



Московский государственный университет имени М.В.Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра автоматизации систем вычислительных комплексов

Бурдюгова Мария Витальевна

**Развитие системы медицинской телеметрии
в части сбора данных электрокардиограммы**

Курсовая работа

Научный руководитель:
к.ф.-м.н.

Бахмуrow Анатолий Геннадьевич

Научный консультант:

Александр Андреевич Любицкий

Москва, 2024

Аннотация

Развитие системы медицинской телеметрии
в части сбора данных электрокардиограммы

Бурдюгова Мария Витальевна

Курсовая работа нацелена на усовершенствование существующей клиентской части сервиса сбора и обработки медицинской телеметрии с устройств интернета медицинских вещей. В ходе работы рассмотрена возможность подключения устройства КР-2 (датчик электрокардиограммы), реализованы: работа с Bluetooth Classic (в дополнение к работе с Bluetooth LE), пользовательский интерфейс для отображения данных, полученных с датчика, их передача на сервер.

Abstract

Development of a medical telemetry system
by means of collecting electrocardiogram data

Abstract

Содержание

1	Введение	5
2	Постановка задачи	6
3	Описание устройства КР-2	7
3.1	Общие сведения	7
3.2	Формат команды	7
3.3	Протокол RFC 1055	8
3.4	Ответные сообщения устройства и обработка ошибок	10
3.5	Данные ЭКГ	10
4	Описание спецификации Bluetooth	12
4.1	Общие сведения	12
4.2	Архитектура и стек протоколов Bluetooth Classic	13
4.2.1	PHY	13
4.2.2	Link Manager Protocol	13
4.2.3	Logical Link Control and Adaptation Protocol	13
4.2.4	RFCOMM	13
4.2.5	SPP	14
4.3	Протокол RFCOMM	14
4.3.1	Общие сведения о Bluetooth Framework и RFCOMM	14
4.3.2	Bluetooth профили	15
4.3.3	Номер канала	15
5	Реализация приложения	17
5.1	Основные подходы и используемые технологии	17
5.2	Программная реализация	18
5.2.1	Графический интерфейс пользователя	18
5.2.2	Взаимодействие с кардиографом	18
5.2.3	Формирование MQTT пакетов из топиков и сообщений	18
5.2.4	Использование сопрограмм	20
6	Тестирование	21

7 Заключение	22
Список литературы	23

1 Введение

Современные технологии в области медицины стремительно развиваются, открывая новые возможности для диагностики и мониторинга состояния здоровья пациентов. Одной из таких инновационных областей является медицинская телеметрия, которая позволяет собирать и анализировать данные о состоянии здоровья пациента в реальном времени с помощью различных устройств интернета медицинских вещей (IoMT).

IoMT в домашних условиях может радикально изменить сферу здравоохранения. Основная идея заключается в том, что носимые устройства (различные датчики) могут постоянно отслеживать жизненно важные показатели пациента - сердечный ритм, артериальное давление, уровень кислорода в крови и другие параметры. Эта информация в режиме реального времени передается врачам, позволяя им дистанционно наблюдать за состоянием пациента, что снижает необходимость в частых визитах пациентов в больницу для рутинных медицинских осмотров.

Кроме того, дистанционный мониторинг позволяет врачам своевременно выявлять ухудшение состояния пациента и вовремя вмешиваться, помогая предотвратить обострение заболевания и повторные госпитализации. Это, в свою очередь, способствует оптимизации использования медицинских ресурсов и снижению расходов на здравоохранение.

Для пациентов IoMT также означает улучшение качества жизни, поскольку они могут получать медицинскую помощь, не выходя из дома, что особенно актуально для пожилых людей, людей с ограниченной мобильностью и хроническими заболеваниями.

На кафедре АСВК ведётся работа по созданию программного средства с открытым исходным кодом для сбора и обработки физиологических данных человека (проект IoMT-LVK). В 2023 году, в рамках соглашения о сотрудничестве с ООО «Нейрософт» была поставлена задача подключения прибора КР-2 этой компании. Интеграция устройства позволит расширить функциональность приложения и осуществлять сбор данных и анализ ЭКГ в условиях кафедры.

