

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Рецензент

_____ 2021 г.
«___»_____

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

_____ Г.И. Радченко
«___»_____ 2021 г.

Разработка программно-аппаратного комплекса для мобильной RFID
идентификации

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090401.2021.594 ПЗ ВКР

Руководитель работы,
к.т.н., доцент каф. ЭВМ
_____ И.Л.Кафтанников
«___»_____ 2021 г.

Автор работы,
студент группы КЭ-222
_____ Н.С.Алдохин
«___»_____ 2021 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
«___»_____ 2021 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Г.И. Радченко
«___» _____ 2021 г.

ЗАДАНИЕ
на выпускную квалификационную работу бакалавра
студенту группы КЭ-222
Алдохину Никите Сергеевичу
обучающемуся по направлению
09.04.01 «Информатика и вычислительная техника»

Тема работы: «Разработка программно-аппаратного комплекса для мобильной RFID идентификации» утверждена приказом по университету от 26 апреля 2021 года №714-13/12

Срок сдачи студентом законченной работы: 7 июня 2021 г.

Исходные данные к работе:

- Финкенцеллер, К. RFID-технологии. Справочное пособие / К. Финкенцеллер; пер. с нем. Союнханова Н.М. — М.: Изд-во Додэка, 2010. — 496 с.;
- ГОСТ Р 58701-2019, "ИДЕНТИФИКАЦИЯ РАДИОЧАСТОТНАЯ ДЛЯ УПРАВЛЕНИЯ ПРЕДМЕТАМИ".

Перечень подлежащих разработке вопросов:

- изучение существующих разработок в области RFID идентификации подвижных объектов и мобильных считывающих систем;

- исследование физических процессов, происходящих при радиочастотном обмене между RFID меткой и считывателем;
- анализ опыта предоставления сервисов аренды индивидуальных транспортных средств;
- проектирование системы мобильной RFID идентификации;
- реализация проекта;
- тестирование.

Дата выдачи задания: 1 декабря 2020 г.

Руководитель работы:

Доцент каф. ЭВМ, к.т.н. _____ /И.Л. Кафтанников /

Студент _____ /Н.С. Алдохин /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2021	
Разработка модели, проектирование	01.04.2021	
Реализация системы	01.05.2021	
Тестирование, отладка, эксперименты	15.05.2021	
Компоновка текста работы и сдача на нормоконтроль	27.05.2021	
Подготовка презентации и доклада	30.05.2021	

Руководитель работы _____ /И.Л. Кафтанников /

Студент _____ /Н.С. Алдохин /

Аннотация

Н.С. Алдохин. Разработка программно-аппаратного комплекса для мобильной RFID идентификации. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭЖН; 2021, 64 с., 39 ил., библиогр. список – 17 наим.

В качестве данной выпускной квалификационной работы представлена разработка системы мониторинга средств индивидуальной мобильности, основанной на технологии радиочастотной идентификации объектов с применением комплексов RFID. Осуществляется обзор физических свойств рабочих процессов средств идентификации и их информационное сопровождение. Производится обзор опыта существующих проектов аренды индивидуальных средств передвижения. Производится разработка структуры диспетчерского пункта средств индивидуальной мобильности. Рассматриваются методы сканирования в условиях перемещения объектов и считывающих устройств в пространстве. Предлагается к рассмотрению модель мониторинга с помощью передвижных сканирующих комплексов как на автомобилях, так и на летательных беспилотных аппаратах.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	10
1.1 ОБЗОР ТЕХНОЛОГИИ RFID И ЕЁ КОМПОНЕНТОВ	11
1.2 ОБЗОР АНАЛОГОВ	28
1.3 ВЫВОД	34
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	35
2.1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	35
2.1.1 Реализуемые задачи	37
2.1.2 Требования к ПО	37
2.2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	38
2.2.1 Условия эксплуатации	38
2.2.2 Пользователи СИМ	39
2.2.3 Характеристики сервера	39
2.2.4 Характеристики аппаратного комплекса	39
2.2.5 Характеристики RFID-меток	39
3 ПРОЕКТИРОВАНИЕ	40
3.1 ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ ОБОРУДОВАНИЯ	40
3.2 АРХИТЕКТУРА РЕШЕНИЯ	44
3.3 ОПИСАНИЕ ДАННЫХ	47
3.4 АЛГОРИТМЫ РЕШЕНИЯ	52
4 РЕАЛИЗАЦИЯ	56
4.1 РЕАЛИЗАЦИЯ МАКЕТА	56
5 ТЕСТИРОВАНИЕ	67
5.1 МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ	67
5.2 ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ	67
ЗАКЛЮЧЕНИЕ	75
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	76
ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД МИКРОКОНТРОЛЛЕРА ESP	78
ПРИЛОЖЕНИЕ Б RNR СКРИПТ ОТОБРАЖЕНИЯ ТЕКУЩЕГО СОСТОЯНИЯ СИМ	80
ПРИЛОЖЕНИЕ В RNR СКРИПТ ЗАПРОСА ИНФОРМАЦИИ О СОСТОЯНИИ АРЕНДЫ ОТ МИКРОКОНТРОЛЛЕРА	81

ПРИЛОЖЕНИЕ Г RNR СКРИПТ ВЗЯТИЯ В АРЕНДУ СИМ	83
ПРИЛОЖЕНИЕ Д RNR СКРИПТ ЗАВЕРШЕНИЯ АРЕНДЫ СИМ.....	85
ПРИЛОЖЕНИЕ Е ДОПОЛНИТЕЛЬНАЯ ПРОГРАММА ТЕСТИРОВАНИЯ МИКРОКОНТРОЛЛЕРА ESP	87

ВВЕДЕНИЕ

Целью данной работы является разработка системы мониторинга объектов в городе с помощью мобильных технологий радиочастотной идентификации. Данная система в наибольшей степени направлена на решение задачи отслеживания местоположения арендуемых средств индивидуальной мобильности (СИМ), то есть, велосипедов, самокатов, и подобного малого транспорта в черте города с помощью радиочастотных меток и множества не только зафиксированных радиочастотных считывателей, но и перемещающихся в пространстве.

Любое введение киберфизических объектов в систему с участием человека требует создания аспекта индивидуальности каждого из таких объектов. Уникальность объектам придают идентификаторы, и другая соответствующая характеристикам каждого изделия информация. Всё это хранится в радиочастотных метках. Такие метки выступают отличным аналогом штриховых и QR кодов, государственных регистрационных автомобильных номеров, так как не требуют прямого оптического взаимодействия с объектом. Кроме этого фактора метки не требуют особенно строгих правил размещения на корпусе, а также они не подвержены порче данных вследствие внешнего загрязнения.

Стационарные считыватели предлагается использовать как с одиночными антеннами, так и с их массивом. В первом случае, если антенна будет неподвижно зафиксирована, станет возможным только регистрировать наличие объекта с установленной на него меткой в некоторой предполагаемой области пространства, совпадающей с областью воздействия электромагнитного сигнала антенны. И такой считыватель с одной антенной имеет статическую привязку к спутниковым координатам. Эта конфигурация подходит для мониторинга проездов, фиксации факта прибытия, убытия, нахождения в точке сканируемого пространства. Другой вариант использования одной антенны – её вращение. Что предоставит возможность

не только увеличить область считывания, но и предполагать о местоположении объекта на основании угла поворота и мощности отражаемого меткой сигнала. В случае с несколькими антеннами, система тоже позволит предполагать о местоположении объекта в пространстве. Достигается это с помощью измерения и сравнения времени отклика метки до каждого считывателя. Перемещающиеся считыватели делают возможным мониторинг объектов, например, на нескольких площадках в черте города. Для этого достаточно обычный считыватель с антенной смонтировать на транспортном средстве. Сканируя объекты, такой считыватель, с привязкой его к спутниковым координатам позволит получать информацию о местоположении объектов.

В современных городах активно развивается сеть средств индивидуальной мобильности (СИМ), то есть самокатов, велосипедов и подобного транспорта. Зачастую такие средства передвижения располагаются хаотично в пешеходной части дороги. Правовые нормы относительно СИМ активно развиваются и появляется проблема идентификации множества небольших устройств. Данный вопрос распознавания конкретного устройства помогает решить радиочастотная идентификация.

Основная цель системы мониторинга объектов в городе с помощью мобильных технологий радиочастотной идентификации – отслеживание передвижения СИМ, их параметров и показателей движения в черте города, помощь в фиксации случаев ДТП и в расследовании причин их возникновения. Из этого следует ряд задач, например, ведение базы данных меток, прикрепляемых к самокатам, запись идентификационной информации в метки, применение алгоритмов определения местоположения и скорости.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Современный человек окружает себя всё большим числом средств передвижения. Идентификация таких средств на данный момент производится с помощью оптических средств отслеживания. То есть с помощью автоматических средств фото- и видеофиксации нарушений ПДД [1]. Единственным идентификатором в таких системах остаётся государственный регистрационный номер на кузове авто. Но на сегодняшний день помимо автомобилей, мотоциклов и прочего транспорта существуют менее габаритные средства передвижения. Это так называемые средства индивидуальной мобильности, то есть самокаты с электрическим двигателем. Их движение менее упорядоченное, чем у автотранспорта, а корпус гораздо меньше. Это порождает ряд трудностей в идентификации таких средств на улицах города и отслеживания их параметров на момент дорожно-транспортного происшествия. Изучение вопроса применения радиочастотной идентификации для мониторинга таких объектов в городе призвано попытаться решить проблему идентификации и предложить модель системы, которая поможет в вопросах регулирования ответственности при управлении подобными устройствами людьми.

Радиочастотная идентификация (англ. Radio Frequency IDentification - RFID) – технология записи и чтения информации без механического контакта между устройствами, использующая радиочастотное электромагнитное излучение [2]. Получение информации происходит с помощью устройства, называемого считывателем (англ. Reader) и специальных электронных элементов, меток (англ. tag), иногда называемых транспондерами (англ. transponder). Технология RFID идентификации основана на сопоставлении объекта и метки, являющейся его неотъемлемой частью. Задачей RFID системы является идентификация объектов при помощи считывания информации, записанной на метке. Метка может содержать и передавать по команде от считывателя данные о типе объекта, стоимости, весе, температуре,

данные логистики, вообще любой информации, которая может храниться в цифровой форме.

В современном городе много различных объектов и сервисов. Многие из них связаны с транспортом. Это является следствием расширения городов по площади и возникающей отсюда необходимостью простым гражданам перемещаться на дальние расстояния, и делать это с максимальным удобством.

В такой ситуации для обеспечения качественной жизнедеятельности и поддержания порядка можно воспользоваться средствами мониторинга объектов. В случае с технологией радиочастотной идентификации стало возможным устанавливать на транспортные средства различных типов упомянутые метки, которые в идеальных условиях однозначно идентифицируют объекты в местах пребывания и перемещения. Привязка мест проверки к координатам автоматически укажет на местоположение объекта во время считывания.

Но также существует несколько других вариантов мониторинга. Например, отслеживание координат объектов с помощью трекеров спутниковой навигации, приблизительное определение местоположения по уровню сигнала мобильной сети, ещё более грубое по точности – с помощью точек беспроводного доступа Wi-Fi, а также использование камер видеонаблюдения. Обзор технологий предлагаемого решения на радиочастотной идентификации, а также аналогов вместе с их достоинствами и недостатками рассматриваются далее.

1.1 ОБЗОР ТЕХНОЛОГИИ RFID И ЕЁ КОМПОНЕНТОВ

RFID система состоит из трех базовых компонентов [2]:

1. считывающее устройство (передатчик/приемник);
2. антенна;
3. радиочастотная метка с встроенной антенной, приемником и передатчиком.

На рынке представлено большое количество разновидностей этих компонентов. Они различаются по типу исполнения, функциональным особенностям. Считыватель может быть представлен в различных форматах – переносной сканер, стационарный для установки в помещениях или на улице. Антенны различаются по размерам и форме. Метки выпускаются как с элементами питания, так и без них, с различными режимами доступа к памяти, разного физического исполнения. На рисунке 1.1.1 представлена топология составляющих RFID-системы.



Рисунок 1.1.1 – Компоненты RFID-системы

Считывающее устройство выполнено единой совокупностью передатчика и приемника. Иногда оно разрабатывается со встроенной антенной. Транспондер – приёмопередающее устройство, которое при восприятии команд на работу со своей информацией – выполняет их. Например, от считывателя поступила команда чтения – отдать запрашиваемые блоки информации. Физически метка состоит из антенны для приёма и передачи информации и микросхемы, хранящей и обрабатывающей информацию. Метки комплектуются встроенным источником питания, который значительно расширяет область считывания и реакции метки. В роли системы сбора информации может выступать рабочая станция при локальном соединении компонентов, или веб-сервер, когда необходима отправка данных от удалённых считывателей.

RFID-метки

Для работы с метками нет необходимости иметь визуальный доступ к ним. Транспондер можно установить на различных частях объекта, и он

сможет участвовать в экосистеме радиочастотной идентификации. Теоретически, для повышения вероятности идентификации объекта, к нему можно прикрепить несколько меток и связать их с ним в информационной системе. RFID-метки, в отличие от других оптических способов идентификации, не подвержены влиянию изменений освещённости, положения объекта в области считывания, а также при определённом исполнении и от неблагоприятных погодных условий.

В настоящее время RFID-метки состоят из двух частей – антенны и интегральной схемы. Благодаря первой происходит приём и передача радиочастотного сигнала, а также осуществляется питание интегральной схемы метки посредством этого же сигнала для формирования ответного сообщения. А вторая необходима для хранения и обработки информации, модулирования и демодулирования радиочастотного сигнала и некоторых других функций. На рисунке 1.1.2 представлена метка с указанными компонентами.

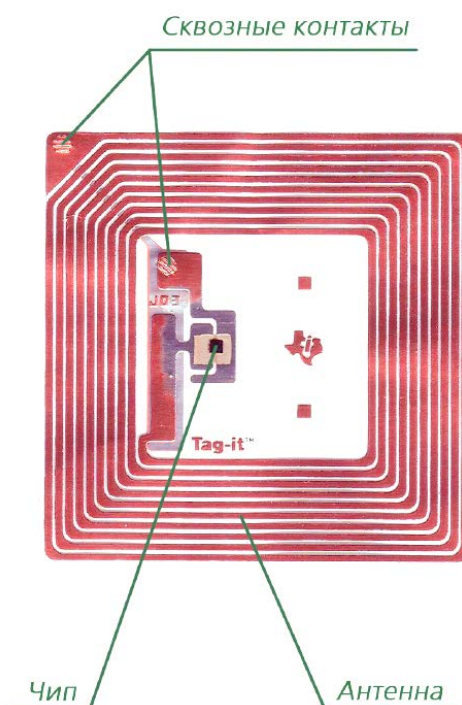


Рисунок 1.1.2 – Основные компоненты пассивной RFID-метки

По способу обеспечения питания метки бывают:

- пассивные. Не имеют встроенного источника энергии. Благодаря индуцированию радиоволн в антенне возникает электрический ток, который служит источником питания для чипа, а также приёма и передачи информации;

- активные. Не используют получаемую от считывателя, потому что имеют встроенный источник питания. Их дальность считывания гораздо больше, чем у пассивных, так как используют энергию источника. Они могут взаимодействовать с различными дополнительными электронными устройствами, а именно датчиками;

- полупассивные. Очень похожи на пассивные метки, однако, встроенное питание в них используется только для обеспечения функционирования чипа, а радиосигнал отражается, а не генерируется.

Также метки имеют различные виды встроенной в них памяти:

- только чтение (англ. Read Only – RO) – такая память предусматривает запись информации при изготовлении метки без каких-либо дальнейших манипуляций. Метки с этим типом памяти можно использовать для идентификации. Но с условием, что к объекту придется привязывать заданный производителем идентификатор. То есть, если потребуются периодически изменять на метке информацию – она становится непригодной для такого типа задач;

- однократная запись и многократное чтение (англ. Write Once Read Many – WORM) – содержат уникальный идентификатор, а также область памяти, в которую можно записать интересующую информацию. Но сделать это можно однократно, только чтение является многократным;

- многократно перезаписываемая (англ. Read and Write – RW) – такие метки содержат идентификатор и блок памяти для чтения (записи) информации. Данные в них могут быть перезаписаны многократно.

По рабочей частоте метки бывают:

- низкочастотные (125—134 кГц);
- средней частоты (13,56 МГц);
- высокочастотные (860—960 МГц и более).

Первые используются для введения меток людям, животным, рыбам. Для последних это хороший вариант для идентификации живых существ при различных биологических природных процессах. Но длина волны не позволяет считывать эти метки на большом расстоянии, а также появляется большое количество коллизий при чтении множества меток в ограниченном пространстве. Системы на частоте 13,56 МГц наиболее стандартизованы, имеют малую стоимость, широко применяются. Например, в системах идентификации личности, контроля доступа, платежах, обмене информацией между мобильными устройствами. Но расстояние чтения не позволяет их использовать на дистанциях более 1,5 метров, а также не решены проблемы коллизий. Системы высокочастотного диапазона могут взаимодействовать с объектами на расстояниях до 25 метров для пассивных меток, а также они оснащены антиколлизийными алгоритмами, что позволяет считывать большое количество меток на широкой площади.

Согласно стандартам EPC Global пассивные метки относят к классам 0 и 1. Метки типа класса 0 оснащаются не перезаписываемым уникальным идентификатором EPC (Electronic Product Code) при производстве, который в перспективе может быть только прочитан. Метки типа класса 1 могут получить этот идентификатор от пользователя единственный раз. Затем доступ к памяти метки будет так же только на чтение. На рисунке 1.1.3 можно увидеть схему размещения данных в памяти.

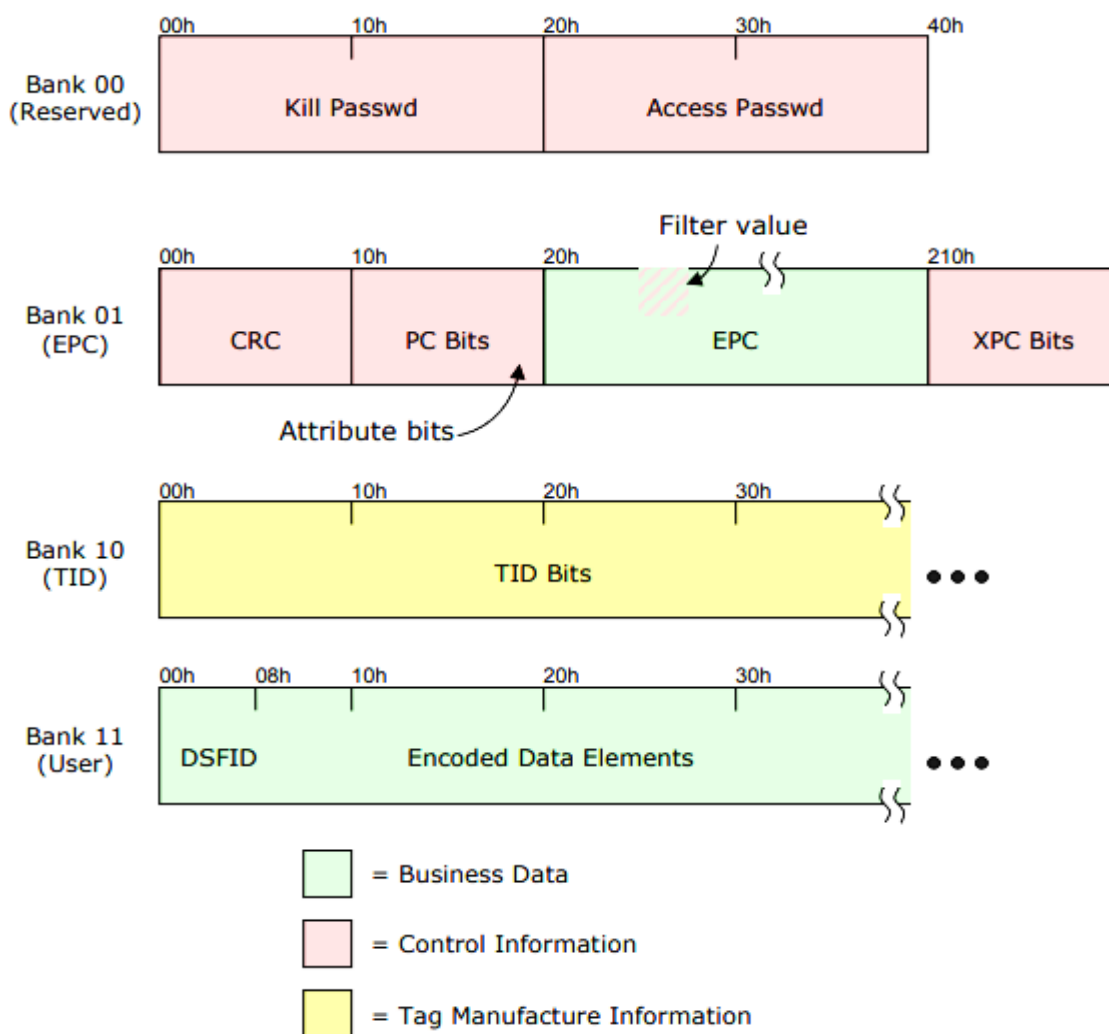


Рисунок 1.1.3 – Схема данных в памяти метки

Условно она разделяется на 4 подраздела[2]:

1. зарезервированный:
 - 1.1. Kill Passwd – пароль на уничтожение метки (32 бита). При ненулевом значении этого поля, зная его, можно послать команду ликвидации метки, после чего та полностью перестанет функционировать;
 - 1.2. Access Passwd – пароль на доступ к метке (32 бита). Если это поле установлено, то доступ к ней будет возможен только при знании этого пароля.
2. EPC (англ. Electronic Product Code – электронный код продукта);
 - 2.1. CRC (англ. Cyclic Redundancy Check – циклическая проверка избыточности) – 16-разрядное значение, вычисленное на

основе содержимого памяти метки, предназначенное для проверки целостности данных;

2.2. PC bits (англ. Protocol Control bits – биты управления протоколом) – 16 бит под различные атрибуты протокола;

2.3. EPC bits (англ. Electronic Product Code – электронный код продукта) – глобальный уникальный идентификатор физического объекта размером 496 бит;

2.4. XPC bits (англ. Extended Protocol Control bits – биты расширенного управления протоколом). 16 бит для расширенных настроек. Доступны, если установлен бит 16h в EPC;

3. TID bits (англ. Transponder ID – идентификатор метки);

4. Encoded Data Elements (Зашифрованный раздел данных). Этот раздел может отсутствовать. Если есть, то обычно занимает от 32 до 512 бит.

