

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О. В. Непомнящий

« ___ » _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

код и наименование направления

Мобильная система регистрации электрической активности головного мозга

тема

Руководитель

подпись, дата

доцент, канд. техн. наук
должность, ученая степень

М. С. Медведев
инициалы, фамилия

Студент

подпись, дата

А. А. Темерова
инициалы, фамилия

Нормоконтролер

подпись, дата

доцент, канд. техн. наук
должность, ученая степень

В. И. Иванов
инициалы, фамилия

Красноярск 2019

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О. В. Непомнящий

« ____ » _____ 2019 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Темеровой Агафье Александровне
Группа КИ15-07Б направление 09.03.01 «Информатика и вычислительная техника», направление (специальность) 09.03.01 "Информатика и вычислительная техника".

Тема выпускной квалификационной работы «Мобильная система регистрации электрической активности головного мозга»

Утверждена приказом по университету № _____ от _____

Руководитель ВКР М.С. Медведев, доцент кафедры вычислительной техники ИКИТ СФУ.

Исходные данные для ВКР: задание на бакалаврскую работу.

Перечень разделов ВКР:

- Введение;
- 1 Анализ предметной области, выявление требований к мобильному приложению;
- 2 Регистрация электроэнцефалограммы;
- 3 Описание системы, описание структурной схемы, описание режимов работы;
- 4 Разработка программного обеспечения;
- Заключение;
- Список использованных источников;
- Список сокращений;
- Приложение А;
- Приложение Б.

Руководитель ВКР _____ М. С. Медведев

Задание принял к исполнению _____ А. А. Темерова

« ___ » _____ 201_ г.

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ предметной области	6
1.1 «Cortex Neurobotix» («Нейроботикс Кортекс»)	7
1.2 «MyEmotiv»	8
1.3 «EEG Analyzer»	8
1.4 Обобщение анализа предметной области.....	9
2 Регистрация электроэнцефалограммы.....	11
2.1 ЭЭГ как метод исследования нервной системы	11
2.2 Характеристики сигнала.....	13
2.3 Цикл сна человека и его фазы	13
2.4 Частотные-амплитудные характеристики сигнала.....	14
2.5 Метод диагностики	15
2.6 Выводы.....	16
3 Описание системы.....	17
3.1 Описание структурной схемы	17
3.1.1 «Блок снятия сигнала»	17
3.1.2 «Блок управления и регистрации».....	17
3.1.3 «Блок приёма и передачи»	18
3.1.4 «Блок исполнительных устройств»	18
3.2 Режимы работы системы.....	18
3.2.1 Автономный режим работы.....	18
3.2.2 Управляемый режим работы	19
4 Разработка программного обеспечения	20
4.1 Разработка программы для микроконтроллера	20
4.1.1 Среда разработки для Arduino	20
4.1.2 Алгоритм работы программы.....	20
4.2 Разработка программы для мобильного приложения	22
4.2.1 Анализ существующих средств разработки.	22

4.2.1.1 Нативная среда разработки.....	22
4.2.1.2 Веб приложение	23
4.2.1.3 Гибридная среда разработки	24
4.2.2 Среда разработки Apache Cordova	24
4.2.3 Ionic Framework.....	25
4.2.4 Typescript	26
4.2.5 Модуль распознавания частотной характеристики сигнала	26
4.2.5.1 Алгоритм распознавания частоты сигнала	26
4.2.5.2 Программная реализация алгоритма	28
4.2.6 Модуль визуализации сигнала D3.js.....	28
4.2.7 Алгоритм работы мобильного приложения.....	29
4.2.8 Скриншоты работы мобильного приложения	36
Заключение	37
Список использованных источников	38
Список сокращений	40
Приложение А	41
Приложение Б.....	44

ВВЕДЕНИЕ

Сохранение сна для всех видов животных предполагает, что сон выполняет жизненно важную функцию. В исследовании [2] показано, что поскольку биологическая активность неизбежно связана с производством продуктов метаболической деградации, сон является важной функцией очистки множества потенциально токсичных отходов ЦНС. Исследование [2] указывает на то, что корковое интерстициальное (между нейронами и глиальными клетками) пространство увеличивается более чем на 60% во время сна, что приводит к эффективному конвективному очищению от олигомеров и амилоидных отложений, которые оказывают неблагоприятное влияние на синаптическую передачу [2] и могут вызвать необратимое повреждение нейронов [2].

Цель сна, вероятно, в том, что восстановительная функция сна может быть связана с переключением головного мозга в функциональное состояние, которое облегчает очистку от продуктов распада нервной деятельности, которые накапливаются во время бодрствования [2].

Болезни ЦНС и психические расстройства часто сопровождаются развитием различных расстройств сна. Так, в госпитальной неврологической популяции частота выявления нарушений сна варьирует от 40 до 83% в зависимости от формы основной патологии [3].

Для диагностики нарушений сна в клинике проводится 12-и или 24-х часовая запись электрической активности мозга при сне и бодрствовании. Диагностика сна в больничных условиях имеет недостатки, такие как: смена обстановки, которая может повлиять на эмоциональное состояние и испортить клиническую картину; если проводится 24-х часовое обследование, то необходимо провести полный день под присмотром врача с его оборудованием. Таким образом, для диагностики состояния головного мозга во время сна и бодрствования лучше бы подошла персональная мобильная носимая система,

которая может записывать полученный результат в любое удобное время и выдать его при необходимости врачу.

Современным решением является приложение на смартфон для основных операционных систем, способное сопрягаться по bluetooth с носимым устройством регистрации электроэнцефалограммы, принимать данные, отправлять команды и анализировать полученные данные, а также с удобным для пользователя интерфейсом.

Цель бакалаврской работы: разработать программное обеспечение для микроконтроллера Arduino для регистрации, оцифровки и передачи данных ЭЭГ, разработать кроссплатформенное мобильное приложение для получения, обработки и проведения диагностики состояния пользователя по полученным данным.

Задачи:

1. Выполнить анализ существующих решений в предметной области;
2. Проанализировать полученную информацию. Выделить необходимые функции программного обеспечения;
3. Проанализировать входные данные, методику регистрации и обработки ЭЭГ;
4. Выбрать средства разработки;
5. Разработать программное обеспечение для переносного устройства регистрации ЭЭГ. В том числе мобильное приложение для платформ iOS, Android.

1 Анализ предметной области

Электроэнцефалография (ЭЭГ) – это исследование биоэлектрических токов головного мозга при помощи специализированного оборудования.

Необходимо получить мобильное программное обеспечение для переносного устройства регистрации ЭЭГ, упрощенная схема которого представлена на рисунке 1.

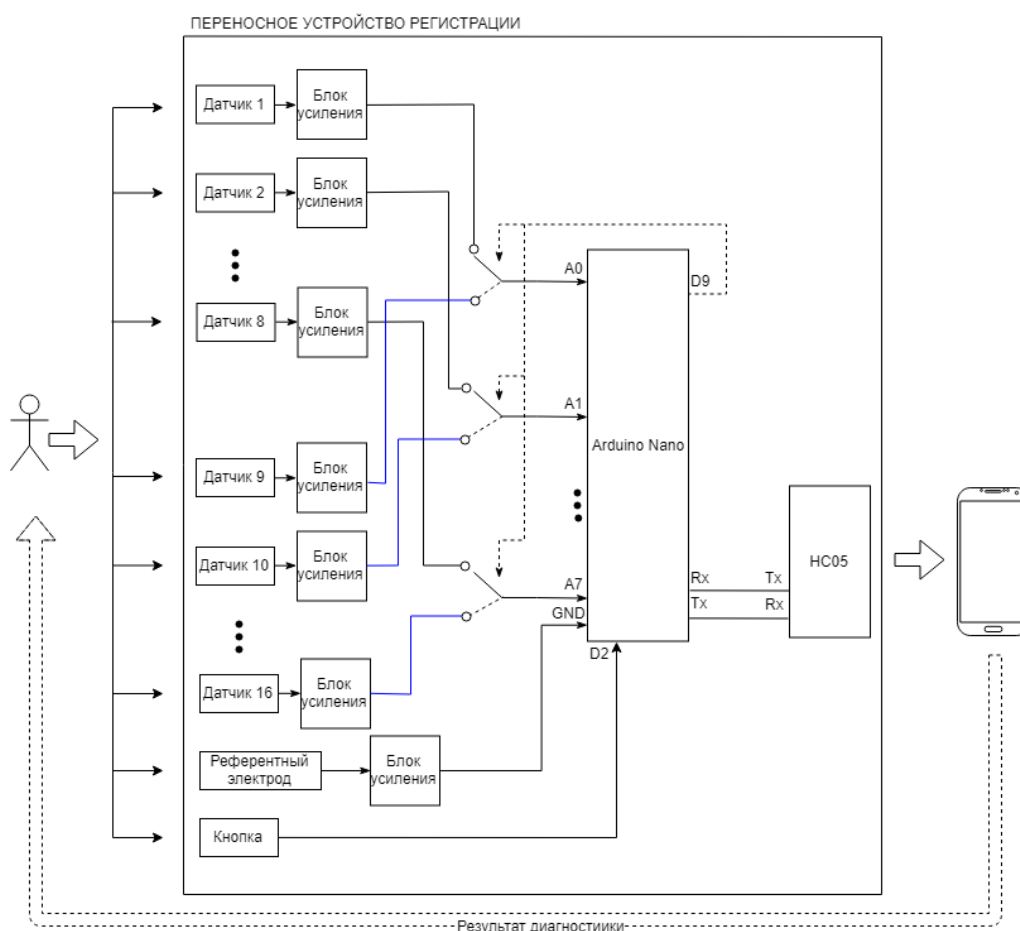


Рисунок 1 – упрощенная схема переносного устройства регистрации ЭЭГ

В качестве конкурентных решений будут рассмотрены 3 программы предназначенные для обработки электроэнцефалограммы:

1. «Нейроботикс Кортекс» (гарнитура «Neuro Play8C»);
2. «MyEmotiv» (гарнитура «Emotiv Eroc»);
3. «EEG Analyzer» (гарнитура «NeuroSky»).