2 Постановка задачи

Целью данной работы является улучшение клиентской части платформы для сбора и обработки физиологических данных о человеке путем добавления поддержки устройства КР-2. Требуется реализовать взаимодействие прибора КР-2 с приложением на смартфоне Android в ходе регистрации ЭКГ.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Изучить принципы разработки мобильных приложений под Android OS, освоить инструменты и фреймворки;
2. Изучить порядок работы с устройствами на основе спецификации Bluetooth Classic;
3. Изучить протокол обмена с устройством КР-2, используя материалы от разработчика;
4. Спроектировать и выполнить доработку интерфейса пользователя приложения с учётом требований работы с КР-2;
5. Провести экспериментальное исследование корректной работы приложения.

3 Описание устройства КР-2

3.1 Общие сведения

Устройство КР-2 представляет собой двухканальный кардиорегистратор, который в процессе тренировки записывает данные ЭКГ, дыхания, а также темп движения и положение тела пациента. Полученные результаты передаются на смартфон с помощью заранее настроенного беспроводного канала Bluetooth Classic.

Кардиограф может работать в одном из следующих основных режимов:

1. Ожидание команды
2. Режим мониторинга ЭКГ

После включения прибор переходит в режим ожидания команды. В режиме измерения ЭКГ прибор передаёт пакеты с данными каналов ЭКГ, каналов акселерометра и дыхания.

3.2 Формат команды

Все команды устройству оформляются в виде пакета протокола. По байтам: версия протокола, код команды, байты данных, контрольная сумма (Рис.1).

Версия протокола	Код команды	Данные	...	Данные	CRC
------------------	-------------	--------	-----	--------	-----

Рис. 1: Формат команды

Длина пакета явно не записывается, она определяется на основе кода команды и данных. Данные могут отсутствовать, то есть минимальный размер пакета будет 3 байта. Команды с ошибкой в контрольной сумме игнорируются устройством.

Контрольная сумма рассчитывается по формуле (1):

$$CRC = 2 + \text{длина данных} + \text{версия протокола} + \text{код команды} + \text{сумма байтов данных} \quad (1)$$

Некоторые поддерживаемые прибором команды:

- ReadDeviceInformation — чтение информации об устройстве;
- ReadDeviceStatus — последняя выполненная команда;
- ReadComponentsStatus — получить заряд батареи и температуру прибора;
- MonitoringMode — перейти в режим мониторинга ЭКГ (или завершить мониторинг ЭКГ);
- PowerOff — выключение устройства.

Ниже приведена таблица с полем данных этих команд:

Символьное имя	Поле данных
ReadDeviceInformation	1 - номер версии встроенного ПО модуля блютуз, 2 - номер ревизии микросхемы АЦП
ReadDeviceStatus	Нет
ReadComponentsStatus	Нет
MonitoringMode	0, 1, 2, 3 – остановка мониторинга и частота выборки 250, 500, 1000 Гц соответственно
PowerOff	Нет

3.3 Протокол RFC 1055

Согласно сообщению разработчиков, перед отправкой каждый пакет кодируется по протоколу RFC 1055, использующему экранирование данных для разделения пакетов в потоке байт.

Экранирование данных (Byte Stuffing) — процесс добавления специальных символов или последовательностей символов в данные для того, чтобы предотвратить их неправильную интерпретацию или конфликт с управляющими символами.

Экранирующие символы, используемые в протоколе RFC 1055, представлены в таблице:

Значение	Кодовое имя	Описание
0xC0	END	Обозначает конец пакета.
0xDB	ESC	Символ экранирования. Указывает, что следующий байт не должен интерпретироваться как байт END или ESC.
0xDC	ESC_END	Экранированный символ END. Указывает, что байт данных имеет то же значение, что и байт END, но не должен интерпретироваться как байт END.
0xDD	ESC_ESC	Экранированный символ ESC. Указывает, что байт данных имеет то же значение, что и байт ESC, но не должен интерпретироваться как байт ESC.

Обработка данных с помощью протокола RFC 1055:

1. В начало списка добавляется специальный символ **END**
2. Обработка каждого байта входных данных:
 - Если текущий байт равен **END**, в список добавляются два байта: **ESC** и **ESC_END**. Это необходимо для того, чтобы избежать путаницы между данными и управляющими символами.
 - Если текущий байт равен **ESC**, в список добавляются **ESC** и **ESC_ESC** для того, чтобы предотвратить интерпретацию следующего байта как начала управляющей последовательности.
 - Все остальные байты добавляются в список без изменений.
3. В конец списка добавляется символ **END**, указывающий на окончание пакета данных.

