

Санкт–Петербургский государственный университет

*Соловьев Дмитрий Николаевич*

Выпускная квалификационная работа

*Система домашней автоматизации с использованием  
децентрализованного подхода в хранении и обработке данных.*

Уровень образования: бакалавриат

Направление 02.03.02 «Фундаментальная информатика и информационные  
технологии»

Основная образовательная программа СВ.5190.2021 «Большие данные и  
распределенная цифровая платформа»

Научный руководитель:

доцент, кафедра компьютерного моделирования и  
многопроцессорных систем,  
к.ф.-м.н., PhD

Корхов Владимир Владиславович

Рецензент:

доцент, кафедра компьютерного моделирования и  
многопроцессорных систем,  
к.т.н.

Дик Геннадий Давидович

Санкт-Петербург  
2025 г.

## Аннотация

С увеличением количества устройств интернет вещей (ИВ) в системах домашней автоматизации (СДА) остро встает вопрос топологий сетей, в которых находятся эти устройства, а также способах хранения и передачи данных между устройствами. Многим системам, использующим централизованный подход в хранении и обработке информации, свойственна существенная потеря функциональности при выходе из строя центрального узла. В ходе исследования были изучены принципы, методологии и архитектуры, применимые к программно-аппаратным комплексам для построения СДА и позволяющие повысить уровень безопасности данных, отказоустойчивость и увеличить скорость передачи информации в рамках системы за счет использования распределенного подхода к хранению и обработке информации, разработан прототип распределенной СДА. Результаты проведенного исследования и предложенная в нем архитектурная модель могут быть использованы не только в системах умных домов, но и в умных городах и на предприятиях.

## Abstract

With the increasing number of Internet of Things (IoT) devices in home automation systems (HAS), the topologies of the networks that house these devices, as well as how data is stored and transferred between devices, are becoming an issue. Many systems that use a centralized approach in storing and processing information are characterized by a significant loss of functionality when a central node fails. In the course of the research the principles, methodologies and architectures applicable to hardware-software complexes for the construction of ADS were studied, which allow to increase the level of data security, fault tolerance and increase the speed of information transfer within the system due to the use of distributed approach to information storage and processing, and a prototype of distributed ADS was developed. The results of the conducted research and the architectural model proposed in it can be used not only in smart home systems, but also in smart cities and enterprises.

# Содержание

<b>1</b>	<b>Введение</b>	<b>4</b>
1.1	Актуальность темы . . . . .	4
1.2	Объект исследования . . . . .	4
1.3	Предмет исследования . . . . .	4
1.4	Постановка задачи . . . . .	4
<b>2</b>	<b>Глава 1</b>	<b>6</b>
2.1	Топологии беспроводных сетей . . . . .	6
2.1.1	Топология точка-точка . . . . .	6
2.1.2	Топология «Звезда» . . . . .	6
2.1.3	Ячеистая топология . . . . .	6
2.1.4	Топология «кластерное дерево» . . . . .	7
2.2	Протоколы беспроводных сетей . . . . .	7
2.2.1	Wi-Fi и Wi-Fi HaLow . . . . .	8
2.2.2	Bluetooth Low Energy (BLE) и Bluetooth Mesh . . . . .	9
2.2.3	Zigbee и Zigbee Pro . . . . .	10
2.2.4	Сравнение представленных протоколов . . . . .	13
2.2.5	Доступность для разработчиков платформ, поддерживающих представленные протоколы . . . . .	15
<b>3</b>	<b>Глава 2</b>	<b>16</b>
3.1	Описание системы . . . . .	16
3.2	Выбранная топология системы . . . . .	18
3.3	Взаимодействие устройств . . . . .	18
3.4	Добавление устройств и настройка сценариев . . . . .	19
3.5	Обработка событий . . . . .	19
<b>4</b>	<b>Глава 3</b>	<b>20</b>
4.1	Графический интерфейс . . . . .	20
4.2	Планировщик . . . . .	21
<b>5</b>	<b>Глава 4</b>	<b>22</b>
5.1	Тестирование и полученные оценки . . . . .	22
5.2	Вывод . . . . .	22
<b>6</b>	<b>Список литературы</b>	<b>23</b>
	<b>Приложение А. Исходные коды и дополнительные материалы</b>	<b>25</b>

# 1 Введение

## 1.1 Актуальность темы

Актуальность и практический аспект анализа систем и выбора оптимальной для построения системы домашней автоматизации:

1. Количество устройств с возможностью интеграции в систему умного дома среди потребителей быстро возрастает год к году. Особенности построения актуальных систем накладывает ограничения на масштабируемость системы, ее надежность и безопасность.
2. В системах домашней автоматизации с появлением новых типов устройств и попытки их интеграции возникают проблемы, связанные со скоростью реакции на события и с отсутствием приоритезации сообщений.

## 1.2 Объект исследования

Объектом научно-исследовательской работы являются системы домашней автоматизации, системы хранения, передачи и обработки информации.

## 1.3 Предмет исследования

Предметом научно-исследовательской работы является оптимизация обработки и передачи данных между устройствами ИВ в системе домашней автоматизации.

## 1.4 Постановка задачи

В рамках данной работы ставится цель провести комплексное исследование существующих СДА на рынке, проанализировать применяемые в них протоколы и выявить основные недостатки. На основании этого выбрать оптимальный протокол и разработать прототип системы, отвечающий сформулированным требованиям и решающий обозначенный список проблем.

Для достижения этой цели были сформулированы начальные условия (требования):

1. Сеть устройств в пределах одной зоны должна строиться по одноуровневому принципу, без централизованных узлов управления.
2. Взаимодействие устройств внутри зоны осуществляется напрямую, без посредников (ЦУД).
3. Для связи устройств из разных зон используются шлюзы-ретрансляторы, объединённые в единую распределённую сеть.

### Практическая значимость предлагаемого подхода:

- Уменьшение задержек, расширение сфер применения: в предлагаемой системе уменьшается количество узлов между устройством-инициатором и устройством-актуатором (при нахождении в одной сети - до нуля), за счет чего увеличивается скорость выполнения команд и повышается общая отзывчивость системы,

что расширяет сферы применения устройств ИВ для построения СДА, ранее нефункциональных ввиду относительно больших задержек.

- Повышение комфорта использования: данные о состоянии устройств, как и статистика, собираемая устройствами, хранится и обрабатывается на самих устройствах, что позволяет обеспечить короткое время восстановления работы после внештатных ситуаций (например, отключения питания) без длительной синхронизации.
- Упрощение настройки системы: благодаря использованию планировщика, автоматически распределяющего сценарии по устройствам, нет необходимости в ручной конфигурации каждого узла.
- Повышение отказоустойчивости: отказ сценария в одной локации возможен лишь при неисправности одного из двух устройств (инициатора и актуатора) и не зависит от иных устройств в сети. Объединение локаций, а также дистанционный мониторинг и управление системой осуществляется посредством распределенной системы шлюзов-ретрансляторов, что избавляет от единой точки отказа системы (точки входа).
- Повышение безопасности: распределенный подход для хранения сценариев и управления устройствами минимизирует получение злоумышленниками доступа ко всем устройствам. Система шлюзов-ретрансляторов, использующих VPN-соединение избавляет от единой точки отказа системы (точки входа).

## 2 Глава 1

### 2.1 Топологии беспроводных сетей

Все перечисленные беспроводные сети работают в одном или нескольких вариантах топологии. На рис. 1 приведены топологии беспроводных сетей различных конфигураций.

