



ИНТЕЛТРАНС
Г Р У П П А К О М П А Н И Й

IT & TRANSPORT

ИТ & ТРАНСПОРТ

Сетевое издание

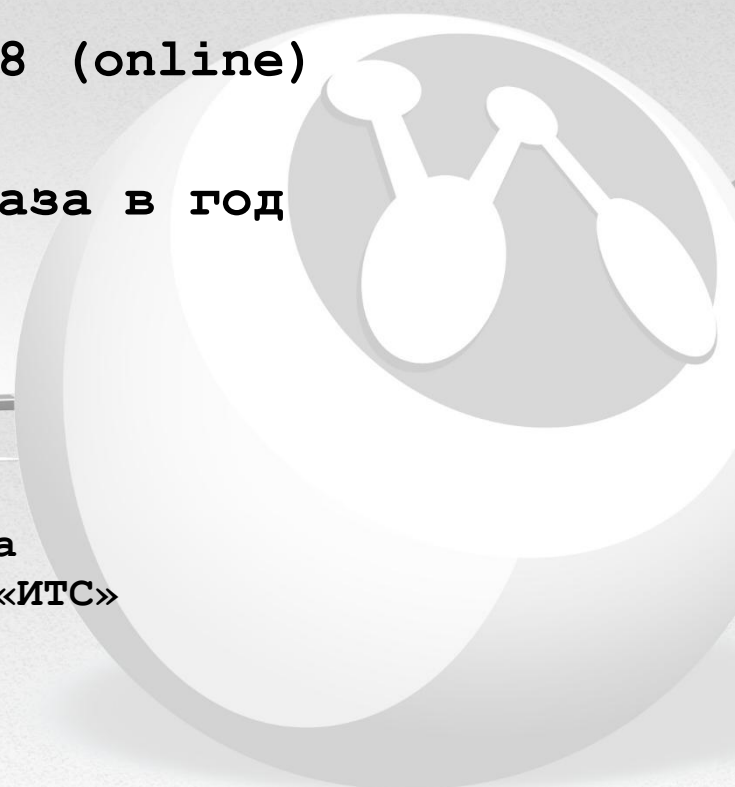
№ 3 (27) – 2024

ISSN 2949–5288 (online)

Выходит: 4 раза в год

Самара
ООО «НПЦ «ИТС»

2024





ИНТЕЛТРАНС
Г Р У П П А К О М П А Н И Й

И Т & Т Р А Н С П О Р Т

И Т & Т Р А Н С П О Р Т

Направления :

**1.2.2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ЧИСЛЕННЫЕ
МЕТОДЫ И КОМПЛЕКСЫ ПРОГРАММ (технические науки)**

**2.3.1. СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА
ИНФОРМАЦИИ (технические науки)**

**2.3.3. АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ
ПРОЦЕССАМИ И ПРОИЗВОДСТВАМИ (технические науки)**

**2.3.5. МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ВЫЧИСЛИТЕЛЬНЫХ МАШИН, КОМПЛЕКСОВ И КОМПЬЮТЕРНЫХ
СЕТЕЙ (технические науки)**

**Главный редактор Михеева Т. И.
доктор технических наук, профессор,
заслуженный деятель науки и образования РФ**

**Самара
ООО «НПЦ «ИТС»
2024**

IT & Transport / ИТ & Транспорт : под ред. профессора Т. И. Михеевой. – Самара, ООО «НПЦ «ИТС», 2024, №3 (27). – 65 с. ил.

ISSN: 2949-5288

В сборнике представлены научные статьи, содержащие результаты исследований в следующих предметных областях: системный анализ, управление и обработка информации, интеллектуальные транспортные системы, геоинформационные системы, информационные технологии, базы данных и знаний, системы искусственного интеллекта, цифровая обработка изображений, управление транспортными процессами, развитие транспортной инфраструктуры, автоматизация и управление технологическими процессами и производствами, математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей, математическое и компьютерное моделирование.

Предназначен для научно-технических работников, инженеров, аспирантов и студентов, обучающихся по специальностям, связанным с информационными технологиями, транспортными системами и процессами.

Авторская позиция и стилистические особенности публикаций полностью сохранены.

Главный редактор – Т. И. Михеева доктор технических наук, профессор,
заслуженный деятель науки и образования РФ

Редакционная коллегия:

Бурдин А.В.	доктор технических наук, доцент
Гераськин М.И.	доктор экономических наук, профессор
Жанказиев С.В.	доктор технических наук, профессор
Заболотнов Ю.М.	доктор технических наук, профессор
Зырянов В.В.	доктор технических наук, профессор
Иващенко А.В.	доктор технических наук, профессор
Минниханов Р.Н.	доктор технических наук, профессор
Прохоров С.А.	доктор технических наук, профессор
Филиппова А.С.	доктор технических наук, профессор
Хайтбаев В.А.	доктор экономических наук, профессор
Зеленко Л.С.	кандидат технических наук, доцент
Золотовицкий А.В.	кандидат технических наук, доцент
Михеев С.В.	кандидат технических наук, доцент
Сапрыкин О.Н.	кандидат технических наук, доцент
Сапрыкина О.В.	кандидат технических наук, доцент
Федосеев А.А.	кандидат технических наук, доцент
Шопин А.Г.	кандидат технических наук

ISBN 978-5-6049535-6-3©

Сетевое издание «ИТ & Транспорт» зарегистрировано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций; регистрационный номер: серия ЭЛ № ФС 77–85276 от 30.05.2023

e-mail: itsspc@yandex.ru

© Адрес страницы сайта: <http://ittransport.ru>

© ООО «НПЦ «ИТС», 2024

УДК 2.3.1

Молев А.М., Михеева Т.И., Клепиков Н.М.
**АДРЕСНОЕ ПОЗИЦИОНИРОВАНИЕ И ЕГО РЕАЛИЗАЦИЯ
В ГЕОИНФОРМАЦИОННОЙ СИСТЕМЕ**

*Самара, Самарский университет имени академика С.П. Королёва
ИнтелТранс*

В данной статье раскрывается метод адресного позиционирования, представляющий собой альтернативный способ задания местоположения пространственных объектов в геоинформационной системе ITSGIS. Дается его определение, рассматриваются сферы его применения, указываются правила построения записи с указанием местоположения и приводится пример такого построения. Также приводятся алгоритмы, которые должны реализовываться геоинформационными системами для поддержки данного вида позиционирования.

Ключевые слова: позиционирование, местоположение, координата, адресный план, реализация, зона близости, геоинформационная система ITSGIS.

Введение

Неотъемлемой характеристикой любой интеллектуальные транспортные геоинформационной системы (ИТСГИС) ITSGIS является способность определять местоположение каждого описываемого ей пространственного объекта. Традиционным способом реализации этой функции является установка глобальных координат каждой вершине объекта в некоторой системе координат, называемая позиционированием [1]. Такой способ локации объектов наиболее удобен при обработке данных с использованием ЭВМ, однако является нечитаемым даже для операторов ITSGIS, не говоря уже об обывателях [2]. Едва ли найдется человек, который сможет без помощи компьютерных сервисов сказать, в каком месте находится точка «53.211967°N, 50.177502°E».

По этой причине предприятиями и государственными учреждениями в качестве записей, идентифицирующих местоположение, обычно используются почтовые адреса как маркеры реально существующих физических объектов – автодорог, строений либо земельных участков [3]. Соответственно, в ГОСТ Р 52438-2005 уточняется, что позиционирование в ITSGIS также может осуществляться опосредованным образом «путем геокодирования». Под гео-

кодированием обычно подразумевают получение геометрических данных на основе описания местоположения в виде почтового адреса [4, 5]. Тем не менее, использование только почтовых адресов для решения задачи позиционирования недостаточно из-за низкой точности в населенных пунктах и в принципе невозможно в незаселенной местности. Эти проблемы требуют разработки другого, более точного способа человекочитаемой адресации географических объектов, остающегося применимым и для вычислительных систем.

Адресное позиционирование

Основной целью метода определения местоположения, который мы будем называть «адресным позиционированием», является облегчение выполнения задач, выполняемых при учете объектов городской инфраструктуры (фонарные столбы, канализационные люки и т.п.). К этим задачам относятся:

- хранение больших массивов пространственной информации, доступной для понимания сотрудникам муниципальных организаций;
- группировка по близлежащим адресам и сортировка больших массивов данных с координатной информацией.

Адресное позиционирование должно отвечать трем требованиям:

- позволять однозначным образом задавать точечное местоположение без потери точности относительно координатного позиционирования;
- оставаться в максимально возможной степени человекочитаемым;
- сохранять способность обозначать любую точку на поверхности Земли.

В общем случае адресное позиционирование представляет собой смесь геокодирования и координатного позиционирования. Запись о местоположении складывается из трех компонентов:

- адреса ближайшего к описываемой точке объекта;
- записи, определяющей положение связанной с выбранным адресным объектом точки отсчета (начала координат);
- набора числовых данных, определяющих сдвиг описываемой точки относительно точки отсчета, который может задаваться в виде декартовых координат – сдвиг по меридианам и сдвиг по параллелям – либо полярных (расстояние и азимутальное направление).

Таким образом, мы получаем систему координат с варьируемой точкой отсчета. Запись о ней должна обозначать видимый и доступный физический ориентир, принадлежащий зданию, например, крайний по одному из направлений угол – самый северный, самый южный и т.д. Такое описание позволяет

однозначно идентифицировать опорную точку как в физическом пространстве, так и на электронной карте.

Рассмотрим пример описания точки в координате « 53.212258° N, 50.178741° E» посредством адресного позиционирования (рисунок 1).

1. Ближайшим к данной точке является здание с адресом «Самарская обл., г. Самара, Московское шоссе, 34». Этот адрес составляет первую часть позиционной записи.
2. В качестве опорной точки будет взят восточный угол данного здания как самый ближайший к этой точке. В общем случае может браться любой – главное, чтобы эта точка была легко различимой.
3. На рисунке 1 $l = 17,8$ м; $\alpha = 128^{\circ}$ (здесь l – расстояние до точки, α – азимутальное направление). Для повышения удобочитаемости вместо точной величины направления может быть взято приблизительное – в данном случае юго-восточное.

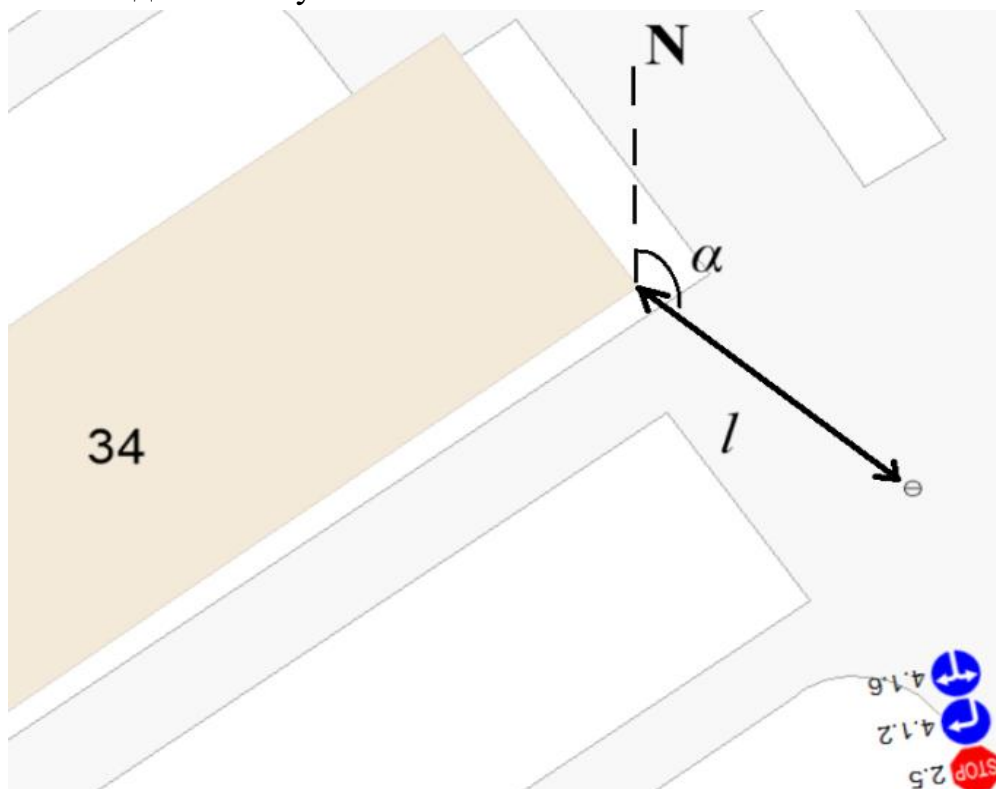


Рисунок 1. Положение точки относительно здания

Таким образом, запись « 53.212258° N, 50.178741° E» будет иметь вид: «В 17,8 метрах (к ЮВ) от восточного угла здания Самарская обл., г. Самара, Московское шоссе, 34 по направлению 128° ».

В некоторых задачах за опорный объект вместо здания удобнее брать начало улицы либо близлежащий перекресток. Тогда для точного определения локации на дороге можно использовать расстояние от начала улицы либо перекрестка и расстояние от центра проезжей части.

Реализация

Для реализации адресного позиционирования ITSGIS должна включать в себя структурированный адресный план, причем обращение к его содержанию должно занимать минимальное время. Адресный план должен содержать информацию обо всех объектах, используемых в качестве ориентиров [6]. Кроме того, система должна реализовывать поиск ближайших ориентиров. Такой поиск может реализовываться двумя способами.

Во-первых, для каждой рассматриваемой точки P может производиться непосредственный поиск ближайшего адресного объекта. Для этого выполняется следующий итеративный алгоритм:

1. Получение множества S всех объектов внутри окружности с центром в P и радиусом R . Начальная величина R зависит от местности расположения P – в городской местности она ниже, чем в сельской, обычно 10 и 50 метров соответственно.
2. При $S \neq \emptyset$ берется объект из S с минимальным расстоянием до P . При $S = \emptyset$ радиус R удваивается. Переход к шагу 1.

Такой алгоритм может выполняться долгое время из-за итеративности, и поэтому может не подойти в случаях, когда необходимо получить местоположения многих тысяч объектов. Поэтому ITSGIS также должна реализовывать второй, опосредованный алгоритм – с сохранением зоны близости для каждого объекта адресного плана при его создании или изменении.

Рассмотрим алгоритм формирования зоны близости. Предварительно указывается радиус зоны близости r и создается массив линий-перпендикуляров M .

1. Находится ближайший к центроиде S нового здания адресный объект O_i и у него вычисляется центроид P_i . Поиск производится среди объектов, расположенных не дальше хотя бы одной линии из M , чем сам адресный объект O_i . При их отсутствии алгоритм заканчивает свою работу.
2. Строится перпендикуляр n_i к отрезку CP_i . От него отсекаются части, не входящие в окружность R с центром в S и радиусом r .
3. При пересечении перпендикуляра n_i с некоторым ранее построенным n_j в точке N от n_i и n_j отсекаются части, расположенные ближе к O_i и O_j , чем к S , соответственно.
4. В массив M заносится n_i и обновляется n_j .
5. Переход к шагу 1.

В результате выполнения из линий M становится возможно построить многогранник вокруг S , который и является зоной близости адресного объекта. На рисунке 2 полученная зона близости обведена черным контуром. Если

многогранник сформировать не удалось, это означает, что в M остались перпендикуляры, не обрезанные с обеих сторон при выполнении пункта 3, что говорит об отсутствии адресных объектов с одной из сторон C в радиусе r . Тогда при формировании зоны близости также используется дуга окружности R , соединяющая границы зоны близости с этих сторон.

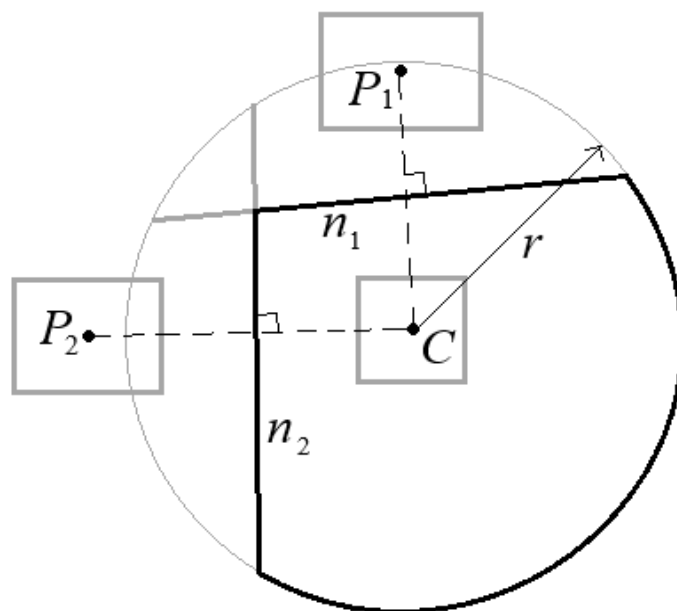


Рисунок 2. Пример построения зоны близости

После создания зоны близости O_i перестраиваются все зоны близости объектов, перпендикуляры к которым попали в массив M .

Применение данного алгоритма позволяет при описании точки вместо перебора всех близких к данной точке объектов ограничиться нахождением зоны близости, в которую входит точка, и на ее основе формировать запись.

Приведенные алгоритмы были реализованы в интеллектуальной транспортной геоинформационной системе «ИТСГИС», что позволило оптимизировать выполняемые с ее помощью работы по проектированию дорожного движения и паспортизации автомобильных дорог.

Список использованных источников

1. ГОСТ Р 52438-2005 Географические информационные системы. Термины и определения. – М. : Стандартинформ, 2006. – 16 с.
2. Семин, П.О. Картографирование российских субъектов малого и среднего предпринимательства по открытым данным налоговой службы / П.О. Семин // Вестник Псковского государственного университета. Серия: Естественные и физико-математические науки. – 2024. – Т. 17. – № 2. – С. 45–64.

3. Потапов, А.Е. Реализация программного интерфейса для работы с Федеральной информационной адресной системой / А.Е. Потапов, О.В. Калинин // Электронный журнал: наука, техника и образование. – 2018. – № СВ1(18). – С. 93–98.
4. Обухов, Л.А. О контроле корректности при геокодировании почтовых адресов / Л.А. Обухов, Е.А. Паниди // ИнтерКарто. ИнтерГИС. – 2021. – Т. 27. – № 2. – С. 114–127.
5. Капралов, Е.Г. Основы геоинформатики / Е.Г. Капралов, А.В. Кошкарев, В.С. Тикунов // В 2 кн. Кн. 1: Учеб. пособие для студ. вузов. – М.: Издательский центр «Академия», 2004. – 352 с.
6. Смолев, А.М. Методы моделирования адресного плана в рамках геоинформационной системы / А.М. Смолев, Т.И. Михеева, А.В. Золотовицкий // ИТ & Транспорт: Сборник научных статей. – 2023. – Т. 24. – С. 4–13.
7. Михеева, Т.И. Интеллектуальная транспортная геоинформационная система ITSGIS. Ядро / Т.И. Михеева, С.В. Михеев, О.К. Головнин, А.В. Сидоров, Е.А. Савинов. – Самара : Интелтранс, 2016. – Т.1. – 171 с.
8. Golovnin, O.K. Interactive content management in the system “Informatriks” / T.I. Mikheeva, S.V. Mikheev, O.K. Golovnin, I.G. Bogdanova // Proceedings of the 14th international workshop on computer science and information technologies CSIT’2012, Ufa – Hamburg – Norwegian Fjords, September 20–26, 2012. – P. 285–289.
9. Михеева, Т.И. Интеллектуальная транспортная геоинформационная система ITSGIS / Т.И. Михеева, С.В. Михеев, О.К. Головнин // Современные проблемы безопасности жизнедеятельности: интеллектуальные транспортные системы : материалы IV Международной научно-практической конференции (Казань, 25–26 февраля 2016 г.). – Казань : ГБУ «Научный центр безопасности жизнедеятельности», 2016. – С. 362–368. – ISBN 978-5-85247-837-5.

A.M. Smolev, T.I. Mikheeva, N.M. Klepikov

ADDRESS-BASED LOCATION DEFINITION AND ITS IMPLEMENTATION BY GEOGRAPHIC INFORMATION SYSTEMS

*Samara, Samara University named after academician S.P. Korolev
IntelTrans*

This article describes the address-based location definition method, which is an alternative way of specifying the location of spatial objects in a geographic in-

formation system. It defines this method, examines its areas of application, specifies the rules for constructing a record indicating the location, and provides an example of such definition. It also provides algorithms that must be implemented by geographic information systems to support this form of location definition.

Keywords: pattern, design, logical data model, modular architecture, geographic information system, visualization, address plan, tool.

УДК 004.02

Михеева Т.И., Клепиков Н.М.
**ИНСТРУМЕНТЫ ОБРАБОТКИ, АНАЛИЗА И РАСПОЗНАВАНИЯ
ИЗОБРАЖЕНИЙ**

*Самарский университет имени академика С.П. Королёва
ИнтелТранс*

В статье обобщаются основные цели обработки изображений для дальнейших разработок, а также описаны основные преимущества использования инструментов обработки и распознавания изображений

Ключевые слова: обработка изображений, анализ, распознавание изображений

Введение

В данной статье будут обобщаться инструменты обработки, анализа и распознавания изображений как программное обеспечение. Данные инструменты будут рассмотрены с перспективой дальнейшего использования в интеллектуальной транспортной геоинформационной системе ITSGIS. В ITSGIS реализована система распознавания дорожных знаков, которую необходимо усовершенствовать с помощью новых инструментов обработки изображений. На разработанную интеллектуальную транспортную геоинформационную систему ITSGIS распознавания знаков по видео получено свидетельство о государственной регистрации программы для ЭВМ № 2021616700. Инструменты обработки, анализа и распознавания изображений необходимы для улучшения точности распознавания образов.

Инструменты обработки, анализа и распознавания изображений как программное обеспечение

Инструменты обработки, анализа и распознавания изображений относят

к разделу «Средства анализа данных» в Классификаторе программ для электронных вычислительных машин и баз данных. Официальное определение Минкомсвязи – «Программное обеспечение, которое должно быть предназначено для обработки и распознавания видеопотоков и изображений, полученных различными способами, включая системы видеонаблюдения, камеры видеонаблюдения, фотографирование, 2D и 3D-сканирование и другие способы...».

Компьютерное зрение использует алгоритмы обработки изображений для анализа и понимания визуальных эффектов из одного изображения или нескольких. Примером компьютерного зрения является идентификация пешеходов и транспортных средств на дороге путем точной классификации и фильтрации миллионов картинок. За прошедшие годы рынок компьютерного зрения значительно вырос. В настоящее время он оценивается в 11,94 млрд. долларов США и, вероятно, достигнет 17,38 млрд. долларов США к 2023 году при среднегодовом темпе роста в 7,80% в период с 2018 по 2023 год.

Это связано с ростом спроса на автономные и полуавтономные транспортные средства, дроны (военного и бытового назначения), носимые устройства и смартфоны. Более того, растущее внедрение Индустрии 4.0 и автоматизации в производственных отраслях еще больше стимулировало спрос на компьютерное зрение.