Таким образом, RFC1055 предлагает простой способ выделения и инкапсуляции пакетных данных в потоке байтов Bluetooth Classic. Это позволяет легко разделять и идентифицировать отдельные пакеты в непрерывном потоке байтов.

3.4 Ответные сообщения устройства и обработка ошибок

Поле кода команды определяет тип сообщения и имеет следующий смысл:

- ReadDeviceInformation, ReadDeviceStatus, ReadComponentsStatus - ответ на соответствующую команду;
- Error = 254 - ошибка выполнения команды;
- MonitoringData = 255 – данные мониторинга ЭКГ, ускорения, дыхания.

Коды ошибок:

Код ошибки	Кодовое имя
0	OK
1	WrongCommandCode
2	WrongCommandDataSize
3	WrongCommandData
4	WrongDeviceStateForCommand
5	HardwareError
6	ProtocolNotSupported
7	Busy

3.5 Данные ЭКГ

Структура пакета данных ЭКГ:

- SampleNumber - номер первого отсчета ЭКГ в данном пакете.
- Status - байт состояния, включающий:
 - Младшие 4 бита (0-3) - отображение разрывов электродов.
 - Биты 4-5 - код частоты дискретизации ЭКГ (0 - 250 Гц, 1 - 500 Гц, 2 - 1000 Гц).
- ЭКГ-данные:
 - ecgData - массив отсчетов ЭКГ для I и II канала. Количество пар отсчетов зависит от частоты дискретизации (25, 50 или 100 пар).

- * Каждый отсчет представлен целым числом со знаком, которое умножается на константу Discrete для получения значения в вольтах.
- accData - массив из 15 байт, содержащий 5 отсчетов ускорения по координатным осям X, Y, Z.
- pgData - массив из 5 отсчетов дыхания, каждый по 24 бита.

Таким образом, пакет данных ЭКГ содержит информацию о номере первого отсчета, состоянии системы, а также фактические данные ЭКГ, ускорения и дыхания.

4 Описание спецификации Bluetooth

4.1 Общие сведения

Bluetooth - это стандарт беспроводной связи, который обеспечивает обмен данными между устройствами с использованием ультракоротких радиоволн. Он отличается от других стандартов беспроводной связи используемыми частотами, правилами передачи и методами шифрования данных.

Bluetooth работает на частотах, близких к 2,4 ГГц, и использует метод расширения спектра со скачкообразной перестройкой частоты (частотно-скачковая модуляция) для передачи данных. Диапазон частот разбит на 79 каналов, и устройство 1600 раз в секунду псевдослучайным образом выбирает один из них для передачи пакетов данных. Устройства, использующие одинаковый способ передачи, объединяются в пикосети, где есть главное, или ведущее, устройство и периферийные, или ведомые, устройства.

Bluetooth поддерживает режим работы в состоянии ожидания (standby), который позволяет сопряженным устройствам запоминать друг друга и автоматически подключаться после выключения или перезагрузки.

Существуют следующие версии и модификации Bluetooth [7]:

- Версия 1.0 появилась в 1999 году. Она не поддерживала анонимное подключение, требовала обмена адресами между устройствами, а связь по ней часто обрывалась. В версии 1.1 скорость передачи данных доходила до 1 Мбит/с.
- Версия 2.0 с поддержкой EDR — enhanced data rate (2004 год). Скорость соединения — до 3 Мбит/с. По сравнению с первой версией протокола сигнал стал стабильнее, а затраты энергии сократились в два раза. Выпущенная позднее версия 2.1 поддерживала технологию Sniff-Subrating, снижающую энергопотребление Bluetooth-модулей в 3–10 раз в зависимости от устройства.
- Версия 3.0 с поддержкой HS, или high speed, вышла в 2009 году и использовала два канала для передачи данных. Если требовалось передать небольшие файлы, то использовался энергосберегающий канал со скоростью до 3 Мбит/с, то есть Bluetooth 2.0, а для передачи больших файлов включался новый HS-канал со скоростью до 24 Мбит/с.

- Версия 4.0 (2010 год) — к старым протоколам был добавлен новый режим с низким энергопотреблением, позволяющий взаимодействовать с датчиками LE — low energy.
- Версия 5.0 (2016 год) — скорость передачи данных увеличилась до 48 Мбит/с.