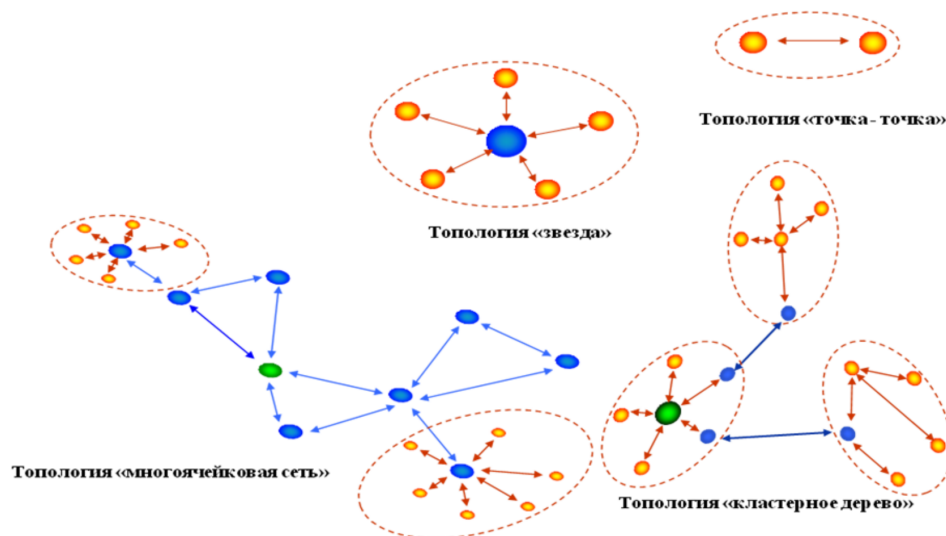


Рис. 1: Топологии беспроводных сетей [1]

#### 2.1.1 Топология точка-точка

Самый простой вариант организации сети из двух устройств. Как правило, узлы этой сети являются равноправными, то есть сеть одноранговая. Эта топология характерна для Bluetooth, ANT, RFID, RuBee, PDC, WI-FI, Insteon, UWB, ZigBee и прочих.

#### 2.1.2 Топология «Звезда»

Эта топология служит основой организации всех современных сетей связи и вычислительных сетей. Данную топологию используют протоколы WI-FI, Insteon, ZigBee, UWB, IDEN, CDMAOne, WIMAX, GSM, GPRS, UTMS.

#### 2.1.3 Ячеистая топология

Ячеистая топология — базовая полносвязная топология компьютерных сетей и сетей связи, в которой каждая рабочая станция сети соединяется со всеми другими рабочими станциями этой же сети. Характеризуется высокой отказоустойчивостью, сложностью настройки и избыточным расходом кабеля в проводных сетях. Каждый узел имеет несколько возможных путей соединения с другими узлами, за счет этого такая топология очень устойчива. Так как исчезновение одного из каналов не приводит к потере соединения между двумя компьютерами. Эта топология допускает соединение большого количества узлов и характерна, как правило, для крупных сетей, она строится из полносвязной путем удаления некоторых возможных связей.

Топология применима для сетей с использованием протоколов UWB, WI-FI, Insteon, ZigBee, UWB, IDEN, CDMAOne, WIMAX, GSM, GPRS, UTMS.

#### 2.1.4 Топология «кластерное дерево»

Топология «Кластерное дерево» образуется в основном в виде комбинаций вышеназванных топологий вычислительных сетей. Основание дерева вычислительной сети располагается в точке (корень), в которой собираются коммуникационные линии информации (ветви дерева). Вычислительные сети с древовидной структурой строятся там, где невозможно непосредственное применение базовых сетевых структур в чистом виде.

## 2.2 Протоколы беспроводных сетей

В данный момент большинство производителей ИВ используют централизованный подход для построения СДА [2]. В качестве центра умного дома (ЦУД) выступает устройство (хаб), объединяющее конечные устройства в единую сеть. Сценарии взаимодействия устройств могут быть локальными (сценарии хранятся в ЦУД) или облачными. Вариант с локальным расположением сценариев более предпочтительный, так как повышается отзывчивость системы и ее отказоустойчивость. Отдельную сложность представляет объединение устройств из разных экосистем: различные комбинации связей «устройство», «ЦУД», «облако».

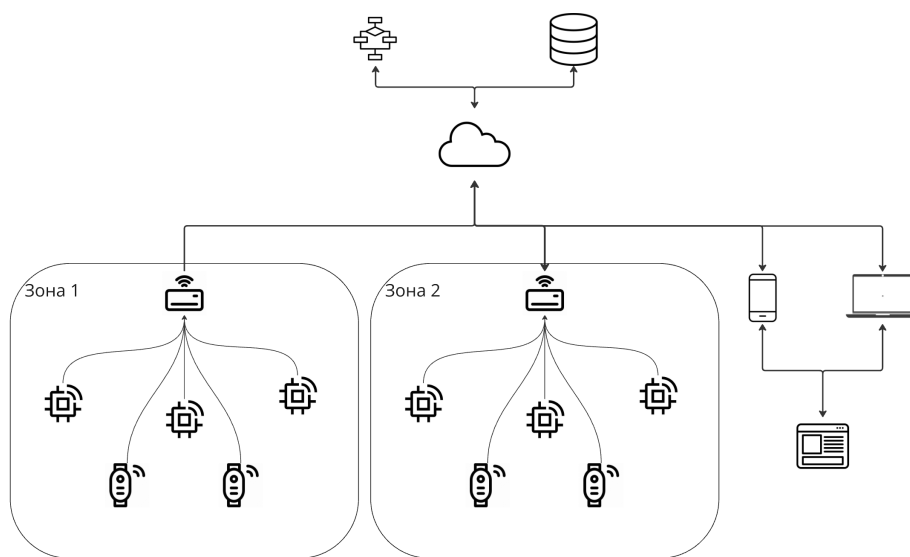


Рис. 2: Пример СДА с облачной обработкой событий

Большинство имеющихся на данный момент решений опираются на топологию «дерево», где в качестве корня рассматривается ЦУД или облако, что проиллюстрировано на рис. 5. При этом структура может подразумевать наличие единственных ветвей из корня, например «облако-ЦУД-устройства» [4]. Тенденции 2023-2024 г. по переносу обработки событий в ЦУД, который ранее использовался лишь для маршрутизации сообщений от устройств к облаку, позволили увеличить надежность и уменьшить задержки в системе, но не исключили имеющиеся проблемы полностью.

Стоит различать аппаратные ограничения, не позволяющие изменить топологию системы, связанные с особенностью работы протокола, например, топологию Bluetooth, позволяющую организовывать только связи типа точка-точка или точка - множество устройств, и архитектуру системы УД, подразумевающую централизованное выполнение сценария в ЦУД либо облаке.

### 2.2.1 Wi-Fi и Wi-Fi HaLow

Протокол Wi-Fi относится к семейству стандартов IEEE 802.11, реализующих беспроводную передачу данных в различных частотных диапазонах:

- 2,4 ГГц — стандарты 802.11b/g/n;
- 5 ГГц — 802.11a/ac;
- 6 ГГц — 802.11ax (Wi-Fi 6E).

Эти протоколы поддерживают высокую пропускную способность и широко распространены благодаря простой интеграции в клиентские устройства и устоявшейся практике применения. Однако они изначально не предназначены для энергоэффективных решений, что ограничивает их применение в системах домашней автоматизации, особенно если подразумевается использование батарейного питания. Так же различные виды конфигурации сети могут оказывать значительное влияние на время автономной работы IoT-устройства. [3]

**IEEE 802.11ah (Wi-Fi HaLow)** — расширение Wi-Fi в суб-ГГц диапазоне для IoT и промышленных задач. Обеспечивает дальность до километра, поддержку до тысяч устройств и низкое энергопотребление, что проиллюстрировано на рис. 3, благодаря механизмам Target Wake Time и RAW. Сохраняет уровень MAC совместимым с Wi-Fi, унаследовав безопасность и QoS.