Распознавание изображений – это способность системы или программного обеспечения идентифицировать объекты, людей, места и действия. Он использует технологии машинного зрения с искусственным интеллектом и обученные алгоритмы для распознавания через систему камер.

Ряд основных целей обработки изображений:

- Визуализация – описывает обработанные данные, придавая видимую форму невидимым объектам.
- Повышение резкости и восстановление изображения – повышает качество обработанных изображений.
- Восстановление изображения – помогает при поиске изображения.
- Измерение объекта – измеряет объекты на изображении.
- Распознавание образов – определяет объекты на изображении, определяет их положение и понимает контекст.

Преимущества использования инструментов обработки, анализа и распознавания изображений

Основные преимущества ПО:

- Распознавание объектов.
- Улучшение качества изображений стационарных и движущихся объектов.

- Мониторинг ситуаций в реальном времени, быстрое реагирование на отклонения.

ТОП – 5 ПО «Инструменты обработки, анализа и распознавания изображений»

Основное ПО дано в таблице.

Таблица 1. ТОП – 5 ПО «Инструменты обработки, анализа и распознавания изображений»

№	Наименование	Функционал
1	ЦИАС	Программа собирает и обрабатывает данные из различных источников, структурирует по заданным метрикам для графической визуализации и анализа.
2	Мега – Дозор	ПО для отслеживания и визуализации дорожных ситуаций. ПО создано на базе нейронных сетей. Регистрация и мониторинг ситуации, передача данных.
3	Beorg Smart Vision	Оцифровка документов и распознавание изображений с применением технологий искусственного интеллекта и качеством выше 99%. Загрузка в информационную систему предприятия.
4	ИнтелВиз	ПО собирает, интерпретирует, хранит и передает данные с видеокамер о движении транспортных средств. Интегрируется с другими системами.
5	N3.Аналитика	Целевое назначение: автоматизация сбора, хранения, очистки, разметки, анализа данных из различных источников. Строит аналитические отчеты с поддержкой визуализации.

Заключение

В результате выполнения научно-исследовательской работы изучены темы, связанные с обнаружением, анализом и распознаванием изображений. Рассмотрены основные преимущества в использовании данных подходов.

Данная технология внедрена в плагин «Дорожные знаки» в среде интеллектуальной транспортной геоинформационной системы ITSGIS.

Реализованы шаги, которые позволяют достичь высоких показателей в обнаружении дорожных знаков с наибольшей скоростью, не требуя больших вычислительных мощностей.

Список литературы

1. Каллан, Р. Основные концепции нейронных сетей / Р. Каллан. – М.: Издательский дом «Вильямс», 2001. – 287 с.
2. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М.: Издательский дом «Вильямс», 2006. – 1104 с.
3. Михеев, С.В. Диагностика состояния транспортной инфраструктуры с использованием нейронных сетей / С.В. Михеев, А.В. Сидоров, А.А. Осьмушин // Современные проблемы науки и образования. – 2013. – №6. – С. 215-216.
4. Клепиков, Н.М. Исследование алгоритмов распознавания дорожных знаков / Н.М. Клепиков, Т.И. Михеева // IT & Transport. – 2021. – №14. – С. 46-57.
5. Михеева, Т.И. Интеллектуальная транспортная геоинформационная система ITSGIS / Т.И. Михеева, С.В. Михеев, О.К. Головнин // Современные проблемы безопасности жизнедеятельности: интеллектуальные транспортные системы. – 2016. – №1. – С. 362-368.
6. Клепиков, Н.М. Методы распознавания дорожных знаков в среде ITSGIS / Н.М. Клепиков, Т.И. Михеева // Перспективные информационные технологии (ПИТ 2021). – Самара, 2021. – С. 470-476.
7. Михеева, Т.И. Интеллектуальная транспортная геоинформационная система ITSGIS. Ядро / Т.И. Михеева, С.В. Михеев, О.К. Головнин, А.В. Сидоров, Е.А. Савинов. – Самара : Интелтранс, 2016. – Т.1. – 171 с.
8. Михеева, Т.И. Структурно-параметрический синтез интеллектуальных транспортных систем / Т.И. Михеева. – Самара : Самар. науч. центр РАН, 2008. – 380 с.
9. Хузятов, Ш.Ш. Проектирование автоматизированных систем управления на основе методики паттерна / Ш.Ш. Хузятов, Р.А. Валиев // Научно-технический вестник Поволжья. – 2015. – № 2. – С. 215–218.
10. Смолев, А.М. Методы моделирования адресного плана в рамках геоинформационной системы / А.М. Смолев, Т.И. Михеева, А.В. Золотовицкий // IT & Transport / ИТ & Транспорт : Сборник научных статей. – Самара : ИнтелТранС, 2023. – Т. 24. – С. 4-13.
11. Джалалов, М.Э. Применение шаблонов проектирования для управления API в микросервисной архитектуре / М.Э. Джалалов // Экономика и качество систем связи. – 2024. – № 1(31). – С. 128-136.

Mikheeva T.I., Klepikov N.M.

IMAGE PROCESSING, ANALYSIS AND RECOGNITION TOOLS

Samara University named after academician S.P. Korolev

IntelTrans

The article reviews the main goals of image processing for further development, as well as describes the main advantages of using image processing and recognition tools.

Keywords: image processing, analysis, image recognition

УДК 004.02

Молодыко К.А., Михеева Т.И.

РОБОВАЯ РОБОТИЗИРОВАННАЯ СИСТЕМА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ИТСГИС НА ОСНОВЕ БЕСПРОВОДНОЙ СЕТИ ПЕРЕДАЧИ ДАННЫХ

Самара, Самарский университет имени академика С.П. Королёва

ИнтелТранс

В данной статье рассматривается функциональная спецификация ядра клиентской части системы. Приводится описание реализации программного интерфейса (API) интеллектуальной транспортной геоинформационной системы «ITSGIS», предназначенного для загрузки информации об объектах различных видов на электронной карте. Большинство типичных задач, выполняемых клиентским приложением, выделены в ядро клиентского приложения. Обосновывается необходимость реализации данного программного интерфейса. Указывается, что для обеспечения взаимодействия компонентов «ITSGIS» с вновь разрабатываемыми модулями требуется устранить или снизить зависимость от конкретных инструментов разработки ПО. В процессе разработки компонентов системы решены задачи хранения настроек приложения и выбора используемых вариантов размеще-

ния журналов (в файле, в системном журнале событий, в базе данных на сервере).

Ключевые слова: программный интерфейс, система поддержки принятия решения, транспортная инфраструктура, интеллектуальная транспортная геоинформационная система «ITSGIS», реализация, модульная архитектура, функциональная спецификация ядра клиентской части компьютерная сеть, интеграция.

Введение

В настоящее время в обществе отмечается значительный прирост различных устройств связи, обеспечивающих мобильную передачу данных по беспроводным сетям, основу которых составляют радиосети.

Отсутствие привязки к проводным телефонным и компьютерным сетям, снижает зависимость владельцев мобильных устройств от определенной инфраструктуры коммуникаций, обеспечивающих обмен информацией в информационной сети Интернет и затраты на развертывание сетей в *ITSGIS*.

В современном мире технологии беспроводной передачи данных все чаще используются для дистанционного управления или контроля различных устройств, например, чайники, работающие через *Bluetooth* или розетки электропитания, которые контролируются через *Wi-Fi*. Вся эта тенденция обращена на удалённый контроль системами управления автоматическими устройствами.

Автоматические устройства актуальны в современной жизни. Они не только упрощают жизнь, как робот пылесос, но и спасают жизни, как робот спасатель, чьи способности могут превосходить человеческие в сотни раз. Так же автоматические устройства можно применять и в военной сфере, например, робот сапер или робот солдат. Очевидно, что для корректной и согласованной работы автоматических устройств они должны находиться в защищенной системе управления. Главной же проблемой реализации такой системы, является ее безопасность, так как передача ведется не через физические интерфейсы, а посредством беспроводной сети.

Так как, несмотря на явные удобства беспроводных сетей, сохранность передачи информации в них подвергается значительным рискам.

Основными рисками уязвимостями беспроводных сетей являются:

- неавторизованный доступ к беспроводной сети при отсутствии соответствующей защиты;

- прослушивание беспроводной сети (*ITSGIS*) – при отсутствии шифрования, передаваемая с помощью радиосигналов информация может быть легко перехвачена.

Цель работы: разработать роевую роботизированную систему обеспечения безопасности на основе беспроводной передачи данных.

Задачи:

- произвести анализ предметной области, роевых роботизированных систем;
- произвести анализ существующих способов обеспечения надежной, безопасной, беспроводной передачи данных;
- определить структуру роевой системы, состоящей из агентов. Выявить основные функции, реализуемые агентами роевой системы;
- разработать протокол взаимодействия агентов роевой системы, обеспечивающий безопасное, надежное, беспроводное соединение;
- спроектировать устройство, реализующее функции агента роевой системы;
- разработать программное обеспечение агента роевой системы.

Анализ предметной области

Современный мир, это мир информационных технологий. Информация стала самым главным ресурсом в прогрессивном обществе. В связи с всеобщей глобализацией информации, человечество начало стремиться к автоматизации, как процессов производства, так и бытовых действий. Развитие автоматизированных устройств, облегчило условия труда и жизни, но еще не достигло своего предела.

Некоторые компании производят автоматизированные устройства для спасения людских жизней, к примеру, иранская компания *RTS Ideas* протестировала спасательный дрон для работы на воде. А робот «*Andros EOD*» от компании *Remotec*, был специально создан для дистанционного обезвреживания взрывных средств, без опасности для человека. Роботизированная пожарная машина «Пеликан» от компании *Fire robot*, противостоит стихии там, где для человека это слишком опасно. Данное направление активно развивается.

Однако с развитием технологий, появляются новые решения, накопившихся проблем. Одним из таких решений, стала система роевого интеллекта. Такая система, как правило, состоит из множества взаимодействующих меж-

ду собой и окружающей средой интеллектуальных агентов. Идея данной системы была позаимствована из биологических систем.

Каждый агент следует очень простым правилам и, несмотря на то, что нет какой-то централизованной системы управления поведения, которая бы указывала каждому из них на то, что ему следует делать, локальные и, в некоторой степени, случайные взаимодействия приводят к возникновению интеллектуального группового поведения, неконтролируемого отдельными агентами. Пример такого взаимодействия можно наблюдать в поведении муравьев, пчел, волков, летучих мышей и других стадных существ. Хорошим примером актуальности данного направления служит научная статья, опубликованная в журнале *Science Robotics*. В ней исследователи описали, как им удалось заставить группу роботов из 300 агентов самоорганизоваться, не следуя заданному шаблону. В будущем данная технология может использоваться как при строительстве сложных конструкций, так и при исследовании космоса и дальних планет с передачей обратно обнаруженной информации.

Системы роевого интеллекта активно используются для создания воздушных, наземных, надводных и подводных автономных платформ, в военных целях. К примеру, были реализованы программы:

- *OFFSET (Offensive Swarm-Enabled Tactics)* – наступательная тактика с поддержкой роя;
- *TOBS (Tactical Offboard Sensing)* – тактическое внешнее наблюдение;
- *LOCUST (Low-Cost UAV Swarming Technology)* – технология роя БЛА низкой стоимости.

Роевые системы невероятно эффективны, но уязвимы к ложным сообщениям, перехватам сообщений и управления, внедрению, искажению сообщений, а также к *DDoS* атакам, что может привести к фатальным последствиям. Поэтому такие системы необходимо усиленно защищать.

Модель взаимодействия открытых сетей

Роевая система работает по стандартам взаимодействия открытых систем. Благодаря *OSI (Open System Interconnection)* создана общая модель протоколов, расположенных на различных уровнях и определяющих взаимодействие между сетевыми устройствами и программным обеспечением. Однако данная модель реализуема не только на сетевых адаптерах, но и на любых модулях беспроводной передачи данных.

Рассмотрим модель взаимодействия открытых систем, показанную в таблице 1.

Таблица 1. Модель OSI

Уровень		Тип Данных	Функции
1	Физический (<i>phisical</i>)	Биты (<i>Bit</i>)	Работа со средой передачи, сигналами и двоичными данными
2	Канальный (<i>data link</i>)	Биты/Кадры (<i>Bit/Frame</i>)	Физическая адресация
3	Сетевой (<i>network</i>)	Пакеты (<i>Packet</i>)	Определение маршрута и логическая адресация
4	Транспортный (<i>transport</i>)	Сегменты/Дейтаграммы (<i>segment/datagram</i>)	Прямая связь между конечными пунктами
5	Сеансовый (<i>session</i>)	Данные	Управление сеансом связи
6	Представления (<i>presentation</i>)		Представление и шифрование данных
7	Прикладной (<i>aplication</i>)		Доступ к сетевым службам

Классификация сетевых угроз

Основная задача по реализации роевой роботизированной системы, это обеспечение безопасной передачи данных. Роевая система имеет ряд уязвимостей, которые вытекают из принципа ее устройства. Однозначно данная система так же подвержена и сетевым угрозам.

Угрозы безопасности классифицируются по характеру на активные и пассивные, по цели реализации, нацеленные на нарушение конфиденциальности, целостности и доступа к информации в системе, по условию начала реализации угрозы, по наличию или отсутствию обратной связи, по расположению злоумышленника, а также по соотношению количества нарушителей и элементов информационной системы. Общая схема классификации угроз безопасности представлена в таблице 2.

Сама роевая система уязвима как к пассивным, так и к активным атакам. Основная защита системы направлена на доступ к информации, так как система состоит из децентрализованных самоорганизующихся агентов, которым необходима бесперебойная связь с системой. Однако вероятность реализации данных угроз можно свести к минимуму за счет аппаратно-программных средств.

Таблица 2. Общая схема классификации угроз безопасности

Квалификация угроз безопасности информации			
По характеру взаимодействия	Пассивные	Активные	
По расположению нарушителя	Межсегментные	Внутрисегментные	
По наличию обратной связи	С обратной связью	Без обратной связи	
По цели воздействия	Целостность информации	Конфиденциальность информации	Доступность информации
По условию начала реализации	По событию в системе	По запросу от системы	Безусловные
По соотношению нарушителей и атакуемых хостов	Один к одному	Один ко многим	Распределенные

Проектирование и реализация роевой роботизированной системы

Общее строение структуры сети включает в себя несколько роботизированных устройств, снабженных модулями приема-передачи и программой взаимодействия роя, реализуемой микроконтроллером.

При рассмотрении роевой роботизированной системы актуальными параметрами являются:

- зона охвата сигнала;
- количество одновременно подключённых устройств;
- энергоэффективность;
- скорость передачи и обработки данных;
- стоимость устройства.

В данной работе рассматривается проектирование роботизированного устройства, основанного на роевой системе, а также основные элементы, роевой системы.

Обоснование и выбор микроконтроллера

В настоящее время популярными для использования являются микроконтроллеры семейств *AVR*, *STM32*, *PIC*, *ESP*. Модули беспроводной связи можно подключить к любому микроконтроллеру через интерфейс, но так как принципы работы микроконтроллеров и их характеристики различаются между собой, то и эффективность передающего-принимающего устройства в различных реализациях будет отличаться, при этом различия будут все более

заметны при увеличении количества подключенных устройств. К примеру, микроконтроллер *ESP8266*, семейства *ESP*, уже имеет вшитый модуль *Wi-Fi*, но при этом не имеет флеш-памяти в электронной схеме *SoC (System on a Chip)*.

Микроконтроллеры семейства AVR

AVR – семейство восьмибитных микроконтроллеров с разделением адресных пространств программ и данных, использующих систему упрощения инструкций для более эффективного декодирования команд.

Процессор *AVR* работает в тридцати двух восьмибитных регистрах и имеет смешанную архитектуру процессора, основанную на *RISC (Reduced Instruction Set Computer)*, но с ограничением методов адресации данных.

Главным недостатком семейства микроконтроллеров *AVR* (рисунок 1) является малая вычислительная мощность восьмиразрядного ядра, а также высокая сложность реализации систем беспроводной передачи данных. Преимуществом является простая настройка, средняя стоимость, не превышающая 600 рублей.

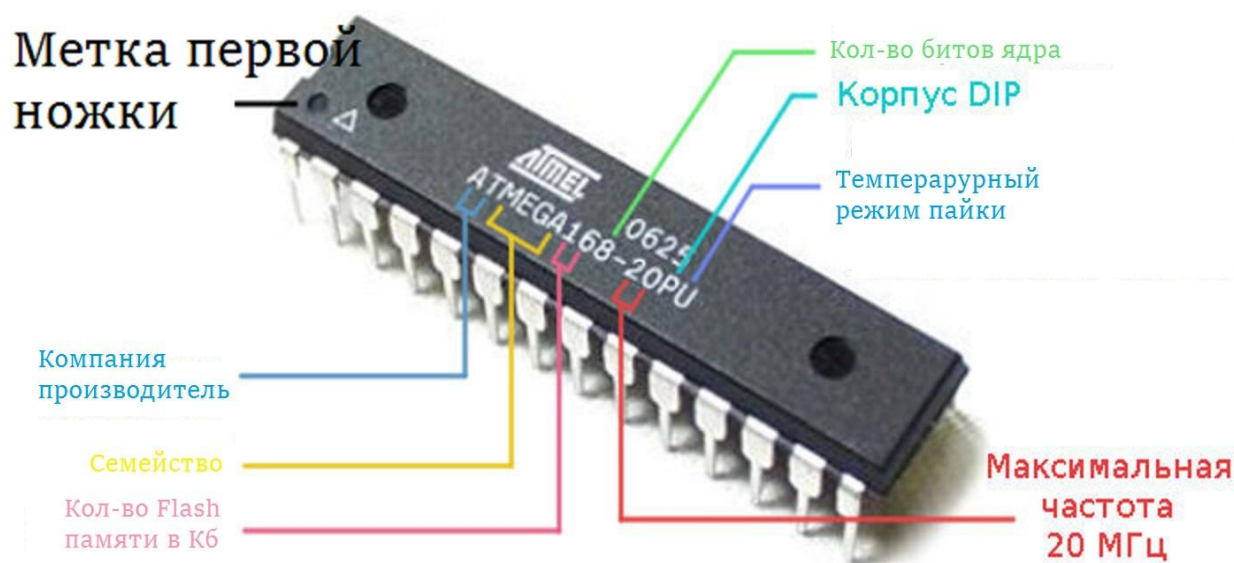


Рисунок 1. Микроконтроллер семейства AVR

Микроконтроллеры семейства STM32

STM32 – семейство тридцати двух битных микроконтроллеров (рисунок 2) с ядром на архитектуре *ARM (Advanced RISC Mashine)*. Каждый микроконтроллер данного семейства состоит из ядра процессора, статической памяти запоминающего устройства с произвольным доступом, флешпамяти и периферийных интерфейсов.

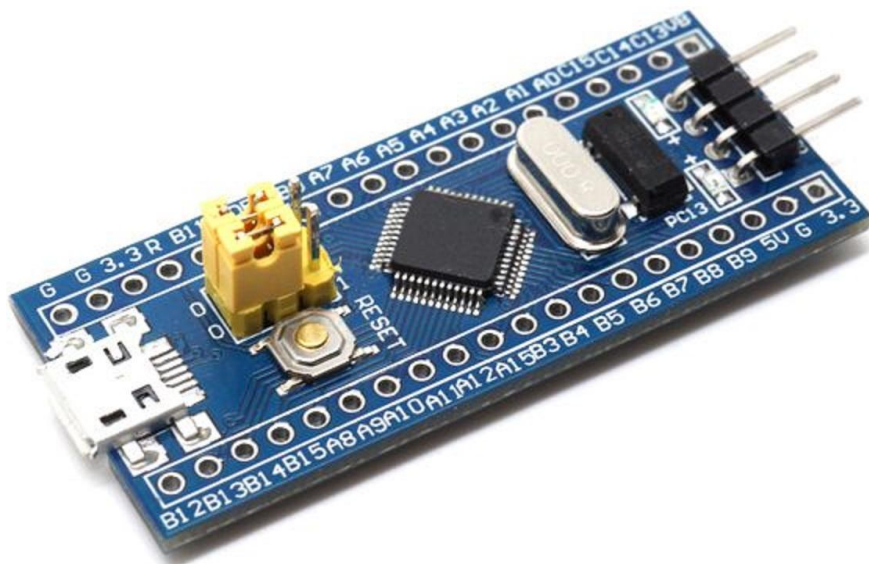


Рисунок 2. Микроконтроллер семейства *STM32* на плате *Blue pill*

Преимуществом данного семейства микроконтроллеров является поддержка программирования любым языком, в частности *JavaScript*, легкая пере прошивка и эмуляция устройств, а также простая реализация систем беспроводной передачи данных. Минусом данного семейства микроконтроллеров является средняя стоимость в 1200 рублей.

Микроконтроллеры семейства *PIC*

PIC – серия восьми, шестнадцати и тридцати двух битных микроконтроллеров с разделением адресных пространств программ и данных (рисунок 3).

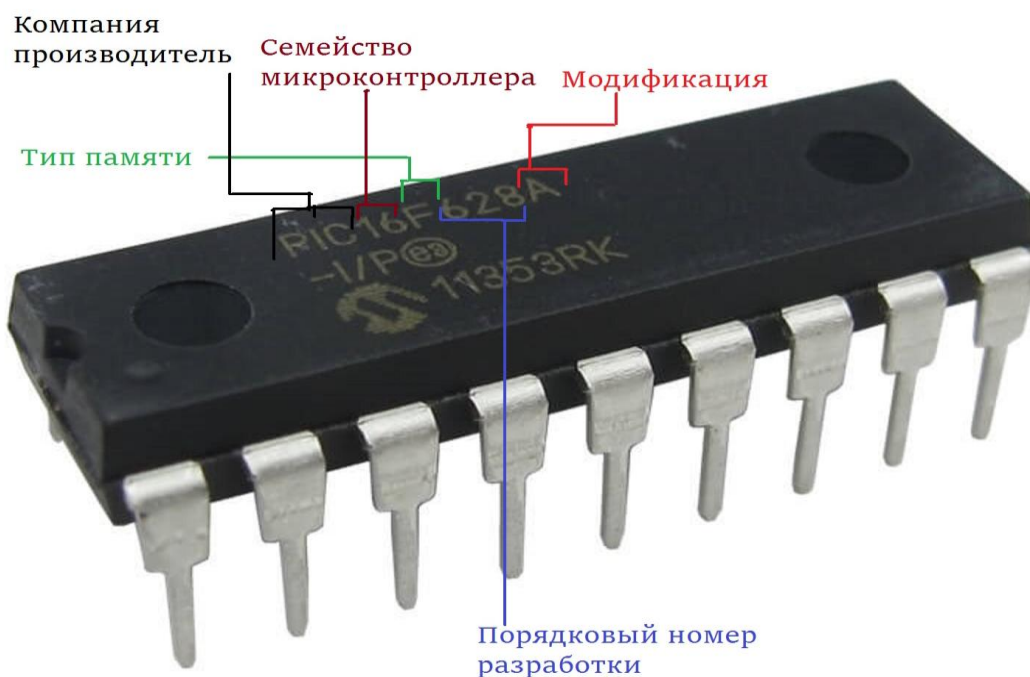


Рисунок 3. Микроконтроллер серии *PIC*

Микроконтроллеры данной серии разбиваются на три больших множества по своей разрядности ядра. Восьми битное ядро микроконтроллеров семейств *PIC10/12/16/* и *PIC18* строятся на архитектуре *BASELINE* или *MID-RANGE*. Данные семейства имеют те же недостатки что и семейство микроконтроллеров *AVR*. Преимуществом является простота программирования и наличие независимых от тактового генератора счётчиков. Семейство тридцати двух битных микроконтроллеров *PIC32* построено на ядре *MIPS32*, с низким энергопотреблением и наилучшем быстродействием. Средняя стоимость таких микроконтроллеров у производителя 17000 рублей.