4.2 Архитектура и стек протоколов Bluetooth Classic

4.2.1 PHY

На физическом уровне (PHY) Bluetooth Classic описывает детали радиоинтерфейса (используемые частоты, тип модуляции, мощность передачи), управления доступом к среде (установление соединения, распределение временных интервалов, адресация, различные состояния устройства Bluetooth и режимы передачи данных).

4.2.2 Link Manager Protocol

Протокол управления соединением (LMP) отвечает за установление и поддержание связи между устройствами Bluetooth. В рамках этого протокола реализованы функции аутентификации и шифрования, а также обмен информацией о размерах пакетов между устройствами.

4.2.3 Logical Link Control and Adaptation Protocol

Протокол управления логическими связями и адаптации (L2CAP) адаптирует кадры верхних уровней к формату кадров базовой полосы и наоборот. L2CAP обеспечивает ориентированные на соединение услуги, поддерживая мультиплексирование высокоуровневых протоколов и фрагментацию пакетов.

4.2.4 RFCOMM

Протокол RFCOMM обеспечивает эмуляцию последовательных портов RS-232 на основе L2CAP, предоставляя до шестидесяти одновременных соединений с Bluetooth-устройством. Этот протокол часто используется в профиле последовательного порта (Serial Port Profile, SPP), который позволяет устройствам обмениваться данными, как если бы они были подключены через последовательный кабель.

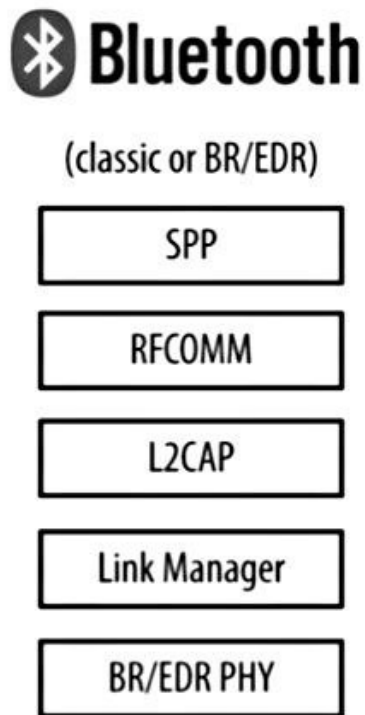


Рис. 2: Стек протоколов Bluetooth Classic

4.2.5 SPP

Bluetooth SPP (Serial Port Profile) - это профиль Bluetooth, который обеспечивает эмуляцию последовательного порта между двумя Bluetooth-устройствами. SPP предоставляет виртуальный последовательный порт, позволяющий устройствам обмениваться данными так же, как через физический последовательный порт и позволяет устройствам обмениваться данными на скоростях до 1 Мбит/с, в зависимости от версии Bluetooth.

4.3 Протокол RFCOMM

4.3.1 Общие сведения о Bluetooth Framework и RFCOMM

Bluetooth RFCOMM (Radio Frequency Communication) - это набор транспортных протоколов, построенных поверх протокола L2CAP, предоставляющих эмулированные последовательные порты RS-232.

Часто RFCOMM ассоциируется с Профилем Серийного Порта (SPP), но это не со-

всем так. Профиль Серийного Порты - это лишь один из многих профилей, основанных на протоколе Bluetooth RFCOMM. Профили Обмена Объектами (OPP), Передачи Файлов (FTP) и многие другие работают на основе RFCOMM.

Bluetooth Framework нативно поддерживает протокол Bluetooth Classic RFCOMM в режимах клиента и сервера. Это означает, что Bluetooth Framework может действовать как клиент RFCOMM (подключаться к другим устройствам с поддержкой Bluetooth) или как сервер RFCOMM (принимать подключения от других устройств с поддержкой Bluetooth). Это также означает, что для общения с устройствами Bluetooth не требуется создавать и/или использовать виртуальные COM-порты.

4.3.2 Bluetooth профили

Для использования технологии Bluetooth устройство должно быть совместимо с множеством профилей Bluetooth (часто называемых сервисами). Профиль Bluetooth (сервис) - это спецификация, касающаяся аспекта беспроводной связи Bluetooth между устройствами. Способ использования технологии Bluetooth устройством зависит от его профильных возможностей. Bluetooth Framework может работать только с профилями, построенными на основе протокола RFCOMM.