1.1 «Cortex Neurobotix» («Нейроботикс Кортекс»)

«Cortex Neurobotix» – это программа для регистрации, предварительной обработки и анализа сигналов электрической активности головного мозга, а также для классификации состояний биопотенциалов головного мозга и формирования управляющих команд по состояниям. Программа опрашивает биоусилители «NVX» и «Нейробелт», предоставляет возможность использовать файл EDF для эмуляции работы усилителя, позволяет настраивать множество параметров регистрации сигналов ЭЭГ, записывает данные в файл формата EDF, позволяет просматривать файлы формата EDF и реализует функционал классификатора интерфейса мозг-компьютер [4].

Достоинства и недостатки представлены в таблице 1, внешний вид программного обеспечения изображен на рисунке 2.

Таблица 1 – Достоинства и недостатки ПО «Cortex Neurobotix»

Достоинства	Недостатки
Поддерживается русский язык	Высокая цена устройства, для которого разработано ПО – 80000 тыс. руб
Есть механизмы для диагностики состояния головного мозга	Нет мобильного приложения
Генерация тестового сигнала	Малое количество каналов

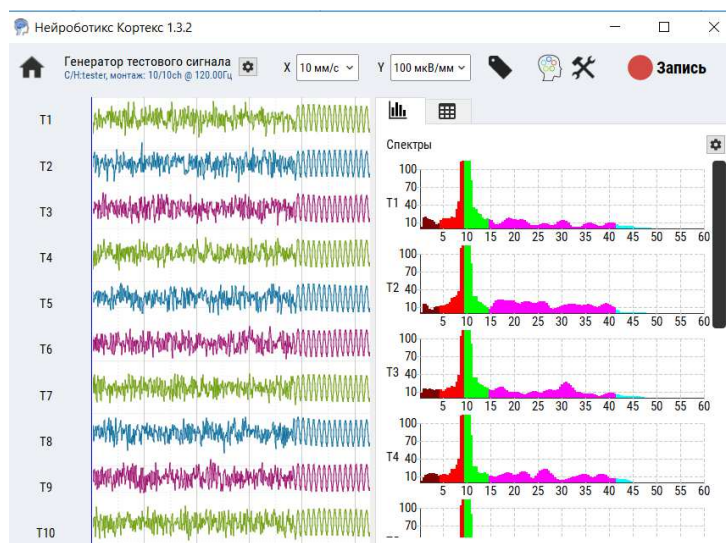


Рисунок 2 – Внешний вид программного обеспечения Cortex Neurobotix

1.2 «MyEmotiv»

«MyEmotiv» – это приложение, разработанное для известной гарнитуры «Emotiv Ерос». Достоинства и недостатки представлены в таблице 2, внешний вид мобильного приложения изображен на рисунке 3.

Таблица 2 – Достоинства и недостатки ПО «MyEmotiv»

Достоинства	Недостатки
Есть мобильное приложение	Высокая цена устройства, для которого разработано ПО: 300 – 1100\$
Поддержка большого количества датчиков в зависимости от устройства. От 5 до 14 активных электродов	Нет поддержки русского языка
	Малое количество каналов
	Не работает с другими устройствами

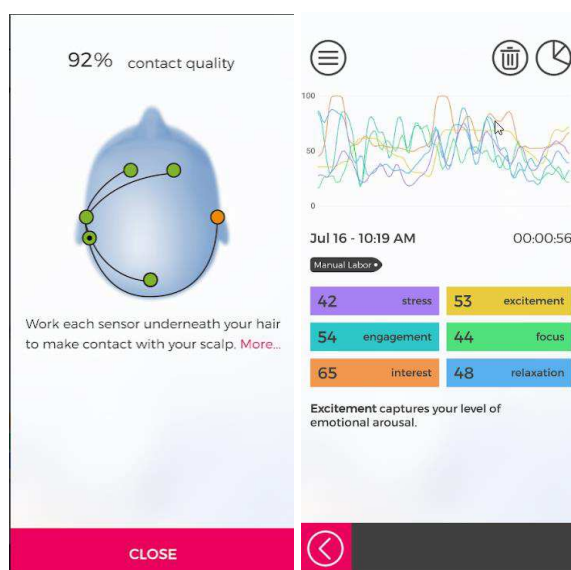


Рисунок 3 – Внешний вид программного обеспечения «MyEmotiv»

1.3 «EEG Analyzer»

«EEG Analyzer» – это одно из многих приложений, разработанное для популярной и доступной гарнитуры от «NeuroSky». Главным недостатком является наличие лишь одного активного электрода.

Достоинства и недостатки «EEG Analyzer» представлены в таблице 3, внешний вид приложения изображен на рисунке 4.

Таблица 3 – Достоинства и недостатки «EEG Analyzer»

Достоинства	Недостатки
Низкая стоимость необходимого оборудования. 5 – 15 тыс. рублей	Всего 1 активный электрод в гарнитуре «NeuroSky»
Есть мобильное приложение	Нет поддержки русского языка
Режим контроля уровня медитации и концентрации	Невозможно зайти в приложение без подключения необходимой гарнитуры
	Не представляет возможности диагностики в виду наличия всего одного электрода

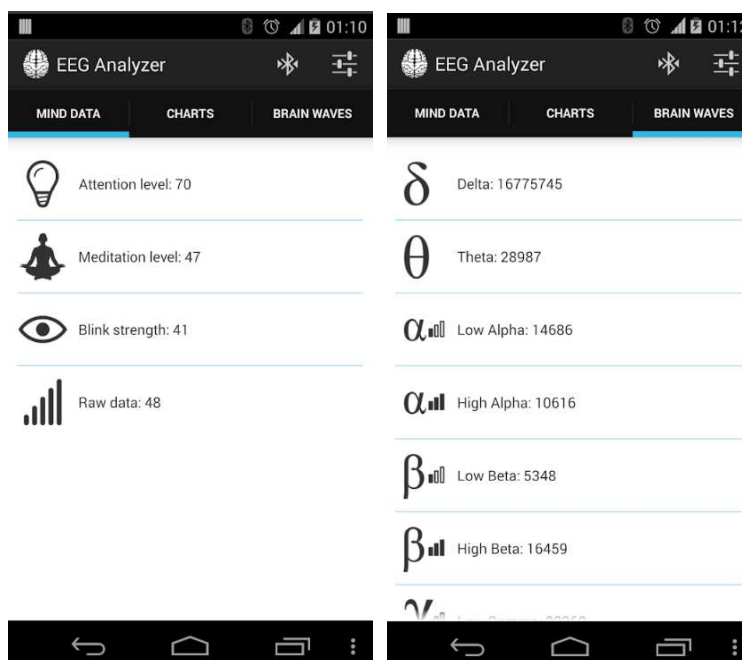


Рисунок 4 – Внешний вид программного обеспечения EEG Analyzer

1.4 Обобщение анализа предметной области

Для обобщения анализа предметной области наиболее сведения сведены в таблицу 4.

Таблица 4 – Сравнение Cortex Neurobotix, EEG Analyzer, MyEmotiv

Название	Необходимое устр-во	Поддержка iOS, Android	Русский язык	Кол-во каналов (датчиков)	Диагностическое значение
Cortex Neurobotix	Neuro Play8C	нет	есть	8 - 10	есть
EEG Analyzer	NeuroSky	есть	нет	1	нет
MyEmotiv	Emotiv Epos	есть	нет	до 14	есть

Обобщая изложенный материал, можно сделать вывод, что рассмотренные решения популярных на российском рынке компаний предоставляют программное обеспечение исключительно для своих собственных гарнитур. Это обуславливается отсутствием единого протокола для работы с различными нейрогарнитурами. Таким образом для переносного устройства регистрации ЭЭГ необходимо написать своё собственное программное обеспечение, обладающее свойствами, представленными в таблице 5. Так как частота дискретизации задаётся программным обеспечением, необходимое значение также вынесено в таблицу 5.

Таблица 5 – Свойства, необходимые для разрабатываемого ПО

Частота дискретизации	Диагностика патологий	Операционные системы	Язык
250 Гц	необходима	iOS, Android	Русский

2 Регистрация электроэнцефалограммы

2.1 ЭЭГ как метод исследования нервной системы

Электроэнцефалография (ЭЭГ) – это исследование биоэлектрических токов головного мозга при помощи специализированного оборудования. Датчики укрепляются на голове с помощью специальных шлемов, состоящих из резиновых жгутов; на мочках ушей устанавливаются электроды заземления в форме прищепок. Электрический потенциал отводится с поверхности кожи головы с помощью датчиков-электродов, и подаётся на входы усилительно-регистрирующих устройств. Сигналы оцифровываются для дальнейшей обработки. Используемая схема отведения биопотенциалов должна отвечать основным требованиям, которым соответствует наиболее часто используемая в клинической практике международная схема «10 – 20%».

Система «10 – 20%» – стандартная система размещения электродов на поверхности головы, которая рекомендована Международной федерацией электроэнцефалографии и клинической нейрофизиологии. Место расположение электродов определяется следующим образом: линия, соединяющая переносицу (назион) и затылочный бугор (инион), делится на 10 равных отрезков. Первый и последний электроды накладываются на расстоянии, соответствующем 10 % общей длины линии, от инициона или назиона. От первого электрода на расстоянии, соответствующем 20 % общей длины линии, накладывается другой электрод и т. д. Таким образом, по линии переносица-затылок (медианная линия, или вертексная линия) накладываются 5 электродов. На линии наружные слуховые проходы (центральная линия) так же накладываются по два электрода на каждое полушарие и макушечный электрод. Линии, параллельные медианной, и проходящие через электроды, наложенные по центральной линии, носят название парасагиттальных и височных (правая и левая). При этом на парасагиттальной линии накладываются по 5 электродов, а на височные – по 3 электрода [5] (см. рисунок 5).