Микроконтроллеры семейства *ESP*

ESP – семейство тридцати двух битных микроконтроллеров, с архитектурой ядра *xtensa*, представленное серией *ESP32* (рисунок 4) и семейством *ESP8266* (рисунок 5). Особенность данных микроконтроллеров заключается в изначально интегрированном модуле *Wi-Fi*, что позволяет изначально обеспечить будущее устройство средством беспроводной передачи данных. Концепция применения данного семейства заключается в *IoT* (*Internet of Things*).

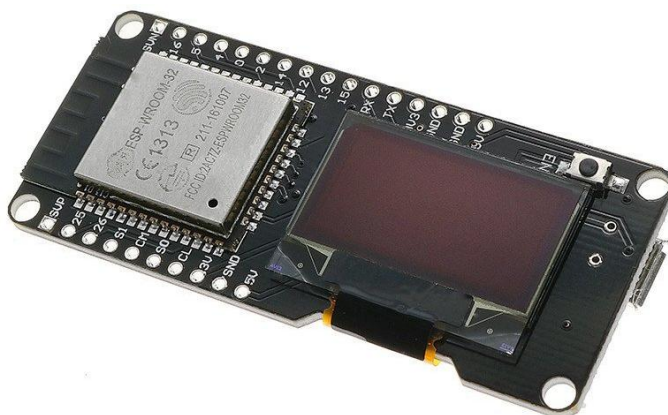


Рисунок 4. Микроконтроллер *ESP32*

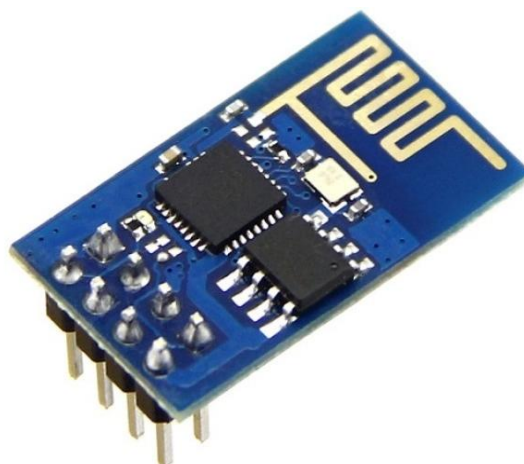


Рисунок 5. Микроконтроллер *ESP8266*

В новой серии микроконтроллеров *ESP32* кроме *Wi-Fi* модуля интегрирован так же и *Bluetooth* модуль, что расширяет его возможности для приема передачи данных. Данная серия широко используется в коммерческих и промышленных целях. Средняя стоимость микроконтроллера 800 рублей.

Обоснование и выбор интерфейсов

Для реализации функционала роботизированного устройства в рамках роевой системы, необходимо обеспечить взаимодействие датчиков, приемопередатчиков и микроконтроллера. Взаимодействие обеспечивают интерфейсы передачи данных. Данные интерфейсы служат для настройки, программирования, а также тестирования устройств и системы. От интерфейсов зависит возможность подключения устройства к различному оборудованию и обмен информацией между роевой роботизированной системой. В роевой системе взаимодействие микроконтроллеров обеспечивается с помощью адаптеров беспроводной связи, посредством интерфейсов.

Существует большое количество интерфейсов, позволяющих реализовать взаимодействие между микроконтрольными устройствами. Рассмотрим некоторые из них.

Интерфейс SWD

SWD (Serial Wire Debug) – последовательная отладка шины, является улучшенной версией интерфейса *JTAG*, требующей только четыре сигнальных выводов вместо двадцати у предшественника. Такое уменьшение благоприятно сказалось на удобстве работы с протоколами отладки, но уменьшила функционал и сферу применения интерфейса. Режим отладки *SWD* стал более надежным, в сравнении с *JTAG* в высокоскоростном режиме, так как программа загрузки *JTAG* в случае большого количества данных может завершиться неудачно. В случае нехватки интерфейсов ввода-вывода, так же можно использовать эмуляцию *SWD*, из-за поддержки меньшего количества контактов. Главным же преимуществом данного интерфейса является его малый размер. Интерфейс *SWD* в сравнении с *JTAG* показан на рисунке 6.

Интерфейс *SWD* в основном необходим для: тестирования печатных плат, прошивки микросхем, отладочных работ при проектировании аппаратнопрограммного комплекса на плате.

Интерфейс SPI

SPI (Serial Peripheral Interface) – трехпроводной периферийный последовательный интерфейс. Данный интерфейс организует обмен данными между несколькими устройствами, удобен для настройки взаимодействия между микроконтроллерами. *SPI* может входить в состав устройства роевой роботизированной системы.

зированной системы для соединения его с другими устройствами использующими микроконтроллерную архитектуру. В случае такого взаимодействия можно получить расширение возможностей системы.

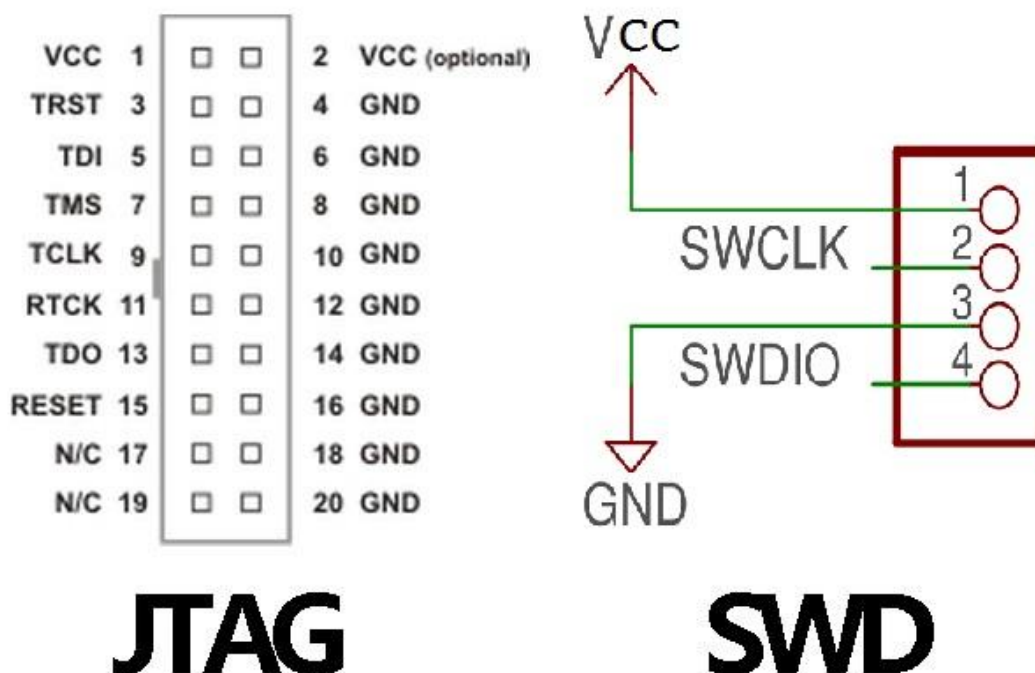


Рисунок 6. Интерфейс SWD в сравнении с JTAG

В данном интерфейсе используются четыре цифровых сигнала:

- *MOSI* – выход ведущего, вход ведомого. Необходим для передачи данных от ведущего к ведомому;
- *MISO* – вход ведущего, выход ведомого. Необходим для передачи данных от ведомого к ведущему;
- *SCLK (SCK)* – последовательный тактовый сигнал, для ведомых устройств;
- *SS* – выбор ведомого.

Работает *SPI* по принципу ведущий-ведомый (рисунок 7). Протокол *SPI* выполняет обмен данными между двумя сдвигowymi регистрами, которые работают в дуплексном режиме. Особенностью данного протокола является его идентичность логике работы сдвигового регистра.

При окончании конфигурации *SPI* интерфейса можно сразу начинать обмен данными с ведомыми устройствами. Так как данный интерфейс не описывает конкретный протокол для передачи данных, то различий между ведущим и ведомым устройствами при настройке в *STM32CubeIDE*, через функции *HAL* нет. Различия между устройствами отображается в параметре режимов.

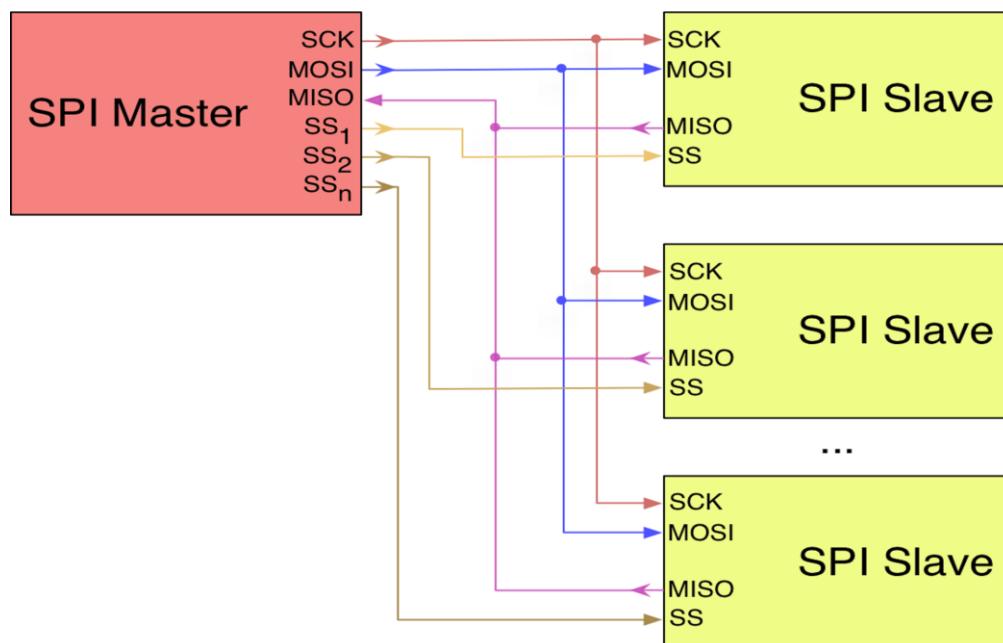


Рисунок 7. SPI взаимодействие ведущий-ведомый

SPI синхронный интерфейс, в котором любая передача синхронизируется с тактовым сигналом ведущего. Частота следования битовых интервалов определяется синхросигналом по *SCLK*, генерируемым ведущим устройством. Ведомые используют данный сигнал для определения момента изменения битов на линии. На обоих устройствах ведется счетчик импульсов синхронизации, который может быть сброшен при выключении *SPI*. Благодаря тактированию одним сигналом и ведущего и ведомого, система является стабильной.

Интерфейс *I2C*

I2C (*Inter-Integrated Circuit*) – последовательный ассиметричный, двухпроводной интерфейс, является аналогом интерфейса *TWI* (*Two-wire Serial Interface*). Данный интерфейс позволяет объединять до 127 устройств, благодаря двунаправленной шине, состоящей из линии данных (*SDA*) и линии тактового сигнала (*SCL*). *I2C* применяется для соединения периферийных компонентов с микроконтроллерами.

Данный интерфейс является синхронным, тактовый сигнал генерирует ведущее устройство, а линией данных может управлять как ведущий, так и ведомый в зависимости от направления передачи. В качестве единицы обмена информацией используется пакет данных, имеющий начальное и конечное условие. Ведущий в начало каждого пакета помещает байт, в котором указана информация об адресе ведомого и направлении передачи. Сами данные передаются массивами по восемь бит, после каждого переданного массива передается бит подтверждения приема.

Работает *I2C* по принципу ведущий-подчиненный (рисунок 8), ведущие устройство инициирует передачу данных, передавая адреса подчиненного устройства и тип передачи на чтение или запись. В случае одновременной инициализации передачи несколькими ведущими устройствами, применяется приоритетный механизм. Устройство, подключенное к шине должно быть либо ведущим, либо подчиненным, но для возможности приема-передачи, должно быть, как минимум одно ведущие устройство.

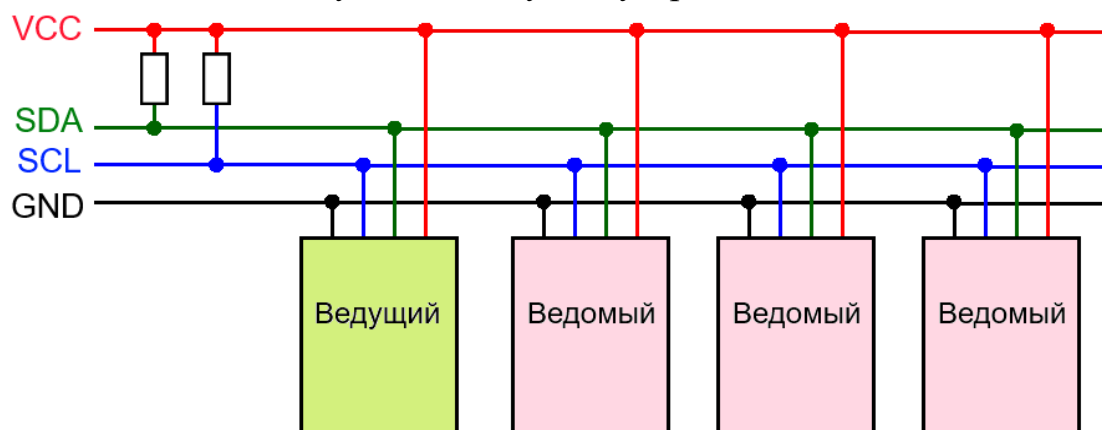


Рисунок 8. Соединение устройств по шине *I2C*

I2C имеет ряд преимуществ таких как: программная определяемость, простая диагностика сбоев и отладка, возможность подключения и отключения устройств без последствий для системы.

Интерфейс *CAN BUS*

CAN (Control Area Network) – последовательная магистраль, обеспечивает подключение устройств периферии в магистраль. Использует протокол, поддерживающий нахождение нескольких ведущих устройств, коррекцию ошибок, высокую помехоустойчивость.

Данный интерфейс предназначен для создания высоконадежных каналов связи в распределенных системах управления. Данные в нем передаются кадрами, состоящими из идентификатора в одиннадцать бит или двадцать девять бит, в зависимости от формата и самого блока с данными до восьми байт. Идентификатор описывает содержимое блока данных, для определения приоритета, при передаче несколькими устройствами.

Скорость передачи данных в данном интерфейсе задается программное, исходя из пропускной способности линии связи, числа подключенных устройств и расстояния.

Стандарт интерфейса поддерживает любой способ передачи данных, включая беспроводной, но чаще используется физический канал связи (рисунок 9).

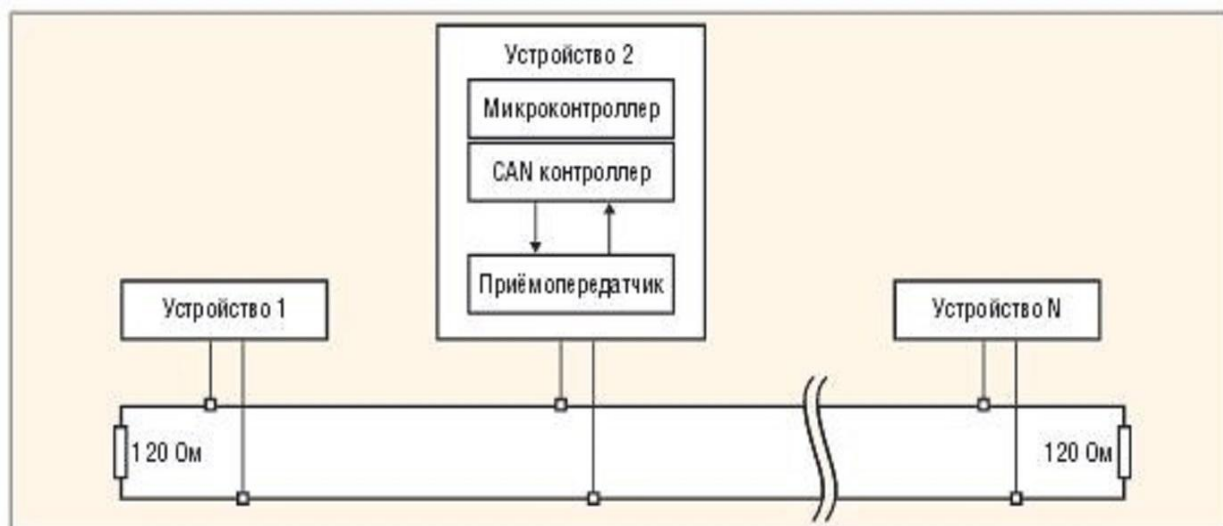


Рисунок 9. Схема подключения по *CAN BUS*

Интерфейс *USB*

USB (Universal Serial Bus) – интерфейс универсальной последовательной шины, рассчитан на подключение до 127 различных устройств. Данный интерфейс имеет четыре версии развития: *USB 1.0*, *USB 2.0*, *USB 3.0* и *USB4*, а также несколько стандартов и типов (рисунок 10). Интерфейс *USB 1.0* и *2.0* работают в полудуплексном режиме, *USB 3.0* и *USB4* в дуплексном, из-за этого в последних версиях возросла скорость передачи данных до 40 гигабит в секунду. В устройстве роевой роботизированной системы используется интерфейс *micro-USB type B*. В спецификации *USB 1.0* существуют всего три типа *A*, *B* и *AB*, первый используется на стороне контроллера, второй, на стороне периферийного устройства, третий тип подходит для обоих.

В *micro-USB type B* используется 5 ножек:

- 1) *Vbus* – ножка напряжения, рассчитана на 5 вольт;
- 2) *PD (Data Positive)* – ножка приема-передачи данных логической единицы;
- 3) *ND (Data Negative)* – ножка приема-передачи данных логического нуля;
- 4) *ID* – определяет конец кабеля, в случае с *type A*, подключена к земле, в случае с *type B*, не выполняет никаких функций;
- 5) *GND* – ножка заземления.

Данные по *USB* интерфейсу передаются дифференциально по принципу кодирования *NRZI (Non Return to Zero Invertive)*, состояния битов определяется по разности потенциалов между линиями данных, при условии напряжения на одной из линий выше относительно линии земли на 2.8 вольта. Логическая единица передается при напряжении на линии *PD* выше, чем на линии

земли на 2.8 вольт, при значении линии *ND* ниже 0.3 вольт, в случае логического нуля наоборот.

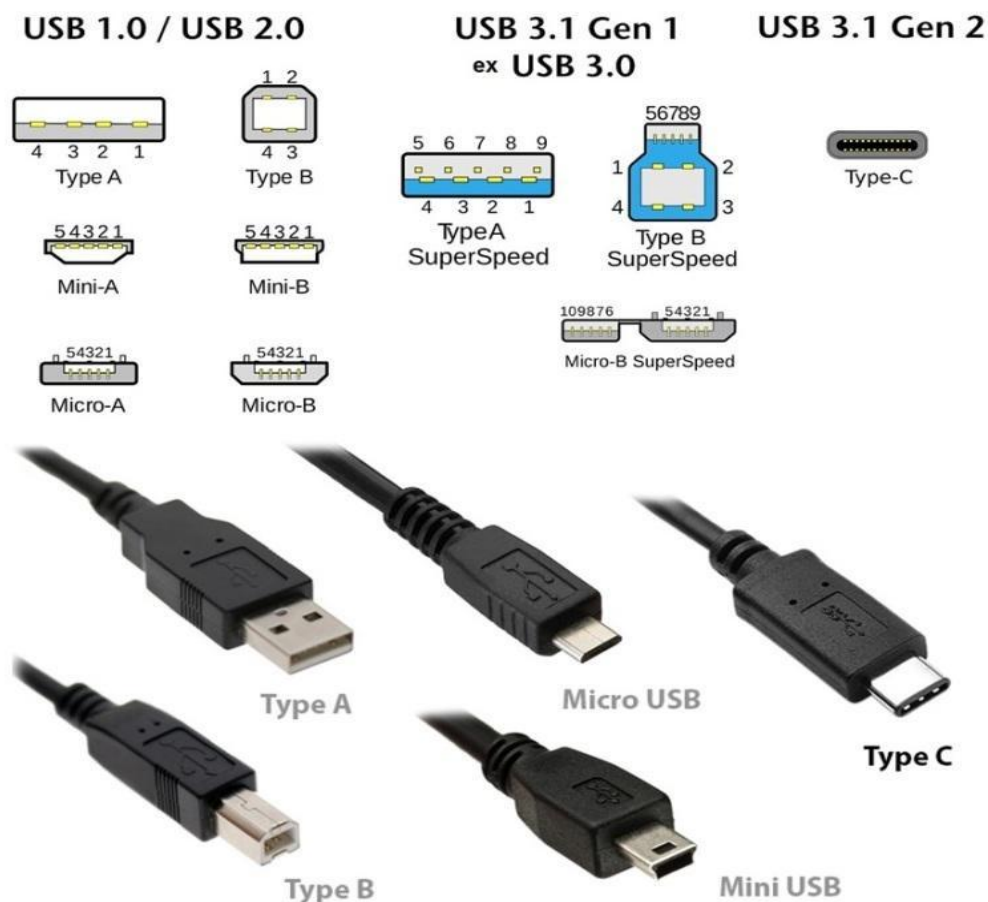


Рисунок 10. USB интерфейсы различных стандартов

Все информация по интерфейсу *USB* передается кадрами, каждый кадр состоит из пакета начала кадра (*SOF, Start Of Frame*), набора транзакций и пакета (*EOF, End Of Frame*). Устройства, оснащенные интерфейсом *USB*, имеют два механизма для автоматического определения функциональности подключаемой системы.

Обоснование и выбор модуля беспроводной передачи данных

Модуль беспроводной передачи данных, отвечает за связь устройств роевой роботизированной системы и определяет ее функционал. Все они работают на технологиях беспроводной передачи данных, таких как:

- *Bluetooth*;
- *GSM*;
- *Wi-Fi*;
- *LoRaWAN* ;
- *Li-Fi*.

Также в отдельную группу можно выделить технологии частотной передачи данных, где данные передаются модулированным излучением.

Для полноценной работы роевой роботизированной системы при ограничениях, необходимо подобрать совместимый с микроконтроллером модуль, который сможет удовлетворить заданным требованиям.