Профиль (сервис) Bluetooth определяется так называемым Универсальным Уникальным Идентификатором (UUID). UUID сервиса сообщает другому устройству, какие функции Bluetooth поддерживает устройство. Подключение RFCOMM обычно устанавливается с использованием UUID сервиса.

Устройство с поддержкой Bluetooth может запросить у другого устройства с поддержкой Bluetooth информацию о поддерживаемых сервисах, отправив запрос к его сервису Протокола Обнаружения Услуг (SDP).

4.3.3 Номер канала

Возникает вопрос: как клиентское устройство Bluetooth может выбрать, к какой службе оно хочет подключиться? У каждой службы RFCOMM есть свой уникальный номер канала. Этот номер канала может быть жёстко задан или может быть динамически назначен, но он всегда уникален.

Когда клиент подключается к удалённому серверу, он всегда подключается к каналу RFCOMM. Если приложение не указывает номер канала в запросе на подключение,

операционная система запрашивает удалённое устройство Bluetooth о его службах и пытается определить номер канала по UUID службы. Если существует две или более служб с одинаковым UUID, операционная система использует первую найденную случайным образом.

Однако в случае нескольких служб с одинаковым UUID приложение может запросить устройство о поддерживаемой службе, найти нужную и указать номер канала в запросе на подключение к требуемой службе. Или, если номер канала жёстко задан на стороне устройства, приложение может просто указать известный номер без запроса служб устройства. Номер канала RFCOMM может быть любым от 1 до 31 включительно.

5 Реализация приложения

В данной главе рассматривается реализация приложения для сбора и отправки данных электрокардиограммы (ЭКГ) в рамках системы медицинской телеметрии. Приложение написано на языке программирования Kotlin с использованием фреймворка Jetpack Compose для создания пользовательского интерфейса. Основные функции приложения включают подключение к кардиографу через Bluetooth, сбор данных ЭКГ, их визуализацию и отправку на сервер через протокол MQTT.

5.1 Основные подходы и используемые технологии

Предоставляя открытую платформу разработки, Android дает разработчикам возможность создавать гибкие и инновационные приложения. Android Software Development Kit (SDK) содержит множество инструментов и утилит для создания и тестирования приложений. Android включает в себя набор написанных на C/C++ библиотек, используемых различными компонентами системы. Эти возможности доступны разработчикам в контексте применения Android Application Framework.

В приложении применяется комплексный подход с использованием различных технологий Android-разработки, включая:

1. Jetpack Compose - фреймворк для декларативного создания UI в Android-приложениях, позволяющий описывать компоненты интерфейса в виде функций.
2. Корутинное Программирование - использование сопрограмм (coroutines) для выполнения асинхронных операций, таких как чтение данных и отправка их на сервер, без блокировки основного потока.
3. Bluetooth API - стандартный API Android для работы с Bluetooth-устройствами, обеспечивающий поиск, подключение и обмен данными.
4. MQTT - легковесный протокол передачи сообщений, используемый для отправки данных на сервер. В приложении используется библиотека для работы с MQTT, предоставляемая HiveMQ.
5. Работа с файловой системой - чтение и запись данных в файлы для временного хранения и последующей обработки.

5.2 Программная реализация

5.2.1 Графический интерфейс пользователя

Пользовательский интерфейс (рис. 3) строится с использованием компоновок Jetpack Compose, таких как Column и Row, которые позволяют удобно размещать элементы интерфейса на экране. Интерфейс включает кнопки для подключения к устройству через Bluetooth, запуска и остановки сессии мониторинга, а также отправки данных на сервер. Jetpack Compose позволяет легко обрабатывать события нажатий на кнопки с использованием лямбда-выражений, что делает код чище и проще в поддержке. Для визуализации данных ЭКГ используется линейный график, который строится на основе массива полученных значений с помощью библиотеки YChartslib, позволяющий настраивать различные параметры графика.

5.2.2 Взаимодействие с кардиографом

Данные от кардиографа принимаются через Bluetooth-соединение. Устройство подключается к кардиографу, после чего начинает считывать данные в реальном времени. Данные принимаются в отдельном потоке для того, чтобы основное приложение не зависало и не блокировалось во время работы с устройством. Это делается для повышения производительности и стабильности приложения, так как считывание данных может занимать значительное время.