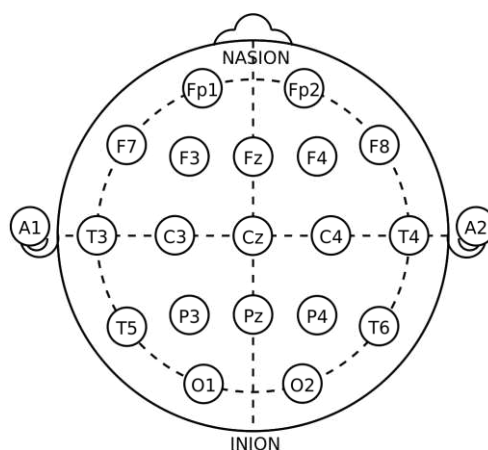


Рисунок 5 – Расположение электродов для регистрации ЭЭГ по международной системе 10-20 [5]

Американское Общество Клинической Нейрофизиологии предусматривает, что 16 каналов одновременной регистрации в настоящее время считается минимальным количеством, которое может охватить области, генерирующие большинство нормальных или патологических паттернов ЭЭГ. Для мониторинга других физиологических показателей могут потребоваться дополнительные каналы [6].

При регистрации ЭЭГ используется два способа отведения потенциалов:

- монополярная – между активным и референтным электродом. В качестве референтного электрода используется электрод на мочке уха;
- биполярная – разность потенциалов между двумя активными электродами.

Для анализа результатов полученной ЭЭГ применяются методы спектрального анализа. Результаты спектрального анализа представляются в виде графиков спектральной плотности-мощности. По любому показателю, такому как абсолютная и относительная мощность, амплитуда, частота, соотношение ритмов, и т.д., может проводиться картирование электрической активности. Примеры картирования представлены на рисунке 6.

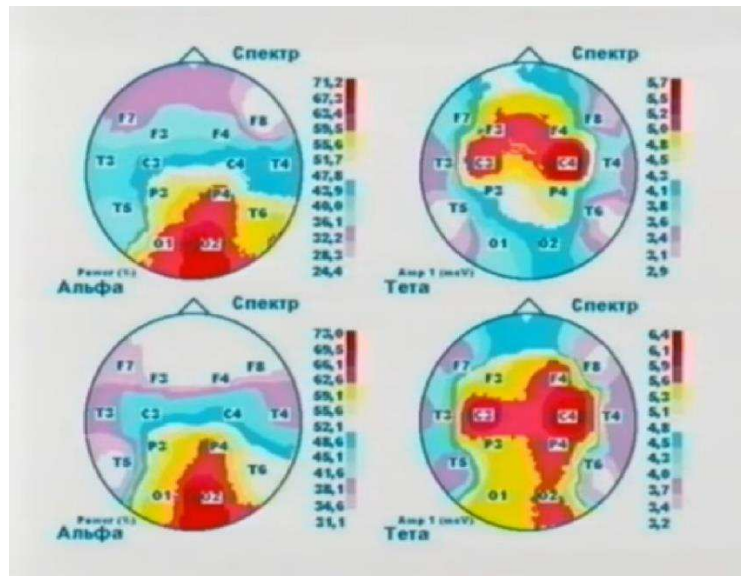


Рисунок 6 – Картирование ЭЭГ [7]

2.2 Характеристики сигнала

Стандартная полоса пропускания при записи ЭЭГ составляет от 0.5 до 70 Гц. Характеристики сигнала ЭЭГ, считающиеся нормой для здорового человека отличны в зависимости от состояния бодрствования или сна, а также от фазы сна.

2.3 Цикл сна человека и его фазы

Сон человека делят на быстрый и медленный.

Быстрый сон, он же парадоксальный, или REM-сон. Название «парадоксальный» связано с тем, что во время этой фазы сна сочетаются полное расслабление мышц и высокая активность мозга [8].

Медленный сон, он же ортодоксальный, или NREM (Not Rapid Eye Movement) сон и отражает тот факт, что для данной фазы не характерны быстрые движения глаз [8]. Медленный сон делится на 4 стадии: дремота, лёгкий сон, умеренно глубокий сон, глубокий сон. Стадии 3 и 4 носят общее название – дельта-сон, что связано с наличием на ЭЭГ специфических дельта-волн.

Дельта-ритм – ритм самой низкой частоты. Характерен для здоровых детей до года, иногда до 6 лет. У взрослых дельта-ритм появляется во время глубокого естественного сна, общего наркоза или комы. В состоянии бодрствования данный ритм возникает при регистрации ЭЭГ от участков мозга, граничащих с патологическими очагами и опухолями [9].

2.4 Частотные-амплитудные характеристики сигнала

Выделяют следующие характеристики ЭЭГ бодрствующего человека (таблица 6).

Таблица 6 – ЭЭГ бодрствующего человека

Название	Частота, Гц	Амплитуда, мкВ	Комментарий	Локализация
Медленный альфа ритм	2,5-6	50 – 80	Редко встречается. Близкие к синусоидальным медленно-частотные волны. Имеет все остальные физиологические и топографические характеристики альфа-ритма. Если не связан с патологией, то является пограничным между патологией и здоровым состоянием	Регистрируется в затылочных отделах при закрытых глазах
Быстрый альфа-ритм	14-18	Около 50	У части здоровых обследуемых вместо Альфа-ритма регистрируют в затылочной части волны частотой 14 - 18Гц и амплитудой ~50мкВ	В затылочной части
Альфа-ритм	8-13	До 100	Регистрируется у 85% здоровых взрослых. Наибольшая амплитуда в состоянии спокойного расслабленного бодрствования при закрытых глазах	Лучше всего ритм выражен в затылочных отделах
Бета-ритм	14-40	До 25	В норме выражен весьма слабо и имеет амплитуду от 5 до 15мкВ. Как правило связан с соматическими сенсорными и двигательными корковыми механизмами. Даёт реакцию активации при угасании на тактильную и двигательную стимуляцию	Лучше всего регистрируется в области центральных извилин, однако также распространяется на задние центральные и лобные извилины.
Мю-ритм	8-13	До 50	Частота амплитуда до не имеет особого клинического значения. отличается от альфа-ритма физиологическими свойствами и топографией	У 5-15% наблюдается в передней области. иногда при двигательной активации

Также при бодрствовании выделяют патологические активности, представленные в таблице 7.

Таблица 7 – Патологические активности при бодрствовании

Название активности	Частота, Гц	Амплитуда, мкВ	Комментарий
Тета активность	4 – 8	Более 40	При некоторых патологиях до 300 мкВ и более
Дельта активность	0,5 – 3,5	До 300	Возникает при регистрации ЭЭГ от участков мозга, граничащих с патологическими очагами и опухолями [9]

2.5 Метод диагностики

Метод диагностики заключается в выделении аномального сигнала для каждого конкретного электрода в соответствии с отведением, указанным на рисунке 7 и в таблице 8.

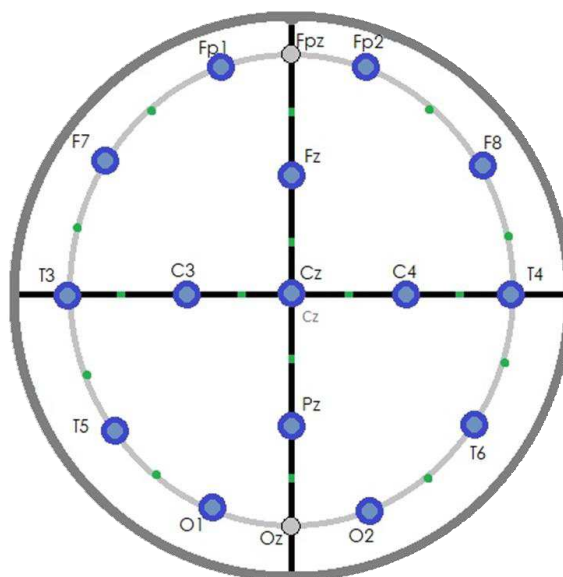


Рисунок 7 – Обозначение расположения датчиков на поверхности кожи головы, F – фронтальная часть головы, С – центральная, Т – височная, Р – теменная, О – затылочная. Референтный электрод не представлен на рисунке, он крепится к мочке уха

Таблица 8 – Соответствие датчиков точкам расположения по системе «10-20»

Электрод	Точка	Электрод	Точка
0	Ref	8	C3
1	Fp1	9	Cz
2	Fp2	10	C4
3	F7	11	T5
4	Fz	12	Pz
5	F8	13	T6
6	T3	14	O1
7	T4	15	O2

Под аномальным сигналом понимается обнаружение:

- частот в спектре сигнала превышающих пороговое значение;
- частот, появление которых может являться признаком патологической зоны. Например, тета и дельта активности в состоянии бодрствования, за исключением активностей со значением амплитуды менее 40 мкВ и локализацией в затылочной зоне;
- частот, находящихся в несвойственной им зоне. Например, альфа-волны в лобных или бета-волны вне лобных долей.

2.6 Выводы

В результате анализа предметной области были сформулированы четкие требования к разрабатываемому программному обеспечению, было решено, что целью данного проекта является клиентское приложение в удобном для пользователя виде для основных операционных мобильных систем, а также программа на микроконтроллер для регистрации и отправки данных на смартфон. Выделен перечень функций, отсутствующих в аналогичных носимых устройствах. Изучен метод диагностики электроэнцефалограммы.

