



УДК 004.75:004.42+004.94  
<https://doi.org/10.37661/1816-0301-2022-19-2-100-116>

Оригинальная статья  
Original Paper

## О среде имитационного моделирования компонентов и приложений в Интернете вещей на основе инструментов визуального потокового программирования и облачных сервисов

А. А. Андрушевич<sup>1</sup>✉, И. С. Войтешенко<sup>2</sup>, О. Ю. Емельянова<sup>2</sup>

<sup>1</sup>Научно-инженерное республиканское унитарное предприятие «Межотраслевой научно-практический центр систем идентификации и электронных деловых операций», ул. Академическая, 15, к. 2, Минск, 220072, Беларусь

✉E-mail: [andrushevich@bsu.by](mailto:andrushevich@bsu.by)

<sup>2</sup>Белорусский государственный университет, пр. Независимости, 4, Минск, 220030, Беларусь

### Аннотация

**Цели.** Целью проведенной аналитической и исследовательской работы являлись разработка и первоначальная оценка возможностей новой среды имитационного моделирования компонентов и приложений в Интернете вещей. Актуальность задачи связана с необходимостью упрощения исследования и тестирования таких приложений в связи с активным развитием и расширением Интернета вещей. При разработке среды имитационного моделирования преследовались следующие основные цели: построение математической модели, реализация программного обеспечения, в рамках которого могут быть проведены эксперименты с данной моделью, обеспечение возможности анализировать результаты и корректировать модель.

**Методы.** Использовались методы имитационного моделирования.

**Результаты.** Анализ актуальности и востребованности результатов исследований показал, что в качестве примера для демонстрации методов и средств решения задачи имитационного моделирования компонентов и приложений в Интернете вещей может быть выбрана реализация части модели приложения «умный дом», управляющей энергоэффективностью жилых помещений. Была разработана математическая модель дома, а также среда для моделирования на основе инструмента визуального потокового программирования Node-RED и облачного сервиса Yandex Cloud / Yandex IoT Core.

**Заключение.** Разработана имитационная модель системы управления энергопотреблением «умного дома», включающая моделирование времени, условий внешней среды, теплопотерь, режимов работы отопительного оборудования и поведения жителей такого дома. На основе реализованной модели проведена первоначальная серия имитационных экспериментов, направленных на проверку базовой функциональности подготовленной среды имитационного моделирования и выбранного примера моделирования системы управления энергопотреблением жилых помещений.

**Ключевые слова:** Интернет вещей, имитационное моделирование, визуальное программирование, облачные вычисления, облачные технологии, управление энергопотреблением «умного дома»

**Для цитирования.** Андрушевич, А. А. О среде имитационного моделирования компонентов и приложений в Интернете вещей на основе инструментов визуального потокового программирования и облачных сервисов / А. А. Андрушевич, И. С. Войтешенко, О. Ю. Емельянова // Информатика. – 2022. – Т. 19, № 2. – С. 100–116. <https://doi.org/10.37661/1816-0301-2022-19-2-100-116>

**Конфликт интересов.** Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 18.03.2022

Подписана в печать | Accepted 15.04.2022

Опубликована | Published 29.06.2022

## About simulation environment for modeling the Internet of Things components and applications using visual flow-based programming tools and cloud services

Alexey A. Andrushevich<sup>1✉</sup>, Iosif S. Vojteshenko<sup>2</sup>, Olga Ju. Emelianova<sup>2</sup>

<sup>1</sup>*Scientific Engineering Republican Unitary Enterprise "Interbranches Research & Development Centre for Identification Systems and e-Business Operations", st. Akademicheskaya, 15, building 2, Minsk, 220072, Belarus*

✉E-mail: andrushevich@bsu.by

<sup>2</sup>*Belarusian State University, av. Nezavisimosti, 4, Minsk, 220030, Belarus*

### Abstract

**Objectives.** The purpose of the analytical and research work was to develop and perform an initial assessment of the capabilities of simulation environment for modeling the Internet of Things (IoT) components and applications. The relevance of the problem is associated with the need to simplify research and testing of such systems as the field is growing. In the implementation of the simulation environment, the following goals were pursued: building a mathematical model; implementation of software, capable of running experiments on that model; providing the user with the ability to analyze results and adjust the model.

**Methods.** Methods of simulation modeling were used.

**Results.** Analysis of the relevance and impact of the research results has led to an appropriate example for demonstrating methods and means of solving the problem of IoT subsystems, components, and applications simulation in the proposed environment. This example has been implemented in the part of the Smart Home application model responsible for the energy efficiency optimization in residential buildings enclosed in a simulation environment based on an integrated software package consisting of the Node-RED visual tool for flow-based programming and the Yandex Cloud / Yandex IoT Core cloud service.

**Conclusion.** A simulation model for managing energy consumption of a "smart home" was developed and implemented using the previously specified software package, including modeling time, environmental conditions, heat loss, operating modes of heating equipment and the behavior of house residents. Based on the implemented model, an initial series of simulation experiments were also carried out, on the one hand, aimed at checking some characteristics of the functionality of the developed simulation environment and the selected example for simulation of residential premises energy management. As a result of the initial experiments, the basic functionality of the integrated software package was proven and demonstrated based on the use of Node-RED and the Yandex Cloud / Yandex IoT Core cloud service for solving problems of simulation modeling of components, subsystems and applications of the Internet of things.

**Keywords:** Internet of Things, simulation, visual programming, cloud computing, cloud technologies, smart home energy management

**For citation.** Andrushevich A. A., Vojteshenko I. S., Emelianova O. Ju. *About simulation environment for modeling the Internet of Things components and applications using visual flow-based programming tools and cloud services*. *Informatika [Informatics]*, 2022, vol. 19, no. 2, pp. 100–116 (In Russ.).

<https://doi.org/10.37661/1816-0301-2022-19-2-100-116>

**Conflict of interest.** The authors declare of no conflict of interest.

**Введение.** В настоящее время человечество переживает период активного развития Интернета вещей, выполняются теоретические работы и практическое внедрение новых технологий в различных его приложениях. В связи с этим возникают вопросы стандартизации используемых подходов и разработки общей информационно-коммуникационной среды. Решением данных вопросов занимаются национальные и международные организации по стандартизации, такие как Государственный комитет по стандартизации Республики Беларусь и W3C (URL: <https://www.w3.org/TR/wot-usecases/>). Настоящая статья посвящена подготовке среды имитационного моделирования, а также исследованию ее функциональных и нефункциональных характеристик на основе одного из инструментов, рекомендованных W3C для использования в сфере

Интернета вещей (URL: <https://www.w3.org/2020/04/pressrelease-wot-rec.html/>), а именно инструмента визуального программирования Node-RED (URL: <https://nodered.org/>) в интеграции с облачным сервисом Yandex Cloud (URL: <https://cloud.yandex.ru/docs/>).

Предложение интегрировать Node-RED с Yandex Cloud / Yandex IoT Core уже высказывалось разработчиками [1], однако адаптация и накопление опыта в результате использования этого инструментария в различных сферах является актуальной задачей.