Активные RFID-метки

Главным образом, такие метки отличаются от пассивных встроенным источником питания (см. рисунок 1.1.4). Собственный источник питания позволяет активным меткам генерировать выходной сигнал большого уровня для увеличения дальности уверенного считывания до сотен метров и использования метки в таких средах как вода, тело животного или человека, или металл [4].

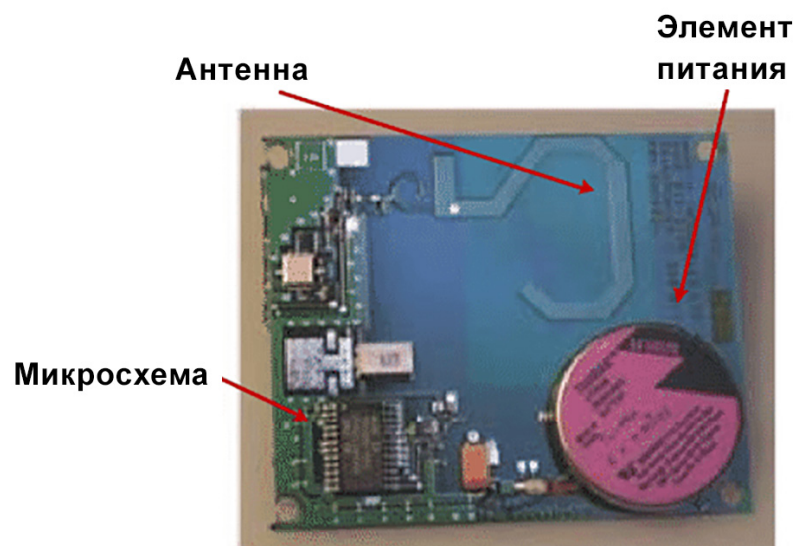


Рисунок 1.1.4 – Активная RFID-метка на текстолите

Тип меток для установки на металл (см. рисунок 1.1.5) позволяет производить установку как с помощью двустороннего скотча, так и с помощью крепёжных болтов для большей надёжности. Такой тип меток предполагается к применению на СИМ сделанных преимущественно из металла.



Рисунок 1.1.5 – Активная RFID-метка для установки на металл с размерами в мм

Средний срок службы элемента питания может варьироваться в пределах от 3 до 10 лет в зависимости от габаритов корпуса самой метки, а также дополнительных датчиков, которыми она может быть оснащена.

Преимущества данного типа меток при применении в текущем проекте следующие:

- контроль СИМ даже при их движении в момент сканирования;
- возможность расположить метку в любых местах корпуса СИМ, в том числе внутри при гарантии считывания с меньшим количеством помех по сравнению с пассивными метками;
- контроль каких-либо параметров СИМ, например, скорость, если метка оснащена дополнительными датчиками;
- возможность расширять возможности взаимодействия метки и СИМ в будущем для улучшения качества мониторинга и введения.

Исследование [5] показало, что определение показателя уровня принимаемого сигнала, RSSI (англ. received signal strength indicator), активных меток не всегда даёт однозначные результаты, а также зависит от очень большого набора параметров, которые достаточно сложно учесть и систематизировать при решении проблемы определения местоположения метки. В этом исследовании изучалось влияние 4 параметров системы на основе значений RSSI: тип материалов, к которым прикреплены метки, препятствия между метками и антенной, расстояние между метками и антенной, относительная ориентация меток и антенны. В качестве критерия оценки был выбран RSSI, поскольку он считается ключевым показателем в большинстве алгоритмов локализации в беспроводных интерфейсах. В этом исследовании было отмечено, что все испытанные материалы (включая алюминий, потолочную доску, гипсокартон, пенопласт, стекло, ковер, глину, фанеру, сталь и картон) в некоторой степени влияют на значения RSSI и что эти эффекты не могут быть ранжированы. Например, было установлено, что сталь, фанера и алюминий являются наиболее доминирующими материалами, влияющими на значения RSSI. При испытаниях на препятствие, следует отметить, что наличие стекла влияло на результаты в присутствии людей на испытательной площадке, в то время как алюминий и гипсокартон привели к

наиболее заметным эффектом при отсутствии людей. Значения RSSI были несовместимы между 2 испытательными стендами (с людьми и без них), и различные испытательные стенды приводили к различным значениям RSSI, даже когда существовала одна и та же преграда. По итогу этого исследования делается вывод, что влияние параметров системы не является последовательным и не оказывает поддающегося количественной оценке влияния на значения и изменчивость RSSI. Результаты показывают, что дисперсия и сила сигналов RSSI напрямую не связаны друг с другом, но они индивидуально зависят от окружающей среды.

RFID-считыватели

Считыватели (англ. reader) – это приборы, которые читают информацию с меток, а также могут записывать в них данные. Эти устройства могут быть постоянно подключенными к учетной системе, или работать автономно.

Виды считывателей:

- стационарные: устанавливаются на любые, зачастую неподвижные объекты: стены, столбы, иные конструкции. Они могут быть выполнены в виде замка, вмонтированы в стол или закреплены рядом с конвейером на пути следования изделий. По сравнению с переносными, считыватели такого типа обычно обладают большей зоной чтения и мощностью, как правило от 1 до 100 метров, реже – около 300 метров, и способны одновременно обрабатывать данные с нескольких десятков меток. Стационарные считыватели подключаются к программируемым логическим контроллерам или к ПК. Задача таких считывателей – поэтапно фиксировать перемещение маркированных объектов в реальном времени, либо идентифицировать положение отмеченных меткой предметов в пространстве;
- мобильные: обладают сравнительно меньшей дальностью действия и зачастую не имеют постоянной связи с программой контроля и учета. Мобильные считыватели имеют внутреннюю память, в которую

записываются данные с прочитанных меток (потом эту информацию можно загрузить в компьютер) и, как и стационарные считыватели, способны записывать данные в метку (например, информацию о произведённом контроле). В зависимости от частотного диапазона метки, дистанция устойчивого считывания и записи данных в них будет различна. Но как правило это не более 10 метров.

Помимо приведенной классификации есть разные реализации считывателей: одноплатный модуль, взаимодействующий с микроконтроллером или другим управляющим устройством с помощью интерфейсов SPI или UART, а также готовое устройство с предустановленной операционной системой (например, Windows CE или Linux), с которым можно взаимодействовать по RS232 или RS485, а также они могут иметь порты USB и UTP Ethernet.

В качестве одного из примеров считывателя, реализованного на одной плате можно рассмотреть "SparkFun Simultaneous RFID Reader-M6E Nano" (см. рисунок 1.1.6). Он основан на модуле "M6E-Nano" от производителя ThingMagic [6].



Рисунок 1.1.6 – Плата "SparkFun M6E Nano"

Данный модуль позволяет читать несколько меток одновременно, а также работать с внешней антенной. Обмен данными с микроконтроллером происходит посредством UART интерфейса. Настройку мощности и производительности под заданные потребности можно производить с помощью программного обеспечения "Universal Reader Assistant". Эту же программу можно использовать для считывания информации. После настройки можно перейти к использованию с микроконтроллером. Плата создана специально в виде расширяемого модуля для «Arduino UNO». Производитель также предоставляет библиотеку, позволяющую работать с микроконтроллером в "Arduino IDE". Функционал библиотеки аналогично программе "Universal Reader Assistant" позволяет настроить считыватель.

Плата считывает метки "EPC Global Gen 2" (соответствующие диапазону 860-960 МГц) со скоростью до 150 штук в секунду. Плата имеет регулируемую выходную мощность от 0 дБм до +27 дБм. Есть возможность считывать метки на расстоянии до 5 метров при подключении внешней антенны. Встроенная позволяет производить считывание на расстоянии до 60 см.

К плюсам данного считывателя относится его исполнение на одной плате, которая может подключаться как расширяемый модуль к Arduino UNO, либо просто использоваться с любым другим микроконтроллером, а также предоставляемая библиотека. Минусом устройства является достаточно малая дальность считывания – 5 метров. Это вплотную подходит для использования на остановках в городе, а иногда может быть недостаточно. Его стоимость составляет 17560 рублей, это средний показатель для такого рода устройств.

Другой вариант одноплатного считывателя – FM-503 от китайского производителя Fonkan [7]. Данная фирма занимается производством радиочастотного оборудования. Основана в 2016 году, самостоятельно занимается разработкой продуктов, которые распространяет по всему миру. Упомянутый считыватель обладает интерфейсами USB, RS-232, UART, выводом под внешнюю антенну и питание 12 вольт. При подключенной

антенне 5.5 dBi заявленная дальность чтения меток до 4 метров. На рисунке 1.1.6 представлен данный считыватель с подключенной к нему антенной.

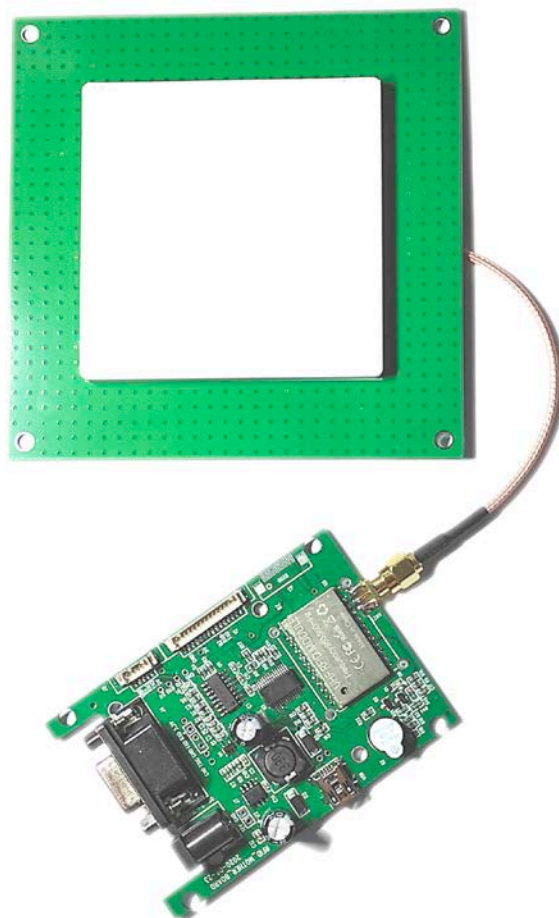


Рисунок 1.1.6 – Считыватель Fonkan FM-503 с подключенной антенной
В качестве готового к использованию без дополнительных действий, не касающихся настройки непосредственно на месте установки, рассматривается считыватель "Motorola FX7400", представленный на рисунке 1.1.7.



Рисунок 1.1.7 – Считыватель "Motorola FX7400"

Данное устройство обладает 4 портами для подключения антенн, выводом для GPIO (англ. general-purpose input/output – порты ввода/вывода), USB type-B, UTP Ethernet. На нём предустановлена операционная система Windows CE 5.0, а также есть доступ к различным настройкам по веб-интерфейсу с другого устройства. Можно отметить наличие API (англ. application programming interface – программный интерфейс приложения), поддерживающего языки Java и C (платформу .NET). Благодаря нему происходит обращение к настройкам и функциям считывателя из других приложений.

Считыватель поддерживает стандарт ISO 18000-6C (EPC Class 1 Gen 2, высокочастотные метки), его выходная мощность от +15 до +30 дБм.

Плюсами данного устройства являются: его полноценная готовность к использованию по назначению, а также различного рода сопутствующее ПО для управления и наличие API. К минусам можно отнести избыточный функционал для изучаемой области мониторинга транспорта на остановках и соединение с сетью Интернет только через UTP Ethernet, а так же цену порядка 38000 рублей, которая обуславливается богатым функционалом.

RFID антенны

Иногда в считывателях уже установлена антенна, однако её дальность считывания не позволяет работать в масштабах пешеходных зон. Если идёт речь о считывании меток на расстоянии в десятки метров, то важным элементом аппаратного комплекса является антенна. На рисунке 1.1.8 представлен пример такой антенны.



Рисунок 1.1.8 – RFID антенна диапазона UHF CLOU CL7205A

Основными параметрами антенны являются:

- ширина луча;
- направленность;
- тип поляризации.

Ширина луча определяется по диаграмме направленности как угол между двумя точками в одной плоскости, где излучение падает примерно на 3 дБ ниже максимальной точки, в пределах которого излучается наибольшая часть энергии сигнала. Его также можно рассматривать как пиковую эффективную величину излучаемой мощности основного направления волны. Существует две ширины луча – азимут (горизонтальный) и угол возвышения (вертикальный). Но чаще всего ширина луча рассматривается как горизонтальный угол на диаграмме направленности. Как правило, ширина

луча ультравысокочастотных антенн колеблется в диапазоне от 60 до 80 градусов. На рисунке 1.1.9 представлен пример диаграммы направленности с подписанными элементами.

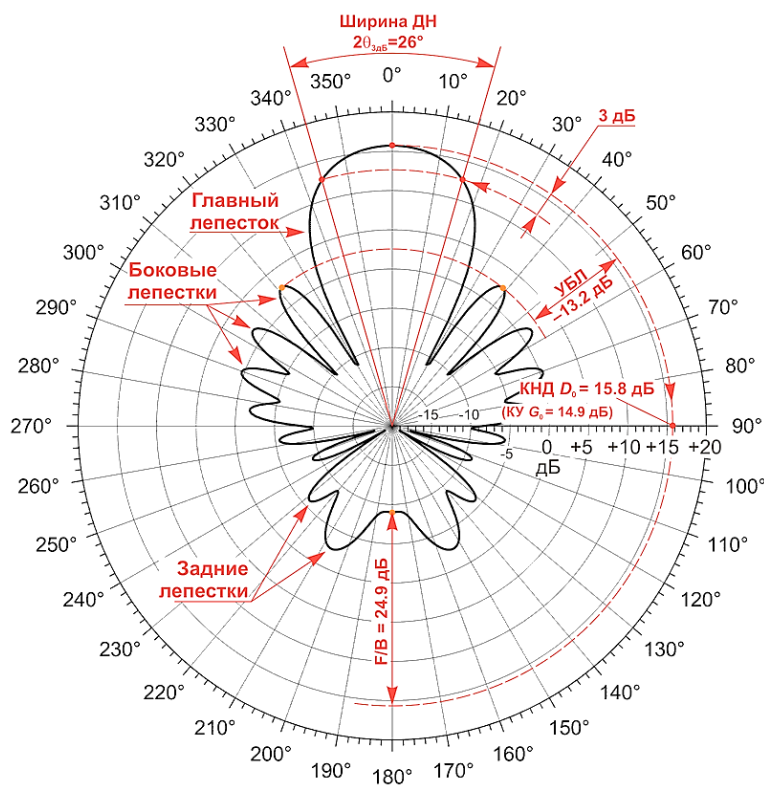


Рисунок 1.1.9 – Диаграмма направленности антенны

Направленность антенны определяется как ее способность фокусироваться в определенном направлении для передачи или приема энергии.

Третьим важным параметром упоминался тип поляризации. При линейной поляризации вектор амплитуды волны изменяется в одной плоскости (см. рисунок 1.1.10), и метка должна располагаться строго вдоль длины антенны. При круговой поляризации вектор описывает окружность в плоскости колебаний (см. рисунок 1.1.11), что даёт возможность располагать метку любым образом.

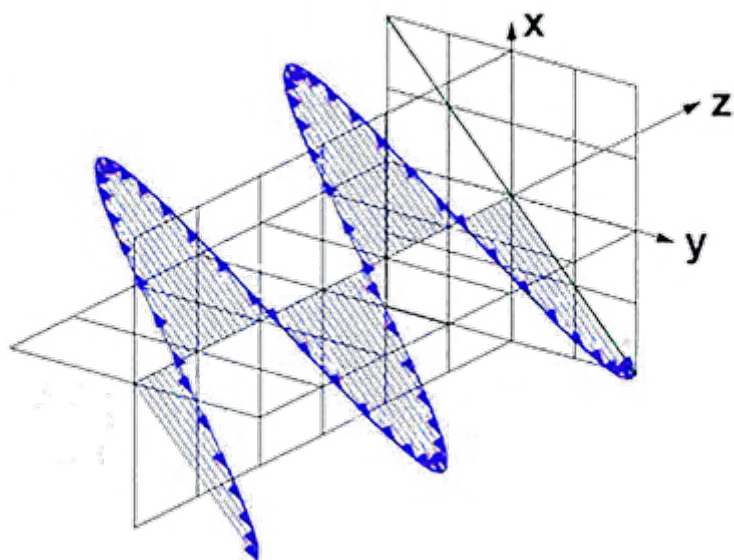


Рисунок 1.1.10 – Колебание вектора амплитуды при линейной поляризации

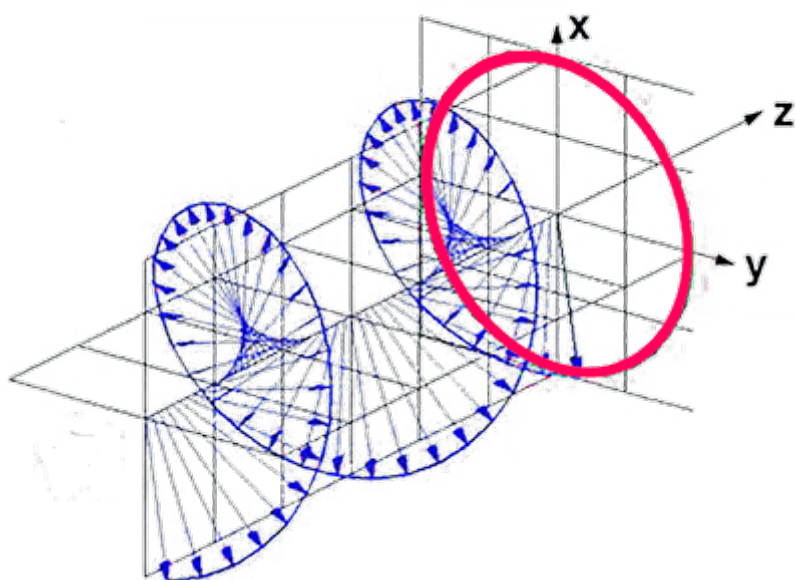


Рисунок 1.1.11 – Колебание вектора амплитуды при круговой поляризации

Например, антенна CLOU CL7205A обладает следующим набором характеристик:

- ширина луча – 70 градусов;
- частота излучения – 840-960 МГц;
- тип поляризации – левая круговая.

1.2 ОБЗОР АНАЛОГОВ

Очевидным и самым распространённым аналогом является мониторинг посредством спутниковой навигации, например, ГЛОНАСС. Основной принцип данного типа навигации лежит в расчете псевдодальности до навигационного космического аппарата (НКА) [8]. Псевдодальность – это разница между показаниями часов приемника и сигнального времени. Сигнал спутника можно интерпретировать как его временную шкалу. Время сигнала всегда сопоставляется по шкале времени спутника. В выбранной шкале времени приемник выбирает необходимое ему. Сверяет показания времени каждого сигнала. За время распространения показания часов меняются, поэтому возникает разница. Она соответствует задержке распространения сигнала. А из задержки вычисляется расстояние до той точки, где находился спутник в момент излучения. На рисунке 1.2.1 представлена простейшая схема взаимодействия компонентов с земли со спутниками на орбите с указанными линиями расстояний r_i . Точность данного метода составляет от 1.2 метра и более, в зависимости от состояния атмосферы в области радиообмена.