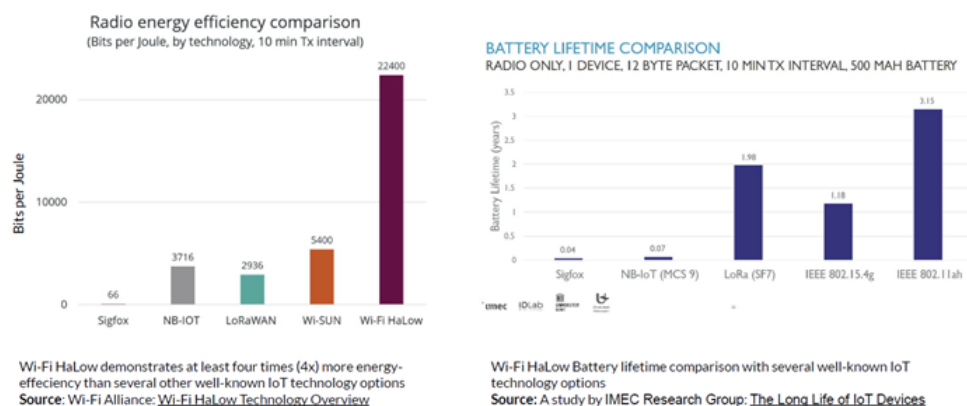


Рис. 3: Сравнение энергоэффективности протоколов [15]

### Основные элементы архитектуры и топологии

- **Диапазон частот:** 863–928 МГц (региональные поддиапазоны).
- **Узлы:**



- STA (датчики/актуаторы) с поддержкой энергосбережения.
  - AP (Access Point) — единственный координатор сети, задаваемый при развёртывании сети администратором.
  - Relay AP (опционально) — специальные узлы (ретрансляторы)
- **Топология:** поддерживаются несколько вариантов: звездная и древовидная с ретрансляцией (ограниченное дерево без полного L2-mesh) для расширения до удалённых STA.

**Преимущества и недостатки:** В качестве преимуществ IEEE 802.11ah можно выделить большую дальность связи (до 3 км вне помещений при использовании ретранслятора Relay AP), высокую плотность подключений (возможность обслуживания сотен устройств одной точкой доступа), а также низкое энергопотребление при малой активности. Вместе с тем, стандарту присущи: низкая пропускная способность по сравнению с Wi-Fi в диапазонах 2,4/5/6 ГГц, более высокая сложность и стоимость модулей по сравнению с устройствами на базе 802.15.4, из-за чего данный протокол не получил широкого распространения среди потребительских устройств, заняв только небольшую нишу промышленных IoT-устройств; а также сложности лицензирования различных диапазонов частот для разных стран. При отказе единственного AP вся BSS теряет синхронизацию и связь STA; стандарт не содержит механизмов внутреннего резервирования или автоматического фейловера.

## 2.2.2 Bluetooth Low Energy (BLE) и Bluetooth Mesh

Bluetooth Low Energy (BLE) — низкоэнергетичная радиотехнология, оптимизированная для передачи небольших объёмов данных с минимальным расходом батареи и поддерживающая энергосберегающие режимы. BLE определяет четыре базовые роли устройств: Central, Peripheral, Broadcaster и Observer, что позволяет строить радиосеть в топологии «Звезда» с центральным узлом (Central) и периферийными устройствами (Peripheral). Частотный диапазон BLE включает 40 каналов в диапазоне 2,4–2,4835 ГГц: 37 каналов для передачи данных и 3 канала для рекламы. Для масштабируемых IoT-решений существует расширение Bluetooth Mesh, позволяющее узлам ретранслировать сообщения и формировать распределённую сеть «many-to-many» поверх BLE.

### Роли устройств

- **Peripheral** — Энергоэффективное устройство, периодически транслирующее (advertise) данные и ждущее подключения от центрального узла (Central) 4b. Характерные примеры: пульсометры, датчики температуры, трекеры активности.
- **Central** — Стационарное устройство, выполняющее сканирование эфирных пакетов и устанавливающее соединения с одним или несколькими периферийными узлами. Часто это смартфоны, планшеты или концентраторы IoT-сетей, которым нужны данные от множества сенсоров.
- **Broadcaster** — Устройство с односторонней рекламой: без установления соединения периодически рассылает рекламные пакеты с полезной информацией

(beacons) 4а. Используется для вещания маячков близости, рекламных кампаний, навигации в помещениях (Indoor positioning system).

- **Observer** — Стационарное устройство-приёмник, которое пассивно сканирует эфир, собирая (несвязанную) информацию из рекламных пакетов Broadcaster'ов и Peripheral'ов. Применяется для мониторинга окружающей среды, подсчёта посетителей, сбора статистики о маячках.

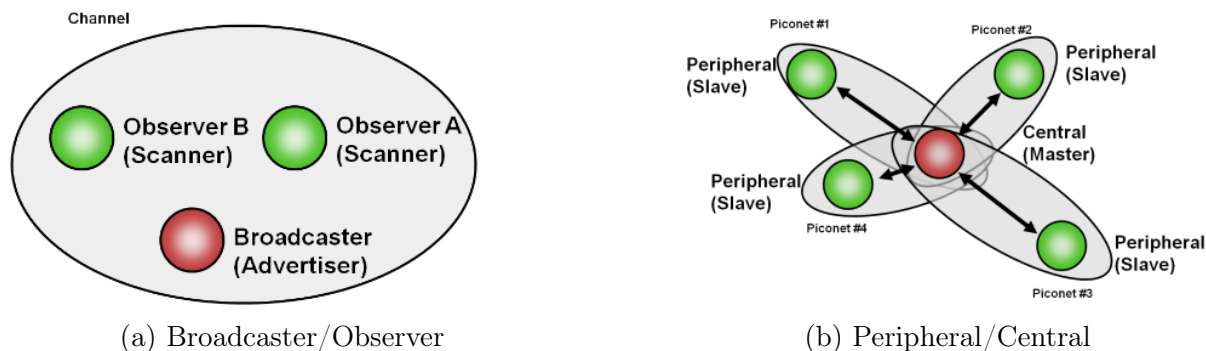


Рис. 4: Профили устройств BLE[17]

**Применимость в системах автоматизации и IoT.** BLE широко используется в умных домах, торговых и продуктовых центрах, медицине и иных сферах ИВ благодаря низкому энергопотреблению (десятки микроампер в спящем режиме) и широкой распространенности среди потребительских устройств.

**BLE Mesh.** Т.к. BLE Mesh лишь расширение стандарта, использующий особым образом режим рекламы для лавинообразного передачи сообщений, по техническим характеристикам он сильно похож на BLE. Для поддержания совместимости было использовано лавинообразное рассылка (flooding [8]), которая с большим количеством устройств может значительно увеличить территорию покрытия, но, с другой стороны, может привести к избыточной загрузке каналов, что после некоторой отметки в количестве устройств может привести к потерям пакетов.

### 2.2.3 Zigbee и Zigbee Pro

Стандарт ZigBee основан на IEEE 802.15.4 и работает в нелицензируемых ISM-диапазонах [10, 9]: 2.4ГГц(глобально доступен и используется во всем мире), 915МГц (США, Канада, Австралия и Израиль). 868МГц (ЕС, Великобритания, Турция)

Области применения включают умные дома и городскую инфраструктуру, где используются системы управления освещением, датчики движения, решения для безопасности и “умные” счётчики в составе городских систем управления; промышленную автоматизацию с круглосуточным мониторингом состояния оборудования, предиктивным обслуживанием, сбором данных с вибрационных и температурных датчиков, дистанционным управлением приводами и оптимизацией технологических процессов; и медицинский мониторинг для удалённого сбора жизненных показателей пациентов и контроля заряда портативных медицинских устройств.

## Основные элементы архитектуры и топологии

- **Узлы:**

- Координатор (ZC) единственный узел, создаёт PAN: сканирует каналы, выбирает свободный, задаёт PAN\_ID, Extended PAN\_ID, ключи безопасности, хранит таблицы маршрутизации и аутентификации. Работает также как маршрутизатор.
- Маршрутизатор (ZR) mains-powered, всегда активен, расширяет покрытие, ретранслирует пакеты, обеспечивает альтернативные пути.
- Конечное устройство (ZED) battery-powered, не ретранслирует, общается только с родителем (координатором или маршрутизатором), включает режим сна.