3 Описание системы

3.1 Описание структурной схемы

Структурная схема представлена на рисунке 8:

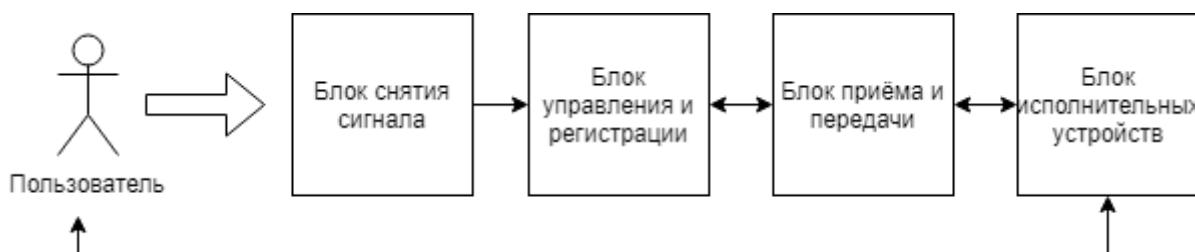


Рисунок 8 – Структурная схема переносного устройства регистрации ЭЭГ

3.1.1 «Блок снятия сигнала»

Каждый датчик и блок усиления представляет собой «Блок снятия сигнала». Данный блок представляет собой аналоговую схему защиты от помех и усиления сигнала.

3.1.2 «Блок управления и регистрации»

В качестве блока управления и регистрации используется микроконтроллер Arduino Nano. «Блок управления и регистрации» подает управляющие сигналы для начала записи сигналов, контролирует процесс записи, принимает управляющие сигналы от bluetooth модуля HC-05 и формирует для него ответ. Запускает процесс передачи данных.

Также к «Блоку управления и регистрации» относится кнопка для запуска регистрации ЭЭГ в автономном режиме (об этом подробнее в параграфе 3.2.1) и аналоговые ключи для поочерёдной регистрации сигналов с электродов.

3.1.3 «Блок приёма и передачи»

В качестве блока приёма и передачи выступает bluetooth модуль HC-05. bluetooth модуль получает информацию от микроконтроллера и выполняет все необходимые действия, для передачи информации на мобильный телефон. принимает управляющие сигналы от мобильного телефона и передаёт их на микроконтроллер.

3.1.4 «Блок исполнительных устройств»

«Блок исполнительный устройств» представляет собой мобильные телефоны, имеющие bluetooth передатчик.

Мобильный телефон принимает, обрабатывает и представляет в удобном пользователю виде полученную от bluetooth модуля информацию. Передаёт полученные от пользователя команды на микроконтроллер bluetooth модуль. Выдаёт пользователю результат диагностики.

3.2 Режимы работы системы

Система работает в двух режимах: автономный и управляемый. Для запуска регистрации ЭЭГ в автономном режиме необходимо нажать на кнопку, расположенную на носимом устройстве. Для запуска управляемого режима пользователю необходимо подать сигнал с «Блока исполнительных устройств» через мобильное приложение.

3.2.1 Автономный режим работы

В автономном режиме работы от кнопки поступает команда начала оцифровки и передачи данных. При повторном нажатии, считывание аналоговых сигналов завершается и устройство работает находится в режиме ожидания команд. Микроконтроллер поочерёдно оцифровывает полученные на вход

аналоговые сигналы и транслирует их через bluetooth модуль. Каждое значение при передаче маркируется входом, с которого было получено. Сопряженные устройства могут подключиться к bluetooth модулю и принять данные.

3.2.2 Управляемый режим работы

Управляемым режимом работы называется регистрация сигналов ЭЭГ, когда команда начала записи отдаётся пользователем через мобильный телефон. Данная ситуация возможна исключительно при наличии bluetooth соединения и сопряжения устройств. Пользователь отдаёт команду, она передаётся на bluetooth модуль, который, в свою очередь, распознаёт команду и передаёт её на микроконтроллер. Микроконтроллер запускает регистрацию, оцифровку и передачу сигналов на мобильный телефон пользователя, где данные подвергаются анализу, диагностике и визуализации. При получении команды остановки, микроконтроллер отключает работу таймера и данные перестают регистрироваться, оцифровываться и передаваться.

4 Разработка программного обеспечения

4.1 Разработка программы для микроконтроллера

4.1.1 Среда разработки для Arduino

Для написания и загрузки программ в Arduino необходимо установить программу для программирования, как Arduino IDE. Arduino IDE – программное обеспечение для пользователей, позволяющее писать свои программы (скетчи) для платформы Arduino. Эта платформа в первую очередь ориентируется на конструкторов-любителей, которые применяют Arduino для построения простых систем автоматизации и робототехники.

Язык программирования Arduino является стандартным C++ (используется компилятор AVR-GCC) с некоторыми особенностями, облегчающими написание программ.

Преимущества Arduino IDE:

- доступность;
- удобный для использования и понимания интерфейс;
- программа совместима со всеми версиями ОС Windows;
- наличие необходимых для работы инструментов;
- встроенный набор примеров программ;
- функции сохранения, экспорта, проверки, поиска, замены скетчей.

4.1.2 Алгоритм работы программы

Принцип обмена данными между устройством и ПЭВМ: программа микроконтроллера позволяет с помощью платы Arduino Nano реализовывать переключение аналоговых ключей, пропуская попеременно на входы A0 – A7 лишь 8 из 16 сигналов для их последующей оцифровки. Этот способ позволяет выбирать какие восемь из шестнадцати аналоговых сигналов будут оцифровываться в текущий момент времени. После, оцифрованные значения

сигналов вычитаются для получения дифференциальной разницы. Далее формируется сообщение, которое будет отправлено в виде последовательности байтов по физическому протоколу передачи данных UART на bluetooth модуль HC-05. HC-05 отправляет полученное сообщение далее по радиосигналу на сопряженное устройство, на котором с помощью проектируемого программного обеспечения будет производиться прием и обработка данных с последующим выводом их в виде графика. Также программа опрашивает вход от датчика касания, ожидая управляющего сигнала. Программа представлена в приложении А. На рисунке 9 представлены алгоритмы программы для Arduino, основной цикл и обработка прерывания по таймеру.

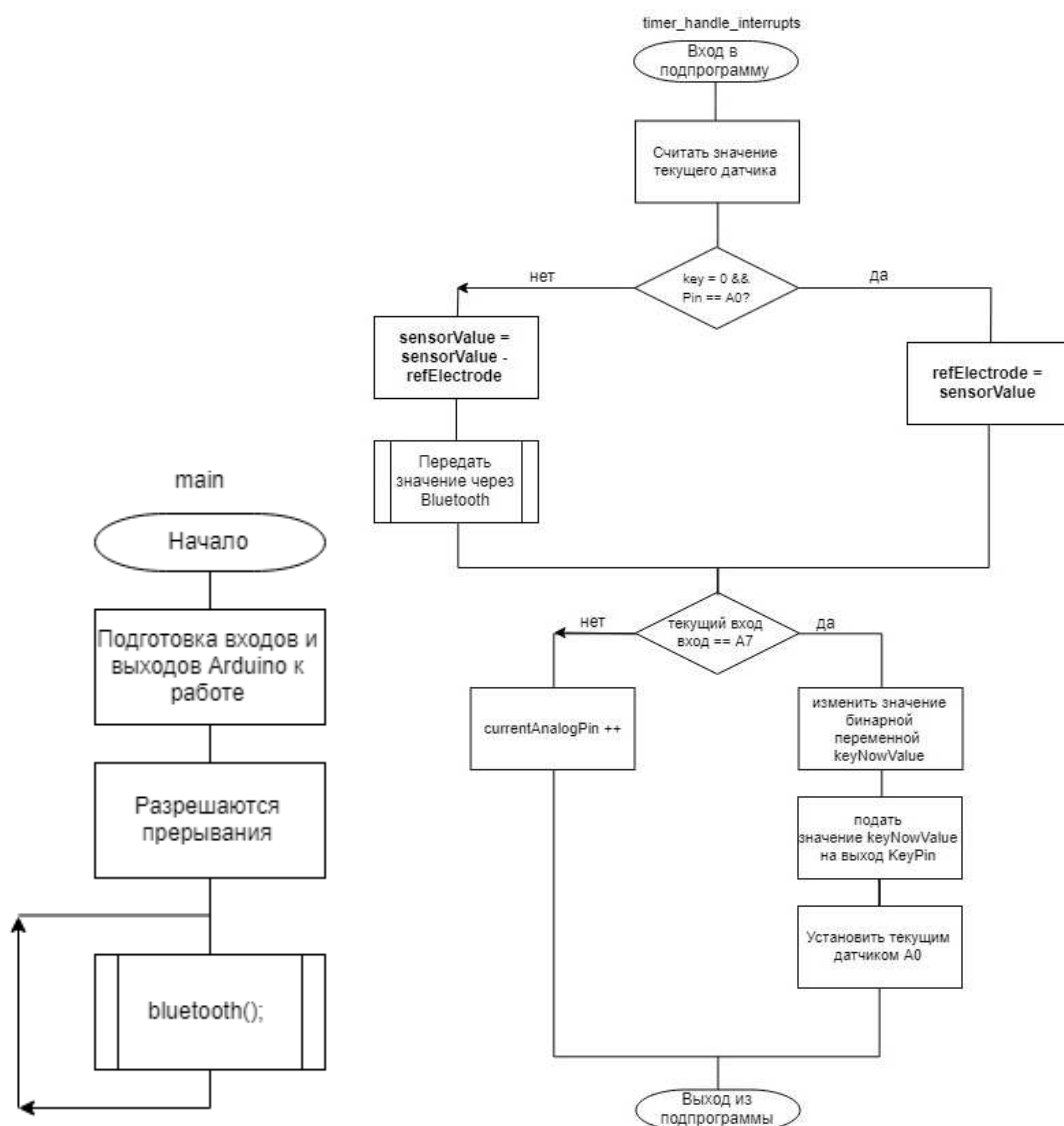


Рисунок 9 – Алгоритмы программы для Arduino

4.2 Разработка программы для мобильного приложения

4.2.1 Анализ существующих средств разработки

4.2.1.1 Нативная среда разработки

Native (с англ. Native – "родной") приложения – это те программные продукты, которые разрабатывались под конкретную операционную систему. Например, если программа была написана под известную многим операционную систему Android, то она будет работать только на ней. У каждой операционной системы есть свой набор основных процедур (API), с помощью которого она осуществляет взаимодействие с разнообразными прикладными программами [11]. API (application programming interface) – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах [11]. Нативные приложения разрабатываются с конкретным API, а следовательно, с конкретной операционной системой.