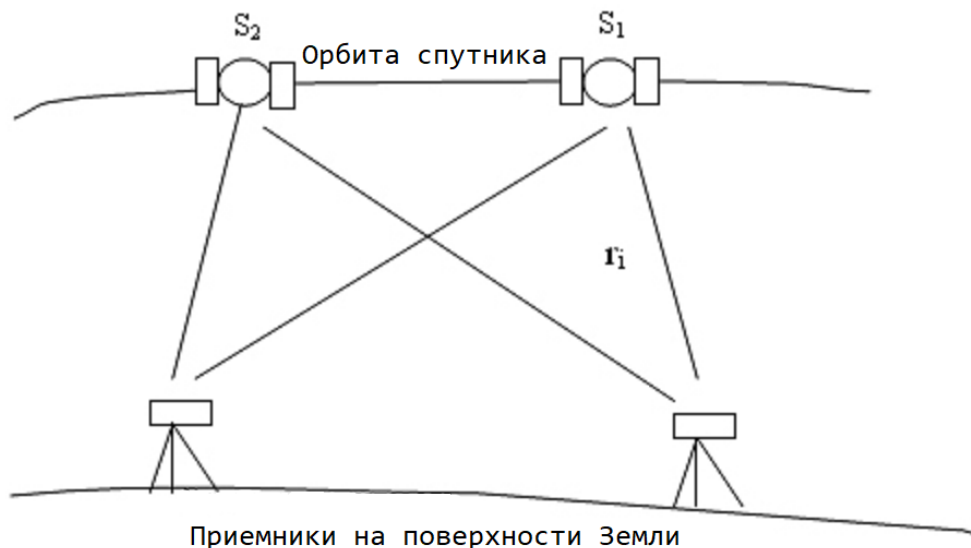


Рисунок 1.2.1 – Схема взаимодействия спутника и приемника

Метод спутниковой навигации позволяет контролировать загруженность городских дорог с помощью внедрения умных светофоров,

получающих данные о пробках онлайн, и установки на автобусы отслеживающего оборудования, кратко называемого трекерами [9, 10]. Для работы системы спутникового мониторинга понадобится установить на транспорт мобильный модуль, состоящий из следующих частей: приемник спутниковых сигналов, модули хранения и передачи координатных данных. Программное обеспечение данной системы с короткими интервалами времени записывает текущие координаты транспортной единицы. Если установлен модуль передачи данных, работающий с помощью мобильной сети Интернет, то координаты передаются в режиме онлайн на сервер и сразу доступны клиентам. Если такой модуль отсутствует, то онлайн мониторинг невозможен, и единственный вариант использования устройства – проверка добросовестного выполнения маршрута на конечных остановках путем переноса данных на какой-либо внешний носитель. Примером может служить такая система на автобусном предприятии ЕМУП МОАП города Екатеринбурга [10]. Спутниковые трекеры используются также в области мониторинга и меньших устройств. Например, для нахождения на карте электрических самокатов, которые можно взять в аренду на время. В США наиболее развиты три сервиса предоставления данного вида малого индивидуального транспорта в аренду: Bird, Lime, Spin. Спутниковые трекеры в самокатах используются для определения местонахождения свободных (не арендуемых кем-то в данный момент времени) устройств [11]. Необходимо это в связи с тем, что оставлять их можно где-угодно, а не на специальных станциях. А сама аренда инициируется сканированием QR-кода на устройстве через приложение арендодателя.

Сотовая связь также позволяет обрабатывать данные о местоположении клиентов. Причем в этой области используются не только методы с применением современных возможностей индустрии информационных технологий, но и основанные на базовых физических принципах распространения и задержки радиосигнала.

Например, первая часть методов базируется на использовании ресурсов и информации из баз данных самих операторов сотовых сетей. В этом случае применяются разные алгоритмы определения местоположения мобильных терминалов по радиосигналам и их характеристикам: по амплитуде в месте приема, направлению распространения сигнала и времени задержки при распространении. А затем происходит сопоставление таких "отпечатков" радиосигнала с образами (англ. signature) мест расположения абонентов (англ. Location Pattern Matching – LPM) [12]. Полученная таким образом информация сравнивается с некоторой накопленной базой аналогичных данных, соответствующих различным вариантам расположения абонента на местности. Точность определения местоположения составляет обычно 1 – 100 метров.

Недостаток технологии в том, что вычислительные действия ложатся на долю базовых станций. При этом информация о местоположении выдается абоненту с периодом повторения 10–20 сек. При условии, что координаты базовых станций известны, для определения местоположения можно использовать метод, основанный на приеме сигналов базовых станций. В отличие от приведенных выше методов, то можно вычислять свое местоположение непрерывно, так как в данном случае нет необходимости делать запросы на базовые станции. Данная технология может быть основана на использовании информации о направлении приема сигнала от базовых станций. Контроллер в приемнике сигналов сможет вычислять собственное местоположение на основе информации о пересечении линий направления приема сигнала, это вносит требования двух и более базовых станций и приемника, оборудованного антеннами с узкой диаграммой направленности [13]. Точность также варьируется в диапазоне от 1 до 100 метров. Взаимодействие компонентов этого представлено на рисунке 1.2.2

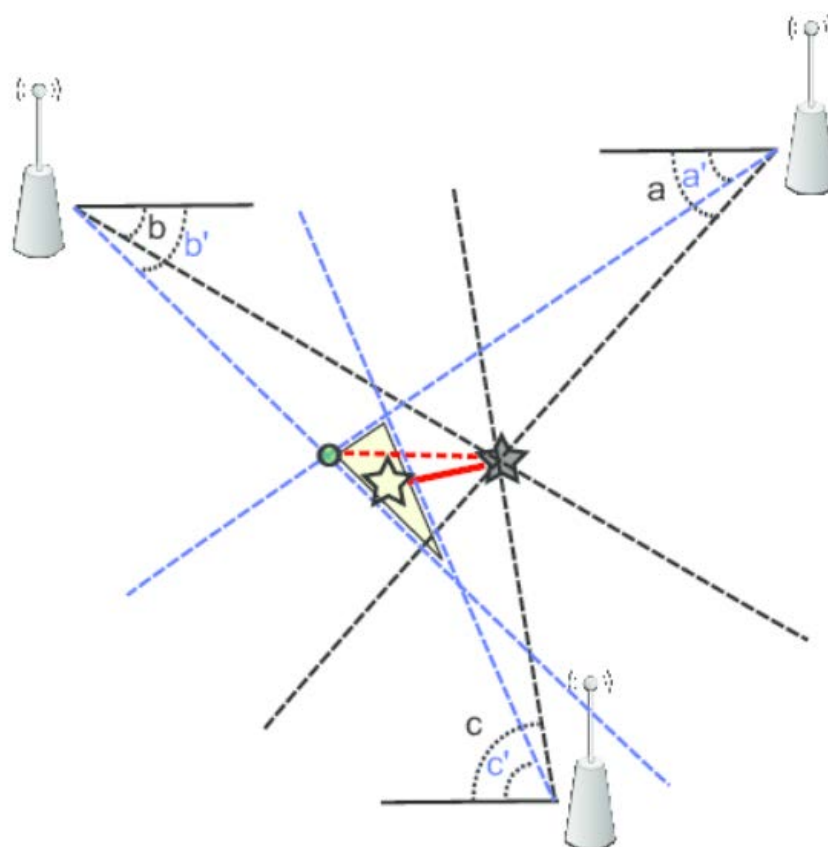


Рисунок 1.2.2 – Определение местоположения по трём базовым станциям

Мониторинг местоположения объектов с некоторыми неточностями может проводиться с помощью сетей беспроводной связи Wi-Fi. Иногда этот метод применяется на какой-либо ограниченной территории, например, для мониторинга персонала и техники на предприятии [14]. Использоваться могут как телефоны, ноутбуки и иные мобильные устройства, так и Wi-Fi-маяки. Инфраструктура обычно представлена в виде набора точек доступа либо объединенных в распределённую беспроводную сеть (Wireless Distribution System – WDS), либо работающих независимо.

При определении местоположения объектов с помощью Wi-Fi используется несколько методов [15]. Вот они, слева направо: метод ближайшей ячейки, метод времени прибытия сигнала, разницы времени прибытия сигнала, амплитуды сигнала, угла прибытия сигнала, а также метод распознавания шаблонов, который был упомянут при использовании сотовых сетей связи. Самый простой из всех перечисленных – метод ближайшей

ячейки. При известном местоположении точки доступа можно определить положение мобильного устройства с точностью около 150 метров.

Методы, основанные на фиксации времени прибытия или отправки сигнала требуют точной синхронизации по времени. Расстояние между источником и приемником сигнала можно вычислить по точному времени прибытия сигнала (Time of Arrival – ToA). Зная скорость распространения сигнала и точное время сигнала в пути, можно вычислить расстояние. Пересечение минимум 3-х окружностей, очерченных расстояниями от базовых станций, дает точное положение мобильного клиента в 2-х мерном пространстве. Добавление 4-й станции может либо уточнить местоположение клиента, либо указать его положение в 3-х мерном пространстве.

Метод разницы во времени прибытия для источников/приемников (Time Difference of Arrival или TDoA) позволяет избавиться от необходимости точной синхронизации времени на приемниках в системе. Основная часть алгоритма опирается на математические вычисления, в частности на гиперболическую латерацию. Если известно расстояние между базовыми станциями и есть возможность зафиксировать время прохождения сигнала от клиента до каждого из передатчиков, для любой пары базовых станций можно построить гиперболу, на которой может находиться клиент. Точное положение клиента находится на пересечении гипербол. Чтобы зафиксировать двумерные координаты объекта – необходимо воспользоваться 3-мя базовыми станциями. Четыре станции позволят найти объект в трёхмерном пространстве.

Метод определения местоположения по амплитуде сигнала прост, но недостаточно надёжен. Он основан на определении потерь сигнала, которые включают в себя сумму составляющих: эталонная мера потери сигнала в dB на 1 метр, расстояние между источником и приемником с учетом затухания сигнала в измеряемой среде, а также другие помехи. К ненадёжности показаний приводит заранее неизвестная окружающая обстановка. Смена атмосферных показателей, стены и другие массивные объекты могут

приводить к значительному изменению характеристик угасания амплитуды и, соответственно, точности определения местоположения. То есть такой метод годится для использования в редко изменяющихся пространствах на открытых территориях.

Метод определения местоположения по углу прибытия сигнала, также называемый ангуляцией (Angle of Arrival – AoA), требует установки специальных секторных антенн, которые могут определить, откуда пришел сигнал. Для определения положения мобильного клиента достаточно получить сигнал от двух источников – точка пересечения направляющих лучей и даст местоположение клиента. Однако этот метод не учитывает случаев отраженных сигналов. В таких ситуациях определить местоположение объекта невозможно.

Ещё одна категория устройств, которые могут помочь в определении местоположения объекта – камеры видеонаблюдения. Например, для идентификации объектов автором одной из статей обозревается возможность использовать статистические методы распознавания образов [16]. Это позволяет разделять объекты на классы, выделять наличие чего-либо на монотонном фоне, выделение движущихся объектов. Другой автор предлагает метод переноса проекций контуров объекта, полученных с двух разных камер, на одну плоскость [17]. Задачу определение местоположения объектов в помещении на основе данных видеонаблюдения автор подразделяет на следующие подзадачи: обнаружение объектов на изображении, определение выпуклого многоугольника на предметной плоскости, являющегося пересечением минимальных выпуклых оболочек проекций одного и того же объекта полученных с разных камер, классификация объектов наблюдения. Также в такой системе необходима информация об эталонном объекте, тогда станет возможным определить угол поворота и координаты объекта. Положительная сторона метода с видеофиксацией в том, что при необходимости сразу видны параметры объекта: цвет (если камера

поддерживает цветной режим съёмки), размеры, иные свойства. Недостаток в том, что захват координат возможен только в поле зрения камеры.

1.3 ВЫВОД

Необходимый минимум для работы RFID системы – это считывающее устройство, антенна и метка на объекте. А также для реализации функционала разработки, в которую она интегрирована, существует привязка к информационной системе. Радиочастотные устройства и метки подразделяются на три диапазона: низкие, средние, и высокие (ультравысокие) частоты. Для использования мониторинга на расстояниях от нескольких до пары десятков метров подходят ультравысокочастотные сканеры и транспондеры. Использование активных меток позволяет расширить диапазон считывания до ста метров. Системы остальных диапазонов частот работают на очень малых расстояниях и не подходят для применения в рамках решаемых задач. Метки содержат несколько областей памяти, в том числе перезаписываемую для хранения какой-либо информации. Пассивные метки дешёвы и очень выгодны при большом тираже, но благоприятнее использование активных меток. Особенно учитывая тот факт, что активная метка может быть укомплектована каким-либо датчиком.

Определять местоположение объекта теоретически можно в радиочастотных системах. Но обычно это приводит к низкому качеству данных с погрешностью до нескольких метров. Особенно этот недостаток касается пассивных меток без встроенного источника питания, где отражённый сигнал в десятки раз менее интенсивен, чем исходящий от антенны считывателя.

Из всех аналогов достаточно хорошими параметрами для мониторинга обладает спутниковая связь трекера на устройстве и спутника на орбите Земли. Варианты с использованием систем мобильной сети сотовой связи и точек беспроводного доступа не подходят под условия решаемых задач, потому что не помогут в идентификации.

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Для реализации поставленных задач проекта требуются следующие базовые компоненты:

1. RFID-модуль считывания с возможностью подключения внешней антенны, а также с поддержкой последовательных интерфейсов для соединения с микроконтроллером или ПК, например, SPI или UART;
2. Микроконтроллер, который производит обмен данным не только с RFID считывателем по последовательному интерфейсу, но и с модулем подключения к сети интернет, например, WiFi-модулем;
3. Системы управления базами данных MySQL, обеспечивающая хранение данных о местоположении транспорта в реальном времени, а также историю маршрутов (полезная информация для диспетчеров);
4. Веб-сервер Apache предоставляющий работу с PHP скриптами. Сервер принимает данные от множества считывателей, расположенных в городе. С помощью веб-интерфейса предоставляется доступ к текущему местоположению машин. Также предоставляются расширенные данные (например, продолжительность остановки, общая длительность маршрута, время между двумя остановками) для диспетчеров, ответственных за определенный маршрут;
5. Графический интерфейс клиентов (для пассажиров и диспетчеров).

2.1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

В системе предусматривается следующий перечень участников: аппаратный комплекс, серверная часть, а также группа лиц, взаимодействующих с сервисом, то есть надзорные органы, пользователи СИМ, владельцы сервисов аренды СИМ (см. рисунок 2.1.1).

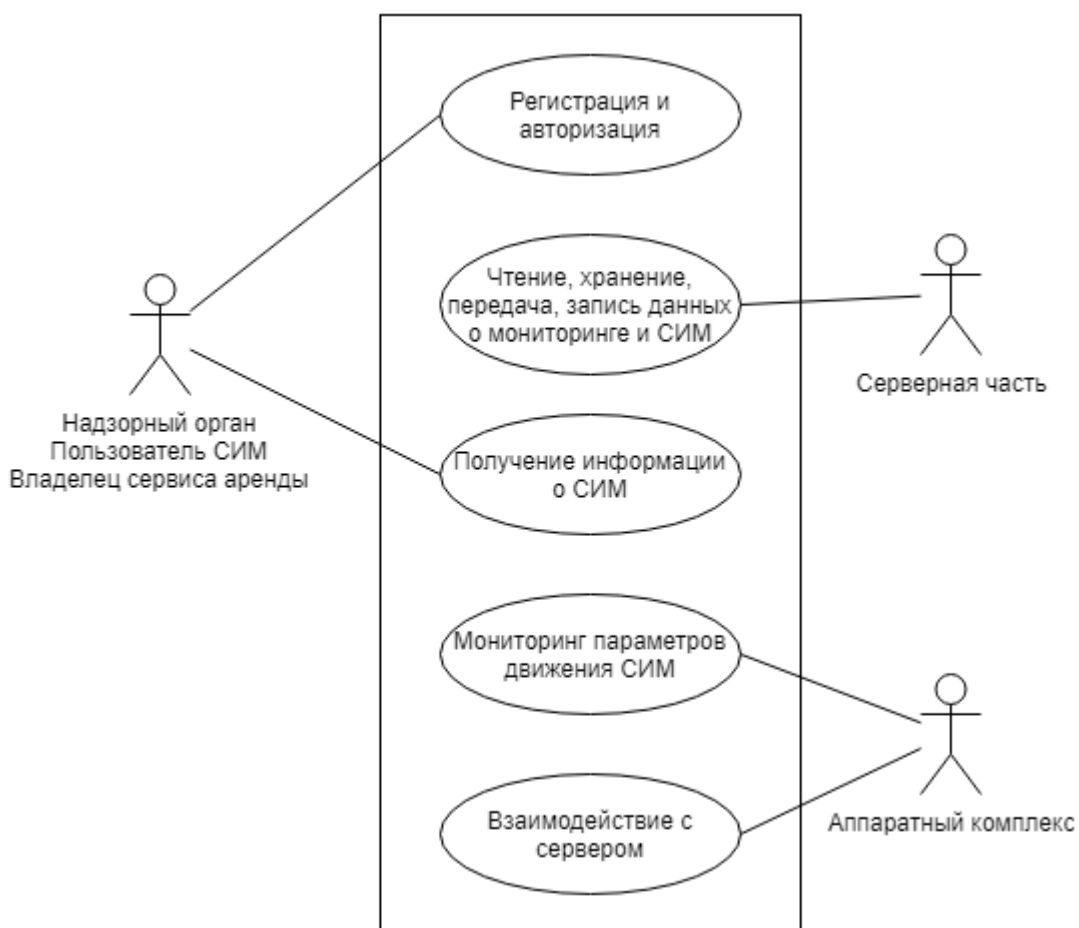


Рисунок 2.1.1 – Диаграмма вариантов использования системы мониторинга СИМ

Получается следующий перечень актеров:

1. Пользователь СИМ – арендатор, любое лицо, имеющее право и способное взять СИМ в аренду у некоторой компании-арендодателя.
2. Надзорный орган – министерство, которое берёт под свою ответственность вопросы регулирования движения СИМ в городе.
3. Владелец сервиса аренды – лицо, которое предоставляет во временное пользование свои СИМ арендаторам.
4. Серверная часть – компонент программно-аппаратного комплекса, отвечающий за чтение, хранение, передачу и запись данных о СИМ.
5. Аппаратный комплекс – компонент, ответственный за сканирование параметров СИМ и взаимодействие с серверной частью.

2.1.1 Реализуемые задачи

Перечень задач разрабатываемой системы следующий:

1. Идентификация СИМ подобно регистрационным номерам автомобилей и других транспортных средств.
2. Приблизительное определение местоположения СИМ в области сканирования.
3. Определение скорости движения СИМ.
4. Запись временной информации о текущем арендаторе или владельце на метку.
5. Ведение статистики по пользователям и СИМ об их возможных нарушениях.
6. Получение прочих параметров для расширения функциональности в будущем.

2.1.2 Требования к ПО

Программное обеспечение подразделяется на клиентов и сервер. Клиентскую часть составляют как пользователи информационной системы, так и аппаратное обеспечение. Клиенты в основном обращаются к сформированным данным, а аппаратная часть их регулярно изменяет или дополняет в соответствии с текущей дорожной обстановкой. То есть надзорный орган может получить статистику по средствам-нарушителям, пользователь СИМ – свою персональную статистику, а владелец сервиса – данные о своём парке устройств. Аппаратный комплекс сканирует идентификатор самого устройства и его параметры движения, а также вносит или стирает данные о текущем временном арендаторе.

Серверная часть должна оперировать данными о пользователях и средствах индивидуальной мобильности. Осуществлять привязку параметров движения и возможных фактов нарушения к СИМ и временным арендаторам, отвечать на запросы клиентов о выдаче определённого набора информации,

взаимодействовать со считывающими аппаратными комплексами на местах сканирования.

Теоретически предполагаемые функциональные возможности сервера по отношению к клиентам:

1. Выдача информации о СИМ.
2. Предоставление данных о запрашиваемом арендаторе или владельце.
3. Вывод различного вида статистики для некоторого сканирующего пункта.
4. Отправка команд считывателю о записи в метку информации о текущем арендаторе или владельце.

А также перечень возможностей аппаратных комплексов на местах считывания:

1. Чтение метки на СИМ.
2. Запись информации о временном владельце на метку.
3. Расчёт примерного местоположения СИМ.
4. Определение скорости движения.
5. Интеграция с сервисом видеофиксации нарушений.

2.2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

2.2.1 Условия эксплуатации

Место эксплуатации – город с населением более 1 миллиона человек с активно развивающимися сервисами проката средств индивидуальной мобильности. Климат умеренный, континентальный, средняя температура зимой около -15 градусов по Цельсию с минимумами примерно -45 градусов по Цельсию каждые 5-7 лет, средняя температура летом около 18 градусов по Цельсию с максимумами до 38 градусов аналогично раз в 5-7 лет. Зима умеренно-холодная, обычно длительная и снежная, метели наблюдаются в течение до 30-и дней. Лето сухое, умеренно теплое, редко – дождливое. Среднегодовое количество осадков – до 430 мм.

Электромагнитный фон осложнен наличием электрического транспорта (троллейбусы, трамвай), активного автомобильного трафика, наружных сетей электрических и другого типа коммуникаций.

2.2.2 Пользователи СИМ

Требования к пользователям включают в себя правила пользования сервисами проката [18]. Пользоваться СИМ должно дееспособное лицо, достигшее возраста 18 лет, по возможности использующее средства защиты, например, велосипедный шлем.

2.2.3 Характеристики сервера

Вычислительные мощности системы должны быть рассчитаны на обработку запросов от 2000 считывателей. Количество определено из расчета, чтобы в среднем на 1 км² приходилось не менее 4 считывающих комплексов. Расчет проведён для города Челябинска, площадь которого 530 км².

В будущем, при доработке правовой части регулирования вопросов, связанных с СИМ может потребоваться пространство для хранения данных регистрации и перемещения отдельно взятых самокатов и других средств мобильности за длительный период времени, от одного года.

2.2.4 Характеристики аппаратного комплекса

Аппаратные комплексы в городе должны безошибочно считывать проезжающие в окрестностях считывания СИМ. Для этого на чтение одной метки должно затрачиваться не более 1 мс, а также должны поддерживаться антиколлизийные алгоритмы. Этим критериям соответствует большинство современных устройств диапазона 860-960 МГц. Необходима поддержка определения RSSI метки для введения алгоритмов анализа движения метки.

2.2.5 Характеристики RFID-меток

Метки на СИМ должны быть защищены от проникновения и изменения к содержащейся в них информации третьими лицами. Это может обеспечиваться путём использования однократно записываемых меток, а также установки пароля на чтение и изменение информации на транспондере.

3 ПРОЕКТИРОВАНИЕ

В этой главе приводится работа над проектированием системы. Сначала рассматриваются варианты использования аппаратной части комплекса в зависимости от ситуаций. Затем приводится архитектура компонентов. После этого проектируются алгоритмы и вводится описание данных.