**Имитационное моделирование при построении и исследовании приложений Интернета вещей.** Приложения Интернета вещей тесно связаны с окружающей средой, где устройства и его составные компоненты представляют собой мобильные гетерогенные объекты, оснащенные датчиками и манипуляторами. Каждое устройство может отличаться от других по функциональности и доступности вычислительных ресурсов, что, в свою очередь, определяется программным и аппаратным обеспечением. Для связи элементов используется коммуникативно-вычислительная инфраструктура Интернета, для которой требуется обеспечение корректности обработки данных и взаимодействия между различными частями рассматриваемой распределенной системы.

Таким образом, приложение Интернета вещей может быть представлено как надстройка над потоком данных об окружающей среде, которая принимает решения, воздействующие на эту среду (рис. 1).

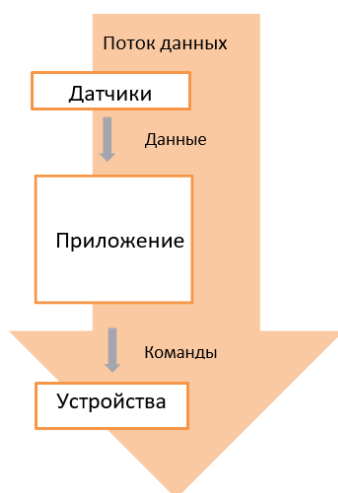


Рис. 1. Схематическое представление приложения в Интернете вещей

*Fig. 1. Schematic representation of Internet of things application*

Всесторонняя оценка работы программно-аппаратных продуктовых приложений в области Интернета вещей в среде их конечного развертывания требует большого количества ресурсов, в первую очередь финансовых и временных. Еще более сложным оказывается решение задачи исследования гипотез в отношении перспективных подходов. В этой связи представляет интерес разработка методов проверки правильности функционирования коммуникативно-вычислительной системы и ее соответствия заданным требованиям. При построении моделей распределенных систем традиционно используются различные языки: Unified Modeling Language (UML, URL: <https://www.uml.org/>), Very large scale Hardware Description Language (VHDL, URL: <http://www.eda-twiki.org/cgi-bin/view.cgi/P1076/WebHome/>), Triad.Net и др. При создании симуляторов компьютерных сетей часто применяется модульная компонентно-ориентированная C++ библиотека для дискретно-событийного моделирования OMNeT++ (URL: <https://omnetpp.org>) совместно с библиотекой сетевых моделей INET Framework (URL: <https://inet.omnetpp.org>). Также следует упомянуть библиотеки NS3 (URL: <https://www.nsnam.org>), GNS3 (URL: <https://www.gns3.com>) и Common Open Research Emulator (CORE, URL: <https://coreemu.github.io/core/>).

При моделировании компонентов и приложений в Интернете вещей важно иметь возможность отделять технические аспекты реализации, такие как выбор конкретных устройств или протоколов взаимодействия, чтобы сосредоточиться на обеспечении корректного поведения программной части. Иначе говоря, требуется разработать метод, который позволял бы в зависимости от конкретной ситуации проверять систему или ее прототип на соответствие заранее определенным критериям, давать оценку качества работы системы путем вычисления заданных показателей.

*Имитационное моделирование* – метод исследования, состоящий в создании и изучении модели, достаточно похожей на исходную систему. Распространенный пакет компьютерного имитационного моделирования динамических систем Simulink из числового пакета MATLAB (URL: <https://www.mathworks.com/products/simulink.html>) является типичным средством визуально-ориентированного программирования. Пользователь имеет возможность на основе готовых графических модулей (блоков) составлять структурные схемы (блок-схемы) исследуемой системы, выполнять имитацию происходящих в системе процессов и анализировать их течение во времени.

Несмотря на наличие визуального представления, применение Simulink требует владения определенными навыками. Кроме того, Simulink отличается относительно высокой стоимостью, в особенности для разработчиков или исследователей, не имеющих доступ в рамках некоторой организации. Поэтому вместо пакета Simulink желательно использовать программные средства с открытым исходным кодом или с оплатой только реально потребленных ресурсов. Что касается процесса разработки и тестирования, то модель, реализованная в Simulink, не может быть напрямую использована в приложениях Интернета вещей. Более того, для уже существующих приложений требуется разработка отдельной модели. Все эти факторы приводят к тому, что применение Simulink в некоторых случаях оказывается невозможным или неэффективным.

Среда имитационного моделирования должна позволять реализовывать широкий круг сценариев моделирования и эмулируемых устройств, а также рассчитывать оценки различных характеристик качества. При этом созданная система должна быть легко изменяемой и дополняемой. 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 переноса разработанной системы между различными средами исполнения, в том числе возможность упрощенного перехода от прототипа к конечному продукту. Поэтому наиболее важным критерием выбора технологии является спектр ее возможностей.

Таким образом, возникает необходимость подготовки и настройки особой среды имитационного моделирования, в рамках которой могли бы проводиться имитационные эксперименты над компонентами и приложениями в Интернете вещей, т. е. сравнение различных сценариев их функционирования при разных условиях окружающей среды. При моделировании таких приложений требуется воспроизведение показателей, связанных со всеми относящимися к системе процессами, в течение длительного виртуального времени. В этой связи под моделированием понимается исследование поведения приложения косвенным путем – при помощи анализа вспомогательных объектов [2].

Наконец, отметим, что сложность может представлять разработка таких требований или критериев моделирования, которые бы учитывали несколько конфликтующих между собой целей. Зачастую задачи приложений Интернета вещей включают в себя несколько предписаний, между которыми требуется найти компромисс, причем на этапе разработки нередко само соотношение еще не определено. Таким образом, необходимо обеспечить достаточную гибкость среды имитационного моделирования, чтобы был возможен легкий переход между различными критериями и показателями качества.

Подводя итоги, можно предложить следующую схему для среды имитационного моделирования компонентов и приложений в Интернете вещей (рис. 2).

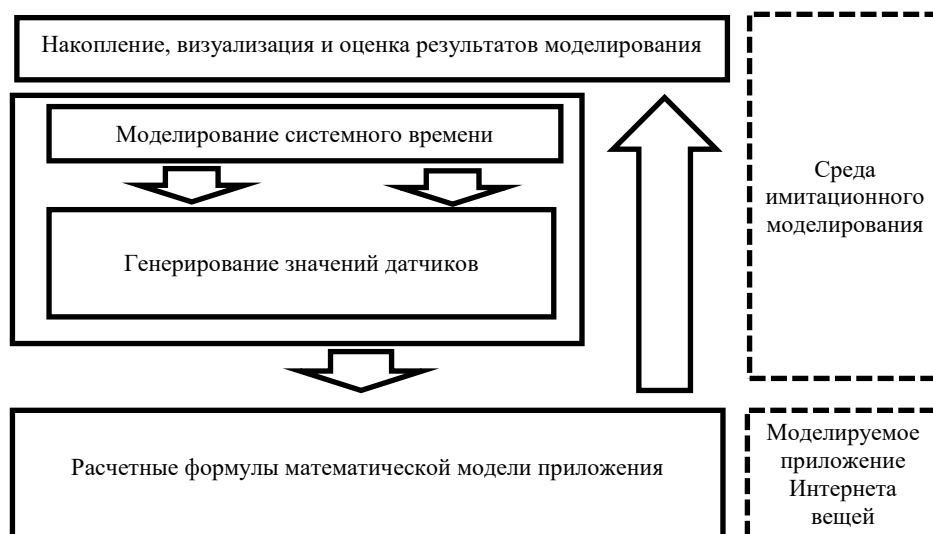


Рис. 2. Схематическое представление среды моделирования приложений в Интернете вещей  
Fig. 2. Schematic representation of modeling environment for Internet of things application

**Программный инструментарий (Node-RED и Yandex IoT Core) и подготовка среды моделирования.** Node-RED – это визуальный инструмент разработки, основанный на парадигме потокового программирования. В Node-RED приложение представляется как сеть так называемых узлов (URL: <https://nodered.org/>), каждый из которых имеет определенную цель: на вход поступают некоторые данные, они обрабатываются внутри, после чего передаются дальше по сети. За перемещение данных между узлами отвечает сеть.

