

Разработка Ядра Платформы Автоматизированного Тестирования Устройств Интернета Вещей

Наталья В. Папуловская, Илья Н. Изотов, Денис Ю. Блиничкин, Андрей Ю. Катаев

Аннотация— В статье рассматривается задача тестирования, возникающая при разработке приборов для систем интернета вещей (IoT). Интеллектуальные устройства, обеспечивающие передачу данных по сети Интернет, становятся всё более востребованы. Например, приборы учета энергоресурсов, передающие данные по беспроводной сети являются неотъемлемой частью системы интернета вещей в секторе ЖКХ. В настоящее время тестирование прошивок счетчиков учета потребления воды и тепла происходит в ручном режиме. Производителю приходится тратить значительное количество человеко-часов для этой работы. В статье описаны результаты научно-исследовательской работы по разработке автоматизированной системы тестирования. Экспериментальной базой исследования выбран статический трехфазный универсальный многотарифный прибор учета электрической энергии Милур 307 с модулем Карат-941LW. Счетчик имеет встроенный специализированный микроконтроллер российского производителя «Миландр». Прибор оснащен интерфейсами связи для обмена данными с внешними устройствами. Предметом тестирования является встроенное программное обеспечение интерфейсного модуля Карат 941LW, реализующее прикладной протокол обмена данными верхнего уровня. Для тестирования устройств Интернета вещей авторами разработан специальный сервис – ядро тестирования. Сервис проводит опрос устройства по указанному сценарию, записанному в YAML-файле. Опрос проводится через протокол MQTT, с которым работает сервер сети LoRaWAN. Разработанная методология тестирования будет использована для создания универсальной платформы тестирования устройств, работающих по беспроводной сети передачи данных.

Ключевые слова— Интернет вещей, тестирование, счетчик энергоресурсов, протокол, LoRaWAN, MQTT.

Статья получена 11 апреля 2021.

Папуловская Наталья Владимировна, Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, доцент, n.v.papulovskaia@urfu.ru

Изотов Илья Николаевич, Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, магистрант, normall777@yandex.ru

Блиничкин Денис Юрьевич, Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, магистрант, dblini4kin@yandex.ru

Катаев Андрей Юрьевич, Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, магистрант, akataev96@gmail.com

I. ВВЕДЕНИЕ

В процессе разработки IoT-систем, в том числе IoT-устройств, важным этапом является тестирование. Тестирование позволяет выявить соответствие между реальным и ожидаемым поведением устройства или программы, осуществить контроль качества, чтобы выпустить на рынок работоспособный продукт.

Чем раньше происходит выявление ошибок в программном обеспечении любой системы, тем ниже расходы на их исправление. Особенно это актуально для предприятий, занимающихся массовым производством электронных устройств, приборов, так называемых «умных» устройств. К таким устройствам мы относим и счетчики, измеряющие потребление воды, тепла или других энергоресурсов и самостоятельно передающие данные изменений по сети интернет. Эти приборы обычно размещаются в квартирах, домах, промышленных помещениях для передачи показаний в систему сбора и учета данных. Очень важно, чтобы эти данные были точными, передавались своевременно и корректно фиксировались в базе данных.

Многие исследователи отмечают, что с ростом рынка интернета вещей производители IoT-устройств стремятся как можно быстрее выпустить новый прибор и не уделяют достаточного внимания вопросам, связанным с тестированием устройств и требованиям информационной безопасности. В работе [1] приводится описание стандартов IEEE для Internet of Things. Отмечается, что для обеспечения надежного функционирования системы IoT необходимо придерживаться стандартов обеспечения безопасности и внедрять тестирование устройств на этапе разработки и выпуска продукции [1]. Методы тестирования умных устройств касаются как функционирования самих устройств, так и взаимодействия с сетями. На сегодняшний день не существует единого подхода и стандарта тестирования устройств интернета вещей.

В работах [2, 3] авторы обращают внимание на то, что классический подход к тестированию сетевого оборудования чаще всего не применим для тестирования IoT-устройств. Полноценное тестирование оборудования Интернета вещей необходимо проводить комплексно, с разных сторон, начиная с проверки аппаратного обеспечения устройств и заканчивая

тестированием функциональности всей системы в целом. Однако некоторые этапы тестирования можно автоматизировать, например, испытание прошивок IoT-устройств.