Технология *Bluetooth*

Bluetooth – технология беспроводной персональной связи. Принцип действия данной технологии основан на использовании радиоволн с частотой в дециметровом диапазоне (УКВ). Используются частоты от 2.402 ГГц до 2.48 ГГц. *Bluetooth* имеет многоуровневую архитектуру, состоящую из основного протокола, протокола замены кабеля, протоколов управления телефонией и заимствованных протоколов. Данные протоколы образуют стек протоколов *Bluetooth*. Примером модулей использующих данную технологию являются *HC-08* и *BT832*.

Модуль *HC-08* (рисунок 11) технологии *BLE* Данный модуль имеет приблизительную стоимость от 250 до 450 рублей. Скорость передачи данных 7,5 Кбит/сек, на рабочей частоте 2,4 ГГц. Радиус действия данного модуля до 80 метров. В активном режиме потребляет ток до 21 мА, в пассивном до 0,4 мкА, при напряжении питания от 3,3 В до 5 В.

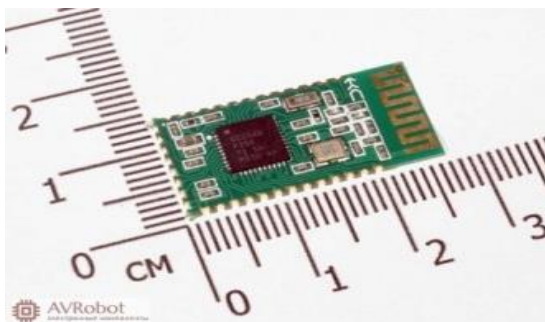


Рисунок 11. *Bluetooth*-модуль *HC-08*

Модуль *BT832* (рисунок 12) технологии *BLE*, имеет приблизительную стоимость от 850 до 1500 рублей. Скорость передачи данных до 6,25 Мбит/сек, на рабочей частоте 2,4 ГГц. Радиус действия данного модуля до 140 метров. В активном режиме потребляет ток до 7,5 мА, в пассивном до 150 мкА, при напряжении питания от 1,7 В до 3,6 В.



Рисунок 12. *Bluetooth*-модуль *BT832*

Технология GSM

Global System for Mobile Communications (GSM, СПС-900) - глобальный стандарт второго поколения цифровой мобильной сотовой связи с разделением каналов по времени и частоте. Обобщенная схема элементов взаимодействия в сети данной технологии представлена на рисунке 13.

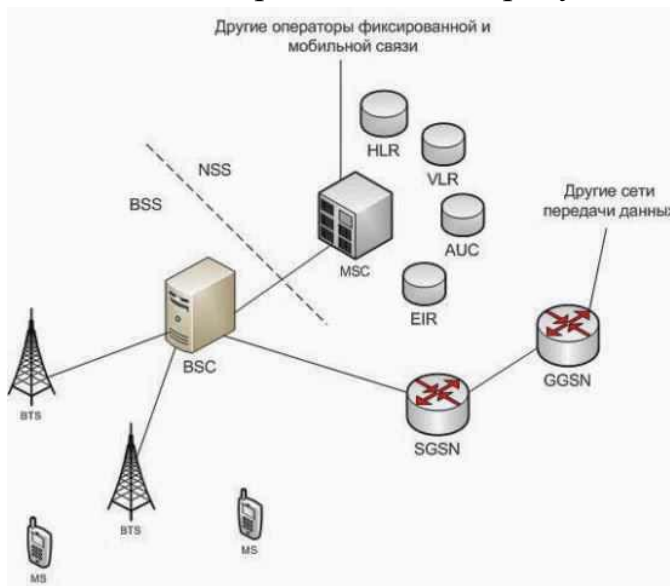


Рисунок 13. Основные элементы, технологии GSM

Сеть GSM делится на две системы. Каждая из них содержит функциональные устройства, которые, являются компонентами сети мобильной радиосвязи. Данные системы:

- *Base Station System (BSS)* – Система базовых станций сотовой связи стандарта GSM. Выполняет функцию по поддержанию и разрушению соединений между абонентским оборудованием, и системой коммуникации, и пакетной сетью передачи данных;
- *Network Switching System (NSS)* – Система коммутации сети сотовой связи стандарта GSM. Выполняет функцию коммутации каналов между абонентами сети и другими внешними сетями.

Примером модулей использующих данную технологию являются M66FA-04-STDN и SIM900D.

Модуль M66FA-04-STDN (рисунок 14) имеет приблизительную стоимость от 1200 до 1650 рублей. Скорость передачи данных 85,6 Кбит/с, на рабочих диапазонах частот 850 МГц, 900 МГц, 1800 МГц, 1900 МГц. Радиус действия данного модуля зависит от областей местоположения GSM, в среднем дальность связи между сотой и устройством от 0,3 до 7 километров, в зависимости от местности. Потребляет ток до 9,3 мА, при напряжении питания от 3,2 В до 4,8 В.



Рисунок 14. GSM-модуль M66FA-04-STDN

Модуль SIM900D (рисунок 15) имеет приблизительную стоимость от 780 до 1000 рублей. Скорость передачи данных 14,4 Кбит/с, на рабочих диапазонах частот 850 МГц, 900 МГц, 1800 МГц, 1900 МГц. Радиус действия данного модуля зависит от областей местоположения GSM, в среднем дальность связи между сотой и устройством от 0,3 до 7 километров, в зависимости от местности. Напряжение питания от 3,2 В до 4,8 В.



Рисунок 15. GSM-модуль SIM900D

Технология Wi-Fi

Wireless Fidelity (Wi-Fi) – беспроводная технология передачи данных. Принцип работы *Wi-Fi* основан на передаче зашифрованных сигналов посредством сверхвысокочастотных волн на небольшие расстояния. Схема *Wi-Fi* сети состоит минимум из точки доступа и клиента, также есть схемы клиент-клиент. Точка доступа транслирует идентификатор имени сети (*Service Set Identifier, SSID*), посредством спецпакетов со скоростью 100 килобит деленных на секунду. Данная скорость считается теоретически наименьшей пропускной способностью беспроводного канала.

Данную технологию активно применяют на микроконтроллерах семейства *ESP*, поэтому наилучшим вариантом является рассмотреть модуль, использующийся в плате. Примером таких плат являются *ESP-14* и *ESP-12E*.

Плата *ESP-14* (рисунок 16) имеет приблизительную стоимость от 300 до 450 рублей. Скорость передачи данных зависит от выбранного стандарта, в среднем 214 Мбит/с, на рабочих диапазонах частот 2,4 ГГц, 5 ГГц. Радиус действия данного модуля зависит от расположения модуля, в помещении или на местности, и от выбранного стандарта, в среднем дальность связи 275 метров. В активном режиме передачи потребляет ток до 200 мА, в пассивном режиме сна без потери связи до 1 мА, в пассивном режиме глубокого сна с откликом по внешнему событию до 0,1мкА, при напряжении питания от 2,95 В до 5,5 В.



Рисунок 16. *Wi-Fi*-модуль на плате *ESP-14*

Плата *ESP-12E* (рисунок 17) имеет приблизительную стоимость от 150 до 300 рублей. Скорость передачи данных зависит от выбранного стандарта, в среднем 214 Мбит/с, на рабочих диапазонах частот 2,4 ГГц, 5 ГГц. Радиус действия данного модуля зависит от расположения модуля, в помещении или на местности, и от выбранного стандарта, в среднем дальность связи 275 метров. Потребляет ток до 215 мА, при напряжении источника питания от 1,7 В до 3,6 В.



Рисунок 17. *Wi-Fi*-модуль на плате *ESP-12E*

Технология *LoraWan*

Сеть *Long Range Wide Area Network (LoRaWAN)*, является одним из типов сетей *Low Power Wide Area Network (LPWAN)*, энергоэффективных сетей дальнего радиуса действия. *LPWAN* сети, беспроводные и имеют широкий радиус охвата, их основным преимуществом является низкое энергопотребление, при этом данные сети имеют малый объем передачи данных не превышающий 1 килобайт. Однако, этого достаточно чтобы передавать телеметрию с конечного устройства на сервер диспетчера. Время жизни таких конечных устройств зависит от расписания передачи данных и объема батареи.

Типовая беспроводная сеть *LoRaWAN* представляет собой совокупность базовых станций, пересылающих сообщения между конечными устройствами и центральным сервером, отличительно характеризуются топологией звезда. Связь между базовыми станциями и центральным сервером осуществляется через проводную сеть передачи данных, а между базовыми станциями и конечными устройствами, через беспроводную сеть передачи данных, использующую широкополосную модуляцию *LoRa* или *FSK*. Связь между базовыми станциями и конечными устройствами является двусторонней, но предполагается, что основной объем данных передается от конечных устройств к базовым станциям. Данная технология обеспечивает скорость передачи в канале беспроводной сети от 0.3 Кбит/с до 50 Кбит/с. Для разделения каналов используется как набор частотных каналов, так и скоростей передачи. Примером модуля использующего данную технологию являются *eRIC*.

Модуль *eRIC* (рисунок 18) имеет приблизительную стоимость от 4 000 до 4 750 рублей. Скорость передачи данных зависит от выбранного стандарта, в среднем 8 Кбит/с, на рабочей частоте 915 МГц.



Рисунок 18. *LoRa*-модуль *eRIC*

Радиус действия данного модуля зависит от расположения модуля, в среднем дальность связи 16 километров. Потребляет ток от 10 мА до 90 мА, при напряжении питания от 3,3 В до 5,5 В.

Технология *Li-Fi*

Li-Fi использует видимый свет электромагнитного спектра от 400 ТГц до 800 ТГц в качестве оптического носителя для передачи данных. Основными компонентами базовой системы *Li-Fi* (рисунок 19) являются, светодиод, который служит источником передачи, и кремниевый фотодиод в качестве приемного элемента.

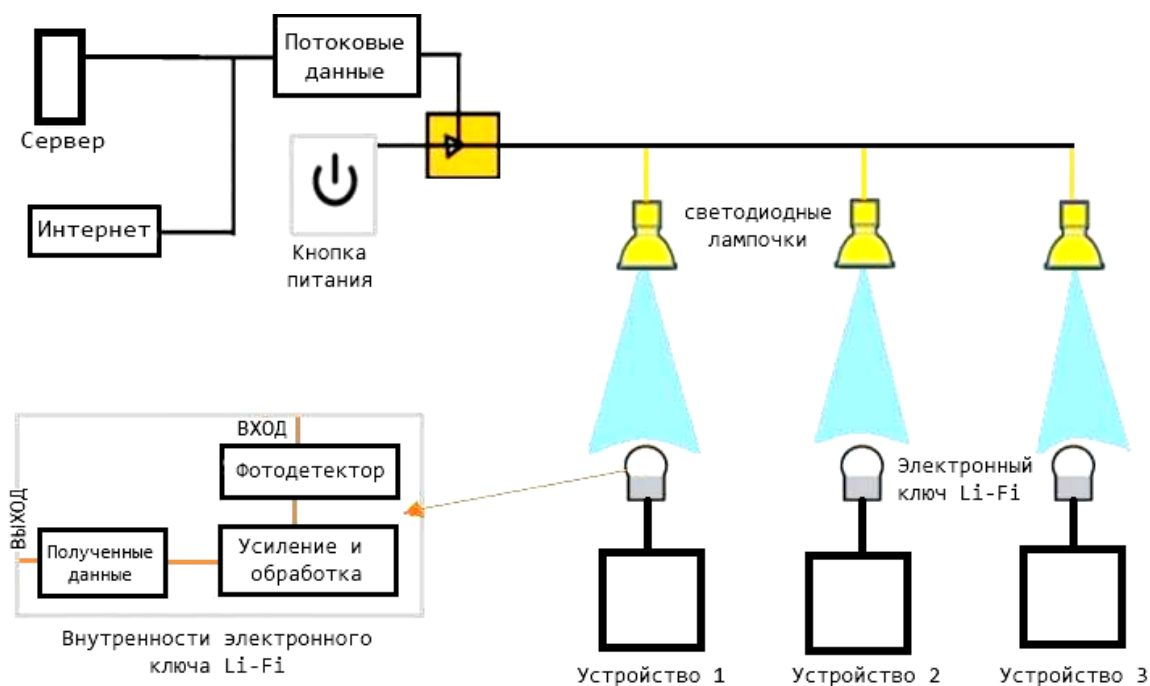


Рисунок 19. Архитектура сети *Li-Fi*

Примером элементов, входящих в структуру сети *Li-Fi* являются фотодиод *BPW20RF* и светодиод *TDS-P001L4J03*.

Фотодиод *BPW20RF* (рисунок 20) принимает длину волны от четырехсот нанометров до тысячи ста нанометров и имеет среднюю стоимость 700 рублей.



Рисунок 20. Фотодиод *BPW20RF*

Светодиод *TDS-P001L4J03* (рисунок 21) издает видимый свет длиной шестьсот нанометров и имеет среднюю стоимость 90 рублей.



Рисунок 21. Светодиод *TDS-P001L4J03*

Технология частотной передачи данных

Данная технология передачи данных основывается на модуляции излучения заданной частоты, в частности инфракрасной части спектра.

Преимуществами данной технологии являются ее помехоустойчивость, простая реализация модулей из недорогих элементов, возможность отражения сигнала, а также достаточная безопасность передачи данных. Недостатком же является малая дальность, зависимость от атмосферных условий и малая скорость передачи данных.

Примером элементов, входящих в структуру инфракрасной сети являются инфракрасный излучатель *TSAL6200* и инфракрасный приемник *TSOP31238*.

Инфракрасный излучатель *TSAL6200* (рисунок 22) на основе технологии *GaAlAs* с высокой мощностью излучения в 60 мВт и длиной волны в 940 нанометров. Средняя стоимость составляет 15 рублей.



Рисунок 22. Инфракрасный диод излучатель *TSAL6200*

Инфракрасный приемник *TSOP31238* (рисунок 23) принимает инфракрасное излучение с несущей частотой 38 кГц, оборудован предусилителем, имеет внутренний фильтр для частоты амплитудной модуляции с полной несущей. Средняя стоимость 70 рублей.



Рисунок 23. Инфракрасный приемник *TSOP31238*

Для разработки устройства, способного выполнять задачи роевой роботизированной системы, был выбран микроконтроллер *STM32F103C8T6* со следующими характеристиками:

- память: 64 Кбайт *Flash*-памяти и 20 Кбайт *SRAM*-памяти;
- *GPIO Pins* в количестве тридцати двух с возможностью внешнего прерывания;
- таймеры: три шестнадцати битных таймера и один шестнадцати битный широтно-импульсно модулированный таймер;
- пятнадцать *PWM Pins*;
- десять аналоговых каналов двенадцати битного аналого-цифрового преобразователя;
- два периферийных устройства *I2C*;
- три периферийных устройства с аппаратным управлением *USART*;
- два периферийных устройства *SPI*;
- *USB 2.0 Full Speed*;
- *CAN 2.0B*.

Также данный микроконтроллер имеет ряд преимуществ перед микроконтроллерами других семейств:

- работа с регистрами, памятью и портами ввода/вывода, а также проведение вычислений в тридцати двух битном режиме, что позволяет достичь большей вычислительной мощности при сохранении тактовой частоты и тепловыделении;
- архитектура вычислительного ядра *Cortex – M3*, ускоряющего вычисления арифметических операций;
- единое адресное пространство для памяти, что упрощает процедуру программирования и созданию функций по организации работы с ОЗУ и внутренней памятью;

- частота работы процессора до 72 МГц;
- наличие большого количества интерфейсов;
- контроллер прерываний, обеспечивающий работу прерываний с разным приоритетом, системные таймеры, использующиеся в качестве внутренних счетчиков и генерации событий по совпадению кода таймера с заранее заданным значением;
- среда программирования *CubeIDE*, использование языка C для написания исходного кода программы.

Для беспроводной передачи данных был выбран модуль ИК-излучения, состоящий из диода *TSAL6200* и ИК-приемника *TSOP31238*.

В рамках данного выбора средств передачи данных и микроконтроллера роевая система имеет ряд ограничений таких как:

- расстояние между устройствами не должно превышать 45 метров;
- не должен нарушаться диапазон рабочих температур от минус 25°C до плюс 85°C;
- угол обзора устройства в 45° не должен быть перекрыт поглощающими ИК-излучение объектами.

Обоснование и выбор программных средств

Для создания роевой роботизированной системы в качестве базы для программирования микроконтроллера используется среда разработки *STM32CubeIDE*. Данный выбор обусловлен полной совместимостью с аппаратными и отладочными средствами для настройки микроконтроллера. Выбор языка программирования C в качестве используемого при реализации программного кода обусловлен удобством при использовании стандартных библиотек для управления устройством.

STM32CubeIDE представляет собой C/C++ платформу разработки для микроконтроллеров *STM32*, обладающую возможностью конфигурации периферии, функциями генерации кода, компиляции кода и отладки.

Основные программные средства *STM32CubeIDE*:

- менеджер проектов для объединения отдельных файлов в группы;
- встроенный редактор исходного текста;
- средства автоматической компиляции и компоновки проекта;
- выбор процессора из собственной базы данных;
- конфигурирование выводов устройства, таймеров и периферии;
- создание проекта, генерирование кода инициализации периферии;
- настройка параметров тактирования;
- расширенные функции отладки с пошаговым режимом;
- мониторинг периферийных регистров и памяти;

- просмотр значений переменных в реальном времени;
- анализ системы и отслеживание в реальном времени;
- инструмент анализа ошибок процессора;
- синтаксическое выделение исходного кода;
- поддержка отладочных средств *ST-LINK (STMicroelectronics)*.

Описание функционала реализуемой роевой системы

В функционал устройства входит прием данных от других устройств, а также их передача на соседние устройства, локальное взаимодействие между собой. Каждый агент системы имеет набор простых правил и не имея центральной системы управления, имеет самоорганизующиеся поведение.

В реализуемой системе необходимо обеспечить взаимодействие множества роботизированных устройств между собой, получить, обработать и ретранслировать информацию (рисунок 24).

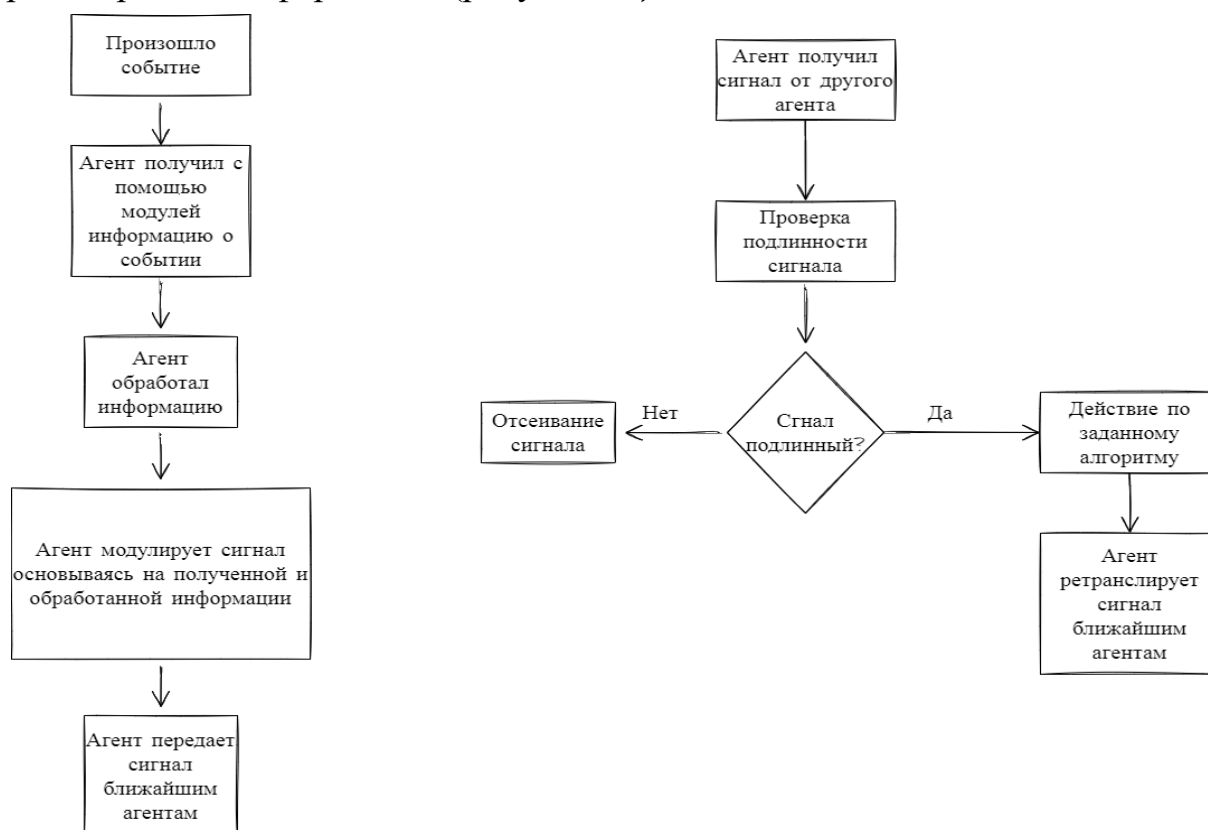


Рисунок 24. UML диаграмма действия агента с информацией

В случае роевой системы, возникают проблемы оптимизации задач и поиска решений, для решения данных проблем используются алгоритмы поисковой оптимизации. Также для самоконтроля и самоорганизации роя необходимо применять методы имитации социального поведения или поведения роя. В реализуемой роевой роботизированной системе ИТСГИС могут применяться следующие методы и алгоритмы:

- бактериальная оптимизация;

- алгоритм роя светлячков;
- сорняковый алгоритм;
- алгоритм кукушки;
- алгоритм обезьян;
- метод роя частиц (МРЧ);
- гармонический поиск;
- гравитационный поиск;
- электромагнитный поиск;
- алгоритм эволюции разума;
- стохастический диффузионный поиск;
- культурный алгоритм;
- миметический алгоритм;
- миграционный самоорганизующийся алгоритм;
- алгоритм мотылька и пламени;
- метод саранчи;
- метод стаи сальп;
- муравьиный алгоритм;
- пчелиный алгоритм.

Сам рой можно определить, как децентрализованную систему, состоящую из множества простых однообразных элементов, взаимодействующих между собой и окружающей средой для решения задач или достижения целей. Рой может формироваться двумя способами: восходящим и нисходящим. В первом случае несколько множеств агентов объединяются в одно множество. Во втором случае одно множество агентов разбивается на несколько не пустых множеств.

Взаимодействие роевой системы по беспроводному каналу показано на рисунке 25.

У каждого агента есть область взаимодействия с другими агентами системы, ограниченная дальностью действия модуля беспроводной передачи данных. Совокупность нескольких агентов образует роевую роботизированную систему. Взаимодействие пользователя с системой происходит через одного, выбранного агента, пользователю необходимо подключиться к агенту и с его помощью взаимодействовать с системой.

При такой реализации роевая система уязвима к перехвату управления и внедрению. В первом случае злоумышленник пытается подключиться к выбранному агенту и через него управлять системой. Во втором случае злоумышленник пытается выдать себя за агента системы.