- **Топология:** ZigBee поддерживает три базовые топологии:

- Звёздная топология (Star) Все узлы подключены непосредственно к координатору. Простая настройка, но ограничена радиусом и зависит от координатора.
- Деревовидная топология (Cluster-Tree) Иерархия: координатор → маршрутизаторы уровней → конечные устройства. Адресация упрощена tree-адресацией, но узлы-маршрутизаторы становятся критическими точками.
- ячеистая топология (Mesh-network) Любой маршрутизатор может ретранслировать пакет между любыми двумя узлами. Поддерживает самовосстановление при отказах и динамическое подключение новых устройств.

Координатор задаётся программно администратором, но при запуске сети он автоматически:

- сканирует каналы ISM;
- выбирает наименее загруженный канал;
- присваивает PAN\_ID;
- задаёт параметры: NWK\_MAX\_CHILDREN, NWK\_MAX\_DEPTH, MAX\_ROUTERS.

В сетях IEEE 802.15.4 с beacon-режимом координатор через равные интервалы от 15 мс до 4 минут рассылает фрейм-маяк, который объявляет начало очередного суперфрейма и указывает доступные слоты доступа[10]; обычно именно координатор является главным источником этих маяков, хотя в древовидных (cluster-tree) конфигурациях роль ретрансляторов могут брать на себя маршрутизаторы, тогда как в обычном mesh-режиме они маяки не посылают; при этом автоматическое резервирование источников маяков не предусмотрено, поэтому отказ узла-маяка приводит к потере синхронизации всей сети, и её приходится вручную восстанавливать.

**Обнаружение маршрута** Процесс начинается с того, что узел-инициатор рассылает по сети широковещательный запрос на установление маршрута (RREQ), при этом указывается определённая метрика, например, количество переходов или качество канала. Узел назначения, получив несколько RREQ от разных маршрутов, выбирает из них наиболее подходящий — согласно метрике — и отправляет одноадресный ответ (RREP) обратно к источнику, используя обратный путь, по которому был получен лучший запрос.

**Передача и ретрансляция** После получения ответа (RREP) начинается передача данных. Информация передаётся от источника к получателю через промежуточные узлы. Каждый промежуточный узел проверяет свою таблицу маршрутизации, чтобы определить, куда направить пакет дальше, и ретранслирует пакет следующему узлу по цепочке. Если ожидаемое подтверждение доставки (ACK) не поступает, инициируется сообщение об ошибке маршрута (RERR). В таком случае начинается повторное построение маршрута — запускается новый процесс рассылки RREQ.

**Шаблоны устройств (Application Profiles)** На уровне приложения Zigbee определяет профили устройств:

- **Публичные профили:** Home Automation, Commercial Building Automation и др.: лампы, выключатели, датчики, термостаты, розетки.
- **Частные профили:** Производители могут создавать собственные профили, но контроллер должен поддерживать их для взаимодействия.

Каждый профиль включает: Profile ID, Device ID, Cluster IDs и назначение энптойтов (адрес и адресат). Контроллер распознаёт и управляет устройством только при совпадении этих идентификаторов.

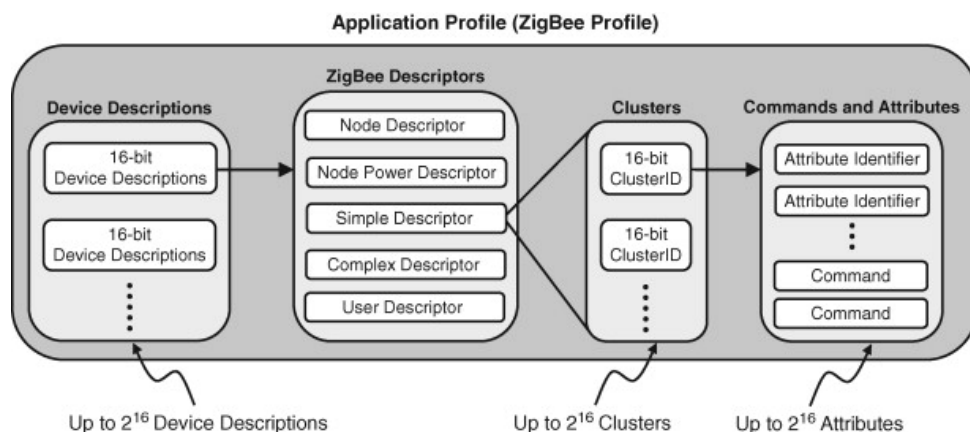


Рис. 5: Профили устройств Zigbee [18]

## Влияние отказов

- **Координатор:** при его отказе сеть не функционирует, требует перезапуска с новым координатором [11].
- **Маршрутизатор:** потеря альтернативных путей, временная деградация доступности узлов, последующее самовосстановление [11].
- **Конечное устройство:** затрагивается только функционал этого узла, остальная сеть продолжает работу.

**Zigbee Pro.** Zigbee Pro представляет собой расширенную реализацию базового протокола Zigbee, оптимизированную для высоконагруженных и масштабируемых сетей [13]. В отличие от классического Zigbee, Про-версия обеспечивает поддержку сетей размером до нескольких тысяч узлов, более гибкую и самовосстанавливающуюся маршрутизацию, улучшенную управляющую функциональность (Many-to-One и Source Routing) и усиленные механизмы безопасности с отдельными ключами для транспортного и сетевого уровней.

В случае отказа координатора или маршрутизатора, сеть на базе Zigbee Pro демонстрирует лучшие возможности по сохранению связи и быстрому восстановлению работоспособности [11] благодаря:

- **Самовосстановлению маршрутов:** оставшиеся маршрутизаторы автоматически перестраивают таблицы маршрутов без участия центрального узла.
- **Source Routing и Many-to-One маршруты:** устройства могут кэшировать рабочие пути и переключаться на альтернативные узлы при недоступности основного, что снижает время простоя.
- **Гибкая архитектура Trust Center:** в Про-режиме можно реализовать избыточные или внешние Trust Center (через облачные сервера или резервные координаторы), что позволяет продолжать авторизацию и обновление ключей даже при выходе из строя основного координатора.

Таким образом, выбор Zigbee Pro вместо «простого» Zigbee особенно оправдан в критичных системах, где недопустима потеря управляемости и долговременная деградация сети при отказах ключевых узлов.

## 2.2.4 Сравнение представленных протоколов

На рис. 6 представлен сравнительный график различных протоколов беспроводных сетей. Была составлена таблица 1.

Wi-Fi Mesh (IEEE 802.11s) и BLE Mesh работают полностью без единого сетевого координатора: в Wi-Fi Mesh каждый узел (Mesh Point) самостоятельно участвует в маршрутизации по протоколу HWMP, а BLE Mesh применяет управляемое наводнение без роли координатора доставки. Zigbee же требует наличия PAN Coordinator'a, который создаёт и управляет сетью, выдавая адреса устройствам-роутерам. Z-Wave полагается на Primary Controller (и Static Update Controller) для добавления узлов и хранения топологии, без которого сама сеть не синхронизируется. Thread формально выбирает «лидера», но все Router-узлы равноправны: нет жёсткого единственного координатора, а маршруты поддерживаются децентрализованно через MLE-сообщения. При этом большинство протоколов (Wi-Fi Mesh, BLE Mesh, Zigbee, Thread) работают в перенаселённой полосе 2.4 GHz – зона 2.4 GHz страдает высоким уровнем помех и зашумлённостью каналов, тогда как Z-Wave (868/915 MHz) избегает этой проблемы.

Для построения распределённой самовосстанавливающейся сети без единой точки отказа и с наименьшей задержкой, лучше выбирать Wi-Fi Mesh или BLE Mesh, хотя они не лишены компромисов: нуждаются в широкой полосе и часто сталкиваются с помехами на частоте 2.4 ГГц. Для снижения влияния радио-шумов 2.4 ГГц и повышения надёжности на больших расстояниях предпочтительнее Z-Wave на суб-ГГц, пусть и с центральным контроллером.