На текущий момент на рынке нет подходящего инструмента, позволяющего проводить тестирование встроенного программного обеспечения автоматически. Эмуляция микроконтроллера для исполнения программ не является эффективным решением, потому что есть непредсказуемое влияние погрешностей эмуляции на результат работы программы [4]. Поэтому, чаще всего тестирование таких устройств проводится вручную. Например, на предприятии «НПО Карат», которое занимается разработкой и производством приборов учета энергоресурсов (водо- и теплоснабжение), для проведения испытаний составляется сценарий тестирования. В нем описаны операции, которые необходимо провести с прибором или модулем. Например, процедуры упаковки, прошивки и регистрация на платформе сбора данных. Также сценарий содержит служебные команды, которые позволяют проверить работоспособность и корректность исполнения функций устройством. Например, правильное формирование архивных записей, отправка данных по расписанию, реакция на возникновение внештатных событий.

Авторы в [5] описывают опыт автоматизации тестирования прошивок модемов. Автоматическими тестами удалось покрыть 75% функций модемов, а время на полноценную проверку прошивки сократилось до 2-3 недель. Тем не менее, компания продолжает проводить ручные испытания для отработки нетипичных сценариев использования.

Программа ReadyAPI [6] позволяет проводить тестирование устройств по протоколу MQTT (message queuing telemetry transport – сетевой протокол, используемый для обмена сообщениями между IoT-устройствами). Протокол MQTT работает поверх TCP/IP и основывается на модели обмена сообщениями по принципу издатель-подписчик. Его цель заключается в сборе данных с большого количества устройств и транспортировке данных в центр обработки данных [7]. Однако, ReadyAPI не обладает функционалом для динамического формирования сообщений (например, отсутствует функция для перевода сообщений в формат base64). Также в инструменте плохо реализовано управление MQTT-топиками. Данный недостаток не позволяет проводить массовое тестирование IoT-устройств, ограничивает применение набора тестов.

Таким образом, задача тестирования устройств интернета вещей на текущий момент не решена, является актуальной и востребованной для производителей устройств. Умное устройство «общается» с приложениями для управления или сбора данных через некоторую сеть. Чаще всего это беспроводные LPWAN сети. Такие сети работают с маленькими данными и на достаточно больших

расстояниях (рис.1). Большинство устройств учета энергоресурсов работает по беспроводной технологии LoRaWAN, которая является протоколом сетей LPWAN. LoRaWAN – широкополосный сетевой протокол (LPWAN) для энергоэффективных сетей дальнего радиуса действия, предназначенный для беспроводного подключения "вещей" с батарейным питанием к интернету в региональных, национальных или глобальных сетях и ориентированный на ключевые требования Интернета вещей (IoT), такие как двунаправленная связь, сквозная безопасность, мобильность и локализация услуг [6].

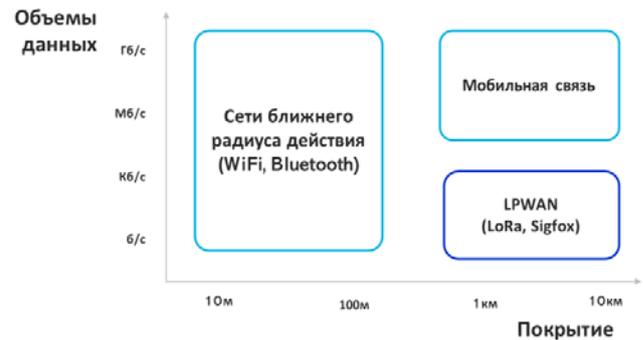


Рис. 1. Сравнение сетей

LoRaWAN имеет три различных класса оконечных устройств для удовлетворения различных потребностей [8].

1. Класс А – оконечные устройства с самой низкой мощностью. Связь всегда инициализируется оконечными устройствами и является полностью асинхронной. Каждая отправка данных может быть произведена в любое время, и за ней следуют два коротких окна, позволяющие при необходимости получить команды управления сетью.

2. Класс В – оконечные устройства с детерминированной задержкой нисходящей линии связи. В дополнение к иницилируемым окнам приема данных класса А, устройства синхронизируются с сетью в запланированные моменты времени. Задержка программируется до 128 секунд для разных приложений, а дополнительное потребление энергии достаточно низкое.

