Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

К ЗАЩИТЕ ДОПУСТИТЬ

Зав. каф. ЭВМ

\_\_\_\_\_\_\_\_\_\_ Б.В. Никульшин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

на тему

ПРОГРАММНЫЙ КОМПЛЕКС «УМНЫЙ ДОМ» С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ ZIGBEE

БГУИР ДП 1–40 02 01 01 006 ПЗ

Студент Я.В. Антонов

Руководитель Ю.А. Луцик

Консультанты:

от кафедры ЭВМ Ю.А. Луцик

по экономической части В.Г. Горовой

Нормоконтролер А.И Стракович

Рецензент

МИНСК 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 7](#_Toc167173818)

[1 ОБЗОР ЛИТЕРАТУРЫ 8](#_Toc167173819)

[1.1 Обзор существующих аналогов 8](#_Toc167173820)

[1.2 MQTT-протокол 15](#_Toc167173821)

[1.3 Протоколы передачи данных 17](#_Toc167173822)

[2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ 20](#_Toc167173823)

[2.1 Блок пользовательского интерфейса 20](#_Toc167173824)

[2.2 Блок соединения с брокером MQTT 22](#_Toc167173825)

[2.3 Блок приема-передачи данных MQTT брокеру 22](#_Toc167173826)

[2.4 Блок обработки данных 23](#_Toc167173827)

[2.5 Блок базы данных 24](#_Toc167173828)

[2.6 Блок тестирования 24](#_Toc167173829)

[2.7 Блок координатора 25](#_Toc167173830)

[2.8 Блок конечных устройств 25](#_Toc167173831)

[2.9 Блок MQTT брокера 26](#_Toc167173832)

[3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ 27](#_Toc167173833)

[3.1 Аппаратная часть 27](#_Toc167173834)

[3.2 Программная часть 28](#_Toc167173837)

[4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 44](#_Toc167173867)

[4.1 Android Studio 44](#_Toc167173868)

[4.2 Добавление системы 45](#_Toc167173869)

[4.3 Вывод списка систем 47](#_Toc167173870)

[4.4 Вывод списка устройств 49](#_Toc167173871)

[4.5 Добавление нового устройства 50](#_Toc167173872)

[4.6 Подключение к MQTT брокеру 53](#_Toc167173873)

[4.7 Подписка и отписка на темы (topic) 54](#_Toc167173874)

[4.7 Получение данных от MQTT брокера 55](#_Toc167173875)

[4.8 Обработка полученных данных 56](#_Toc167173876)

[4.9 Отправка данных в MQTT брокер 57](#_Toc167173877)

[4.10 Взаимодействие с базой данных 57](#_Toc167173878)

[5 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ 61](#_Toc167173879)

[5.1 Модульное тестирование 61](#_Toc167173880)

[5.2 Ручное тестирование 67](#_Toc167173881)

[6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 69](#_Toc167173882)

[6.1 Руководство по использованию устройства координатор. 69](#_Toc167173883)

[6.2 Системные требования мобильного устройства 72](#_Toc167173884)

[6.3 Руководство по использованию мобильного приложения 73](#_Toc167173885)

[6.4 Добавление системы 74](#_Toc167173886)

[6.5 Изменение и удаление системы 75](#_Toc167173887)

[6.6 Открытие списка устройств и добавление устройства 76](#_Toc167173888)

[6.6 Изменение и удаление устройства 77](#_Toc167173889)

[6.6 Руководство программиста 77](#_Toc167173890)

[6.7 Установка IDE Android Studio 78](#_Toc167173891)

[6.8 Java Development Kit 79](#_Toc167173892)

[6.12 MQTT сервер 80](#_Toc167173893)

[6.13 Установка ПО ESP32 81](#_Toc167173894)

[6.14 Установка ПО E72 82](#_Toc167173895)

[6.15 Разработка и установка ПО конечного устройства 83](#_Toc167173896)

[7 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ И   
 РЕАЛИЗАЦИИ НА РЫНКЕ ПРОГРАММНОГО КОМПЛЕКСА   
 «УМНЫЙ ДОМ» С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ ZIGBEE 86](#_Toc167173897)

[7.1 Описание функций, назначения и потенциальных пользователей   
 программного комплекса 86](#_Toc167173898)

[7.2 Расчет затрат на разработку программного комплекса 86](#_Toc167173899)

[7.3 Расчет экономического эффекта от реализации программного   
 комплекса на рынке 88](#_Toc167173904)

[7.4 Расчет показателей экономической эффективности разработки и   
 реализации программного средства на рынке 90](#_Toc167173905)

[7.5 Вывод об экономической целесообразности реализации проектного   
 решения 90](#_Toc167173906)

[ЗАКЛЮЧЕНИЕ 91](#_Toc167173907)

[СПИСОКИСПОЛЬЗОВАННЫХИСТОЧНИКОВ 92](#_Toc167173908)

[ПРИЛОЖЕНИЕ А 94](#_Toc167173909)

[ПРИЛОЖЕНИЕ Б 126](#_Toc167173910)

[ПРИЛОЖЕНИЕ В 127](#_Toc167173911)

# ВВЕДЕНИЕ

С появлением передовых технологий, жизнь человека становится более комфортной и продуктивной. Развитие технологий в области «умного дома» привело к тому, что в доме теперь можно автоматизировать и контролировать множество процессов, что делает жизнь более удобной, безопасной и эффективной. От управления освещением и климатом до мониторинга безопасности и энергопотребления – все это становится доступным благодаря «умным» технологиям.

Сегодня множество компаний предлагают различные решения для «умного дома». Некоторые из них специализируются на «умных замках» и системах безопасности, другие - на «умных термостатах» и системах управления климатом, а третьи - на интегрированных системах «умного освещения». Более того, с появлением стандартов связи, таких как Zigbee и Z-Wave, устройства разных производителей могут взаимодействовать между собой, что упрощает интеграцию различных компонентов системы «умного дома».

Существуют как готовые решения, представляющие собой набор устройств с уже встроенным программным обеспечением для управления ими, так и DIY-проекты, где пользователи могут самостоятельно прошивать устройства и создавать собственные решения под свои потребности.