3.1 ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ ОБОРУДОВАНИЯ

Для решения задач мониторинга и отслеживания параметров движения СИМ в черте города предлагается ряд вариантов использования:

- 1) идентификация СИМ в городе;
- 2) интеграция с сервисом видео фиксации;
- 3) определение местоположения;
- 4) определение скорости.

Самая очевидная возможность радиочастотных технологий в данном контексте – идентификация малого транспортного средства. Задача размещения регистрационного номера на самокате или велосипеде с последующим считыванием его автоматическими оптическими комплексами может быть затруднена из-за малых габаритов. Здесь радиочастотная метка помогает безошибочно определить зарегистрированное СИМ (рисунок 3.1.1).



Рисунок 3.1.1 – Идентификация СИМ

Наличие только радиочастотной метки на индивидуальном транспорте скорее всего не поможет выяснить обстоятельства ДТП, если оно произойдет. Поэтому перспективной возможностью становится объединение видеофиксации и считывания на особенно оживлённых участках улиц. Такой метод поможет, потому что только видеофиксация при отсутствии меток бессильна в идентификации транспортных средств. Так как СИМ на дороге передвигаются в хаотичном потоке людей (рисунок 3.1.2), на тротуарах, и редко присутствуют специальные полосы для велосипедов и подобного транспорта.

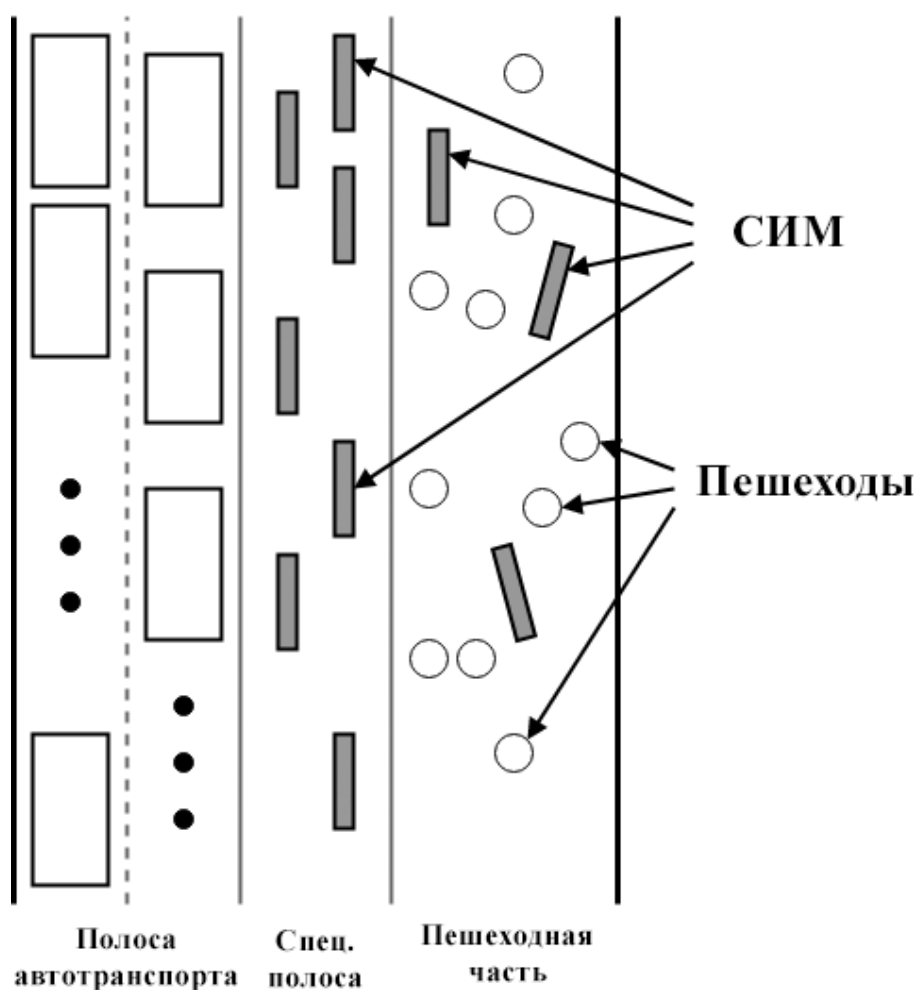


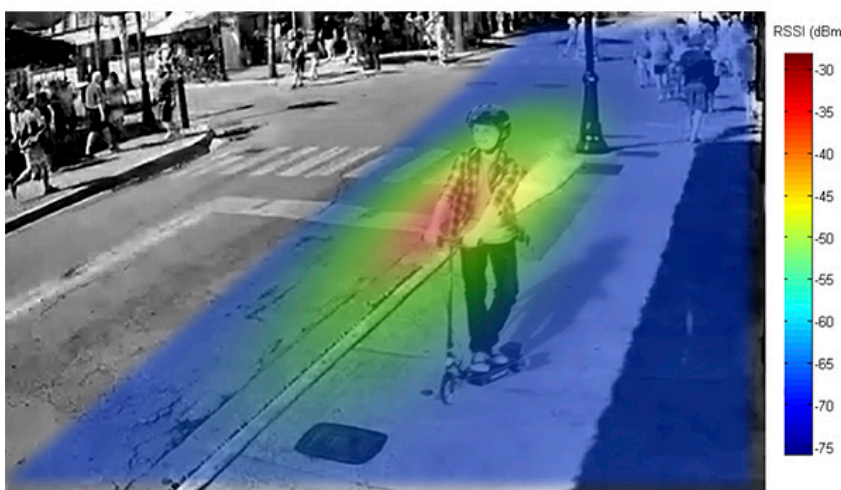
Рисунок 3.1.2 – Размещение СИМ на дороге в городе

Можно совместить пространство поля считывания с тем, что находится в кадре и наложить карту покрытия на изображение (см. рисунок 3.1.3). Это понадобится, чтобы затем при перемещении СИМ с меткой в поле зрения

объектива камеры появилась возможность на основании RSSI данной метки определить, что это именно тот самокат, который находится на изображении.



(a)



(б)

Рисунок 3.1.3 – Мониторинг с помощью камеры наблюдения (а), наложение карты RSSI в dBm от RFID-метки (б)

Также можно примерно предполагать о местоположении СИМ на улице. Из идеи следует два пути реализации: ToF (Time of Flight – время полёта), основанный на измерении промежутка времени между отправкой запроса на чтение и получением ответа (см. рисунок 3.1.3); а также на основе значения RSSI. В первом случае требуется наличие высокоточного таймера-счетчика. Чем выше разрешающая способность счетчика, тем более точным может быть определение.

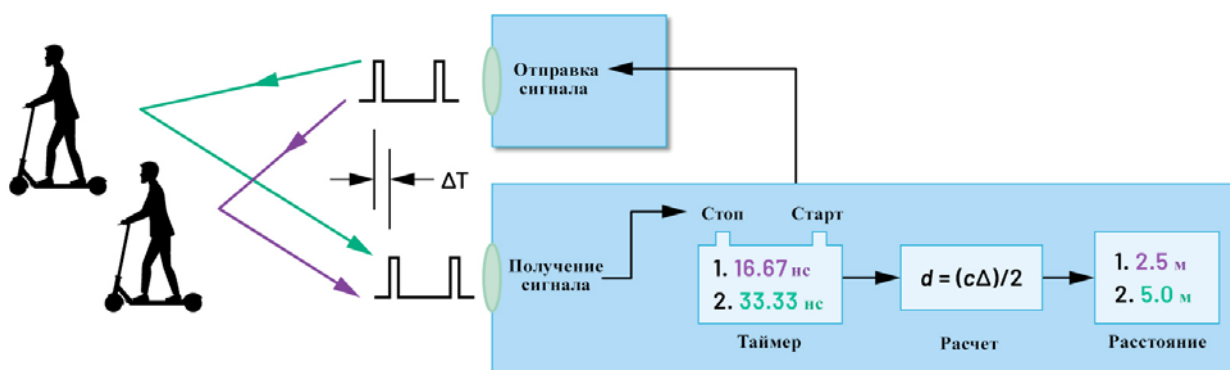


Рисунок 3.1.4 – Определение расстояния по временному промежутку между отправкой и приемом сигнала

При предполагаемом определении местоположения аппаратными комплексами на местах станет возможным вычисление скорости перемещения. Для этого необходимо выявлять местоположение метки в первый отрезок времени, а затем во второй. Полученная разница делится на временной интервал между двумя измерениями, и таким образом вычисляется скорость перемещения. Вторым вариантом определения этого параметра – разработка метки с датчиком скорости (см. рисунок 3.1.4). Наиболее распространено два типа таких сенсоров: шестерня датчиком Холла и оптопара с энкодером. При разработке активной радиочастотной метки, её микроконтроллер периодически, например, раз в секунду мог бы опрашивать датчик скорости, и фиксировать показания в пользовательскую область памяти.

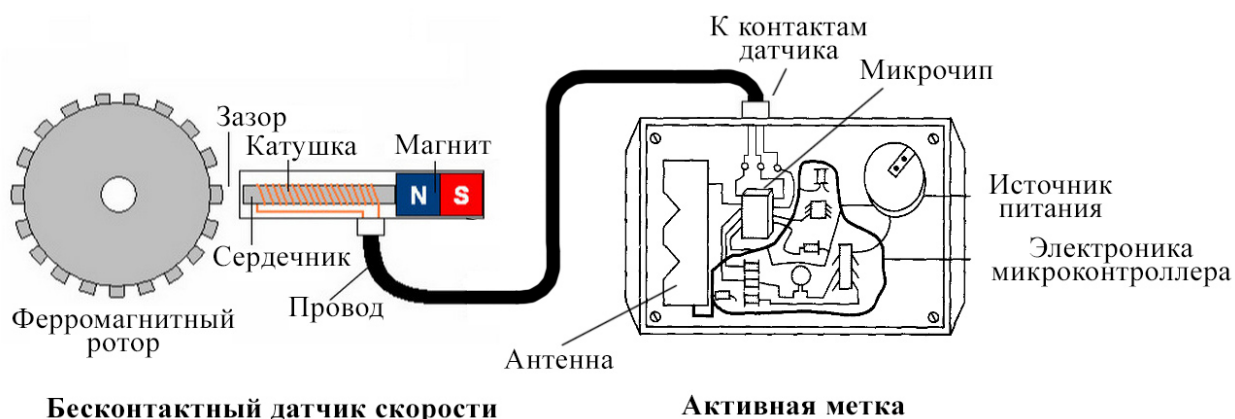


Рисунок 3.1.5 – Схематичная модель активной радиочастотной метки с датчиком скорости

3.2 АРХИТЕКТУРА РЕШЕНИЯ

Для функционирования системы понадобится следующий набор компонентов: считыватель, антенны, микроконтроллер, средство подключения к сети Интернет, а также веб-сервер, принимающий сканируемые данные. На рисунке 3.2.1 показана упрощенная топология минимально необходимых объектов для работы системы.

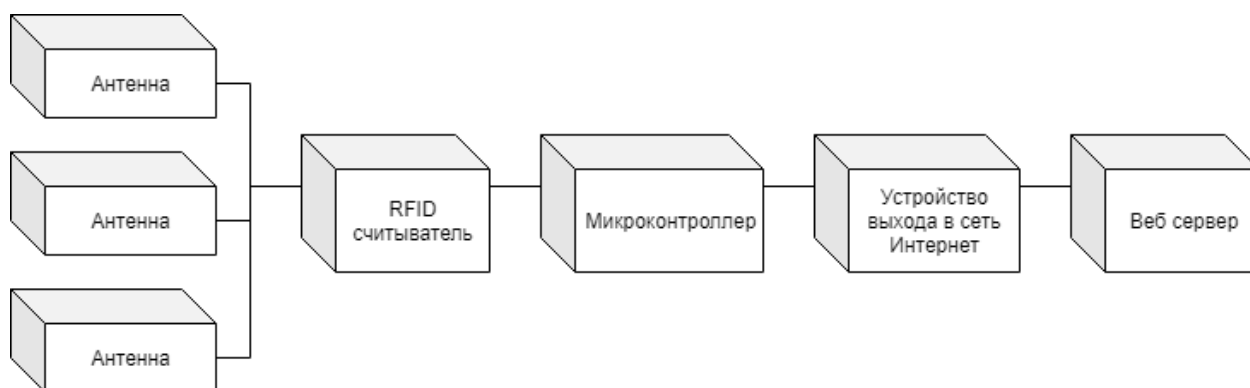


Рисунок 3.2.1 – Архитектура решения

Считыватель RFID с помощью одной или нескольких встроенных или внешних антенн получает данные, записанные на метках и обработанную информацию передаёт на микроконтроллер, который формирует запрос для сервера и передаёт его устройству выхода в интернет. Таким устройством может быть GSM или WiFi модуль, либо плата с разъёмом Ethernet.

В черте города вблизи общедоступных беспроводных точек доступа в сеть Интернет оптимальным решением будет использование модуля WiFi. Вблизи стационарных объектов, либо на внутренней территории предприятия можно использовать как WiFi, так и Ethernet интерфейсы. А для мобильных передвижных пунктов сканирования остаётся только использование GSM, так как зона покрытия WiFi для этих целей слишком мала.

Считывающее оборудование можно установить стационарно на отдельно стоящих конструкциях, но для предотвращения порчи внешнего облика города лучше всего размещать его на опорах уличного освещения (рисунок 3.2.2) или светофоров (рисунок 3.2.3). В помощь первому варианту можно производить установку на передвижных устройствах. Хорошим

вариантом будет использование маршрутных транспортных средств, выполняющих перевозку пассажиров, например, на автобусах и микроавтобусах (рисунок 3.2.4). При развитии технологий беспилотных летательных аппаратов можно сделать их носителями средств радиочастотной идентификации (рисунок 3.2.5).



Рисунок 3.2.2 – Считывающее оборудование на опоре освещения

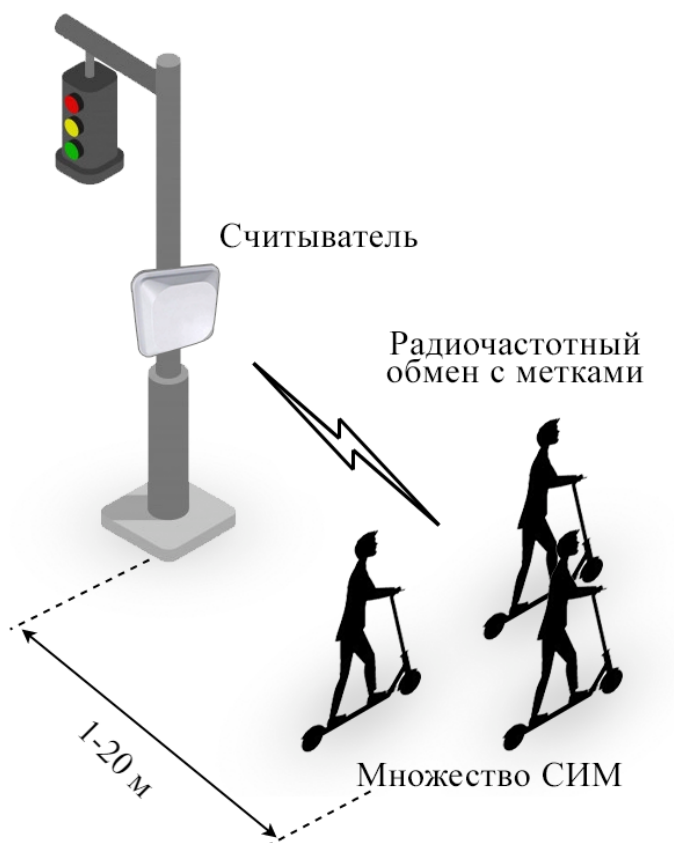


Рисунок 3.2.3 – Считывающее оборудование на опоре светофора

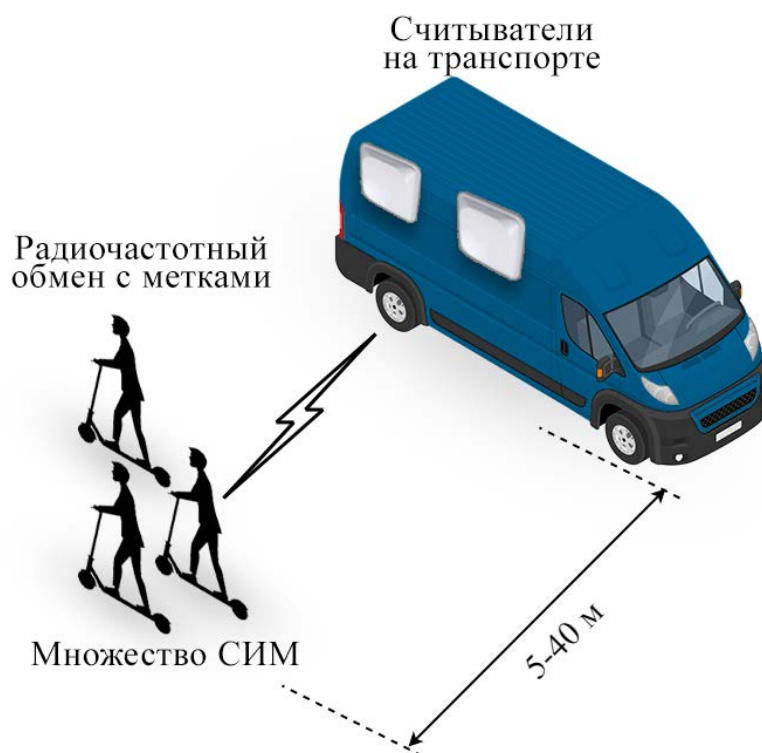


Рисунок 3.2.4 – Микроавтобус с антеннами и RFID-оборудованием



Рисунок 3.2.5 – Беспилотный летательный аппарат с RFID-считывателем

Таким образом, получается система в которой на местах постоянно работают средства радиочастотного мониторинга, сканирующие метки, и отправляющие идентификаторы СИМ на сервер. Сервер предназначен для принятия решений по модификации информации на метке, если это необходимо, о чем посылает ответный сигнал аппаратной части.

Наиболее осуществимыми на сегодняшний день являются способы установки считывателей на опоры освещения и светофоров или знаков. А также на городской маршрутный транспорт. Такой транспорт зачастую охватывает большинство городских улиц, где может быть скопление индивидуальных средств передвижения, что поможет отслеживать их параметры перемещения.

3.3 ОПИСАНИЕ ДАННЫХ

Чтобы протестировать аппаратный комплекс и его роль с системе идентификации СИМ необходима база данных на сервере, которая хранила бы набор минимально необходимых данных. Общая схема базы с использованием UML диаграмм представлена на рисунке 3.3.1.

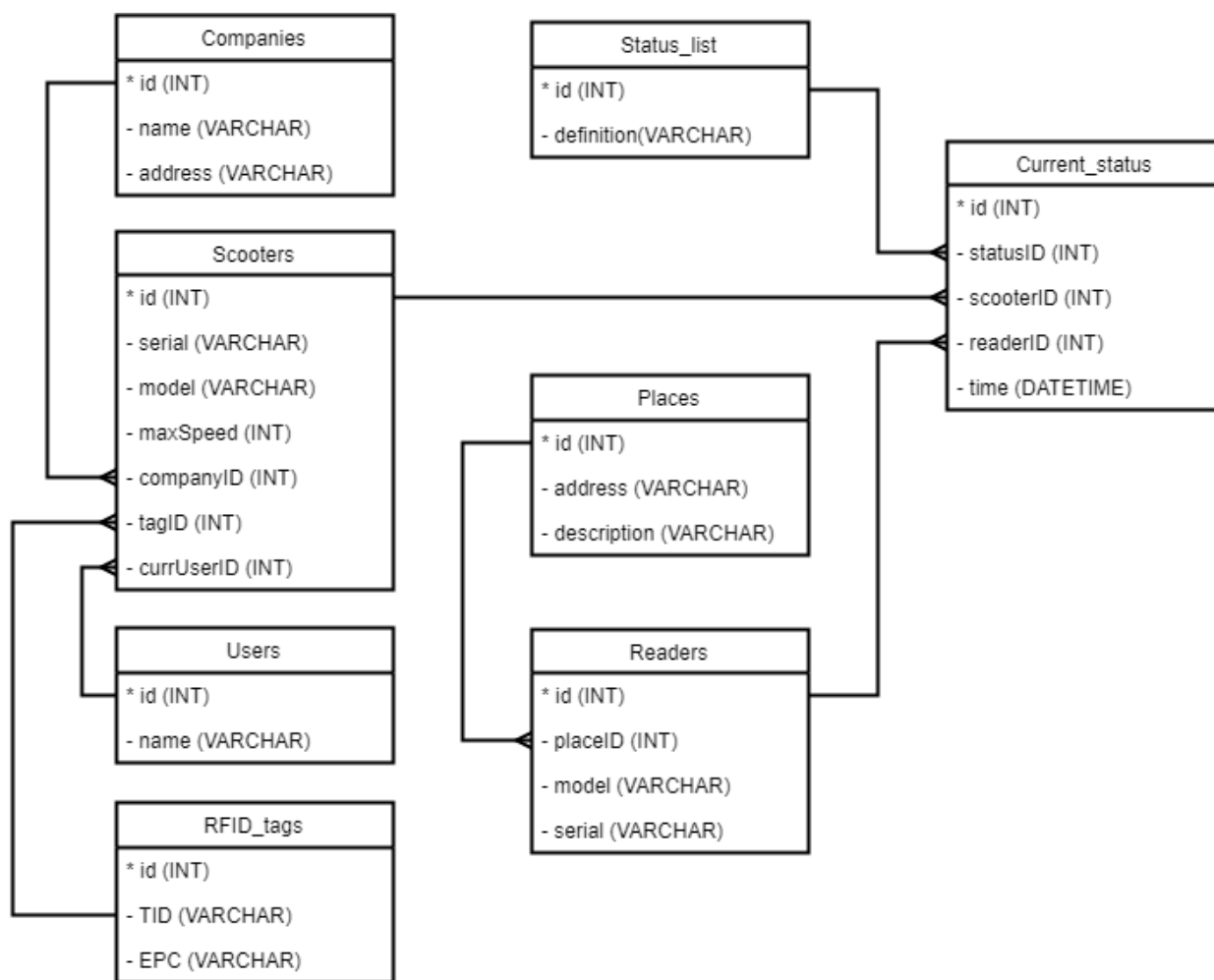


Рисунок 3.3.1 – Схема спроектированной базы данных

В базе данных используются следующие типы данных: Int (целочисленный), varchar (строка переменной длины), datetime (дата и время).