Приложение Интернета вещей представляется как отдельный поток Node-RED, но оно может быть реализовано независимо и подключено 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 моделируемым приложением и анализ его функционирования.

Yandex IoT Core – основной сервис экосистемы Yandex Cloud для Интернета вещей. Он предназначен для двустороннего 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 могут асинхронно получать данные из нее. Основными элементами Yandex IoT Core являются устройство и реестр, которые обмениваются данными и командами через MQTT-брокер (URL: <https://cloud.yandex.ru/docs/>). 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: получает и обрабатывает сообщения, контролирует их доставку MQTT-клиентам (устройствам и реестрам). Обмен данными может происходить только через MQTT-брокер, который обеспечивает необходимый уровень качества обслуживания при обмене сообщениями. Чтобы устройства и реестры могли получать сообщения через MQTT-брокер, они должны быть подписаны на нужные топики.

Yandex IoT Core дополняет Node-RED, создавая возможность коммуникации с моделируемым приложением Интернета вещей при имитационном моделировании. Таким образом, приложения Интернета вещей состоят из двух частей: первая связана с передачей данных и команд между хранилищем и устройствами, а вторая осуществляет работу с полученными данными (рис. 3).

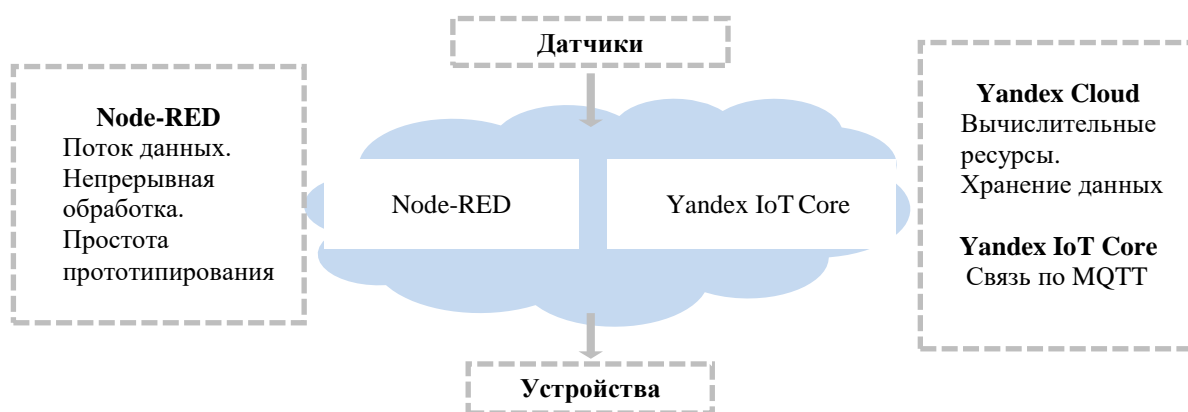


Рис. 3. Схематическое представление приложения в Интернете вещей на основе Node-RED и Yandex IoT Core  
 Fig. 3. Schematic representation of Internet of things application based on Node-RED and Yandex IoT Core

**Преимущества и ограничения Node-RED и Yandex IoT Core.** При использовании в сфере Интернета вещей Node-RED предлагает несколько преимуществ по сравнению с другими вариантами. Во-первых, он является инструментом для визуального программирования, что позволяет облегчить создание прототипов, интегрировать все стадии разработки системы. При этом Node-RED достаточно гибок, чтобы решать большинство имеющихся в области Интернета вещей задач. Во-вторых, Node-RED можно запускать на большом количестве платформ. Использование Node-RED не накладывает ограничений на формат данных, кроме того, что он должен быть представлен как объект JavaScript. Таким образом, прототип можно использовать как основу для дальнейшей работы над продуктом. В-третьих, разработанные сценарии и правила не зависят от физической реализации системы, поскольку Node-RED предоставляет возможности для создания абстрактных устройств. При необходимости физические устройства можно позже привязать к системе через эти абстрактные устройства.

Вместе с тем использование Node-RED сопровождается дополнительными тратами на пересылку и обработку данных. Например, большую трудность представляют ограничения по частоте событий. На практике, если сигнал посылается чаще 10 раз в секунду, в приложении на основе Node-RED появляются ощутимые задержки. Кроме того, задачи некоторых приложений могут быть слишком сложными для функциональности, предлагаемой Node-RED, хотя это частично может решаться пользовательскими дополнениями.

**Подготовка среды к проведению экспериментов.** 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, реализована с помощью Node-RED. Для сбора и передачи данных был подключен Yandex IoT Core. Далее опишем ключевые детали проектирования и реализации.

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 (URL: <https://cloud.yandex.ru/docs/>). Он 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 помощью узлов MQTT Input и MQTT Output.

**Моделирование времени и значений датчиков.** Немаловажным свойством приложений 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. Под *внешним временем* будем понимать время, которое занимает моделирование с точки зрения внешнего наблюдателя. *Внутреннее,*

или *системное, время* – это время, которое проходит внутри модели. Моделируемое приложение работает только с системным временем. Разработка приложений Интернета вещей чаще всего требует моделирования внутрисистемного времени со значительным ускорением относительно внешнего времени. Таким образом, если за один шаг симуляции для внешнего наблюдателя проходит одна секунда, то для приложения может проходить одна минута, один час, одни сутки и т. д. Производительность обработки сигналов в Node-RED на практике ограничена 10 сигналами в секунду. Соответственно, при временном разрешении внутри системы в один час можно добиться ускорения по сравнению с реальным временем в 36 000 раз, а при временном разрешении в минуту – только в 600.

При реализации моделирования времени с помощью Node-RED интервал внешнего времени поддерживается с помощью узла Inject. Поскольку поток все время активен, то 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.

Исходные данные при генерации показаний датчиков помещаются в 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.

Эмуляция работы датчика зависит от характера его показаний (дискретные или непрерывные). Примером дискретного датчика является датчик присутствия. В Node-RED такой датчик моделируется путем проверки условия, причем узел выдает сообщение, если условие оказывается истинным. Аналогичным образом можно действовать для дискретных датчиков, имеющих более одного значения. Примером непрерывного датчика может служить датчик температуры. Расчет температуры выполняется в нескольких узлах Node-RED, поскольку на ее значение влияют многие факторы. Некоторые из этих узлов относятся к обработке данных, полученных от моделируемого приложения. Отдельный узел был выделен для передачи данных о температуре на каждом шаге симуляции, поскольку предполагается, что приложение должно регулярно получать новые данные о температуре внутри подконтрольного ему помещения. Набор узлов, предназначенных для генерации показаний датчиков и управления временем, показан на рис. 4. Здесь и далее изображения взяты из визуального представления исходного текста программного обеспечения, созданного внутри среды Node-RED.

