные.
- После успешной аутентификации пользователи получают доступ к функционалу приложения.

2. Заполнение профиля:

- Пользователи имеют возможность добавлять и изменять свои личные данные в своем профиле.
- Обновленные данные сохраняются на сервере для последующего использования.

3. Отображение графиков данных ЭКГ:

- Пользователи могут запрашивать просмотр графиков данных ЭКГ, полученных от Electron приложения.
- Графики отображаются в интерфейсе приложения для удобного изучения пользователем своих данных.

4. Отображение показаний здоровья и рекомендаций:

- Пользователям предоставляется информация о их текущем состоянии здоровья, основанная на данных ЭКГ и других медицинских параметрах.
- Кроме того, пользователи могут получать рекомендации по улучшению своего здоровья на основе анализа предоставленных данных.

Проектированием и разработкой этого модуля занимался студент группы 382006-2 Юнин Даниил Дмитриевич.

4. Заключение

В ходе выполнения выпускной квалификационной работы был разработан программно-аппаратный комплекс для мониторинга показателей сердца человека, основанный на методе электрокардиографии (ЭКГ).

В ходе работы были достигнуты следующие результаты:

1. Изучение принципов взаимодействия с микроконтроллером

Были изучены основы программирования и взаимодействия с микроконтроллерами, что позволило эффективно реализовать аппаратную часть комплекса.

2. Исследование фреймворков для разработки сервера

Были проанализированы различные фреймворки и выбраны наиболее подходящие для реализации серверной части комплекса.

3. Подбор компонентов для аппаратной части

Были тщательно отобраны компоненты, учитывая требования к функциональности, надежности и энергоэффективности устройства.

4. Моделирование схемы устройства

С помощью специализированных программных инструментов была создана трехмерная модель устройства с распиновкой и необходимыми подключениями.

5. Разработка серверной части и базы данных

Было создано и реализовано серверное ПО и база данных для приема, хранения и обработки данных, полученных от аппаратной части комплекса.

В дальнейшем развитии проекта можно рассмотреть несколько ключевых шагов. Во-первых, к нашему веб-приложению можно добавить мобильное приложение. Это позволит расширить доступность нашего комплекса для пользователей, позволяя им просматривать свои показатели сердечной активности непосредственно через их мобильные устройства.

Во-вторых, мы можем оптимизировать инфраструктуру, используя контейнеризацию с помощью Docker. Мы можем разнести базу данных и серверную часть на разные машины, что повысит отказоустойчивость и масштабируемость нашего комплекса.

5. Список литературы

1. Townsend, K., Cufi, C., Akiba, & Davidson, R. (2014). *Getting Started with Bluetooth Low Energy*. O'Reilly Media, Inc. ISBN 978-1-4919-4951-1.
2. Зудбинов, Ю. И. (2003). *Азбука ЭКГ (Издание 3)*. Ростов-на-Дону: Феникс. ISBN 5-222-02964-6.
3. Синельников, Р. Д. (1979). *Атлас анатомии человека*. Москва: Медицина. Том 2.
4. Brawnwald, L. D. (1992). *Heart disease*. Стр. 122.
5. Python documentation. [Электронный ресурс]. URL: <https://docs.python.org/> (дата обращения: 06.06.2024).
6. ESP32 Documentation. [Электронный ресурс]. URL: <https://www.espressif.com/en/products/socs/esp32/overview> (дата обращения: 06.06.2024).
7. Рабаи, Ж. М., Чандракасан, А., & Николич, Б. (2007). *Цифровые интегральные схемы. Методология проектирования = Digital Integrated Circuits*. Москва: Вильямс. ISBN 0-13-090996-3.
8. Гринберг, М. (2018). *Flask Web Development: Developing Web Applications with Python*. Санкт-Петербург: Наука.
9. Су, Л. С., & Обе, Р. О. (2019). *PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database*. Москва: Питер.
10. Cochrane, K. (2019). *Docker Cookbook: Over 100 practical and insightful recipes to build distributed applications with Docker, 2nd Edition*. Москва: Питер.
11. Pallets Projects. Flask Documentation [Электронный ресурс]. URL: <https://flask.palletsprojects.com/> (дата обращения: 05.05.2024).
12. SQLAlchemy Documentation [Электронный ресурс]. URL: <https://docs.sqlalchemy.org/en/20/> (дата обращения: 05.05.2024).

13. Чен, Ц. Ч., Лин, Ш. Ю., Чан, В. И., и др. (2021). Измерение физиологических сигналов через одежду с помощью портативной системы электрокардиограммы. *Сенсоры*, 21(11), 3668. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8197543/>
14. Фу, Ю., Жао, Ч., Дун, Ю., Ванг, К., и др. (2020). RNN-Aided Human Velocity Estimation from a Single IMU. *Sensors*, 20(13), статья 3651.
URL: <https://www.mdpi.com/1424-8220/20/13/3651/htm>
15. Espressif Systems. ESP32-C3.
URL: <https://www.espressif.com/en/products/socs/esp32-c3>
16. Espressif Systems. ESP32-S3.
URL: <https://www.espressif.com/en/products/socs/esp32-s3>
17. LabData. Активные сухие электроды для биопотенциалов ЭКГ, ЭЭГ, ЭМГ. URL: https://labdata.ru/article/dry_elecrodes

6. Приложение

Полная версия исходного кода проекта расположена в github репозитории. Ссылка на проект прикреплена ниже:

https://github.com/danielbulgakov/ecg_esp32/