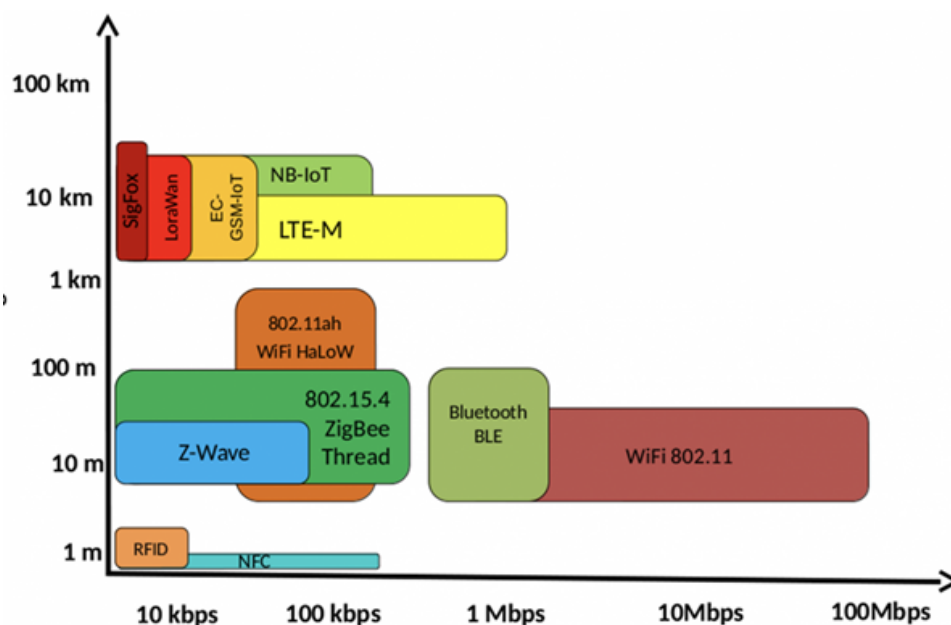


Рис. 6: Сравнение беспроводных протоколов передачи данных по дальности и скорости передачи. [14]

Имя	Пропускная способность	Дальность	Энергопотребление	Протокол маршрутизации
WiFi Mesh	до 1,300 Мбит/с	до 100 м в помещении и до 300 м на открытом пространстве	от нескольких мВт до нескольких сотен мВт	HWMP с возможностью изменения
BLE Mesh	до 125 Кбит/с	до 100 м в помещении и до 300 м на открытой местности	от нескольких мкВт до нескольких мВт	Flooding
Z-wave	до 100 Кбит/с	до 100 м в помещении и до 300 м на открытой местности	от нескольких мкВт до нескольких мВт	Закрытый протокол
Zigbee	от 20 до 250 Кбит/с	от 10 до 100 м в помещении и до 400 м на открытом пространстве	от нескольких мкВт до нескольких мВт	AODV с возможностью изменения

Таблица 1: Сравнение беспроводных mesh-технологий

### 2.2.5 Доступность для разработчиков платформ, поддерживающих представленные протоколы

В среде Arduino IDE наиболее зрелая поддержка реализована для плат серии ESP32 DevKitC на базе модуля ESP-WROOM-32: доступен API для построения Wi-Fi Mesh, однако готовых примеров в составе «arduino-esp32» нет, а поддержка BLE Mesh на уровне Arduino отсутствует, требуя прямой работы с ESP-IDF. Платы ESP32-C6 и ESP32-H2 оснащены аппаратным радио IEEE 802.15.4 (Zigbee/Thread), но в текущих версиях core для Arduino-ESP32 их поддержка нестабильна и не включает демонстрационных примеров; для полноценной работы требуется использовать свежие ветки «arduino-esp32» или компилировать примеры из ESP-IDF.

Со стороны STMicroelectronics наиболее популярна плата P-NUCLEO-WB55RG (Nucleo-WB55RG) на базе STM32WB55, где через STM32duino core и библиотеку STM32duinoBLE доступны примеры по BLE, но для Thread и Zigbee необходимо обращаться к STM32CubeWB SDK и демонстрациям на сайте производителя. Аналогично, плата STM32W-RFCKIT поддерживается лишь в рамках STM32CubeWB и не имеет официальных Arduino-библиотек или готовых sketch'ей.

## 3 Глава 2

### 3.1 Описание системы

Централизованные СДА, имеющие ЦУД, обладая несомненными преимуществами, такими как простота настройки и реализации, также имеют ряд недостатков: низкая скорость реакции, невысокая надежность и безопасность, сильно зависящая от ЦУД и облачной инфраструктуры. Для интеграции устройств, находящихся в разных сетях, используется централизованная координация через единое облако. На рис. 7 представлена архитектура предлагаемой гибридной системы, подразумевающей прямое взаимодействие устройств в СДА.

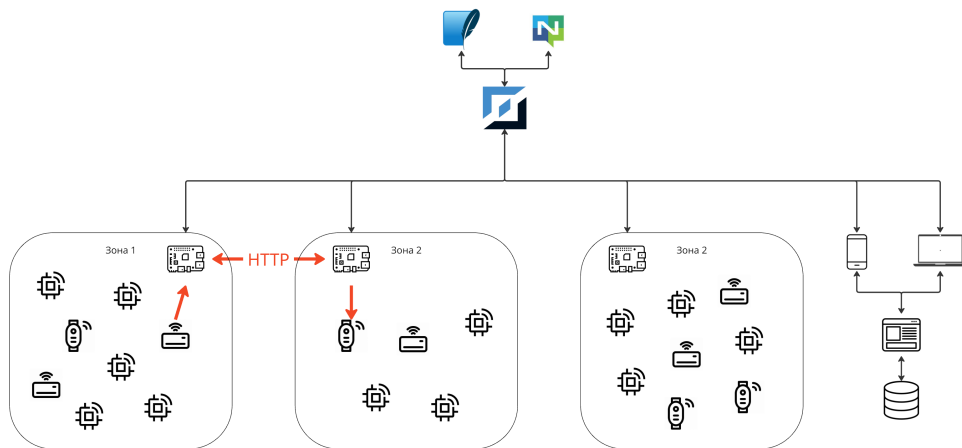


Рис. 7: Пример СДА с распределенной структурой

Для взаимодействия физически разнесенных устройств предлагается использование шлюзов-ретрансляторов, которые выполняют функцию объединения зон в единую сеть, распределяют по устройствам обновленную конфигурацию сети, а также предоставляют функции управления умным домом извне. Предложенная система уменьшает количество промежуточных узлов между устройством, порождающим событие, и устройством исполнителем (при нахождении устройств в одной зоне до нуля). Данное решение кроме повышения надежности системы также уменьшает задержку передачи информации, повышает отказоустойчивость системы и ее безопасность [5, 6].



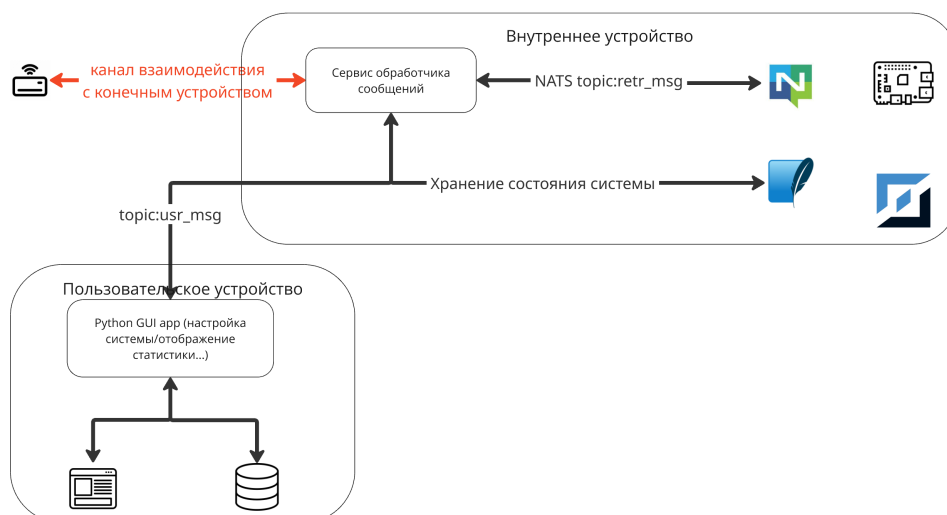


Рис. 8: Компонентная структура и процессы взаимодействия в кластере K0s предложенной системы

Шлюзы-ретрансляторы – кластер K0s – при выходе из строя какого-то узла, например, зоны К, перестанут обрабатываться межзональные события, связанные с зоной К. При этом функционал внутри и вне зоны К не изменится.