3. Класс С – оконечные устройства с самой низкой задержкой. Устройство постоянно «прослушивает» сеть во все время, пока не осуществляет отставку информации. Сетевой сервер может инициировать передачу данных в любое время, задержка отсутствует. Однако данный класс подходит для устройств с постоянным поддержанием энергии.

Асинхронная отправка данных подразумевает передачу данных только после выхода прибора на связь (в случае приборов класса А) или в любой момент времени (для приборов класса С). Выход на связь по расписанию позволяет экономить заряд батареи, поскольку прибор основную часть времени находится в спящем режиме.

Базовые станции прослушивают эфир, принимая сигнал интерфейсного модуля и транслируя его на

сервер сети. Сетевой сервер производит аутентификацию устройств и кадров в сети и управляет потоками сообщений, передавая данные на серверы приложений. Концепция обмена данными в сети LoRaWAN представлена на рисунке 2.

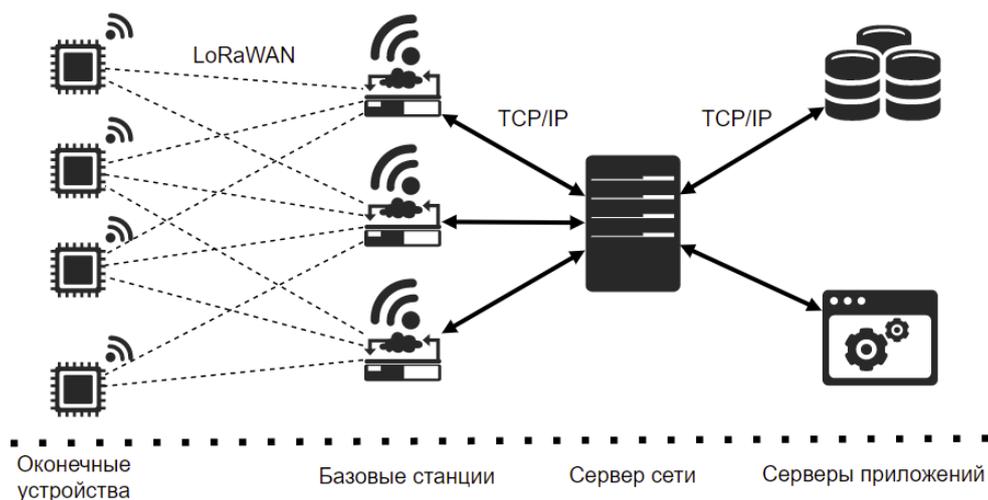


Рис. 2. Топология LoRaWAN

Для создания LoRaWAN сети необходим отдельный сетевой сервер. Примером данного сервера является стек технологий, применяемый на «НПО Карат», под названием ChirpStack. ChirpStack предоставляет компоненты с открытым исходным кодом для сетей LoRaWAN. Вместе они образуют готовое к использованию решение, включающее удобный веб-интерфейс для управления устройствами и API для интеграции. Модульная архитектура позволяет интегрироваться в существующие инфраструктуры. Все компоненты лицензированы по лицензии MIT и могут использоваться в коммерческих целях. Сети LoRaWAN работают в нелицензируемом диапазоне частот: 868 МГц. (Россия), обеспечивают надежную связь в условиях промышленных помех при этом, конечные устройства имеют низкое энергопотребление.

Нами было принято решение о создании платформы для автоматизированного тестирования встроенного программного обеспечения в устройства Интернета вещей.

Цель научно-исследовательской работы – разработать методологию тестирования «умных» устройств, на основе которой создать универсальную автоматизированную платформу тестирования устройств Интернета вещей.

Платформа должна представлять собой гибкую и универсальную систему и обладать следующими возможностями:

- создавать и детально настраивать сценарии тестирования;
- проводить испытания и формировать интерактивный отчет;
- управлять IoT-устройствами благодаря интеграции с разными платформами управления устройствами;
- собирать статистику о качестве связи.

II. ОБОРУДОВАНИЕ ДЛЯ ТЕСТИРОВАНИЯ

В качестве экспериментальной базы, позволяющей начать работу над платформой, было выбрано тестируемое устройство – электросчетчик Милур 307 с

модулем Карат-941LW. Устройство предоставлено компанией «НПО Карат». Счетчик Милур 307 – это статический трехфазный универсальный многотарифный прибор учета электрической энергии. Счетчик имеет встроенный специализированный микроконтроллер российского производства, разработка компании «Миландр». Прибор оснащен различными интерфейсами связи для обмена данными с внешними устройствами.

