

ИМИТАЦИОННЫЕ ЭКСПЕРИМЕНТЫ ПРИ МОДЕЛИРОВАНИИ ЭЛЕМЕНТОВ "УМНОГО" ДОМА НА ОСНОВЕ ИНТЕГРАЦИИ NODE-RED С YANDEX IOT CORE

Андрушевич А. А., Емельянова О. Ю., Войтешенко И. С.

Белорусский государственный университет, Минск, Беларусь,
e-mail: Andrushevich@bsu.by; volha.emelianova@gmail.com; voit@bsu.by

1 Система интернета вещей как надстройка над потоком данных об окружающей среде.

С достаточно абстрактной точки зрения системы интернета вещей может считаться «черным ящиком», имеющим входы и выходы, в которой по информации, поступающей через входы, принимаются решения, выдаваемые на выходы. Ее особенностью является тесная связь с той или иной сферой применения, а также большие объемы потребляемых данных, непрерывно поступающих в систему. Таким образом, система интернета вещей может быть представлена как надстройка над потоком данных об окружающей среде, которая может принимать решения, воздействующие на эту среду (см. рис 1).



Рис. 1. Схематическое представление системы управления энергоэффективностью жилища.

2. Инструментальные средства разработки и коммуникационный протокол.

Большинство систем интернета вещей состоит из двух частей. Первая из них связана с передачей данных и команд между хранилищем и устройствами, а вторая заключается в работе с полученными данными.

В данной работе применяется сервис Yandex IoT Core, программа Node-RED, коммуникационный протокол MQTT.

Сервис Yandex IoT Core – составная часть облачной платформы Yandex Cloud (Яндекс.Облако).

Основными элементами Yandex IoT Core являются устройство и реестр, которые обмениваются данными и командами через MQTT-брокер [1]. Устройство может отправлять телеметрические данные и получать команды. Реестр – набор логически связанных между собой устройств.

MQTT – сетевой протокол, использующий архитектуру издатель-подписчик и работающий поверх TCP/IP. Этот протокол поддерживается даже если пропускная способность сети мала или конечные устройства обладают ограниченными ресурсами для обработки данных [2].

Node-RED – визуальный инструмент разработки, основанный на парадигме потокового программирования. 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 [3].

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.

3. Система имитационного моделирования

В работе используется подход, основанный на отделении системы интернета вещей 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, задачей тестирующей является генерация и передача этих данных, а также сбор и анализ информации о реакции на них.

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.

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.

Для исполнения среды Node-RED была создана виртуальная машина в облачном сервисе Yandex Compute Cloud.

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.

Задачей этой части является генерация реалистичных показаний датчиков с учетом модели внешнего мира и поведения системы. Набор эмулируемых датчиков зависит от тестируемой системы.

Подход к моделированию условий окружающей среды похож для всех показателей, кроме времени, которое также можно отнести к ним.

Свойством многих 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. В частности, был использован такой интервал, при котором каждую секунду внутрисистемное время увеличивалось на час.

Тестируемая система интернета вещей должна получать данные таким же образом, как она бы получала их от реальных датчиков. Таким образом, появляется необходимость передачи сгенерированных данных об окружающей среде в тестируемую систему таким образом, который был бы неотличим от ожидающегося при окончательном использовании системы.

Для решения этой задачи использовались узлы MQTT Output, позволяющие публиковать данные в MQTT-топик, чтобы их потом могла считывать система интернета вещей, подписавшись на него. Как было описано ранее, для получения данных от датчиков система также использует MQTT-узлы. Таким образом, задача на стороне тестирующей системы сводится к тому, чтобы публиковать данные в правильные топики и в правильном формате. Соответствие топиков обеспечивается настройкой MQTT-узлов.

Аналогичным образом тестирующая система может собирать данные о работе тестируемой системы. Для получения данных по протоколу MQTT используются узлы MQTT Input, подписанные на те топики, в которые тестируемая система выдает команды для устройств.

4. Практика имитационных экспериментов.

В качестве базового варианта выбрана модель, при которой отопление всегда включено (работает 100% времени).

Вариант 1. Вводится верхняя и нижняя границы, между которыми должна находиться температура в помещении. Таким образом, отопление будет включаться по достижении нижней границы и выключаться по достижении верхней.

Вариант 2. Система оснащена датчиком местоположения пользователя, причем верхняя и нижняя граница температуры снижаются, если в помещении нет людей.

Были выбраны следующие параметры эксперимента: период виртуального времени – с 0:00 3 марта по 23:00 9 марта, то есть 7 дней, шаг – 1 час, то есть общая продолжительность – 168 шагов. Температурные границы были установлены в 18 °C и 22 °C при наличии людей в помещении и в 16 °C и 20 °C при их отсутствии. Для задания поведения человека по методу агентного моделирования были выбраны два состояния «дома» и «не дома», переход между которыми совершается в 11:00 и 19:00 каждого дня по системным часам. Для оценки систем используются два показателя: доля времени, в которое была включена система отопления, как оценка энергопотребления относительно максимума и «штраф за комфорт», линейно зависящий от разницы желаемой и наблюдаемой температуры.

Как и ожидалось, вариант 1 позволяют значительно (на 40%) сократить энергопотребление по сравнению базовым. При этом разграничение между присутствием и отсутствием людей в помещении позволило сэкономить еще 10 %. Штраф за комфорт в обеих системах незначителен. Температура в помещении никогда не опускалась ниже, чем на 1 °C меньше требуемой.

Литература

1. Техническая документация Yandex Cloud [Электронный ресурс]. – Режим доступа: <https://cloud.yandex.ru/docs/> – Дата доступа: 05.12.2020.
2. Boyd, B. Building Real-time Mobile Solutions with MQTT and IBM MessageSight / B. Boyd [et al.] – Vervante, 2014. – 264 p.
3. Node-RED [Electronic resource] – Mode of access: <https://nodered.org/> – Date of access: 02.12.2020.