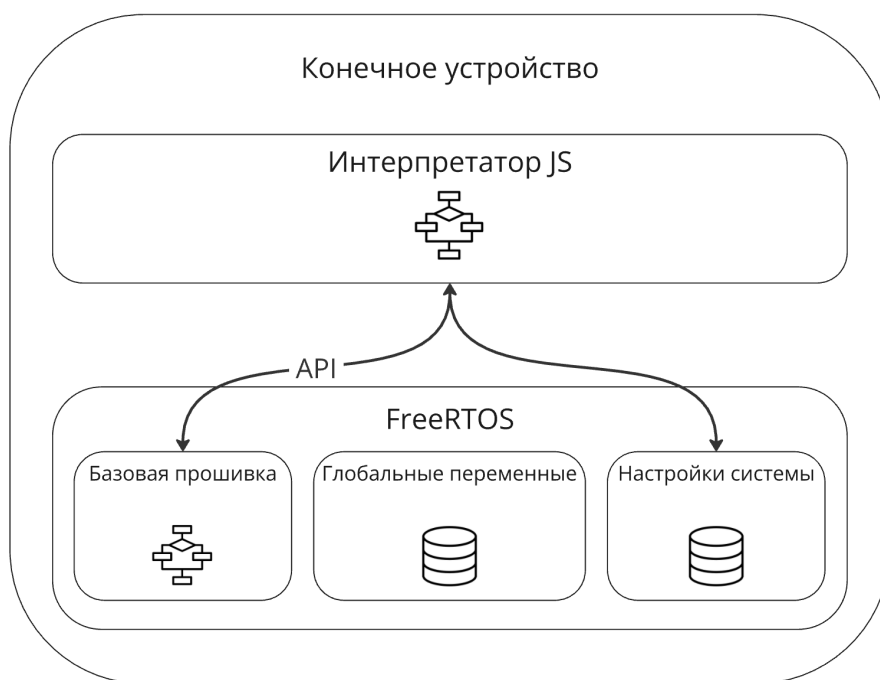


Рис. 9: Реализация конечного устройства

Програмная часть каждого конечного устройства представляет из себя двухуровневую систему, как показано на рис. 9: низкоуровневую прошивку, написанную с использованием FreeRTOS и предоставляющую API для высокоуровневой части для управления, и высокоуровневую прошивку, написанную на языке JavaScript и запускающуюся посредством интерпретатора на конечном устройстве. JS часть может

быть легко изменена извне через настройку системы. Такой подход позволяет вынести логику взаимодействия из ЦУД в конечные устройства и передавать данные напрямую и сразу локально исполнять сценарии. Во время прототипирования были использованы различные протоколы передачи информации, в итоге были использованы наиболее гибкие в настройке радиомодули SI4432.

## 3.2 Выбранная топология системы

Для передачи сообщений между конечными устройствами описываемая СДА использует радиомодули, взаимодействующие между собой посредством широковещательной передачи (broadcast), при которой каждое сообщение может быть принято любым узлом в зоне радиодоступа. Сигнал принимают все модули, но обрабатывают его лишь те, чей фильтр (например, по адресу или типу пакета) его пропускает. С точки зрения классических сетевых определений, говорить о чёткой топологии здесь затруднительно: физического соединения между узлами нет, а сама передача осуществляется через общую радиосреду. Такая структура ближе всего к логической топологии типа "шина" (bus), поскольку сигнал распространяется по общему каналу связи и доступен всем участникам сети. Однако корректнее описывать такую организацию как сеть с разделяемой средой доступа (shared medium) и логикой point-to-multipoint, где один узел может потенциально вещать всем, но приём зависит от контекста.

## 3.3 Взаимодействие устройств

Для связи устройств в рамках сети предлагается использование протокола для обмена сообщений по шаблону «наблюдатель» (Observer). В котором каждое устройство сверяет полученное сообщение со списком реакций на оповещения, и, если оповещение есть в списке, запускает выполнение команды. Наглядно демонстрацию отличий Observer от Pub-Sub можно увидеть на рис. 10.

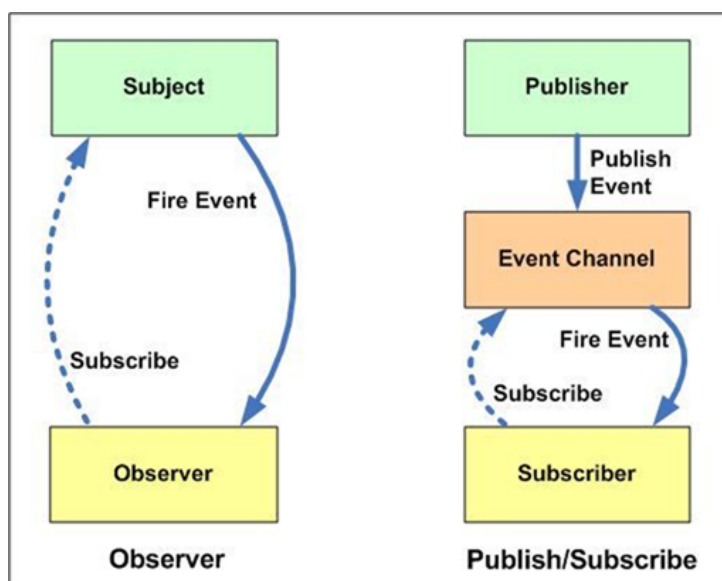


Рис. 10: Отличие observer от Publisher-Subscriber [16]

### 3.4 Добавление устройств и настройка сценариев

Система домашней автоматизации состоит из конечных устройств, связанных по радиоканалу, и шлюзов-ретрансляторов, представляющих собой одноплатные компьютеры с установленными сервисами **k0s**. Настройка системы выполняется через единое графическое приложение, содержащее кнопку для начала настройки, поле для изменения параметров и кнопку для применения изменений.

#### 1. Перевод системы в режим настройки.

Пользователь нажимает кнопку «Начать настройку системы» в графическом интерфейсе. Приложение отправляет управляющую команду на все шлюзы-ретрансляторы, которые, в свою очередь, передают её конечным устройствам. Устройства переходят из режима эксплуатации в режим настройки и готовы к приёму новой конфигурации.

#### 2. Внесение и отправка изменений.

Через предоставленное текстовое или графическое поле в приложении вносятся необходимые изменения конфигурации. Это могут быть новые сценарии взаимодействия, логика работы исполнительных устройств и параметры ретрансляции сообщений. После завершения редактирования пользователь нажимает кнопку «Применить настройки», и новая конфигурация передаётся на шлюзы-ретрансляторы.

#### 3. Распространение конфигурации на конечные устройства.

Шлюзы-ретрансляторы получают обновлённую конфигурацию и передают её по радиоканалу на исполнительные устройства. Датчики не требуют настройки, так как они только передают сообщения. Исполнительные устройства анализируют входящие сигналы и запускают соответствующие действия по заданным сценариям. После успешного применения настроек система может быть переведена обратно в режим работы тем же интерфейсом.