Для подключения прибора к сети LoRaWAN используется интерфейсный модуль Карат 941LW. Модуль представляет собой устройство, которое позволяет удаленно взаимодействовать с электросчетчиком. Связь между электросчетчиком и модулем осуществляется по протоколу Modbus [9]. Модуль способен выполнять команды, спускаемые сервером приложений LoRaWAN или управляющей платформой, и отправлять архивные данные. Модуль работает в классе С, постоянно прослушивая эфир, поскольку имеет активное питание от электросчетчика.

Предметом тестирования в данном случае является встроенное программное обеспечение интерфейсного модуля Карат 941LW, реализующее прикладной протокол обмена данными верхнего уровня. Модуль работает корректно, если данные передаются согласно сценарию использования. Например, при смене суток ожидается передача часовых и суточных архивов электросчетчика.

Автоматизация тестирования интерфейсного модуля Карат 941LW, установленного в счетчике Милур 307, позволит решить задачи научно-исследовательской работы, поскольку на данный момент испытания подобного оборудования на предприятии производятся вручную.

III. ОПИСАНИЕ ТЕСТИРОВАНИЯ УСТРОЙСТВ

Для ручной проверки применяются вспомогательные корпоративные инструменты для тестирования, например, программа LoraData (рис. 3). LoraData позволяет регистрировать тестируемые устройства на

сервере сети, передавать управляющие команды и получать ответные сообщения от приборов.

