

Министерство образования и науки Российской Федерации  
Санкт-Петербургский государственный университет  
Факультет прикладной математики и процессов управления

**Научно исследовательская работа по теме:**  
Система домашней автоматизации с использованием  
децентрализованного подхода в хранении и обработке  
данных.

Автор работы	<div></div> <div>подпись</div>	<div>Соловьев Д.Н.</div> <div>фамилия и инициалы</div>
Научный руководитель	<div></div> <div>подпись</div>	<div>Корхов В.В.</div> <div>фамилия и инициалы</div>
Председатель комиссии	<div></div> <div>подпись</div>	<div></div> <div>фамилия и инициалы</div>

## 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.

# **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 Цель работы**

Целью работы является исследование систем представленных на рынке, изучение имеющихся протоколов, выбор оптимального, разработка прототипа системы.

## **1.5 Научная новизна и практическая значимость**

Уменьшение задержек, расширение сфер применения: в предлагаемой системе уменьшается количество узлов между устройством, порождающим событие, и устройством исполнителем (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 в случае восстановления системы после сбоя из-за внештатной ситуации, например отключение электроэнергии, сразу возобновить работу системы без ожидания синхронизации состояния. Упрощение настройки системы: так как для настройки системы используется планировщик, распределяющий сценарии по устройствам, нет необходимости настраивать каждое устройство по-отдельности. Кроме того, повышается отказоустойчивость системы: в рамках одной локации невыполнение сценария возможно при неисправности только одного из двух устройств: устройства-инициатора или устройства-исполнителя. Повышается безопасность системы: распределенный подход для хранения сценариев

и управления устройствами минимизирует получение злоумышленниками доступа ко всем устройствам. Объединение локаций, а также дистанционный мониторинг и управление системой осуществляется посредством шлюзов-ретрансляторов с использованием VPN соединения, что избавляет от единой точки отказа системы (точки входа) и Нижеизложенный прототип может быть использован для построения адаптивной системы изменения освещения для подстройки под потребляемый пользователем контент для большего эффекта погружения (использование осветительных приборов в качестве фонового освещения). Так же данная система ввиду уменьшения задержек по сравнению с аналогами может быть использована для предсказания положения пользователя в умном здании и подстройки окружающего пространства под нужды пользователя с минимальной задержкой. Например, предсказываемое изменение состояния устройств, находящихся в одной локации с пользователем. В качестве сфер применения можно рассматривать не только системы домашней автоматизации, но и системы умных предприятий. Так данная система будет востребована на участках, связанных с риском для жизни, где использование проводов не представляется возможным, либо излишне затруднит построение “умного” предприятия. Например, металлургические предприятия.

## 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.

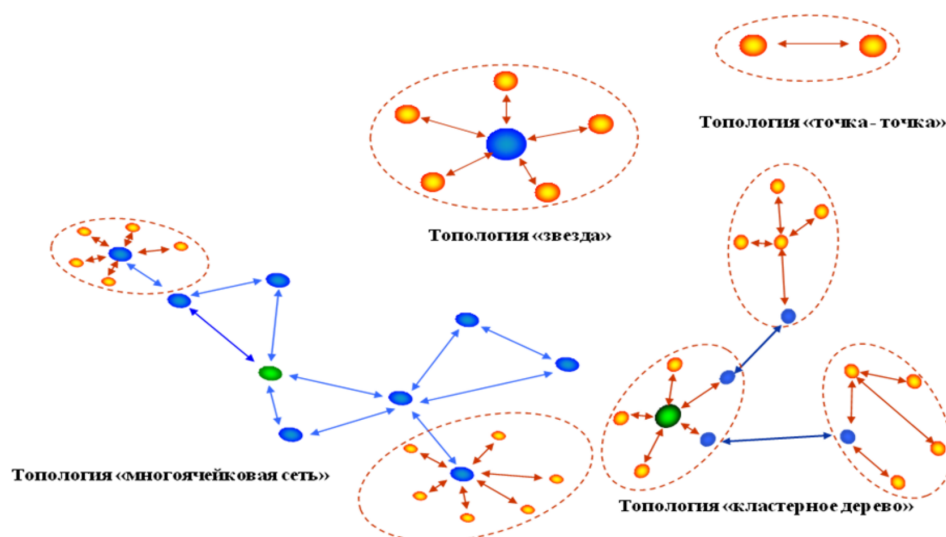


Рис. 1: Топологии беспроводных сетей

#### 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 Топология «многоячейковая сеть» (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 Краткий обзор имеющихся альтернатив

В данный момент большинство производителей ИВ используют централизованный подход для построения СДА [1]. 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 предпочтителен, так как повышается отзывчивость системы и ее отказоустойчивость. Отдельную сложность представляет объединение устройств из разных экосистем: различные комбинации связей «устройство», «ЦУД», «облако».