### 3.5 Обработка событий

Каждое устройство-исполнитель хранит набор сценариев. Сценарии связаны с событиями. При выполнении события устройство сверяет события-активаторы, соответствующие сценариям, и, если события совпали, начинает выполнение соответствующего сценария. Сценарии представляют собой фрагменты кода JS, интерпретируемые на конечных устройствах. Их передача осуществляется в виде текстового формата JSON. Происходит их поэтапное выполнение.

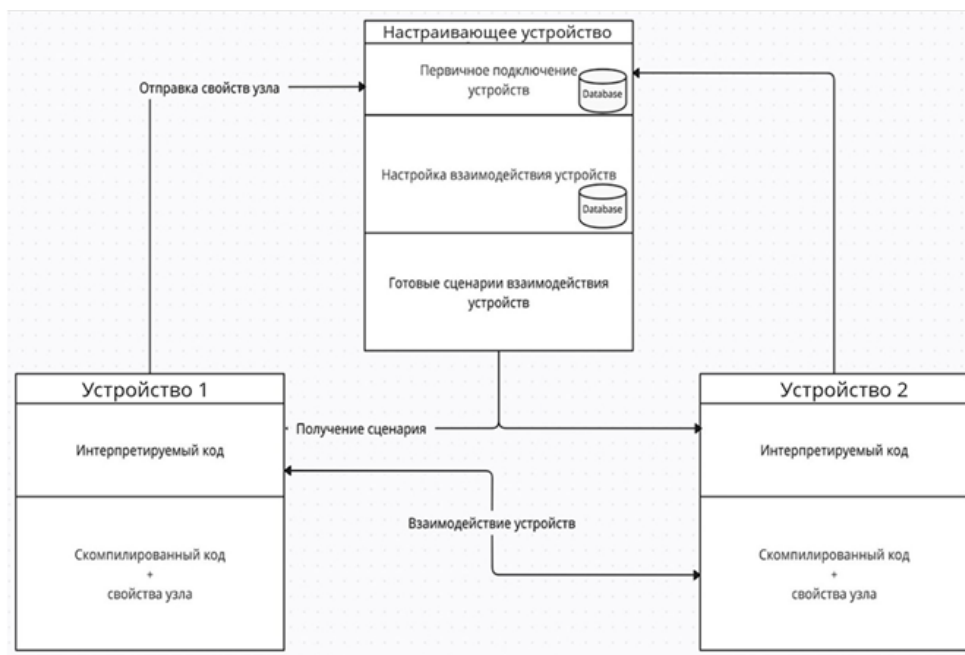


Рис. 11: Пример взаимодействий конечных устройств

## 4 Глава 3

### 4.1 Графический интерфейс

Было написано графическое приложение рис. 12 на языке Python с использованием фреймворка Qt, в котором производилась настройка системы и ее отладка.

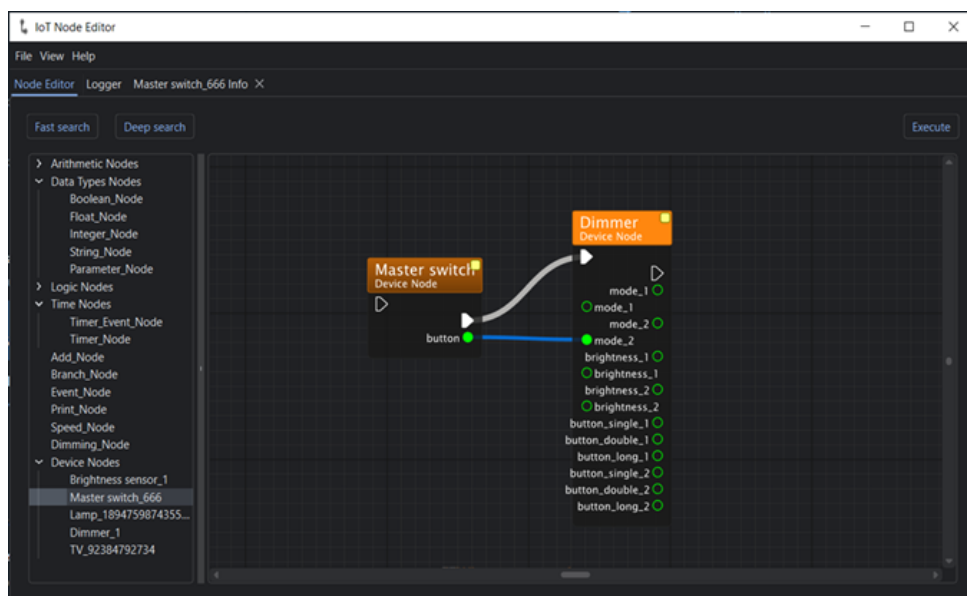


Рис. 12: Графическое приложение с созданным внутри сценарием для конечных устройств

Графический интерфейс позволяет прописывать и более сложные сценарии взаимодействия устройств, что можно увидеть на рис. 13

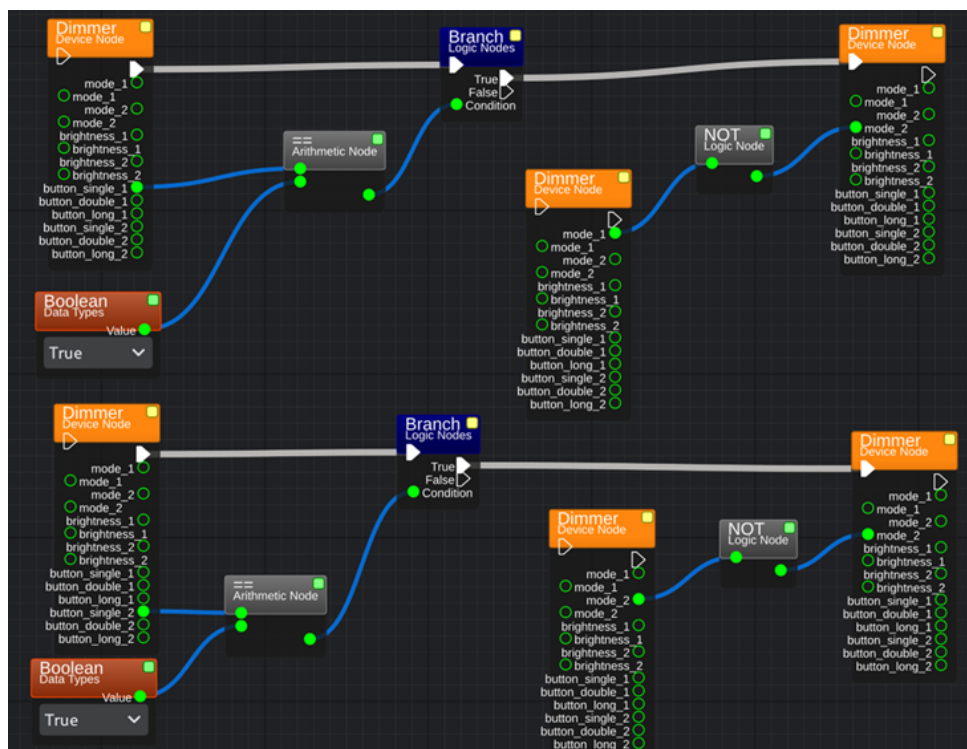


Рис. 13: Пример более сложного сценария взаимодействия устройств

## 4.2 Планировщик

Было опробовано два варианта распределения сценариев: их хранение и исполнение на устройстве-отправителе, порождающем событие и хранение сценариев на устройстве-исполнителе. В рамках проведенных экспериментов было установлено, что некоторые сложные команды, последовательный набор действий для первого варианта проводят к значительному увеличению времени обработки и излишней загроможденности радиоканала, что потенциально уменьшает максимальное число устройств в системе. Планировщик осуществляет преобразование визуальных узлов и связей в JS код, выполняя синтаксический анализ, преобразованием узлов в лексемы с последующим объединением их в единый исполняемый файл. Так же в задачи планировщика входит разбиение файла настроек на компоненты связанности и определение целевого устройства, на которое будет отправлен сценарий. Также для дальнейшего тестирования система была сконфигурирована для работы с использованием ЦУД и для распределенной работы.