Преимущества нативного мобильного приложения:

- позволит поддерживать высокую производительность;
- позволит обрабатывать большое количество данных на стороне клиента;
- гибкость в реализации;
- нативная разработка может использовать все возможности мобильной операционной системы;
- экономичность расходования аккумулятора.

Одним из главных недостатков нативного мобильного приложения можно выделить трудоемкость реализации проектов что, влияет на конечную стоимость продукта.

4.2.1.2 Веб приложение

Веб-приложение – клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера, а за сервер отвечает веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами [12].

Преимущества веб-приложений:

- простая установка веб-приложений. Для использования приложения необходимо устройство с установленным браузером и соединение с интернетом;
- обновление веб-приложений проще. Для того чтобы совершить обновление веб-приложения, его необходимо обновить только на сервере, и все пользователи смогут работать с новой версией;
- веб-приложения универсальны и практичны для конечного пользователя. Достаточно установить веб-приложение на сервер, работающий под любой современной ОС, и можно пользоваться им через интернет на любом компьютере или мобильном устройстве;
- веб-приложения облегчают организацию хранения данных.

Недостатки веб-приложений:

- для работы с веб-приложением необходим постоянный доступ к интернету;
- необходимо содержать сервер;
- качество работы приложения определяется скоростью интернет-соединения.

4.2.1.3 Гибридная среда разработки

Гибридная среда разработки совмещает в себе свойства нативной разработки и веб-приложений. От веб-приложений здесь представлены кроссплатформенность и лёгкость поддержки, но при этом с помощью специализированных средств разработки полученное программное обеспечение представляет собой полноценное клиентское приложение. Преимущества гибридных мобильных приложений:

- кроссплатформенность. Код пишется один раз как для веб приложения, после чего может быть преобразован в приложение под любую популярную операционную систему;
- лёгкость поддержки. Добавление нового функционала происходит единожды сразу для всех операционных систем;
- независимость работы от наличия интернет-соединения.

4.2.2 Среда разработки Apache Cordova

Apache Cordova – это платформа разработки мобильных приложений с открытым исходным кодом [13]. Она позволяет использовать стандартные веб-технологии, такие как HTML5, CSS3 и JavaScript для кроссплатформенной разработки, избегая родного языка разработки для каждой из мобильных платформ. Приложения выполняются внутри обертки, нацеленной на каждую платформу, и полагаются на стандартные API для доступа к датчикам устройства, данным и состоянию сети. Принцип работы Apache Cordova представлен на рисунке 10.

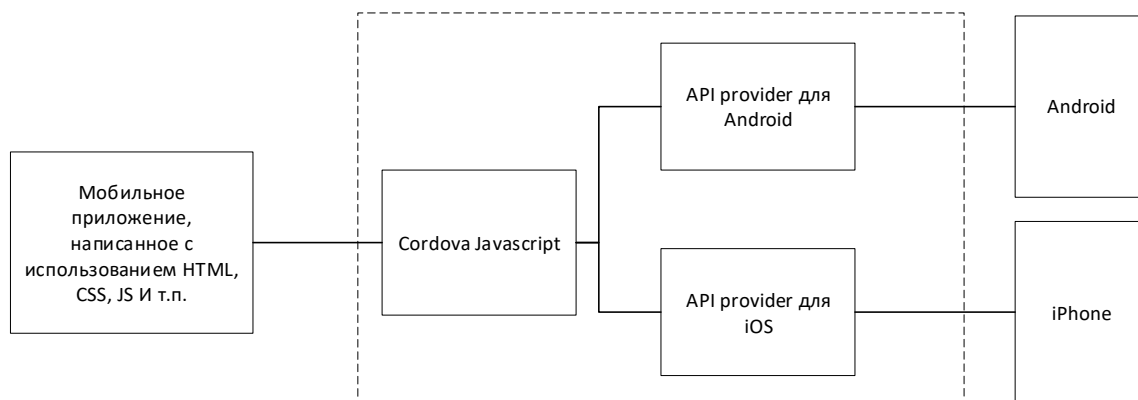


Рисунок 10 – Принцип работы Apache Cordova

Именно кроссплатформенность написанных с использованием Apache Cordova приложений является основным фактором в пользу использования при написании клиентской части приложения.

4.2.3 Ionic Framework

Ionic – это полноценный SDK с открытым исходным кодом для разработки мобильных приложений [14]. Оригинальная версия была выпущена в 2013 году и построена на AngularJS и Apache Cordova. Более поздние версии, известные как «Ionic 3» (или просто Ionic), построены на Angular. Ionic предоставляет инструменты и услуги для разработки гибридных мобильных приложений с использованием веб-технологий, таких как CSS, HTML5 и Sass.

Модель работы Ionic Framework представлена на рисунке 11.

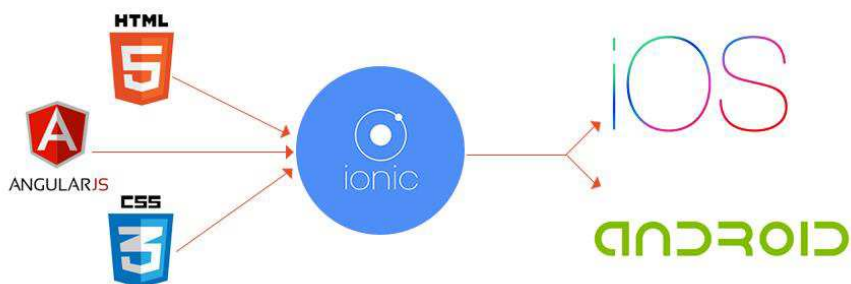


Рисунок 11 – Модель работы Ionic Framework

Широкий функционал встроенных средств разработки и унаследованная от Cordova кроссплатформенность вместе с бесплатностью являются основными аргументами в пользу его использования в данной работе.

4.2.4 TypeScript

TypeScript [15] – язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript.

TypeScript отличается от JavaScript возможностью явного статического назначения типов, поддержкой использования полноценных классов (как в традиционных объектно-ориентированных языках), а также поддержкой подключения модулей, что призвано повысить скорость разработки, облегчить читаемость, рефакторинг и повторное использование кода, помочь осуществлять поиск ошибок на этапе разработки и компиляции, и, возможно, ускорить выполнение программ.

Компилятор TypeScript называется tsc, написан на языке TypeScript и может быть скомпилирован в стандартный JavaScript, и может быть запущен на любом движке JavaScript, например, в браузере. Компилятор идет вместе с сервером сценариев, который может запускать компилятор. Также он доступен в виде пакета для node.js, который использует node.js в качестве сервера.

Язык широко поддерживается множеством IDE, удобен в использовании и является перспективным и с использованием Apache Cordova может использоваться как инструмент для разработки как под Android, так и под iOS.

4.2.5 Модуль распознавания частотной характеристики сигнала

4.2.5.1 Алгоритм распознавания частоты сигнала

Алгоритм распознавания частоты сигнала основывается на спектральном анализе получаемых данных.

Спектральный анализ заключается в определении частотного состава сигнала и оценивании его спектральных характеристик, являющихся функциями частоты [10]. Различают следующие методы спектрального анализа: фильтровые (полосовой анализ), бесфильтровые (на основе дискретных преобразований Фурье, Уолша, Хаара, Хартли, вейвлет-анализ), параметрические (на основе параметрических моделей случайных сигналов), методы текущего, скользящего и скачущего анализа, последовательного и параллельного, одноканального и многоканального, выполняемого в реальном времени [10].

Особенности классического спектрального анализа на основе дискретного преобразования Фурье (ДПФ) связаны с оценкой спектра сигнала по его реализациям конечной длины, т. е. на конечном интервале наблюдения. При этом предполагается, что за пределами этого интервала сигнал является периодическим продолжением считанной реализации с периодом, равным или большим ее длительности. Широкое применение анализаторов спектра на основе ДПФ обусловлено наличием высокоэффективных вычислительных алгоритмов быстрого преобразования Фурье (БПФ) [10]. Преобразование Фурье является математической основой спектрального анализа сигнала.

Алгоритмы быстрого преобразования Фурье (БПФ) — это способы быстрого вычисления ДПФ $X(jk) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)kn}$, устраняющие свойственную ДПФ вычислительную избыточность. Они основываются на свойствах комплексной экспоненты $e^{-j(2\pi/N)kn}$, для удобства обозначаемой W_N^{kn} ($W_N = e^{-j(2\pi/N)kn}$) ее симметрии $W_N^{(N-k)n} = W_N^{(N-n)k} = (W_N^{kn})^*$ и периодичности $W_N^{(k+N)(n+N)} = W_N^{kn}$ с периодом, равным длине обрабатываемой реализации сигнала N (числу точек БПФ). В соответствии с последним свойством экспоненте $W_N^{pkn} = W_{N/p}^{kn}$ отвечает период N/p , где p — целые числа, на которые делится N . Использование данных свойств в алгоритмах БПФ исключает большое число повторяющихся при вычислении ДПФ операций. Общий принцип БПФ заключается в разбиении ДПФ исходной последовательности на ДПФ под

последовательностей меньшей длины, вплоть до минимально возможной (равной основанию БПФ), через которые и вычисляется ДПФ исходной последовательности. Разбиение означает прореживание последовательностей во временной или в частотной области. В связи с этим различают БПФ с прореживанием по времени и БПФ с прореживанием по частоте. В отличие от ДПФ, БПФ может вычисляться только по определенному числу точек N , соответствующему целой степени его основания m : $N = mL$, где L – это число этапов прореживания: $L = \log_m N$. Чаще всего применяют БПФ по основанию 2 [10].

4.2.5.2 Программная реализация алгоритма

Для реализации БПФ существуют готовые библиотеки и плагины, такие как «Fast Foirier Transform». Библиотека «Fast Foirier Transform» («FFT») принимает на вход массив значений, также необходимо выставить частоту дискретизации, в нашем случае 250 Гц. На выходе получается массив из структур, в каждой из которых хранятся амплитуда и частота полученного спектра. Пример использования FFT приведён в листинге 1.

