

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра технологий программирования

ЕМЕЛЬЯНОВА
Ольга Юрьевна

ИНТЕГРАЦИЯ NODE-RED С YANDEX IOT CORE ДЛЯ ПРИМЕНЕНИЯ В
СФЕРЕ ИНТЕРНЕТА ВЕЩЕЙ

Дипломная работа

Научный руководитель:
кандидат технических наук,
доцент И.С. Войтешенко

Допущена к защите

«__» _____ 2021 г.

Зав. кафедрой технологий программирования
доктор технических наук, профессор,
заслуженный деятель науки
Республики Беларусь А.Н. Курбацкий

Минск, 2021

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ И ТЕРМИНОВ	4
РЕФЕРАТ	5
РЭФЕРАТ	6
ABSTRACT	7
ВВЕДЕНИЕ	8
ГЛАВА 1 ПРОБЛЕМА ТЕСТИРОВАНИЯ И ОЦЕНКИ КАЧЕСТВА В СФЕРЕ ИНТЕРНЕТА ВЕЩЕЙ	10
1.1 Общая характеристика интернета вещей	10
1.1.1 Типичные требования, предъявляемые к системам интернета вещей	11
1.1.2 Особенности систем интернета вещей в контексте их тестирования	13
1.2 Тестирование систем интернета вещей. Использование имитационного моделирования.....	14
1.3 Обзор технологий интернета вещей и возможностей их применения в имитационном моделировании.....	17
1.3.1 Парадигма потокового программирования. Инструмент разработки Node- RED	18
1.3.1 Облачные технологии и интернет вещей. Yandex IoT Core	22
ГЛАВА 2 ЗАДАЧА ОБЕСПЕЧЕНИЯ ЭНЕРГОЭФФЕКТИВНОСТИ УМНОГО ДОМА	25
2.1 Системы управления энергопотреблением	25
2.1.1 Среда работы систем управления энергопотреблением	26
2.1.2 Основные компоненты	29
2.2 Требования и показатели качества	31
2.3 Возможные подходы к обеспечению эффективности использования энергии	32
2.3.1 Анализ основных подходов	33
2.3.2 Структура эксперимента по выделению наиболее перспективных подходов.....	34
ГЛАВА 3 ИСПОЛЬЗОВАНИЕ YANDEX IOT CORE И NODE-RED ДЛЯ ТЕСТИРОВАНИЯ СИСТЕМ ИНТЕРНЕТА ВЕЩЕЙ	36
3.1 Имитационная модель окружающей среды	36
3.1.1 Моделирование времени	37

3.1.2 Моделирование условий внешней среды	38
3.1.3 Моделирование тепловых потерь и работы устройств ОВиК.....	39
3.1.4 Моделирование поведения жителей дома	42
3.2 Оценка работы тестируемой системы.....	44
3.3 Проведение экспериментов.....	45
3.3.1 Генерация показаний датчиков с помощью Node-RED	46
3.3.2 Разработка и подключение тестируемой системы	48
3.3.3 Пользовательский интерфейс. Возможности по конфигурации системы пользователем.....	49
ГЛАВА 4 ПРОВЕДЕНИЕ ИМИТАЦИОННЫХ ЭКСПЕРИМЕНТОВ. ОЦЕНКА РЕЗУЛЬТАТОВ	53
4.1 Имитационные эксперименты	53
4.2 Выводы в области управления энергопотреблением	58
4.3 Анализ использования Node-RED и Yandex IoT Core.....	61
ЗАКЛЮЧЕНИЕ	64
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	66

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ И ТЕРМИНОВ

IoT	интернет вещей (Internet of Things)
MQTT	передача телеметрических сообщений с использованием очередей (Message Queuing Telemetry Transport)
Yandex Cloud	набор облачных сервисов от компании Yandex
Node-RED	инструмент потокового программирования
Yandex IoT Core	облачный сервис для интернета вещей
HTTP	протокол передачи гипертекста (Hypertext Transfer Protocol)
JSON	текстовый формат обмена данными, основанный на JavaScript (JavaScript Object Notation)
ОВиК	отопление, вентиляция и кондиционирование
ВИЭ	возобновляемые источники энергии

РЕФЕРАТ

Дипломная работа, 58 с., 14 рис., 6 табл., 4 приложения, 21 источник.

Ключевые слова: ИНТЕРНЕТ ВЕЩЕЙ, ДОМАШНЯЯ АВТОМАТИЗАЦИЯ, СИСТЕМА УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ, ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ, ОБЛАЧНЫЕ ТЕХНОЛОГИИ, ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ.

Объект исследования — системы интернета вещей, инструмент визуального программирования Node-RED, облачный сервис Yandex IoT Core, а также методы имитационного моделирования и тестирования для оценки качества систем управления энергопотреблением.

Цели работы — оценить возможность интеграции Node-RED и Yandex IoT Core, применения этих технологий в сфере интернета вещей. Разработать имитационную модель с использованием этих технологий и проверить ее состоятельность в испытании нескольких подходов к обеспечению эффективности энергопотребления в сфере домашней автоматизации

Методы исследования — анализ источников в сфере тестирования систем интернета вещей, а также связанных с использованием домашней автоматизации для обеспечения эффективности энергопотребления, прототипирование, имитационное моделирование, анализ данных.

В результате разработана программа для имитационного моделирования систем интернета вещей с использованием Node-RED и Yandex IoT Core. На основе проведенных экспериментов сформулированы рекомендации по применению нескольких подходов к обеспечению энергоэффективности.

Область применения — для тестирования и исследования в сфере интернета вещей, в частности, для оценки качества систем управления энергопотреблением.

РЭФЕРАТ

Дыпломная праца, 58 с., 14 рыс., 6 табл., 4 дадаткі, 21 крыніца.

Ключавыя словы: ІНТЭРНЭТ РЭЧАЎ, ДАМАШНЯЯ АУТАМАТЫЗАЦІЯ, СІСТЭМА КІРАВАННЯ ЭНЕРГАСПАЖЫВАННЕМ, ВІЗУАЛЬНАЕ ПРАГРАММАВАННЕ, ВОБЛАЧНЫЯ ТЭХНАЛОГІІ, ІМІТАЦЫЙНАЕ МАДЭЛІРАВАННЕ.

Аб’ект даследвання — сістэмы інтэрнэту рэчаў, інструмент візуальнага праграмавання Node-RED, воблачны сэрвіс Yandex IoT Core, а таксама метады імітацыйнага мадэлявання і тэставання для ацэнкі якасці сістэм кіравання энергаспажываннем.

Мэты працы — ацаніць магчымасць інтэграцыі Node-RED з Yandex IoT Core, выкарыстання гэтых тэхналогій у сферы інтэрнэту рэчаў. Распрацаваць імітацыйную мадэль з выкарыстаннем гэтых тэхналогій і праверыць яе грунтоўнасць у выпрабаванні некалькіх падыходаў да забеспячэння эфектыўнасці энергаспажывання ў сферы хатняй аўтаматызацыі.

Метады даследвання — аналіз крыніц у сферы тэставання сістэм інтэрнэту рэчаў, а таксама звязаных з выкарыстаннем хатняй аўтаматызацыі для забеспячэння эфектыўнасці энергаспажывання, сазданне прататыпаў, імітацыйнае мадэліраванне, аналіз дадзеных.

У выніку была распрацавана праграма для мадэліравання сістэм інтэрнэту рэчаў з выкарыстаннем Node-RED і Yandex IoT Core. На аснове праведзенных эксперыментаў сфармуляваны рэкамендацыі па ужыванні некалькіх падыходаў да забеспячэння энергаэфектыўнасці.

Вобласць ужывання — для тэсціравання і даследвання у сферы інтэрнэту рэчаў, асабліва для ацэнкі якасці сістэм кіравання энергаспажываннем.

ABSTRACT

Thesis, 58 p., 14 fig., 6 tables, 4 appendices, 21 sources.

Keywords: INTERNET OF THINGS, HOME AUTOMATION, ENERGY MANAGEMENT SYSTEM, VISUAL PROGRAMMING, CLOUD TECHNOLOGIES, SIMULATION MODELING.

Object of study — the Internet of Things systems, Node-RED visual programming tool, Yandex IoT Core cloud service, as well as simulation and testing methods used for evaluation of energy management systems.

Objectives — evaluate the feasibility of integration of Node-RED with Yandex IoT Core, their application in the field of internet of things, as well as to develop a simulation model using these technologies and to test its validity by using it to test and compare several approaches for ensuring energy efficiency in home automation.

Method — analysis of sources in the field of IoT systems testing, as well as those related to the use of home automation for energy efficiency, prototyping, simulation, data analysis.

As a result, a method for modeling IoT systems was developed and implemented using Node-RED and Yandex IoT Core. Based on the experiment conducted, recommendations for several energy efficiency approaches have been formulated.

Application area — for testing and research in the field of Internet of Things, in particular for assessment and evaluation of energy management systems.

ВВЕДЕНИЕ

Развитие информационных технологий в последние десятилетия привело к появлению такой концепции, как интернет вещей. В то время как привычный интернет связывает компьютеры, в рамках новой идеи к нему подключаются в том числе и обычные вещи: осветительное оборудование, бытовые приборы, отопительные системы, автомобили и многие другие. Предполагается, что внедрение глобального интернета вещей приведет к далеко идущим последствиям. Станет возможным получить доступ к данным в ранее невиданных масштабах, что, в свою очередь, приведет к оптимизации и автоматизации многих процессов реального мира. В каком-то смысле переход к интернету вещей является логическим продолжением информатизации и компьютеризации всех сфер жизни.

При всех преимуществах, которые должны принести системы интернета вещей, при внедрении подобных систем разработчики сталкиваются с рядом сложностей. Как и любая развивающаяся сфера, интернет вещей страдает от недостатка стандартизации и отсутствия общепринятых инструментов. Разработка систем интернета вещей сопряжена с уникальными сложностями, связанными с их погруженностью в поток данных окружающей среды. При этом нередко необходимо обеспечить корректность работы систем, поскольку особенностью среды их работы является высокая степень ответственности. Поэтому необходимо исследование подходов к тестированию и изучению таких систем. В данной работе изучается подход, основанный на имитационном моделировании с использованием Node-RED и Yandex IoT Core. Node-RED является гибким программным средством, позволяющим реализовать различные методы имитационного моделирования в сочетании с созданием прототипов систем и графического интерфейса для проведения экспериментов, в то время как Yandex IoT Core позволяет интегрировать систему с облачным сервером, передавать и обрабатывать данные.

Таким образом, задачей данной работы является, с одной стороны, анализ технологий Node-RED и Yandex IoT Core, а с другой, разработка подхода к моделированию и тестированию систем интернета вещей с их использованием.

В главе 1 дается общая характеристика проблемы тестирования систем интернета вещей. Анализируются особенности таких систем, типичные требования, предъявляемые к ним. Рассматриваются возможные подходы к их тестированию. Дается обоснование использованию имитационного моделирования. Анализируется роль парадигмы потокового программирования и вдохновленного ей инструмента разработки Node-RED, а также облачных технологий на примере

Yandex Cloud и сервиса Yandex IoT Core в сфере интернета вещей, потенциальные возможности и причины их использования для имитационного моделирования.

В главе 2 дается более детальная характеристика использования домашней автоматизации для управления энергопотреблением — прикладной области в рамках сферы интернета вещей. Рассматривается среда их работы, основные компоненты. Описываются типичные требования и показатели качества для таких систем. Анализируются существующие подходы.

В главе 3 описывается процесс разработки системы тестирования и моделирования систем интернета вещей с использованием Node-RED и Yandex IoT Core, включая математическую модель происходящих процессов и способы ее реализации.

В главе 4 описывается пример использования разработанной системы для проведения экспериментов, позволяющих сравнить несколько подходов, на основе которых делаются рекомендации по их использованию. Проводится анализ использования Node-RED и Yandex IoT Core.

ГЛАВА 1

ПРОБЛЕМА ТЕСТИРОВАНИЯ И ОЦЕНКИ КАЧЕСТВА В СФЕРЕ ИНТЕРНЕТА ВЕЩЕЙ

1.1 Общая характеристика интернета вещей

Развитие микроэлектроники привело к миниатюризации и удешевлению электронных компонент, благодаря чему стало возможным встраивать компьютерные чипы в различные вещи, а также использовать множество миниатюрных датчиков. Коммуникационные компоненты подобных устройств позволили объединять их в одну сеть для обмена данными и командами. Так появилась концепция интернета вещей.

Каждая вещь по отдельности может предложить некоторую долю автоматизации: например, лампочка в комнате может включаться по датчику движения — однако в интернете вещей предметы объединены в такую систему, которая может учитывать большее количество факторов одновременно. Последствием такой конфигурации является широкая доступность информации для анализа и принятия решений. Таким образом, интернет вещей позволяет автоматизировать и оптимизировать процессы, к которым он применяется, что во многом снимает нагрузку и ответственность с пользователя, что в особенности полезно в сферах, требующих постоянного контроля. Наиболее часто системы интернета вещей предназначены для оказания фоновой поддержки или слежения за различными каждодневными процессами.

В качестве модели для описания работы систем интернета вещей предлагается обработка потока данных. Центральной частью рабочего цикла системы является реакция на происходящие события, однако стоит отметить, что не все датчики работают по такой модели: иногда контролирующее устройство получают наблюдаемые данные через фиксированные промежутки времени. Таким образом, система постепенно пропускает через себя происходящие данные, принимая решения об отправлении команд подконтрольным устройствам по необходимости. Кроме того, зачастую отдельным требованием является хранение получаемых данных, чтобы позднее использовать их для анализа.

Уходя от абстракции потока данных, стоит отметить, что системы интернета вещей обычно работают в неоднородной среде, где устройства как составные элементы представляют собой мобильные гетерогенные объекты, оснащенные датчиками и манипуляторами. Каждое устройство может отличаться от других по функционалу и доступным ресурсам, причем роль играет как программное, так и

аппаратное обеспечение. Для связи элементов воедино используется коммуникативно-вычислительная инфраструктура интернета. Таким образом, требуется обеспечение корректности как обработки данных, так и взаимодействия между различными частями рассматриваемой распределенной системы.

Учитывая перечисленные особенности, представляет интерес проверка правильности функционирования такой коммуникативно-вычислительной системы и ее соответствия заданным требованиям. При этом нередко возможно отделить технические аспекты реализации, такие как выбор конкретных устройств или протоколов взаимодействия, чтобы сконцентрироваться на обеспечении корректного поведения программной части. Отметим также сходную задачу определения пригодности того или иного подхода, возникающую в процессе проведения исследований в сфере интернета вещей. Иначе говоря, требуется разработать метод, который бы позволил в зависимости от конкретной ситуации:

1. проверить систему или ее прототип на соответствие заранее определенным критериям,
2. или же дать оценку качества работы системы путем вычисления заданных показателей.

Процесс решения одной из или обеих перечисленных задач для данного программного продукта или прототипа будем называть тестированием.

1.1.1 Типичные требования, предъявляемые к системам интернета вещей

Учитывая, что центральное место в процессе тестирования в контексте данной работы занимает вычисление некоторых показателей для некоторой системы, следует определить, какого рода требования предъявляются к подобным продуктам.

Существует множество вариантов классификации и систематизации возможных требований к программному обеспечению, равно как и показателей, по которым определяется соответствие этим требованиям. В стандарте ISO/IEC TR 9126—2:2003 предлагается шесть категорий метрик для оценки поведения программных продуктов: функциональность, надежность, удобство использования, эффективность, возможность сопровождения и переносимость. Для расчета этих показателей систему необходимо поместить в условия реального использования [19, с. 4]. Все перечисленные метрики могут быть использованы для тестирования

любого программного обеспечения, однако в конкретной сфере, в частности — в сфере интернета вещей, некоторые категории более полезны, чем другие.

Проверка программного обеспечения на функциональность зачастую занимает основное место в тестировании. По одной из классификаций требования делятся на функциональные и нефункциональные, то есть все остальные. Фундаментально, под функциональностью будем понимать соответствие результатов работы системы ожиданиям пользователя. К этой же категории относят показатели внешней совместимости и безопасности системы [19, с. 6—7]. Таким образом, понятие функциональности само по себе включает в себя многие из основных критериев оценки системы.

Функциональные требования можно назвать наиболее важными и в контексте интернета вещей. Задачей систем интернета вещей является автоматизация и оптимизация различных процессов, нередко в сферах с высокой степенью ответственности, где требуется точное соответствие разработанным предписаниям, иначе последствия могут быть тяжелыми. Поэтому требуется обеспечить отсутствие ошибок в поведении системы настолько, насколько это возможно. Стороннее вмешательство также может быть проблемой: успешная атака, например, на «умный дом» потенциально может привести к контролю злоумышленником таких важных для качества жизни систем, как отопление или освещение. Кроме того, через системы интернета вещей нередко проходят конфиденциальные данные, из которых можно делать чувствительные выводы о их пользователях.

Среди нефункциональных требований в качестве особенно важных можно выделить требования к надежности и эффективности. Несмотря на то, что большинство систем интернета вещей не требуют принятия решений в жестком реальном времени, реакция на происходящие события должна быть своевременной, а отказ может привести к катастрофическим последствиям. Для оценки надежности предлагаются метрики зрелости, отказоустойчивости, способности к восстановлению, а для оценки эффективности — метрики своевременности реакции системы и загрузки ресурсов [19, с. 16, 46]. Дополнительной причиной использования требований к эффективности является ограниченность ресурсов большинства устройств интернета вещей.

Другие категории требований, такие как удобство использования, возможность сопровождения, переносимость, могут также применяться к системам интернета вещей, однако их автоматизация расчёта таких показателей возможна в значительно меньшей степени. Например, под удобством использования понимается простота понимания, изучения и управления работой системы [19, с.

28—29], что важно и в сфере интернета вещей, однако для оценки этих показателей необходимо тестирование реальными пользователями.

Подводя итоги, отметим, что большинство описанных требований являются широкими категориями, а для конкретной системы или по крайней мере предметной области необходимо уточнение выбранных метрик. Пример такого анализа будет описан в дальнейших главах.

1.1.2 Особенности систем интернета вещей в контексте их тестирования

Как было описано ранее, в контексте данной работе под тестированием понимается вычисление некоторых показателей, на основании которых оценивается качество работы системы, возможно, с промежуточным шагом в виде проверки соответствия ее заранее заданным требованиям. Такая задача может возникать как при наличии готового продукта, для которого необходимо обеспечить корректность его поведения, так и при предварительном исследовании предметной области для определения наиболее перспективных подходов.

Среда работы систем интернета вещей зачастую такова, что изменения можно проследить только на больших промежутках времени. Например, для проверки корректной реакции на сезонные факторы необходимо тестировать систему в течение года или более, потому что меньший период времени не охватит весь возможный спектр. Вместе с тем некоторые события настолько редки, что они могут не встретиться даже в таком долгом исследовании. В качестве примера можно привести экстремально высокие или экстремально низкие температуры, которые могут встречаться один раз в несколько лет.

Иногда возможно протестировать готовую систему с учетом всех особенностей ее физической реализации. С другой стороны, нередко продукт существует только в виде прототипа или не существует вообще, а необходимо протестировать тот или иной теоретически разработанный подход. В таком случае построение физической реализации системы сопряжено с большими затратами ресурсов, в том числе денежных. Отметим, что даже в случае существования готовой системы может иметь смысл отделение ее аппаратной реализации от программного обеспечения. В частности, становится возможным воспроизвести некоторые заданные ситуации, чтобы проверить корректность некоторых изменений. Кроме того, как было указано ранее, время исследования может представлять проблему, если требуется отследить изменения, происходящие на

больших временных промежутках. В таком случае может иметь смысл использование технологий, позволяющих ускорить этот процесс.

Несмотря на названные сложности, невозможно недооценить значимость тестирования в сфере интернета вещей. Высокие требования к функциональности, надежности, эффективности делают эту стадию необходимой. С другой стороны, может быть удобным, особенно на начальных этапах разработки, отойти от физической реализации системы и использовать программные технологии.

В случае использования тех или иных технологий симуляции, эмуляции или имитации стоит обратить внимание на то, что требуется построение отдельной тестовой инфраструктуры для решения этой задачи. Нередко требуется определенная адаптация тестирующей и тестируемой систем для их совместной работы.

Наконец, отметим, что сложность может представлять разработка таких требований или критериев тестирования, которые бы учитывали несколько конфликтующих между собой целей. Зачастую задачи систем интернета вещей включают в себя несколько предписаний, между которыми требуется найти компромисс, причем на этапе исследования нередко само соотношение еще не определено. Таким образом, необходимо обеспечение определенной гибкости тестирующей системы, чтобы было возможно без затруднений переходить между различными показателями качества.

1.2 Тестирование систем интернета вещей. Использование имитационного моделирования

Подытожим описанное, кратко перечислив основные особенности систем интернета вещей. Такие системы тесно погружены в окружающую среду и постоянно получают информацию о ней через датчики. Как моделирование показаний этих датчиков, так и оценка работы системы невозможны в ручном режиме из-за большого объема и отсутствия человеко-машинных интерфейсов. Эти особенности приводят к тому, что всесторонняя оценка работы программных продуктов в области систем интернета вещей в среде их конечного развертывания требует большого количества ресурсов, в первую очередь денежных и временных. Еще сложнее оказывается решить задачу исследования гипотез в отношении перспективных подходов.

Таким образом, возникает необходимость разработки особой тестирующей программы, в рамках которой могли бы проводиться эксперименты над

тестируемой системой, то есть сравнение различных сценариев ее функционирования при различных условиях окружающей среды. Возникает несколько степеней свободы, по которым можно сравнивать поведение системы: изменение условий внешней по отношению к системе среды, изменение поведения или возможных состояний управляемых устройств, изменение алгоритмов, по которым принимает решения тестируемая система. Немаловажно, что при тестировании такой системы требуется детальное воспроизведение показателей, связанных со всеми процессами, относящимися к системе, в течение длительного виртуального времени.

Имитационное моделирование как раз и является разновидностью моделирования, которое позволяет удовлетворить описанные требования: детальное воспроизведение процессов, длительный период виртуального времени, отсутствие ограничений на структуру и объем входных и выходных данных [1].

Рассмотрим некоторые методы имитационного моделирования, представляющиеся полезными в сфере интернета вещей.

В методе системной динамики система представляется как совокупность потоков, накопителей, вспомогательных переменных и подмоделей со своими элементами. При данном подходе реализуется математическая модель, представляющая собой динамическую систему одновременных уравнений, интегрированная с источниками статистических данных [2, с. 17—18]. Например, такие параметры, как температура и влажность в помещении, могут быть представлены как функции нескольких переменных: внешней температуры, влажности, присутствия человека, состояния системы отопления и так далее. В процессе моделирования в каждый момент нужные показатели рассчитываются по этим формулам. В системе также может быть представлена обратная связь [2, с. 23]. В предыдущем примере температура в помещении влияет на управление отоплением, которое в свою очередь влияет на температуру.

Агентный метод состоит в том, чтобы индивидуально задать поведение отдельных агентов системы, из которого возникает глобальное поведение системы. Под агентами понимаются различные элементы системы: социальные, экономические, технические, экосистемные. Для каждого агента определяется множество его состояний, переходы между ними, вызывающие их события, временные задержки, совершаемые ими действия [2, с. 37—38]. Например, агентом, действующим в системе «умный дом», может быть житель этого дома. В простейшей модели можно представить два состояния «дома» и «не дома», между которыми агент переходит по достижению выбранного виртуального времени.

Возможны также гибридные методы, при которых в процессе моделирования сочетаются несколько подходов. В контексте интернета вещей наиболее подходящим представляется как раз такой метод. В частности, отдельные элементы системы могут быть представлены как агенты, состояния которых рассчитываются с помощью системной динамики.

В завершение раздела коротко коснемся других подходов, элементы которых могут быть интегрированы в такую гибридную систему. Во-первых, это дискретно-событийная парадигма, в которой фокусом моделирования являются ресурсы, очереди, задержки, переходы по событиям [2, с. 52]. В гибридной модели может использоваться свойство дискретности такой модели вместо непрерывных потоков системной динамики. Для учета в системе недетерминированности окружающей среды могут быть использованы методы стохастического моделирования, в котором состояние системы описывается в том числе и случайными величинами [2, с. 69]. Для подбора наиболее верной модели может быть применено эволюционное моделирование.

Прежде чем перейти к обсуждению использования Node-RED и Yandex IoT Core для имитационного моделирования, проведем краткий анализ одного из распространенных существующих решений в этой области. Simulink — это визуальная среда для проектирования и моделирования, включающая в себя графический редактор с различными компонентами для построения моделей, а также интегрированная с MATLAB для поддержки численных методов решения [17]. Создание модели в этой среде ведется путем соединения блоков, отвечающих за простые операции, такие как суммирование, умножение на константу, интегрирование по времени и другие. После этого пользователь может запустить моделирование и просмотреть результаты как в виде файлов с данными, так и графически.

Благодаря использованию MATLAB среда Simulink может предложить высокую точность моделирования, разумеется, при правильной разработке модели. Например, система может подобрать шаг автоматически, чтобы обеспечить наибольшую точность. Кроме того, в среде Simulink доступен широкий выбор компонент моделей, что позволяет применять ее практически в любой предметной области.

Тем не менее, это решение не лишено проблем, в первую очередь связанных с относительной сложностью вхождения, даже несмотря на использование визуального представления. Являясь профессиональным продуктом, Simulink также отличается относительно высокой стоимостью, в особенности для разработчиков или исследователей, не имеющих доступ в рамках некоторой

организации. Что касается процесса разработки и тестирования, модель, разработанная в Simulink, не может быть напрямую использована, более того, для существующей системы требуется разработка отдельной модели. Все эти факторы приводят к тому, что использование Simulink в некоторых случаях оказывается невозможным или неэффективным.

1.3 Обзор технологий интернета вещей и возможностей их применения в имитационном моделировании

Из анализа имитационного моделирования как подхода к тестированию интернета вещей, а также существующих решений этой задачи, можно выделить некоторые факторы, на которые стоит ориентироваться при выборе технологий для создания среды симуляции. В том числе, требуется выделить недостатки существующих решений, возможно, делающие их неприменимыми для той или иной задачи. В таком случае разработка нового подхода или системы является особенно актуальной.

Инструменты для имитационного моделирования должны позволять реализовывать широкий круг сценариев тестирования и эмулируемых устройств, а также рассчитывать различные оценки качества. При этом созданная система должна быть легко изменяемой и дополняемой. Из-за этого наиболее важным критерием выбора технологии является спектр ее возможностей.

Simulink является распространенным инструментом для имитационного моделирования, однако он имеет высокий порог вхождения как в смысле требуемых навыков, так и в смысле средств. Поэтому хотелось бы разработать более интуитивно понятный подход. При этом визуальная среда видится как преимущество, хотя только ее недостаточно. Кроме того, процесс разработки со стороны пользователя упрощается, когда существуют доступные компоненты для решения некоторых задач в формате библиотеки.

В качестве дополнительного требования можно также выдвинуть возможность перенесения разработанной системы между различными средами исполнения, в том числе упрощенный переход от прототипа к конечному продукту.

Наконец, при анализе Simulink еще одним недостатком следует назвать цену, высокий размер которой связан с моделью доступа и распространения. В противоположность этой среде желательно использовать программные средства с открытым исходным кодом или во всяком случае с оплатой только реально потребленных ресурсов.

1.3.1 Парадигма потокового программирования. Инструмент разработки Node-RED

Традиционно, в соответствии с принципами фон Неймана, компьютерная программа моделировалась как последовательность происходящих в определенном порядке операций [6]. В такой модели фокусом являются команды, в то время как данные являются статичными. Каждая команда может изменить значение, но не положение данных, после чего процессор переходит к следующей команде. Разделяемая память в модели фон Неймана не представляет проблем для однопоточных программ, однако приводит к сложностям при попытке распараллеливания. В качестве альтернативы была предложена архитектура, основанная на потоках данных. Программа в таком случае представляется как направленный граф, в котором данные «перетекают» между командами [6]. Каждая отдельная операция функционирует как черный ящик с четко определенными входами и выходами.

Развитием идеи программирования потоков данных является потоковое программирование. В этой парадигме программа представляется как сеть процессов, внутреннее содержание которых игнорируется. В сети определяются соединения между процессами, через которые они могут передавать порции данных. Разработанные с использованием описанной выше структуры программы естественным образом являются слабо связанными. Узлы в сети связаны только тем, какой формат данных они принимают на вход и производят на выходе.

Node-RED — это визуальный инструмент разработки, основанный на парадигме потокового программирования. Он был разработан компанией IBM и написан на языке программирования JavaScript. В Node-RED приложение представляется как сеть так называемых узлов. Каждый узел имеет определенную цель: на вход поступают некоторые данные, они обрабатываются внутри, после чего передаются дальше по сети. За перемещение данных между узлами отвечает сеть [15].

Для объединения и организации узлов в единую концептуальную единицу в Node-RED используется поток. Поток представляется как отдельная страница в редакторе и предлагает промежуточный уровень контекста между отдельным узлом и всей системой в целом. Не все узлы в рамках потока должны быть связаны между собой, однако слово «поток» также может использоваться для обозначения набора связанных между собой узлов.

Каждый узел в потоке ожидает либо сообщения от предыдущего узла, либо какого-либо внешнего события, не связанного с деятельностью потока. В качестве

примера внешнего события можно привести HTTP-запрос. После получения сообщения или возникновения события узел переходит к обработке. Завершив ее, узел может послать сообщение. Каждый узел может иметь только один входной порт, но любое количество выходных портов. Узлы передают друг другу сообщения, которые являются JavaScript-объектами. Соединения между узлами называются проводами.

Кроме как через сообщения, информация может передаваться между узлами через контекст. В Node-RED определено три уровня контекста: уровень узла, когда информация доступна только узлу, который ее произвел, уровень потока, когда информация доступна всем узлам потока, и глобальный [15].

Средой исполнения Node-RED является Node.js. Разработка также ведется через веб-браузер. Внутри редактора потоков пользователь может добавлять узлы и соединять их проводами. Также есть возможность импортировать и экспортировать потоки в виде JSON-файлов. Node-RED допускает реализацию своих узлов на JavaScript, для визуализации используется HTML.

Внутри каждый узел реализует шаблон «Наблюдатель». Благодаря этому узлы, соединенные проводом, могут уведомляться о событиях. Входные узлы в потоке могут подписываться на внешние сервисы, прослушивать порты или обслуживать HTTP-запросы. Узел-обработчик получает данные и совершает с ними заданные действия, после чего уведомляет последующие узлы с помощью событий. Узел также может генерировать дополнительные события, устанавливать связь с внешними сервисами или операционной системой [21]. Также существуют узлы конфигурации, в которых можно хранить общую для всего потока конфигурационную информацию.

Программу Node-RED можно разделить на три уровня: глобальный, уровень потока и уровень узла. На глобальном уровне находится вся программа: все узлы, их параметры и связи между ними. Запуск на исполнение и сохранение глобального контекста производятся одновременно. При этом исполнение непрерывно, возможно только обновление и перезапуск всей программы одновременно. Каждый узел может принять данные в любой момент и все узлы работают параллельно.

Уровень потока играет роль в двух случаях. Во-первых, разделяемые между несколькими узлами переменные могут быть сохранены в контекст потока. Во-вторых, разделение программы по нескольким потокам, представленным отдельными вкладками в визуальном редакторе Node-RED, помогает в структуризации. Заметим, что все потоки исполняются одновременно, хотя некоторые потоки могут быть отключены.

Наконец, особенный интерес представляет уровень узла. Если не учитывать возможность добавить самостоятельно разработанный узел, наиболее общим по своему функционалу узлом является Function. В таком узле можно записать любой алгоритм с помощью JavaScript, причем можно определить разное поведение для запуска, остановки и каждой активации узла. Такой узел может иметь несколько выходов.

Для некоторых распространенных задач вместо узла Function можно использовать узлы более узкого назначения, которые, тем не менее, обобщают некоторую операцию. В частности, одним из таких узлов является узел Change, который позволяет изменять свойства сообщений и устанавливать значения переменных контекста. Для того, чтобы отправить сообщение в одну из нескольких ветвей потока в зависимости от входных данных или переменных контекста, используется узел Switch. Этот узел отправляет сообщение во все свои выходы, соответствующие указанным в нем правилам, хотя он может быть задан и таким образом, чтобы отправлять только по первому подходящему правилу.

В то же время существуют и специализированные узлы, предназначенные для решения определенной задачи: разбор того или иного формата данных, обработка запросов с использованием заданного протокола, файлового ввода-вывода, отображения объектов графического интерфейса. В Node-RED также возможно подключать сторонние коллекции узлов для тех задач, для которых не удалось найти подходящий узел в коллекции по умолчанию.

Кроме визуального редактора, в распоряжении пользователя Node-RED также есть несколько других представлений, помогающих анализировать поведение системы. Наиболее универсальным является так называемая «приборная панель», на которой демонстрируются элементы графического интерфейса, задаваемые специальными узлами. При этом возможно как выводить информацию на экран в виде текста, графиков и изображений, так и использовать такие элементы управления, как кнопки и поля ввода.

Node-RED предлагает возможность ручного запуска сообщения по потоку с помощью узла Inject, а также чтения передаваемых сообщений с помощью узла Debug. Все сообщения, считываемые узлом Debug, отображаются на отдельной панели отладки. Можно добавить несколько таких узлов.

При использовании в сфере интернета вещей Node-RED предлагает несколько преимуществ по сравнению с другими вариантами: визуальное представление, разнообразие поддерживаемых платформ, возможность абстракции относительно физической реализации системы. Разумеется, как и любой дополнительный слой системы, Node-RED вносит дополнительные затраты

на пересылку и обработку данных, но в большинстве систем это допустимо. Кроме того, задачи некоторых систем могут быть слишком сложны для функционала, предлагаемого Node-RED, хотя это частично и может решаться пользовательскими дополнениями. Несмотря на это, инструмент Node-RED хорошо подходит для разработки систем интернета вещей, причем его преимущества в этой области также являются его сильными сторонами с точки зрения моделирования и тестирования таких систем. Подробнее опишем возможный метод использования Node-RED в этой области.

Сама система интернета вещей представляется как отдельный поток Node-RED, хотя она и может быть реализована независимо и просто быть подключена через сетевые узлы. В отдельном потоке или потоках реализуется программа для проведения имитационных экспериментов. В нее входят источники данных, генераторы различных показателей, в том числе и системного времени, а также узлы, отвечающие за коммуникацию с тестируемой системой и анализ ее поведения.

Использование Node-RED не накладывает ограничений на формат данных, кроме того, что он должен быть представлен как объект JavaScript. Большую трудность представляют ограничения по частоте событий. На практике было испытано, что при посылании сигнала чаще, чем 10 раз в одну секунду, в системе появляются видимые задержки. Таким образом, при временном разрешении внутри системы в один час можно добиться ускорения по сравнению с реальным временем в 36000 раз, а при временном разрешении в минуту — только в 600. Дальнейшее уменьшение периода виртуального времени между двумя сигналами имеет ограниченное практическое применение, поскольку масштабы требуемых временных промежутков экспериментов в системах интернета вещей достигают периодов от дня до года.

Node-RED представляется подходящим инструментом для имитационного моделирования, поскольку он предлагает большое разнообразие различных узлов, а также способы их объединения таким образом, чтобы воспроизвести практически любые нужные показатели. Благодаря использованию визуального программирования построение сложных связей между показателями упрощается с точки зрения разработчика, а вместе с тем и пользователю становится легче понимать строение системы. Задача тестирования дополнительно упрощается, если сама тестируемая система была реализована с помощью Node-RED.

1.3.1 Облачные технологии и интернет вещей. Yandex IoT Core

Сервис Yandex IoT Core был запущен компанией Яндекс как составная часть облачной платформы Yandex Cloud, также известной как Яндекс.Облако. Данный сервис предлагает решения для интернета вещей в формате «программное обеспечение как услуга». В такой модели поставщик предоставляет готовое программное обеспечение, которое он полностью обслуживает сам, а пользователь может получить доступ к его функциям. Таким образом, данный сервис позволяет значительно сократить затраты на разработку и создание инфраструктуры интернета вещей.

Как было описано ранее, большинство систем интернета вещей состоит из двух частей (рисунок 1.1). Первая из них связана с передачей данных и команд между хранилищем и устройствами, а вторая заключается в работе с полученными данными. При этом первая часть одновременно характеризуется четкими нефункциональными требованиями, для удовлетворения которых хорошо подходят облачные хранилища, и большой схожестью среди различных систем интернета вещей. Поэтому команда Яндекса при разработке сервисов для использования в сфере интернета вещей сфокусировалась в первую очередь на предоставлении возможности быстро и надежно получать данные от устройств и отправлять им команды.

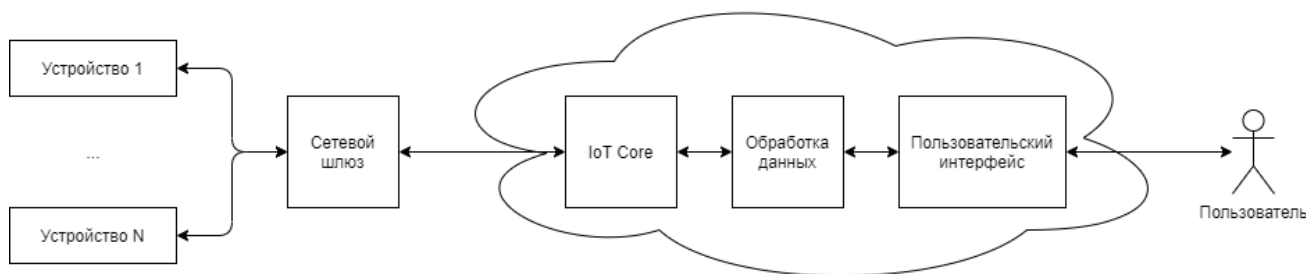


Рисунок 1.1 — Архитектура системы интернета вещей с использованием облачных технологий

Помимо самого Yandex IoT Core, в экосистеме сервисов для интернета вещей также имеются такие сервисы, как Yandex Cloud Functions и Yandex Object Storage. Сервис IoT Core представляет собой MQTT-брокер с набором дополнительных функций. Сервис Cloud Functions позволяет запускать программы пользователя для предобработки данных без создания виртуальных машин. Сервис Object Storage дает возможность хранить большие массивы данных, в том числе архивные записи,

накопленные за большой период работы системы. В дополнение к этим сервисам в Yandex Cloud также есть предложения для хранения и анализа данных, например, Yandex Managed Service for ClickHouse.

Основным сервисом Yandex Cloud, предлагающим решения в области интернета вещей, является Yandex IoT Core. Данный сервис предназначен для двухстороннего обмена данными между облаком и устройствами с использованием протокола MQTT. В этом протоколе для передачи данных используются именованные очереди, иначе называемые темами или топиками, в которые можно записывать данные. Подписавшись на события очереди, объекты может асинхронно получать данные из нее.

Основными элементами Yandex IoT Core являются устройство и реестр, которые обмениваются данными и командами через MQTT-брокер [4]. Под устройством понимается экземпляр физического устройства интернета вещей. Устройство может отправлять телеметрические данные и получать команды. Реестр — это набор логически связанных между собой устройств. Устройства в реестре могут взаимодействовать между собой, отправляя данные и получая команды. Реестр может читать данные и отправлять команды.

MQTT-брокер отвечает за обмен MQTT-сообщениями между устройствами и реестрами. С одной стороны, он получает и обрабатывает сообщения, а с другой, контролирует их доставку MQTT-клиентам, то есть устройствам и реестрам. Обмен данными может происходить только через него. Каждое сообщение содержит тему, или топик, которые классифицируют данные. Клиенты получают только сообщения из тех тем, на которые они подписаны. MQTT-брокер обеспечивает необходимый уровень качества обслуживания при обмене сообщениями. Yandex IoT Core поддерживает два уровня: QoS 0, когда сообщение отправляется только один раз и доставка не гарантируется, и QoS 1, когда сообщение обязательно доставляется как минимум один раз, однако допускается получение дубликатов.

Чтобы устройства и реестра могли получать сообщения через MQTT-брокер, они должны быть подписаны на нужные топик. Для особо важных сообщений, например показаний датчиков, требующих немедленной реакции, особенно если возможна потеря соединения, Yandex IoT Core предлагает перманентные топик, которые сохраняют последнее сообщение, отправленное в него. Это сообщение будет отображено при восстановлении соединения, даже если в этот момент не было опубликовано новых сообщений [4].

Что касается имитационного моделирования, Yandex IoT Core дополняет Node-RED, создавая возможность коммуникации с тестируемой системой. Скорее всего, устройства в физической реализации системы интернета вещей в любом

случае будут связаны с управляющим центром. В таком случае достаточно систему только адаптировать, заменив существующие узлы коммуникации на узлы связи с Yandex IoT Core. При этом облачные технологии позволяют расширить возможности тестирующей системы: например, возможно хранить результаты симуляции в базе данных или запускать те или иные действия в зависимости от приходящих команд. Вычислительные ресурсы, доступные на облачном сервере, также могут быть полезны, например для более глубокого анализа данных, создания более совершенных алгоритмов с использованием машинного обучения.

ВЫВОДЫ

Первая глава была посвящена рассмотрению и решению следующих проблем:

1. Проведен анализ сферы интернета вещей. Выделены основные особенности систем интернета вещей, влияющие на процесс их тестирования.
2. Рассмотрены основные подходы к тестированию систем интернета вещей.
3. Обосновано использование имитационного моделирования.
4. Рассмотрены основные технологии, используемые в работе.

ГЛАВА 2

ЗАДАЧА ОБЕСПЕЧЕНИЯ ЭНЕРГОЭФФЕКТИВНОСТИ УМНОГО ДОМА

2.1 Системы управления энергопотреблением

Изменение климата, вызванное выбросами парниковых газов в атмосферу, остается одной из важнейших проблем, стоящих перед человечеством в настоящее время. В качестве основных направлений ее решения рассматривается использование возобновляемых источников энергии с одной стороны и повышение энергоэффективности с другой. Иначе эти два пути можно охарактеризовать как изменение подхода к производству и потреблению соответственно.

Домохозяйства вносят значительный вклад в общее энергопотребление. Согласно данным Национального статистического комитета Республики Беларусь, в структуре конечного потребления топливно-энергетических ресурсов в 2019 году жилищный сектор занимал почти 27% [5, с. 54]. Таким образом, оптимизация энергопотребления в этой сфере представляется целесообразной.

Для повышения энергоэффективности домохозяйств было разработано множество технологий, например, теплоизоляция стен, крыши, более эффективные котлы отопления. Традиционно большинство решений являлось «статическими», то есть воздействовать на их поведение после установки практически невозможно. В противоположность им в настоящее время набирают популярность интерактивные подходы, к которым относятся технологии, требующие настройки и взаимодействия с человеком после установки.

Вопрос о том, какой из подходов более эффективен, остается открытым. Некоторые исследования показывают, что простые интерактивные технологии, такие как программируемые термостаты, не только не снижают энергопотребление, но даже увеличивают его [16]. Наиболее вероятной причиной этого эффекта является то, что пользователи зачастую используют их неправильно, намеренно или нет. Интерактивные технологии в целом более требовательны к пользователю, что часто приводит к их неверному использованию [11].

К другой стороны, при использовании статических технологий всегда остается возможность улучшить энергоэффективность путем использования интерактивных технологий. Исследования так называемых «зеленых» зданий показали, что и они страдают от жестких схем энергопотребления, однако даже простые схемы автоматического управления позволяют снизить затраты энергии [7]. Кроме того, аргументом в пользу развития интерактивных технологий является

сама возможность взять контроль над потреблением энергии, которую они предлагают пользователю.

Использование интернета вещей позволяет пойти дальше в двух важных направлениях. С одной стороны, использование множества датчиков дает возможность контролировать большое количество различных показателей, чтобы обеспечивать максимальный комфорт по минимальной цене. С другой, относительно более мощные компьютеры, потенциально — облачные сервера, позволяют достичь значительной степени автоматизации и снять с пользователя часть ответственности за верную настройку системы.

Общая концепция использования домашней автоматизации в контексте обеспечения энергоэффективности проста: система интернета вещей управляет различными бытовыми приборами, основываясь на данных об окружающей среде. Ее задачей является достижение максимального уровня комфорта при минимальном использовании энергии.

Коротко коснемся особенностей. В таком контексте удобно классифицировать бытовую технику в зависимости от того, возможно ли планировать ее работу, или иначе говоря, может ли устройство работать в любое время и без участия человека. Это сделано для того, чтобы система могла назначать работу таких приборов на удобное время. Кроме того, система может быть дополнена более редкими устройствами локального производства и хранения энергии, что позволяет сокращать потребление от сети.

Развитием идеи домашних систем управления энергопотреблением является новая модель производства и распределения энергии — так называемая «умная энергосистема», характеризующаяся двунаправленным потоком данных и энергии между ее элементами. Все узлы сети могут производить энергию, отдавая ее при необходимости. В такой системе возможно реализовать динамическое управление всеми этапами производства энергии и потенциально приблизиться к максимально возможной эффективности.

2.1.1 Среда работы систем управления энергопотреблением

Системы домашней автоматизации по определению предназначены для использования в различных домах, и системы управления энергопотреблением — не исключение. Когда речь идет о продуктах, предназначенных для использования в одном домохозяйстве, следует различать квартиры и индивидуальные жилые дома, поскольку допустимые стратегии регуляции энергопотребления, а также

возможные устанавливаемые устройства различаются. Что касается квартир, оптимизация потребления энергии значительно затруднена, поскольку многие коммунальные услуги предоставляются централизованно или во всяком случае совместно с другими жителями дома. Представляется целесообразным ориентироваться на полный спектр возможностей, доступных в индивидуальных жилых домах, учитывая при этом ограничения по их применению в квартирах.

Для определения наиболее перспективных направлений рассмотрим структуру конечного потребления энергии в домашних хозяйствах. По данным Национального статистического комитета Республики Беларусь, на отопление в среднем затрачивается 59,2% от суммарной использованной энергии, на горячее водоснабжение — 17,8%, на приготовление пищи — 13%, на бытовые приборы и освещение — 10% [3]. Структура энергопотребления по различным типам жилых домов показана в таблице 2.1. Заметим, что затраты энергии на отопление еще выше относительно других при рассмотрении основной целевой среды — индивидуальных жилых домов. Вместе с тем отметим, что для квартир с центральным отоплением больший относительный вес имеют иные направления использования энергии.

Таблица 2.1 — Структура общего конечного потребления энергии в домашних хозяйства [3]

	Дома с автономным отоплением	Дома с центральным отоплением	Квартиры с автономным отоплением	Квартиры с центральным отоплением
Отопление	70,6	73,4	66,6	52,0
Горячее водоснабжение	9,8	9,2	10,9	23,0
Приготовление пищи	10,8	9,6	12,7	14,3
Бытовые приборы и освещение	8,8	7,8	9,8	10,7

Таким образом, кажется разумным концентрироваться в первую очередь на отоплении, а затем — на горячем водоснабжении, для чего представляются полезными решения, регулирующие работу устройств на основе показаний датчиков об окружающей среде. Использование систем управления энергопотреблением нередко предполагает переход на автономное отопление и

водоснабжение, поскольку в таком случае возможен больший контроль над использованием энергии.

Вместе с тем, несмотря на относительно небольшую их долю в общей структуре потребления энергии домохозяйствами, не стоит игнорировать и другие направления: приготовление пищи, бытовые приборы, освещение. Одной из основных причин является распространенность некоторых простых оптимизационных решений, к которым относятся, в частности, автоматическое управление освещением, планирование использования бытовых приборов.

Одним из важных преимуществ систем управления энергопотреблением на основе технологий интернета вещей является широта их проникновения в разные аспекты окружающей среды и повседневной жизни пользователей. В отличие от узконаправленных решений, таких как программируемые термостаты или включение-выключение света по датчику движения, системы домашней автоматизаций способны учитывать большее количество факторов. Чтобы нечто подобное было возможным, необходимо обеспечить достаточную гибкость системы для подключения к ней различных устройств, от котлов отопления до электроплит, а потенциально и возобновляемых источников энергии, локальных средств хранения и других приборов, традиционно не относимых к бытовой технике.

Заметим, что подключаемые к интернету вещей устройства должны быть способными как минимум к коммуникации по сети и автоматическому управлению, причем могут использоваться команды, не имеющие прямых аналогов среди возможностей ручного управления. Прямой централизованный контроль не обязателен, в качестве альтернативы система управления энергопотреблением может снабжать устройства достаточными данными, с использованием которых они должны сами принимать решения [14]. Полезным может оказаться и дополнение бытовой техники дополнительными датчиками, значения которых также передаются по сети. Таким образом, требуется широкая модификация используемых бытовых приборов.

Другим преимуществом является то, что благодаря встроенным микропроцессорам становится возможным использовать программное обеспечение для автоматизации, например составления расписания работы устройств отопления. Потенциально коммуникативно-вычислительные возможности интернета вещей позволяют даже связывать группы домашних хозяйств и поставщиков энергии в системы, где достигается еще большая эффективность энергопотребления за счет широкой доступности данных о процессе производства энергии, требованиях и шаблонах поведения пользователей.

В то время как для повышения эффективности работы системы желательно подключить к ней максимальное количество устройств и датчиков, следует учитывать, что с точки зрения пользователей требуется минимальное вмешательство в их жизнь. Разумеется, степень допустимого может отличаться, но для того, чтобы обеспечить максимальный охват, наилучшей стратегией будет отталкиваться от более строгих требований. Важно отметить, что здесь есть два частично противоречащих друг другу аспекта. Для простоты использования цель разработчиков должна состоять в автоматизации как можно большего количества задач, чтобы пользователю не приходилось принимать решения, сопряженные с необходимостью глубокого изучения предметной области, анализа своего образа жизни и сходных областей. Чтобы повысить качество работы такой системы, необходимо увеличить количество получаемых ей данных, а также степень управляемости подконтрольных ей элементов, что связано с внедрением новых или заменой существующих устройств и датчиков. С другой стороны, помимо удобства использования пользователей также интересует простота вхождения, для чего необходимость переоборудования жилища имеет негативные последствия.

В качестве завершения обсуждения среды работы систем управления энергопотреблением отметим, что в перспективе разработанные для индивидуальных домохозяйств технологии возможно расширить на иные типы зданий: многоквартирные дома, офисные и производственные здания и другие. В первую очередь это касается средств обеспечения комфортного нахождения в помещении: поддержка температуры, влажности воздуха, освещение, водоснабжение и так далее. Для определения возможностей оптимизации энергопотребления в других направлениях требуется дополнительный анализ.

2.1.2 Основные компоненты

В литературе наблюдается значительное разнообразие устройств, предлагаемых к использованию в контексте систем управления энергопотреблением. Наиболее часто рассматриваются устройства ОВиК — отопления, вентиляции и кондиционирования — такие как обогреватель воздуха, водонагреватель, кондиционер. Несколько реже включается освещение [14]. Что касается бытовой техники, редко включаются устройства, которые работают по требованию пользователя, например компьютер. Часто встречаются такие приборы, как холодильник, стиральная машина, посудомоечная машина, духовка [14]. Таким

образом, выбор устройств согласуется с анализом наиболее перспективных направлений потребления энергии, проведенном в подразделе 2.1.1.

Некоторые исследования дополняют традиционные пассивные домашние хозяйства технологиями производства и хранения энергии. Прогнозируется, что со временем они станут более доступными [8]. Решения, включающие средства хранения энергии: аккумуляторы, батареи, электро- и гибридные автомобили — встречаются несколько чаще, чем учитывающие вклад ВИЭ (солнечные батареи, ветрогенераторы) [14]. Такие устройства по-прежнему мало распространены, однако их включение представляется перспективным для целей экологически чистого развития.

В то время как ВИЭ, электромобили, аккумуляторы редки, но доступны для обычного пользователя, некоторые предлагаемые компоненты и стратегии основываются на внедрении других технологий. Частым допущением является тарификация различных видов коммунальных услуг в соответствии с текущим спросом и предложением, из чего следует подключение к энергосистеме через интеллектуальный счетчик, объединяющий индивидуальные домохозяйства в глобальную сеть. Внедрение этой технологии позволит сократить потери энергии, усовершенствовать планирование производства энергии с учетом прогнозируемого спроса, сократить затраты пользователей, сделать техническое обслуживание более эффективным, уменьшить экологический ущерб [12]. Несмотря на предполагаемые преимущества, пока что неизвестно, будет ли развитие энергосистем идти в этом направлении, поэтому практическое внедрение в рамках индивидуальных домохозяйств пока что невозможно.

Таким образом, можно выделить следующие возможные физические компоненты: бытовая техника, источники энергии, устройства хранения энергии, умные коммунальные услуги. Список приведен по убыванию доступности. Что касается практической реализации подобных систем, о ее возможности можно говорить только в том случае, когда ее компоненты не требуют появления тех или иных новых технологий или подходов к производству и распределению энергии.

Для эффективного управления подконтрольными устройствами в систему также включаются датчики, считывающие такие показания окружающей среды, как температура, влажность, освещенность и тому подобные. Конкретный набор датчиков будет зависеть от требований к системе и доступных в ней устройств. Дополнительно могут применяться технологии определения пребывания человека в помещении с помощью датчиков движения или местоположения.

Что касается логической архитектуры систем управления энергопотреблением, в качестве основных компонент можно выделить

отслеживание и хранение данных о состоянии устройств, управление и планирование их работы, а также оповещение о нежелательных ситуациях [18]. При этом предполагается, что система может обучаться, то есть выделять шаблоны поведения пользователей, тенденции потребления и производства энергии. Наконец, система предоставляет пользовательский интерфейс для наблюдения и управления, корректно обрабатывает команды.

2.2 Требования и показатели качества

Среда работы систем управления энергопотреблением — жилое помещение, домохозяйство, даже офисное или производственное здание — приводит к многоцелевому характеру таких систем [14], то есть к необходимости нахождения компромисса между несколькими задачами. В то время как глобальной целью всегда является повышение эффективности использования энергии, на ее достижения налагаются ограничения, в частности, по комфорту пользователей. Кроме того, формулировка конкретной задачи может различаться в зависимости от контекста: снижение выбросов, денежная экономия, балансировка нагрузки на энергосистему — вот некоторые возможные направления.

Цели управления потреблением энергии могут быть выражены через стоимость. Хотя наиболее очевидным и преобладающим фактором является цена потребляемой энергии, в общую формулу расчета также могут включаться такие факторы, как затраты на начальную установку системы, штраф за вклад в общую нагрузку, прогнозируемый износ оборудования, налог на выбросы парниковых газов [14]. Таким образом, система может следовать единой цели — снижение получаемых по этой формуле денежных затрат. Выбор корректного соотношения может представлять сложность, если нет устоявшихся денежных значений для некоторых целей.

Основная задача из тех, которые сложно представить в денежном эквиваленте, — это комфорт пользователей, то есть их неудобства, связанные с качеством услуг, предоставляемых при доставке энергии. Система не должна приводить к значительному изменению их образа жизни. Для оценки влияния системы на комфорт пользователей могут использоваться различные штрафные функции: ограничения по значению, отклонение, штраф за отсутствие сервиса [14]. Иначе говоря, если стоимость представляет собой некоторое соотношение, которое требуется минимизировать, то требования по комфорту налагают ограничения на возможные стратегии.

Описанные выше соображения могут применяться в тестировании общих подходов к оптимизации потребления энергии. Например, может потребоваться оценить выгоды от добавления того или иного устройства и сравнить их с ценой приобретения и установки, или же выбрать из нескольких разрабатываемых алгоритмов наилучший.

С точки зрения конечного пользователя к системе могут предъявляться такие требования, как доступ к историческим данным и тенденциям, возможность удаленного управления устройствами, предупреждение о нежелательных ситуациях [18]. Конфиденциальность данных, надежность и эффективность системы также играют роль в оценке ее качества. Подобные требования скорее характерны для продуктов практической направленности, предназначенных для прямого использования по назначению, а не исследовательских прототипов.

2.3 Возможные подходы к обеспечению эффективности использования энергии

Прежде чем перейти к анализу конкретных подходов, укажем несколько общих соображений.

Понятие эффективности использования энергии зависит от выбранных критериев оценки. Возможные критерии были приведены в предыдущем разделе. Выбор конкретных требований зависит от целей пользователя, поэтому вне такого контекста принять решение о том, какие подходы лучше, бывает невозможно, хотя некоторые факторы играют роль для любых задач. При выборе стоит ориентироваться на существующую ситуацию. Необходимо проанализировать, в каких областях возможно получение наибольших преимуществ, а где использование энергии и так близко к минимальному. В некоторых случаях достаточно лишь небольших изменений особенно неэффективных алгоритмов работы устройств, чтобы значительно уменьшить потребление энергии.

Не стоит упускать из виду и эффективность подконтрольных устройств. Нередко более простым способом снизить энергопотребление будет использование лучших материалов и аппаратуры. При этом система домашней автоматизации ограничена возможностями устройств.

Вместе с тем стоимость внедряемого решения может оказаться выше предлагаемой им экономии, или же от пользователей требуется смириться со значительным вмешательством в их образ жизни. В таком случае снижение энергопотребления вряд ли окажется приоритетом.

При анализе эффективности требуется не только теоретическое рассмотрение, но и проведение экспериментов, поскольку могут обнаружиться непредвиденные сочетания факторов, приводящие к неожиданным результатам.

2.3.1 Анализ основных подходов

Наиболее простым подходом к снижению энергопотребления является включение и отключение устройств в зависимости от нахождения людей в помещении. Такой метод позволяет обеспечить значительное уменьшение использования энергии по сравнению с ситуацией, когда бытовая техника остается включенной, даже когда в ее работе нет необходимости [7]. Тем не менее, данный подход ограничен тем, что дает лишь небольшие преимущества по удобству в том случае, когда пользователь и так делает нечто подобное вручную.

Отметим, что использование подобных алгоритмов в контексте ОВиК, в особенности в сочетании со способностью системы к самообучению и предсказанию местоположения пользователей, приводят к значительному снижению энергопотребления на поддержание комфортной температуры [20]. В то же время использование программируемых вручную термостатов сопряжено с частыми ошибками, приводящими к увеличению энергопотребления [11], а использование интеллектуальных термостатов без способности к предсказанию поведения пользователей приводит к потерям в комфорте [20].

Кроме проверки нахождения людей в помещении, существует потенциал для использования сезонных и дневных вариаций показателей окружающей среды для управления ОВиК [7]. Таким образом, в качестве относительно доступного и простого решения в сфере оптимизации энергопотребления можно предложить систему, основывающуюся на комбинации датчиков для оценки окружающей среды и нахождения пользователей, и управляющую устройствами исходя из этого. В такую систему рекомендуется включить максимальное количество устройств, которые не должны быть доступны по требованию пользователя, а связаны с обеспечением необходимого уровня комфорта. В первую очередь, к ним относятся устройства ОВиК и освещения.

Для уменьшения зависимости от централизованных производителей энергии и сопряженных с этим денежных затрат предлагается использование возобновляемых источников энергии (ВИЭ). Их эффективность зависит от географических условий, так что необходимы исследования на основе погодных

данных выбранной территории, чтобы оценить перспективы применения солнечных батарей, ветроэлектрических установок и прочих подобных устройств.

Одной из основных проблем ВИЭ является их нестабильность. Для уменьшения воздействия этого фактора система может быть дополнена средствами хранения энергии [18]. Когда условия окружающей среды позволяют производство энергии, аккумуляторы заряжаются, после чего она может быть использована, когда существует необходимость.

Часто предлагается подход, основанный на планировании использования гибкой бытовой техники, из чего следует выравнивание спроса, в том числе снижение пиковой нагрузки, благодаря чему общая эффективность увеличивается [13]. Тем не менее, такой эффект будет заметен только на уровне больших групп домохозяйств, что будет препятствовать его внедрению. Для индивидуальных пользователей метод имеет смысл в том случае, если используется динамическое ценообразование на уровне энергосистемы. Тогда следует назначать использование бытовых приборов на те периоды, когда стоимость энергии минимальна. Заметим, что подобный механизм также может быть полезен в сочетании с локальными ВИЭ, когда использование некоторых устройств может быть запланировано на периоды их высокой производительности. Таким образом, планирование использования бытовой техники может выступать в качестве альтернативы установке устройств хранения энергии.

Помимо установки специализированных батарей, существует возможность использовать электро- или гибридные автомобили в качестве средства хранения энергии [9]. Разумеется, мобильный характер такого устройства приводит к тому, что оно не всегда будет доступно. Кроме того, пользователи нередко ожидают, что автомобиль должен быть заряжен, когда они желают его использовать, дополнительно ограничивая потенциал такого средства хранения.

Все описанные подходы выиграют от прогнозирования таких факторов, как производительность ВИЭ, распределение спроса по времени и используемым устройствам, цена на энергию в случае ее изменчивости.

2.3.2 Структура эксперимента по выделению наиболее перспективных подходов

Прежде чем приступить к планированию и проведению эксперимента, необходимо сформулировать тестируемую гипотезу или подход, а также определить базовый случай. Например, может сравниваться система управления

энергией, использующая солнечные панели, со встроенными батареями и без. От выбора базовой конфигурации системы будут зависеть рассчитанные преимущества и недостатки, поэтому важно проанализировать ситуацию, чтобы определить ее корректно.

В качестве следующего шага требуется выделить особенности функционирования выбранного подхода с помощью анализа предметной области. В частности, необходимо определить, какие факторы способствуют и препятствуют его работе. В качестве примера можно привести ВИЭ: для конкретной технологии следует определить, при каких условиях окружающей среды она способна производить энергию.

На основе выделенных факторов необходимо создать имитационную модель, генерирующую нужные показатели. В том числе следует определить конфигурацию эксперимента, в частности, входные данные об окружающей среде, такие как температура, влажность, облачность, скорость ветра и другие, в зависимости от требуемых для расчета показателей. Также требуется выбрать длительность проведения эксперимента и интервал между двумя шагами моделирования.

Наконец, необходимо подключить к модели прототип системы, провести имитационный эксперимент и проанализировать его результаты.

ВЫВОДЫ

Во второй главе были решены следующие задачи:

1. Охарактеризована задача обеспечения энергоэффективности умного дома, в том числе среда ее работы и основные компоненты.
2. Выделены требования к таким системам, их показатели качества.
3. Разработаны возможные подходы к обеспечению энергоэффективности, а также структура экспериментов для выделения наилучших.

ГЛАВА 3

ИСПОЛЬЗОВАНИЕ YANDEX IOT CORE И NODE-RED ДЛЯ ТЕСТИРОВАНИЯ СИСТЕМ ИНТЕРНЕТА ВЕЩЕЙ

3.1 Имитационная модель окружающей среды

В главе 1 был предложен подход к моделированию и тестированию систем интернета вещей, при котором она отделяется от программы, отвечающей за генерацию и поставку данных. В то время как задачей тестируемой программы является корректная реакция на поступающие от датчиков данные об окружающей среде, от тестирующей системы требуется генерация и передача этих данных, а также сбор и анализ информации о реакции на них. Под моделированием понимается исследование поведения системы косвенным путем, то есть при помощи анализа вспомогательных объектов [2, с. 15].

Хотя полное разделение двух программ невозможно, поскольку тестирующая система должна знать о том, какие данные и в какой форме поступают на входы и выходы, тестируемая система может быть подключена таким образом, что с ее точки зрения поступающие сгенерированные данные неотличимы от реальных. Воспользуемся тем фактом, что любая система интернета вещей по определению должна соединяться с физическими устройствами, то есть в ней присутствуют входы и выходы, через которые передаются сообщения с использованием определенного протокола. Кроме того, при понимании области использования систем, которые необходимо протестировать или исследовать возможно сузить количество моделируемых показателей.

Рассмотрим ключевые детали реализации тестирующей программы. Важной частью является модель окружающей среды — набор правил, по которым генерируются показания датчиков с учетом как данных о внешней среде, так и состояний устройств системы, причем необходимо достичь реализма в определенном смысле, обычно означающем, что результаты работы модели должны совпадать с аналогичным сценарием в реальном мире с заданной степенью точности. Таким образом, задача построения имитационной модели состоит в нахождении приближенных к реальности правил, по которым формируются данные. Выбор правил зависит от предметной области, доступных вычислительных ресурсов и необходимой точности моделирования. Точность генерируемых данных зависит не только от правил, но и от данных, к которым они применяются. Поэтому необходимо решить вопрос об источнике данных внешней среды. Возможным вариантом является использование исторических данных. Кроме того, может быть

известно ожидаемое распределение данных, тогда можно использовать методы генерации случайных величин. В то же время нередко бывает и так, что требуется проверить систему в заданных условиях: например, требуется протестировать реакцию системы на резкое изменение условий окружающей среды, возникновение нежелательной ситуации и так далее. В таком случае остается только использовать требуемые формулы для генерации данных.

Подход к моделированию условий окружающей среды основывается на общих принципах имитационного моделирования. Для некоторых показателей имеет смысл использовать методы системной динамики, в особенности носящих непрерывный характер, таких как температура. Для дискретных показателей представляется более подходящим агентное или событийное моделирование. Моделирование на основе использования агентов в особенности полезно, когда в системе присутствуют действующие лица, характеризующиеся своим поведением. Кроме того, зачастую оказывается выгодно использовать гибридный подход, совмещающий два или более «чистых» метода имитационного моделирования.

3.1.1 Моделирование времени

Особенным показателем в контексте имитационного моделирования систем интернета вещей является внутреннее время. Немаловажным свойством систем интернета вещей и данных, с которыми они работают, является большая длительность обрабатываемых и управляемых процессов. К примеру, нагревание комнаты может занимать несколько часов, а значительные изменения температуры окружающей среды наблюдаются в масштабе от дня до года в зависимости от требуемой амплитуды. Поэтому требуется моделирование внутрисистемного времени со значительным ускорением относительно внешнего. Под внешним временем будем понимать время, которое занимает моделирование с точки зрения внешнего наблюдателя. В частности, удобным будет измерять время одного шага симуляции: может быть одна секунда, одна десятая или сотая секунды и так далее. Внутреннее, или системное, время — это время, которое проходит внутри модели, иначе говоря, с точки зрения тестируемой системы. Для моделирования наиболее часто необходимо «ускорить» системное время, то есть если за один шаг симуляции для внешнего наблюдателя проходит одна секунда, то для системы может проходить одна минута, один час и так далее. При этом ускорение ограничено как возможностями используемой среды, доступными в ней ресурсами, так и объемом и трудоемкостью вычислений на каждом шаге.

В случае использования наиболее простого представления с фиксированным шагом время можно представить следующим образом:

$$t = i * step, i = \{0, \dots, n\}, \quad (1)$$

где i — текущая итерация, $step$ — размер временного интервала, n — общее количество шагов.

Для того, чтобы избежать вычислений с числами с плавающей точкой, возможно также вести учет шагов от нуля до заданного общего числа. В таком случае время в часах или других единицах выражается через количество шагов за такую единицу времени, являющееся отдельным параметром модели.

Выбор промежутка внутрисистемного времени, который занимает один шаг, зависит от многих факторов. С одной стороны, для повышения точности моделирования необходимо в общем случае шаг уменьшить. Иногда бывает возможность рассчитать необходимую длину из требуемой точности. На реалистичность моделирования также влияет соответствие временного промежутка частоте принятия решений системой: если некоторые события слишком коротки по времени, то они могут быть вовсе не зафиксированы при моделировании.

С другой стороны, существуют факторы, регулирующие минимально возможную длину шага, в первую очередь связанные с доступными для проведения моделирования ресурсами. Если моделирование одного шага достаточно трудоемко, то достичь достаточно долгих периодов моделирования в целом возможно только увеличивая общее внешнее время. Кроме того, степень, в которой тестируемая система может увеличить скорость реагирования на внешние по отношению к ней события, ограничена, что устанавливает верхнюю границу «ускорения» внутреннего времени по отношению ко внешнему.

3.1.2 Моделирование условий внешней среды

Системы управления энергопотреблением требуют данных как о состоянии и работе подконтрольных им устройств, так и об окружающей среде. Для корректного моделирования таких показателей требуется не только использовать верные способы их расчета, но и задать верные константные параметры. Такие параметры могут быть получены путем изучения статистических данных об окружающей среде в заданной местности. В то же время, нередко требуется изучить

реакцию системы на особенную ситуацию. В таком случае параметры могут быть входными данными теста или эксперимента.

В контексте обеспечения эффективного использования энергии для обогрева помещения наиболее важным показателем внешней среды является ее температура. Заметим, что он является внутренним по отношению к тестирующей системе (имитационной модели), на работу тестируемой системы он оказывает влияние опосредованно, в частности, через расчет внутренней температуры контролируемого помещения, влияние на значение которой он оказывает, увеличивая или уменьшая потери тепла.

Учитывая, что промежуток внутрисистемного времени, в течение которого будет проводиться эксперимент, может значительно различаться в зависимости от задачи, моделирование температуры окружающей среды принимает различные формы. В случае коротких промежутков времени, применяемых, в частности, для проверки нормальной работы системы, оценки качества работы отдельных ее компонентов и так далее, простым способом моделирования является использование средней температуры, сложенной с суточным колебанием.

$$T_{\text{внеш}}(t) = T_{\text{ср}} + A \cos(\omega t + \phi), \quad (2)$$

где $T_{\text{внеш}}$ — среднесуточная температура, A — амплитуда колебаний температуры, $\omega = \frac{2\pi}{24}$ — частота колебаний, ϕ — фаза. Более интуитивно понятным является представление

$$T_{\text{внеш}}(t) = T_{\text{ср}} + A \cos(\omega(t - h_0)), \quad (3)$$

где h_0 — время суток, в которое температура должна быть максимальной.

На более длительных промежутках времени играть роль начинают годовые колебания температуры, а также фактор случайности. Случайность может играть роль как в выборе средней температуры суток, так и в процессе ее колебаний.

3.1.3 Моделирование тепловых потерь и работы устройств ОВиК

Для расчета затрат на отопление дома необходимо определить некоторые параметры, в частности, его геометрию и использованные материалы. Под геометрией понимается форма, измерения стен и крыши, количество и размеры окон и тому подобные параметры жилища. В частности, задаются ширина и длина

дома, а также его высота без крыши. Для крыши задается угол ее наклона. Для окон задается ширина и высота. Кроме того, дополнительным параметром окон и стен является их толщина. Используемые материалы задаются через их тепловые свойства, откуда можно рассчитать тепловое сопротивление дома, зная толщину стен и окон. Используемые параметры показаны в таблице 3.1.

Таблица 3.1 — Используемые в имитационном моделировании параметры

Параметр	Значение	Единица измерения	Параметр	Значение	Единица измерения
Параметры дома					
$l_{\text{дома}}$	30	м	$w_{\text{окна}}$	1	м
$w_{\text{дома}}$	10	м	$k_{\text{стены}}$	0,038	Вт/м·К
$h_{\text{дома}}$	4	м	$L_{\text{стены}}$	0,2	м
$\alpha_{\text{крыши}}$	30	°	$k_{\text{окна}}$	0,78	Вт/м·К
$N_{\text{окон}}$	6	ед.	$L_{\text{окна}}$	0,1	м
$h_{\text{окна}}$	1	м	$\rho_{\text{воздуха}}$	1,225	кг/м ³
Параметры нагревателя					
L	1	кг/с	c	1005,4	Дж/кг·К
$T_{\text{нагр}}$	50	°С	p	0.1616	у. е./кВт·ч

Для расчета тепловых потерь необходимо знать площадь поверхности, соприкасающейся с внешней средой. У дома имеется два типа поверхностей: поверхность стены и поверхность окна. Исходя из заданных параметров, возможно рассчитать площадь окон по формуле (4).

$$S_{\text{окон}} = N_{\text{окон}} * h_{\text{окна}} * w_{\text{окна}}, \quad (4)$$

где $N_{\text{окон}}$ — количество окон в доме, $h_{\text{окна}}$ — высота окна, $w_{\text{окна}}$ — ширина окна.

Оставшаяся площадь внешней поверхности дома является площадью стен. В качестве упрощения будем считать, что крыши дома имеет ту же толщину и состоит из того же материала, что и его стены. Учитывая площадь окон, рассчитаем ее по формуле (5).

$$S_{\text{стен}} = 2h_{\text{дома}}(w_{\text{дома}} + l_{\text{дома}}) + \frac{1}{2}w_{\text{дома}}^2 \tan(\alpha_{\text{крыши}}) + \frac{w_{\text{дома}}l_{\text{дома}}}{\sin(\alpha_{\text{крыши}})} - S_{\text{окон}} \quad (5)$$

где $l_{\text{дома}}$ — длина дома, $w_{\text{дома}}$ — ширина дома, $h_{\text{дома}}$ — высота дома без крыши, $\alpha_{\text{крыши}}$ — угол наклона крыши.

Зная площадь стен дома, рассчитаем тепловое сопротивление по формуле (6).

$$R_{\text{стен}} = \frac{L_{\text{стены}}}{k_{\text{стены}} * S_{\text{стен}}}, \quad (6)$$

где $L_{\text{стены}}$ — толщина стены, $k_{\text{стены}}$ — коэффициент теплопроводности материала стен. Формула (7) позволяет рассчитать тепловое сопротивление окон аналогичным образом.

$$R_{\text{окон}} = \frac{L_{\text{окна}}}{k_{\text{окна}} * S_{\text{окон}}}, \quad (7)$$

где $L_{\text{окна}}$ — толщина окна, $k_{\text{окна}}$ — коэффициент теплопроводности материала окон. Эквивалентное тепловое сопротивление рассчитаем по формуле (8) исходя из теплового сопротивления стен и окон.

$$R_{\text{экв}} = \frac{R_{\text{стен}} * R_{\text{окон}}}{R_{\text{стен}} + R_{\text{окон}}}. \quad (8)$$

Скорость нагревания воздуха зависит от его массы, поэтому для конечного расчета внутренней температуры при использовании нагревателя потребуется рассчитать массу воздуха в доме. Плотность воздуха показана в таблице [II] и соответствует значению на уровне моря.

$$M_{\text{возд}} = \left(h_{\text{дома}} l_{\text{дома}} w_{\text{дома}} + \frac{1}{4} w_{\text{дома}}^2 l_{\text{дома}} \tan(\alpha_{\text{крыши}}) \right) \rho_{\text{возд}}, \quad (9)$$

где $\rho_{\text{возд}}$ — плотность воздуха.

С другой стороны, необходимо задать характеристики используемого обогревателя, чтобы определить количество подаваемой тепловой энергии. Соотношение между подачей тепла от обогревателя и потерей тепла из-за внешней среды определяет направление и скорость изменения температуры внутри помещения. Простой моделью будет устройство, определяющееся температурой подаваемого горячего воздуха и скоростью его подачи. Значения использованных параметров показаны в таблице [III].

Для вычисления температуры на следующем шаге моделирования необходимо рассчитать скорость теплового потока от обогревателя, то есть количество полученной от него тепловой энергии за единицу времени.

$$\frac{Q_{\text{нагр}}}{\Delta t} = (T_{\text{нагр}} - T_{\text{внутр}})Lc, \quad (10)$$

где $T_{\text{нагр}}$ — температура воздуха внутри обогревателя, L — скорость потока подаваемого им воздуха, c — удельная теплоемкость воздуха, Δt — временной промежуток. Кроме того, понадобится такой же расчет для потерь.

$$\frac{Q_{\text{потерь}}}{\Delta t} = \frac{T_{\text{внутр}} - T_{\text{внеш}}}{R_{\text{экв}}}. \quad (11)$$

Наконец, можем найти, насколько нагрелся или остыл дом, то есть температуру воздуха внутри на следующем шаге.

$$T_{\text{внутр}}(t + \Delta t) = cM_{\text{возд}} \left(\frac{Q_{\text{нагр}}}{\Delta t} - \frac{Q_{\text{потерь}}}{\Delta t} \right) \Delta t. \quad (12)$$

3.1.4 Моделирование поведения жителей дома

В контексте управления потреблением энергии, в особенности, когда речь идет об устройствах ОВиК, имеет смысл выделить три возможных состояния, связанных с поведением людей: «дом пуст» относится к случаю, когда внутри нет ни одного человека; «жители активны», когда хотя бы один человек находится в доме и бодрствует; «жители спят», когда нет ни одного бодрствующего человека в доме.

Для определения наличия хотя бы одного активного человека в подконтрольном помещении можно использовать датчики движения или присутствия [20]. Распространенным механизмом их работы является анализ инфракрасного излучения. В частности, к этому типу относятся пассивные инфракрасные датчики. Заметим, что для использования в системах домашней автоматизации стоит выбирать более дешевые, простые в установке, наименее навязчивые устройства, даже если это означает определенные потери в точности в

виде ложноположительных или ложноотрицательных результатов, ведь эта сфера отличается относительно невысокой степенью ответственности.

Для моделирования поведения такого датчика требуется разработка правила, по которому определяется активность пользователей. В качестве наиболее простой модели можно предложить отправление сообщений в рамках заданного промежутка времени. В пределах этого промежутка эмуляция датчика состоит в том, чтобы отправлять сигнал на каждый шаг моделирования, иначе датчик сохраняет молчание. Более сложная модель может включать в себя вероятностные отклонения от такого заданного расписания, однако это требует дополнительных исследований поведения людей. Пример функции, моделирующей показания датчика присутствия, показан в формуле (13).

$$f(h) = \begin{cases} 1, & h \in A \\ 0, & h \notin A \end{cases} \quad (13)$$

где h — текущее внутрисистемное время, A — множество шагов, на которых датчик присутствия должен сработать. Чтобы отделить друг от друга состояния «жители спят» и «дом пуст», следует использовать дополнительный датчик на входной двери [20]. Такой датчик дает данные о входах и выходах через эту дверь, из чего система может делать выводы о наличии людей в доме. Математическая модель такого датчика аналогична модели датчика движения, за исключением наличия нескольких состояний, отраженных в формуле (14). Развитием этой идеи является использование камер, хотя такой сценарий имеет смысл в основном в том случае, когда есть другие цели их использования в дополнение к управлению ОВиК, ведь их покупка и установка оказывается относительно более дорогой.

$$f(h) = \begin{cases} 1, & h \in A \\ -1, & h \in B \\ 0, & h \notin A \cup B \end{cases}, \quad (14)$$

где h — текущее внутрисистемное время, A — множество шагов, на которых датчик должен реагировать на вход через дверь, B — множество шагов, на которых датчик должен реагировать на выход. Моделирование такого дверного датчика можно проводить аналогично описанному выше случаю для датчиков движения. Следует составить расписание приходов-уходов для жителей дома и в соответствующие этим точкам шаги отправлять нужные сообщения. При этом требуется синхронизация с эмулятором датчиков движения, чтобы показания датчиков были согласованы.

3.2 Оценка работы тестируемой системы

Для отслеживания работы системы необходимо собирать статистику о ее поведении: например, как часто и в какие моменты она включала и отключала отопление. Из этих данных возможно рассчитать такие показатели, как потребление электроэнергии. В дополнение к информации о поведении системы также доступны сгенерированные данные и параметры системы, такие как время или температура в помещении. Например, комфорт пользователей может определяться как способность системы поддерживать температуру в пределах заданных значений. Для этого используется сгенерированные показания датчиков, к которым система также имеет доступ.

Учитывая, что с точки зрения пользователя целью системы управления энергопотреблением является минимизация затрат на энергию, в качестве основной оценки качества работы таких систем предложим общую стоимость использованной энергии. Для упрощения расчетов будем считать, что отсутствуют потери при преобразованиях одного вида энергии в другую. В частности, для вычисления стоимости потребленной энергии используем константу — стоимость одного киловатт-часа. Таким образом, для расчета стоимости энергии можно использовать формулу (15). Стоит отметить, что задача оценки эффективности использования энергии усложняется, если предполагается, что стоимость энергии является функцией от времени, что может наблюдаться в случае, когда цена динамически зависит от спроса.

$$P = p \sum E_i, \quad (15)$$

где p — цена за единицу энергии, E_i — энергия, потребленная i -м устройством.

В Node-RED устройства, потребляющие энергию, были подключены к узлу расчета ее суммарной стоимости.

Менее очевидной является оценка комфорта пользователей системы. В данной работе требования по поддержке того или иного температурного режима и тому подобные запросы были инкорпорированы в систему как ограничивающие условия. Тем не менее, по выходным данным проведенного эксперимента возможно рассчитать более сложные оценки, если возникает такая необходимость. В частности, наиболее часто после использования требований как ограничений в системе предлагается оценка, основанная на линейном отклонении от целевых

значений [14]. В качестве альтернативы можно также предложить расчет доли времени, проведенной в состоянии, отличном от желаемого. Такая оценка является немного более простой для расчета, поскольку на каждом временном шаге она имеет двоичное значение.

3.3 Проведение экспериментов

Для проведения экспериментов над тестируемой системой необходимо подключить систему таким образом, чтобы она получала необходимые ей сгенерированные данные. Соответственно, набор моделируемых показателей отчасти определяется областью применения тестируемой системы, отчасти — реализацией конкретной системы или прототипа. Таким же образом можно рассуждать в отношении вычисления оценок качества работы системы, с тем отличием, что на них влияют выходные данные.

В дополнение к вопросу подключения и передачи данных между средой моделирования и тестируемой системой следует коснуться разработки пользовательского интерфейса, через который будет проводиться управление процессом имитационного исследования. В частности, такой интерфейс имеет две основные компоненты: элементы настройки и управления экспериментом, а также представление выходных данных процесса симуляции, включая возможности экспортировать результаты для анализа.

Использование Node-RED в сочетании с Yandex IoT Core позволяет решить все вышеописанные вопросы, от моделирования данных до выдачи результатов. Среда имитационного моделирования, включающая в себя генерацию датчиков, подключение системы, визуализацию информации и пользовательский интерфейс, была реализована с помощью Node-RED. Для сбора и передачи данных был подключен Yandex IoT Core. Далее опишем ключевые детали проектирования и реализации.

Одним из немаловажных преимуществ Node-RED является то, что с помощью этого инструмента возможно создать простой прототип системы в той же среде, в которой ведется или будет вестись основная разработка. Средства визуального программирования Node-RED в сочетании с использованием облачных технологий, в частности, Yandex IoT Core, позволяют создавать прототипы систем интернета вещей без использования реальных устройств, что позволяет проработать архитектуру системы до ее физической реализации. Кроме того, Node-RED также предоставляет средства для простой визуализации получившейся системы, что

позволяет создать прототип графического интерфейса в рамках этой же среды. Заметим, что эти же свойства делают особенно удобным тестирование такой системы в рамках имитационных экспериментов.

Для исполнения среды Node-RED была создана виртуальная машина в облачном сервисе Yandex Compute Cloud. Этот сервис является частью Yandex Cloud и предоставляет масштабируемые вычислительные мощности для создания виртуальных машин и управления ими [4]. Данный сервис предлагает широкий выбор настроек виртуальных машин, от различных операционных систем до тонкой настройки используемых ресурсов. Также был создан реестр системы в Yandex IoT Core, включающий в себя несколько устройств. Соединение с этими объектами было установлено с использованием узлов MQTT Input и MQTT Output.

3.3.1 Генерация показаний датчиков с помощью Node-RED

Исходные данные для генерации могут быть помещены в каждый отдельный узел, однако в данной работе для этого был выделен отдельный узел для конфигурации. Неизменяемые данные задаются при развертывании потока, при этом есть возможность установить часть данных в начальное для эксперимента состояние через графический интерфейс. Возможности по настройке системы пользователем будут подробнее описаны в подразделе 3.3.3.

Прежде чем перейти к описанию процесса для датчиков, коснемся реализации моделирования времени с помощью Node-RED, хотя время в представленном в данной работе виде и представляет интерес в первую очередь при проведении имитационных экспериментов. Интервал внешнего времени поддерживается с помощью узла Inject. Поскольку поток все время активен, а хотелось бы уметь запускать эксперимент и просматривать его результаты, был добавлен узел, считающий шаги симуляции в пределах от нуля до заданного общего количества. По достижении последнего шага узел выдает сигнал об окончании эксперимента на отдельный выход. Для удобства демонстрации также использовался узел, отвечающий за получение внутрисистемного времени в часах. Реализация узлов, отвечающие за управление виртуальным временем, показаны в приложении А.

Что касается эмуляции работы датчиков, реализация этого процесса зависит от характера его показаний. В частности, их можно классифицировать на дискретные и непрерывные. К дискретным отнесем те датчики, которые передают данные, относящиеся к конечному множеству возможных сообщений. Если же

показания датчиков выбираются из некоторого числового промежутка значений, то такой датчик назовем непрерывным.

Обычно дискретные датчики возникают в том случае, когда требуется реагирование на некоторое событие. Наиболее простым вариантом является датчик, множество значений которого состоит из одного элемента, означающего, что событие произошло. Примером такого датчика может быть датчик присутствия, математическая модель которого была описана в подразделе 3.1.4. В Node-RED такой датчик моделируется путем проверки условия, причем узел выдает сообщение, если условие оказывается истинным. Аналогичным образом можно действовать для датчиков, имеющих более одного значения, только необходимо проверить больше условий.

Непрерывные датчики, напротив, реже реагируют на события, а зачастую должны постоянно передавать измеряемые им значения через некоторый заданный промежуток времени. Примером такого датчика может служить датчик температуры. В Node-RED расчет температуры выполняется в нескольких узлах, поскольку на ее значение влияют многие факторы, как было показано в подразделе 3.1.2 и 3.1.3. Некоторые из этих узлов относятся к обработке данных, полученных от тестируемой системы. Эти узлы описываются в подразделе 3.3.2. Тем не менее, отдельный узел был выделен для передачи данных о температуре на каждом шаге симуляции, ведь предполагается, что система должна регулярно получать новые данные о температуре внутри подконтрольного ей помещения. Набор узлов, предназначенных для генерации показаний датчиков и управления временем, показан на рисунке 3.1. Их внутреннее устройство находится в приложении А.

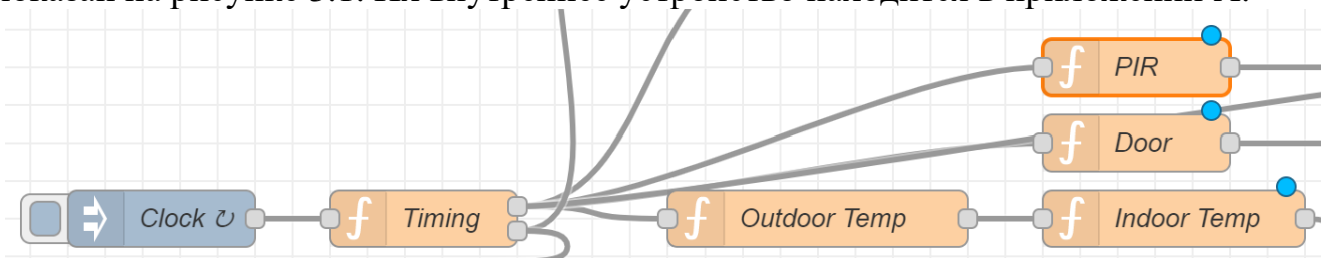


Рисунок 3.1 — Основные узлы генерации показаний датчиков

Модель окружающего мира может быть в любой момент дополнена, если появилась необходимость в моделировании нового показателя. В систему добавляется новый узел, который соединяется с узлом, генерирующим время. Наиболее подходящим узлом, особенно в случае сложных правил получения данных, является Function. Внутри этого узла записывается алгоритм. При этом могут быть также добавлены другие узлы для расчета, если требуется учет каких-либо новых факторов.

3.3.2 Разработка и подключение тестируемой системы

В отличие от случая, когда тестируется готовый программный продукт, при тестировании теоретически разработанных подходов требуется сперва создать прототип системы, использующей выбранный подход. Поскольку реализация сильно связана с конкретным исследованием, то коснемся только общих соображений. При использовании в сфере интернета вещей Node-RED предлагает несколько преимуществ по сравнению с другими вариантами. Во-первых, Node-RED является инструментом для визуального программирования, что позволяет не только сделать простую автоматизацию даже тем, кто не знаком с разработкой, но и облегчить создание прототипов, интегрировать все стадии разработки системы. При этом Node-RED достаточно гибок, чтобы решать большинство встающих в области интернета вещей задач. Во-вторых, Node-RED возможно запускать на большом количестве платформ. Сюда относятся как Raspberry Pi, так облачные серверы. Таким образом, разработанный прототип можно использовать как основу для дальнейшей разработки продукта. В-третьих, разработанные сценарии и правила не зависят от физической реализации системы, поскольку Node-RED предоставляет возможности для получения данных и управления абстрактными устройствами, которые могут быть связаны с физическими отдельно.

Что касается подключения системы или прототипа к среде моделирования, требуется интеграция с узлами или потоками как генерации данных, так и анализа и оценки качества, что приблизительно соответствует входам и выходам такой системы. Сначала разберем общий принцип работы с датчиками.

Основным датчиком для работы с устройствами ОВиК в контексте управления энергопотреблением является датчик температуры внутри контролируемого помещения. Типичным использованием такого датчика является включение и отключение обогревателя в зависимости от текущей температуры в помещении и требований по допустимым ее пределам. Возможные узлы для обработки температурных показаний показаны на рисунке 3.2. Их внутреннее устройство показано в приложении Б.

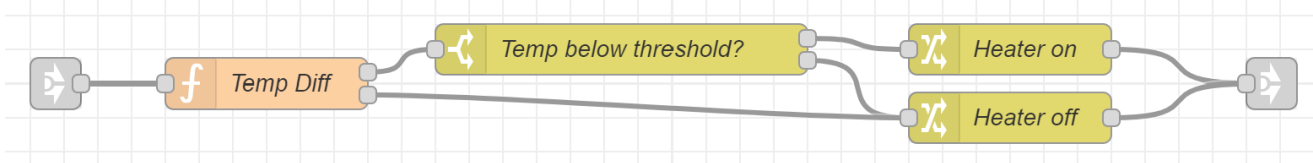


Рисунок 3.2 — Узлы для обработки показаний датчика температуры

Для уточнения необходимости использования энергии имеет смысл использование датчиков присутствия. Например, система может поддерживать

несколько состояний в виде конечного автомата, переходы между которыми осуществляются по сообщениям от датчиков. Управление устройствами в таком случае будет зависеть от установленного состояния. Узлы системы, предназначенные для реагирования на сообщения от таких датчиков, изображены на рисунке 3.3. Их внутреннее устройство показано в приложении Б.

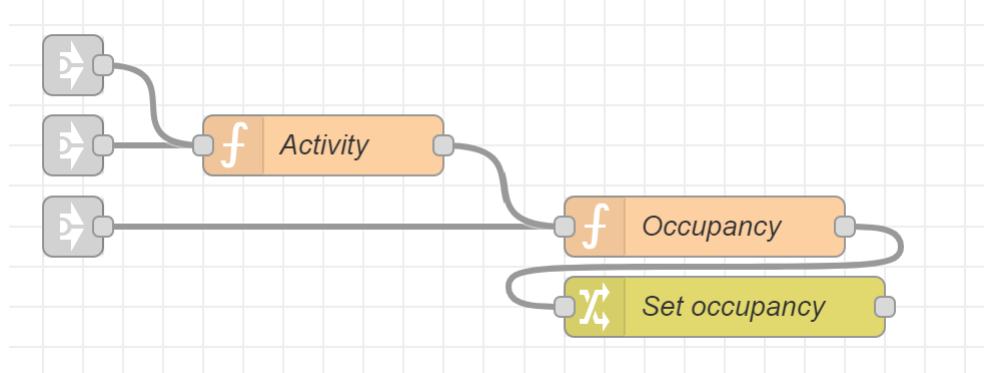


Рисунок 3.3 — Узлы обработки данных от датчиков движения

Немаловажной задачей коммуникативной части тестирующей системы является также получение данных от тестируемой системы. Существуют два направления их использования: во-первых, это анализ качества работы системы; во-вторых, это генерация показателей датчиков. Реализация узлов для обработки данных показана в приложении В, а визуальное представление — на рисунке 3.4.

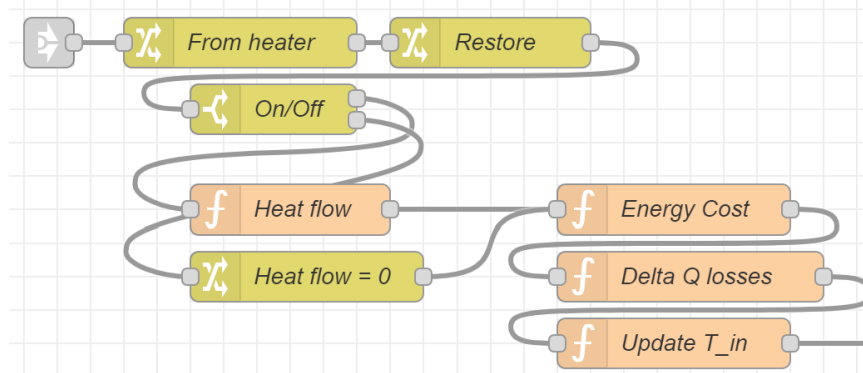


Рисунок 3.4 — Узлы расчета внутренней температуры

3.3.3 Пользовательский интерфейс. Возможности по конфигурации системы пользователем

Для того, чтобы система была более наглядной или интуитивно понятной, желательна визуализация процесса ее работы. В частности, возможно демонстрировать генерируемые показания датчиков или условия окружающей

среды, такие как температура окружающей среды, температура в помещении, или же результаты работы системы, например, потребленная энергия или ее стоимость.

Чтобы визуализировать данные, Node-RED предлагает особые графические узлы, позволяющие создать элементы интерфейса, такие как текстовые поля, графики, на так называемой «панели управления». Например, на графике можно показать изменение температуры или накопление стоимости энергии, а с помощью текста — стоимость энергии или количество шагов моделирования. Для сбора данных использовались MQTT-узлы с подключением к Yandex IoT Core. Узлы, отвечающие за сбор и передачу данных, а также за их отображение на графике, показаны на рисунке 3.5. Их внутреннее устройство показано в приложении Г.

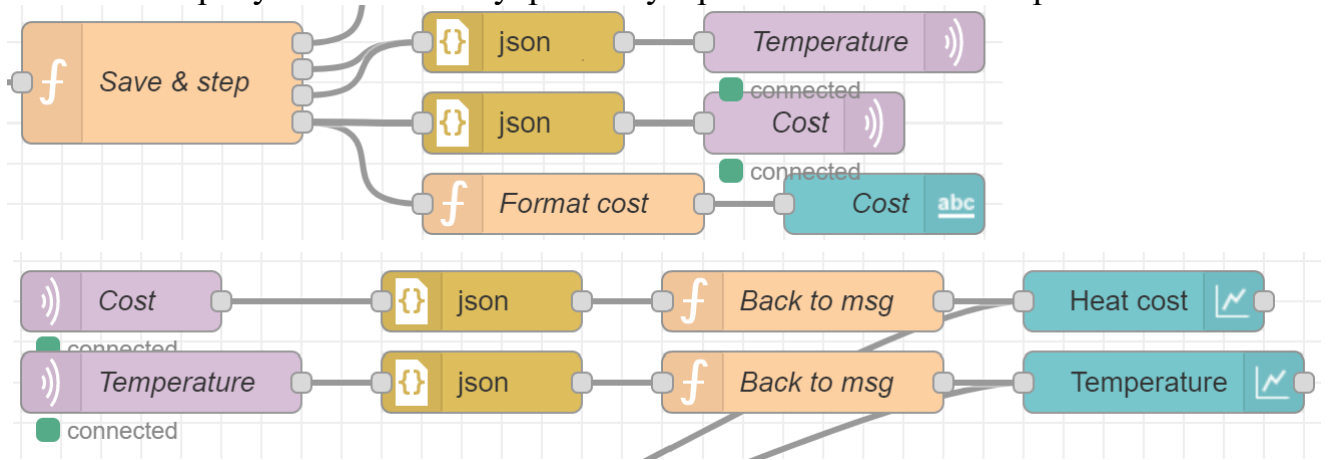


Рисунок 3.5 — Узлы для сбора, передачи, визуализации данных

Помимо визуализации, собранные данные также сохраняются в файл, который также доступен через пользовательский интерфейс. Этот файл создается при запуске эксперимента и дополняется при поступлении новых данных. Тут же можно выгрузить настройки системы, чтобы учесть их при дальнейшем анализе. Узлы для создания, обновления и скачивания файлов показаны на рисунке 3.6.

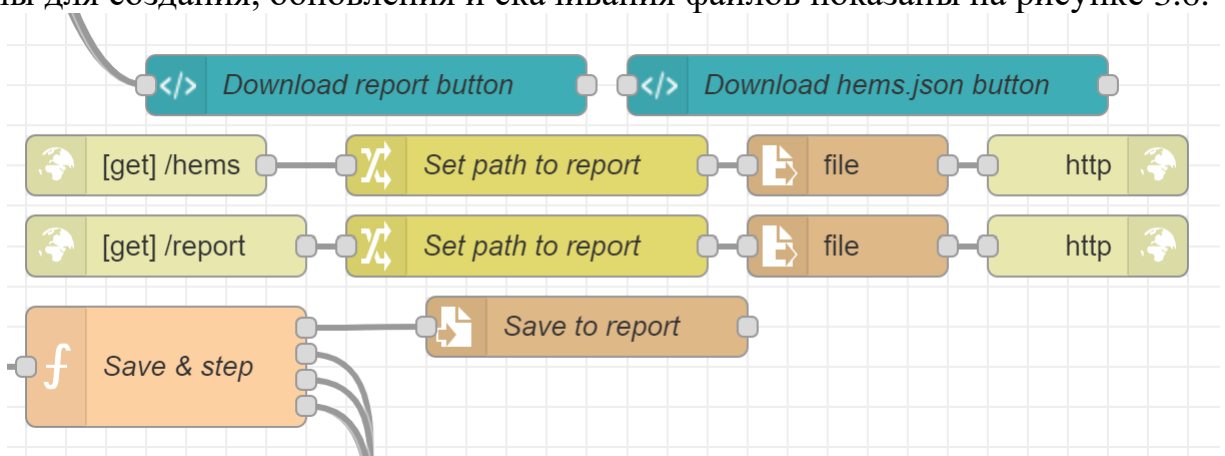


Рисунок 3.6 — Узлы для работы с файлами

На эту же вкладку панели управления была помещена кнопка для запуска моделирования. Таким образом, при запуске эксперимента пользователь может видеть динамическое изменение температуры, потребление энергии, количество пройденных шагов, а по его окончании — выгрузить сгенерированные данные, которые можно затем проанализировать отдельно. На верхнем графике показано накопление общей стоимости затраченной энергии в процессе симуляции, а на нижнем — температура внутри и вне дома. Отметим, что данные для этих графиков отправлялись каждый час виртуального времени, или каждые десять шагов. Это было сделано для того, чтобы не нагружать графическую часть, требующую большого количества ресурсов. Вкладка с элементами графического интерфейса, описанными выше, показана на рисунке 3.7.

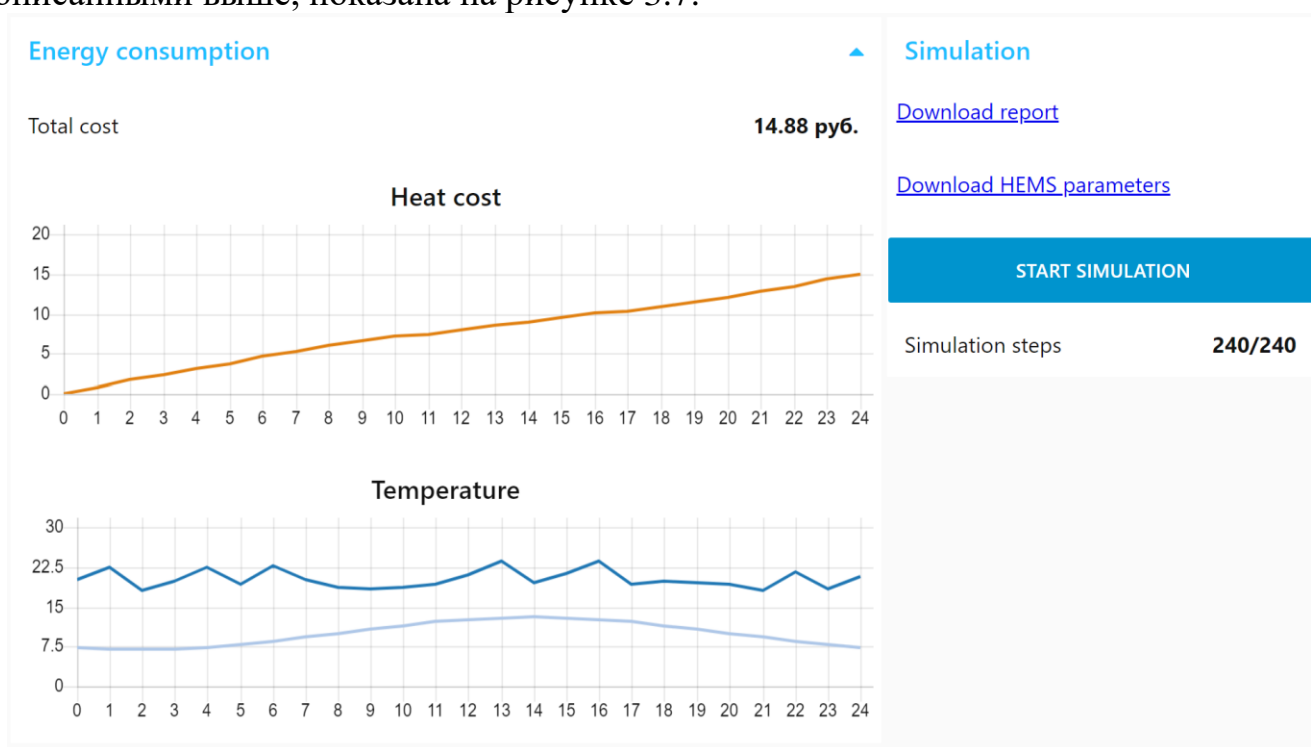


Рисунок 3.7 — Вкладка «Симуляция»

Пользователю также доступны некоторые возможности по конфигурации системы. Пользователь может настроить две группы параметров: параметры моделирования и параметры системы. К первой группе относятся такие параметры, как общее количество шагов, а также их частота, выраженная как количество шагов в одном часе по внутрисистемному времени. Что касается второй группы, то она, в свою очередь, может быть разделена на параметры тестируемой системы, такие как требуемая температура, и настройки включаемых модулей. В частности, пользователь может отключать те или иные функции системы, чтобы проверить реакцию системы, сравнить разные стратегии управления энергопотреблением.

Под часть пользовательского интерфейса, предназначенную для конфигурации, была выделена отдельная вкладка панели управления, показанная на рисунке 3.8, недоступная в то время, когда идет процесс моделирования. Это было сделано для того, чтобы пользователь не мог случайно изменить параметры в ходе проведения эксперимента и таким образом нарушить его выполнение.

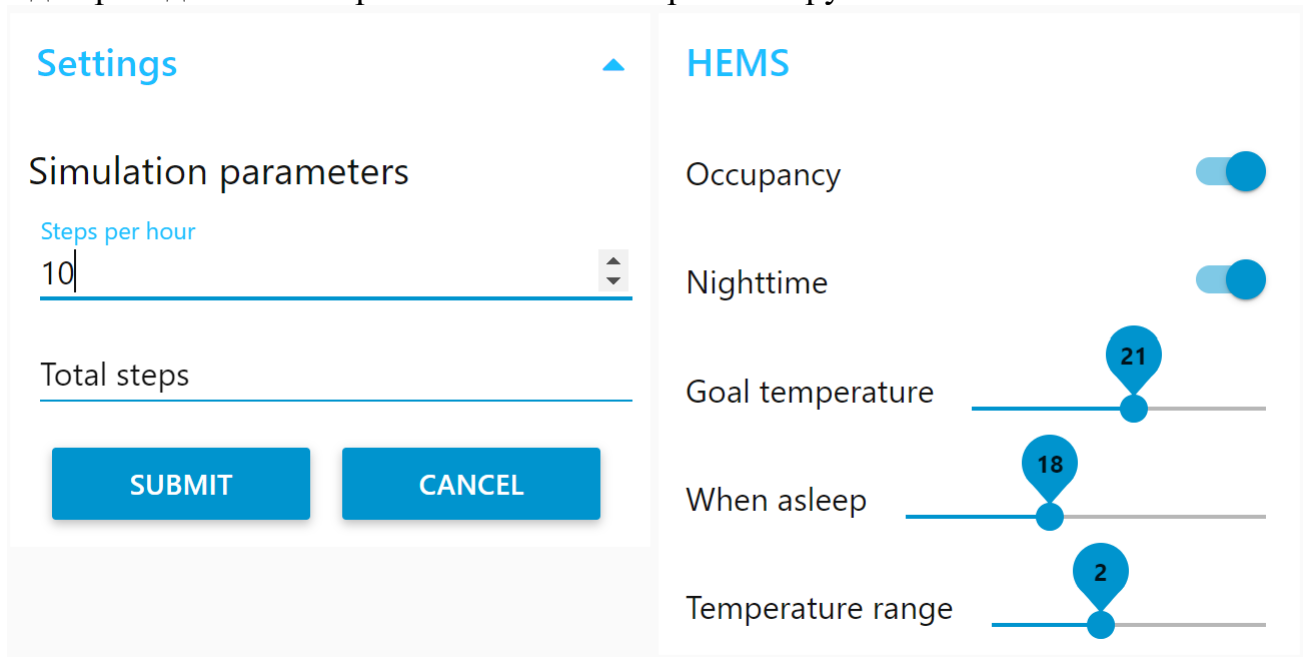


Рисунок 3.8 — Вкладка «Настройки»

ВЫВОДЫ

В третьей главе были решены следующие задачи:

1. Разработана модель окружающей среды.
2. Реализовано взаимодействие с тестируемой системой для оценки ее работы.
3. Создан пользовательский интерфейс для управления проведением экспериментов.

ГЛАВА 4 ПРОВЕДЕНИЕ ИМИТАЦИОННЫХ ЭКСПЕРИМЕНТОВ. ОЦЕНКА РЕЗУЛЬТАТОВ

4.1 Имитационные эксперименты

Для демонстрации возможности проведения имитационных экспериментов, а также анализа и оценки использования для этого Node-RED и Yandex IoT Core, была поставлена следующая подзадача: сравнить несколько алгоритмов управления устройствами ОВиК, отличающиеся степенью и способом использования поведения пользователей системы.

В качестве базового варианта системы управления энергопотреблением была разработана система, включающая и выключающая отопление, основываясь на температуре внутри подконтрольного ей помещения. Система отправляет команду на включение, когда разница между целевой и реальной температурой достигает заданного порога, и на выключение, когда температура вновь находится в нужных пределах. Получаемый температурный режим в течение одних суток показан на рисунке 4.1, где синяя линия показывает температуру в помещении, оранжевая — температуру снаружи, а красным отмечены желаемые пределы.

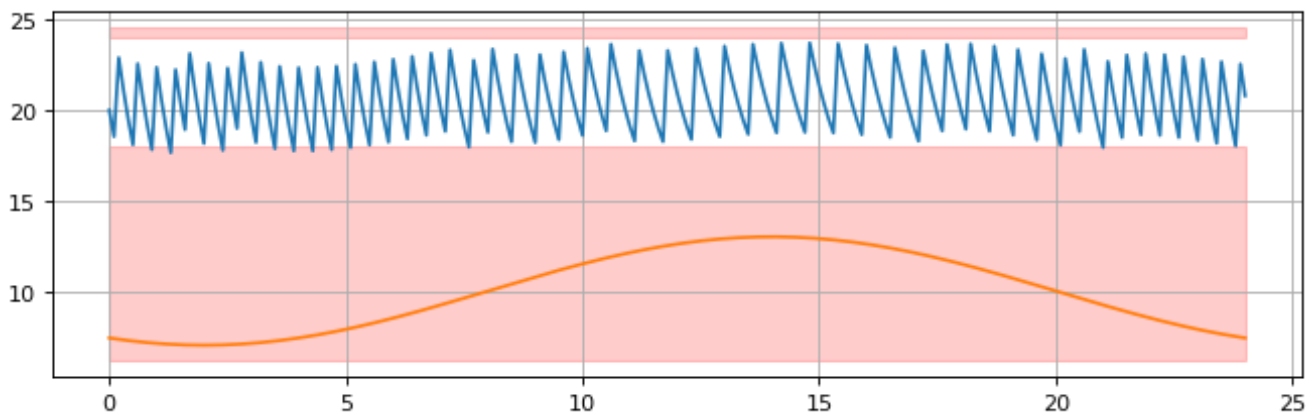


Рисунок 4.1 — Температурный режим базовой системы

Для сравнения были выбраны две возможные стратегии уменьшения энергопотребления, основывающиеся на отслеживании поведения пользователей. Первая из них состоит в том, чтобы наблюдать за активностью людей, находящихся в доме, с использованием датчиков движения. Использование такой стратегии оправдано, поскольку воздействие холода не воздействует на качество сна, в то время как излишне высокая температура его снижает [10]. Вторая использует датчик на входной двери, чтобы отслеживать моменты, когда люди покидают

помещение или входят в него. В таком случае возможно отключать устройства ОВиК, когда они не требуются.

Стратегия, основанная на активности, работает следующим образом. К системе подключается датчик движения. Пока он посылает сигналы, система находится в «активном» состоянии и поддерживает один заданный температурный уровень. После каждого сигнала система ждет определенное время, которое сбрасывает по приходу каждого нового сигнала. Однако если за это время не появилось новых сообщений, то система переходит в «неактивное» состояние, в котором система поддерживает другую, меньшую температуру, как показано на рисунке 4.2.

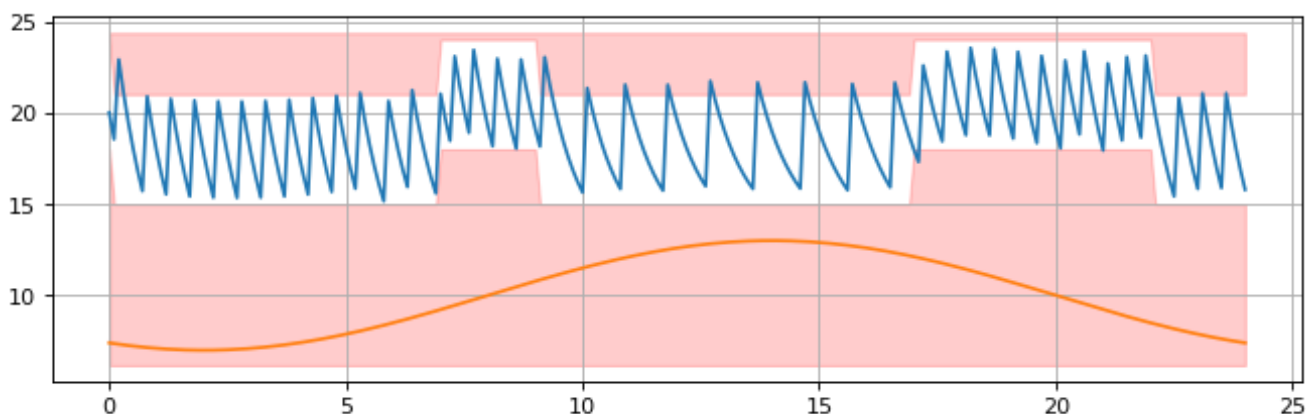


Рисунок 4.2 — Температурный режим с учетом активности пользователей

Другая предложенная стратегия позволяет отслеживать состояние «дом пуст», но не активность пользователей. Для этого к системе подключается датчик, реагирующий на использование входной двери. Таким образом система отключать отопление в те моменты, когда в помещении никого нет. Возможный температурный режим изображен на рисунке 4.3.

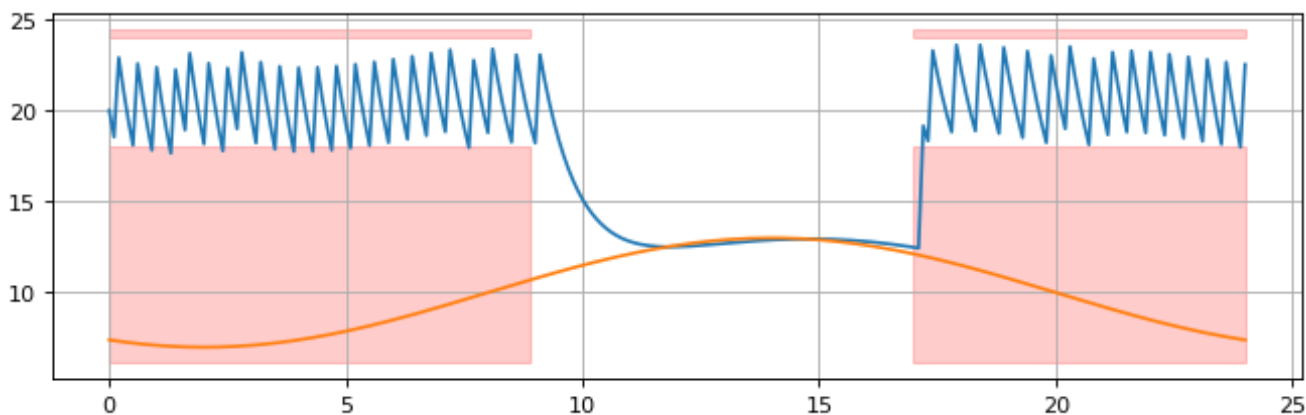


Рисунок 4.3 — Температурный режим с учетом наличия людей в помещении

Как можно заметить, предложенные методы не противоречат друг другу, поэтому имеет смысл также проверить их совместную работу. В таком случае система определяет состояние «дом пуст» аналогично предыдущему сценарию, а в противоположном случае действует в зависимости от показаний датчиков движения. Соответствующий температурный режим в течение суток показан на рисунке 4.4.

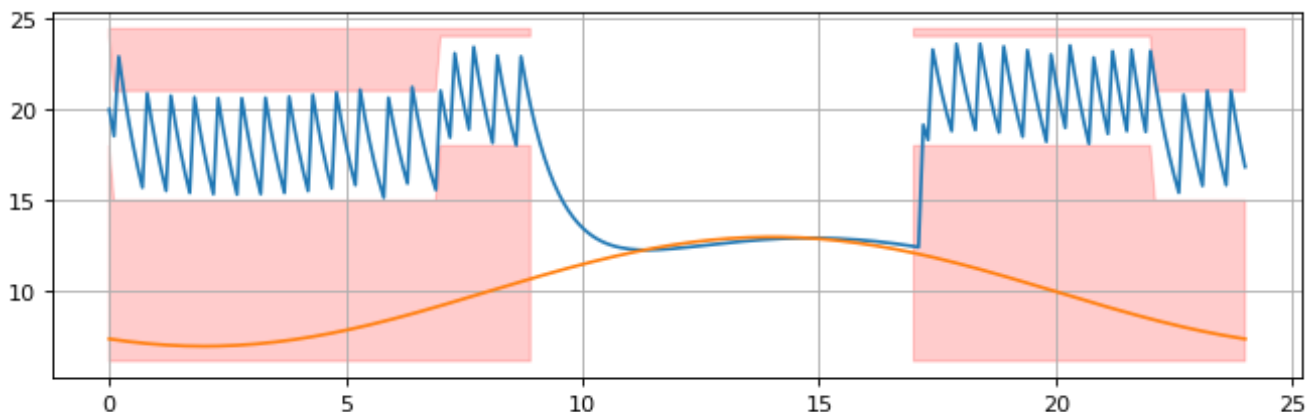


Рисунок 4.4 — Температурный режим при использовании обеих стратегий

Основной метрикой функциональности систем управления энергопотреблением является общая стоимость энергии. Таким образом, чтобы оценить качество и перспективность выделенных подходов, требуется сравнить системы по итоговому количеству потребленной энергии. Для этого поставим такой эксперимент: система работает в течение 24 часов по виртуальному времени. При этом для моделирования был выбран размер шага в 6 минут, то есть 10 шагов в час. Формулы, включающие в себя время, используют именно внутрисистемное время, а не количество шагов, если не указано иначе. Температура окружающей среды изменяется по формуле (16).

$$T_{\text{внеш}} = 10 + 3 \cos\left(\frac{\pi}{12}(t - 14)\right) \quad (16)$$

где t — текущий временной шаг в часах.

Поведение пользователя моделируется следующим образом: считается, что пользователь спит с 22:00 до 7:00, активен с 7:00 до 9:00 и с 17:00 до 22:00, остальное время находится вне контролируемого помещения. Целью пользователя является минимизация энергопотребления устройств ОВиК. Вместе с тем пользователь имеет требования по комфорту, показанные в таблице 4.1, связанные с желаемой температурой. Отметим, что для некоторых из подходов, в частности,

связанных с распознаванием активности жителей, более точной являлась бы метрика с более узкими рамками, однако такой показатель не являлся бы универсальным и не позволил бы сравнить все системы между собой.

Таблица 4.1 — Требования по комфорту

Состояние	Минимальная температура, °С	Максимальная температура, °С
Дом пуст	-	-
Жители активны	18	24
Жители спят	15	24

Гипотеза состоит в том, что базовая система окажется наименее эффективной с этой точки зрения, самой же эффективной окажется система, использующая обе стратегии. В результате проведенного имитационного эксперимента были получены следующие результаты. Общая стоимость затраченной энергии составила 14,87 условных единиц для базовой системы, для стратегии с учетом активности — 12,64, для стратегии с учетом наличия людей в помещении — 11,52, для объединенной стратегии — 9,88. На графике, изображенном на рисунке 4.5, можно также видеть динамическое накопление общей стоимости. В частности, заметим, что стратегии, включающие отслеживание наличия людей в помещении, выигрывают за счет длительных промежутков без употребления энергии. Таким образом, выдвинутая гипотеза подтвердилась.

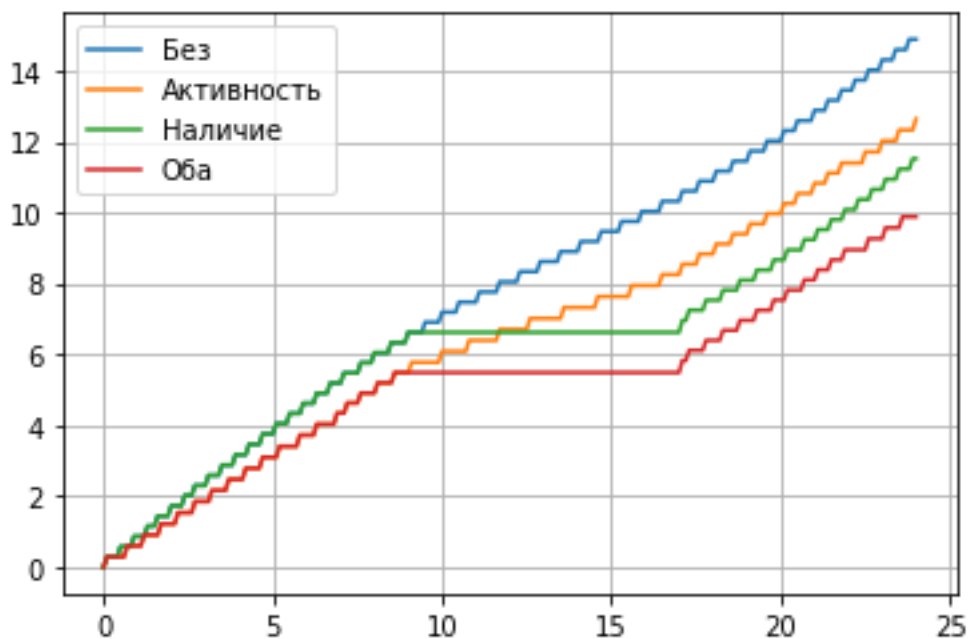


Рисунок 4.5 — Накопление потребления энергии

В качестве дополнительного критерия оценки тестируемых подходов оценим комфорт пользователя. Для того, чтобы сравнить оценки при разной длительности эксперимента, рассчитаем среднее значение по формуле (17).

$$M = \frac{1}{N} \sum_{i=0}^N m(T, h_i), \quad (17)$$

где N — количество шагов моделирования, $m(T, h_i)$ — функция оценки при температуре T на шаге h_i .

Использовалось два варианта расчета i -й оценки. При расчете по формуле (18) оценка дается в зависимости от количества единиц времени, таких как шаг симуляции, в течение которых наблюдалась температура за пределами требуемых значений.

$$m(T, h) = \begin{cases} T_H(h) - T(h), & \text{если } T(h) < T_H(h) \\ 0, & \text{если } T_H(h) \leq T(h) \leq T_B(h) \\ T(h) - T_B(h), & \text{если } T(h) > T_B(h) \end{cases} \quad (18)$$

где $T_H(h)$ — нижняя граница допустимой температуры на шаге тестирования h , $T(h)$ — текущая температура на шаге тестирования h , $T_B(h)$ — верхняя граница допустимой температуры на шаге тестирования h . Другую оценку комфорта можно получить по формуле (19), где рассчитывается среднее отклонение от целевой температурной области.

$$m(T, h) = \begin{cases} 1, & \text{если } T(h) < T_H(h) \text{ или } T(h) > T_B(h) \\ 0, & \text{если } T_H(h) \leq T(h) \leq T_B(h) \end{cases} \quad (19)$$

Полученные результаты показаны в таблице 4.2. Можно видеть, что наилучшие результаты показал базовый вариант, однако третья и четвертая системы незначительно от него отличаются. Отметим, что значительных отклонений от требуемых значений не наблюдалось, что подтверждает возможность поддержки достаточного уровня комфорта всеми системами. Также стоит заметить, что наиболее часто нарушения наблюдаются при переходе между состояниями, как можно видеть на рисунках выше. Это связано с тем, что система не успевает достаточно быстро нагреться или остыть. Возможным решением будет

уточнение условий для управления обогревателем, чтобы система изменяла температуру более плавно.

Таблица 4.2 — Оценка комфорта пользователя для каждой системы

Система	Доля времени вне заданных пределов	Среднее отклонение от заданных пределов
1	0,0415	0,0072
2	0,1112	0,0472
3	0,0456	0,009
4	0,0539	0,0196

В заключение предварительного анализа эксперимента отметим, что ко всем системам предъявлялись одинаковые требования по комфорту, чтобы их возможно было сравнить, из-за чего в вычислениях использовались достаточно мягкие границы. В связи с этим отметим, что первая и третья системы поддерживают достаточно высокую температуру в ночное время, что может быть нежелательным для некоторых пользователей.

Для проверки полученных результатов также был проведен более длительный эксперимент продолжительности в семь внутрисистемных дней. Показания датчиков моделировались аналогичным образом. Полученные данные представлены в таблице 4.3.

Таблица 4.3 — Результаты длительного эксперимента

Система	Потребление энергии		Комфорт	
	Общее	Среднее за день	По времени	По отклонению
1	104,13	14,87	0,0464	0,009
2	87,62	12,52	0,1201	0,0573
3	78,89	11,27	0,05	0,0136
4	70,99	10,14	0,0577	0,0197

Видим, что показатели изменились незначительно, причем если абсолютные значения и изменились, то отношения между рассматриваемыми моделями — нет, из чего можно сделать вывод о справедливости первого эксперимента, а также подтвердить его результаты.

4.2 Выводы в области управления энергопотреблением

Анализ разработанных систем проведем в два этапа. Во-первых, рассмотрим результаты экспериментов с точки зрения обобщенных показателей качества,

предназначенных для оценки программных продуктов в любых сферах. Например, здесь можно использовать показатели качества из стандарта ISO/IEC TR 9126—2:2003: функциональность, надежность, удобство использования, эффективность, возможность сопровождения и переносимость. Особое внимание обратим на функциональность, надежность и эффективность. Причины такого выбора были описаны в подразделе 1.1.1. Разумеется, одним из важных отличий программного продукта от его прототипа является конкретизация требований к нему, поэтому при тестировании готовых систем анализ будет отличаться. Тем не менее, рассмотрим четыре разработанных прототипа на выполнение перечисленных категорий метрик качества.

Основное место занимает проверка программного обеспечения на соответствие требованиям функциональности, однако при этом критерии зачастую требуется разрабатывать в соответствии с конкретной предметной областью и ожидаемым применением разрабатываемой системы. В рассмотренном примере требования по функциональности состояли в том, чтобы, во-первых, снизить энергопотребление, а во-вторых, контролировать нахождение определенных показателей в заданных пределах. Подробнее выводы с этой точки зрения будут описаны ниже, однако можно отметить, что все прототипы выполняют определенные таким образом требования функциональности.

О категориях эффективности и надежности можно судить по появлению некоторых нежелательных событий. Примером может быть постоянное нахождение обогревателя во включенном состоянии, в результате чего температура в помещении выйдет за пределы желаемых пользователем значений. Кроме того, такая система также была бы неэффективной в своем использовании энергии. С другой стороны, примером отказа системы можно назвать состояние, когда система не посылает никаких команд, что также привело бы к выходу температуры за пределы требуемого режима. Ни в одном из экспериментов не наблюдалось значительного выхода за пределы требуемых значений, то есть нежелательных событий не было, что свидетельствует в пользу рассматриваемых прототипов.

Другие категории метрик достаточно сложно оценить программным образом, то есть без участия человека, однако можно провести анализ удобства использования. Можно видеть, что система практически не требует участия человека для своей работы, за исключением установки требуемой температуры. Тем не менее, существует риск неверного использования такой системы, например попытки ручной настройки, что приведет к снижению ее эффективности.

Из анализа общих категорий метрик качества программного обеспечения делаем вывод, что все рассматриваемые системы удовлетворяют базовым

требованиям, однако следует провести более глубокий анализ, чтобы сформулировать рекомендации по выбору между стратегиями.

Напрямую из проведенных экспериментов и рассчитанных метрик потребления энергии и комфорта можно сделать вывод о том, что наилучшей является четвертая система, использующая как факторы для принятия решений активность и присутствие людей в помещении. В таблице 4.4 показано, насколько повысилось или понизилось употребление энергии между каждой парой систем, причем доля считается относительно энергопотребления системы, указанной в столбце. Показатели были рассчитаны по данным недельного эксперимента. Видим, что по сравнению с наименее эффективной системой потребление энергии удалось снизить на 32%. В то же время использование следующей по эффективности стратегии оптимизации энергопотребления повышает его на 11%. Потери в комфорте незначительны и скорее всего останутся незамеченными для конечных пользователей, причем понижение температуры во время сна может даже повышать комфорт, хотя этот фактор в вычислениях не учитывался. Таким образом, с точки зрения функциональности следует порекомендовать систему, использующую данные об активности и присутствии людей в помещении.

Таблица 4.4 — Относительное использование энергии

Система	1	2	3	4
1	100%	119%	132%	147%
2	84%	100%	111%	124%
3	76%	90%	100%	111%
4	68%	81%	90%	100%

С той же точки зрения не рекомендуется использовать вторую систему, которая учитывает только активность пользователей. Хотя она позволяет снизить потребление энергии относительно худшего варианта на 16%, ее использование сопряжено с относительно большими потерями в комфорте. С практической точки зрения также стоит отметить, что использование датчиков движения может приводить к ложноотрицательным результатам, снижая удобство использования системы.

Несмотря на наблюдаемую экономию, стоит отметить, что четвертая система требует относительно большего вмешательства в личную жизнь пользователей, а также затрат на установку. С этой точки зрения стоит отметить, что даже базовая система потенциально снизит затраты энергии по сравнению со случаем, когда отопление включено весь холодный сезон года, и при этом повысит комфорт жителей.

Наконец, отметим, что в некоторых случаях установка подобной системы домашней автоматизации хоть и приведет к снижению энергопотребления, однако в меньшей степени, чем лучшая теплоизоляция, более эффективные приборы и другие инженерно-технические решения. С другой стороны, энергопотребление также можно снизить, если пользователь готов терпеть некоторое снижение комфорта, например установка более низкой целевой температуры.

4.3 Анализ использования Node-RED и Yandex IoT Core

В главе 3 было продемонстрировано, каким образом возможно использовать Node-RED и Yandex IoT Core в имитационном моделировании с целью тестирования и исследования систем интернета вещей, в частности — систем управления энергопотреблением. Проведение экспериментов, описанное в разделе 4.1, подтвердило гипотезу о возможности такого использования перечисленных инструментов. Тем не менее, требуется более глубокий анализ процесса разработки и употребления среды имитационного моделирования, чтобы сделать выводы о преимуществах и недостатках такого подхода в сравнении с другими.

В главе 1 при теоретическом рассмотрении вопроса о тестировании систем интернета вещей путем имитационного моделирования было проанализировано существующее решение в этой области — так называемый Simulink, являющийся частью пакета прикладных программ MATLAB. Этот продукт предназначен в первую очередь для исследований различных физических систем, в том числе может быть применен и для моделирования жилища вместе с системой управления энергопотреблением. Среда Simulink позволяет достичь высокой точности за счет использования сложных численных алгоритмов расчета. Вместе с тем порог вхождения относительно высок с точки зрения как навыков, так и средств. Поэтому в качестве альтернативы был предложен подход, основанный на интеграции Node-RED и Yandex IoT Core.

Основным преимуществом разработанного в данной работе подхода является то, что он более интуитивен, в особенности для людей с опытом разработки программного обеспечения. Для более сложных задач в Node-RED имеется узел Function, который позволяет создать небольшую программу на языке JavaScript. Такие узлы удобны для моделирования показателей, поскольку позволяют записать алгоритм в привычном формате, причем разработанная программа запускается в браузере при активации этого узла сообщением. При этом Node-RED, как и Simulink, позволяет связывать компоненты в сеть визуально, что упрощает

понимание строения системы. Сообщения, передаваемые между узлами, являются объектами JavaScript, которые описываются форматом JSON, что позволяет с легкостью задавать необходимые данные.

Визуальная составляющая и широкая библиотека компонентов также играют роль в обеспечении относительно более легкого вхождения. Благодаря большому количеству доступных узлов часто встречающиеся задачи можно автоматизировать. В особенности это полезно для создания графического интерфейса, поскольку от пользователя требуется только настройка нескольких параметров, в отличие от полной реализации внешнего вида с использованием HTML или сторонних средств разработки. В случае, если необходимо автоматизировать выполнение некоторой операции, для которой не доступны ранее созданные узлы, Node-RED также предлагает возможность добавить собственный узел.

Node-RED также относится к программным продуктам с открытым исходным кодом и не требует денежных затрат для использования, что может быть значительным преимуществом для небольших команд. Кроме того, доступность исходного текста Node-RED означает возможность не только создания своих узлов, но и индивидуальной настройки всей среды.

Что касается интеграции Node-RED с Yandex IoT Core, то она позволяет пересылать данные между различными компонентами, в том числе на облачный сервер. В данной работе использовалась только малая часть всех возможностей — обычный сбор данных для дальнейшей их визуализации, а также использование виртуальной машины для запуска Node-RED. В перспективе использование облачных технологий позволяет сохранять результаты в базу данных, а также проводить их анализ для выявления тенденций и более сложных зависимостей. Использование виртуальной машины также дает возможность удаленного доступа, в том числе упрощая работу и демонстрацию результатов между несколькими участниками. Yandex IoT Core также можно связать с сервисом Yandex Cloud Functions, что позволяет выполнять те или иные операции при получении данных от модели или тестируемой системы.

При всех преимуществах разработанного подхода он также не лишен и недостатков, ограничивающих его применение. В сравнении с Simulink описанная модель проигрывает в точности из-за недоступности сложных численных методов. Разумеется, некоторого улучшения можно добиться, реализовав их самостоятельно, однако один этот факт значительно усложняет разработку до такой степени, что было бы проще использовать Simulink. Тем не менее, стоит отметить, что потенциально такая возможность существует, поскольку Node-RED предлагает

широкие возможности по персонализации и индивидуальной настройке среды работы.

В Node-RED также невозможна реализация моделей, требующих трудоемких вычислений, со значительной степенью ускорения внутреннего времени. Этот недостаток может также проявляться в случаях, когда необходима небольшая длина шага, в частности, менее одной минуты по внутрисистемному времени. Графический интерфейс в особенности страдает от этой проблемы, поскольку он с трудом справляется уже с потоком сообщений чаще чем раз в секунду, хотя этот эффект более заметен на сложных визуальных объектах, таких как графики. Однако и для прочих элементов с трудом можно достичь частоты сообщений в секунду больше 10, то есть минимальная длина шага симуляции по внешнему времени составляет 0,1 с. Соответственно, моделирование в течение одного дня при длине шага в 1 минуту по виртуальному времени занимает 2,4 минуты, одного месяца или 30 дней — 72 минуты, одного года — 14,6 часов. Таким образом, необходимые временные ресурсы для моделирования оказываются значительными даже для частоты в 1 минуту.

Несмотря на выделенные недостатки, подход, основанный на интеграции Node-RED и Yandex IoT Core, показал себя хорошо и может быть рекомендован к использованию.

ВЫВОДЫ

В четвертой главе были решены следующие задачи:

1. Проведены имитационные эксперименты.
2. Проанализированы результаты и их значение с точки зрения сравнения методов обеспечения энергоэффективности.
3. Выделены преимущества и недостатки использования разработанного метода тестирования.

ЗАКЛЮЧЕНИЕ

В данной работе ставилась следующая задача: исследовать возможность применения инструмента разработки Node-RED и облачного сервиса Yandex IoT Core в сфере интернета вещей. В частности, целью была разработка подхода к моделированию и тестированию систем интернета вещей с их использованием.

В качестве подготовки была проанализирована концепция интернета вещей, технологические предпосылки ее появления и возникающие при внедрении систем интернета вещей сложности. Были рассмотрены основные требования, выдвигаемые к таким программным продуктам, особенности их тестирования. В качестве основного подхода было предложено использование имитационного моделирования с помощью Node-RED и Yandex IoT Core.

В качестве пробной сферы для проверки предложенного подхода была выбрана домашняя автоматизация, в частности системы управления энергопотреблением. Были изучены предлагаемые в литературе подходы к решению этой задачи, выделены наиболее перспективные на основе использования статистических данных об энергопотреблении домохозяйств в Республике Беларусь. Были выделены основные компоненты таких систем, на основе которых была разработана математическая модель дома, включающая показатели окружающей среды, необходимые для решения этой задачи.

Разработанная модель была реализована с использованием Node-RED и Yandex IoT Core, после чего она была испытана для сравнения нескольких подходов в сфере обеспечения эффективности работы устройств ОВиК. Разработанная модель позволила оценить преимущества, получаемые при дополнении простейшей системы, основанной на контроле температуры, некоторыми поведенческими компонентами.

В ходе выполнения данной работы были рассмотрены следующие вопросы и решены следующие задачи:

1. Изучена концепция интернета вещей и основные особенности этой сферы, включая анализ качества программных продуктов.
2. Рассмотрены проблемы тестирования систем интернета вещей с особым фокусом на использование имитационного моделирования, а также возможность использования Node-RED и Yandex IoT Core.
3. Проведен анализ использования домашней автоматизации для управления энергопотреблением, включая среду их работы, основные компоненты.

4. Описаны основные подходы к обеспечению эффективности использования энергии.
5. На основе выделенных компонентов создана математическая модель показателей окружающей среды, использующихся для управления потреблением энергии.
6. Разработана система тестирования и моделирования систем интернета вещей с использованием Node-RED и Yandex IoT Core.
7. Проведен имитационный эксперимент в сфере управления энергопотреблением с анализом его результатов.

Таким образом, проведенное в рамках данной работы исследование позволяет сделать вывод о возможности использования инструмента разработки Node-RED в сочетании с облачными технологиями, представляемыми сервисом Yandex IoT Core, и о преимуществах такого использования, к которым можно отнести легкость вхождения, удобство реализации, возможность переноса разработанного прототипа в среду применения. Подход рекомендуется к использованию при оценке систем интернета вещей или их прототипов.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Левенцов, В.А. Аспекты концепции «Индустрия 4.0» в части проектирования производственных процессов / В.А. Левенцов, А.Е. Радаев, Н.Н. Николаевский // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Экономические науки. — 2017. — С. 19—31.
2. Акопов, А.С. Имитационное моделирование / А. С. Акопов. — М.: Издательство Юрайт, 2016. — 389 с.
3. Потребление энергии в домашних хозяйствах Республики Беларусь [Электронный ресурс] / Нац. Стат. Комитет Респ. Беларусь. — Минск, 2021. — Режим доступа: <https://www.belstat.gov.by/> — Дата доступа: 28.04.2021.
4. Техническая документация Yandex Cloud [Электронный ресурс]. — Режим доступа: <https://cloud.yandex.ru/docs/> — Дата доступа: 07.05.2021.
5. Энергетический баланс Республики Беларусь: стат. Сборник / Нац. Стат. Комитет Респ. Беларусь; редкол.: И.В. Медведева (председатель) [и др.]. — Минск, 2020. — 152 с.
6. Johnston, W.M. Advances in Dataflow Programming Languages / W.M. Johnston, J.R.P. Hanna, R.J. Millar // ACM Computing Surveys. — 2004. — Vol. 36. — P. 1—34.
7. Pan, J. An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments / J. Pan [et al.] // IEEE Internet of Things Journal. — IEEE, 2015. — Vol. 2. — P. 527—537.
8. Stojkoska, B.L.R. A review of Internet of Things for smart home: Challenges and solutions / B.L.R. Stojkoska, K.V. Trivodaliev // Journal of Cleaner Production. — Elsevier, 2017. — Vol. 140. — P. 1454—1464.
9. Pedrasa, M.A.A. Coordinated Scheduling of Residential Distributed Energy Resources to Optimize Smart Home Energy Services / M.A.A Pedrasa, T.D. Spooner, I.F. MacGill // IEEE Transactions on Smart Grid. — IEEE, 2010. — Vol. 1. — P. 134—143.
10. Okamoto-Mizuno, K. Effects of thermal environment on sleep and circadian rhythm / K. Okamoto-Mizuno, K. Mizuno // Journal of Physiological Anthropology. — Springer, 2012. — Vol. 31.
11. Pritoni, M. Energy efficiency and the misuse of programmable thermostats: The effectiveness of crowdsourcing for understanding household behavior / M. Pritoni [et al.] // Energy Research & Social Science. — Elsevier, 2015. — Vol. 8 — P. 190—197.

12. Shrouf, F. Energy management based on Internet of Things: practices and framework for adoption in production management / F. Shrouf, G. Miragliotta // *Journal of Cleaner Production*. — Elsevier, 2015. — Vol. 100. — P. 235—246.
13. Asare-Bediako, B. Integrated Energy Optimization with Smart Home Energy Management Systems / B. Asare-Bediako, P.F. Ribeiro, W.L. Kling // *3rd IEEE PES Innovative Smart Grid Technologies Europe*, Berlin, 14—17 Oct. 2012 / IEEE. — New York, 2013. — P. 1—8.
14. Beaudin, M. Home energy management systems: A review of modelling and complexity / M. Beaudin, H. Zareipour // *Renewable and Sustainable Energy Reviews*. — Elsevier, 2015. — Vol. 45 — P. 318—335.
15. Node-RED [Electronic resource] — Mode of access: <https://nodered.org/> — Date of access: 13.05.2021.
16. Adua, L. Reviewing the complexity of energy behavior: Technologies, analytical traditions, and household energy consumption data in the United States / L. Adua // *Energy Research & Social Science*. — Elsevier, 2020. — Vol. 59.
17. Simulink [Electronic resource] — Mode of access: mathworks.com/help/simulink/ — Date of access: 10.05.2021.
18. Zhou, B. Smart home energy management systems: Concept, configurations, and scheduling strategies / B. Zhou [et al.] // *Renewable and Sustainable Energy Reviews*. — Elsevier, 2016. — Vol. 61. — P. 30—40.
19. Software engineering — Product quality — Part 2: External metrics: ISO/IEC TR 9126-2:2003. — Intr. 07.2003. — Geneva: International Organization for Standardization, 2003. — 86 c.
20. Liu, J. The Smart Thermostat: Using Occupancy Sensors to Save Energy in Homes / J. Liu [et al.] // *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, Zurich, Nov. 2010, / Association for Computing Machinery. — New York, 2010. — P. 211—224.
21. Blackstock, M. Toward a Distributed Data Flow Platform for the Web of Things (Distributed Node-RED) / M. Blackstock, R. Lea // *Proceedings of the 5th International Workshop on Web of Things*. — 2014. — P. 34—39.

```

function timing(msg) {
  let t = global.get("simtime");
  let total_steps = global.get("total_steps");

  if (t > total_steps) {
    msg.payload = "END";
    return [null, msg];
  } else if (t >= 0) {
    msg.payload = t;
    return [msg, null];
  }
}

function outdoorTemp(msg) {
  let t = global.get("simtime") / global.get("steps_per_hour");
  let ampl = global.get("T_out_ampl");
  let T_var = ampl * Math.cos(2 * Math.PI / 24 * (t - 14));

  msg.T_out = global.get("T_out") + T_var;
  return msg;
}

function indoorTemp(msg) {
  msg.T_in = global.get("T_in");
  return msg;
}

function PIR(msg) {
  let h = (global.get("simtime") / global.get("steps_per_hour")) % 24;

  if ((h >= 7 && h < 9) || (h >= 17 && h < 22)) {
    msg.payload = "active";
    return msg;
  }
}

function door(msg) {
  let h = (global.get("simtime") / global.get("steps_per_hour")) % 24;

  if (h == 9) {
    msg.payload = "leave";
    return msg;
  } else if (h == 17) {
    msg.payload = "enter";
    return msg;
  }
}

```

```

function tempDiff(msg) {
    let occupancy = global.get("hems.occupancy");

    if (occupancy === "AWAY") {
        return [null, msg];
    } else {
        let currT = msg.payload;
        let goalT = global.get("hems.goal_temp")[occupancy];

        msg.diff_T = goalT - currT;
        return [msg, null];
    }
}

function activity(msg) {
    let curr_time = global.get("simtime");

    if (msg.payload === "RESET") {
        context.set("last_active", -1);
    } else if (msg.payload === "active") {
        context.set("last_active", curr_time);
        msg.payload = "on";
        return msg;
    } else { // timing signal
        let last_active = context.get("last_active");
        if (curr_time - last_active > 1) {
            msg.payload = "off";
            return msg;
        }
    }
}

function occupancy(msg) {
    var state = context.get("state");

    if (state === "ACTIVE") {
        if (msg.payload === "off") {
            state = "ASLEEP";
        } else if (msg.payload === "leave") {
            state = "AWAY";
        }
    } else if (state === "AWAY") {
        if (msg.payload === "enter") {
            state = "ACTIVE";
        }
    } else if (state === "ASLEEP") {
        if (msg.payload === "on") {
            state = "ACTIVE";
        }
    }

    context.set("state", state);
    msg.payload = state;
    return msg;
}

```

```

function heatFlow(msg) {
  let T_heater = global.get("T_heater");
  let T_indoor = msg.T_in;
  let Mdot = global.get("Mdot");
  let c = global.get("c");

  let heat_flow = (T_heater - T_indoor) * Mdot * c;
  msg.delta_Q_heater = heat_flow;
  return msg;
}

function energyCost(msg) {
  let time = 1/global.get("steps_per_hour");
  let heat_flow = msg.delta_Q_heater;
  let cost_kWh = global.get("cost_kWh");

  let cumulative_cost = global.get("total_cost") +
    heat_flow * time * cost_kWh;
  global.set("total_cost", cumulative_cost);

  msg.cost = cumulative_cost;
  return msg;
}

function deltaQLosses(msg) {
  T_in = msg.T_in;
  T_out = msg.T_out;
  R_eq = global.get("R_eq");

  delta_Q_losses = (T_in - T_out) / R_eq;
  msg.delta_Q_losses = delta_Q_losses;
  return msg;
}

function updateTin(msg) {
  var T_in = global.get("T_in");
  let delta_Q_heater = msg.delta_Q_heater;
  let delta_Q_losses = msg.delta_Q_losses;
  let M_air = global.get("M_air");
  let c = global.get("c");

  delta_T_in = (delta_Q_heater - delta_Q_losses) / (M_air * c);
  T_in += delta_T_in / global.get("steps_per_hour");

  global.set("T_in", T_in);

  msg.payload = delta_T_in;
  return msg;
}

```

```

function saveAndStep(msg) {
  let steps_per_hour = global.get("steps_per_hour");
  let steps = global.get("simtime");
  let hours = steps / steps_per_hour;

  let occupancy = global.get("hems.occupancy");

  let simdata = `${hours},${msg.T_in},${msg.T_out},${msg.cost},${occupancy}`
  msg.payload = simdata;

  global.set("simtime", steps + 1);

  if (steps % steps_per_hour === 0) {
    indoors = {
      "payload": {
        "payload": msg.T_in,
        "timestamp": hours,
        "topic": "in"
      }
    };
    outdoors = {
      "payload": {
        "payload": msg.T_out,
        "timestamp": hours,
        "topic": "out"
      }
    };
    cost = {
      "payload": {
        "payload": msg.cost,
        "timestamp": hours,
        "topic": "cost"
      }
    };
    return [msg, indoors, outdoors, cost];
  } else {
    return [msg, null, null, null];
  }
}

function formatCost(msg) {
  let cost = msg.payload.payload.toFixed(2) + " руб.";
  msg.payload = cost;
  return msg;
}

function backToMsg(msg) {
  let message = msg.payload;
  for (const [key, value] of Object.entries(message)) {
    msg[key] = value;
  }
  return msg;
}

```