Информация о взаимодействующих с системой компаниях хранится в таблице «Companies», в которой содержатся минимально необходимые данные (см. таблицу 3.3.1).

Таблица 3.3.1 – Таблица «companies»

Название столбца	Тип данных	Назначение
Id	Int	Первичный ключ, уникальный идентификатор
Name	Varchar	Наименование компании
Address	Varchar	Адрес компании

Для хранения данных о СИМ требуется таблица «Scooters», в которой содержатся не только идентификационные данные каждого средства, но и привязка к компании, метке, временному пользователю (см. таблицу 3.3.2).

Таблица 3.3.2 – Таблица «Scooters»

Название столбца	Тип данных	Назначение
Id	Int	Первичный ключ, уникальный идентификатор
Serial	Varchar	Серийный номер средства
Model	Varchar	Модель средства
MaxSpeed	Int	Максимальная скорость
CompanyID	Int	Внешний ключ таблицы «companies», позволяет определить, какой организации принадлежит средство
TagID	Int	Внешний ключ таблицы «RFID_tags», осуществляет привязку СИМ к метке
CurrUserID	Int	Внешний ключ таблицы «Users», осуществляет привязку к пользователю или арендатору

Перечень всех пользователей системы перечислен в таблице «Users» (см. таблицу 3.3.3).

Таблица 3.3.3 – Таблица «Users»

Название столбца	Тип данных	Назначение
Id	Int	Первичный ключ, уникальный идентификатор
Name	Varchar	ФИО пользователя одной строкой

Перечень всех меток, используемых в системе перечислен в таблице «RFID_tags» (см. таблицу 3.3.4).

Таблица 3.3.4 – Таблица «RFID_tags»

Название столбца	Тип данных	Назначение
Id	Int	Первичный ключ, уникальный идентификатор
TID	Varchar	Содержимое
EPC	Varchar	Долгота, для местоположения

Таблица «Status_list» (см. таблицу 3.3.5) предназначена для перечисления видов состояния СИМ. На момент разработки и тестирования предлагается два состояния: Арендован, Свободен.

Таблица 3.3.5 – Таблица «Status_list»

Название столбца	Тип данных	Назначение
Id	Int	Первичный ключ, уникальный идентификатор
Definition	Varchar	Описание статуса

Для запоминания в системе всех мест, где установлены считыватели спроектирована таблица «Places» (см. таблицу 3.3.6).

Таблица 3.3.6– Таблица «Places»

Название столбца	Тип данных	Назначение
Id	Int	Первичный ключ, уникальный идентификатор
Address	Varchar	Адрес местоположения в формате город, улица, дом
Description	Varchar	Описание, где установлен считыватель по указанному адресу

Для учета самих считывателей, которые подключены к информационной системе, предназначена таблица «Readers» (см. таблицу 3.3.7).

Таблица 3.3.7 – Таблица «Readers»

Название столбца	Тип данных	Назначение
Id	Int	Первичный ключ, уникальный идентификатор
PlaceID	Int	Внешний ключ таблицы «Places», предназначен для привязки места к конкретному считывателю
Model	Varchar	Модель считывателя
Serial	Varchar	Серийный номер считывателя

Таблица «Current_status» (см. таблицу 3.3.8) необходима для отображения текущего состояния всех СИМ.

Таблица 3.3.8 – Таблица «Current_status»

Название столбца	Тип данных	Назначение
Id	Int	Первичный ключ, уникальный идентификатор
StatusID	Int	Внешний ключ таблицы «Status_list», для отображения состояния СИМ
ScooterID	Int	Внешний ключ таблицы «Scooters», для привязки самого СИМ
ReaderID	Int	Внешний ключ таблицы «Readers», для привязки последнего взаимодействующего с меткой считывателя.
Time	Datetime	Дата и время взятия в аренду или её завершения

3.4 АЛГОРИТМЫ РЕШЕНИЯ

Одна из задач – идентификация СИМ в городе. При разработке алгоритма необходимо учитывать разные модели поведения считывателей. В основном их две. Первая заключается в том, что микроконтроллер считывателя самостоятельно проводит регулярный опрос находящихся в поле считывания меток и при их наличии отправляет на порт ввода-вывода информацию о них. Как правило, передаётся уникальный идентификатор. Вторая модель поведения требует регулярной отправки команд считывателю на чтение (запись) того или иного блока памяти. Считыватель Fonkan FM-503 работает по второй модели.

После получения необходимой информации с метки (в рамках решаемой задачи идентификации это уникальный идентификатор метки и область пользовательской перезаписываемой памяти) данная информация отправляется на сервер (укрупненный алгоритм взаимодействия аппаратного комплекса и считывателя представлен на рисунке 3.3.1). Тот, в свою очередь, определяет, кто управляет данным СИМ, которому соответствует идентификатор метки, и возвращает идентификатор пользователя, либо команду о сбросе информации о пользователе, если средство после аренды оставили под считывателем и закончили аренду или пользователь на метке и в базе данных на сервере совпадают.

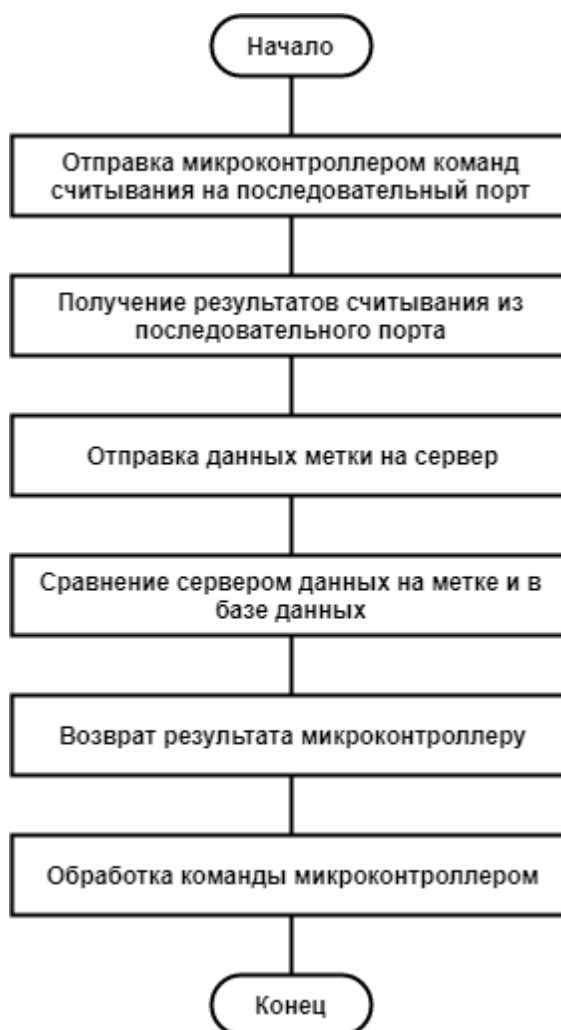


Рисунок 3.3.1 – Укрупненный алгоритм системы идентификации

При проектировании алгоритма аппаратного комплекса (рисунок 3.3.2) предусматривалась следующая последовательность действий: инициализация порта ввода-вывода в соответствии со скоростью порта считывателя (38400 бод/с); подключение к беспроводной сети доступа в Интернет; отправка последовательно команд на чтение пользовательской области памяти, затем уникального идентификатора метки с ожиданием ответа после каждой; обработка полученной от считывателя строки и отправка её на сервер; ожидание команды от сервера; при необходимости, перезапись информации на метке.

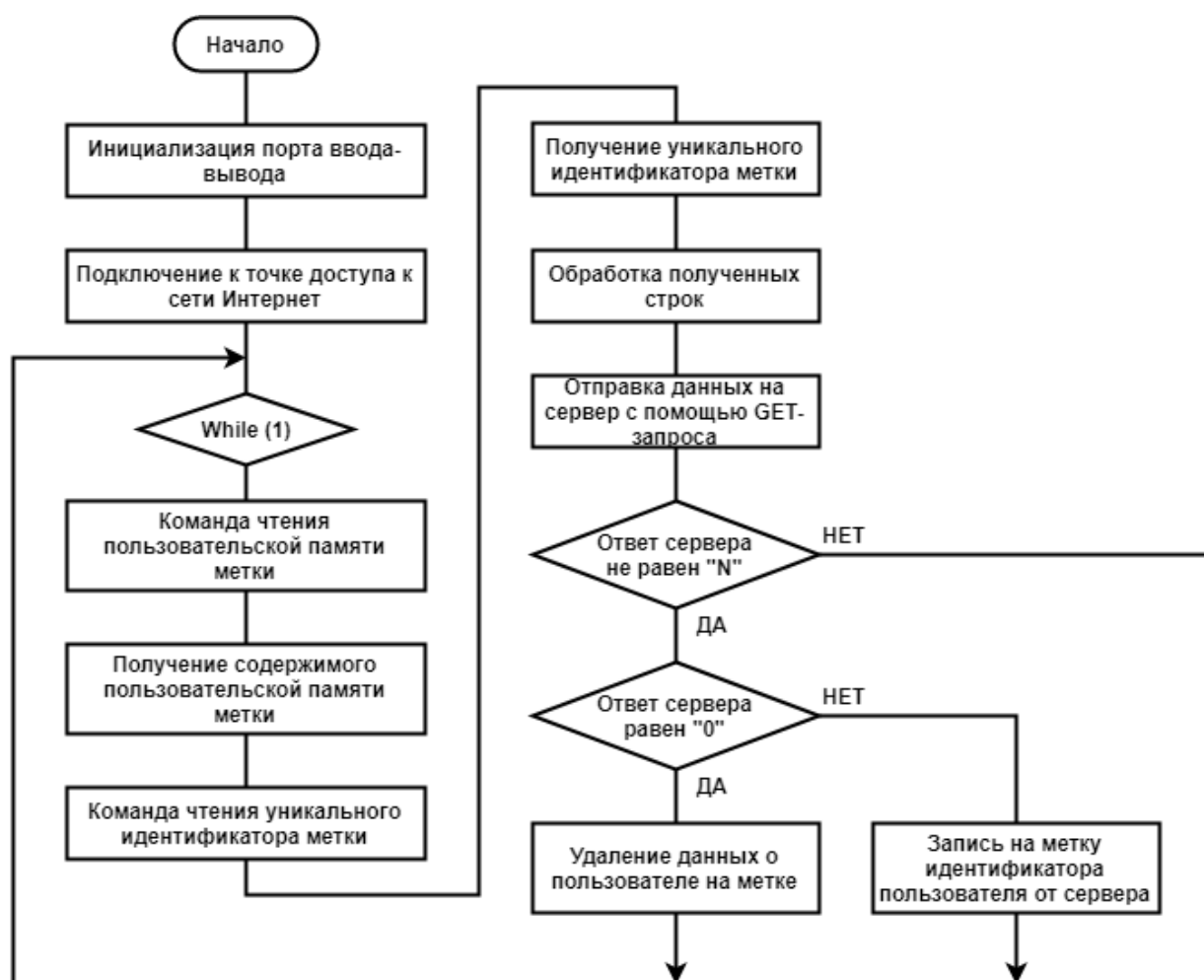


Рисунок 3.3.2 – Алгоритм подключенного к считывателю микроконтроллера

У сервера есть набор скриптов на языке php. Два из них (приложение Г и приложение Д) инициируют начало аренды и её завершение. Устроены они одинаковым образом: проверить наличие пользователя в базе данных запросом к таблице «Users» (см. таблицу 3.3.3), а также арендуемого СИМ в таблице «Scooters» (см. таблицу 3.3.2), затем, в случае начала аренды в таблице «Scooters» обновляется идентификатор пользователя, а в таблице «Current_status» (см. таблицу 3.3.8) для этого СИМ меняется статус на «Арендован» и обновляется временная метка. В случае завершения аренды в таблице «Scooters» идентификатор пользователя меняется на Null-значение, а в таблице «Current_status» для этого СИМ меняется статус на «Свободен».

Следующий скрипт (приложение Б) служит для проверки правильности внесения данных о СИМ при инициировании аренды, или при её завершении. Он использует один select-запрос к базе данных, с помощью которого из

таблиц «Companies», «Scooters», «Users» и «Status_list» производит выборку со следующими столбцами: модель скутера, название компании, состояние аренды, имя пользователя (может быть пустым, если состояние «Свободен»), временная метка последнего взаимодействия пользователя со скутером.

Последний скрипт сервера (приложение В) используется для вызова микроконтроллером, подключенным к считывателю. Он проверяет, находится ли запрошенное СИМ в аренде, и сравнивает идентификатор пользователя с метки, если таковой имеется, с аналогичным в базе (рисунок 3.3.3.). На основе этих сравнений возвращается либо идентификатор текущего пользователя, либо его сброс на метке, либо не совершать никаких действий, если информация на метке актуальна.

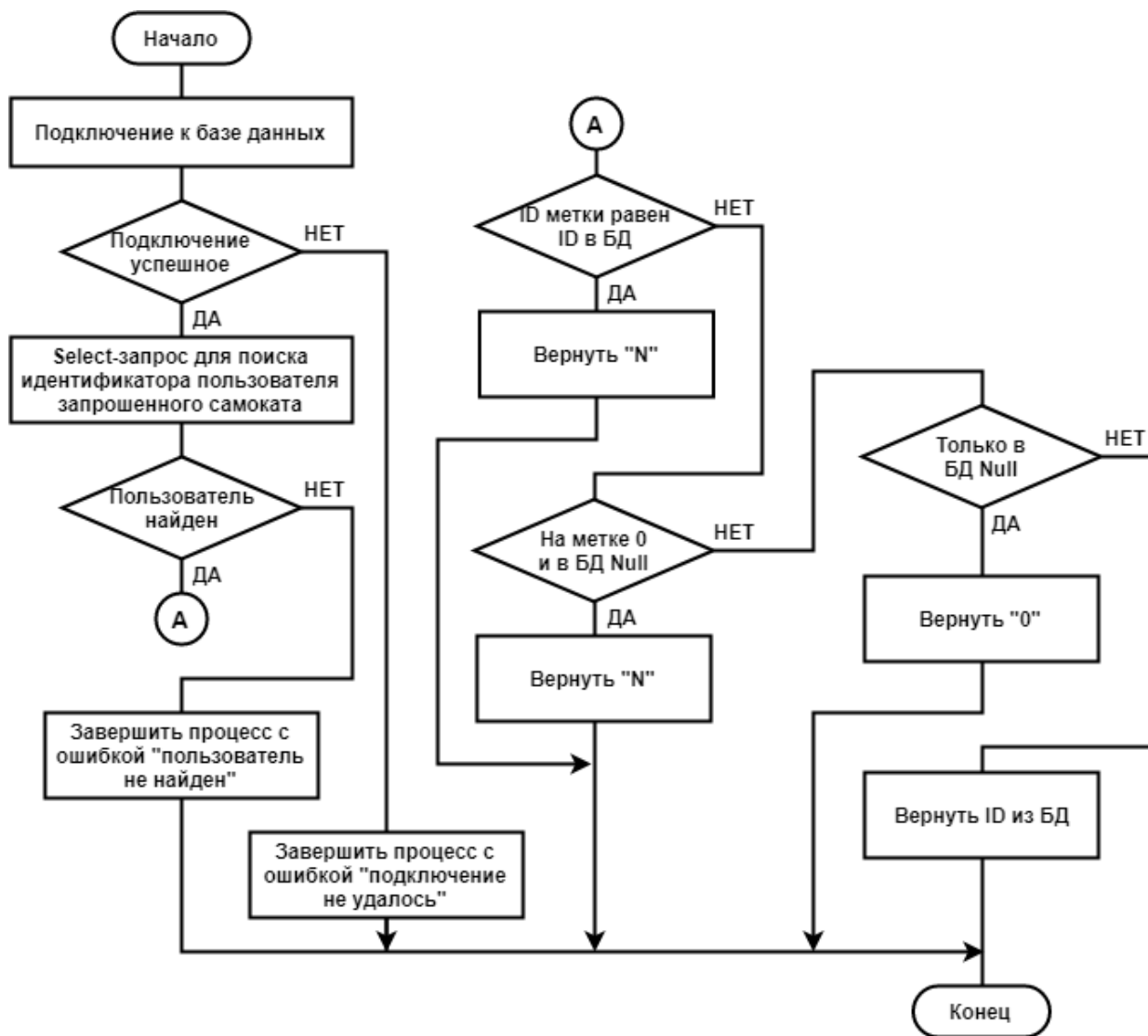


Рисунок 3.3.3 – Алгоритм скрипта проверки аренды

4 РЕАЛИЗАЦИЯ

Для моделирования работы системы и взаимодействия её ключевых компонентов реализован макет с применением макетной платы "NodeMCU" с микроконтроллером "ESP8266", с модулем в своём составе, поддерживающим возможность подключения к беспроводной сети WiFi, и радиочастотного считывателя меток FM-503 производства китайской компании Fonkan.

Для сервера осуществлена сборка, состоящая из веб-сервера Apache версии 2.4 [19], системы управления базами данных (СУБД) MySQL версии 8.0 [20] и языка PHP версии 7.3.2 [21].

4.1 РЕАЛИЗАЦИЯ МАКЕТА

Макет из перечисленных выше плат собирается и подключается к серверу для проверки работоспособности модели. Подробное описание компонентов далее в этом разделе.

Плата NodeMCU для логики и подключения к сети WiFi

NodeMCU – плата (см. рисунок 4.1.1) для разработки на базе чипа ESP8266 (версия ESP12E), который представляет собой WiFi модуль, подключаемый по UART протоколу. Данная плата позволяет упростить разработку устройств, которым необходимо соединение с сервером т.к. на ней уже реализованы все необходимые компоненты: разъём micro-USB, стабилизатор напряжения питания, выводы чипа разведены на контактные ножки со стандартным шагом 2.54 мм, что позволяет использовать устройство в макетной плате. Кроме того, плата поставляется с прошивкой NodeMCU, что позволяет программировать ее с помощью языка Lua или с помощью Arduino IDE. Полные технические характеристики перечислены в таблице 4.1.3.

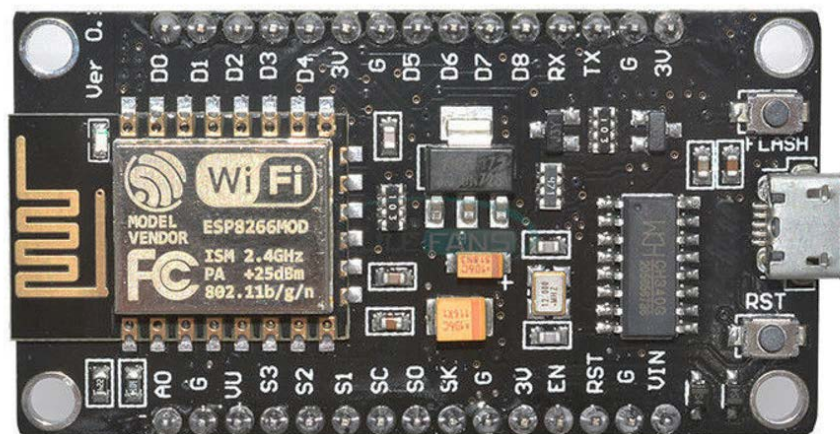


Рисунок 4.1.1 – Внешний вид платы NodeMCU

Таблица 4.1.1 – Технические характеристики NodeMCU

Параметр	Показатель
Напряжение питания	4.5 – 9 В
Потребляемый ток	Во время обмена – 70-200 мА Режим ожидания – 0.2 мА
Стандарт WiFi	802.11 b/g/n
Цифровые порты ввода-вывода	11 штук, некоторые из них могут быть сконфигурированы под потребности протоколов передачи данных
Аналоговые порты ввода-вывода	1 штука
Ток портов ввода-вывода	15 мА
Поддерживаемые протоколы	UART, I ² C, SPI
Рабочая температура	От -40 до +125 градусов Цельсия
Размеры	47 x 31 мм

Ультравысокочастотный считыватель Fonkan FM-503

Данный считыватель позволяет работать с метками стандарта ISO-18000-6C (ГОСТ ИСО/МЭК 18000-6-2013), то есть соответствующими диапазону 860-960 МГц, который планируется применять в реальной системе. Размер самой платы без учета выводов 60,4 на 88,4 миллиметров (рисунок

4.1.2). На ней размещены порты mini-USB и RS-232, разъем для одной антенны и для питания. А также множество посадочных мест под контакты 2.54 мм.