Схема взаимодействия агентов показана на рисунке 26.

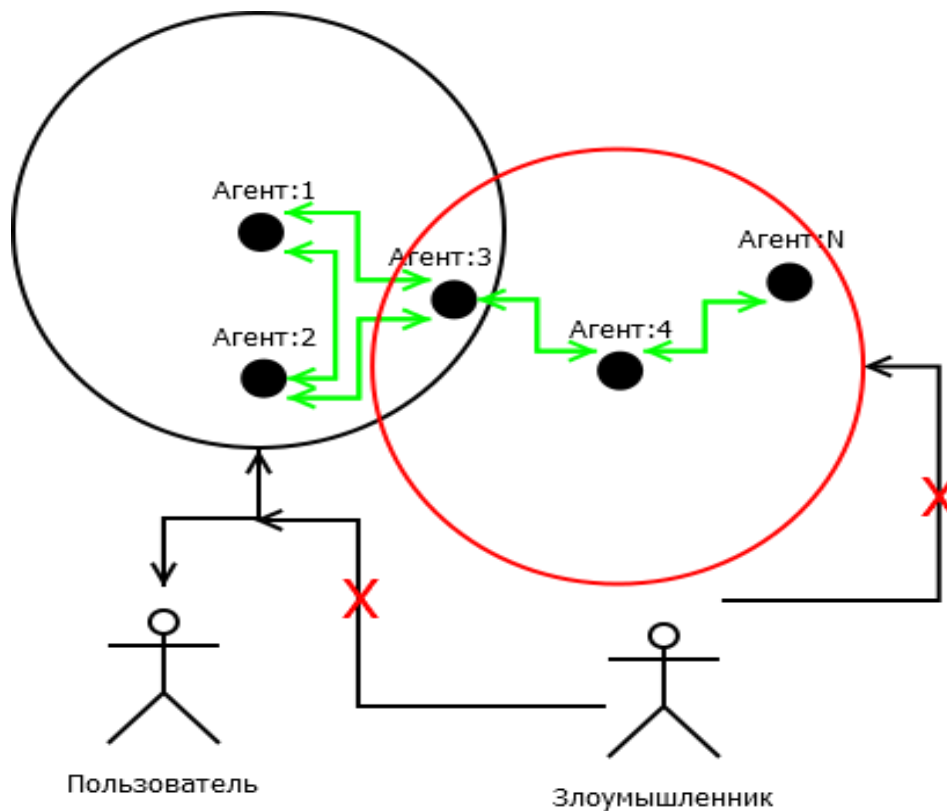


Рисунок 25. Взаимодействие роевой системы по беспроводному каналу

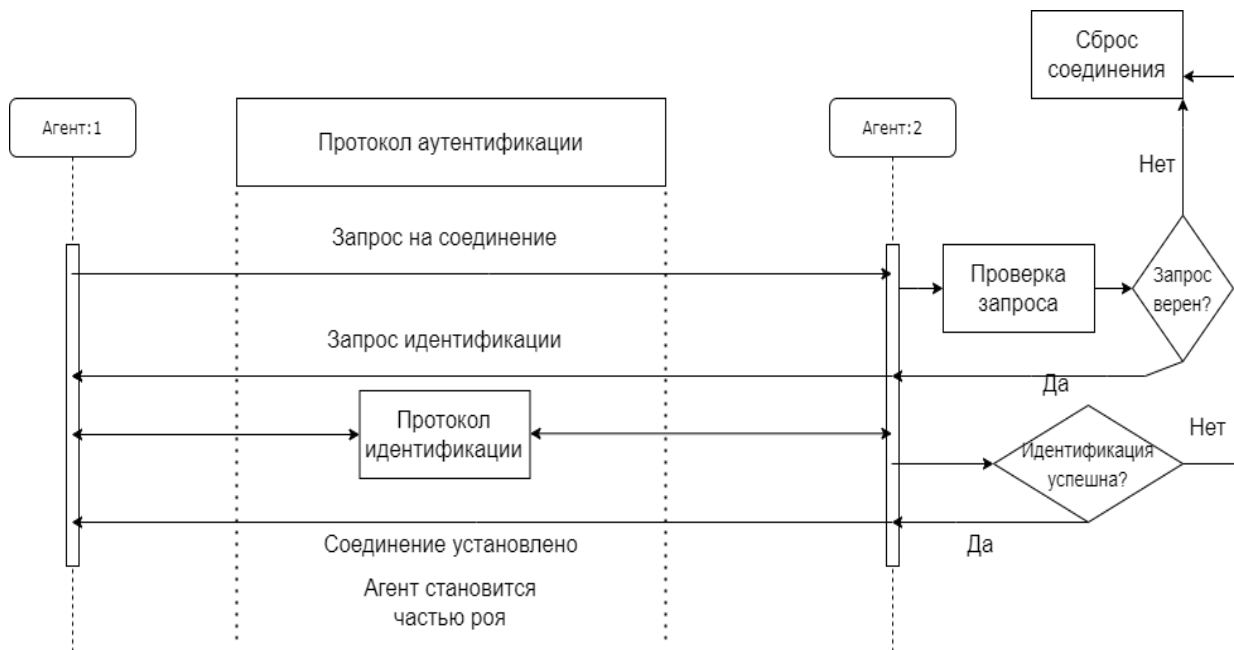


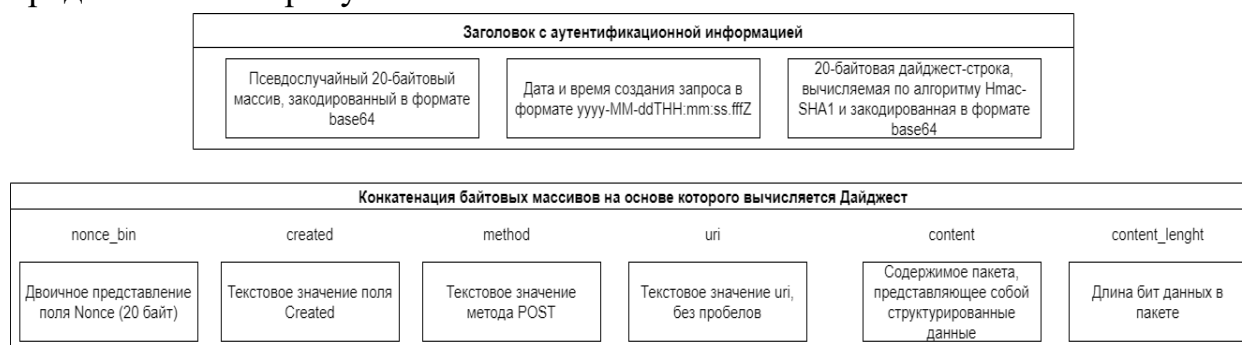
Рисунок 26. Схема взаимодействия агентов

При реализации данной схемы вероятность успешной атаки на рой, используя уязвимости внедрения, перехвата управления и ложных сообщений, зависит от используемых в протоколах алгоритмов, для идентификации и аутентификации агента.

Агент адресат должен сформировать запрос на соединение, по заданной форме в протоколе аутентификации, после чего данный пакет кодируется ал-

горитмом циклического избыточного кода и посылается агенту адресанту. На каждом шаге получения пакета от агента адресата проверяется правильность формирования аутентификационной информации и заголовок пакета. Агент может стать частью роевой системы, в случае если правильно сформированы пакеты данных, при передаче не была нарушена аутентификация и пройдена идентификация без блокировки. В конце каждого пакета данных ведется отсчет принятых и обработанных пакетов.

Аутентификационная информация и отсчетные части пакетов данных представлены на рисунке 27.



Аутентификационный заголовок читается корректным, если вычисленное адресантом значение дайджеста совпадает с переданным в заголовке, а GMT-время, переданное в поле Created, отличается от времени адресанта не более чем на 10 секунд.

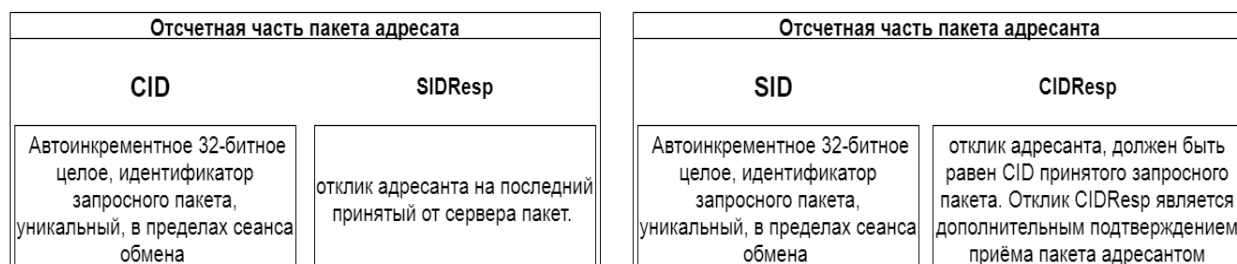


Рисунок 27. Аутентификационная информация и отсчетные части пакетов данных

Аутентификационный заголовок пакета данных (*ECNC-Auth*) состоит из случайного 20-байтового массива (*Nonce*), закодированного в формате *base64*, даты и время создания запроса (*Created*) в формате «уууу-ММ-ддТНН:мм:сс.fffZ» и 20-байтовой хэш-строки (*Digest*), вычисляемой по алгоритму *Hmac-SHA1* и закодированной в формате *base64*.

Хэш-строка представляет собой результат работы алгоритма *Hmac-SHA1* на конкатенации битовых массивов:

1. двоичное представление поля *Nonce* (20 байт);
2. текстовое значение поля *Created*;
3. текстовое значение метода *POST*;
4. текстовое значение *uri*, без пробелов;

5. содержимое пакета, представляющее собой структурированные данные;
6. длина бит данных в пакете.

Аутентификационный заголовок считается корректным, если вычисленное адресантом значение дайджеста совпадает с переданным в заголовке, а *GMT*-время, переданное в поле *Created*, отличается от времени адресанта не более чем на 10 секунд.

Агент адресат и агент адресант ведут отчет пакетов данных. Адресат, формируя пакет данных, добавляет в конец автоинкрементное 32-битное целое, идентификатор запросного пакета, уникальный, в пределах сеанса обмена (*CID*) и отклик адресанта на последний принятый от сервера пакет (*SIDResp*). Адресант, формируя пакет данных, добавляет в конец автоинкрементное 32-битное целое, идентификатор запросного пакета, уникальный, в пределах сеанса обмена (*SID*) и отклик (*CIDResp*), который должен быть равен *CID* принятого запросного пакета. Отклик *CIDResp* является дополнительным подтверждением приёма пакета адресантом.

Вероятность успешной атаки на рой, используя уязвимость ложных сообщений, зависит от используемых в протоколе алгоритмов, для вычисления хэш-функции и проверки подписи сообщения.

В качестве метода идентификации используется алгоритм Фейга-Фиата-Шамира, основанный на протоколе доказательства без разглашения. Хэш-функция вычисляется по алгоритму *SHA1*, проверка целостности информации выполняется по алгоритму *Hmac*. Помехоустойчивость системы обеспечивается использованием циклического избыточного кода *CRC*. Данная система позволяет изменять алгоритмы на более стойкие без изменения структуры системы.

Аппаратная реализация

Для работы микроконтроллера необходимо подключить тактовый генератор, интерфейсы программирования, разъемы для разграничения памяти (переключение режима работы процессора), взаимодействие между агентами осуществляется через инфракрасный интерфейс.

Для аппаратной реализации необходимо разработать принципиальную схему, которая включает в себя схемы: приемника сигналов, передатчик сигналов, обвязки микроконтроллера, системы питания, интерфейсной части роботизированного устройства, управления приводами робота.

1) схема *TSAL* (рисунок 28) – выполняет функцию передатчика. В зависимости от взаимного расположения устройств для управления внешними приводами подключенного реле, приема-передатчики, в расположении точка-

точка достаточно разместить на одной стороне печатной платы. В случае треугольника, на плате устройства передатчик от устройства приемника размещается под углом 45 градусов. В случае расположения на одной прямой, приема-передатчики располагаются с противоположенных сторон платы;

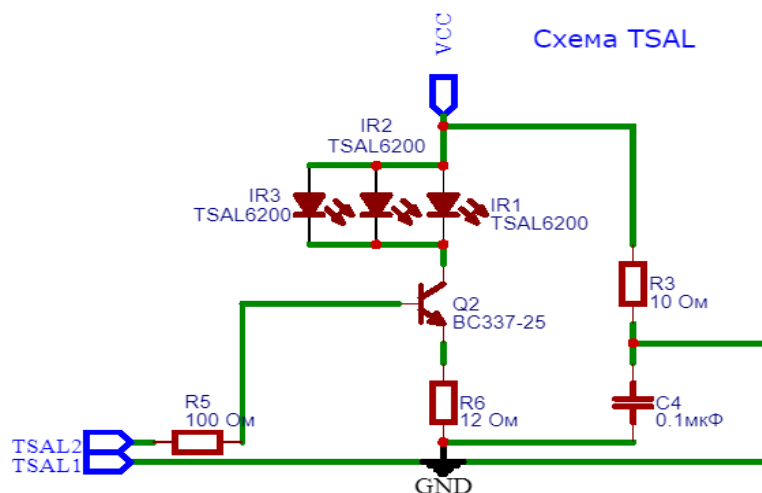


Рисунок 28. Схема TSAL

2) схема TSOP (рисунок 29), выполняют функцию приемника. В зависимости от расположения устройств имеет три точки размещения на плате;

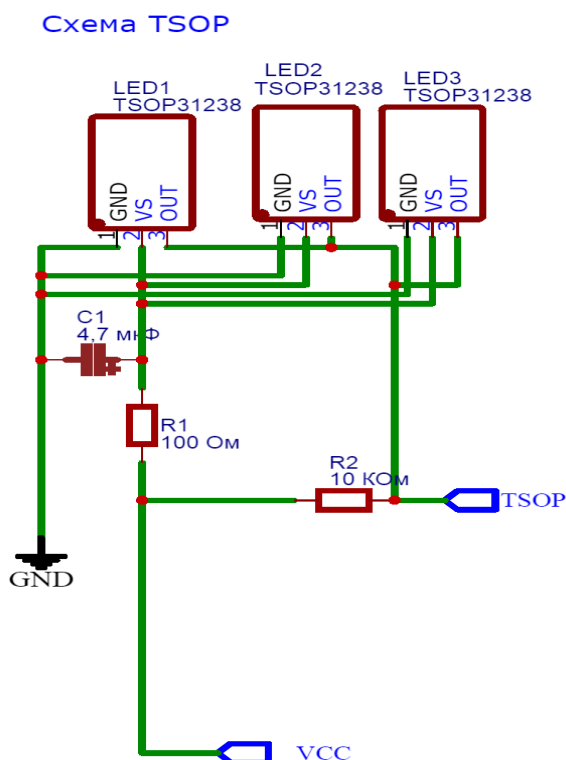


Рисунок 29. Схема TSOP

3) схема подачи питания показана на рисунке 30, эта схема позволяет преобразовать питание устройства от источника 12 вольт в 5 вольт;

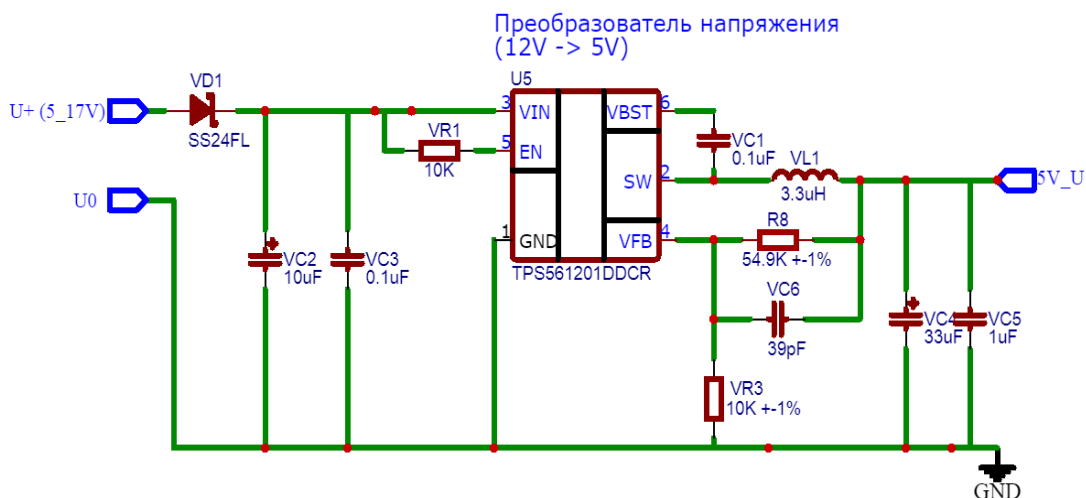


Рисунок 30. Преобразователь напряжения из 12 В в 5 В

4) схема питания микроконтроллера показана на рисунке 31, эта схема позволяет преобразовать питание устройства от преобразователя подачи питания с 5 вольт до 3,3 вольт;

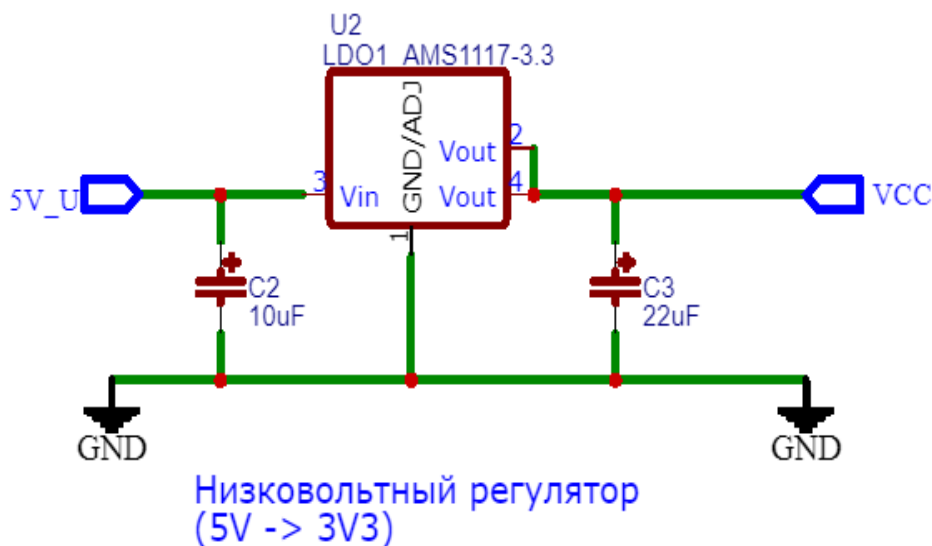


Рисунок 31. Преобразователь напряжения из 5 В в 3,3 В

5) схема реле управляющего сигнала (рисунок 32), выполняет функцию передачи преобразованного сигнала подключаемому устройству, имеет три вида сигнала комбинации которых возможно использовать для управления;

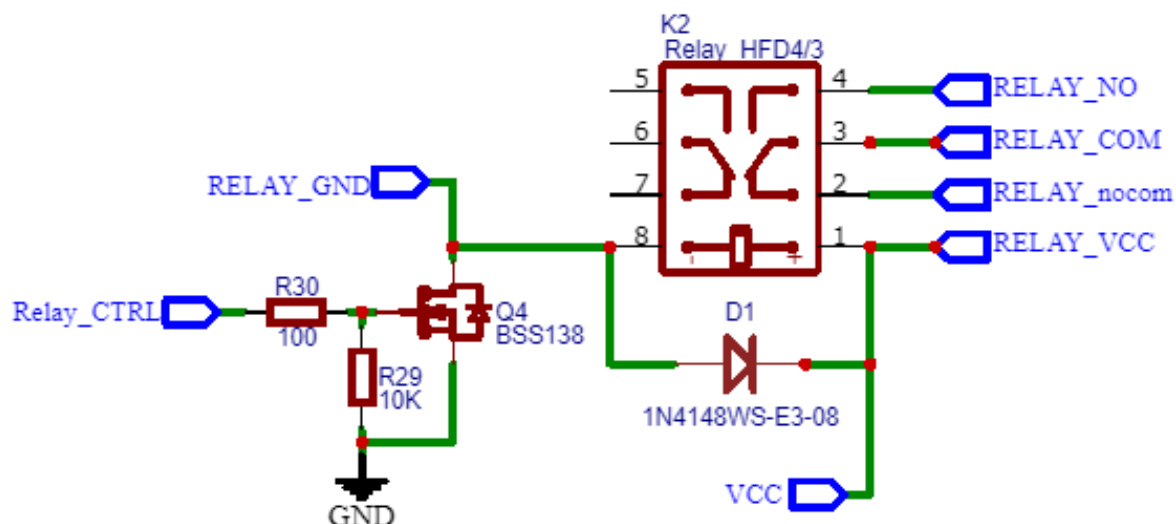


Рисунок 32. Реле управляющего сигнала

6) схема кнопки рестарта (рисунок 33), данная кнопка сбрасывает параметры микроконтроллера до заводской настройки, вшитой в оболочку;

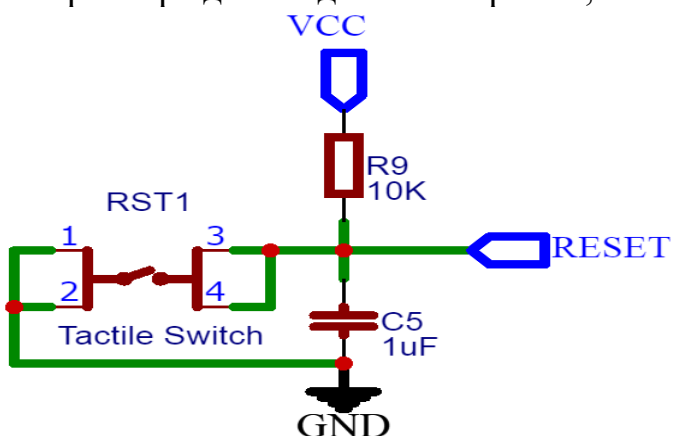


Рисунок 33. Кнопка рестарта

7) схема резонатора на 8 МГц (рисунок 34). Для функционирования и работы микроконтроллера необходимо синхронизировать выполнения операций в нем, для этого необходим тактовый резонатор;

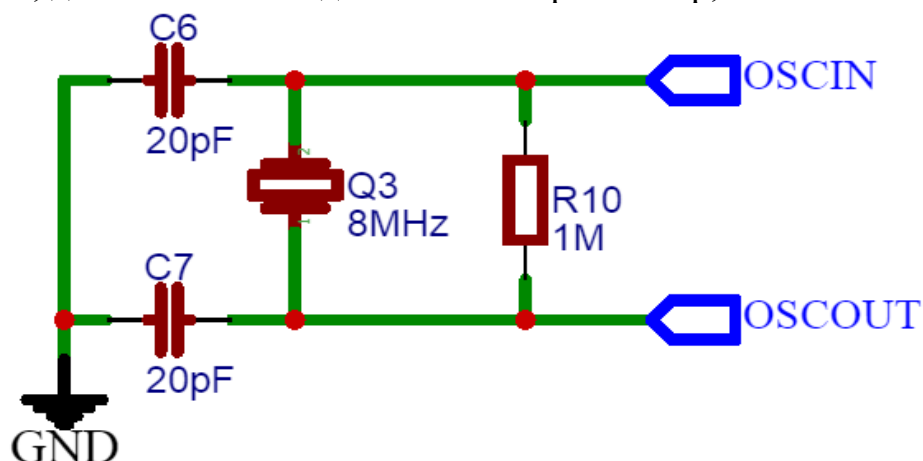


Рисунок 34. Резонатор на 8 МГц

8) схема резонатора на 32,768 кГц (часовой кварц) (рисунок 35). Для отсчета реального времени и выполнения задач имея первоначально понятную точку отчета, необходим часовой кварц:

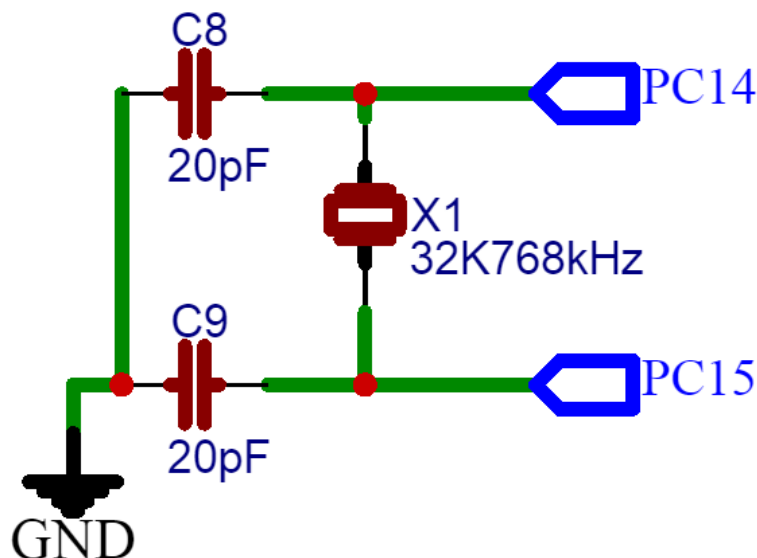


Рисунок 35. Резонатор на 32,768 кГц

9) схема разъема CWF-6R (male) (рисунок 36). Для подключения агента к дополнительным управляемым устройствам, в частности к приводам, необходимо средство подключения, в роли которого выступает данный разъем;

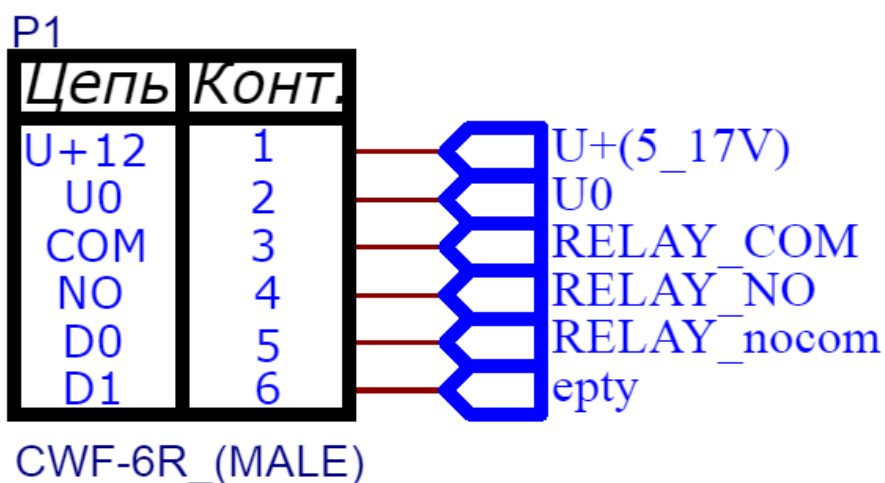


Рисунок 36. Разъем CWF-6R (male)

10) схема разъема USB-micro (рисунок 37). Для подключения периферийных устройств, объединение множества устройств в одно моноустройство необходим интерфейс, которым и является USB-micro;

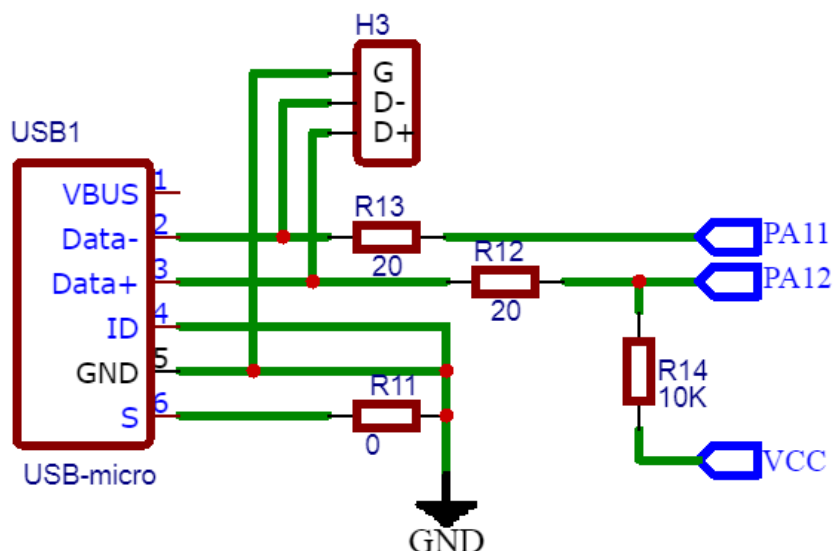


Рисунок 37. Разъем *USB-micro*

11) схема динамика-пищалки (рисунок 38). Данный модуль необходим для звукового уведомления пользователя, наглядно показывающего выполнение заданного алгоритма действий;

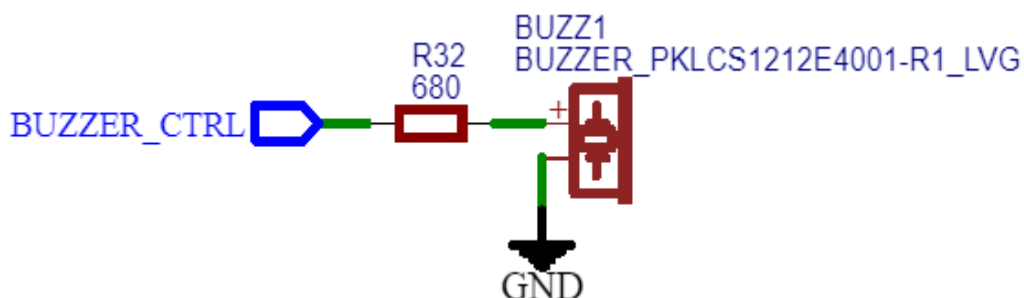


Рисунок 38. Динамик-пищалка

12) схема перемычек памяти микроконтроллера (рисунок 39). Данные перемычки необходимы для взаимодействия с памятью микроконтроллера, их корректное замыкание дает возможность работать с основной флеш-памятью, системной памятью и встроенной памятью *SRAM*;

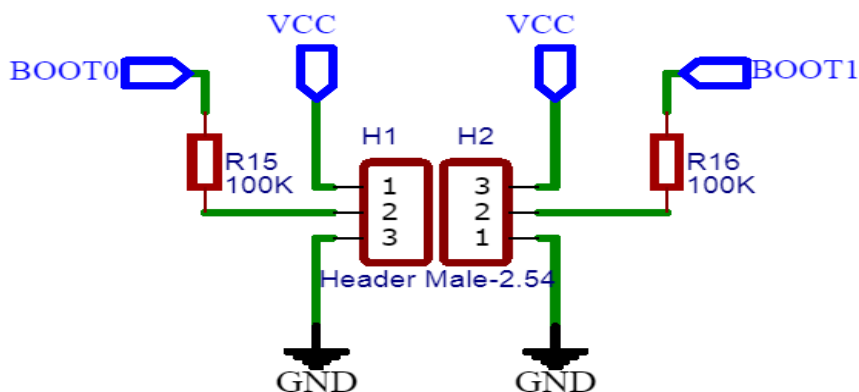


Рисунок 39. Перемычка управления памятью

13) схема микроконтроллера *STM32F103C8T6* (рисунок 40). Главная часть устройства, выполняющая все вычислительные, обрабатывающие и контрольные функции;

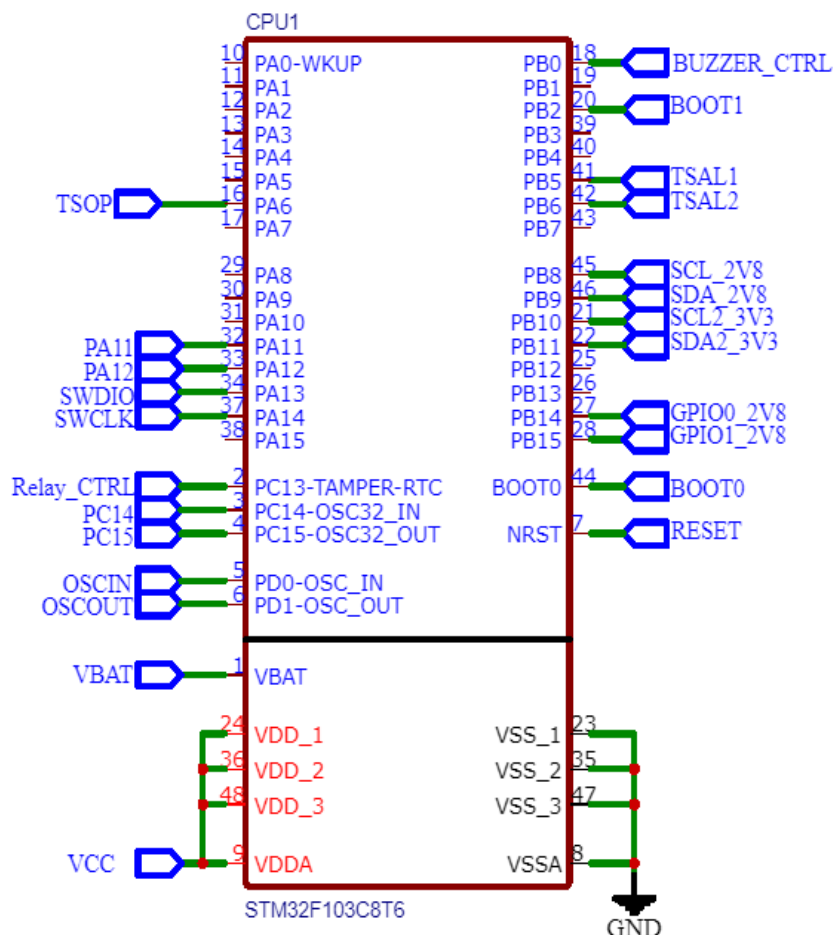


Рисунок 40. Процессор *STM32F103C8T6*

14) схема перемычки программирования процессора (рисунок 41). Для загрузки данных в микроконтроллер, программирования и отладки использовался интерфейс *SWD*.

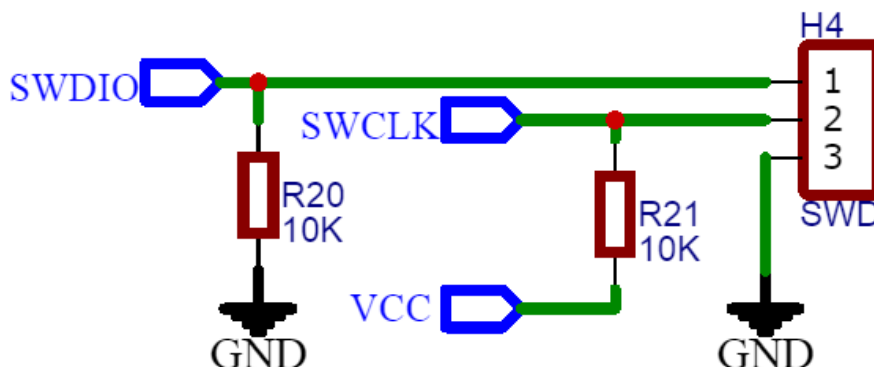


Рисунок 41. Перемычка программирования процессора

Данные модули в совокупности на плате (рисунок 42) образуют одно устройство для роевой роботизированной системы.

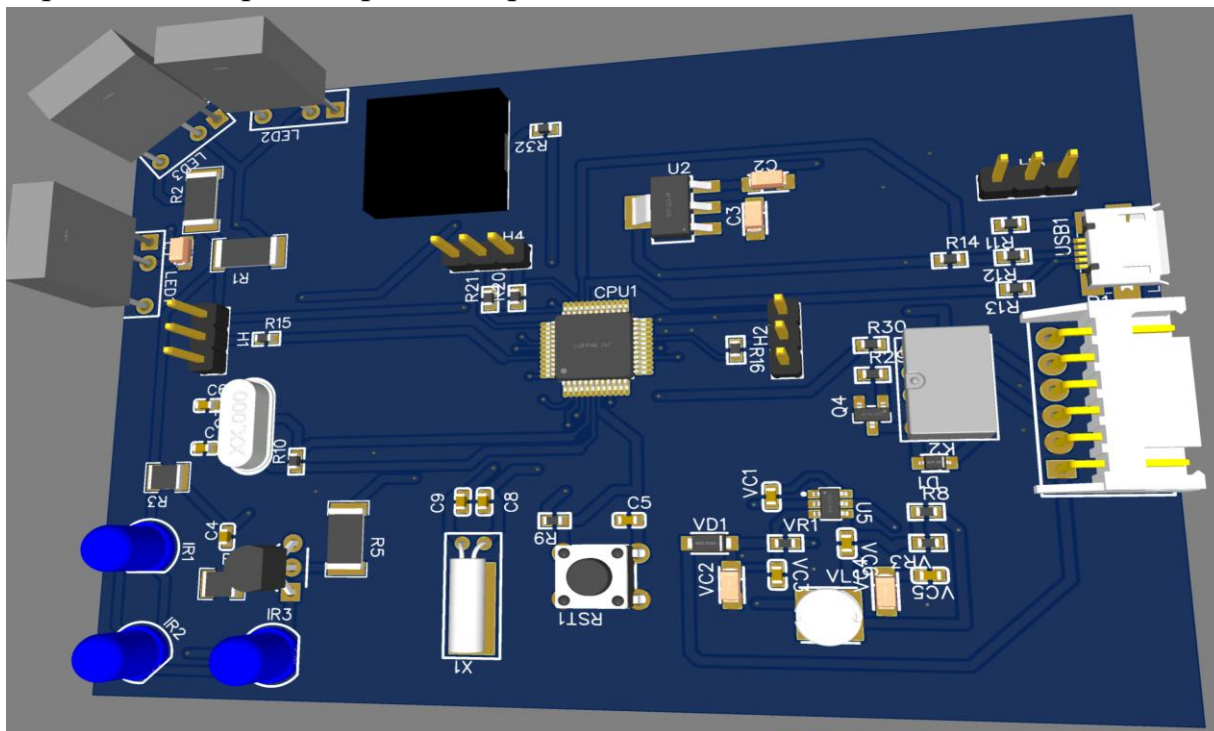


Рисунок 42. 3D модель платы роевого устройства

На рисунке 43 представлены слои и компоненты модели печатной платы.

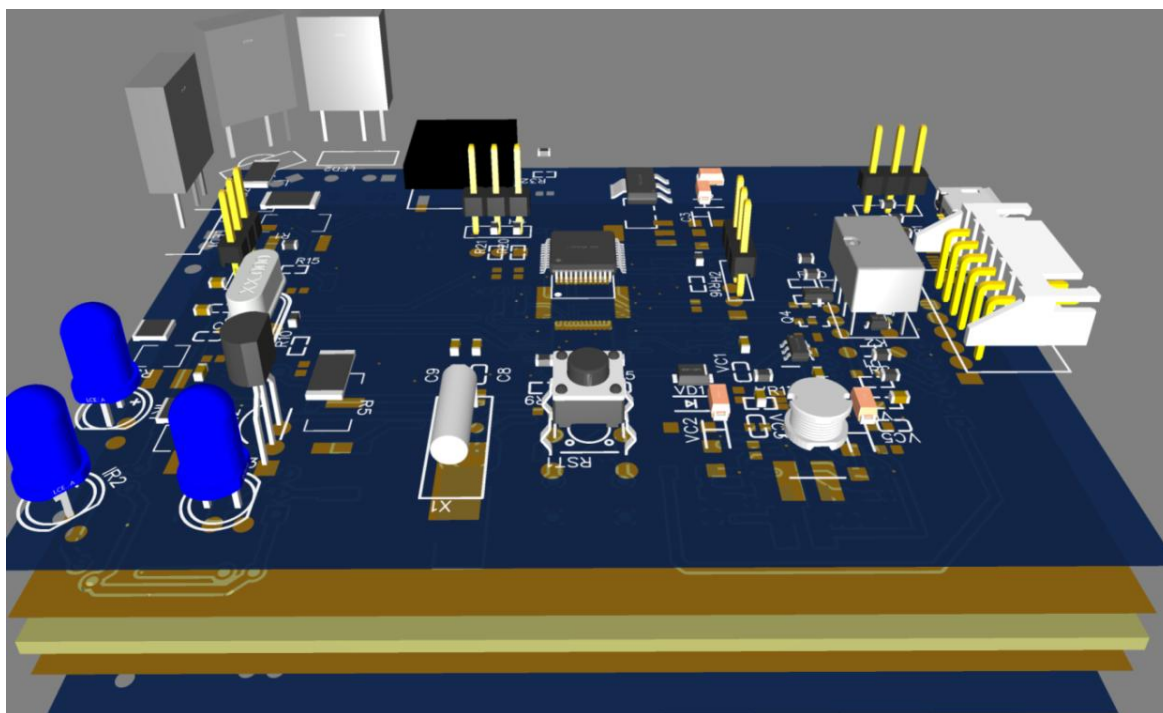


Рисунок 43. Слои и компоненты платы роевого устройства

На рисунке 44 приведена схема расположения компонентов и шелкография.

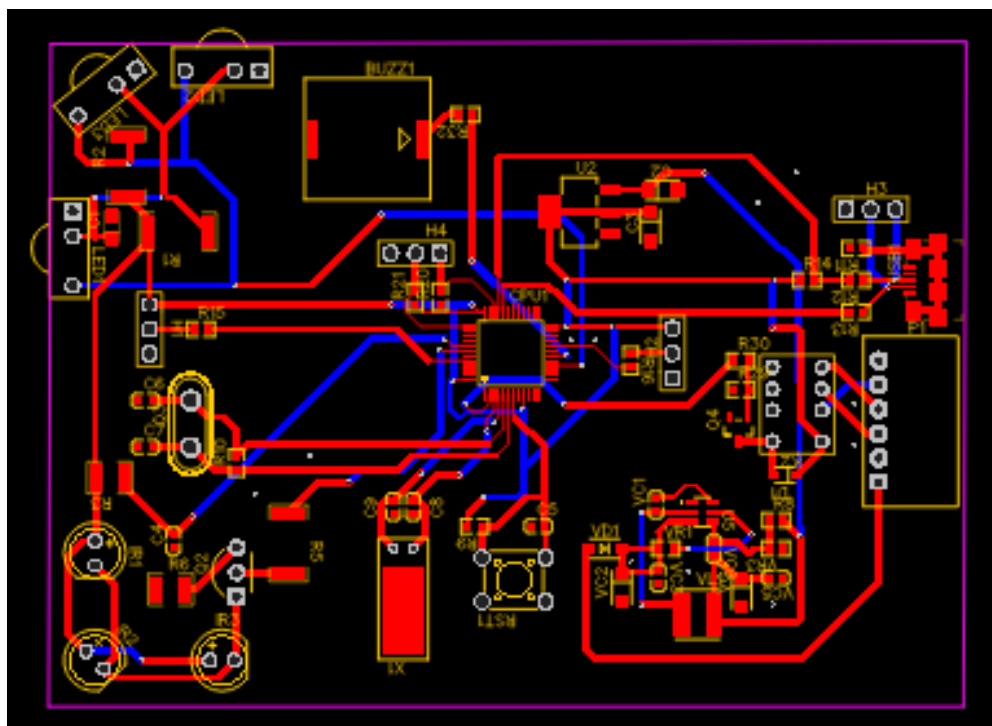
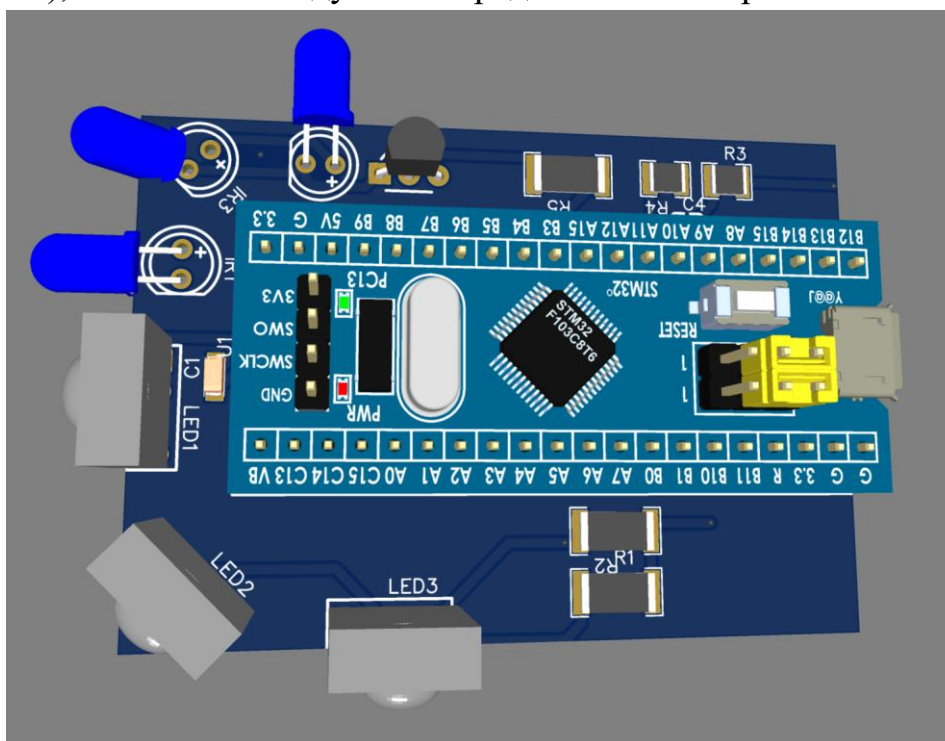


Рисунок 44. Схема расположения компонентов и шелкография

Для тестирования программной и аппаратной части роевое системы был спроектирован и реализован отладочный модуль на плате *STM32f103 blue pill* (рисунок 45), с базовыми модулями передачи *TSOP* и приема *TSAL*.



[illegible]

Рисунок 47. Контакты платы *STM32f103 Blue Pill*

Ядро микроконтроллера *STM32F103C8T6* поддерживает два режима работы: потоковый (*Thread*) и режим обработки (*Handler*). Процессор запускается в потоковом режиме при непрерываемом, фоновом выполнении инструкций и переключается в режим обработки исключительных ситуаций. Процессор может выполнять код программы в привилегированном или не-

привилегированном режиме. В привилегированном режиме, процессор имеет доступ ко всем наборам инструкций, а в непривилегированном режиме часть инструкций отключаются.