Листинг 1 – Пример использования Fast Foirier Transform

```
export function FFT(data: number[]): FFTResult[] {
  const phasors = fft.fft(data);
  const f = fft.util.fftFreq(phasors, 250);
  const m = fft.util.fftMag(phasors);
  return f.map((fr, i) => {
    return {
      f: fr,
      m: m[i]
    };
  });
}
```

4.2.6 Модуль визуализации сигнала D3.js

Для визуализации и создания графиков по имеющимся данным существует множество библиотек, которые выдают по результатам работы рисунки в разных

форматах и для любых целей. Визуализация частотно-амплитудной, частотной и амплитудной по отдельности характеристик не является исключением.

Модуль визуализации получает уже готовые данные от «Модуля анализа» таким образом, визуализировать их не составляет труда. Одной из таких библиотек является D3.js.

D3.js (или просто D3) — это JavaScript-библиотека для обработки и визуализации данных с невероятно огромными возможностями. Примеры графиков, созданных с помощью библиотеки D3 изображены на рисунке 12.

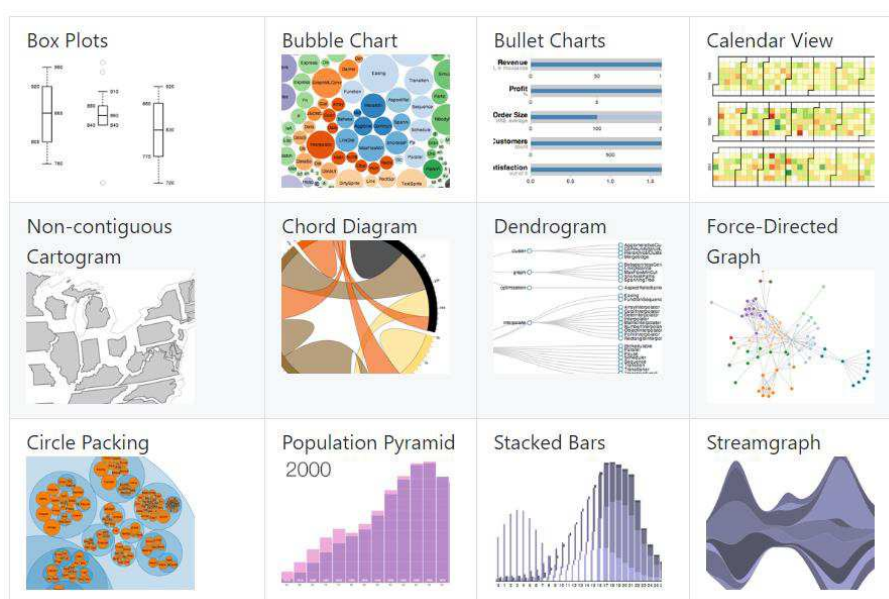


Рисунок 12 – Примеры графиков, созданных с помощью D3.js библиотеки [16]

4.2.7 Алгоритм работы мобильного приложения

Диаграмма прецедентов представлена на рисунке 13.

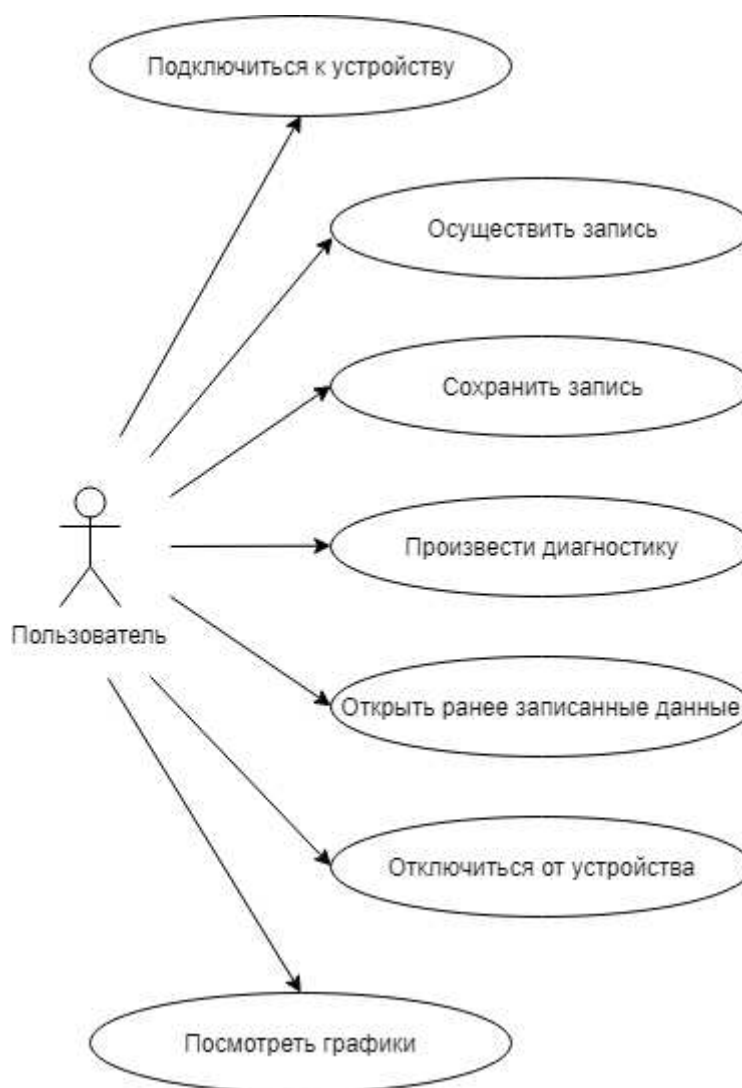


Рисунок 13 – Диаграмма прецедентов

Текстовое описание прецедентов:

Название прецедента: «Подключиться к устройству».

Цель сценария: установить bluetooth соединение с микроконтроллером.

Предусловия: пользователь находится на главной странице приложения, нажимает кнопку «Показать список устройств».

Основной сценарий:

А. Пользователю выводится список сопряженных с телефоном устройств и для каждого по кнопке «Подключиться».

Б. Пользователь выбирает с каким устройством установить соединение, нажимает «Подключиться».

В. При установлении соединения выводится информация о том, что соединение установлено, а также появляется кнопка для разрыва соединения.

Постусловия: соединение установлено, мобильное приложение способно передавать команды на микроконтроллер, а также ожидает поступления данных и готово к их обработке.

Условие ввода в действие альтернативных сценариев:

Условие 1. Соединение неожиданно разорвано без нажатия кнопки «Отключиться».

А. Мобильное приложение больше не получает данные и не может отдавать команды на микроконтроллер.

Б. Пользователь может возобновить соединение, повторив действия по подключению.

Название прецедента: «Осуществить запись».

Цель сценария: произвести регистрацию сигналов, поступающих с датчиков на микроконтроллер, осуществить передачу по bluetooth модулю данных в мобильное приложение, произвести накопление данных.

Предусловия: Пользователь нажал кнопку «Старт».

Основной сценарий:

А. Запускается таймер, видимый пользователю.

Б. На микроконтроллер передаётся команда начала записи. Запускается таймер на микроконтроллере для срабатывания прерываний по таймеру.

В. Амплитуды с аналоговых входов считываются с заранее определённой частотой, оцифровываются, сравниваются с нулевым входом и результат маркируется номером входа и передаётся через bluetooth соединение.

Г. Данные принимаются на стороне мобильного устройства и накапливаются в сервис пакетами по 16 значений для дальнейшей обработки.

Д. Производится подсчет количества пакетов.

Е. Ожидается действие от пользователя, а именно: нажатие кнопки «Стоп».

Ж. При получении команды «Стоп» мобильное приложение отправляет команду остановки записи и передачи на микроконтроллер.

Постусловия: в сервисе хранится массив данных в определённом порядке для дальнейшей обработки.

Условие ввода в действие альтернативных сценариев:

Условие 1. Остановки записи не произошло.

А. При превышении определённого пользователем лимита по времени, запись останавливается, регистрация прекращается. Пользователю выдаётся сообщение о превышении лимита длительности записи.

Название прецедента: «Сохранить запись».

Цель сценария: сохранить в памяти данные, полученные от микроконтроллера.

Предусловия: пользователь подключился к устройству, нажал «Старт», после чего нажал «Стоп». Мобильное приложение предложило пользователю сохранить данные, пользователь согласился.

Основной сценарий:

А. Данные сохраняются в памяти телефона в том виде, в котором были получены и накоплены. Название генерируется из текущих даты и времени.

Б. Пользователю выдаётся сообщение о том, что данные сохранены.

Постусловия: данные сохранены и доступны для загрузки.

Условие ввода в действие альтернативных сценариев:

Условие 1. Память мобильного телефона заполнена.

А. Пользователю выдается сообщение о том, что память телефона переполнена и запись невозможна.

Б. При освобождении памяти пользователем возможна повторная попытка.

Название прецедента: «Произвести диагностику».

Цель сценария: произвести обработку полученных данных и диагностику состояния пользователя исходя из полученных данных, отобразить графики, вывести результат диагностики пользователю.

Предусловия: пользователь нажал кнопку «СТОП».

Основной сценарий:

А. Полученные в мобильном приложении данные разбиваются на 16 массивов.

Б. Вычисляется ближайшая меньшая степень двойки от количества пакетов, такое количество данных обрабатывается FFT библиотекой.

В. Результат полученный FFT библиотекой передаётся в модуль отрисовки графиков. Полученные графики для каждого массива данных выводятся на вкладках «График» и «Спектр».

Г. Среди полученных спектров ищем нежелательные частоты, значение амплитуд которых превышает норму. При обнаружении таких значений, пользователю выводится предупреждение и рекомендация.

Постусловия: пользователь получает графики, визуализирующие амплитуды и спектры полученных значений, результат диагностики и рекомендации.

Условия ввода в действие альтернативных сценариев:

Условие 1. Нежелательных значений не найдено.