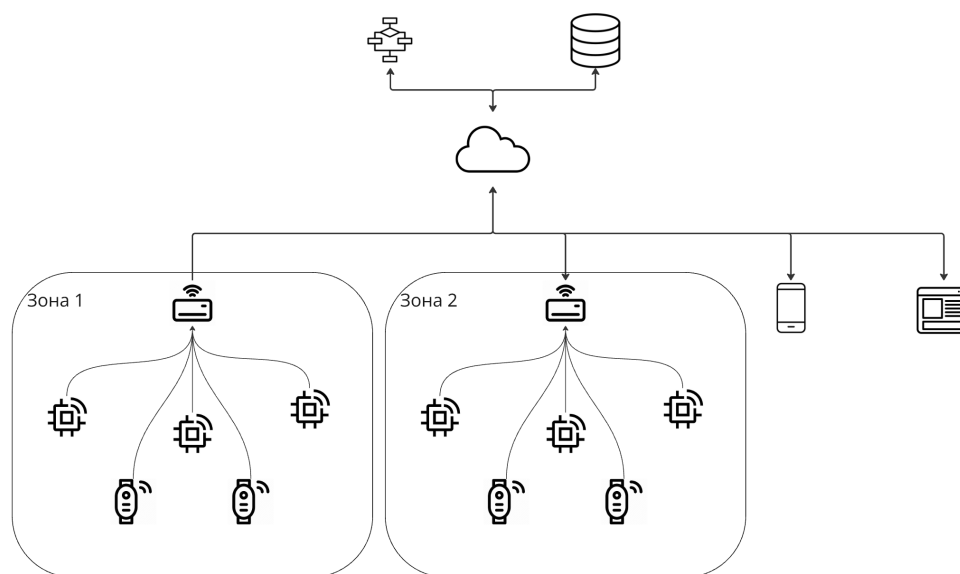


Рис. 2: Пример СДА с облачной обработкой событий

Большинство имеющихся на данный момент решений опираются на топологию «дерево», где в качестве корня рассматривается ЦУД или облако, что проиллюстрировано на рис. 2. При этом структура может подразумевать наличие единственных ветвей из корня, например «облако-ЦУД-устройства» [2]. 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 Mesh**

Протокол беспроводной передачи данных, основанный на IEEE 802.11, работающий на частоте 2.4 ГГц (802.11b), 5 ГГц (IEEE 802.11ac), 6 ГГц (IEEE 802.11ah). Ячеистая топология доступна в 802.11s. Доступны различные протоколы маршрутизации в рамках ячеистой топологии [19], но имеет требование наличия HWMP [18] в качестве базового протокола. Устройства, использующие данный стандарт, как правило представляют собой систему WLAN расширения зон покрытия, в которых роутеры ретранслируют сигнал для увеличения дальности. Протокол мало пригоден для построения систем домашней автоматизации ввиду высокого потребления электроэнергии.

### **2.2.2 Bluetooth Low Energy (BLE) и Bluetooth 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[22]), 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. Так же метод является неоптимальным при небольшом количестве устройств ввиду счетчика TTL который на каждом шаге уменьшается и может обнулиться, не дойдя до адресата [20].

### **2.2.3 Zigbee**

Протокол, основанный на IEEE 802.15.4, описывающий построение беспроводной персональной сети. Спецификация Zigbee ориентирована на приложения, требующие гарантированной безопасной передачи данных при относительно небольших скоростях и возможности длительной работы сетевых устройств от автономных источников питания (батарей). Основная особенность технологии Zigbee заключается в том, что она при малом энергопотреблении поддерживает не только простые топологии сети («точка-точка», «дерево» и «звезда»), но и самоорганизующуюся и самовосстанавливающуюся ячеистую (mesh) топологию с ретрансляцией и маршрутизацией сообщений. Кроме того, спецификация Zigbee содержит возможность выбора алгоритма маршрутизации в зависимости от требований приложения и состояния сети. Основными областями применения технологии Zigbee являются беспроводные сенсорные сети, автоматизация жилья («Умный дом» и «Интеллектуальное здание»), медицинское оборудование, системы промышленного мониторинга и управления, а также бытовая электроника и «периферия» персональных компьютеров. Способность к самоорганизации и самовосстановлению, ячеистая (mesh-) топология, защищенность, высокая помехоустойчивость, низкое энергопотребление и отсутствие необходимости