Однако, несмотря на все преимущества «умного дома», вопрос цены остается значимым. Многие интегрированные системы все еще остаются дорогостоящими, что может отпугнуть потенциальных пользователей. Поэтому разработка доступных по цене устройств является важным направлением развития этой отрасли.

Цель дипломного проекта заключается не только в прошивке устройств и разработке приложения для управления ими, но и в создании гибкой и доступной системы «умного дома», которая бы удовлетворяла потребности разнообразных пользователей. Это включает в себя не только технические аспекты, но и удобство использования, безопасность и экономическую целесообразность.

Для достижения поставленной цели дипломного проекта важно разбить её на конкретные задачи:

– прошивка и настройка устройств сети;

– проектирование архитектуры системы «умный дом»;

– разработка программного обеспечения;

– тестирование и отладка приложения.

Каждая из этих задач важна для создания успешного проекта по разработке гибкой и доступной системы «умного дома», удовлетворяющей потребности разнообразных пользователей.

# 1 ОБЗОР ЛИТЕРАТУРЫ

## 1.1 Обзор существующих аналогов

1.1.1 Система «Хайт Про»

«Хайт Про» [1] (рисунок 1.1) – это один из аналогов систем «умного дома», который следует изучить для понимания того, какие функциональности уже реализованы на рынке и какие подходы используются в этой области.



Рисунок 1.1 – Система «Хайт Про»

Вышеназванная система представляет собой компактное устройство, которое без труда вставляется в стандартную розетку 220 В. Оно работает на специальной радиочастоте 868 МГц и функционирует без подключения к интернету. Компактный корпус этого устройства укомплектован кнопкой управления, двумя индикаторными светодиодами и двумя разъемами: RJ-45 и USB. Программирование и настройка системы, а также добавление новых устройств осуществляются при помощи приложения HiTE PRO.

Среди функциональных возможностей этого устройства выделяются: способность управлять освещением и приводами, обеспечивая открытие и закрытие штор и жалюзи. Кроме того, система обладает разнообразными датчиками, которые могут контролировать движение, температуру, влажность, протечки воды, а также открытие дверей и окон. Интегрированные сценарии позволяют создавать удобные и интуитивно понятные автоматизированные процессы.

Эта система «умного дома» легко интегрируется в пять популярных экосистем: Apple HomeKit, Google Home, HiTE PRO, Tuya Smart и «Умный дом Яндекса». Кроме того, комплексом можно управлять голосом при помощи таких голосовых ассистентов, как Google Assistant, «Яндекс.Алиса», Siri и «Маруся».

В комплект этого устройства входит все необходимое для начала использования: основа «умного устройства», реле для управления освещением, беспроводной выключатель, а также датчики открытия двери и протечки воды.

Несмотря на многочисленные функции и преимущества, пользователи отмечают некоторые недостатки:

– иногда в системе возникают сбои, и связь между блоком управления и реле теряется;

– отсутствие функций обнаружения пожара;

– обновления программного обеспечения выпускаются не так часто, как хотелось бы пользователям.

Помимо перечисленных функциональных возможностей и недостатков, важно отметить, что компания предоставляет эффективную техническую поддержку для пользователей устройства. Техническая поддержка оперативно отвечает на запросы пользователей и готова помочь с решением любых возникающих проблем или вопросов.

1.1.2 «Умный дом» от «Белтелеком»

«Умный дом» от компании «Белтелеком» [2] представляет собой простое и экономичное решение, которое включает в себя все основные функции для обеспечения комфорта и ресурсосбережения в помещениях. Эта система предоставляет абонентам информацию о состоянии различных объектов в помещениях и позволяет им управлять этими объектами в соответствии с их потребностями.

Пользователи имеют возможность настраивать алгоритмы работы системы под свои индивидуальные требования, создавая сценарии событий. В этих сценариях можно настроить систему таким образом, чтобы не только уведомлять пользователя о срабатывании датчиков, но и выполнять определенные действия с помощью устройств. Например, можно настроить систему так, чтобы она автоматически регулировала температуру в помещении при определенных условиях.

С помощью этой системы пользователи могут интегрировать датчики задымления, движения и открытия дверей/окон для обеспечения безопасности своего дома. Датчики задымления могут оперативно реагировать на возможные пожары, предупреждая об опасности и активируя аварийную сигнализацию. Датчики движения и открытия дверей/окон могут использоваться для контроля доступа и обеспечения безопасности внутри дома.

Кроме того, система поддерживает интеграцию различных устройств, таких как видеокамеры для наблюдения за обстановкой внутри и вокруг дома, сирены для предупреждения о возможных инцидентах, «умные розетки» для управления электроприборами издалека, а также датчики температуры, влажности и протечки воды для контроля за условиями в помещении и предотвращения аварийных ситуаций.

Система предлагает гибкие варианты оповещения пользователей о событиях. Оповещение может осуществляться через Push-уведомления на мобильных устройствах, SMS-сообщения или электронные письма. Пользователи могут выбирать наиболее удобный для себя способ оповещения или использовать несколько вариантов одновременно.

Благодаря своей гибкости и простоте использования, система «Умный дом» от «Белтелеком» пользуется популярностью среди пользователей, которые ценят возможность настройки системы под свои потребности и удобные варианты оповещения.



Рисунок 1.3 – «Умный дом» от «Белтелеком»

Данный продукт также имеет ряд недостатков:

– зависимость от оператора;

– процесс подключения и настройки устройств может быть сложным для некоторых пользователей.

1.1.3 «Умный дом» от «Xiaomi»

Данный производитель предоставляет широкий ассортимент разнообразной техники, в том числе технику, которая отвечает требованиям «умного дома» [3]. В их ассортименте можно найти устройства для обеспечения безопасности и поддержания чистоты в помещении, что делает их продукцию универсальным решением для создания современного «умного дома».

Для удобного управления и взаимодействия с этими устройствами предоставляется специальное приложение (см. рисунок 1.4), обеспечивающее полную совместимость устройств одной экосистемы и их простую настройку. Благодаря фокусу данного производителя на широкий потребительский рынок, все устройства легко подключаются и не требуют значительных временных затрат на настройку.