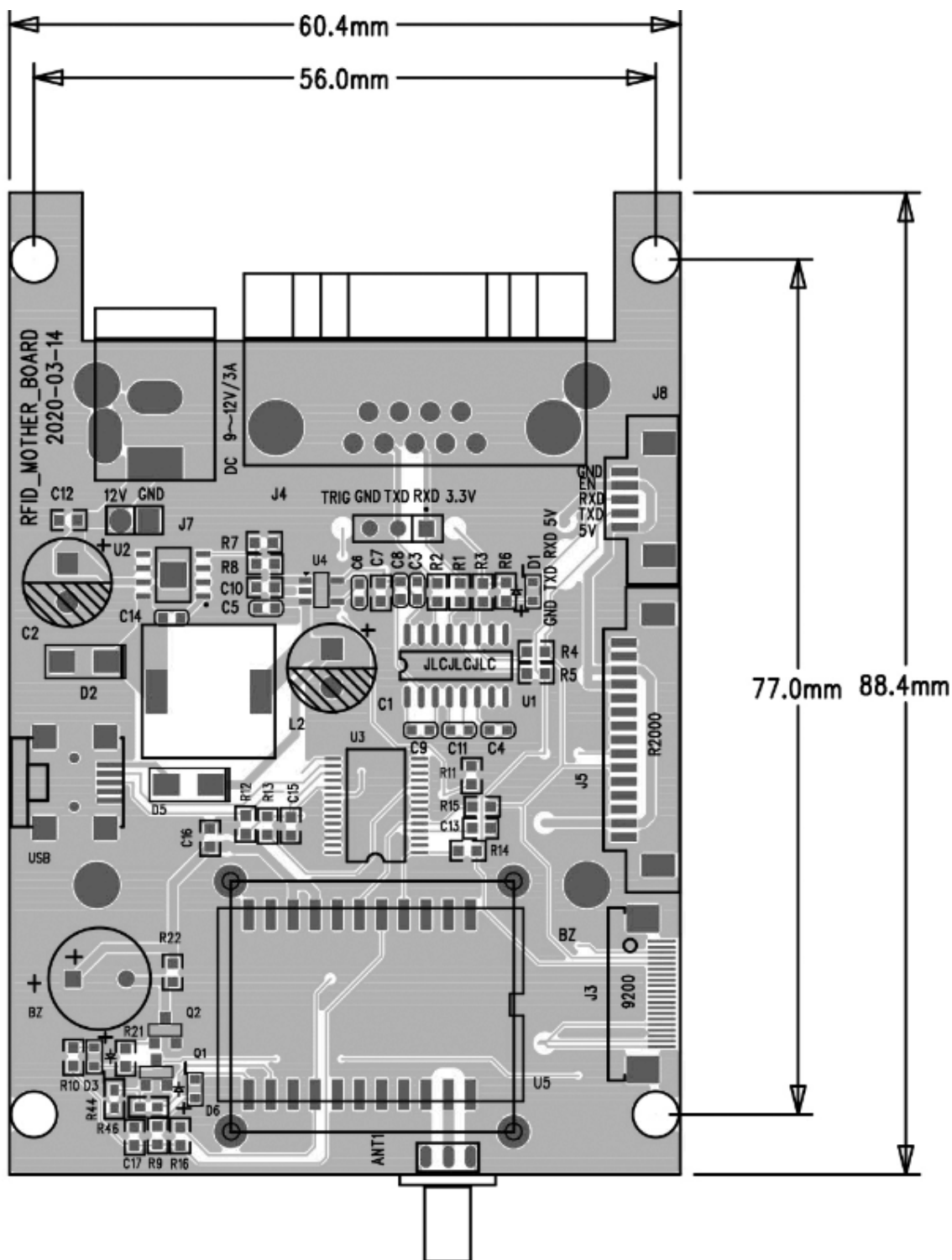


Рисунок 4.1.2 – Схема платы Fonkan FM-503

Перечень характеристик, указанных в таблице 4.1.2, удовлетворяет условиям выполнения проекта.

Таблица 4.1.2 – Характеристики считывателя Fonkan FM-503

Параметр	Показатель
Напряжение питания	5В при питании от USB, до 16 В при подключении к разъему питания
Потребляемый ток	Во время обмена – 280 мА Режим ожидания – 10 мА
RFID-протокол	ISO-18000-6C/ EPC class1 gen2, аналог ГОСТ ИСО/МЭК 18000-6-2013
Рабочая частота считывателя	865-868 МГц
Выходная мощность радиочастотного сигнала	26,5 dbm
Максимальное время радиочастотной сессии	0,5 мс
Максимальная скорость порта ввода-вывода	26 Кбит/с
Тип модуляции/демодуляции	Амплитудная
Поддерживаемые протоколы	UART
Рабочая температура	От -40 до +85 градусов Цельсия
Температура хранения	От -55 до +100 градусов Цельсия
Размеры	60,4 x 88,4 мм

Антенна считывателя FA-305A

Антенна, характеристики которой представлены в таблице 4.1.3, позволяет считывать и записывать информацию на метках. Она выполнена в монолитном керамическом корпусе, который зафиксирован на печатной плате из текстолита. С обратной стороны припаян провод с разъёмом SMA типа

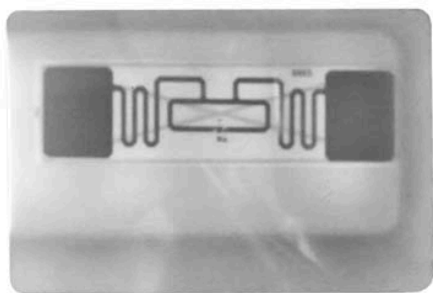
"папа". Заявленная дальность считывания с такой антенной до 4 метров по центральной оси.

Таблица 4.1.3 – Характеристики антенны FA-305A

Параметр	Показатель
Рабочая частота	865-868 МГц
Коэффициент усиления	5,5 dbi
Тип поляризации	Круговая
Разъём	SMA-мама
Материал	Печатная плата и керамика
Рабочая температура	От -40 до +85 градусов Цельсия
Температура хранения	От -55 до +100 градусов Цельсия
Размеры	120 x 120 x 5,8 мм

Радиочастотные метки

Имитация меток на мобильных передвижных устройствах производится с помощью меток-наклеек и карточек (рисунок 4.1.3). В действительно применяющихся системах карточки не будут задействованы, потому что это больше средство идентификации человека где-либо, электронный пропуск, но аналогичная со всеми остальными системами разметка памяти позволяет их применить для тестирования. А наклейки более приближены к реальной системе, потому что могут быть установлены на пластмассовые элементы СИМ. Типичные характеристики для пассивных меток диапазона 860-960 МГц приведены в таблице 4.1.4



(а)



(б)

Рисунок 4.1.3 – Метка-карта на просвет мощным источником освещения (а), метки-наклейки в специальном рулоне (б)

Таблица 4.1.4 – Характеристики меток

Параметр	Метка-наклейка	Метка-карта
Рабочая частота	860-960 МГц	
Стандарт	ISO18000-6C, EPC класс 1 Gen2	
Модель микроконтроллера	ImPinj M4QT	Higgs-3
Объем памяти пользователя	128 бит	
Дальность чтения и записи	До 6 м в зависимости от антенны считывателя	До 10 м в зависимости от антенны считывателя
Рабочая температура	От -30 до +85 градусов Цельсия	От -35 до +75 градусов Цельсия
Температура хранения	От -40 до +80 градусов Цельсия	
Размеры	50 x 50 мм	85 x 54 мм

Сборка макета

Базовый алгоритм простой: периодические запросы на считывание идентификаторов меток, отправка на сервер, и получение от него дальнейших инструкций, например, запись или удаление пользовательских данных с

метки. После написания кода программы для микроконтроллера и сервера следует сборка макета.

По схеме на рисунке 4.1.4 прослеживается, что микроконтроллер работает со считывателем по протоколу UART Universal Asynchronous Receiver-Transmitter – универсальный асинхронный приёмопередатчик), предназначение линий следующее:

- RX – Receiver. Приёмник входящего сигнала. Служит для получения данных;
- TX – Transmitter. Передатчик входящего сигнала. Служит для отправки данных;
- VIN – Voltage In. Входное напряжение;
- GND – Ground. Заземление.

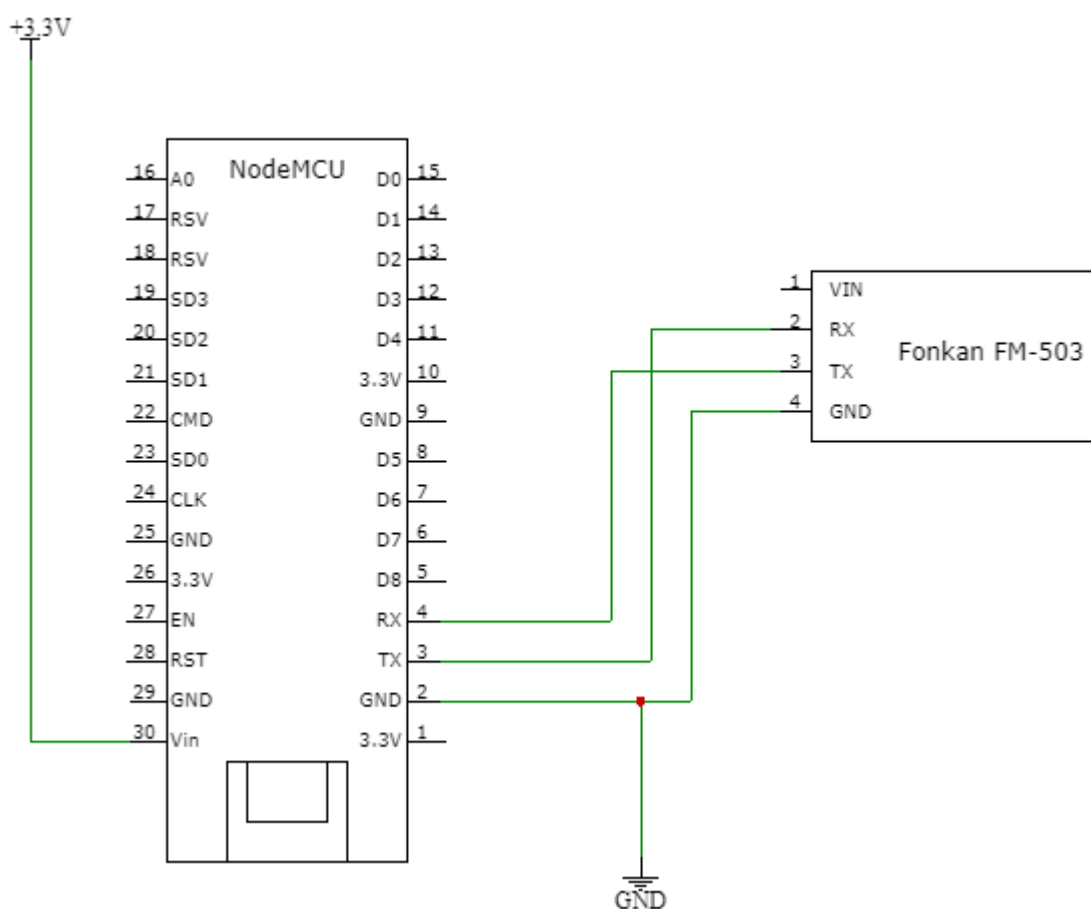


Рисунок 4.1.4 – Схема соединения компонентов макета

По данному интерфейсу микроконтроллер ESP8266 посылает команды считывателю, а тот, в свою очередь, передаёт ответы на запросы. Перечень необходимых для выполнения команд представлен в таблице 4.1.5.

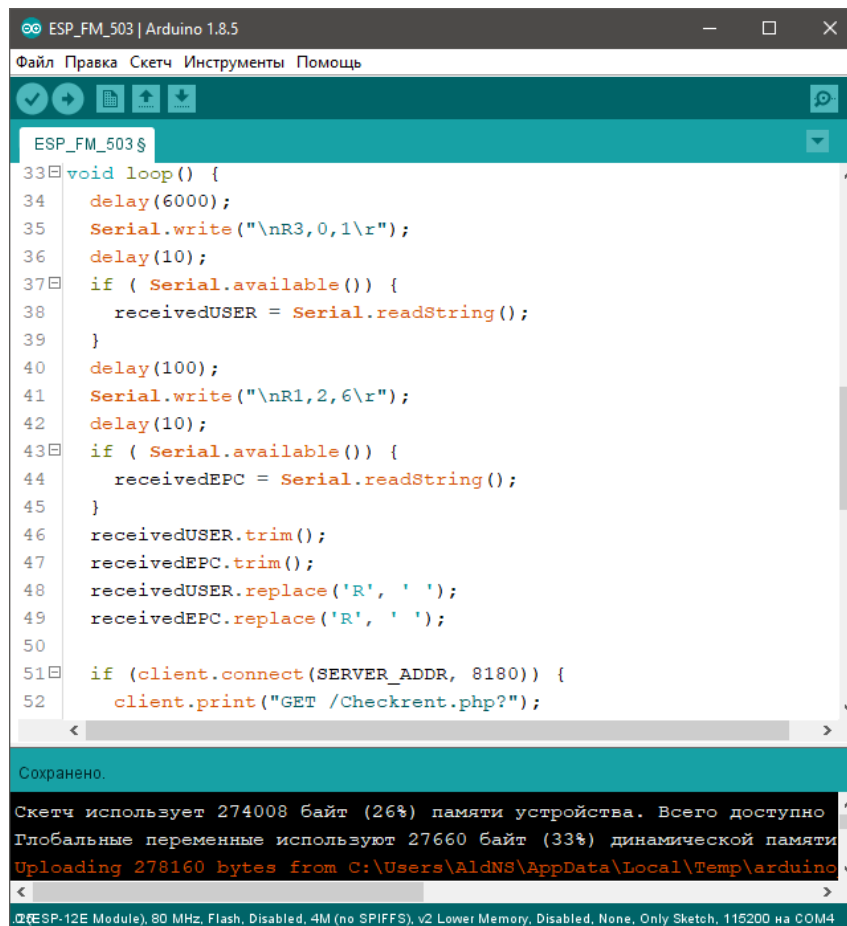
Таблица 4.1.5 – Команды считывателя FM-503

Формат команды и справка	Формат ответа и справка	Описание
Q	Q<ничего или EPC> Состав EPC: PC+EPC+CRC16	Отобразить идентификатор EPC
R<область>,<адрес>,<длина>	R<ничего или данные>	Считать область памяти метки
W<область>,<адрес>,<длина>,<данные> Области: 0: reserved 1: EPC 2: TID 3: Пользовательская Доступные адреса: 0 ~ 3FFF Длина данных: 1 ~ 1E	W<OK> или <код ошибки> Коды ошибок: 0: другое 3: переполнение памяти 4: память заблокирована В: Недостаточно мощности сигнала F: Неспецифическая ошибка	Записать в область памяти метки

Программирование

Для программирования микроконтроллеров используется Arduino IDE (англ. Integrated Development Environment – интегрированная среда разработки). С помощью неё, а также встроенных библиотек гораздо быстрее и удобнее разрабатывать алгоритм работы микроконтроллера, чем при использовании обычных текстовых редакторов и других средств разработки.

Программы, написанные в этой среде называются скетчами. Язык программирования похож по синтаксису на C-подобные. К тому же, в момент загрузки скетча в микроконтроллер, программа действительно конвертируется в C++ и компилируется. На рисунке 4.1.5 представлено окно программы Arduino IDE [22] во время написания скетча.



```
ESP_FM_503 $
33 void loop() {
34   delay(6000);
35   Serial.write("\nR3,0,1\r");
36   delay(10);
37   if ( Serial.available() ) {
38     receivedUSER = Serial.readString();
39   }
40   delay(100);
41   Serial.write("\nR1,2,6\r");
42   delay(10);
43   if ( Serial.available() ) {
44     receivedEPC = Serial.readString();
45   }
46   receivedUSER.trim();
47   receivedEPC.trim();
48   receivedUSER.replace('R', ' ');
49   receivedEPC.replace('R', ' ');
50
51   if (client.connect(SERVER_ADDR, 8180)) {
52     client.print("GET /Checkrent.php?");
```

Сохранено.

Скетч использует 274008 байт (26%) памяти устройства. Всего доступно
Глобальные переменные используют 27660 байт (33%) динамической памяти
Uploading 278160 bytes from C:\Users\AldNS\AppData\Local\Temp\arduino

ESP-12E Module), 80 MHz, Flash, Disabled, 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 на COM4

Рисунок 4.1.5 – Окно программы Arduino IDE

Использовались библиотеки «ESP8266WiFi» для подключения к сети Wi-Fi, «WiFiClient» для подключения к веб-серверу в качестве клиента, «Serial» для работы с портом ввода-вывода. В коде программы (приложение А) микроконтроллера ESP используются следующие функции:

1. Serial.Begin – относится к библиотеке «Serial», принимает один аргумент, скорость обмена информацией по последовательному порту ввода-вывода. Скорость установлена 38400 бод/с.
2. WiFi.Begin – относится к библиотеке «ESP8266WiFi», принимает два аргумента, наименование сети Wi-Fi и пароль для неё.
3. WiFi.Status – относится к библиотеке «ESP8266WiFi», не принимает аргументов, возвращает значение, подключен микроконтроллер к сети беспроводной передачи данных или нет.

4. `Serial.write` – относится к библиотеке «Serial», принимает в качестве аргумента строку, которую по байтам передаёт на последовательный порт.
5. `Serial.readString` – относится к библиотеке «Serial», возвращает строку, полученную из последовательного порта.
6. `Trim` – относится к библиотеке «String», удаляет из строки пробелы и специальные символы (табуляция, конец строки, перевод каретки и т.п.) в начале и в конце строки.
7. `Replace` – относится к библиотеке «String», заменяет символ, указанный в первом аргументе на символ, указанный во втором аргументе.
8. `Client.print` и `client.println` – относятся к библиотеке «WiFiClient», служат для формирования строки GET-запроса веб-серверу.
9. `Client.readString` – относится к библиотеке «WiFiClient», с помощью неё принимается от сервера ответ на запрос.
10. `IndexOf` – относится к библиотеке «String», возвращает индекс первого вхождения указанной в аргументе подстроки.
11. `Substring` – относится к библиотеке «String», вырезает из исходной строки подстроку, начиная включительно с индекса, указанного первым аргументом, и не включительно заканчивая индексом во втором аргументе.
12. `Sprintf` – функция языка C++, производящая вывод строки, указанной вторым аргументом в массив символов типа `char`, указанный первым аргументом.
13. `toInt` – относится к библиотеке «String», конвертирует число в строке в соответствующее значение целочисленного типа.

По окончании реализации некоторого заявленного функционала получилось устройство (рисунок 4.1.6), которое позволяет считывать уникальные идентификаторы меток, а затем отправляет данные на удалённый веб-сервер с помощью беспроводного подключения к сети Интернет. Сервер

получает данные идентификаторов и отправляет в ответ команды о необходимости изменить пользователя на метке, стереть идентификатор пользователя на метке, или не производить никаких действий над меткой. А также реализована серверная часть на основе веб-сервера Apache. Сервер может хранить данные о пользователях, СИМ, компаниях, владеющих СИМ, метках на СИМ, текущем состоянии самокатов, то есть арендованы они или нет. Также сервер может имитировать начало и отмену аренды СИМ при помощи соответствующих php-скриптов. Для просмотра текущего состояния на сервере создан скрипт, возвращающий таблицу с текущим состоянием каждого СИМ в системе.