5.2.3 Формирование MQTT пакетов из топиков и сообщений

В MQTT протоколе топики используются для адресации сообщений. В приложении топики формируются следующим образом:

```
ecg/{user_id}/{device_mac}/{frequency}/{flag}
```

Данные ЭКГ отправляются для конкретного пользователя, устройства и с определенной частотой. Например, топик может выглядеть так:

```
ecg/12345L/88:6B:0F:8B:7F:29/250/3
```

Сообщения содержат данные ЭКГ и временные метки, что позволяет точно отслеживать, когда было получено каждое измерение. Пример сообщения:



Рис. 3: Графический интерфейс (данные для графика получены с устройства КР-2 при подключённых электродах)

```
value: [4601670, 4601610],  
timestamp: "2024-05-24T10:00:00"
```

Когда пользователь хочет отправить данные на сервер, приложение считывает данные из файлов, формирует сообщения в формате JSON, и отправляет их на сервер. Приложение подключается к MQTT брокеру, используя аутентификацию. Затем данные отправляются по соответствующим топикам. После успешной отправки приложение отключается от брокера.

5.2.4 Использование сопрограмм

Сопрограммы в Kotlin используются для выполнения асинхронных задач. В приложении они применяются для выполнения задач, которые могут занять продолжительное время: отправка данных на сервер через MQTT. Асинхронность позволяет избежать блокировки основного потока приложения, что обеспечивает более плавную работу и отзывчивость пользовательского интерфейса.

6 Тестирование

Поиск и подключение устройства КР-2 к приложению прошли успешно, но первые же тесты показали необходимость оптимизации и использования механизмов многопоточности, коллбэков и сопрограмм, что и было реализовано.

Далее приложение было протестировано с помощью имитатора датчика электрокардиограммы, разработанного Голубковой М.С. [2], в результате чего были обнаружены потери данных. Причиной являлось то, что считывание байтов из сокета и журналирование ЭКГ происходило в одном и том же потоке. В результате было принято решение выполнять эти операции в разных потоках.

7 Заключение

В рамках данной курсовой работы было рассмотрено развитие клиентской части сервиса сбора и обработки медицинской телеметрии путем добавления поддержки устройства КР-2.

Пользователь приложения получил возможность подключаться к кардиографу при помощи технологии Bluetooth Classic, проводить сессии мониторинга ЭКГ, настраивая необходимые параметры: приложение предлагает выбор частоты дискретизации - 250 Гц, 500 Гц или 1000 Гц, что позволяет получать данные с необходимой степенью детализации. Считанные данные можно отображать в виде графика и отправлять на сервер при помощи протокола MQTT.

Список литературы

- [1] *Фролов, А. В.* Разработка приложения сбора данных с нескольких устройств интернета вещей, подключенных одновременно, на примере медицинской телеметрии. — 2023.
- [2] *Голубкова, М. С.* Разработка имитатора датчика электрокардиограммы в составе приложения имитации датчиков интернета вещей. — 2024. <https://github.com/IoMT-LVK/papers/blob/main/client>.
- [3] *Фролов, А. В.* Развитие клиентской части сервиса сбора и обработки медицинской телеметрии. — 2022.
- [4] *Чайчиц, Д. А.* Разработка и реализация клиентской части платформы для сбора и обработки медицинской телеметрии / Д. А. Чайчиц // *ПРОГРАММНЫЕ СИСТЕМЫ И ИНСТРУМЕНТЫ Тематический сборник*. — 2021. — Vol. 20.
- [5] *Бодров, А.О.* Построение масштабируемой платформы сбора медицинской телеметрии / А.О. Бодров, А.Г. Бахмуров // *ПРОГРАММНЫЕ СИСТЕМЫ И ИНСТРУМЕНТЫ Тематический сборник*. — 2021. — Vol. 21. — Рр. 20–32.
- [6] *Аникевич, Ю.В.* Разработка и реализация платформы для сбора и обработки физиологических данных о человеке / Ю.В. Аникевич, А.Г. Бахмуров, Д.А. Чайчиц // *ПРОГРАММНЫЕ СИСТЕМЫ И ИНСТРУМЕНТЫ Тематический сборник*. — 2020. — Vol. 20. — Рр. 36–48.
- [7] *Bluetooth SIG, Inc.* Bluetooth. — 2022. <https://www.bluetooth.com>.
- [8] *Sons, John Wiley* . An Overview of Enabling Technologies for the Internet of Things / John Wiley Sons // *Internet of Things A to Z: Technologies and Applications* / Ed. by Qusay F. Hassan. — 2018. — Рр. 79–112.