А. Пользователь получает информацию о том, что нежелательных значений не выявлено.

Условие 2. Данных недостаточно для обработки.

А. Пользователю выводится сообщение о том, что данных для обработки недостаточно и предлагается запустить запись с начала.

Название прецедента: «Открыть ранее записанные данные».

Цель сценария: загрузить данные из памяти устройства при их наличии.

Предусловия: пользователь открыл жестом слева направо всплывающее меню и выбрал в нём пункт «Загрузить».

Основной сценарий:

А. Все доступные для загрузки файлы в памяти приложения отображаются списком. Пользователь выбирает необходимый ему файл для загрузки.

Б. Из памяти телефона загружается файл, в котором хранится длительность записи, дата и время записи, а также список амплитуд в заранее известном порядке, готовый к обработке алгоритмами визуализации и диагностики.

Постусловия: данные загружаются, производится обработка, визуализация, диагностика, результат диагностики предоставляется пользователю.

Условие ввода в действие альтернативных сценариев:

Условие 1. Нет заранее записанных файлов в памяти телефона.

А. Приложение выдает пустой список и сообщение, что нет данных для загрузки.

Название прецедента: «Отключиться от устройства».

Цель сценария: остановить получение данных мобильным телефоном.

Предусловия: пользователь установил соединение с микроконтроллером. Нажал кнопку «Отключиться».

Основной сценарий:

А. Мобильное приложение перестаёт ожидать данные от микроконтроллера. Соединение разрывается.

Постусловия: bluetooth соединение разорвано.

Условие ввода в действие альтернативных сценариев:

Условие 1. Соединение разорвалось без участия пользователя.

А. Пользователь не имеет возможности «Отключиться» от устройства, так как не видит соответствующей кнопки.

Название прецедента: «Перейти на вкладку графиков/спектров».

Цель сценария: перейти на другое окно приложения.

Предусловия: Пользователь находится в любом окне приложения из трёх: «Домой», «График», «Спектр».

Основной сценарий:

А. Пользователь нажимает на необходимую кнопку в меню навигации внизу экрана.

Б. Текущее окно переключается на выбранное.

Постусловия: открывается окно в зависимости от пожелания пользователя.

Основной функционал приложения заключается в обработке полученных данных. На рисунке 14 представлен алгоритм работы приложения при нажатии кнопки «СТАРТ».

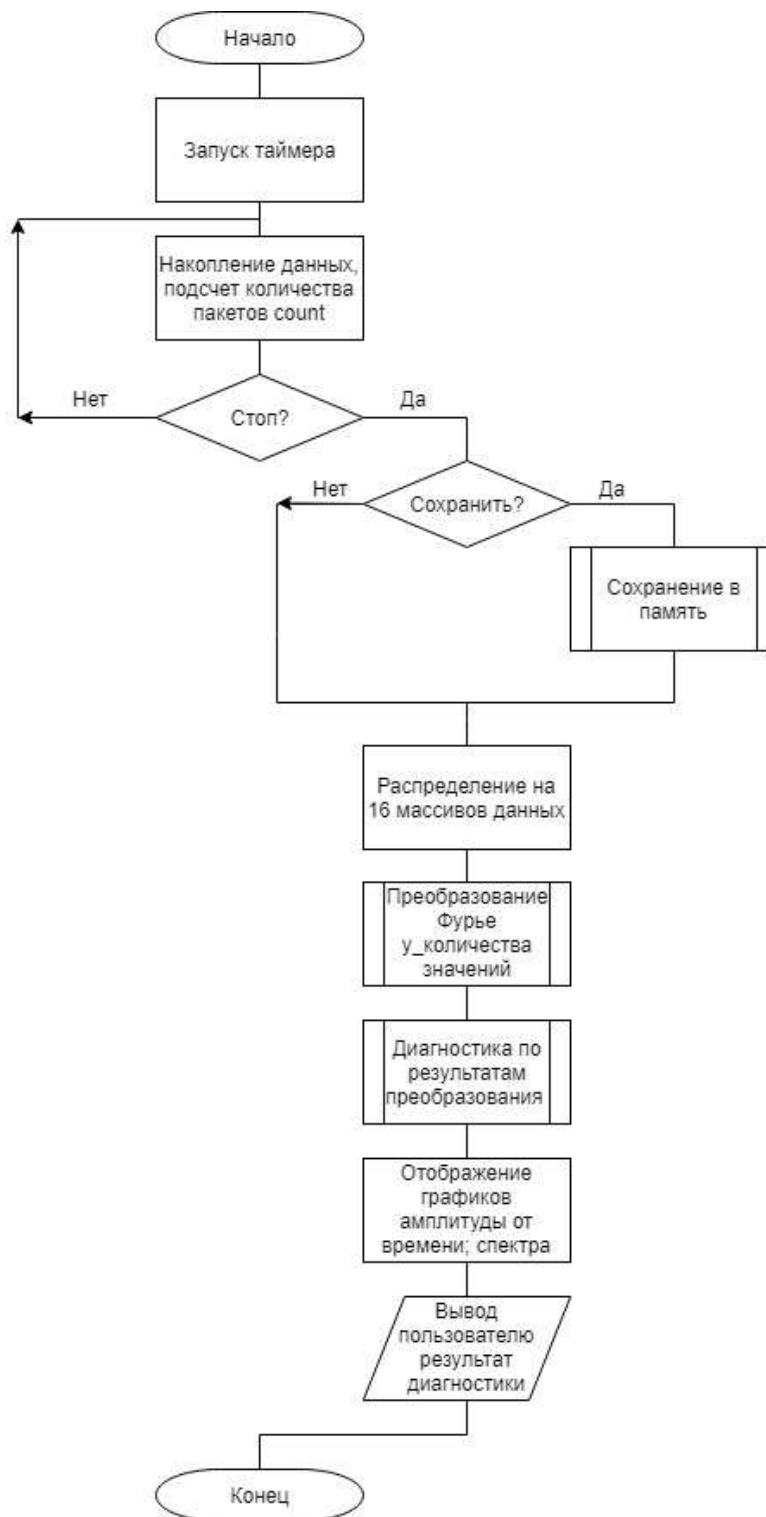


Рисунок 14 – Алгоритм обработки и визуализации данных в мобильном приложении.

4.2.8 Скриншоты работы мобильного приложения

Скриншоты мобильного приложения представлены на рисунках 15, 16.



Рисунок 15 – Скриншоты мобильного приложения

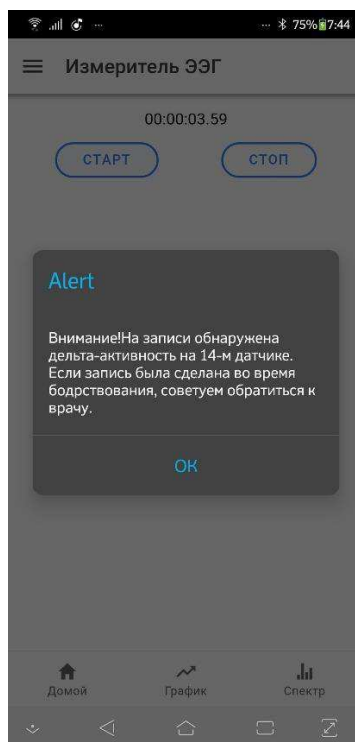


Рисунок 16 – Скриншот мобильного приложения. Результат анализа спектра.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы по разработке и проектированию мобильной системы регистрации электрических потенциалов головного мозга были решены следующие задачи:

- изучены основы регистрации и обработки электроэнцефалограммы;
- разработана структурная схема системы;
- разработаны алгоритмы работы программного обеспечения;
- изучены основы гибридной кроссплатформенной разработки;
- составлены и протестированы на практике программы взаимодействия Arduino и мобильного телефона;
- реализован частотный анализ сигнала и визуализация сигнала;
- реализовано обнаружение возможно патологических состояний в соответствии с методом, указанным в пункте 2.5.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Стандарт организации «Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности» [Электронный ресурс] : Номер документа: СТО 4.2-07-2014. – Режим доступа: <http://about.sfu-kras.ru/node/8127>.
2. Xie, L., Kang, H., Xu, Q., Chen, M. J., Liao, Y., Thiagarajan, M., ... Nedergaard, M. (2013). Sleep Drives Metabolite Clearance from the Adult Brain. *Science*, 342(6156), 373–377. doi:10.1126/science.1241224.
3. Левин Я.И., Ковров Г.В. Нарушения сна и их фармакологическая коррекция у неврологических больных. *Психиат и психофармакотер* 2003; 5: 2: 1 – 15.
4. Neurobotics «Кортекс» Руководство по эксплуатации» Версия 1.2, Москва, Зеленоград, 2018.
5. Международная система размещения электродов «10 – 20» [Электронный ресурс] : Материал из Википедии – свободной энциклопедии : Версия 53307763, сохранённая в 03:34 UTC 13 марта 2013 / Авторы Википедии // Википедия, свободная энциклопедия. – Электрон. дан. – Сан-Франциско: Фонд Викимедиа, 2013. – Режим доступа: <https://ru.wikipedia.org/?oldid=53307763>.
6. Визуальная ЭЭГ [Электронный ресурс] : Американское Общество Клинической Нейрофизиологии Guideline 1: Минимальные технические требования (MTR) при записи клинической ЭЭГ. – Режим доступа: http://eeg-online.ru/standards/rec_mtr_acns.htm.
7. Электроэнцефалография. Методы исследования нервной системы. ЭЭГ [Электронный ресурс] : Electroencephalography. – Режим доступа: <https://www.youtube.com/watch?v=8SE26hB9y7Y>.
8. Фазы и стадии сна [Электронный ресурс] : Experimental Psychic. – Режим доступа: <https://experimental-psychic.ru/fazy-stadii-sna>.

9. Электроэнцефалограмма головного мозга: что показывает, расшифровка результатов [Электронный ресурс] : ОкейДок. – Режим доступа: <https://okeydoc.ru/elektroencefalogramma-golovnogo-mozga>.