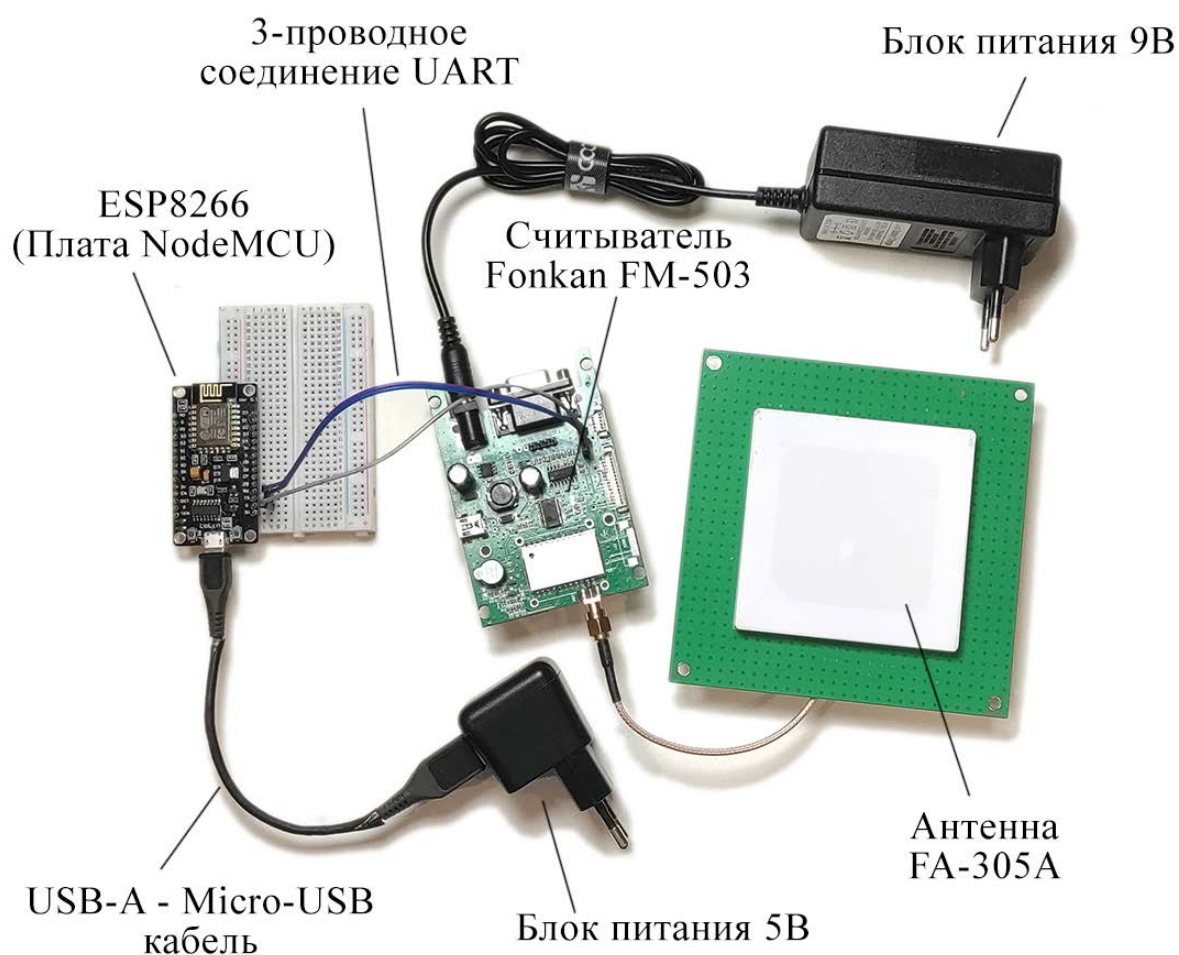


Рисунок 4.1.6 – Соединенные компоненты макета

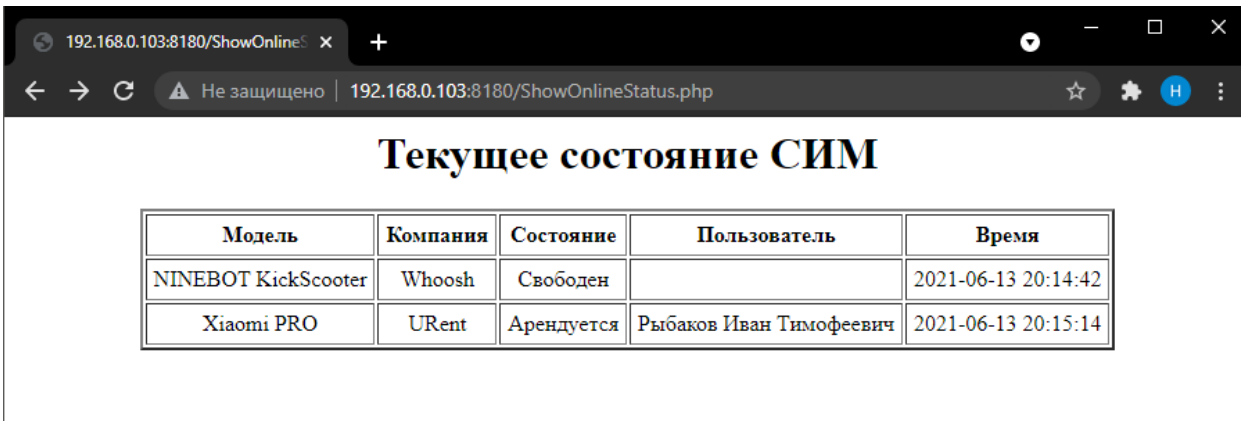
5 ТЕСТИРОВАНИЕ

5.1 МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ

Так как система представляет собой программно-аппаратный комплекс и в результате данной работы уделено большее внимание его аппаратной части и её взаимодействию с сервером, проводилось альфа-тестирование реализованного функционала. Альфа-тестирование – имитация реальной работы с системой разработчиком, либо реальная работа с системой потенциальными пользователями. Этот тип тестирования может проводиться как на ранней стадии разработки продукта, так и для законченного продукта в качестве внутреннего приёмочного тестирования. Иногда альфа-тестирование выполняется под отладчиком или с использованием окружения, которое помогает быстро выявлять найденные ошибки. Обнаруженные ошибки могут быть переданы тестировщикам для дополнительного исследования в окружении, подобном тому, в котором будет использоваться программа.

5.2 ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ

Для проверки работы скриптов запросы сначала проводились в веб-браузере. На рисунке 5.2.1 показан результат выполнения запроса на отображение текущего состояния СИМ. Первый самокат, записанный в систему, никем не занят, а последний пользователь закончил аренду 13 июня в 20:14. Второй арендуется вымышленным пользователем, Рыбаковым Иваном Тимофеевичем.



Модель	Компания	Состояние	Пользователь	Время
NINEBOT KickScooter	Whoosh	Свободен		2021-06-13 20:14:42
Xiaomi PRO	URent	Арендуется	Рыбаков Иван Тимофеевич	2021-06-13 20:15:14

Рисунок 5.2.1 – Текущее состояние СИМ

Затем проверяется в базе данных идентификатор Рыбакова Ивана Тимофеевича. Он равен 4, выделен на рисунке 5.2.2 среди остальных пользователей. А также проверяется идентификатор самоката, который он арендует (рисунок 5.2.3). Судя по отчету на рисунке 5.2.1, это самокат Xiaomi PRO, ему соответствует идентификатор метки 2 в базе данных. В таблице меток просматривается идентификатор второй метки (рисунок 5.2.4).

id	userName
1	Кудряшова Мирослава Артёмовна
2	Кузнецов Михаил Алексеевич
3	Касаткина Алёна Васильевна
4	Рыбаков Иван Тимофеевич
5	Ильина Алиса Алиевна

Рисунок 5.2.2 – Таблица пользователей

id	serial	model	maxSpeed	companyID	tagID	currUserID
1	8T1MWXRF354BOQJEGVPZLNUH	NINEBOT KickScooter	25	1	1	NULL
2	SZDPNC5UO8IB02V31KYXM6AW	Xiaomi PRO	25	2	2	4

Рисунок 5.2.3 – Таблица самокатов

id	TID	EPC
1	E280110C2000784B	300833B2DDD9014000000000
2	E280110C2000784E	300833B2DDD9014000000001

Рисунок 5.2.4 – RFID-метки с идентификаторами.

Далее проверяется скрипт, выполнение которого должен запрашивать микроконтроллер. На рисунке 5.2.5 представлена строка запроса с параметрами интересующего нас пользователя. Это идентификатор метки на его самокате (равен E280110C2000784E), а также его личный идентификатор пользователя (равен 4). Допустим, на метке уже был идентификатор пользователя, и вставим это значение в параметр запроса. На рисунке 5.2.5 видно, что сервер вернул ответ "N", что воспримется микроконтроллером как отмена каких-либо действий записи на метку.

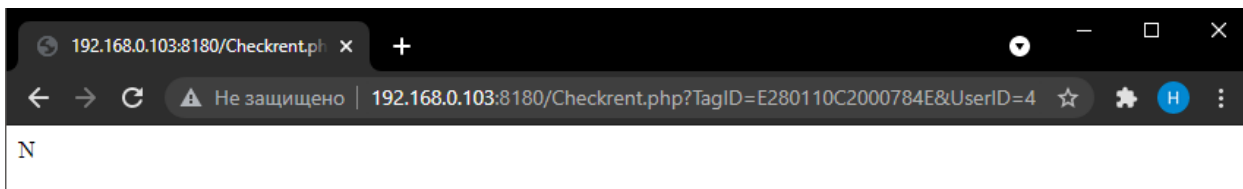
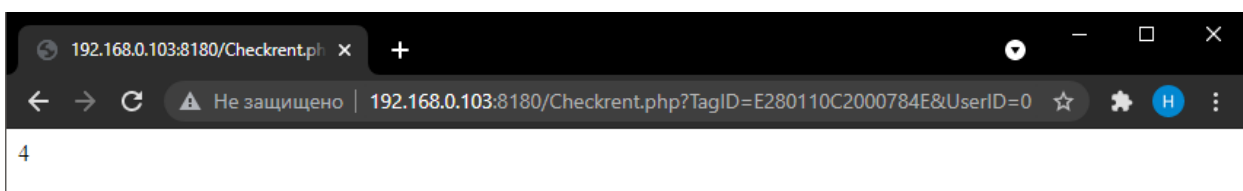
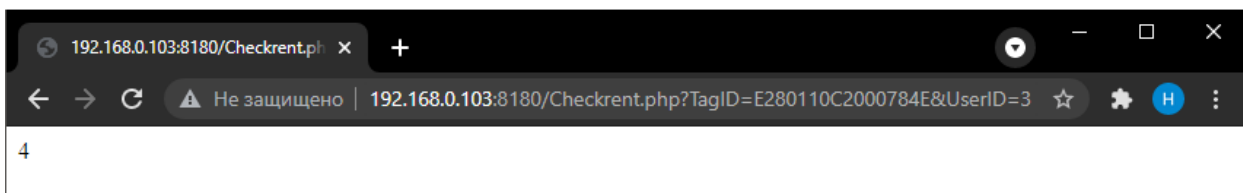


Рисунок 5.2.5 – Возврат значения "N" скриптом проверки аренды

Затем поочерёдно проверяется возвращаемое значение при указании нулевого значения с метки, а также идентификатора другого пользователя (рисунок 5.2.6). Это означает, что пользователь первый раз после начала своей аренды попадает в поле считывателя, а сервер возвращает идентификатор текущего пользователя, вследствие чего микроконтроллер, который получит ответ, перезапишет идентификатор на метке на правильный.



(а)

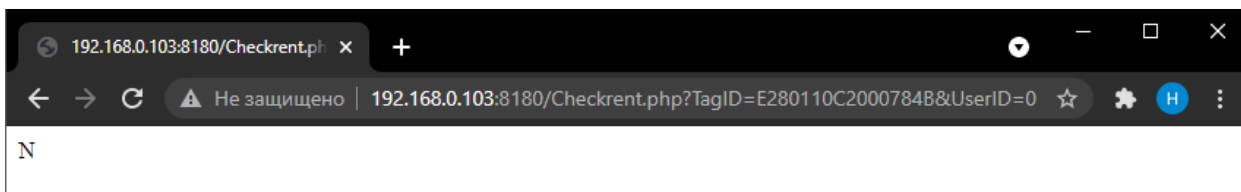


(б)

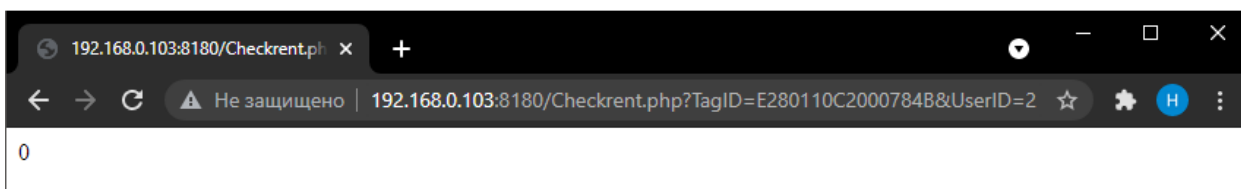
Рисунок 5.2.6 – Получение правильного идентификатора текущего пользователя при задании значения 0 с метки (а), при задании значения идентификатора предыдущего пользователя (б)

Точно таким же образом проверяется работа системы, если самокат не арендован. Такому самокату, по рисункам 5.2.3 и 5.2.4 соответствует метка с идентификатором E280110C2000784B. Проверяется работа алгоритма сначала заданием 0 в параметр идентификатора метки, а затем какого-либо идентификатора пользователя. В первом случае возвращаемое значение "N", это означает, что данные (их отсутствие) пользователя на метке соответствуют текущему состоянию самоката. Задание идентификатора пользователя 2

имитирует ситуацию, когда на метке он был записан, но сейчас его необходимо удалить, о чём говорит возвращаемое нулевое значение.



(а)



(б)

Рисунок 5.2.7 – Получение ответа "N" при условии свободного самоката и отсутствия данных пользователя на метке (а), получение нуля при задании значения идентификатора предыдущего пользователя (б)

Случай взятия самоката в аренду имитируется скриптом Rent (приложение Г). В случае на рисунке 5.2.8 берётся в аренду первый самокат первым пользователем из базы данных. На рисунке 5.2.9 статус самоката обновился, и появился пользователь (отсутствие пользователя у первого самоката подтверждается рисунком 5.2.1).

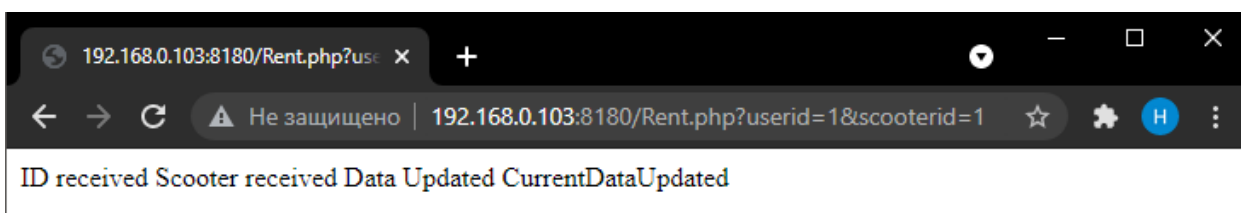


Рисунок 5.2.8 – Выполнение скрипта взятия в аренду

Модель	Компания	Состояние	Пользователь	Время
NINEBOT KickScooter	Whoosh	Арендуются	Кудряшова Мирослава Артёмовна	2021-06-14 22:58:23
Xiaomi PRO	URent	Арендуются	Рыбаков Иван Тимофеевич	2021-06-13 20:15:14

Рисунок 5.2.9 – Текущее состояние СИМ после взятия в аренду первого в таблице самоката

Затем тестируется скрипт Unrent (приложение Д), который имитирует завершение аренды самоката. Например, на рисунке 5.2.10 Рыбаков с идентификатором 4 завершает аренду своего самоката Xiaomi с идентификатором 2. На рисунке 5.2.11 в текущем состоянии СИМ этот самокат освобожден.

ID received Scooter received Data Updated CurrentDataUpdated

Рисунок 5.2.10 – Выполнение скрипта завершения аренды

Модель	Компания	Состояние	Пользователь	Время
NINEBOT KickScooter	Whoosh	Арендуются	Кудряшова Мирослава Артёмовна	2021-06-14 22:58:23
Xiaomi PRO	URent	Свободен		2021-06-14 23:07:17

Рисунок 5.2.11 – Текущее состояние СИМ после снятия аренды со второго в таблице самоката

Для проверки, что микроконтроллер ESP8266 получает данные от сервера и формирует правильную строку записи для считывателя создан

тестовый код (приложение Е), где на последовательный порт посылается вся информация о состоянии платы в данный момент времени. На мониторе последовательного порта видно, что в условиях, когда микроконтроллер посылает данные, которые предположительно могли быть получены с метки, когда самокат на самом деле никем не занят, он формирует правильную строку записи нулевого значения в область пользовательской памяти.

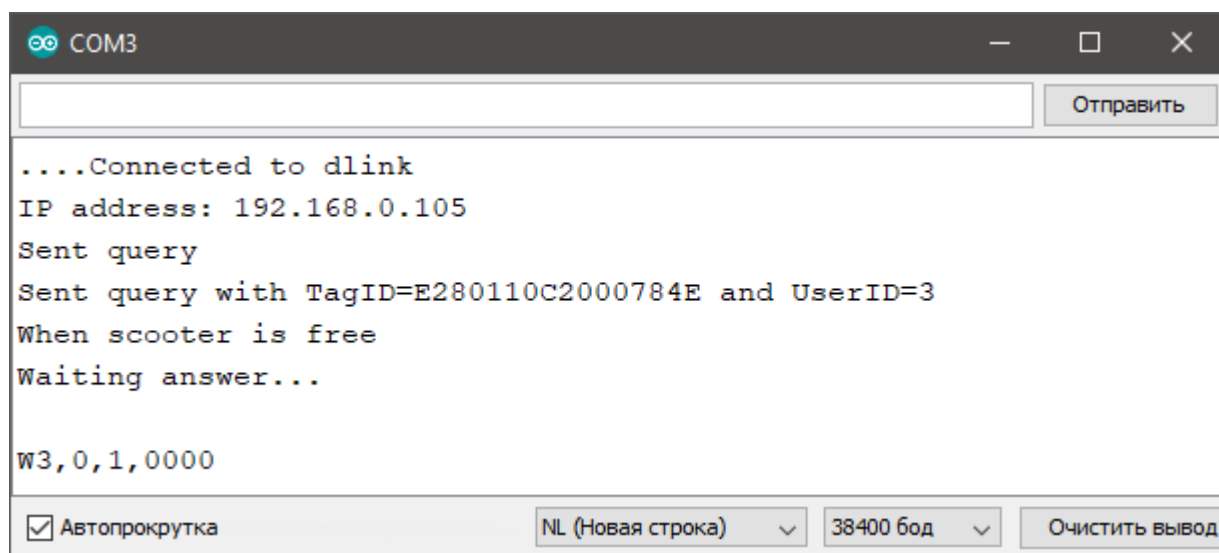


Рисунок 5.2.12 – Монитор последовательного порта среды Arduino IDE

Для проверки, что на метке записывается верная информация была проведена следующая процедура:

1. Самокат с идентификатором 2 был в аренде у пользователя с идентификатором 4.
2. На метке отсутствовала информация о пользователе перед чтением, что было проверено программой для работы со считывателем, поставляемой производителем (рисунок 5.2.13).
3. Микроконтроллер был подключен к считывателю, а метка поднесена в поле считывания.
4. Через несколько секунд считыватель снова был подключен к программе производителя, а на метке проверена пользовательская область памяти, на которой появился идентификатор пользователя, который арендует самокат (рисунок 5.2.14).

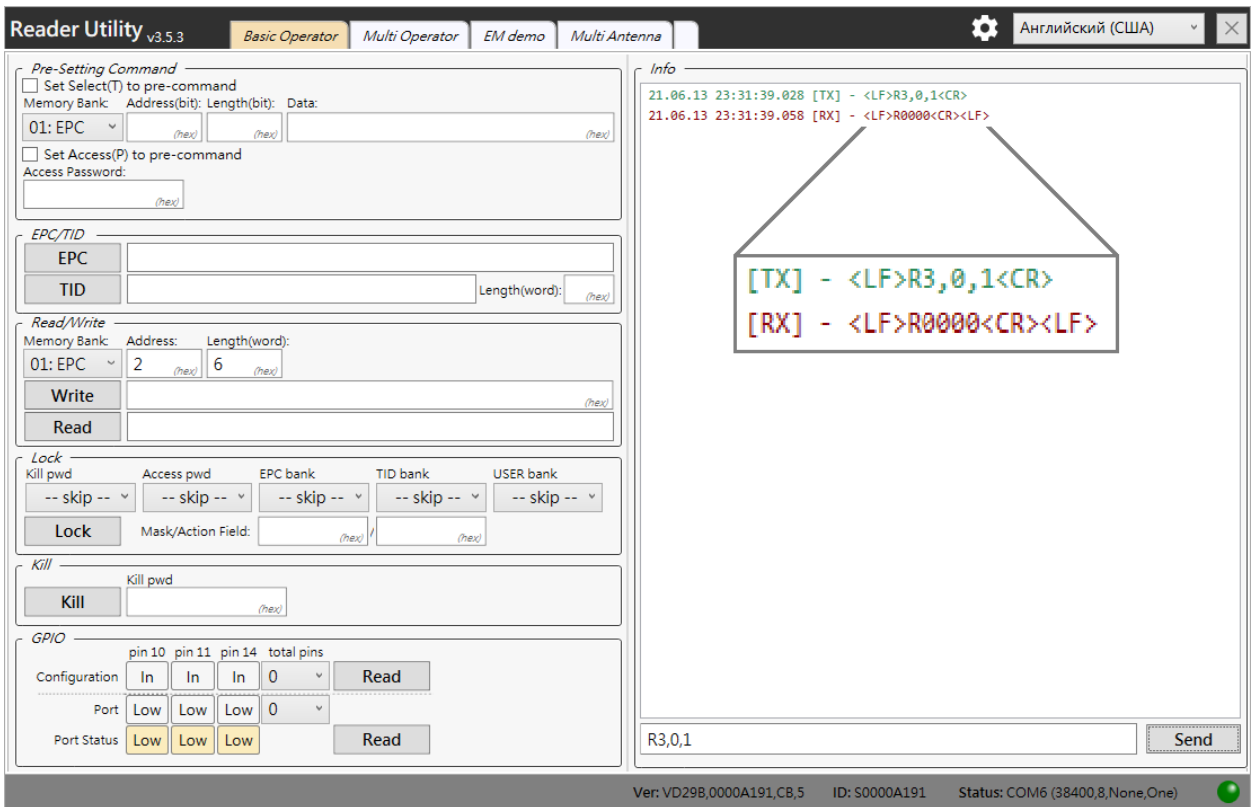


Рисунок 5.2.13 – Окно программы Reader Utility до записи идентификатора пользователя в метку

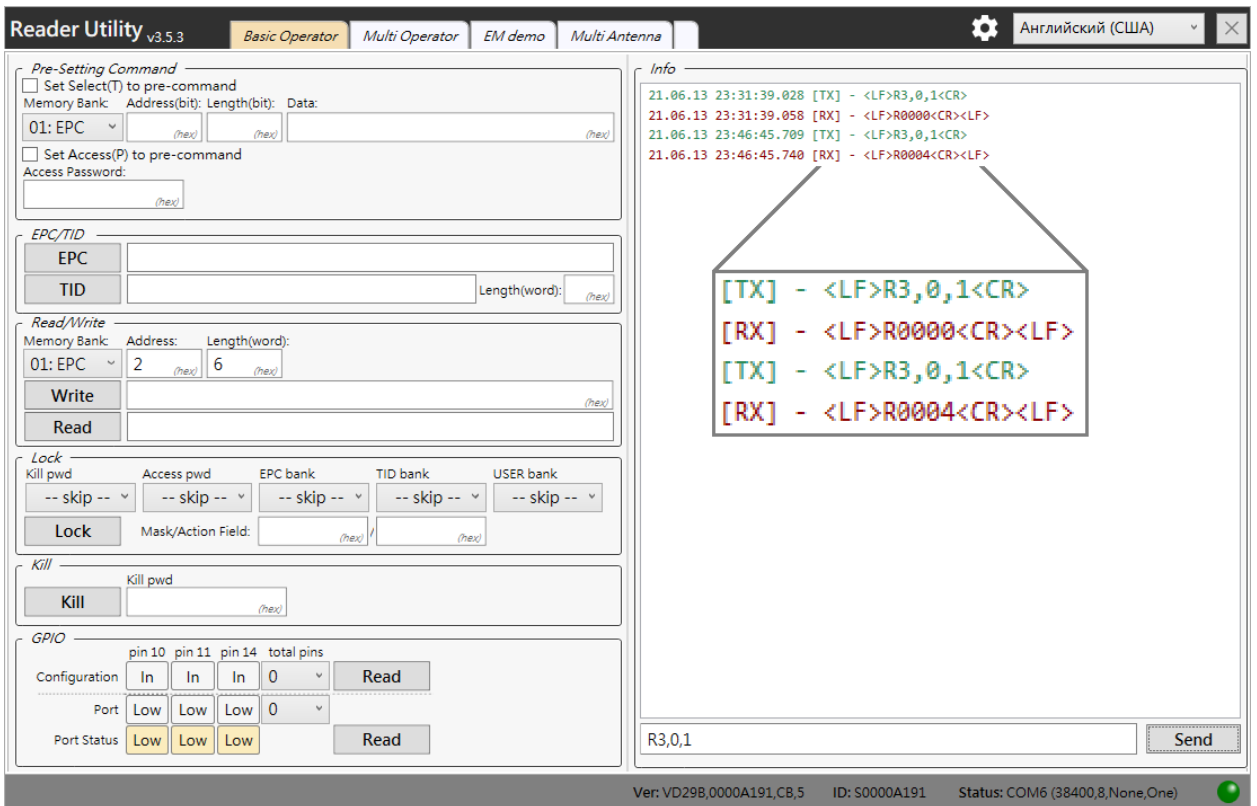


Рисунок 5.2.14 – Окно программы Reader Utility после записи идентификатора пользователя в метку

На рисунке 5.2.15 изображен монитор порта Arduino IDE во время выполнения основной программы (приложение). На нём видно, как сначала посылается по две команды чтения пользовательской памяти и идентификатора метки, а в момент поднесения метки в зону считывания отправляется команда записи в пользовательскую область памяти идентификатора текущего арендатора самоката. Это означает, что на метке до этого была неправильная информация о пользователе. Затем функция записи не вызывалась даже при нахождении метки в зоне считывания, так как информация на ней стала актуальной.