получения частотного разрешения делают Zigbee-сеть подходящей основой для беспроводной инфраструктуры систем позиционирования в режиме реального времени (RTLS). В России, как и в большинстве стран мира за исключением США, ЕС и Австралии разрешенная частота составляет 2.4 ГГц, которая является общей для других технологий, таких как WiFi и Bluetooth, что может привести к избыточной загруженности каналов и большой потере сообщений.

#### 2.2.4 Сравнение представленных протоколов

Был составлен сравнительный график различных протоколов передачи сообщений.

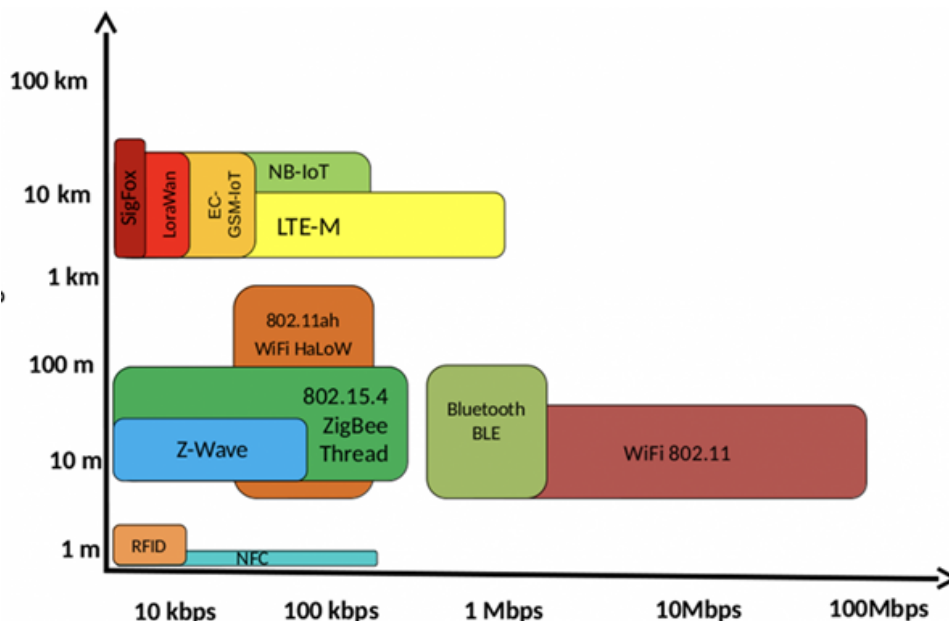


Рис. 3: Сравнение беспроводных протоколов передачи данных по дальности и скорости передачи.

Так же была составлена сравнительная таблица. 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 распределенной 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 2.4ГГц. Для снижения влияния радио-шумов 2.4ГГц и повышения надёжности на больших расстояниях предпочтительнее Z-Wave на суб-ГГц,



пусть и с центральным контроллером.

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 Постановка задачи

После анализа существующих решений и выявления недостатков СДА, присутствующих на рынке, были сформулированы начальные условия (требования).

1. Использование одноуровневой сети для связи устройств в рамках единой зоны.
2. Взаимодействие устройств осуществляется напрямую, без посредников в виде ЦУД.
3. Взаимодействие устройств из разных зон обеспечивается благодаря шлюзам-ретрансляторам, объединенными в единую сеть.

### 3.2 Краткое описание предлагаемой 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. 4 представлена архитектура предлагаемой гибридной системы, подразумевающей прямое взаимодействие устройств в СДА.