10. Цифровая обработка сигналов. Версия 1.0 [Электронный ресурс] : курс лекций / А. С. Глинченко. – Электрон. дан. (3 Мб). – Красноярск : ИПК СФУ, 2008. – (Цифровая обработка сигналов : УМКД № 50-2007 / рук. творч. коллектива А. С. Глинченко). – Режим доступа: http://files.lib.sfu-kras.ru/ebibl/umkd/50/u_lectures.pdf.

11. API [Электронный ресурс] : Материал из Википедии – свободной энциклопедии : Версия 98552928, сохранённая в 06:13 UTC 10 марта 2019 / Авторы Википедии // Википедия, свободная энциклопедия. – Электрон. дан. – Сан-Франциско: Фонд Викимедиа, 2019. – Режим доступа: <https://ru.wikipedia.org/?oldid=98552928>.

12. Веб-приложение [Электронный ресурс] : Материал из Википедии – свободной энциклопедии : Версия 94938607, сохранённая в 08:39 UTC 7 сентября 2018 / Авторы Википедии // Википедия, свободная энциклопедия. – Электрон. дан. – Сан-Франциско: Фонд Викимедиа, 2018. – Режим доступа: <https://ru.wikipedia.org/?oldid=94938607>.

13. Apache Cordova [Электронный ресурс] : The Apache Software Foundation. – Режим доступа: <https://cordova.apache.org>.

14. Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular [Электронный ресурс] : Ionic. – Режим доступа: <https://ionicframework.com>.

15. TypeScript – JavaScript that scales [Электронный ресурс] : 2012-2019 Microsoft. – Режим доступа: <https://www.typescriptlang.org>.

16. D3 – wiki – Galery [Электронный ресурс] : Github.com. – Режим доступа: <https://github.com/d3/d3/wiki/Gallery>.

СПИСОК СОКРАЩЕНИЙ

Гц – герц;

мкВ – микровольт;

ОС – операционная система;

ПО – программное обеспечение;

ЦНС – центральная нервная система;

ЭЭГ – электроэнцефалограмма;

API – application programming interface.

ПРИЛОЖЕНИЕ А

Программное обеспечение для Arduino

```
#include <SoftwareSerial.h>
#include <Time.h>
#include "timer-api.h"
#define KeyPin 9
#define ButtonPin 2
#define Sensor0Pin A0
#define Sensor1Pin A1
#define Sensor2Pin A2
#define Sensor3Pin A3
#define Sensor4Pin A4
#define Sensor5Pin A5
#define Sensor6Pin A6
#define Sensor7Pin A7
// Назначение прерывания
int buttonInt = 0;
volatile bool currentMod = false; //false - выкл
volatile int keyNowValue = LOW;
volatile byte currentAnalogPin = A0;
volatile int sensorValue = 0; //значение амплитуды
volatile bool needLn = true;
volatile int refElectrode = 0;
int triggerPin = 7; //Trigger назначается пину 7
int echoPin = 8; //Echo назначается пину 8
int LED = 13; //пин встроенного светодиода
int info = 0; //переменная для информации, поступающей от Bluetooth модуля
int state = 0; //переменная для отображения состояния
int numbPinPrint=0;

void setup(){
  Serial.begin(9600);
  pinMode(triggerPin, OUTPUT); //определение пинов
  pinMode(echoPin, INPUT);
  pinMode(LED, OUTPUT); //определение пина для светодиода
  pinMode(KeyPin, OUTPUT);
  pinMode(ButtonPin, INPUT);
  pinMode(Sensor0Pin, INPUT);
  pinMode(Sensor1Pin, INPUT);
  pinMode(Sensor2Pin, INPUT);
  pinMode(Sensor3Pin, INPUT);
  pinMode(Sensor4Pin, INPUT);
  pinMode(Sensor5Pin, INPUT);
  pinMode(Sensor6Pin, INPUT);
  pinMode(Sensor7Pin, INPUT);
  attachInterrupt(buttonInt, swap, RISING);
  interrupts();
}
```

```

}
void loop() {
  bluetooth();
}

void bluetooth()
{
  if (Serial.available() > 0)
  {
    info = Serial.read(); //...и присваиваем ее переменной "info"
    state = 0;
    if (info == '1')
    {
      if (state == 0)
      {
        startRead();
        digitalWrite(LED, HIGH);
        state = 1;
      }
    }
    else if (info == '2')
    {
      if (state == 0)
      {
        stopRead();
        digitalWrite(LED, LOW);
        state = 1;
      }
    }
    else
    {
      if (state == 0)
      {
        stopRead();
        Serial.println("error command");
        state = 1;
      }
    }
  }
}

void swap()
{
  currentMod = !currentMod;
  if (currentMod == true) { startRead();}
  else { stopRead();}
}

void startRead()
{
  keyNowValue = LOW;
  currentAnalogPin = A0;
  currentMod = true;
  timer_init_ISR_20KHz(TIMER_DEFAULT);

```

```

}

void stopRead()
{
  currentMod = false;
  timer_stop_ISR(TIMER_DEFAULT);
}
void timer_handle_interrupts(int timer)
{
  // дополнительный множитель периода
  static int countTimer = 4;

  if(countTimer == 0)
  {
    sensorValue = analogRead(currentAnalogPin);
    if ((!keyNowValue)&&(currentAnalogPin == A0))
    {
      refElectrode = sensorValue;
    }else
    {
      sensorValue = sensorValue - refElectrode;
    }
    if (!keyNowValue) {numbPinPrint = currentAnalogPin - 14;} else {numbPinPrint =
currentAnalogPin - 6;}
    Serial.print(numbPinPrint);
    Serial.print(" ");
    Serial.print(sensorValue);
    Serial.print(" ");

    if (currentAnalogPin == A7)
    {
      keyNowValue = !keyNowValue;
      digitalWrite(KeyPin, keyNowValue);
      currentAnalogPin = A0;
      needLn = !needLn;
      if (needLn)
      {
        Serial.println();
      }
    }
    else
    {
      currentAnalogPin ++;
    }
    countTimer = 4;
  }else
  {
    countTimer--;
  }
}

```

ПРИЛОЖЕНИЕ Б

«home.page.ts»

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { BluetoothSerial } from '@ionic-native/bluetooth-serial/ngx';
import { BehaviorSubject, Subscription } from 'rxjs';
import { makeGraph } from '../lib/graph';
import { FFT } from '../lib/fft';
import { MainServiceService } from '../main-service.service';
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage implements OnInit {

  public result: any = "";
  public list: any[] = [];
  public connection: Subscription;

  public arduinoResponses$: BehaviorSubject<any[]> = new BehaviorSubject<any[]>([]);

  constructor(
    public mainService: MainServiceService,
    private bluetoothSerial: BluetoothSerial
  ) {}
  ngOnInit(): void {
    this.bluetoothSerial.subscribeRawData().subscribe((res) => {
      console.log(res);
    });
  }
  writeStart() {
    this.bluetoothSerial.write('1').then((result) => {
      console.log(result);
      this.result = JSON.stringify(result);
    }).catch(err => console.error(err));
  }
  writeStop() {
    this.bluetoothSerial.write('2').then((result) => {
      console.log(result);
      this.result = JSON.stringify(result);
    }).catch(err => console.error(err));
  }
  showDeviceList() {
    this.bluetoothSerial.list().then(result => {
      console.log(result);
      this.list = result;
    });
  }
}
```

```

}
connectToDevice(address) {
  const responses = [];
  this.connection = this.bluetoothSerial.connect(address).subscribe((result) => {
    console.log(result);
    this.result = JSON.stringify(result);
    this.bluetoothSerial.subscribe('\r\n').subscribe(data => {
      responses.push(data);
      this.mainService.arduinoResponses$.next(responses);
      console.log('data', data);
    });
  });
}
disconnect() {
  if (this.connection) {
    this.connection.unsubscribe();
    this.connection = null;
  }
}
}
}

```

Одна из функций диагностики в «spectre.page.ts»

```

GetDiagnose(array, numberOfDevice){
  let FFT = this.SliceArrayFFT(array);
  for(let i = 0; i < FFT.length; i++){
    let elem = FFT[i];
    if(elem.f > 0.5 && elem.f <= 3.5 && elem.m > 40){
      return "Внимание! На записи обнаружена дельта-активность на " +
        numberOfDevice.toString() + "-м датчике. Если запись была сделана во время бодрствования,
        советуем обратиться к врачу. " + "\n";
    }
    else if(elem.f > 3.5 && elem.f < 8 && elem.m > 40){
      return "Внимание! На записи обнаружена Тета-активность на " +
        numberOfDevice.toString() + "-м датчике. Если запись была сделана во время бодрствования,
        советуем обратиться к врачу. " + "\n";
    }
    else if(numberOfDevice >= 1 && numberOfDevice <= 7 && elem.f > 8 && elem.f <= 13 &&
    elem.m > 50 && elem.m < 100){
      return "Внимание! На записи обнаружена Альфа-активность на " +
        numberOfDevice.toString() + "-м датчике. Советуем обратиться к врачу. " + "\n";
    }
    else if(numberOfDevice >= 11 && numberOfDevice <= 15 && elem.f > 14 && elem.f < 70
    && elem.m >= 5 && elem.m <= 15){
      return "Внимание! На записи обнаружена Бета-активность на " +
        numberOfDevice.toString() + "-м датчике. Советуем обратиться к врачу. " + "\n";
    }
    else{ return ""; }
  }
}
}

```


Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой


О. В. Непомнящий

«28» 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Мобильная система регистрации электрической активности головного мозга
тема

Руководитель


подпись, дата

25.06.19 доцент, канд. техн. наук
должность, ученая степень

М. С. Медведев
инициалы, фамилия

Студент


подпись, дата

25.06.19

А. А. Темерова
инициалы, фамилия

Нормоконтролер


подпись, дата
27.06.19

доцент, канд. техн. наук
должность, ученая степень

В. И. Иванов
инициалы, фамилия

Красноярск 2019