Тем не менее, стоит отметить, что у продукции этого производителя есть и некоторые недостатки. Например, некоторые датчики могут быть дешевыми и менее надежными, а обновления приложения для «умного дома» выпускаются не так часто, как хотелось бы. Кроме того, большинство устройств требуют постоянного подключения к электросети и Wi-Fi для полноценной работы.

Таким образом, данный производитель предлагает доступное решение для знакомства с концепцией «умного дома. Однако, у них также имеются и более дорогие варианты аппаратуры с расширенными функциональными возможностями и повышенной надежностью.

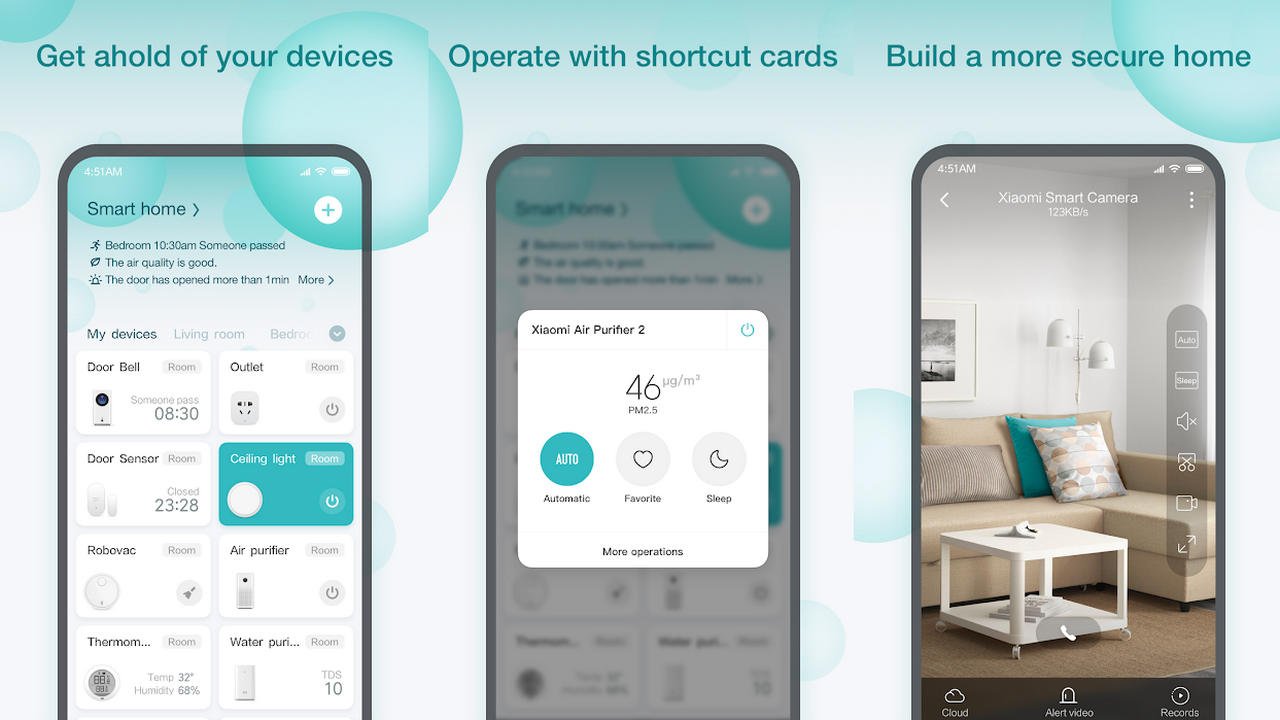


Рисунок 1.4 – Приложение Xiaomi Smart Home

1.1.4 Координатор от Яндекс

Для обеспечения функционирования всех устройств в «умном доме» необходимо наличие центрального устройства, которое выступает в качестве координатора, обеспечивая связь и взаимодействие между пользователем и системой. Одним из вариантов такого решения являются продукты компании Яндекс [4], предоставляющие широкий ассортимент устройств, способных выполнять функции координаторов (см. рисунок 1.5). Эти устройства не только обеспечивают надежное соединение и взаимодействие с другими устройствами в комплексе, но также предоставляют пользователю удобный интерфейс для управления всем комплексом. Благодаря интеграции с различными экосистемами и голосовыми ассистентами, устройства от Яндекса обеспечивают гибкость и удобство использования.

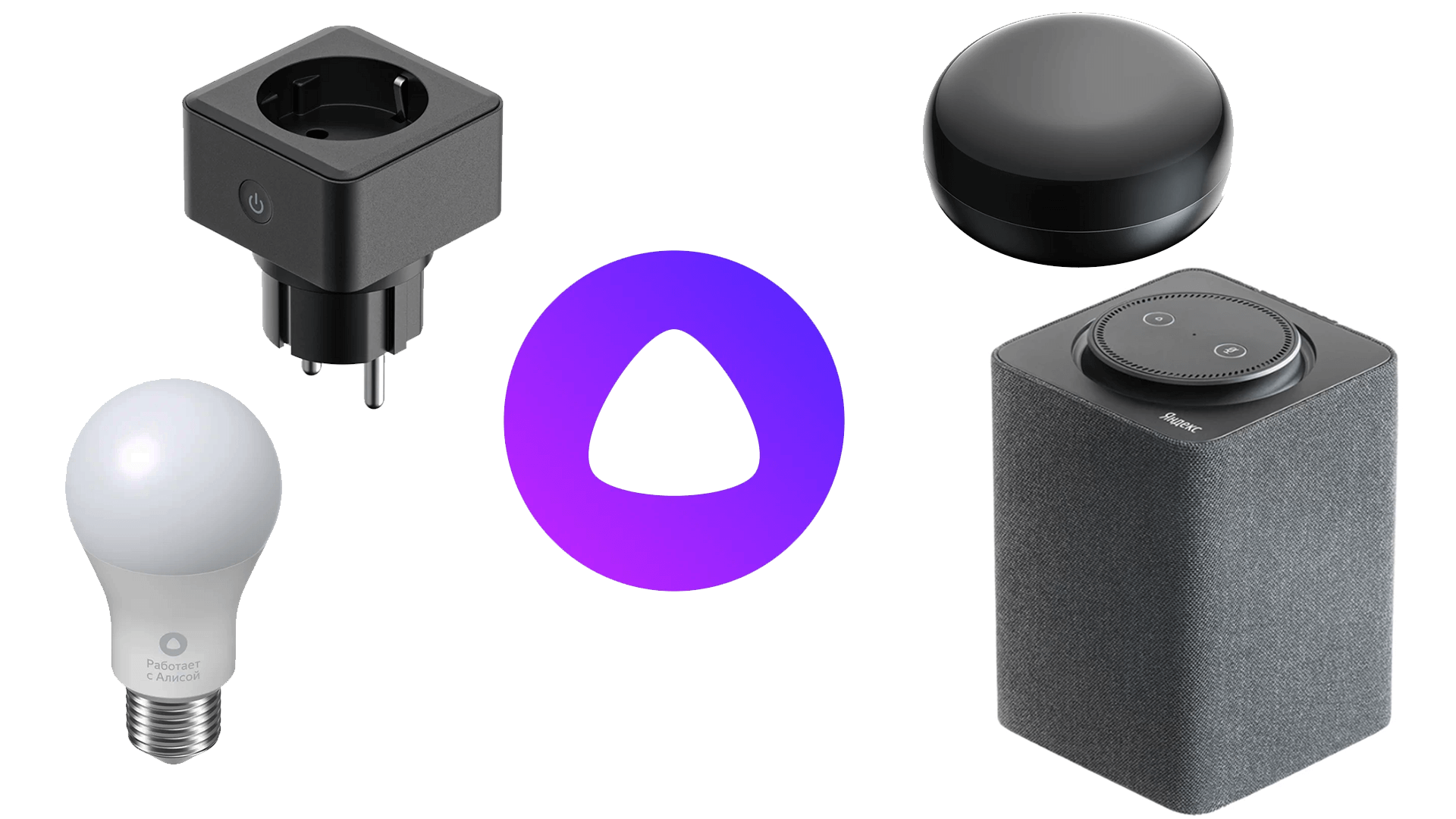


Рисунок 1.5 – «Умный дом» от Яндекс

Наиболее распространённым вариантом устройства, выступающего в роли координатора в «умном доме», является «умная колонка» с голосовым помощником Алиса (см. рисунок 1.6). Это устройство способно подключать различные оконечные устройства для управления всеми аспектами «умного дома».

«Умная колонка» Алиса предоставляет возможность управления, информирования и выполнения различных функций как с использованием мобильного приложения, так и с помощью голосовых команд. Это делает взаимодействие с «умным домом» более удобным и интуитивно понятным для пользователей, позволяя им контролировать своё пространство с легкостью и эффективностью.

Без интернет-соединения ПО неспособно функционировать должным образом. Обработка голосовых команд происходит не на устройстве, а на серверах «Яндекса». После отправки команды «Алисе» она передается на сервер для обработки, а затем возвращается на устройство для выполнения. Даже при значительной задержке процесс обработки остается быстрым и эффективным.

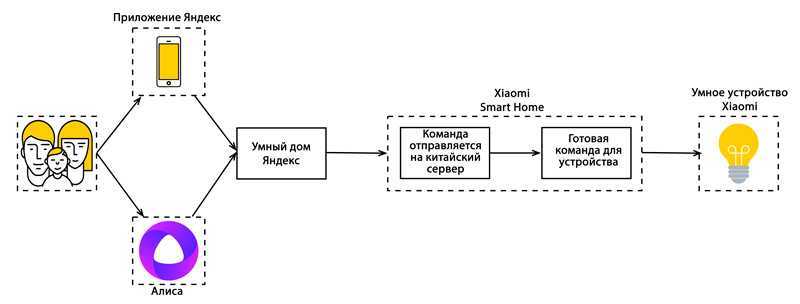


Рисунок 1.6 – Схема запросов в «Умном доме» от Яндекс

«Умный дом» от Яндекса уже интегрирован с множеством устройств от таких производителей, как Philips, Redmond, Rubetek, Samsung и Xiaomi (см. рисунок 1.7). В дополнение компания разработала и представила собственные устройства, включая «умную лампочку, розетку и пульт». Эти устройства обладают различным функционалом, который дополняет экосистему «умного дома».



Рисунок 1.7 – Совместимые устройства с координатором от Яндекс

Плюсом выбора продукта от Яндекс является то, что устройства данной компании прекрасно взаимодействуют с устройствами популярных производителей, работающими по протоколу Zigbee. Это позволяет пользователям легко интегрировать разнообразные устройства в свой «умный дом», выбирая из широкого ассортимента рынка и создавая гибкую систему под свои нужды.

Дополнительно, управление системой осуществляется с помощью мобильного приложения с современным интерфейсом (см. рисунок 1.8), что делает использование и контроль «умного дома» максимально удобным для пользователя. Это приложение позволяет настраивать интерфейс в соответствии с индивидуальными предпочтениями и потребностями, обеспечивая интуитивно понятное и персонализированное взаимодействие с системой. Такой подход делает использование «умного дома» более эффективным и удобным для каждого пользователя.



Рисунок 1.8 – Мобильное приложение от Яндекс

Однако, наряду с многочисленными преимуществами, у данного решения имеются и определенные недостатки:

1. Подписочная система. Для полноценного функционирования некоторых возможностей требуется подписка, что может стать дополнительным финансовым бременем для пользователей.
2. Высокая стоимость оборудования. Несмотря на качество и функциональность устройств, их стоимость может быть значительной, что делает данное решение недоступным для определенного сегмента потребителей.

В итоге, хотя данный производитель предлагает весьма привлекательные решения с широкими возможностями интеграции в домашнюю среду, высокая стоимость оборудования и необходимость в подписке могут создать некоторые препятствия для пользователей.

## 1.2 MQTT-протокол

В силу того, что большинство «умных домашних устройств» находятся в локальной сети, для того чтобы они могли функционировать в качестве устройств интернета вещей (IoT), необходимо осуществить их подключение к сети Интернет. Благодаря этому, собирать данные с датчиков и управлять конечными устройствами «умного дома» будет возможно из любой точки планеты. Подобный подход существенно упрощает коммуникацию между человеком и системой, ведь в настоящее время доступ в Интернет из мобильных устройств обеспечен почти во всех частях мира.

В большинстве случаев для управления конечными устройствами «умного дома» используется телефон с приложением системы, и в редких случаях веб-версии приложений данных систем. Такой подход делает мобильный телефон еще одним устройством IoT, содержащий различные возможности, включая экран для визуализации информации, полученной от датчиков, и контроллера конечных устройств.

В сети «умного дома» конечные устройства и координатор общаются при помощи протоколов Zigbee, Z-Wave и др., но для коммуникации между координатором и мобильным телефоном данный протокол не подходит. Для данной цели более приемлим протокол MQTT (Message Queue Telemetry Transport) [6], являющийся протоколом потоковой передачи информации с ограниченой производительностью процессора, то есть устройствами IoT. Протокол MQTT, построенный на основе TCP/IP, используется в сетях с низкой пропускной способностью, что делает его идеальным выбором для «умного дома».

Отличия протокола MQTT, выделяющие его среди других протоколов:

* нейтрален к содержимому сообщения;
* подходит для дистанционных коммуникаций «один ко многим» и приложений с разделением;
* имеет функцию LWT (Last Will and Testament, «последняя воля и завещание») для оповещения о нештатном отключении клиента;
* основан на протоколе TCP/IP для основных коммуникационных задач;
* разработан для доставки сообщений с определенной частотой: «максимум один раз», «минимум один раз» и «ровно один раз».

Система связи, использующая MQTT (см. рисунок 1.9), включает в себя издателей, сервер-брокер и одного или нескольких клиентов (подписчиков). Издатель не нуждается в настройке для определения количества или расположения подписчиков, получающих сообщения. Подписчики также не требуют настройки для определения конкретного издателя. В системе может присутствовать несколько брокеров, распространяющих сообщения. Издатель помещает сообщения в темы (topic), откуда их могут читать клиенты, подписанные на данную тему. Издатель и подписчик не могу коммуницировать напрямую, поэтому они не знают о существовании друг друга. В это же время подписчики могут читать сообщения из многих тем, на которые они подписаны, как и издатели имеют возможность помещать свои сообщения в различные темы.

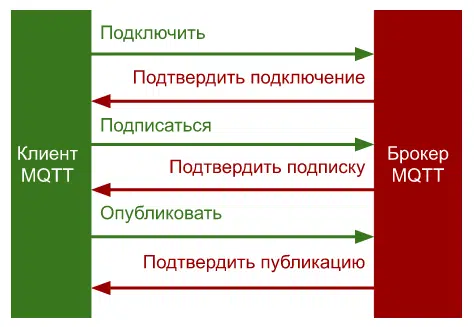


Рисунок 1.9 – Схема простого взаимодействия по протоколу MQTT

Данные, отправленные или полученные через брокер MQTT, будут представлены в двоичном формате, поскольку MQTT является бинарным протоколом. Это означает, что для того чтобы понять содержимое сообщения, необходимо выполнить интерпретацию двоичных данных.

Топики в MQTT составлены из символов в кодировке UTF8 и имеют структуру, аналогичную дереву файловой системы в UNIX. Этот подход обеспечивает удобство в названии сущностей таким образом, что они понятны человеку. Например:

– home/kitchen/temperature;

– home/sleeping-room/pressure;

– home/outdoor/light.

Такая организация позволяет наглядно отслеживать передаваемые данные и упрощает разработку и отладку кода, не требуя запоминания числовых адресов расположения данных.

Устройства, использующие MQTT, взаимодействуют с брокером посредством определенных типов сообщений. Ниже перечислены основные из них:

* Connect – установка соединения с сервером-брокером;
* Disconnect – оборвать соединение с сервером-брокером;
* Publish – опубликовать данные в топик;
* Subscribe – подписаться на топик;
* Unsubscribe – отписаться от топика.

Для удобства в уже готовых системах каждому конечному устройству назначается свой топик с определенными вложенными разделами. Именно по этим топикам происходит рассылка уведомлений о изменениях определенных параметров и управление устройствами. Когда сообщения публикуются в соответствующий топик, устройство, подписанное на этот топик, получает сообщение и реагирует на него.

## 1.3 Протоколы передачи данных

1.3.1 Протокол I2С

I2C (Inter–Integrated Circuit)[7] – это протокол последовательной связи, который обеспечивает взаимодействие между различными компонентами внутри электронных устройств.



Рисунок 1.11 – Схема шины I2С

Применение данного стандарта для передачи информации способствует сокращению количества взаимосвязей между интегральными схемами, что приводит к уменьшению числа требуемых контактов и дорожек. Также встроенный I2C-протокол исключает необходимость в дополнительных средствах, таких как дешифраторы адресов, и другой внешней логике для согласования.

Для передачи информации используются лишь две линии (см. рисунок 1.11):

* SDA (для данных);
* SCL (для синхронизации).

Каждое устройство определяется как ведущее или ведомое, а также обладает уникальным (в пределах шины) адресом. Максимальное количество устройств, подключаемых к одной шине, ограничено ее емкостью – 400 пФ.

Ведущее устройство – то, которое инициирует передачу данных и контролирует сигналы синхронизации, а также завершает передачу данных. Оно может отправлять данные или запрашивать их.

Ведомое устройство – любое адресуемое устройство по отношению к ведущему. Оно может принимать данные или отправлять их по запросу ведущего.

Передатчик – устройство, передающее данные по шине.

Приемник – устройство, принимающее данные с шины.

Отношения ведущий-ведомый и приемник-передатчик не постоянны и зависят от направления передачи данных в текущий момент времени. Обычно стандарт I2C предполагает наличие одного ведущего устройства в один момент времени, однако допускается наличие нескольких ведущих устройств на шине без нарушения работы системы. В таком случае применяется процедура арбитража для предотвращения ошибок.

Каждое прикрепленное устройство имеет свой собственный адрес. Производитель задает первые три бита адреса, а последний бит указывает на запись или чтение устройства. Такая структура обеспечивает простую передачу данных. 7-битная адресация шины I2C позволяет подключить до 128 устройств.

1.3.2 Протокол Zigbee

Zigbee – это беспроводной протокол, специально разработанный для создания сетей малой дальности с низким энергопотреблением [8]. Он широко используется для передачи данных в устройствах Интернета вещей (IoT) и смарт-домах. Протокол Zigbee работает в частотном диапазоне 2,4 ГГц и использует технологию малого радиуса действия, что обеспечивает низкое энергопотребление и надежную связь. Основные характеристики протокола включают в себя низкое энергопотребление, минимальные задержки передачи данных, возможность создания крупных сетей до тысяч устройств, самоорганизацию и безопасность данных.

ZigBee устанавливает ячеистую сеть устройств, соединенных между собой с помощью маломощных радиосигналов. Всякое устройство в сети, действуя как маршрутизатор, может принимать и перенаправлять сообщения другим устройствам, что обеспечивает избыточность связи и повышает надежность сети.

Уникальная схема адресации в ZigBee позволяет однозначно идентифицировать устройства в сети. Это обеспечивает возможность использования функций безопасности, таких как шифрование и аутентификация, для защиты сети от несанкционированного доступа.

Сети Zigbee имеют несколько типов структур, которые могут быть использованы в различных сценариях в зависимости от требований к конкретному приложению. Вот некоторые из основных структур сетей Zigbee:

1. Звезда (Star). В звездной структуре одно устройство, обычно координатор сети, служит центром и связывается напрямую со всеми другими устройствами в сети. Это простая и надежная структура, но она имеет ограничения по расстоянию и масштабируемости.
2. Дерево (Tree). В данной структуре устройства связаны между собой в иерархической форме, создавая иерархию маршрутизаторов и конечных устройств. Это позволяет увеличить дальность связи и масштабируемость сети.
3. Сеть с маршрутизацией (Mesh). Сеть с маршрутизацией является наиболее распространенной структурой в Zigbee. Она позволяет устройствам действовать как маршрутизаторы, перенаправляя сообщения между устройствами, даже если они находятся вне прямой радиочастотной области. Это обеспечивает более высокую надежность и устойчивость связи.
4. Кластерные структуры. Эти структуры позволяют группировать устройства по функциональным кластерам или областям, что облегчает организацию и управление сетью.

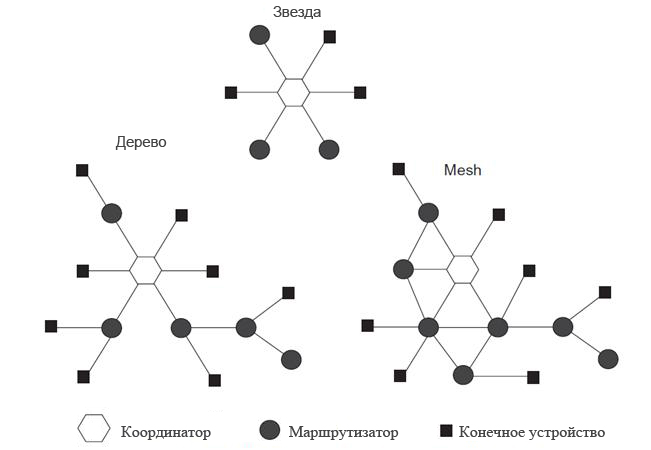


Рисунок 1.12 – Топология Zigbee

# 2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

При разработке программного обеспечения для «умного дома» ключевым моментом является структурирование его компонентов в логические блоки, каждый из которых отвечает за определенные функциональные задачи.

Целью разбиения программного обеспечения на логические блоки является облегчение понимания его работы и структуры, а также упрощение процесса разработки и поддержки. Каждый блок выполняет определенные функции и имеет четко определенный интерфейс для взаимодействия с другими блоками.

Рассмотрим следующие основные блоки структуры программного обеспечения:

– блок интерфейса приложения;

– блок соединения с брокером MQTT;

– блок приема-передачи данных брокеру MQTT;

– блок обработки данных;

– блок базы данных;

– блок тестирования;

– блок координатора;

– блок конечных устройств;

– блок MQTT брокера.

Структурная схема, изображенная на чертеже ГУИР.400201.006 С1, иллюстрирует взаимосвязь между различными блоками программного обеспечения.

В конечном итоге, все модули блока взаимодействуют и образуют единую систему, позволяющую корректно реагировать на запросы и изменения субъекта. В числе прочего, это позволяет обеспечивать пользователю интерфейс и комфорт взаимодействия с системой умного дома, путем предоставления ему актуальной информаци и выполнения необходимых функций. Далее будет описано именно практическое назначение и способ работы каждого из блоков, для наглядности их реализации.

## 2.1 Блок пользовательского интерфейса

Блок интерфейса приложения играет ключевую роль в системе «умного дома», поскольку он обеспечивает взаимодействие пользователя с системой. Его основной задачей является создание удобного и интуитивно понятного пользовательского интерфейса, который позволяет пользователю управлять и мониторить различными компонентами «умного дома». Ниже приведено подробное описание функциональности и особенностей блока интерфейса приложения:

1. Пользовательский опыт. Блок интерфейса разрабатывается с учетом пользовательского опыта, чтобы обеспечить удобство использования для широкого круга пользователей. Это включает в себя понятную навигацию, простоту в использовании элементов управления и понятные инструкции.
2. Визуализация данных. Интерфейс приложения предоставляет пользователю визуализацию данных о состоянии устройств в системе и системы в целом. Это может быть представление в виде графиков, диаграмм, таблиц или других подходящих форматов, которые помогают пользователю лучше понять текущее состояние системы.
3. Элементы управления. Блок интерфейса содержит различные элементы управления, позволяющие пользователю взаимодействовать с компонентами «умного дома». Это могут быть кнопки, ползунки, переключатели, поля ввода и другие элементы, позволяющие изменять параметры устройств, включать и выключать функции, а также добавлять свои системы и устройства в них.
4. Отображение уведомлений. Интерфейс приложения также поддерживает отображение уведомлений пользователю о различных событиях и состояниях системы. Это могут быть предупреждения о срабатывании сигнализации, информация о текущих работающих процессах, оповещения о необходимости обслуживания устройств и т.д.
5. Настройки и персонализация. Блок интерфейса предоставляет пользователю возможность настройки и персонализации системы в соответствии с его предпочтениями. Это могут быть настройки устройств и другие параметры, которые делают систему более удобной и функциональной для конкретного пользователя.
6. Адаптивность приложения. В случае мобильного приложения блок интерфейса также оптимизирован для работы на мобильных устройствах, обеспечивая удобство использования на смартфонах и планшетах. Это включает в себя адаптивный дизайн, оптимизированный для различных размеров экранов, а также использование жестов для управления.

Благодаря функциональности и особенностям блока интерфейса приложения пользователь может легко и удобно управлять системой «умного дома», получая необходимую информацию и выполняя нужные действия в соответствии с его потребностями и предпочтениями.

При разработке проектов под Android с использованием IDE Android Studio широко используется подход с определением визуального интерфейса в специальных файлах XML. Эти XML-файлы являются ресурсами разметки и содержат определения визуального интерфейса в формате XML-кода. Этот подход аналогичен использованию HTML-файлов для определения веб-интерфейса в веб-разработке, где логика приложения отделена от визуальной составляющей и реализована с помощью JavaScript.

Использование XML для объявления пользовательского интерфейса позволяет отделить интерфейс приложения от его логики, что позволяет изменять визуальное оформление без необходимости внесения изменений в Java-код. Например, в приложении можно определить различные разметки для разных ориентаций экрана, разных размеров устройств, различных языков и так далее. Кроме того, объявление интерфейса в XML упрощает визуализацию его структуры и облегчает процесс отладки, что делает разработку более эффективной и удобной.

После объявления в XML, компоненты становятся доступными для манипуляций из основного кода приложения. Это достигается путем связывания XML-компонентов с объектами в Java-коде с помощью механизма «надувания», который позволяет получить доступ к компонентам по их уникальному идентификатору. После связывания компоненты можно изменять, управлять и реагировать на действия пользователя из основной части проекта.

## 2.2 Блок соединения с брокером MQTT

Блок соединения с брокером MQTT является важной частью системы «умного дома», ответственной за установление и поддержание связи с сервером MQTT. Этот блок обеспечивает надежное соединение между приложением и брокером MQTT, что позволяет обмениваться сообщениями и данными между устройствами системы.

Основные задачи блока соединения с брокером MQTT включают:

1. Установка соединения. Приложение должно установить соединение с сервером MQTT перед тем, как пользователь открывает страницу с устройствами, чтобы начать отправку и получение сообщений для отображения актуальных данных, если это возможно. Этот процесс включает в себя инициализацию сетевого соединения и передачу необходимых параметров для аутентификации и подключения к серверу MQTT.
2. Поддержание связи. После установления соединения блок должен поддерживать активное соединение с сервером MQTT, чтобы обеспечить непрерывный обмен сообщениями. Это включает в себя периодическую отправку «ping» запросов для подтверждения активности соединения и переподключение в случае его разрыва.
3. Обмен сообщениями. После успешного установления и поддержания соединения блок должен обеспечивать передачу сообщений между приложением и сервером MQTT. Это включает в себя отправку сообщений от приложения к брокеру для управления устройствами «умного дома» и получение обновлений состояния устройств от брокера.
4. Обработка ошибок. Блок должен обрабатывать возможные ошибки при установлении и поддержании соединения с брокером MQTT, такие как потеря сетевого соединения или недоступность сервера. В случае ошибок он должен предпринимать соответствующие действия для восстановления связи и продолжения работы.

## 2.3 Блок приема-передачи данных MQTT брокеру

Блок приема-передачи данных MQTT брокеру выполняет функцию обмена сообщениями между приложением и брокером MQTT в системе «умного дома». Этот блок играет важную роль в передаче управляющих команд от приложения к устройствам «умного дома» через брокер MQTT, а также в получении обновлений состояния устройств от брокера и передаче их в приложение для дальнейшей обработки.

Основные задачи блока приема-передачи данных MQTT брокеру включают отправку и получение сообщений, а также управление подписками на топики MQTT и обработку ошибок. Блок отправляет управляющие команды от приложения к устройствам «умного дома» через брокер MQTT и получает от них сообщения с обновлениями состояния устройств или другой важной информацией. Полученные сообщения обрабатываются блоком для извлечения нужной информации и принятия соответствующих действий. Блок также управляет подписками на топики MQTT, определяя, на какие топики приложение подписано для получения сообщений от брокера, что позволяет эффективно управлять потоком данных и получать только необходимую информацию. При возникновении ошибок блок обрабатывает их, предпринимая соответствующие действия для восстановления связи и обеспечения непрерывной работы системы.

## 2.4 Блок обработки данных

Информация, полученная из различных источников в блок обработки данных, где преобразовываются для последующего использования в различных блоках.

Блок получает «сырые сообщения» в виде JSON от брокера MQTT, содержащие информацию о состоянии устройств в системе «умного дома» или другие важные данные. Эти сообщения могут содержать информацию о температуре, освещенности, статусе устройств и других параметрах, которые необходимо обработать и отобразить пользователю в виде виджетов.

В функциональную часть блока также входит обязанность принимать данные, введенные пользователем через поля ввода в интерфейсе приложения. Эти данные могут включать в себя команды управления устройствами, настройки параметров системы и другую информацию, которую необходимо обработать и применить.

Кроме того, блок обрабатывает события, связанные с действиями пользователя, такие как нажатие кнопок для переключения состояния светодиодов и других управляемых устройств. Он интерпретирует эти события и принимает соответствующие действия, например, отправку сформированной команды на изменение состояния устройства.