При работе в любом режиме, инструкции потока и обработки выполняются в привилегированном, таким образом ограничения на доступ к процессорным ресурсам отсутствуют.

Компоненты отладочного модуля роевого устройства

Для сборки модулей использующихся в устройстве, необходимы электронные компоненты. При выборе компонентов был проведен сравнительный анализ по стоимости закупки. Для отладочного модуля необходимы:

- 1) *TSAL6200* ик-диод, *TSOP31238* ик-приемник;
- 2) резисторы на 12 Ом, 10 Ом, 100 Ом, 10кОм;
- 3) конденсаторы на 4,7 мкФ, 0,1 мкФ;
- 4) *BC337-25* транзистор;
- 5) *STM32F103C8T6 Blue pill*.

Данные компоненты необходимы не только для отладочного модуля, но и для самого устройства роевой системы, однако для самостоятельного устройства неприемлемо использовать готовую плату *STM32f103 Blue Pill*.

Для закупки компонентов отладочного модуля были выбраны три магазина в ближайшей доступности и проведен оценочный анализ цены.

При подсчете средняя стоимость закупки отладочно модуля (не считая программатора, проводов, пинов, отладочной платы) составила 850 рублей. Сама же закупка для полной реализации модуля суммарно составила 2000 рублей.

Программная реализация устройства

Для реализации программной части роевой роботизированной системы в среде программирования *STM32CubeIDE*, необходимо выполнить несколько базовых этапов.

Первый этап предварительной настройки и конфигурации проекта. Для выполнения данного этапа необходимо создать новый проект в *STM32CubeIDE* и выбрать необходимый микроконтроллер (рисунок 48).

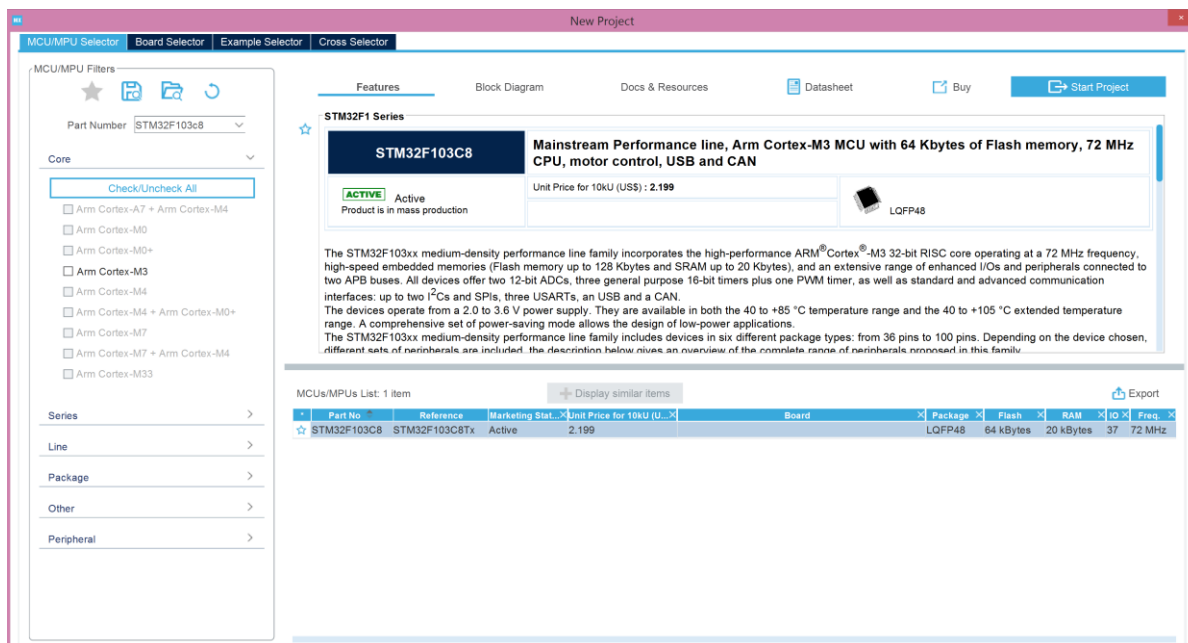


Рисунок 48. Выбор микроконтроллера в *STM32CubeIDE*

После подтверждения выбора представляется рабочая область настройки проекта (рисунок 49).

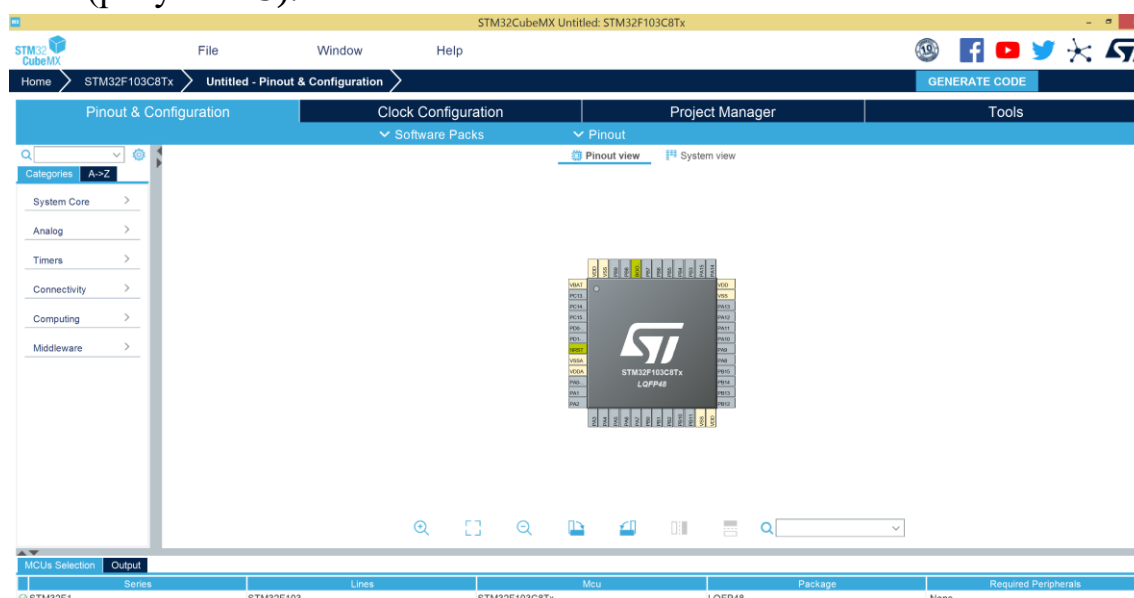


Рисунок 49. Рабочая область настройки проекта в *STM32CubeIDE*

После этого наступает второй этап, настройка управляющих сигналов. Данная настройка управляющих выходов процессора необходима для взаимодействия с модулями посредством цифрового сигнала, обработки полученного сигнала и синхронизации с кварцевыми резонаторами. Во время этого этапа программно задается функция ножек микропроцессора.

Для того чтобы перепрограммировать микроконтроллер в дальнейшем без перезагрузки и сброса внутренних настроек, необходимо заранее задать

системную конфигурацию, которая выделить две ножки для загрузки данных и тактового сигнала (рисунок 50).

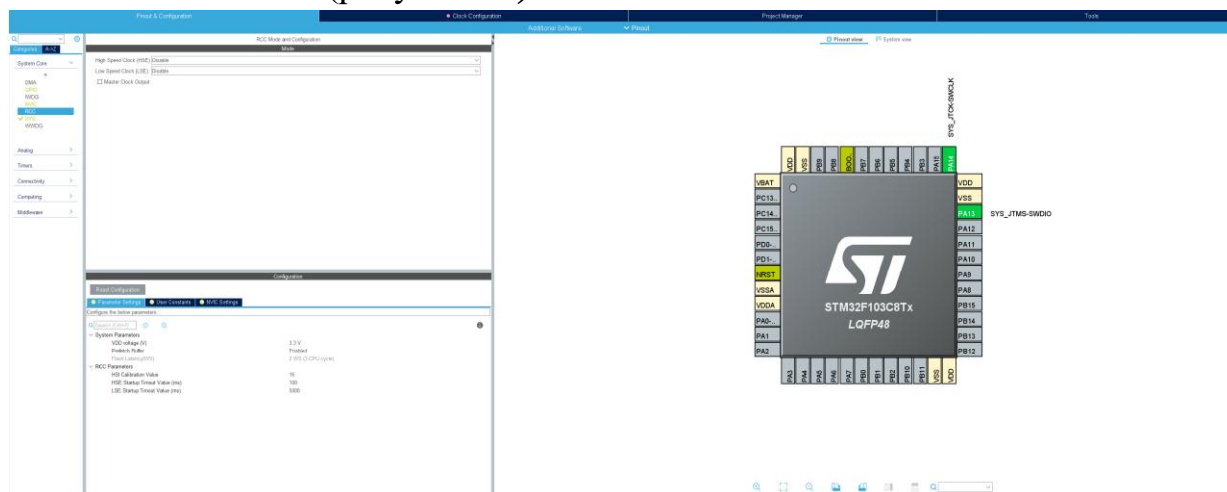


Рисунок 50. Настройка системной конфигурации ножек микроконтроллера в *STM32CubeIDE*

Для работы микроконтроллера необходимо настроить резонатор частот. В первую очередь назначить ножки микроконтроллера, которые будут отвечать за прием частотного сигнала (рисунок 51). Так необходима высокая стабильность частоты работы микроконтроллера, а также привязка ко времени, необходимо использовать внешние тактовые резонаторы, в частности резонатор на 32 768 Гц, так как при делении на 15-ти разрядном двоичном счетчике, данный резонатор дает отсчет в одну секунду времени. Данный резонатор может быть использован для формирования сигналов, растянутых по времени.

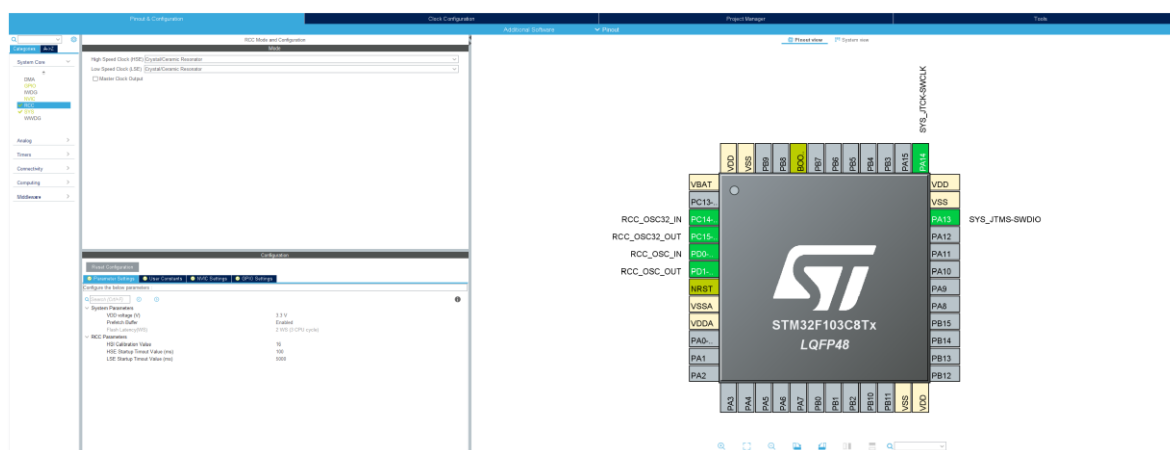


Рисунок 51. Настройка ножек микроконтроллера для резонатора частот в *STM32CubeIDE*

После назначения системных и частотных ножек, настраивается конфигурация резонатора, в *STM32CubeIDE* есть внутренний калькулятор, который помогает в настройке и выборе частот (рисунок 52).

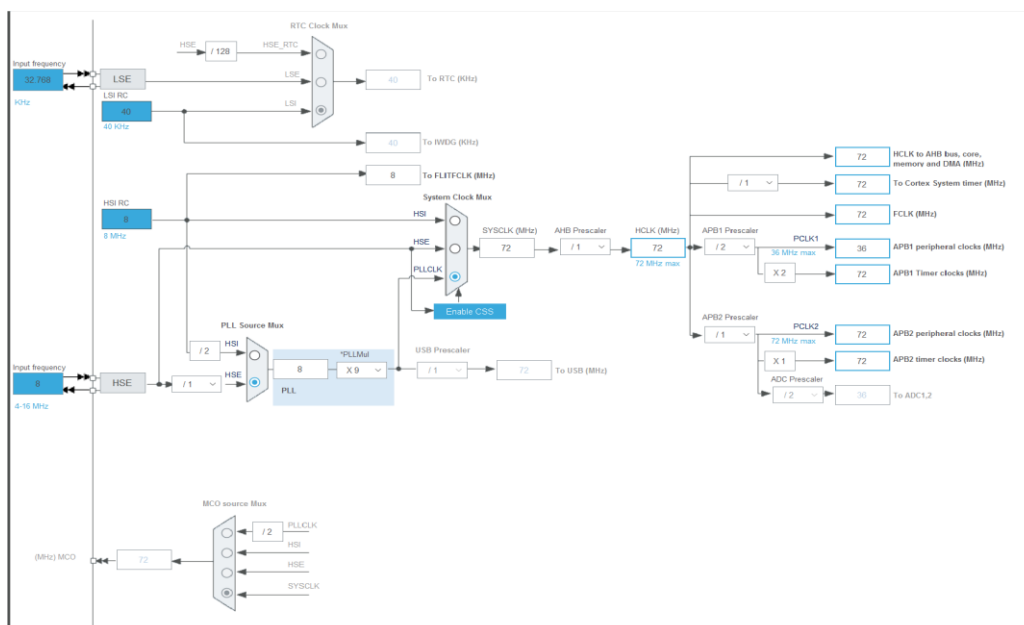


Рисунок 52. Настройка частот тактирования в *STM32CubeIDE*

После этого базовая настройка микроконтроллера считается завершенной и можно приступать к программированию микроконтроллера. На этапе программирования необходимо задать ножки микропроцессора которые будут отвечать за прием или передачу, цифровых или аналоговых сигналов управления (рисунок 53).

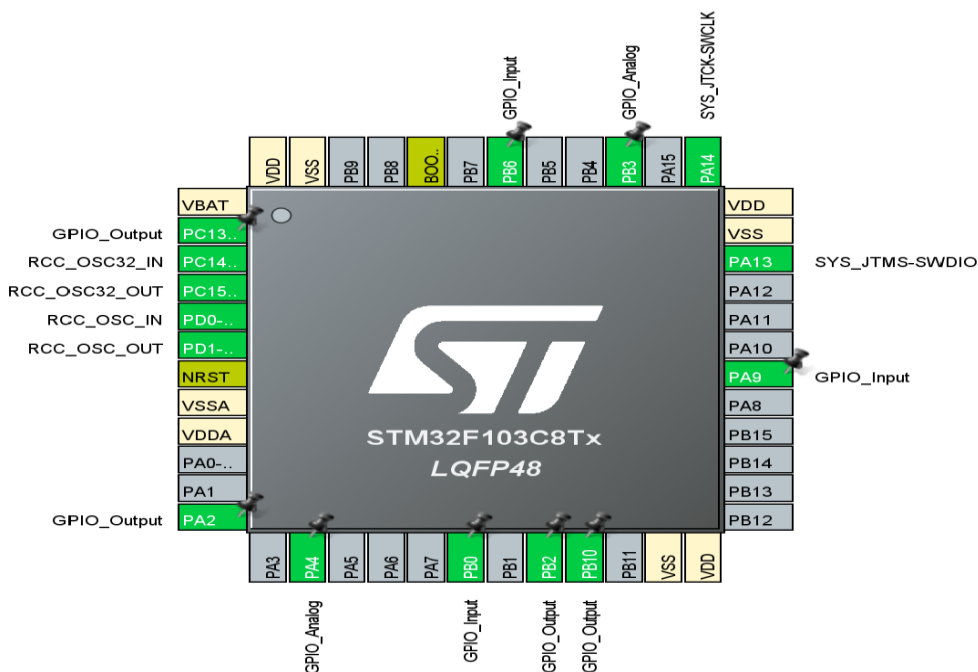


Рисунок 53. Настройка ножек приема или передачи сигналов
в *STM32CubeIDE*

После того как необходимые ножки выбраны и настроены, необходимо подготовить интерфейс для загрузки программы в микропроцесор. *STM32CubeIDE* имеет функцию подбора кода, которая не требует написания

вручную задач под ножки выделенные на резонатор и системную конфигурацию, а так же других зарезервированных ножек.

Для загрузки написанной программы и настроек в микроконтроллер необходим программатор (рисунок 54). Компания *STM* имеет специально разработанный программатор *ST-LINK V2* с внутренним отладчиком и несколькими интерфейсами для подключения микроконтроллеров, но средняя цена таких устройств равна 2000 рублей. Поэтому в качестве программатора был выбран китайский аналог *ST-LINK V2*.



Рисунок 54. Программатор *ST-LINK V2* для *STM8* и *STM32*

Перед началом загрузки программы необходимо очистить память микроконтроллера (не удаляя защищенные адреса), для этого необходимо воспользоваться программой *STM32CubeProgrammer* (рисунок 55) или более старой версией *STM32 ST-LINK Utility*. Это необходимо, в частности, для предотвращения появления ошибок и правильной работы микроконтроллера.

Некоторые микроконтроллеры, в частности установленные на платы, уже имеют пробную программу для проверки их работоспособности, к сожалению некоторые производители не выполняют системную конфигурацию, во время проверки и это приводит к блокировке микроконтроллера. В этом случае необходимо провести очистку памяти микроконтроллера, подключаясь к нему в режиме перезагрузки.

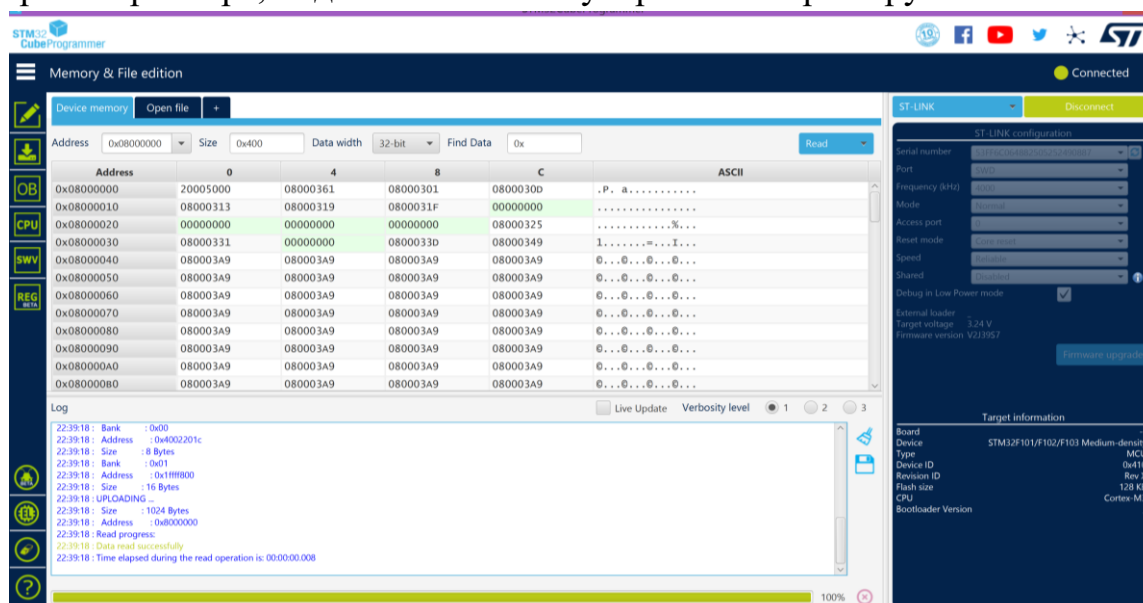


Рисунок 55. Основное окно *STM32CubeProgrammer*

Программа *STM32 ST-LINK Utility* (рисунок 56) имеет меньший функционал, но способна обойти блокировку программаторов, при обновлениях прошивки.

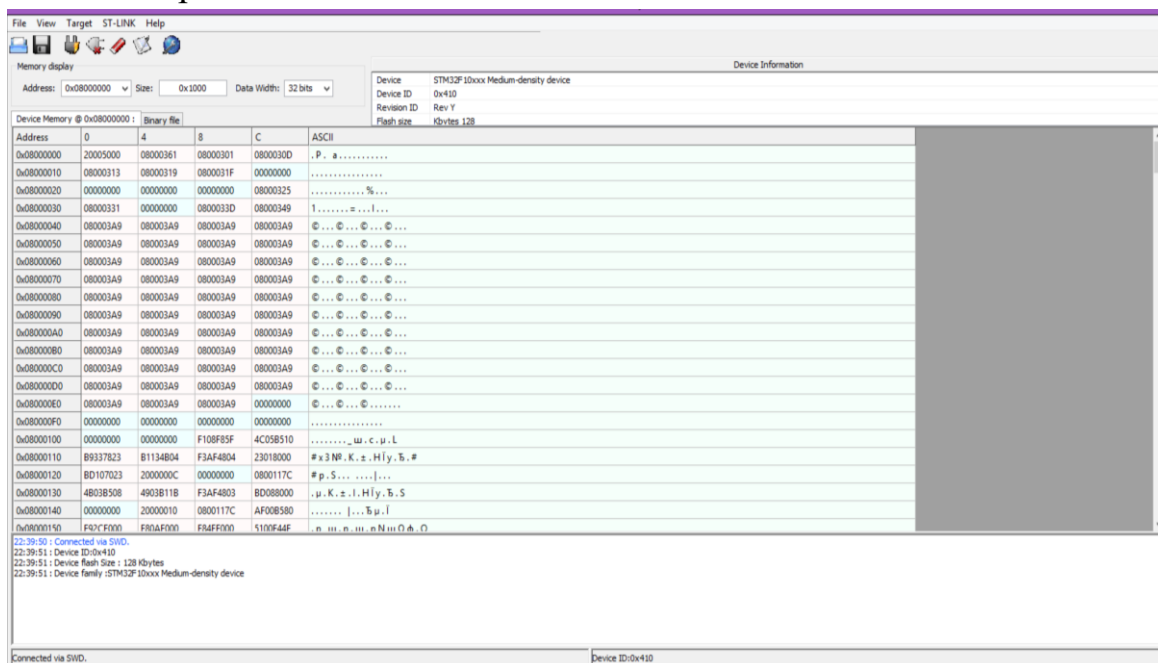


Рисунок 56. Основное окно *STM32 ST-LINK Utility*

При передачи информации используется процесс управления мощностью свечения диода методом затухающего, пульсирующего включения и выключения.

На этапе генерации ключей (рисунок 57) необходим доверенный центр который будет публиковать произведение двух простых чисел и два параметра безопасности k и t , где k – длина вектора, а t – количество раундов проверки. Агент идентифицирует себя только открытым ключом, не раскрывая секретный ключ.