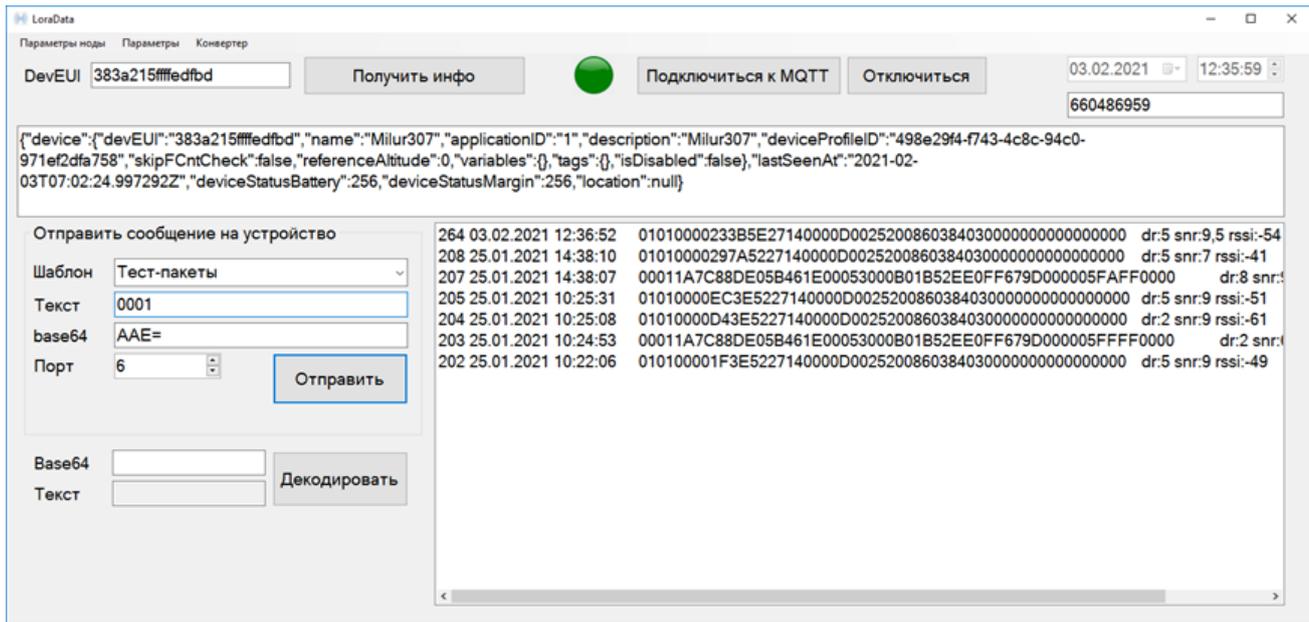


Рис. 3. Интерфейс программы ручного тестирования LoraData



Рис. 4. Схема обмена данными

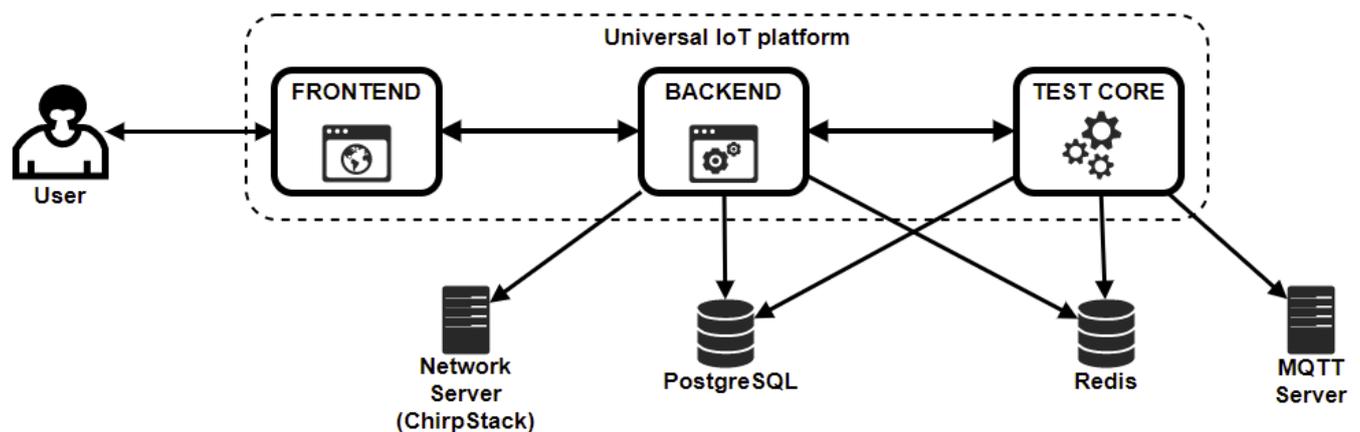


Рис. 5. Схема взаимодействия компонентов платформы тестирования

В данном примере в качестве протокола передачи данных на уровне приложения используется проприетарный протокол компании «НПО Карат». Управляющие и информационные команды имеют формат последовательности шестнадцатеричных

значений. Значения расположены друг за другом в порядке, определенном в протоколе передачи данных совместимых устройств Карат-9xx LW. Полезная нагрузка в полях данных имеет смысловое значение и представлена в десятичной системе счисления.

Ручное тестирование устройств в среднем отнимает 5 часов рабочего времени. При выходе новой версии встроенного программного обеспечения тестировщик

IV. РЕШЕНИЕ

Разработанное ядро автоматизированной платформы тестирования позволяет автоматически тестировать сетевые модули согласно подготовленному протоколу испытаний. Концептуальная схема обмена данными представлена на рисунке 4.

Система тестирования общается с сервером сети (ChirpStack) через API для регистрации и управления устройствами. Сервер сети взаимодействует с базовой станцией. Базовая станция транслирует сигналы на тестируемые модули LoRaWAN.

Систему тестирования можно логически поделить на три части: *front-end*, *back-end* и ядро тестирования (рис. 5). Платформа разработана с использованием технологий ASP.Net (*back-end* и ядро), а для *front-end* части применяется фреймворк *React*. В качестве СУБД *PostgreSQL*, для временного хранения данных используется *Redis*.

На вход сервису тестирования передается сценарий, содержащий несколько тестов. Сценарий задается в YAML-формате, который имеет минималистичный и понятный вид. Для простого теста достаточно двух параметров: команда для отправки и ожидаемое ответное сообщение. Для создания более специфичных тестов в файле можно изменить параметры, заданные по

вынужден повторно выполнять испытания. Усовершенствование процедуры проверки устройств позволит сократить ручную работу. умолчанию. Содержимое *yaml*-файла со сценарием тестирования представлено на рисунке 6.

```

1 convert_to_base64: true
2 retry: 1
3 timeout_retry: 15
4 scenario:
5   - test_name: "Check 0x0100 pack"
6     command: "0001"
7     result: "0001*"
8
9   - test_name: "Check 0x0101 pack"
10    command: "0101"
11    result: "0101*"
12
13  - test_name: "Check 0x1111 pack (for fail)"
14    command: "1111"
15    result: "1111*"
16
17
    
```

Рис. 6. Содержимое *yaml*-файла – сценарий тестирования

Процесс тестирования заключается в следующем: приложение отправляет на устройство запрос с командой и получает от него ответ. Взаимодействие происходит с помощью протокола MQTT. Далее, приложение анализирует полученный ответ и сравнивает его с ожидаемым откликом. При получении некорректного ответа регистрируется ошибка, означающая наличие ошибки в прошивке устройства. По результатам тестирования формируется отчет о работоспособности устройства. Пример отчета тестирования представлен на рисунке 7.