Информация, приходящая от полей ввода, связанных с данными системы и ее устройствами передаются в блок обработки данных. Далее происходит создание и поддержка внутренней модели системы «умного дома», включая список доступных устройств, их параметры и текущее состояние. Он обновляет эту модель на основе полученной информации от MQTT и действий пользователя, чтобы обеспечить актуальное состояние системы и корректное функционирование устройств, например, добавление последних полученных данных от датчика, для последующего корректного отображения информации при перезагрузке приложения.

## 2.5 Блок базы данных

Информация, получаемая из блока обработки данных, записывается в базу данных, где она сохраняется в специальных таблицах, предназначенных для хранения данных о системе и устройствах. Чаще всего для этого используется SQLite.

В таблице «системы» хранятся данные, необходимые для установления связи с брокером MQTT, такие как адрес сервера, порт, учетные данные и название системы для удобства отображения.

В таблице «устройства» содержатся поля для хранения последних данных, полученных от MQTT брокера, уникальный адрес устройства для подключения к сети Zigbee, изображение самого устройства, выбранное пользователем, каналы управления (например, для переключения датчика или светодиода), тип устройства и его дружественное имя.

Данная информация позволяет системе эффективно управлять подключением к MQTT и обеспечивать надежную передачу данных в системе умного дома. Информация может храниться в базе данных в «сыром виде», что обеспечивает гибкую обработку данных для различных задач в приложении.

## 2.6 Блок тестирования

Блок тестирования представляет собой важный компонент разработки комплекса «умного дома», ответственный за проверку корректности работы приложения и его компонентов. В этом блоке проводятся несколько видов тестирования, включая модульное и пользовательское тестирование, с целью обнаружения и устранения ошибок и недочетов в приложении.

Модульное тестирование направлено на проверку отдельных модулей приложения, таких как блоки обработки данных, соединения с брокером MQTT и интерфейса приложения. Пользовательское тестирование проводится для оценки удобства использования приложения и выявления потенциальных проблем, с которыми может столкнуться конечный пользователь.

Блок тестирования включает в себя разработку тестовых сценариев, проведение тестов, анализ результатов и исправление выявленных ошибок. Целью этого блока является обеспечение качества и надежности приложения перед его выпуском в продакшн и после каждого обновления. Тестирование позволяет выявить и устранить проблемы до того, как они окажутся в реальных условиях эксплуатации, что способствует повышению удовлетворенности пользователей и обеспечивает успешное функционирование комплекса «умного дома». Кроме того, благодаря систематическому тестированию, становится возможным внедрять новые функции и улучшения, минимизируя риск возникновения непредвиденных проблем.

## 2.7 Блок координатора

Блок координатора ответственен за взаимодействие с устройствами Zigbee в системе умного дома. Он обеспечивает связь с устройствами Zigbee, управляет ими и получает от них данные. Координатор выполняет конвертацию данных, полученных от устройств Zigbee, в формат, понятный брокеру MQTT, и обратно, обеспечивая эффективную передачу информации между устройствами и приложением.

В состав блока координатора входит прошивка устройства, которая представляет собой программное обеспечение, устанавливаемое на устройство координатора.

Координатор подключается к брокеру MQTT, обеспечивая передачу данных между конечными устройствами и мобильным приложением пользователя. Это позволяет системе умного дома контролировать и управлять устройствами через стандартный протокол обмена сообщениями.

Сопряжение конечных устройств с координатором осуществляется через протокол Zigbee, который обеспечивает беспроводную связь между ними. После сопряжения устройства могут обмениваться информацией о состоянии с координатором и участвовать в работе комплекса.

Для дополнительной настройки и управления координатором можно использовать HTTP протокол. Кроме того, координатор позволяет использовать Lua скрипты. Lua - это легкий и гибкий скриптовый язык программирования, который позволяет создавать сценарии для автоматизации различных задач. Например, с помощью Lua скриптов можно настраивать расписания работы устройств, настраивать реакцию системы на определенные события или условия, а также выполнять другие задачи, улучшающие функциональность и удобство использования комплекса.

## 2.8 Блок конечных устройств

Блок конечных устройств в системе умного дома выполняет задачу получения данных от датчиков и формирования соответствующих кластеров для передачи в сети Zigbee. Кластеры представляют собой логические группы данных, которыми устройства Zigbee обмениваются между собой. Каждый кластер имеет свое определенное назначение, например, кластер может содержать данные о температуре, влажности, освещенности и т.д., в зависимости от типа датчика.

Конечные устройства, такие как датчики, работают на уровне протокола Zigbee и формируют данные в соответствии с необходимыми кластерами. Например, датчик температуры будет отправлять данные о текущей температуре в специально определенный кластер для температурных данных.

Полученные кластеры данных затем передаются в сеть Zigbee и доставляются до координатора. Координатор, в свою очередь, правильно интерпретирует эти данные, конвертируя их в формат, понятный брокеру MQTT.

Кроме того, для работы конечных устройств необходимо настроить контакты, которые используются для подключения датчиков или для взаимодействия с ними. Настройка контактов устройства позволяет определить, какие сигналы будут приниматься или отправляться через эти самые контакты, что важно для правильного взаимодействия между устройствами.

## 2.9 Блок MQTT брокера

Блок MQTT брокера представляет собой центральный элемент в системе умного дома, отвечающий за обмен сообщениями между всеми устройствами и компонентами системы. Основные функции блока MQTT брокера включают в себя:

1. Прием и передача сообщений. Брокер MQTT принимает сообщения от различных устройств и компонентов системы, а затем передает их нужным адресатам в соответствии с подписками на топики.
2. Маршрутизация сообщений. Он обеспечивает маршрутизацию сообщений между различными устройствами и компонентами системы, гарантируя доставку сообщений тем получателям, которые подписаны на соответствующие топики.
3. Управление подписками. Брокер управляет подписками на топики MQTT, определяя, какие устройства и компоненты получают определенные сообщения.

Установка сервера MQTT на публичный адрес обеспечивает доступ к системе умного дома из любой точки мира, где есть доступ к интернету. Это означает, что пользователи могут контролировать и управлять своими «умными устройствами» удаленно, независимо от их местоположения, используя приложение.