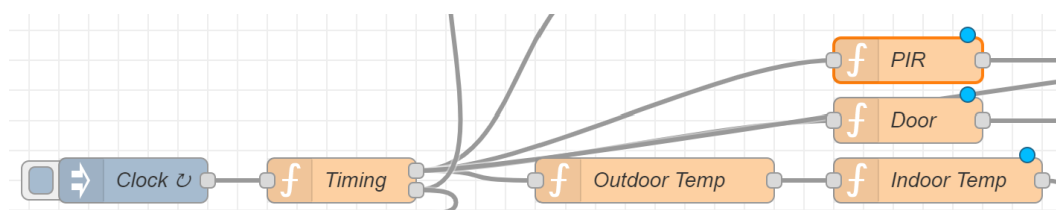


Рис. 4. Основные узлы генерации показаний датчиков

Fig. 4. Main sensor data generation nodes

Модель окружающего мира может быть в любой момент дополнена (при появлении необходимости в моделировании нового показателя) за счет добавления в систему нового узла, который соединяется с узлом, генерирующим время.

**Пользовательский интерфейс. Возможности конфигурации среды пользователем.** Чтобы среда имитационного моделирования была более наглядной и интуитивно понятной, желательно визуализировать процесс ее работы.

Для визуализации данных Node-RED предлагает особые графические узлы, позволяющие создавать элементы интерфейса, такие как текстовые поля, графики, на 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 рис. 5.

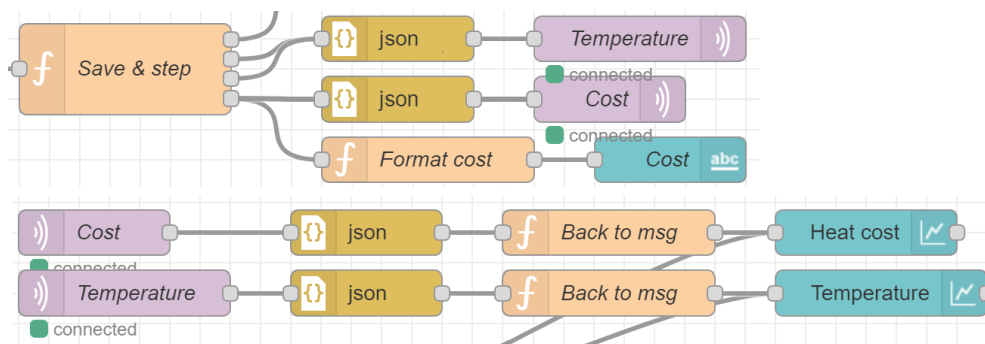


Рис. 5. Узлы для сбора, передачи и визуализации данных

Fig. 5. Nodes for data gathering, transmission and visualization

Помимо визуализации, собранные данные сохраняются в файл, который 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 изображены на рис. 6.

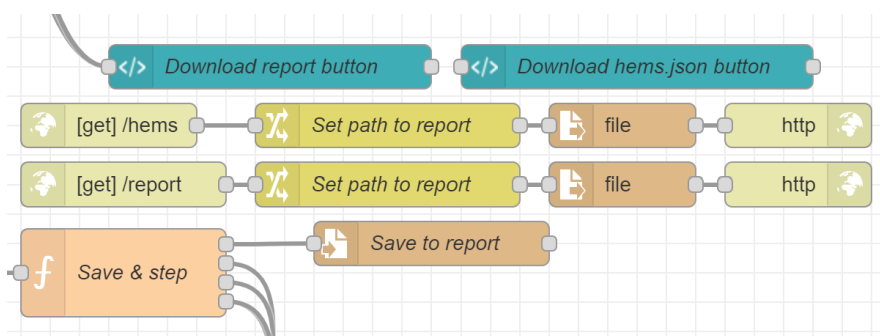


Рис. 6. Узлы для работы с файлами

Fig. 6. Nodes for file management

Таким образом, при запуске эксперимента пользователь может видеть динамическое изменение внешней температуры и температуры внутри дома, накопление общей стоимости затраченной энергии, количество пройденных шагов симуляции. По окончании эксперимента можно выгрузить сгенерированные данные для их анализа. Вкладка с элементами графического интерфейса, описанными выше, показана на рис. 7.

Пользователю также доступны возможности изменения конфигурации системы. Можно настроить две группы параметров: параметры среды моделирования и параметры моделируемого приложения. К первой группе относятся общее количество шагов и их частота, выраженная как количество шагов в одном часе по внутрисистемному времени. Вторая группа может быть разделена на параметры моделируемого приложения и настройки включаемых модулей. В частности, пользователь может отключать те или иные функции приложения, чтобы проверить реакцию приложения или сравнивать разные стратегии управления энергопотреблением. Под конфигурационную часть пользовательского интерфейса была разработана отдельная вкладка в панели управления (рис. 8).



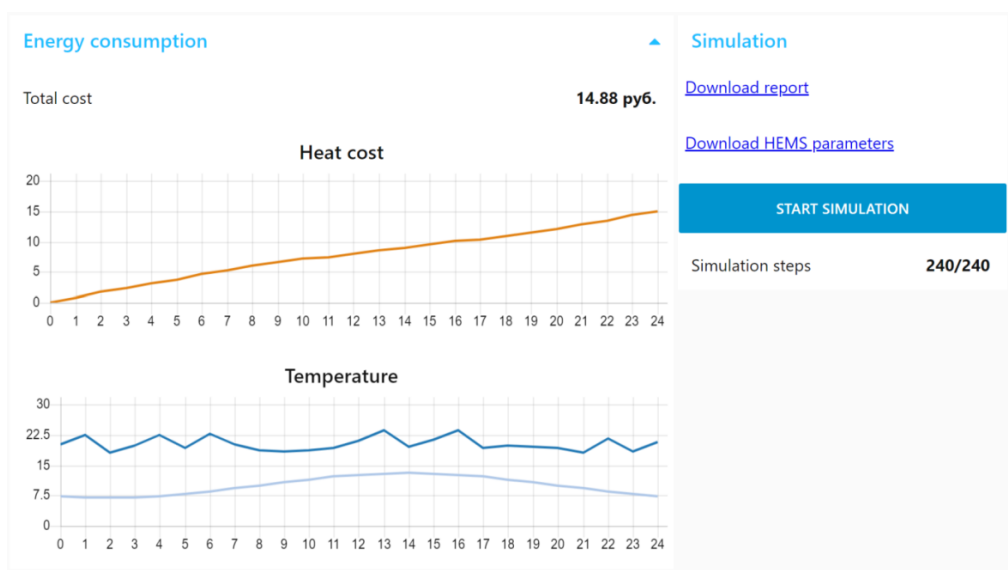


Рис. 7. Графический интерфейс пользователя для проведения симуляции

Fig. 7. Graphical user unterface for simulation execution

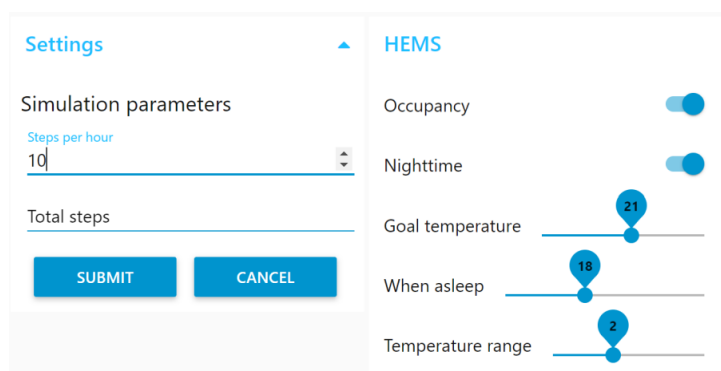


Рис. 8. Конфигурационная часть графического интерфейса пользователя

Fig. 8. Configuration part of graphical user unterface

**Моделируемое приложение и математическая модель Интернета вещей.** В качестве моделируемого приложения Интернета вещей будем рассматривать систему управления энергоэффективностью «умного дома» [3]. Домохозяйства вносят значительный вклад в общее энергопотребление. В структуре конечного потребления топливно-энергетических ресурсов в Республике Беларусь в 2019 г. жилищный сектор занимал почти 27 % [4]. Традиционно большинство решений в сфере управления энергоэффективностью являлись «статическими», т. е. 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 [5, 6].

Прежде чем приступить к планированию и проведению имитационного эксперимента, необходимо сформулировать тестируемую и базовую гипотезы, например, в виде сценария. От выбора базовой конфигурации системы будут зависеть рассчитанные преимущества и недостатки, поэтому важно определить ее корректно. В дальнейшем следует определить конфигурацию эксперимента, в частности специфицировать входные данные об окружающей среде в зависимости от требуемых для расчета показателей, выбрать длительность проведения эксперимента и интервал между шагами моделирования. В итоге необходимо подключить модель прототипа, провести имитационный эксперимент и проанализировать его результаты.

**Моделирование условий внешней среды.** В контексте обеспечения эффективного использования энергии для обогрева помещения наиболее важным показателем внешней среды является ее температура. Заметим, что этот показатель является внутренним по отношению к среде имитационного моделирования. На работу моделируемого приложения он оказывает влияние через расчет внутренней температуры контролируемого помещения, увеличивая или уменьшая потери тепла.

В случае коротких промежутков времени, применяемых для проверки корректности работы и оценки качества отдельных компонентов, простым способом моделирования является использование суммы средней температуры и суточного колебания:

$$T_{\text{внеш}}(t) = T_{\text{ср}} + A \cos(\omega t + \phi), \quad (1)$$

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

$$T_{\text{внеш}}(t) = T_{\text{ср}} + A \cos(\omega(t - h_0)), \quad (2)$$

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

На более длительных промежутках времени начинают играть роль годовичные колебания температуры, а также фактор случайности. Случайностью могут определяться как характер и амплитуда колебаний температуры в течение дня, так и ее среднее значение.

**Моделирование тепловых потерь и работы устройств отопления, вентиляции и кондиционирования.** Для расчета затрат на отопление были определены значения параметров, используемых в имитационном моделировании, в том числе параметры дома:  $l_{\text{дома}} = 30$  м,  $w_{\text{дома}} = 10$  м,  $h_{\text{дома}} = 4$  м,  $\alpha_{\text{крыши}} = 30^\circ$ ,  $N_{\text{окон}} = 4$  ед.,  $h_{\text{окна}} = 1$  м,  $w_{\text{окна}} = 1$  м,  $k_{\text{стены}} = 0,038$  Вт/м·К,  $L_{\text{стены}} = 0,2$  м,  $k_{\text{окна}} = 0,78$  Вт/м·К,  $L_{\text{окна}} = 0,1$  м,  $\rho_{\text{воздуха}} = 1,225$  кг/м<sup>3</sup> – и параметры нагревателя:  $L = 1$  кг/с,  $T_{\text{нагр}} = 50$  °С,  $c = 1005,4$  Дж/кг·К,  $p = 0,1616$  у.е./кВт·ч.

На основании значений указанных параметров при некоторых упрощениях были рассчитаны тепловые потери дома. Для этого использовались формулы для площади стен, площади окон, теплового сопротивления стен, теплового сопротивления окон, массы воздуха в доме, скорости теплового потока от обогревателя и скорости тепловых потерь. На основании этих величин определяется температура воздуха внутри помещения на следующем шаге симуляции:

$$T_{\text{внутр}}(t + \Delta t) = T_{\text{внутр}}(t) + cM_{\text{возд}}(Q_{\text{нагр}} - Q_{\text{потерь})\Delta t, \quad (3)$$

где  $t$  – внутрисистемное время симуляции,  $\Delta t$  – отрезок времени между шагами симуляции,  $c$  – удельная теплоемкость воздуха,  $Q_{\text{нагр}}$  – скорость нагревания от обогревателя,  $Q_{\text{потерь}}$  – скорость тепловых потерь,  $M_{\text{возд}}$  – масса воздуха в помещении. Узлы расчета внутренней температуры представлены на рис. 9.

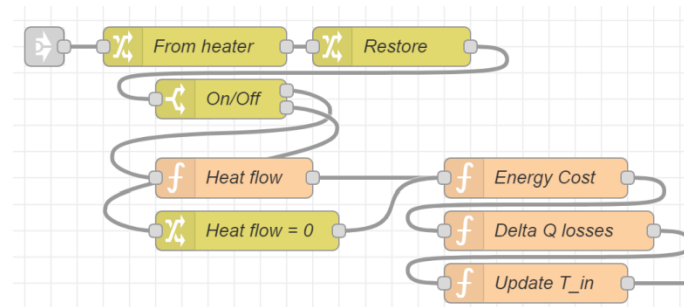


Рис. 9. Узлы расчета внутренней температуры помещения

Fig. 9. Nodes for internal room temperature calculation

**Моделирование работы устройств отопления, вентиляции и кондиционирования и поведения жителей дома.** Основным датчиком для работы с устройствами отопления, вентиляции и кондиционирования (ОВиК) в контексте управления энергопотреблением является датчик температуры внутри контролируемого помещения. В качестве базового сценария управления энергопотреблением была разработана система, включающая и выключающая отопление в зависимости от температуры внутри подконтрольного ей помещения. В этом сценарии отправляется команда на включение, когда разница между целевой и реальной температурой достигает заданного порога, и на выключение, когда температура вновь находится в нужных пределах. Возможные узлы для обработки температурных показаний изображены на рис. 10.

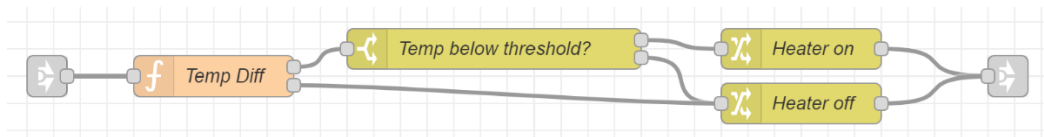


Рис. 10. Узлы для обработки показаний датчика температуры

Fig. 10. Nodes for temperature sensor data processing

Получаемый температурный режим в течение одних суток показан на рис. 11, где верхняя линия отражает температуру в помещении, нижняя – температуру снаружи, а отсутствием заливки отмечены желаемые пределы изменения температуры внутри помещения.

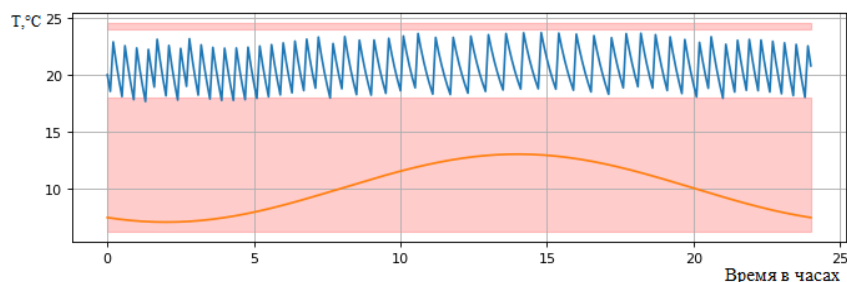


Рис. 11. Температурный режим базового сценария

Fig. 11. Temperature conditions of the base scenario

В течение дня могут возникать промежутки времени, когда дом пустует и нет определенных требований по температуре, что позволяет уменьшить потребление энергии. Такая ситуация фиксируется с помощью датчиков присутствия. Например, система может поддерживать несколько состояний в виде конечного автомата, переходы между которыми осуществляются по сообщениям от датчиков. 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. Узлы системы, предназначенные для реагирования на сообщения от таких датчиков, изображены на рис. 12.

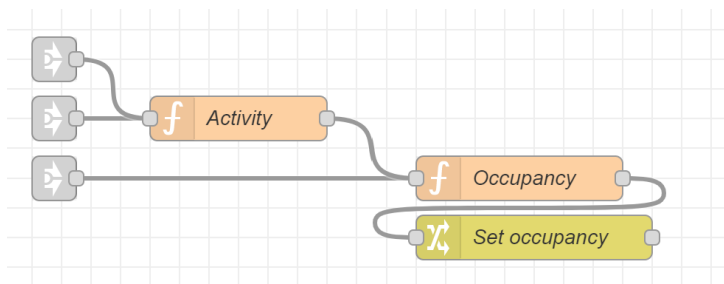


Рис. 12. Узлы обработки данных от датчиков движения

Fig. 12. Nodes for movement sensor data processing

Для сравнения с базовым сценарием были выбраны две стратегии уменьшения энергопотребления, основывающиеся на отслеживании поведения пользователей. Первая из них состоит в том, чтобы наблюдать за активностью людей, находящихся в доме, с помощью датчиков движения. Применение такой стратегии оправдано, поскольку воздействие холода не влияет на качество сна, в то время как излишне высокая температура его снижает [7]. Вторая стратегия использует датчик на входной двери, чтобы отслеживать моменты, когда люди покидают помещение или входят в него. В этом случае можно 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 условленное время, которое сбрасывает по приходу нового сигнала. Вместе с тем если за это условленное время не появилось новых сигналов, то система переходит в неактивное состояние, в котором поддерживает меньшую температуру (рис. 13).

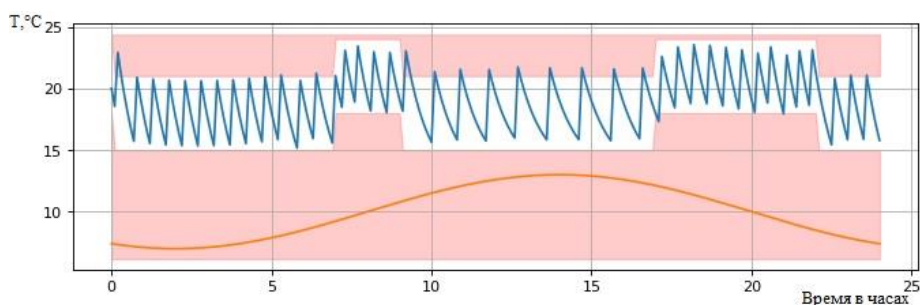


Рис. 13. Температурный режим с учетом активности пользователей

*Fig. 13. Temperature regime taking into account the activity of users*

Вторая стратегия позволяет отслеживать состояние «дом пуст», а не активность пользователей. Для этого к системе подключается датчик, реагирующий на использование входной двери. Таким образом, система отключает отопление в те моменты, когда в помещении никого нет. Возможный температурный режим изображен на рис. 14.

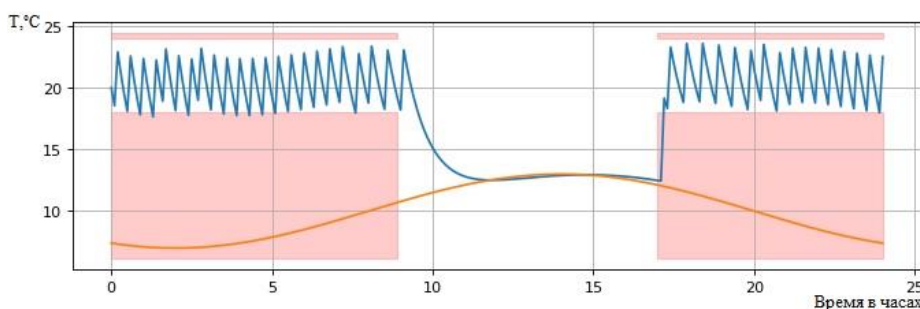


Рис. 14. Температурный режим с учетом наличия людей в помещении

*Fig. 14. Temperature regime, taking into account the presence of people in the room*

Можно заметить, что предложенные стратегии не противоречат друг другу, поэтому имеет смысл проверить их совместную работу. В этом случае система определяет состояние «дом пуст» согласно предыдущей стратегии. Если же такое состояние не зафиксировано, то система действует в зависимости от показаний датчиков движения. Соответствующий температурный режим в течение суток показан на рис. 15.

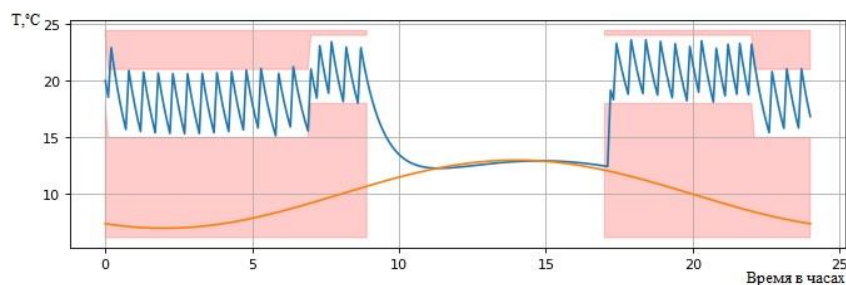


Рис. 15. Температурный режим при использовании обеих стратегий одновременно

Fig. 15. Temperature conditions when using both strategies

**Проведение имитационных экспериментов и их результаты.** Для демонстрации возможности проведения имитационных экспериментов, а также анализа и оценки использования для этого Node-RED и Yandex IoT Core была поставлена задача сравнить несколько сценариев управления устройствами ОВиК, отличающихся степенью и способом учета поведения пользователей приложения.

Основной метрикой функциональности систем управления энергопотреблением является общая стоимость затраченной энергии. Для оценки качества и перспективности сформулированных выше сценариев был поставлен эксперимент: система работает в течение 24 ч по виртуальному времени. При этом для моделирования выбран размер шага в 6 мин, т. е. 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 согласно выражению

$$T_{\text{внеш}} = 10 + 3 \cos\left(\frac{\pi}{12}(t - 14)\right), \quad (4)$$

где  $t$  – текущий временной шаг в часах.

Поведение пользователя моделируется следующим образом: 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 требования к комфорту, предъявляемые пользователями, зависят от ряда состояний: когда жители активны, температура не должна превышать 24 °C и не опускаться ниже 18 °C; когда жители спят, минимальная допустимая температура понижается до 15 °C; если дом пуст, то требований по температуре нет.

Гипотеза заключается в том, что базовый сценарий окажется наименее эффективным в контексте описанных выше критериев, самой же эффективной будет система управления, использующая обе стратегии уменьшения энергопотребления. В результате проведенного имитационного эксперимента были получены следующие результаты: общая стоимость затраченной энергии составила 14,87 условных единиц для базового сценария, для стратегии с учетом активности – 12,64, для стратегии с учетом наличия людей в помещении – 11,52, для объединенной стратегии – 9,88.

На рис. 16 можно видеть динамическое накопление общей стоимости энергии. В частности, заметим, что стратегии, включающие отслеживание наличия людей в помещении, выигрывают за счет длительных промежутков без употребления энергии. Таким образом, выдвинутая гипотеза подтвердилась.

Дополнительно оценим тестируемые стратегии оптимизации с точки зрения комфорта пользователя. Для того чтобы сравнить оценки при разной длительности эксперимента, рассчитаем среднее значение:

$$M = \frac{1}{N} \sum_{i=0}^N m(T, h_i), \quad (5)$$

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

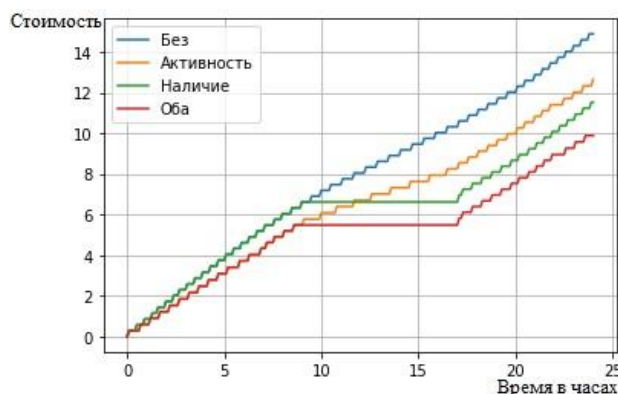


Рис. 16. Накопление потребления энергии

Fig. 16. Accumulation of energy consumption

Использовались два варианта расчета  $i$ -й оценки. При расчете по формуле (6) оценка дается в зависимости от количества единиц времени, в течение которых наблюдалась температура за пределами требуемых значений:

$$m(T,h) = \begin{cases} T_H(h) - T(h), & \text{если } T(h) < T_H(h); \\ 0, & \text{если } T_H(h) \leq T(h) \leq T_B(h); \\ T(h) - T_B(h), & \text{если } T(h) > T_B(h), \end{cases} \quad (6)$$

где  $T_H(h)$  – нижняя граница допустимой температуры на шаге тестирования  $h$ ,  $T(h)$  – текущая температура на шаге симуляции  $h$ ,  $T_B(h)$  – верхняя граница допустимой температуры на шаге симуляции  $h$ . Другую оценку комфорта можно получить с помощью формулы, по которой рассчитывается среднее отклонение от целевой температурной области:

$$m(T,h) = \begin{cases} 1, & \text{если } T(h) < T_H(h) \text{ или } T(h) > T_B(h); \\ 0, & \text{если } T_H(h) \leq T(h) \leq T_B(h). \end{cases} \quad (7)$$

Полученные результаты представлены в табл. 1. Видно, что наилучшие результаты показал базовый (первый) сценарий, однако третий и четвертый сценарии по результатам незначительно отличаются от базового. В целом отметим, что значительных отклонений от требуемых значений не наблюдалось. Данный факт подтверждает возможность поддержки достаточного уровня комфорта всеми сценариями. Наиболее часто нарушения наблюдаются при переходе между состояниями (см. рис. 13–15). Это связано с тем, что помещение не успевает достаточно быстро нагреться или остыть. Возможным решением будет уточнение условий для управления обогревателем, чтобы система управления более плавно изменяла температуру.

Таблица 1  
 Оценка комфорта пользователя для каждого сценария

Table 1  
 Assessment of user comfort for each scenario

Сценарий Scenario	Доля времени вне заданных пределов Fraction of time outside the specified limits	Среднее отклонение от заданных пределов Average deviation from the specified limits
Первый (базовый)	0,0415	0,0072
Второй (активность жителей)	0,1112	0,0472
Третий (наличие жителей)	0,0456	0,0090
Четвертый (обе стратегии второго и третьего сценариев)	0,0539	0,0196

По итогам предварительного анализа эксперимента отметим, что в целях сравнения сценариев в вычислениях были приняты наиболее широкие границы. Таким образом, требования по комфорту были более свободными, чем в отдельных сценариях. Обратим внимание на то, что первый и третий сценарии поддерживают достаточно высокую температуру в ночное время, а это может быть нежелательным для некоторых пользователей.

Для проверки полученных результатов также был проведен более длительный эксперимент продолжительностью семь внутрисистемных дней. Показания датчиков моделировались аналогичным образом, данные представлены в табл. 2.

Таблица 2  
Результаты длительного эксперимента

Table 2  
Results of a long-term experiment

Сценарий <i>Scenario</i>	Потребление энергии <i>Power consumption</i>		Комфорт <i>Comfort</i>	
	Общее <i>General</i>	Среднее за день <i>Daily Average</i>	По времени <i>By time</i>	По отклонению <i>By deviation</i>
Первый	104,13	14,87	0,0464	0,0090
Второй	87,62	12,52	0,1201	0,0573
Третий	78,89	11,27	0,0500	0,0136
Четвертый	70,99	10,14	0,0577	0,0197

Из табл. 2 видно, что показатели изменились незначительно, причем если абсолютные значения и изменились, то отношения между рассматриваемыми моделями – нет. Это позволяет сделать вывод о справедливости результатов первого эксперимента.

**Обсуждение результатов имитационных экспериментов и практические выводы.** По одной из классификаций требования при разработке программного обеспечения (ПО) делятся на функциональные и нефункциональные. Под функциональностью будем понимать соответствие результатов работы ПО ожиданиям пользователя. Целью эксперимента являлась первоначальная оценка возможностей среды имитационного моделирования компонентов и приложений в Интернете вещей на основе инструмента визуального потокового программирования Node-RED и облачного сервиса Yandex Cloud / Yandex IoT Core. В рассмотренном примере приложения Интернета вещей функциональные требования состояли в снижении стоимости энергопотребления и в нахождении показателей температурного комфорта жителей дома в заданных пределах. Результаты проведенных имитационных экспериментов говорят о том, что предлагаемая среда моделирования и выбранный пример приложения удовлетворяют обозначенным требованиям функциональности.

Среди нефункциональных требований важно выделить требования к надежности и эффективности. При всей ограниченности ресурсов компонентов и приложений в Интернете вещей их реакция на изменение окружающей среды должна быть своевременной. Нежелательным поведением приложения в выбранном примере могло бы быть постоянное нахождение обогревателя во включенном состоянии. В результате температура в помещении выйдет за пределы комфортных значений. Кроме того, такая система отопления была бы неэффективной в своем использовании энергии. Ни в одном из проведенных экспериментов не наблюдалось значительного выхода за пределы требуемых значений, что предварительно свидетельствует о надежности и эффективности разработанных прототипов.

Другие категории требований при разработке ПО бывает сложно оценить программным образом без участия человека. С точки зрения удобства использования отметим, что разработанное моделируемое приложение не требует участия человека для своей работы, за исключением установки требуемой температуры. Тем не менее существует риск неверного применения данного приложения при попытках ручного управления отоплением, что приведет к снижению эффективности приложения.

Можно сделать практический вывод, что предлагаемая среда моделирования и рассматриваемый пример моделируемого приложения удовлетворяют базовым требованиям при разработ-



ке ПО. Вместе с тем следует провести дополнительные эксперименты и дать более глубокий анализ их результатов, чтобы сформулировать рекомендации по выбору между рассмотренными сценариями. Дальнейшая исследовательская работа будет заключаться в разработке моделей, проведении экспериментов и анализе результатов для других приложений в Интернете вещей.

**Заключение.** В настоящей публикации авторы представили исследование по разработке и первоначальной оценке возможностей среды имитационного моделирования компонентов и приложений в Интернете вещей на основе интегрированного программного комплекса, состоящего из инструмента визуального потокового программирования Node-RED и облачного сервиса Yandex Cloud / Yandex IoT Core. Для проверки работы данной среды в качестве примера приложения была выбрана система управления энергопотреблением жилых помещений. Для демонстрации ее функциональности проведена серия имитационных экспериментов, которые включили в себя моделирование времени, условий внешней среды, теплопотерь, режимов работы отопительного оборудования и поведения жителей дома. В результате первоначальных экспериментов была обоснована и подтверждена стратегия оптимизации расхода энергии отопительных приборов на основании показателей температурного комфорта и поведения жителей дома.

**Вклад авторов.** *А. А. Андрушевич* – обоснование концепции и анализ актуальности статьи, консультирование по вопросам создания среды имитационного моделирования на основе Node-RED и Yandex Cloud / Yandex IoT Core, формулирование выводов; *И. С. Войтешиенко* – сбор, анализ и интерпретация результатов работы, критический пересмотр содержания текста, утверждение окончательного варианта статьи для публикации; *О. Ю. Емельянова* – существенный вклад в реализацию работы, написание текста.

#### Список использованных источников

1. Сурков, А. Знакомство с Node-RED и потоковое программирование в Yandex IoT Core [Электронный ресурс] / А. Сурков. – Режим доступа: <https://habr.com/ru/company/yandex/blog/519600/>. – Дата доступа: 05.11.2021.
2. Акопов, А. С. Имитационное моделирование / А. С. Акопов. – М. : Юрайт, 2016. – 389 с.
3. Stojkoska, B. L. R. A review of Internet of Things for smart home: Challenges and solutions / B. L. R. Stojkoska, K. V. Trivodaliev // *J. of Cleaner Production*. – 2017. – Vol. 140. – P. 1454–1464.
4. Энергетический баланс Республики Беларусь : стат. сборник / Нац. Стат. Комитет Респ. Беларусь ; редкол. : И. В. Медведева (председатель) [и др.]. – Минск, 2020. – 152 с.
5. Energy efficiency and the misuse of programmable thermostats: The effectiveness of crowdsourcing for understanding household behavior / M. Pritoni [et al.] // *Energy Research & Social Science*. – 2015. – Vol. 58. – P. 190–197.
6. An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments / J. Pan [et al.] // *IEEE Internet of Things J.* – 2015. – Vol. 2. – P. 527–537.
7. Okamoto-Mizuno, K. Effects of thermal environment on sleep and circadian rhythm / K. Okamoto-Mizuno, K. Mizuno // *J. of Physiological Anthropology*. – 2012. – Vol. 31. – P. 105–114.

---

---

#### References

1. Surkov A. Znakomstvo s Node-RED i potokovoe programirovanie v Yandex IoT Core. *Introduction to Node-RED and Streaming Programming in Yandex IoT Core*. Available at: <https://habr.com/ru/company/yandex/blog/519600/> (accessed 05.11.2021) (In Russ.).
2. Akopov A. S. Imitacionnoe modelirovanie. *Simulation*. Moscow, Jurajt, 2016, 389 p. (In Russ.).
3. Stojkoska B. L. R., Trivodaliev K. V. A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 2017, vol. 140, pp. 1454–1464.
4. Jenergeticheskij balans Respubliki Belarus': statisticheskij sbornik. *Energy Balance of the Republic of Belarus: Statistical Compendium*. Nacional'nyj statisticheskij komitet Respubliki Belarus'; In I. V. Medvedeva, I. S. Kangro, Zh. N. Vasilevskaja, O. A. Dovnar, E. I. Kuharevich, ..., A. S. Snetkov (eds.). Minsk, 2020, 152 p. (In Russ.).

5. Pritoni M., Meier A. K., Aragon C., Perry D., Peffer T. Energy efficiency and the misuse of programmable thermostats: The effectiveness of crowdsourcing for understanding household behavior. *Energy Research & Social Science*, 2015, vol. 8, pp. 190–197.

6. Pan J., Jain R., Vu T., Paul S. An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments. *IEEE Internet of Things Journal*, 2015, vol. 2, pp. 527–537.

7. Okamoto-Mizuno K., Mizuno K. Effects of thermal environment on sleep and circadian rhythm. *Journal of Physiological Anthropology*, 2012, vol. 31, pp. 105–114.

### Информация об авторах

*Андрушевич Алексей Андреевич*, младший научный сотрудник, научно-инженерное республиканское унитарное предприятие «Межотраслевой научно-практический центр систем идентификации и электронных деловых операций».

E-mail: andrushevich@bsu.by

*Войтешенко Иосиф Станиславович*, кандидат технических наук, доцент, доцент кафедры технологий программирования факультета прикладной математики и информатики, Белорусский государственный университет.

E-mail: voit@bsu.by

*Емельянова Ольга Юрьевна*, выпускница кафедры технологий программирования факультета прикладной математики и информатики, Белорусский государственный университет.

E-mail: volha.emelianova@gmail.com

### Information about the authors

*Alexey A. Andrushevich*, Junior Researcher, Scientific Engineering Republican Unitary Enterprise "Interbranches Research & Development Centre for Identification Systems and e-Business Operations".

E-mail: andrushevich@bsu.by

*Iosif S. Vojteshenko*, Ph. D. (Eng.), Assoc. Prof., Assoc. Prof. of the Department of Programming Technologies of the Faculty of Applied Mathematics and Informatics, Belarusian State University.

E-mail: voit@bsu.by

*Olga Ju. Emelianova*, Graduate of the Department of Programming Technologies, Faculty of Applied Mathematics and Informatics, Belarusian State University.

E-mail: volha.emelianova@gmail.com