## 5 Глава 4

### 5.1 Тестирование и полученные оценки

Для корректности тестирования и сравнимости подходов, ввиду отличия протоколов передачи данных, использования разных микроконтроллеров в существующих системах и иных отличий, исходная система была переконфигурирована для работы как с ЦУД, так и без ЦУД. Таким образом, проведя замеры задержек от возникновения события до реакции, можно оценить численно преимущество предложенной архитектуры. В таблице приведены результаты замеров на каждом этапе: время передачи пакета сообщения, время обработки события для каждой системы, что позволило использовать эти данные без привязки к конкретной реализации. Были проведены замеры времени задержек выполнения программы на микроконтроллере STM32F411 (25МГц), выступавшим в качестве конечного устройства, и одноплатным компьютером Raspberry pi 3b+. Тестирование одноплатного компьютера проводилось на системе Raspberry Pi OS 19.11, базирующейся на Debian 12. Использовался GCC 15 с опциями: «-mcpu=native, -O2». При тестировании микроконтроллеров использовалась STM32duino FreeRTOS v10.3.2 с библиотекой RadioLib v7.1.2. Радиомодули, представленные SI4432, использовались со стандартными параметрами.

п/п	Тип операции	Время операции
1	Передача в рамках одной зоны	197 мс
2	Выполнение 1 примитива интерпретатора на микроконтроллере	18 мс
3	Выполнение 1 примитива интерпретатора на одноплатном компьютере	0,24 мс
4	Общая задержка системы при использовании ЦУД	407 мс
5	Общая задержка системы без использования ЦУД	231 мс
6	Задержка СУД «Умный дом с Алисой» от Яндекс	~470 мс

Таблица 2: Среднее время выполнения операций

Использовалось готовое решение на базе протокола Zigbee с ЦУД в виде YNDX-00510 [7] и лампой YNDX-00558. Лучшим способом сокращения задержек в СУД является уменьшение задержек в передаче информации. Предложенный выше подход позволяет почти в вдвое сократить задержку (в рамках тестирования спроектированной СУД) и более чем втрое по сравнению с представленными на рынке решениями.

### 5.2 Вывод

Предложен подход к построению распределенной системы домашней автоматизации, проведено сравнение с решением, представленным на рынке. В дальнейшем предполагается исследовать возможные уязвимости данной системы, построить и протестировать прототип распределенной системы с использованием шлюзов-ретрансляторов, объединенных в единую сеть посредством VPN.

## 6 Список литературы

- [1] Колыбельников А. И. Обзор технологий беспроводных сетей // ТРУДЫ МФТИ. 2012. Т. 4. № 2 С. 3–29.
- [2] Шерчков А. В. Анализ существующих технических решений «Умный дом» // Молодежный вестник Уфимского государственного авиационного технического университета. 2020. № 1 С. 153–155.
- [3] Montori F., Contigiani R., Bedogni L., Is WiFi suitable for energy efficient IoT deployments? A performance study // 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI). 2017. pp. 1-5, doi: 10.1109/RTSI.2017.8065943.
- [4] Тукмачева Ю. А. Обзор и анализ автоматизированных систем «Умный дом», представленных на российском сегменте рынка // Вестник науки. 2022. Т. 4. № 10 С. 139–144.
- [5] Некрасов П. В., Жариков А. М., Козин Д. А. Обеспечение безопасности беспроводных каналов связи киберфизических систем типа «Умный дом» // Безопасность информационных технологий. 2024. Т. 31. № 1 С. 54–62.
- [6] Vardakis G., Hatzivasilis G., Koutsaki E., Papadakis N. Review of smart-home security using the internet of things // Electronics. 2024. No 13 P. 33–43.
- [7] Центр умного дома «Яндекс Хаб» модель YNDX-00510. Инструкция по эксплуатации [Электронный ресурс]: URL:<https://yastatic.net/s3/doc-binary/src/support/smart-home/hub/hub-manual.pdf> (дата обращения: 17.03.2025).
- [8] Specifications and Documents Mesh Protocol [Электронный ресурс]: URL:[https://www.bluetooth.com/specifications/specs/html/?src=MshPRT\\_v1.1/out/en/index-en.html](https://www.bluetooth.com/specifications/specs/html/?src=MshPRT_v1.1/out/en/index-en.html) (дата обращения: 11.05.2025).
- [9] Zigbee Specification Revision 23 [Электронный ресурс]: <https://csa-iot.org/wp-content/uploads/2023/04/05-3474-23-csg-zigbee-specification-compressed.pdf> (дата обращения: 20.05.2025).
- [10] IEEE Std 802.15.4-2020: IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs) [Электронный ресурс]: [https://standards.ieee.org/standard/802\\_15\\_4-2020.html](https://standards.ieee.org/standard/802_15_4-2020.html) (дата обращения: 20.05.2025).
- [11] Richard Alena, Ray Gilstrap, Jarren Baldwin, Thom Stone, Pete Wilson. Fault Tolerance in ZigBee Wireless Sensor Networks // NASA Ames Research Center, Moffett Field [Электронный ресурс]: URL:[https://ntrs.nasa.gov/api/citations/20110012426/downloads/20110012426.pdf?utm\\_source=chatgpt.com](https://ntrs.nasa.gov/api/citations/20110012426/downloads/20110012426.pdf?utm_source=chatgpt.com) (дата обращения: 20.05.2025).
- [12] ZigBee Cluster Library Specification (07-5123-08), Connectivity Standards Alliance [Электронный ресурс]: <https://zigbeealliance.org/wp-content/uploads/2021/10/07-5123-08-Zigbee-Cluster-Library.pdf> (дата обращения: 20.05.2025).

- [13] Zigbee Pro Stack User Guide (JN-UG-3101), NXP Semiconductors [Электронный ресурс]: <https://www.nxp.com/docs/en/user-guide/JN-UG-3101.pdf> (дата обращения: 20.05.2025).
- [14] Сравнительный анализ стандартов связи для сетей IoT [Электронный ресурс]: URL:<https://habr.com/ru/companies/msw/articles/720518/> (дата обращения: 11.05.2025).
- [15] Wi-Fi HaLow™ and LoRaWAN: How do the technologies compare? [Электронный ресурс]: URL:<https://dev.wi-fi.org/zh-hans/beacon/y-zachary-freeman/wi-fi-halow-and-lorawan-how-do-the-technologies-compare> (дата обращения: 11.05.2025).
- [16] Observer vs Pub-Sub [Электронный ресурс]: URL:<https://habr.com/ru/articles/270339/> (дата обращения: 11.05.2025).
- [17] Generic Access Profile (GAP) Roles [Электронный ресурс]: URL:<https://developerhelp.microchip.com/xwiki/bin/view/applications/ble/introduction/bluetooth-architecture/bluetooth-host-layer/gap/roles/> (дата обращения: 19.05.2025).
- [18] Bachelor Seminar Paper [Электронный ресурс]: URL:[https://www.auto.tuwien.ac.at/bib/pdf\\_TR/TR0148.pdf](https://www.auto.tuwien.ac.at/bib/pdf_TR/TR0148.pdf) (дата обращения: 20.05.2025).



## Приложение А. Исходные коды и дополнительные материалы

1. Solovjev D. Distributed smart home cluster [Электронный ресурс]: URL:<https://github.com/Dmitrij-Solovjev/distributed-smart-home-cluster> (дата обращения: 11.05.2025).  
Примечание: Кластер на базе K0s с сервисами для ретрансляции сообщений и общей настройки системы.
2. Solovjev D. IoT Base Device [Электронный ресурс]: URL:<https://github.com/Dmitrij-Solovjev/IoT-Base-Device> (дата обращения: 11.05.2025).  
Примечание: Прошивка для микроконтроллера на FreeRTOS, реализующая логику домашней автоматизации.