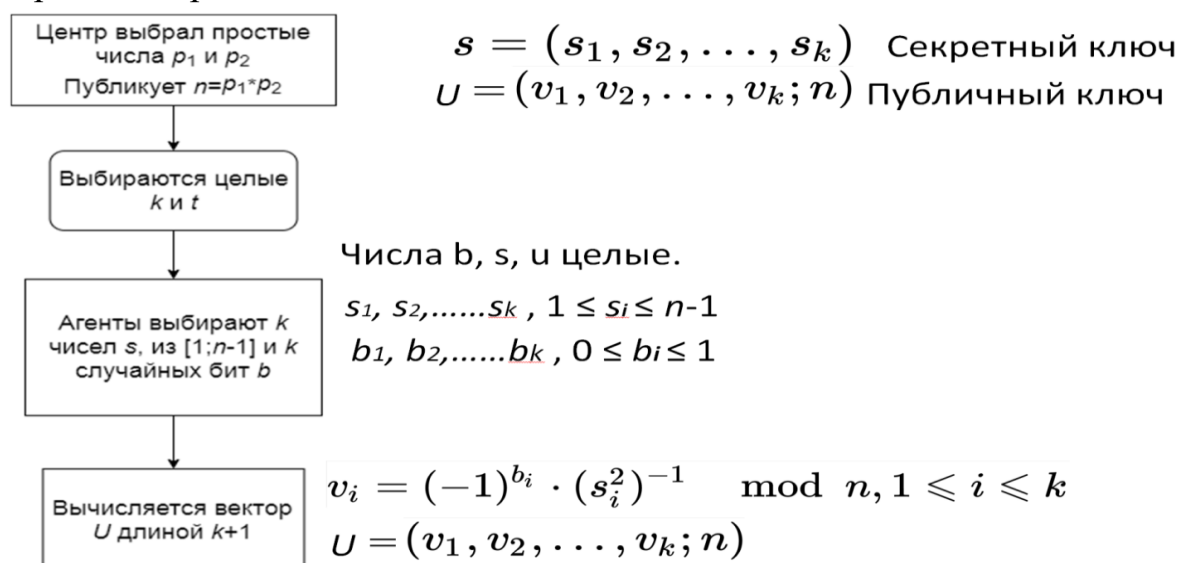


Рисунок 57. Этап генерации ключей

На этапе проверки знания секретом (рисунок 58), претендент должен доказать, что обладает секретом, при этом, не разглашая его.

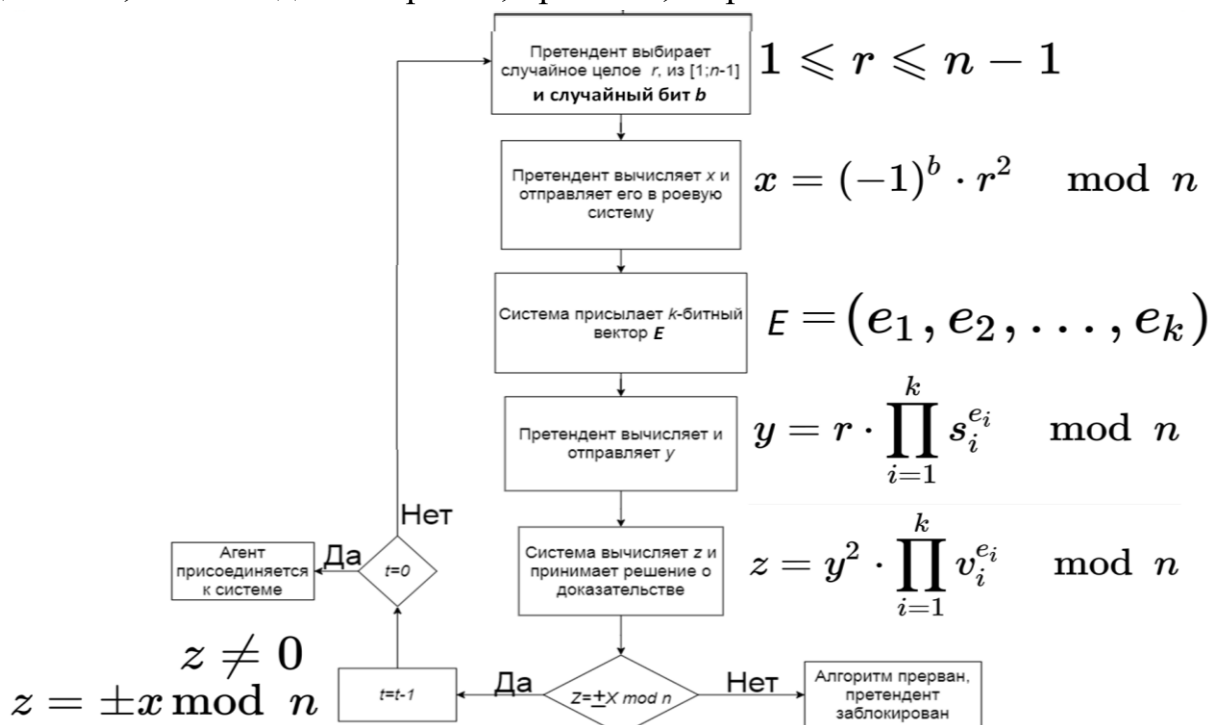


Рисунок 58. Этап проверки знания секретом

Безопасность данного алгоритма исходит из предположения, что разложение произведения простых чисел на множители вычислительно невозможная задача. Таким образом можно получить ложноположительный ответ, если угадать значения вектора E_i , вероятность такой атаки злоумышленника на протокол составляет 2^{-kt} .

Для проверки аутентификации используется алгоритм *Hmac-SHA1*. Код аутентификации сообщений (рисунок 59), использующий хэш-функции (*Hmac*). позволяет гарантировать что передаваемые данные не были изменены злоумышленником.

Для повышения похозащищенности канала приема-передачи между агентами, а так же дополнительной мерой защиты от ложных сообщений, используется алгоритм циклического избыточного кода(*CRC*) (рисунок 60).

Данный алгоритм базируется на математических свойствах деления с остатком многочленов над полем $GF(2)$. Циклический избыточный код является остатком от деления многочлена соответствующего входным данным, на фиксированный порождающий многочлен.

Код передачи данных посредством формирования полезного сигнала, широтно-импульсной модуляцией, частоты 38 кГц. Модуляция происходит через внешний тактовый резонатор на 8 МГц. Код формирования пакетов, установления и поддержания сигнала, представлен в приложении Д.

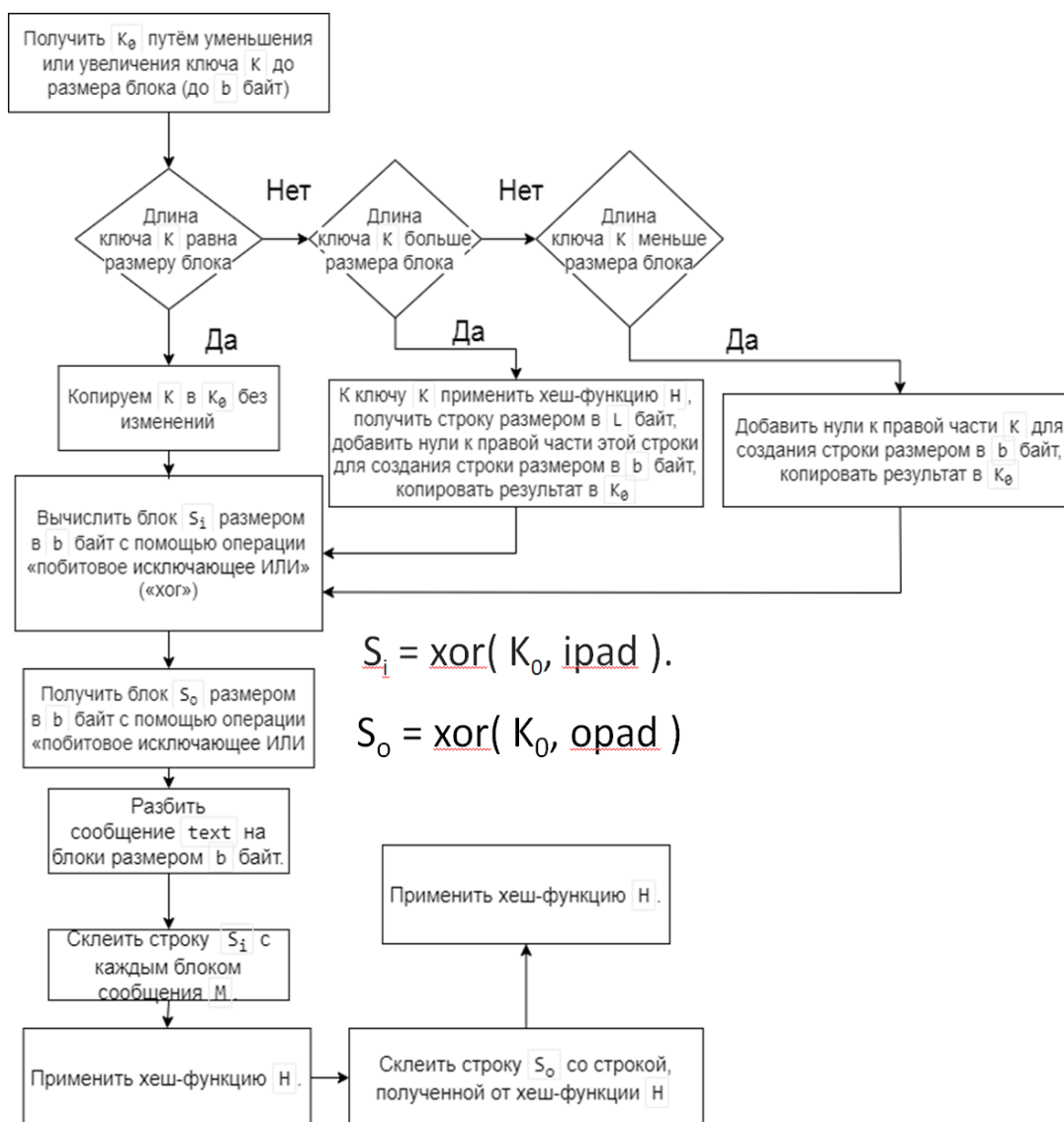


Рисунок 59. Алгоритм Hmac

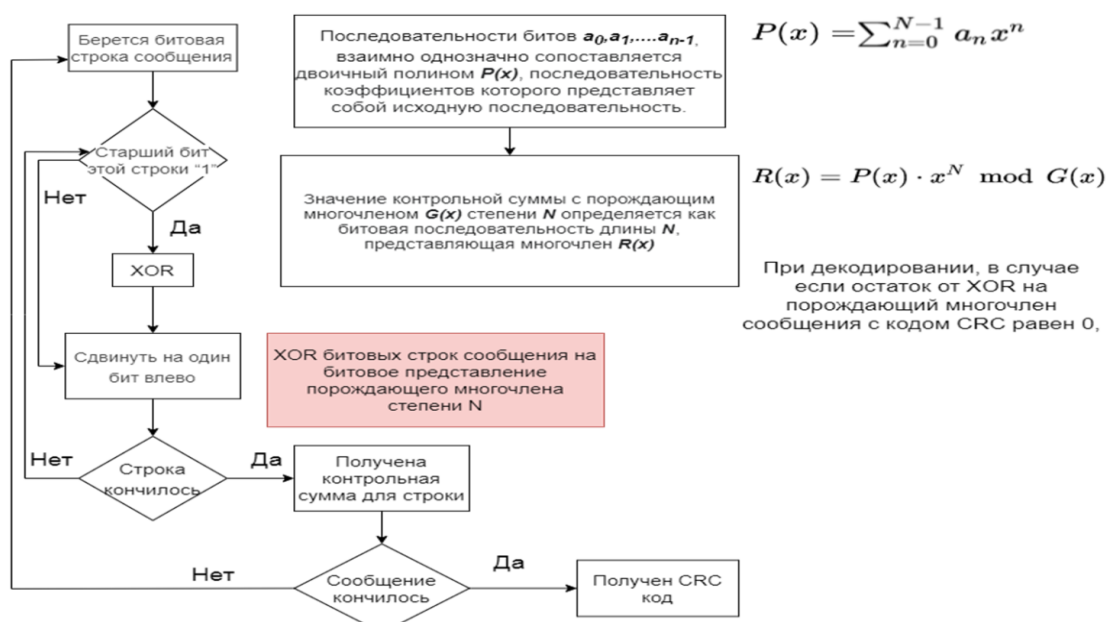


Рисунок 60. Алгоритм циклического избыточного кода

Заключение

В ИТСГИС разработана роевая роботизированная система обеспечения безопасности, на основе защищенной и надежной беспроводной передачи данных. Проведен анализ предметной области, роевых роботизированных систем. Проведен анализ существующих способов обеспечения надежной безопасной, беспроводной передачи данных. Выбрано семейство микроконтроллеров, способ передачи данных и способ взаимодействия устройств. Определена структура роевой системы, состоящей из агентов. Выявлены основные функции, реализуемые агентами роевой системы. Разработан протокол взаимодействия агентов роевой системы, обеспечивающий безопасное, надежное, беспроводное соединение. Спроектирован, реализован и запрограммирован отладочный модуль роевой роботизированной системы. Разработано программное обеспечение агента роевой системы.

Также определена структура безопасной роевой системы, использующей протоколы шифрования, аутентификации и идентификации агентов при беспроводной передачи данных.

В качестве вычислительного устройства выбрано семейство микроконтроллеров *STM32*, проектирование отладочного модуля и агентов выполнялось в программе *EasyEDA*, программирование выполнялось с помощью программ *STM32CubeProgrammer*, *STM32 ST-LINK Utility* и *STM32CubeIDA*.

Список использованных источников

1. Емельянов, Д.А. Фильтрация сетевого контента в образовательных учреждениях / Д.А. Емельянов // М. : Педагогическое образование в России – 2018. – URL: <https://cyberleninka.ru/article/n/filtratsiya-setevogo-kontenta-v-obrazovatelnyh-uchrezhdeniyah/viewer>
2. Чемодуров, А.С. Обзор средств фильтрации трафика в корпоративной сети / А.С. Чемодуров, А.Ю. Карпутина // М. : Концепт – 2015. № 2. – С. 1-6.
3. Интеллектуальная транспортная геоинформационная система ИТСГИС. Ядро / Т.И. Михеева, С.В. Михеев, О.К. Головнин, А.В. Сидоров, Е.А. Савинов. – Самара : Интелтранс, 2016. – Т.1. – 171 с. – ISBN 978-5-9906857-4-1.
4. Михеев, А.В. Исследование методов сбора статистических данных о трафике в ip-сетях передачи данных / А.В. Михеев // электронный научный архив Уральского федерального университета им. Б. Н. Ельцина. – 2016. – URL: <http://elar.urfu.ru/bitstream/10995/36237/1/ittsm-2016-16.pdf>.
5. Головнин, О.К. Паттерны поддержки принятия решений по дислокации технических средств организации дорожного движения /

- Т.И. Михеева, О.К. Головнин // Перспективные информационные технологии (ПИТ 2013) : труды Международной научно-технической конференции. – Самара : Издательство Самарского научного центра РАН, 2013. – С. 267–272.
6. Олифер, Н.А. Средства анализа и оптимизации локальных сетей / Н.А. Олифер, В.Г. Олифер // IT-портал. – 1998. – Режим доступа: http://citforum.ru/nets/optimize/locnop_07.shtml.
 7. Михеев, С.В. Интеллектуальная транспортная геоинформационная система ITSGIS / Т.И. Михеева, С.В. Михеев, О.К. Головнин // Современные проблемы безопасности жизнедеятельности: интеллектуальные транспортные системы : материалы IV Международной научно-практической конференции (Казань, 25–26 февраля 2016 г.). – Казань : ГБУ «Научный центр безопасности жизнедеятельности», 2016. – С. 362–368. – ISBN 978-5-85247-837-5.
 8. Кадыров, М.М. Разработка программного модуля для фильтрации сетевого трафика / М.М. Кадыров // М. : Молодой ученый. – 2017. – № 4 (138). – С. 15-18. – URL: <https://moluch.ru/archive/138/38793/>.
 9. Головнин, О.К. Применение паттерна «Рабочий поток» в геоинформационной составляющей интеллектуальной транспортной системы / О.К. Головнин // Математика. Компьютер. Образование : тезисы докладов двадцатой международной конференции. – Пущино, 2013. – Режим доступа : <http://www.mce.su/rus/archive/abstracts/mce20/sect101361/doc174037>.
 10. Mikheeva, T. Recognition of Urban Transport Infrastructure Objects Via Hyperspectral Images / O. Saprykin, A. Fedoseev, T. Mikheeva // In Proceedings of the 2nd International Conference on Vehicle Technology and Intelligent Transport Systems (VENITS-2016). Rome, Italy: SCITEPRESS, 2016. – P. 203-208.
 11. Михеева, Т.И. Метод синтеза системы зонального сетцентрического управления транспортными процессами / Т.И. Михеева, С.В. Михеев, О.К. Головнин // Известия Самарского научного центра Российской академии наук. – 2016. – Т. 18, № 4 (4). – С. 799-807.
 12. Михеев, С.В. Диагностика состояния транспортной инфраструктуры с использованием нейронных сетей / С.В. Михеев, А.В. Сидоров, А.А. Осьмушин // Современные проблемы науки и образования. – 2013. – №6. – С. 215-216.
 13. Михеева, Т.И. Системный анализ объектов транспортной инфраструктуры в геоинформационной среде / Т.И. Михеева // Программные продукты и системы. – 2018. – № 1. – С. 12-18.

K.A. Molodyko, T.I. Mikheeva

**SWARM ROBOTIC SUPPORT SYSTEM SECURITY ITSGIS BASED ON
WIRELESS DATA NETWORKS**

*Samara, Samara University named after academician S.P. Korolev
IntelTrans*

This article discusses the functional specification of the core of the client part of the system. The article describes the implementation of the software interface (API) of the intelligent transport geoinformation system "ITSGIS", designed to download information about objects of various types on an electronic map. Most of the typical tasks performed by a client application are allocated to the core of the client application. The necessity of implementing this software interface is justified. It is indicated that in order to ensure the interaction of ITSGIS components with newly developed modules, it is necessary to eliminate or reduce dependence on specific software development tools. In the process of developing system components, the tasks of storing application settings and selecting the log placement options used (in a file, in the system event log, in a database on the server) were solved.

Keywords: software interface, decision support system, transport infrastructure, intelligent transport geoinformation system "ITSGIS", implementation, modular architecture, functional specification of the core of the client part of the computer network, integration.

СПИСОК АВТОРОВ СБОРНИКА

Михеева Татьяна Ивановна, доктор технических наук, профессор кафедры информационных систем и технологий Самарского университета (национального исследовательского университета), генеральный директор группы компаний «ИнтелТранС» (ITSGIS).

Михеев Сергей Владиславович, кандидат технических наук, доцент кафедры организации и управления перевозками на транспорте Самарского университета (национального исследовательского университета).

Смолев Александр Михайлович, диссертация Самарского университета им. академика С.П. Королева (национального исследовательского университета), программист отдела программирования научно-производственного центра «Интеллектуальные транспортные системы».

Клепиков Никита Михайлович, диссертация Самарского университета им. академика С.П. Королева (национального исследовательского университета), кафедра технической кибернетики института информатики, математики и электроники.

Молодыко Константин Алексеевич, диссертация Самарского университета им. академика С.П. Королева (национального исследовательского университета), кафедра технической кибернетики института информатики, математики и электроники.

СОДЕРЖАНИЕ

<i>Смолев А.М., Михеева Т.И., Клепиков Н.М. АДРЕСНОЕ ПОЗИЦИОНИРОВАНИЕ И ЕГО РЕАЛИЗАЦИЯ В ГЕОИНФОРМАЦИОННОЙ СИСТЕМЕ.....</i>	<i>4</i>
<i>Михеева Т.И., Клепиков Н.М. ИНСТРУМЕНТЫ ОБРАБОТКИ, АНАЛИЗА И РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ</i>	<i>10</i>
<i>Молодыко К.А., Михеева Т.И. РОЕВАЯ РОБОТИЗИРОВАННАЯ СИСТЕМА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ИТСГИС НА ОСНОВЕ БЕСПРОВОДНОЙ СЕТИ ПЕРЕДАЧИ ДАННЫХ.....</i>	<i>14</i>

Сетевое издание**IT & TRANSPORT / ИТ & ТРАНСПОРТ****Редакционная коллегия****Главный редактор:**

Михеева Татьяна Ивановна – доктор технических наук, профессор, заслуженный деятель науки и образования РФ», (Самара)

Заместители главного редактора:

Прохоров Сергей Антонович – д.т.н., заместитель генерального директора ООО «НПЦ «ИТС», заслуженный работник высшей школы РФ, академик Академии навигации и управления движением, доктор технических наук, профессор, (Самара),

Филиппова Анна Сергеевна – д.т.н., «Башкирский государственный педагогический университет им. М. Акмуллы», (Уфа)

Редакционный совет:

Бурдин Антон Владимирович – д.т.н., «Санкт-Петербургский Государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича», (Санкт-Петербург),

Гераськин Михаил Иванович – д.э.н., Самарский университет, (Самара),

Жанказиев Султан Владимирович – д.т.н., интеллектуальные транспортные системы МАДИ, (Москва),

Заболотнов Юрий Михайлович – д.т.н., Самарский университет, (Самара),

Зырянов Владимир Васильевич – д.т.н., РГСУ, (Ростов-на-Дону),

Ивашенко Антон Владимирович – д.т.н., Передовая медицинская инженерная школа Самарского государственного медицинского университета Министерства здравоохранения Российской Федерации, (Самара),

Минниханов Рифкат Нургалиевич – д.т.н., ГБУ «Безопасность дорожного движения», (Казань),

Хайтбаев Валерий Абдурахманович – д.э.н., Самарский университет, (Самара), Самарский университет, (Самара),

Зеленко Лариса Сергеевна – к.т.н., Самарский университет, (Самара),

Золотовицкий Аркадий Владимирович – к.т.н., Самарский университет, (Самара),

Михеев Сергей Владиславович – к.т.н., ООО «Научно-производственный центр «Интеллектуальные транспортные системы», (Самара),

Сапрыкин Олег Николаевич – к.т.н., Самарский университет, (Самара),

Сапрыкина Ольга Валерьевна – к.т.н., Самарский университет, (Самара),

Федосеев Александр Андреевич – к.т.н., ОАО «РКУ «Прогресс», (Самара),

Шопин Андрей Геннадьевич – к.т.н., ООО «СМС-информационные технологии», (Самара).

Издательство ООО «НПЦ «ИТС»
443099, Самарская обл., г.о. Самара, г. Самара,
вн. р-н Самарский, ул. Куйбышева, д. 71, оф. 2

Подписано к выпуску 15.11.2024.

Распространяется бесплатно

Учредитель

ООО «НПЦ «ИТС»

Адрес редакции журнала «ИТ & Транспорт»
443125, г. Самара, пр. Кирова, 328, 67