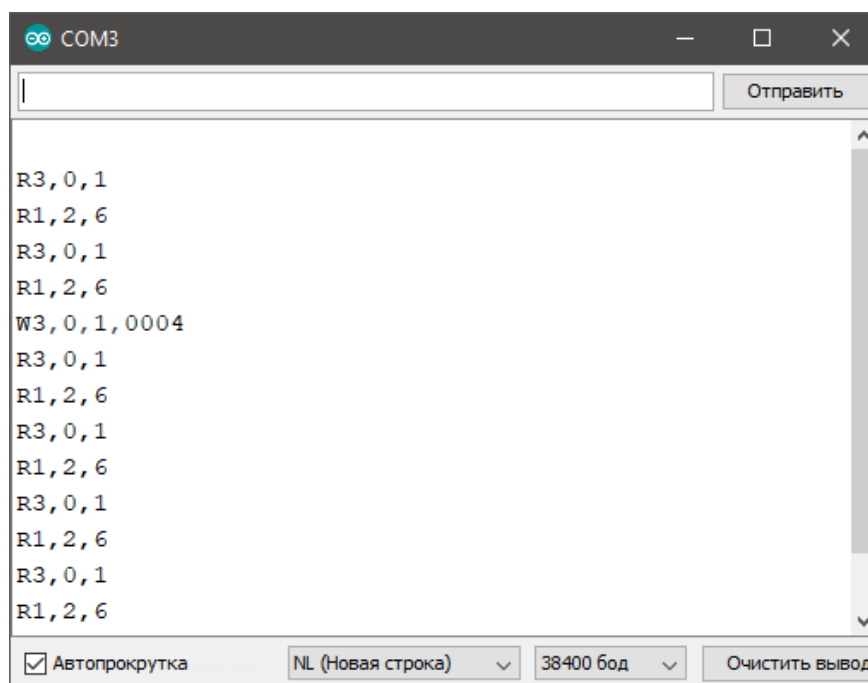


Рисунок 5.2.15 – Монитор порта во время выполнения основной программы

В результате альфа-тестирования можно сделать вывод, что система работает как предполагалось: считыватель получает информацию как от одной, так и от множества меток, данная информация поступает на устройство связи (плату с модулем WiFi), которое отправляет запрос на сервер. Веб-сервер, в свою очередь, принимает запросы и отклоняет заведомо ошибочные (с неправильными параметрами, либо вообще не связанными с системой данными).

ЗАКЛЮЧЕНИЕ

Данная работа заключалась в проектировании и реализации системы мониторинга объектов в городе с помощью мобильных технологий радиочастотной идентификации. По итогу выполнения выполняется задача отслеживания местоположения арендуемых средств индивидуальной мобильности (СИМ), то есть, велосипедов, самокатов, и подобного малого транспорта в черте города с помощью радиочастотных меток.

Радиочастотные метки являются хорошей альтернативой государственным регистрационным знакам автомобилей, мотоциклов и прочей техники. Преимущество меток в том, что нет необходимости визуального контроля достаточно малых мобильных устройств передвижения. Их корпуса достаточно малы для установки регистрационных номеров, а владельцы или арендаторы двигаются в хаотичном потоке на тротуарах, а не по полосам.

Было предложено устанавливать считыватели на опорах освещения, светофоров, знаков, а также на транспорте (автобусах и микроавтобусах). В настоящий момент – это наиболее реализуемый вид размещения считывающего оборудования. Идея размещать считыватели на беспилотных летательных аппаратах пока ещё далека от реализации в силу их физических возможностей: малого времени полёта от одного заряда аккумулятора, и небольшой грузоподъёмности большинства моделей.

По итогу выполнения был собран макет с применением считывателя FM-503 производства компании Fonkan и микроконтроллера ESP8266 с модулем подключения к беспроводной сети WiFi. Получившаяся система считывает идентификатор метки, сервер сопоставляет его с имеющимся в базе, и определяет, используется ли ещё СИМ кем-либо или нет. В случае, если не используется, при повторном обнаружении метки где-либо с неё удаляется информация о временном владельце, либо записывается о новом. Система успешно прошла альфа тестирование.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Зиборов, О. В. Использование средств автоматической фото- и видеофиксации нарушений ПДД в целях осуществления надзора за дорожным движением / О. В. Зиборов, П. Г. Андреев // Управление деятельностью по обеспечению безопасности дорожного движения: состояние, проблемы, пути совершенствования. – 2019. – № 1(2). – С. 201-204.
2. Финкенцеллер, К. RFID-технологии. Справочное пособие / К. Финкенцеллер; пер. с нем. Союнханова Н.М. — М.: Изд-во Додэка, 2010. — 496 с.
3. Гудин, М. Технология RFID: реалии и перспективы / М. Гудин, В. Зайцев // Компоненты и технологии. – 2003. – № 4(30). – С. 42-44.
4. Обзор технологий и стандартов RFID систем / Н. А. Верзун, Д. М. Воробьева, А. М. Колбанев, М. О. Колбанев // Информационные технологии и телекоммуникации. – 2018. – Т. 6. – № 1. – С. 1-11.
5. Çaliş, G. Analysis of the variability of RSSI values for active RFID-based indoor applications / G. Çaliş // Turkish Journal of Engineering & Environmental Sciences. – 2013. – Т. 37. – №. 2. – P. 186-207
6. Веб-сайт «SparkFun». [Электронный ресурс]. URL: <https://www.sparkfun.com/products/14066>. Дата обращения: 10.03.2021.
7. Веб-сайт «Fonkan». [Электронный ресурс]. URL: http://www.fonkan.com/en_product_lists_5340.html. Дата обращения 15.04.2021
8. Соловьев, Ю.А. Системы спутниковой навигации / Ю.А. Соловьев — М.: Изд-во Эко-Трендз, 2000. — 267 с.
9. Андреев К.П. Разработка мероприятий по оптимизации городской маршрутной сети / К.П. Андреев, В.В. Терентьев, И.Е. Агуреев // Грузовик. – 2017. – № 8 – С. 6-9.
10. Черницын, С.А. Спутниковая система мониторинга общественного транспорта в Екатеринбурге / С.А. Черницын // Научное творчество молодежи – лесному комплексу России: материалы X Всероссийской науч.-техн. конф. студентов и аспирантов и конкурса по программе «Умник» – 2014. – Ч. 1. – С. 412–413.
11. Веб-сайт «Habr». [Электронный ресурс]. URL: <https://habr.com/ru/company/pochtoy/blog/415549>. Дата обращения 02.04.2021

12. Varshney, U. (2003). Location management for mobile commerce applications in wireless Internet environment. *ACM Transactions on Internet Technology*, 3(3), 236–255.
13. Комраков, Д.В. Технологии позиционирования наземных подвижных объектов в сетях GSM / Д.В. Комраков // Технические науки в России и за рубежом (II): материалы международной заочной научной конференции – 2012. – С. 38–40.
14. Дудяк, Е.И. Методы определения координат сотрудников и техники предприятия с использованием технологии Wi-Fi / Е.И. Дудяк // Техника радиосвязи. – 2015. – №. 1. – С. 67-77.
15. Сулейманов, Д.Ф. Метод определения координат мобильных устройств по радиосигналу wi-fi точек // Научно-технический вестник информационных технологий, механики и оптики. – 2007. – № 41. – С. 40-43.
16. Васин, Н.Н. Алгоритмы идентификации объектов в системах видеонаблюдения с использованием статистических методов / Н.Н. Васин, Р.Р. Диязитдинов // Инфокоммуникационные технологии. – 2011. – Т. 9. – № 3. – С. 23-28.
17. Камалутдинов, Р.Х. Определение местоположения объектов в помещении на основе данных видеонаблюдения / Р.Х. Камалутдинов, В.А. Суздальцев // За нами будущее: взгляд молодых ученых на инновационное развитие общества : сборник научных статей Всероссийской молодежной научной конференции: в 4 томах – Курск: Юго-Западный государственный университет. – 2020. – С. 338-340.
18. Веб-сайт «Whoosh» [Электронный ресурс]. URL: <https://whoosh.bike/faq>
Дата обращения 01.05.2021.
19. Веб-сайт «Apache» [Электронный ресурс]. URL: <https://httpd.apache.org>
Дата обращения 16.05.2021.
20. Веб-сайт «MySQL» [Электронный ресурс]. URL: <https://www.mysql.com>
Дата обращения 16.05.2021.
21. Веб-сайт «PHP» [Электронный ресурс]. URL: <https://www.php.net> Дата обращения 18.05.2021.
22. Веб-сайт «Arduino» [Электронный ресурс]. URL: <https://www.arduino.cc>
Дата обращения 21.05.2021.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД МИКРОКОНТРОЛЛЕРА ESP

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#define SERVER_ADDR "192.168.0.103"

const char* ssid      = "dlink";
const char* password = "00123638300";
String receivedEPC;
String receivedUSER;
String receivedREADERID;

WiFiClient client;

String receivedStr;
int messageLength;
char temp[16];

void setup() {
  Serial.begin(38400);
  while (!Serial) {
    //do nothing
  }
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}

void loop() {
  delay(6000);
  Serial.write("\nR3,0,1\n");
  delay(10);
  if ( Serial.available()) {
    receivedUSER = Serial.readString();
  }
  delay(100);
  Serial.write("\nR1,2,6\n");
  delay(10);
  if ( Serial.available()) {
```

```

receivedEPC = Serial.readString();
}
receivedUSER.trim();
receivedEPC.trim();
receivedUSER.replace('R', ' ');
receivedEPC.replace('R', ' ');

if (client.connect(SERVER_ADDR, 8180)) {
  client.print("GET /Checkrent.php?");
  client.print("TagID=");
  client.print(receivedEPC);
  client.print("&UserID=");
  client.print(receivedUSER);
  client.println(" HTTP/1.1");
  client.print("Host: ");
  client.println(SERVER_ADDR);
  client.println("Connection: close");
  client.println();
  client.println();
  delay(100);
  if (client.connected())
  {
    String answer = client.readString();
    int startSearch = answer.indexOf("UTF-8") + 6;
    String resp = answer.substring(startSearch, answer.length());
    if (resp != "N"){
      if (resp == "0"){
        Serial.write("\nW3,0,1,0000\r");
      }
      else {
        char buf[14];
        int n = sprintf(buf, "\nW3,0,1,%04d\r", resp.toInt());
        Serial.write(buf);
      }
    }
  }
  client.stop();
}
}

```

ПРИЛОЖЕНИЕ Б

PHP СКРИПТ ОТОБРАЖЕНИЯ ТЕКУЩЕГО СОСТОЯНИЯ СИМ

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "individualmobility";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: \n" . $conn->connect_error);
}
$sql = "SELECT scooters.model, companies.compName, status_list.definition,
users.userName, time FROM companies INNER JOIN scooters ON
scooters.companyID=companies.id INNER JOIN current_status ON
scooters.id=current_status.scooterID LEFT JOIN users on
users.id=scooters.currUserId, status_list WHERE
current_status.statusID=status_list.id";
$result = $conn->query($sql);
if (!$result) { trigger_error('Invalid query: ' . $conn->error);}
if ($result->num_rows > 0) {
    echo "<h1 align = \"center\"><b>Текущее состояние СИМ</b></h1>";
    echo "<table border=\"2\" cellpadding=\"5\" align =
\"center\"><tr><th>Модель</th><th>Компания</th><th>Состояние</th><th>Пользоват
ель</th><th>Время</th></tr>";
    while($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td align = \"center\">" . $row["model"] . "</td>";
        echo "<td align = \"center\">" . $row["compName"] . "</td>";
        echo "<td align = \"center\">" . $row["definition"] . "</td>";
        echo "<td align = \"center\">" . $row["userName"] . "</td>";
        echo "<td align = \"center\">" . $row["time"] . "</td>";
        //for ($j = 0 ; $j < 3 ; ++$j) echo "<td>$row[$j]</td>";
        echo "</tr>";
    }
} else { echo "0 results";}
$conn->close();
?>
```


ПРИЛОЖЕНИЕ В

PHP СКРИПТ ЗАПРОСА ИНФОРМАЦИИ О СОСТОЯНИИ АРЕНДЫ ОТ МИКРОКОНТРОЛЛЕРА

```
<?php
date_default_timezone_set("Asia/Yekaterinburg");

//Checkrent.php?TagID=xxxxxx&UserID=xxxx

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "individualmobility";

$tagid = $_GET['TagID'];
$userid = $_GET['UserID'];

//создание соединения
$conn = new mysqli($servername, $username, $password, $dbname);

//проверка соединения
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

//поиск пользователя по TID
$sql1 = "SELECT currUserID FROM scooters INNER JOIN rfid_tags ON
rfid_tags.id=scooters.tagID WHERE scooters.tagID=(SELECT rfid_tags.id WHERE
rfid_tags.TID='$tagid')";
$result = $conn->query($sql1);

if ($conn->query($sql1) == FALSE) {
    echo "\nError: " . $sql1 . "<br>" . $conn->error;
}
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $usrID = $row["currUserID"];
    }
}
```

```
else {
    die("UNKNOWN USER");//echo "0 results";
}

if ($userid==$usrID){
    echo "N";
}
elseif ($usrID == NULL and $userid == 0){
    echo "N";
}
elseif ($usrID == NULL) {
    echo "0";
}
else {
    echo "$usrID";
}

$conn->close();

?>
```

ПРИЛОЖЕНИЕ Г

PHP СКРИПТ ВЗЯТИЯ В АРЕНДУ СИМ

```
<?php
date_default_timezone_set("Asia/Yekaterinburg");

// Rent.php?userid=x&scooterid=x

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "individualmobility";

$date = date('Y-m-d H:i:s');
$userid = $_GET['userid'];
$scooterid = $_GET['scooterid'];

//создание соединения
$conn = new mysqli($servername, $username, $password, $dbname);

//проверка соединения
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
    echo "Connection SQL error";
}

//поиск пользователя
$sql1 = "SELECT id FROM `users` WHERE id='$userid'";
$result = $conn->query($sql1);

if ($conn->query($sql1) == TRUE) {
    echo "ID received\r\n";
} else {
    echo "Error: " . $sql1 . "<br>" . $conn->error;
}

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $usrID = $row["id"];
    }
} else {
    die("UNKNOWN USER");//echo "0 results";
}
}
```

```

//поиск СИМ
$sql3 = "SELECT id FROM `scooters` WHERE id='$scooterid'";
$result = $conn->query($sql3);
if ($conn->query($sql3) == TRUE) {
    echo "Scooter received\r\n";
} else {
    echo "Error: " . $sql3 . "<br>" . $conn->error;
}

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $sctrID = $row["id"];

        $sql4 = "UPDATE `scooters` SET `currUserID` = '$usrID' WHERE
`scooters`.`id` = '$sctrID'";
        if ($conn->query($sql4) == TRUE) {
            echo "Data Updated\r\n";
        } else {
            echo "Error: " . $sql4 . "<br>" . $conn->error;
            echo "UNKNOWN USER ID OR SCOOTER ID";
        }
    }
    $sql6 = "UPDATE `current_status` SET `statusID` = '1', `time` =
'$date' WHERE `scooterID` = '$sctrID'";
    $result = $conn->query($sql6);
    if ($conn->query($sql6) == TRUE) {
        echo "CurrentDataUpdated\r\n";
    } else {
        echo "Error: " . $sql6 . "<br>" . $conn->error;
    }
}
$conn->close();

?>

```

ПРИЛОЖЕНИЕ Д

PHP СКРИПТ ЗАВЕРШЕНИЯ АРЕНДЫ СИМ

```
<?php
date_default_timezone_set("Asia/Yekaterinburg");

// Unnent.php?userid=x&scooterid=x

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "individualmobility";

$date = date('Y-m-d H:i:s');
$userid = $_GET['userid'];
$scooterid = $_GET['scooterid'];

//создание соединения
$conn = new mysqli($servername, $username, $password, $dbname);

//проверка соединения
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
    echo "Connection SQL error";
}

//поиск пользователя
$sql1 = "SELECT id FROM `users` WHERE id='$userid'";
$result = $conn->query($sql1);

if ($conn->query($sql1) == TRUE) {
    echo "ID received\r\n";
} else {
    echo "Error: " . $sql1 . "<br>" . $conn->error;
}

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $usrID = $row["id"];
    }
} else {
    die("UNKNOWN USER");//echo "0 results";
}
}
```

```

//поиск СИМ
$sql3 = "SELECT id FROM `scooters` WHERE id='$scooterid'";
$result = $conn->query($sql3);
if ($conn->query($sql3) == TRUE) {
    echo "Scooter received\r\n";
} else {
    echo "Error: " . $sql3 . "<br>" . $conn->error;
}

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $sctrID = $row["id"];

        $sql4 = "UPDATE `scooters` SET `currUserID` = NULL WHERE
`scooters`.`id` = '$sctrID'";
        if ($conn->query($sql4) == TRUE) {
            echo "Data Updated\r\n";
        } else {
            echo "Error: " . $sql4 . "<br>" . $conn->error;
            echo "UNKNOWN USER ID OR SCOOTER ID";
        }
    }
    $sql6 = "UPDATE `current_status` SET `statusID` = '2', `time` =
'$date' WHERE `scooterID` = '$sctrID'";
    $result = $conn->query($sql6);
    if ($conn->query($sql6) == TRUE) {
        echo "CurrentDataUpdated\r\n";
    } else {
        echo "Error: " . $sql6 . "<br>" . $conn->error;
    }
}
$conn->close();
?>

```

ПРИЛОЖЕНИЕ Е

ДОПОЛНИТЕЛЬНАЯ ПРОГРАММА ТЕСТИРОВАНИЯ МИКРОКОНТРОЛЛЕРА ESP

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#define SERVER_ADDR "192.168.0.103"

const char* ssid      = "dlink";
const char* password = "00123638300";
String receivedEPC;
String receivedUSER;
String receivedREADERID;

WiFiClient client;

String receivedStr;
int messageLength;
char temp[16];

void setup() {
  Serial.begin(38400);
  while (!Serial) {
    //do nothing
  }
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  delay(6000);
  if (client.connect(SERVER_ADDR, 8180)) {
    client.print("GET /Checkrent.php?");
```

```

client.print("TagID=E280110C2000784E");
client.print("&UserID=3");
client.println(" HTTP/1.1");
client.print("Host: ");
client.println(SERVER_ADDR);
client.println("Connection: close");
client.println();
client.println();
Serial.println("Sent query");
if (client.connected())
{
    Serial.println("Sent query with TagID=E280110C2000784E and UserID=3");
    Serial.println("When scooter is free");
    Serial.println("Waiting answer...");
    String answer = client.readString();
    int startSearch = answer.indexOf("UTF-8") + 6;
    String resp = answer.substring(startSearch, answer.length());
    if (resp != "N"){
        if (resp == "0"){
            Serial.write("\nW3,0,1,0000\r");
        }
        else {
            char buf[14];
            int n = sprintf(buf, "\nW3,0,1,%04d\r", resp.toInt());
            Serial.write(buf);
        }
    }
}
else {
    Serial.print("may not connected");
}
client.stop();
}
else {
    Serial.println("--> connection failed\n");
}
}

```