🔗 Платформа для тестирования устройств Интернета вещей

Главная / Группы испытаний / Карат / Испытания / Карат-941 / Результаты

КОЛОНКИ ФИЛЬТР РАЗМЕР ЭКСПОРТ

Дата создания	Название теста	Отправленная команда	Ожидаемый отклик	Полученный отклик	Результат
25.05.2021 19:57	Проверка команды 0001	0001	0001*	00011A7C88DE05B461E00053000B01B52EE0F...	✓
25.05.2021 19:57	Проверка команды 0101	0101	0101*	010100007F10ED27140000D002520086038403...	✓
25.05.2021 19:58	Проверка команды 1111	1111	1111*	11910201	✗

1-3 of 3

Рис. 7. Пример отчета тестирования

В файле отчета отображаются технология тестирования, ожидаемый отклик и результат теста.

Также указывается дата и время окончания тестирования. Представленный способ автоматического тестирования позволяет покрыть большую часть ручных

тестов. Тем не менее, некоторые специфичные тесты устройств требуется выполнять вручную. Например, проверить факт отключения реле электросчетчика при передаче специальной команды.

V. РЕЗУЛЬТАТЫ

В результате работы реализован следующий функционал:

- интеграция с сервером сети ChirpStack для получения данных об устройствах;
- отправка, получение и валидация ответа (такой функционал на данный момент покрывает 40% ручных тестов)
- редактор сценариев с подсветкой синтаксиса YAML;
- формирование отчета с результатами проведенных тестов по форме схожей с протоколом тестирования, используемом на предприятии «НПО Карат».

На данный момент работа над приложением продолжается. Необходима доработка механизма ядра тестирования для покрытия тестами большего числа функций.

VI. ЗАКЛЮЧЕНИЕ

Тестирование является важным этапом при разработке программного обеспечения, особенно при создании приборов и электронных модулей. В нашей работе создано ядро тестирования и разработана методология автоматизированного тестирования встроенного программного обеспечения модулей связи, работающих по технологии LoRaWAN. Для этого был проведен анализ проприетарного протокола и принципов ручного тестирования, которыми пользуются в настоящий момент на предприятии. Для выполнения тестирования прибора учета ядро тестирования отправляет на MQTT-сервер специальное сообщение, а затем считывает полученный ответ и сравнивает его с ожидаемым.

В дальнейшем планируется на базе разработанного приложения создать платформу, которая будет поддерживать работу с другими системами и технологиями. Особое внимание планируется уделить технологии NB-IoT (Narrow Band Internet of Things).

БЛАГОДАРНОСТИ

Авторы выражают благодарность компании НПО Карат в лице руководителя Инженерного центра Дмитриева Григория Александровича и генерального директора Ледовского Сергея Дмитриевича за помощь и предоставленное оборудование для апробации системы тестирования.

БИБЛИОГРАФИЯ

- [1] Наралиев Н.А., Самаль Д.И. Обзор и анализ стандартов и протоколов в области Интернет вещей. Современные методы тестирования и проблемы информационной безопасности IoT. *International Journal of Open Information Technologies*. 2019; 7(8): 94–104.
- [2] Кулик В. А., Киричек Р.В., Кучерявый А.Е. Программно-аппаратный комплекс для тестирования устройств интернета

вещей. *Информационные технологии и телекоммуникации*. 2015; 4(12): 67–76. Режим доступа: <http://www.sut.ru/doci/nauka/review/4-15.pdf>