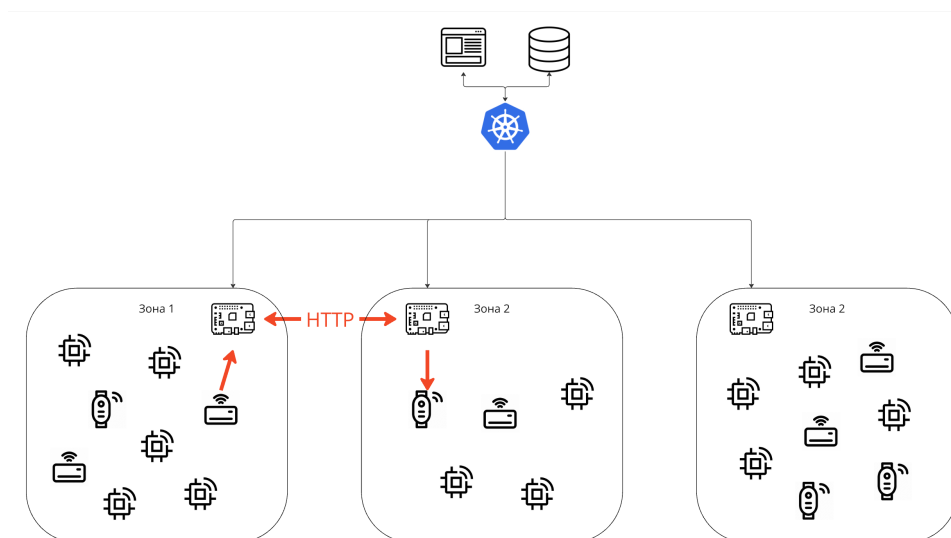


Рис. 4: Пример СДА с распределенной структурой

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 функцию 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 [4, 5].

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.

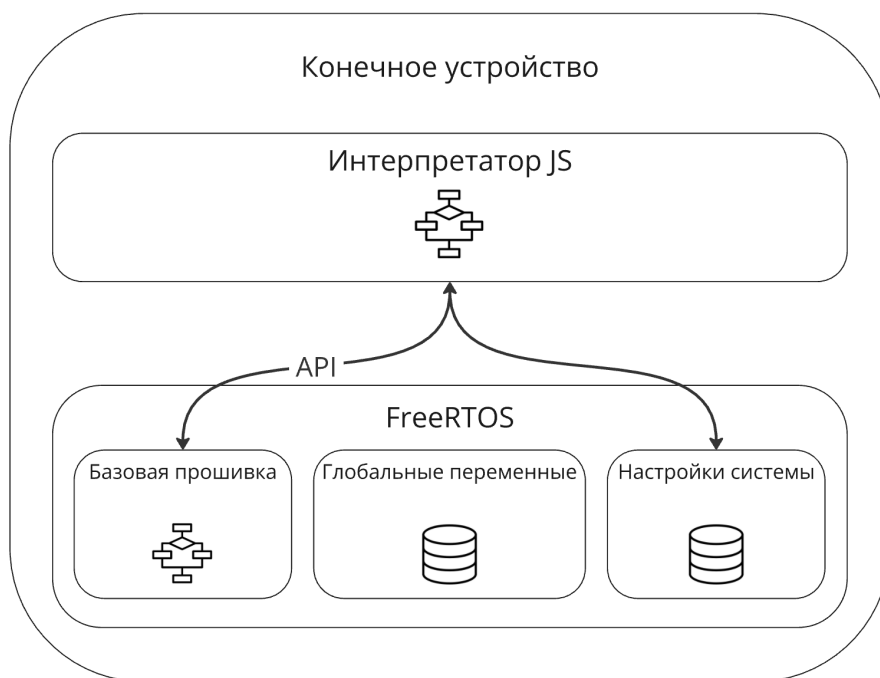


Рис. 5: Пример реализации конечного устройства

Программная часть каждого конечного устройства представляет из себя двухуровневую систему, как показано на рис. 5: 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.

### 3.3 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 четкой 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.

### 3.4 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. 7.

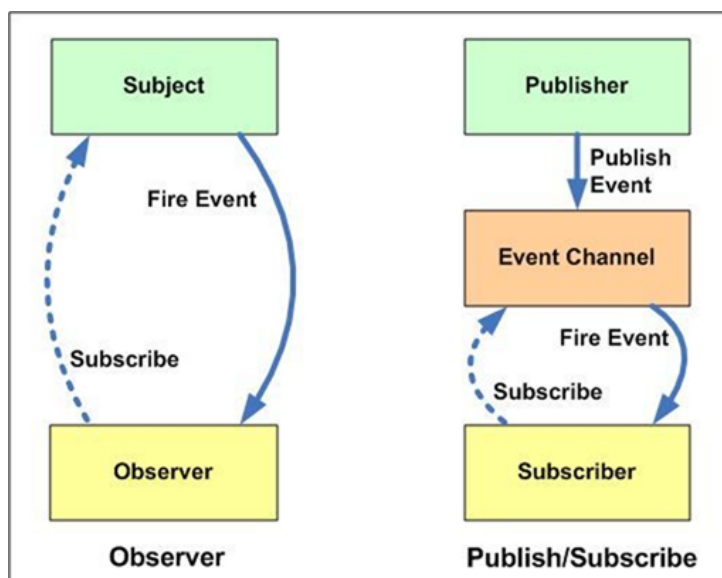


Рис. 6: Пример реализации конечного устройства

### 3.5 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.

### 3.6 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.

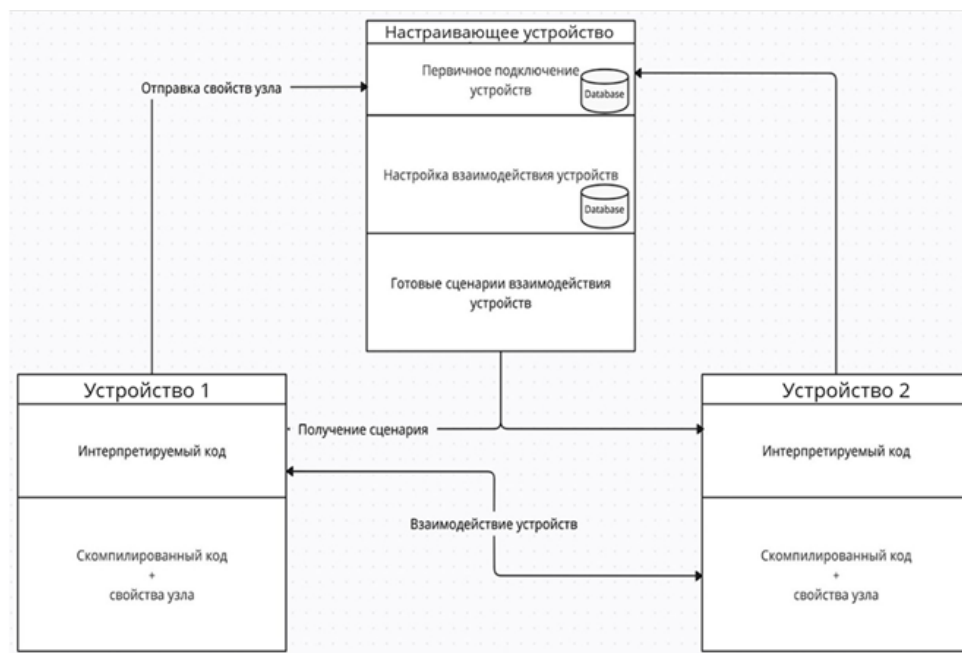


Рис. 7: Пример взаимодействий конечных устройств

## 4 Глава 3

### 4.1 Графический интерфейс

Было написано графическое приложение рис. 8 на языке Python с использованием фреймворка Qt, в котором производилась настройка системы и ее отладка.

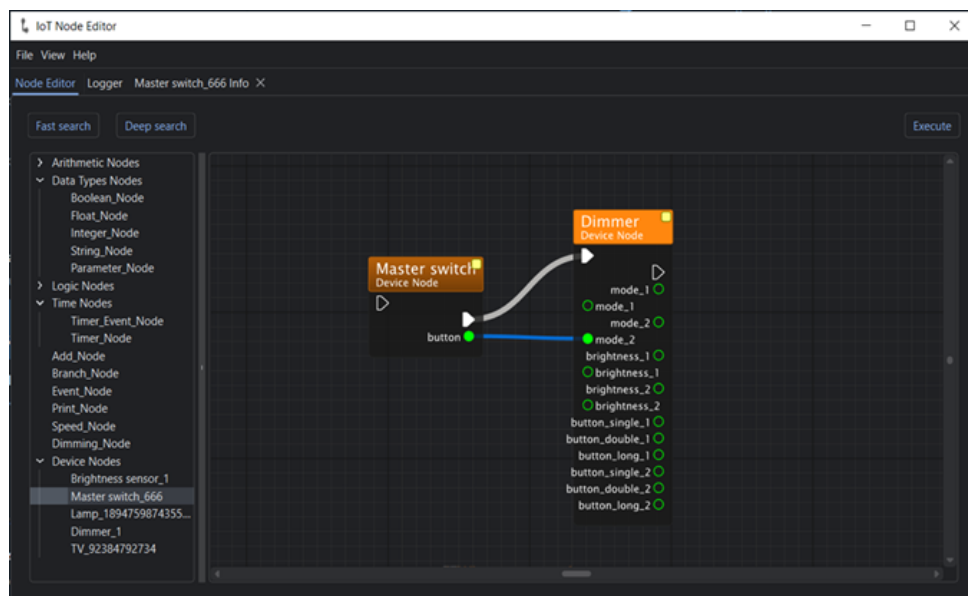


Рис. 8: Пример реализации конечного устройства

Графический интерфейс позволяет прописывать и более сложные сценарии взаимодействия устройств, что можно увидеть на рис. 9

### 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 приводят к значительному увеличению времени обработки и излишней загроуженности 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 последующем 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.

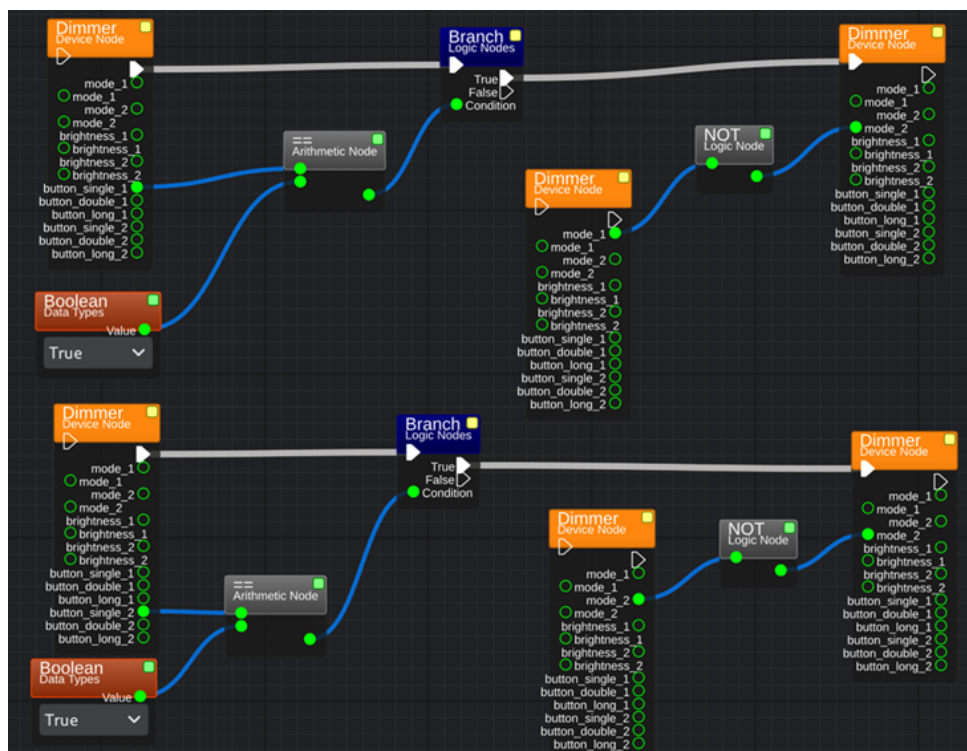


Рис. 9: Пример реализации конечного устройства

## 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.

Использовалось готовое решение на базе протокола Zigbee с ЦУД в виде YNDX-00510 [6] 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.

Таблица 2: Среднее время выполнения операций

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

## 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.



## Список литературы

- [1] Шерчков А. В. Анализ существующих технических решений «Умный дом» // Молодежный вестник Уфимского государственного авиационного технического университета. 2020. № 1 С. 153–155.
- [2] Тукмачева Ю. А. Обзор и анализ автоматизированных систем «Умный дом», представленных на российском сегменте рынка // Вестник науки. 2022. Т. 4. № 10 С. 139–144.
- [3] Колыбельников А. И. Обзор технологий беспроводных сетей // ТРУДЫ МФТИ. 2012. Т. 4. № 2 С. 3–29.
- [4] 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.
- [5] 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.
- [6] 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).