- [3] Долгушев Р. А., Киричек Р.В., Кучерявый А. Е. Обзор возможных видов и методов тестирования интернет вещей. *Информационные технологии и телекоммуникации*. 2016; 4(2): 1–11.
- [4] Козлов А. Тестирование прошивок микроконтроллеров AVR [Электронный ресурс] // Платформа Medium. 2020. Режим доступа: https://medium.com/@anton_oxide/testirovanie-proshivok-mikrokontrollov-avr-f079045419c (дата обращения: 28.03.2021)
- [5] Юлаева Э., Иванов Р., Павлов Д., Автоматизированное тестирование прошивок в iRZ: от идеи до реализации. *Беспроводные технологии*. 2019; (4): 52–54.
- [6] ReadyAPI – платформа для проведения функционального тестирования с поддержкой протокола MQTT. Режим доступа: <https://smartbear.com/product/ready-api/overview/>
- [7] Селезнёв С.П., Яковлев В.В. Архитектура промышленных приложений IoT и протоколы AMQP, MQTT, JMS, REST, CoAP, XMPP, DDS. *International Journal of Open Information Technologies*. 2019; 7(5):17–28.
- [8] What is the LoRaWAN® Specification? [Электронный ресурс] // LoRa Alliance. 2019. Режим доступа: <https://loralliance.org/about-lorawan> (дата обращения: 26.01.2021).
- [9] Спецификация протокола Modbus [Электронный ресурс] // Организация Modbus. 2012. Режим доступа: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf, свободный (дата обращения 21.03.2021)

Core Platform Development for IoT-devices Automated Testing

N. V. Papulovskaya, I. N. Izotov, D. Yu. Blinichkin, A. Yu. Kataev

Abstract— The article deals with the testing problem that arises when developing devices for the Internet of Things (IoT) systems. Smart devices that provide data transmission over the Internet are becoming more and more popular. For example, energy metering devices that transmit data wirelessly are an integral part of the Internet of Things system in the housing and utilities sector. Currently, the firmware testing of water and heat consumption meters is performed manually. The manufacturer must spend a significant number of man-hours on this work. The article describes the results of research work on the development of an automated testing system. The experimental basis of the study is a static three-phase universal multi-tariff electric energy metering device Milur 307 with the Karat-941LW module. The meter has a built-in specialized microcontroller of the Russian manufacturer "Milander". The device is equipped with communication interfaces for data exchange with external devices. The subject of testing is the embedded software of the Karat 941LW interface module, which implements the top-level application data exchange protocol. For testing IoT devices, the authors have developed a special service—the testing core. The service polls the device according to the specified scenario recorded in the YAML file. The survey is conducted via the MQTT protocol, which is used by the LoRaWAN network server. As a result of testing, a report is generated. The developed testing methodology will be used to create a universal testing platform for devices operating over a wireless data network.

Keywords— IoT, testing, energy metering device, protocol, LoRaWAN, MQTT.

REFERENCES

- [1] Naraliyev N.A., Samal D.I. Review and analysis of standards and protocols in the field of Internet of Things. Modern testing methods and problems of information security IoT. *International Journal of Open Information Technologies*. 2019; 7(8): 94–104 (In Russ.)
- [2] Kulik V.A., Kirichek R.V., Koucheryavy A.E. Hardware-software complex for testing internet of things devices. *Telecom IT. Electronic scientific journal*. 2015; 4(12): 67–76. Available at: <http://www.sut.ru/doci/nauka/review/4-15.pdf> (In Russ.)
- [3] Dolgushev R., Kirichek R., Koucheryavy A.: An Overview of Possible Testing Types and Methods for the Internet of Things. *Telecom IT*. 2016; 4(2): 1–11 (In Russ.)
- [4] Kozlov A. Why do we need to test AVR microcontrollers? 2020; Available at: <https://maddevs.io/blog/avr-mcu-testing/>
- [5] Yulaeva E., Ivanov R., Pavlov D., Avtomatizirovannoe testirovanie proshivok v iRZ: ot idei do realizacii. *Besprovodnye tekhnologii*. 2019; (4): 52–54 (In Russ.)
- [6] ReadyAPI. Accelerating API Quality Through Testing. Available at: <https://smartbear.com/product/ready-api/overview/>
- [7] Seleznev S., Yakovlev V. Industrial Application Architecture IoT and protocols AMQP, MQTT, JMS, REST, CoAP, XMPP, DDS. *International Journal of Open Information Technologies*. 2019; 7(5):17–28 (In Russ.)
- [8] What is the LoRaWAN® Specification? LoRa Alliance. 2019. Available at: <https://lora-alliance.org/about-lorawan>.
- [9] MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3. 2012. Available at: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf