Введение в разработку и сопровождение систем, реализующих парадигму интернета вещей

В. А. Галатенко[1], К. А. Костюхин[2]

[1]Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», Москва, РФ, galat@niisi.ras.ru;

[2]Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», Москва, РФ, kost@niisi.ras.ru

Аннотация. В статье рассматривается архитектура типичной системы интернета вещей, выделяется набор требований к ее компонентам, и на основе этих требований формулируются требования к средствам разработки и сопровождения систем, реализующих парадигму интернета вещей.

Ключевые слова: интернет вещей, средства разработки, архитектура, требования

1. Введение

Интернет вещей (Internet of Things, IoT) описывает сеть физических объектов — «вещей», — обладающих аппаратными и программными интерфейсами для получения данных и обмена ими с другими устройствами и системами через Интернет. Сложность этих устройств варьируется от обычных бытовых предметов до промышленных механизмов [1].

Ключевой характеристикой IoT является тот факт, что этим устройствам необходимо постоянно взаимодействовать между собой (и, возможно, с пользователем) для передачи, обработки и приема постоянно меняющейся информации.

Можно сказать, что наделение повседневных объектов элементарным интеллектом и коммуникативными навыками дает нам широкий спектр технологических помощников:

- биометрическая одежда, оснащенная датчиками;
- беспилотные аппараты с автоматическим планированием доставки и построением маршрута;
- автоматический мониторинг и управление бытовой техникой в домах (концепция «умного» дома);
- «умные» медицинские приборы;
- системы управления в промышленном секторе и сельском хозяйстве;
- системы управления городской инфраструктурой (концепция «умного» города).

Этот список можно продолжать и дальше.

Реализация такого видения интернета вещей требует, чтобы компьютеры продолжали становиться меньше, «умнее» и поддерживали широкий спектр протоколов связи. В то время как соответствующие модификации аппаратного обеспечения ни у кого не вызывают сомнения, мало кто понимает, что существенных изменений требует также и встроенное программное обеспечение. Это также предъявляет ряд новых требований к средствам разработки систем интернета вещей.

Целью статьи является анализ архитектуры типичной системы интернета вещей и формулировка требований для аппаратного и программного обеспечения таких систем. В свою очередь, из этих требований логично следуют требования к средствам разработки систем интернета вещей, которые также будут сформулированы.

2. Архитектура IoT

На рисунке 1 представлена типичная архитектура системы, реализующей парадигму интернета вещей.

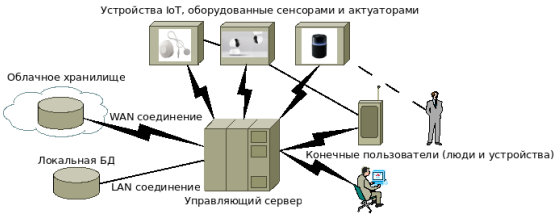

Рис. 1. Архитектура системы IoT

Такая система обычно состоит из компонентов четырех типов:

- устройства, оборудованные датчиками (сенсорами и актуаторами);
- компоненты, обеспечивающие коммуникацию;
- подсистемы обработки данных;

- устройства, предоставляющие пользовательский интерфейс.

2.1. Устройства, оборудованные датчиками

Это могут быть как совсем примитивные устройства, измеряющие температуру окружающей среды, так и достаточно сложные устройства, ведущие непрерывную видеотрансляцию.

Кроме того, устройства могут быть оснащены датчиками, для обеспечения непосредственной коммуникации с конечным пользователем, например, умная колонка обладает микрофоном для приема голосовых команд. В любом случае на первом этапе данные собираются из окружающей среды с помощью соответствующей аппаратуры.

2.2. Компоненты, обеспечивающие коммуникацию

Затем собранные данные отправляются на сервер или в облако. Здесь следует отметить широкий спектр возможных способов и протоколов связи [2]. Некоторые устройства обмениваются данными по беспроводной сети, используя 802.11 (Wi-Fi), Bluetooth, RFID, сети сотовой связи или технологии глобальной сети с низким энергопотреблением (LPWAN), такие как LoRa, SigFox или NB-IoT. Проводная связь подходит для стационарных устройств.

Каждый вариант не является идеальным и имеет компромиссы между энергопотреблением, дальностью действия и шириной полосы пропускания. Выбор наиболее подходящего варианта подключения зависит от конкретного применения системы интернета вещей, но все они выполняют одну и ту же задачу: осуществление коммуникации с сервером, с другими устройствами IoT или с конечным пользователем, которым может быть как человек, так и устройство.

2.3. Обработка данных

Как только данные попадают на сервер, в облако или на другое устройство IoT, начинается этап их обработки.

Это может быть простая проверка, например, того, что показания температуры находятся в пределах допустимого диапазона. Или же использование систем искусственного интеллекта для идентификации объектов в полученном видео (например, злоумышленников в доме).

Если результат обработки данных удовлетворяет заданным критериям, то обычно система просто продолжает работу в штатном режиме. Если же происходит нештатная с точки зрения логики работы системы ситуация, например, температура повысилась до критических значений или в доме появился неидентифицированный объект, то тогда, как правило, результат работы системы поступает конечному пользователю. Отметим, что конечный пользователь также может быть получателем результатов и штатной работы системы, например, пользователь так называемой «умной» колонки.

2.4. Пользовательский интерфейс

Результат обработки данных поступает к конечному пользователю по-разному. Это может быть сделано с помощью оповещения пользователя (электронная почта, текстовое сообщение, уведомление и т.д.). Например, текстовое оповещение, когда в холодильных камерах компании слишком высокая температура.

Кроме того, у пользователя может быть интерфейс, предоставляющий ему доступ к «сырым» необработанным данным. Например, к системе видеонаблюдения, чтобы самому увидеть «злоумышленника» и принять решение, что делать дальше.

Следует иметь ввиду, что пользователь также может иметь возможность выполнить определенные действия и повлиять на систему. Например, он может удаленно регулировать температуру в холодильной камере с помощью приложения на своем телефоне.

Заметим, что некоторые действия могут быть выполнены автоматически. Вместо того чтобы ждать, пока пользователь отрегулирует температуру, система может сделать это автоматически с помощью предопределенных правил. Аналогичным образом, система предупреждения незаконного вторжения может вместе с оповещением пользователя автоматически уведомлять соответствующие органы.

2.5. Протоколы передачи данных

Устройства IoT и конечные пользователи подключаются друг к другу, серверу или облаку через различные каналы связи, такие как Ethernet, Wi-Fi или модем 4G/3G.

Базовым протоколом связи обычно является UDP или TCP IP-протокол. Для простоты разработки и поддержки стандартизации используются такие протоколы, как MQTT, CoAP, XMPP, AMQP [3].

Протокол выбирается с учетом объема и частоты передачи данных, которыми необходимо обмениваться внутри системы IoT.

2.6. Управление данными

Управление данными включает в себя потоковую передачу данных, фильтрацию данных и хранение данных (в случае потери подключения к серверу или облаку).

Главное требование здесь заключается в том, чтобы свести к минимуму задержки и обеспечить корректность данных при передаче.

2.7. Менеджеры подключений

Эти компоненты отвечают за бесперебойное

подключение к серверу, облаку, а также обеспечивают надежность соединения устройств IoT между собой и с конечными пользователями.

2.8. Управляющий сервер IoT

Управляющий сервер должен иметь встроенные интеллектуальные возможности для анализа и принятия решения о том, какие данные следует передавать по сети дальше для обработки, а какие данные можно кэшировать для автономной обработки, чтобы сэкономить затраты на передачу данных и вычислительную мощность основной системы.

3. Основные технологии, используемые в IoT

Ниже приведены некоторые из наиболее часто используемых технологий в IoT.

Метки **RFID** [4] (Radio Frequency IDentification, радиочастотный код) и хранящийся на них EPC (Electronic Product Code, электронный код продукции) уникальный код, позволяющий компаниям отслеживать свою продукцию на протяжении всего жизненного цикла.

NFC [5] (Near Field Communication, связь ближнего действия) используется для обеспечения двустороннего взаимодействия между электронными устройствами, находящимися на расстоянии порядка 10 см друг от друга. Применяется, в основном, в смартфонах для совершения бесконтактных, например, платежных, операций.

Bluetooth используется для коммуникации на небольших расстояниях (примерно 10 м) между компонентами системы IoT. Преимущественно в портативных устройствах.

Z-Wave [6] – это технология радиочастотной связи с низким энергопотреблением. В основном, используется для управления бытовыми приборами посредством передачи простых управляющих команд с малыми временными задержками.

Sigfox [7] – сотовая беспроводная связь большой дальности действия с низкой пропускной способностью. Низкое энергопотребление гарантирует длительную работу удаленных устройств при минимальной зарядке аккумулятора или техническом обслуживании. Применяется в системах удаленного управления, мониторинга и слежения.

LTE-M [8] (Long-Term Evolution for Machine-Type Communications) разработана над стандартными сотовыми протоколами для обеспечения коммуникации устройств IoT между собой. В частности, эта технология должна удовлетворять таким требованиям, как низкая стоимость устройств, связь с большим покрытием, длительное время автономной работы устройств. Применяется, например, при реализации концепции «умных» городов.

NB-IoT [9] (Narrowband IoT) – это технология беспроводной связи, относящаяся к категории глобальных сетей с низким энергопотреблением (low-power wide-area networks, LPWAN), позволяющих подключать устройства, которым требуются небольшие объемы данных, низкая пропускная способность и длительное время автономной работы. Применяется, например, для считывания информации с различных датчиков: бытовых приборов учета, медицинских датчиков и т.д.

Zigbee [10] – беспроводная технология, разработанная как открытый стандарт подключения для создания недорогих беспроводных сетей передачи данных IoT с низким энергопотреблением. Предназначена, в основном, для передачи данных через зашумленные радиочастотные среды. Применяется в системах управления и мониторинга, при реализации концепции «умных» домов.

LoRaWAN [11] (Long Range Wide Area Network) – технология беспроводной связи, также относящаяся к классу LPWAN. Применяется в системах мониторинга, распределенного управления, в частности, при решении логистических задач.

Wi-Fi – это типичный способ коммуникации в локальных, например, домашних, системах IoT. При работе в локальной сети позволяет производить обмен достаточно большими массивами данных.

4. Требования к программному и аппаратному обеспечению систем IoT

Чтобы понять, какие средства разработки использовать для создания систем IoT, необходимо выделить требования к программному и аппаратному обеспечению, на которых строится система IoT. Далее в этом разделе мы сформулируем эти требования и уже на их основе опишем необходимые требования к средствам разработки систем IoT.

4.1. Вычислительная мощность

Управляющий сервер (пул серверов) системы IoT должен обладать серьезной вычислительной мощностью для обеспечения синхронной обработки большого количества входящих и исходящих запросов, работы с БД, а также, возможно, и для обеспечения работы некоторых систем искусственного интеллекта, например, нейросетей.

4.2. Оптимальный код и аппаратная архитектура

Устройства, используемые в системах IoT, обладают достаточно ограниченными ресурсами, как по размеру встроенной памяти, так и по быстродействию. Поэтому создаваемый программный код систем IoT должен иметь возможность оптимизации, как по размеру, так и по быстродействию. Отметим, что и аппаратная архитектура устройства IoT должна оптимально подходить для решения задач, которые выполняются на этом устройстве.

4.3. Широкий спектр поддерживаемых протоколов коммуникации

В системах IoT используется множество способов коммуникации устройств, поэтому, в первую очередь, на аппаратном уровне необходимо обеспечить гибкие возможности поддержки связи, используя, например, технологии из списка, приведенного в разделе 3.

4.4. Быстрая разработка

Поскольку системы IoT стремительно развиваются, их программное обеспечение должно постоянно оставаться актуальным и отвечать часто меняющимся требованиям. Поэтому средства и методы разработки должны быть адаптированы к такой ситуации. Разработка должна быть максимально простой с использованием современных инструментов.

4.5. Кроссплатформенность, переносимость и следование стандартам

С учетом большого количества устройств IoT с разной архитектурой необходимо, чтобы разработанное ПО с минимальными издержками переносилось на различные устройства или же изначально создавалось так, чтобы работать на широком спектре устройств. Достичь этого можно, например, используя интерпретируемые языки в качестве средств расширения функциональности, а также следуя промышленным стандартам в заданной предметной области. Кроме того, для встраиваемых систем и систем с критической миссией существуют собственные стандарты кодирования [12], которым также необходимо следовать.

4.6. Стандартизованные интерфейсы и масштабируемость

Для тех устройств IoT, которые взаимодействуют с конечными пользователями, важно иметь единообразный пользовательский интерфейс. Кроме того, для улучшения масштабируемости системы IoT, необходимо стандартизовать интерфейсы взаимодействия между устройствами.

4.7. Безопасность

Все устройства IoT должны быть защищены от несанкционированного доступа и устойчивы к взлому. При обеспечении информационной безопасности предпочтение следует отдавать аппаратным средствам. Чем ниже уровень, на котором применяются средства противодействия атакам, тем надежнее защищено устройство. В частности, крайне желательно иметь аппаратные средства шифрования.

4.8. Надежность

Все компоненты системы IoT должны быть надежными. Компонент, постоянно нуждающийся в перезагрузке, практически бесполезен. Поэтому и аппаратное, и программное обеспечение, создаваемое для Интернета вещей, должны соответствовать этому требованию. они должны быть надежными. Хотя любой язык и/или инструмент должны быть способны создавать надежное программное обеспечение, в качестве критериев следует рассматривать простоту и полноту тестирования.

4.9. Стабильность

Для компонентов системы IoT характерно свойство, что к моменту выхода с конвейера устройство морально устаревает. Необходимо обеспечить процесс обновления программного обеспечения «на лету», в полевых условиях. При этом устройство должно продолжать стабильно работать. Более того, отсюда проистекает еще одно требование к средствам разработки. При переходе на новые версии инструментов для создания аппаратно-программного обеспечения сами эти инструменты должны продолжать стабильно работать, обеспечивая обратную совместимость с текущим проектом.

5. Заключение

В работе была проведен анализ архитектуры типичной системы интернета вещей. На основе анализа выделены следующие требования.

Требования к аппаратуре.

- Разработка архитектуры аппаратной платформы, исходя из задач IoT, которые будут под ее управлением выполняться.
- Аппаратная поддержка протоколов и технологий связи, используемых в системах IoT.
- Аппаратная поддержка средств информационной безопасности, например, наличие аппаратных средств шифрования.

Требования к средствам разработки ПО.

- Наличие алгоритмов и средств оптимизации, нацеленных на разные способы оптимизации кода, например, по размеру кода или по быстродействию.
- Простая среда разработки (IDE), поддерживающая разработку на уровне исходного

языка, а также учитывающая особенности архитектуры целевой системы.
- Наличие простых средств расширения функциональности «на лету», например, с использованием интерпретируемых языков.
- Следование стандартам как кодирования, так и на используемые средства и языки разработки.
- Следование стандартам в той предметной области, для которой создается система IoT.
- Использование стандартных и безопасных коммуникационных протоколов (например, на уровне вызовов стандартизованных библиотек).
- Использование проверенных сертифицированных средств разработки, регулярные обновления инструментальной системы, особенно обновления системы безопасности.
- Использование систем тестирования, обеспечивающих максимальное тестовое покрытие.
- Обеспечение обратной совместимости средств разработки.

В качестве направления дальнейшей работы видится внедрение принципов контролируемого выполнения [13] при разработке и поддержке систем интернета вещей.

«Публикация выполнена в рамках государственного задания по проведению фундаментальных исследований по теме «Исследование и реализация программной платформы для перспективных многоядерных процессоров» (FNEF-2022-002).»

Introduction to the Development and Maintenance of Systems Implementing the Internet of Things Paradigm

Vladimir Galatenko, Konstantin Kostiukhin

Abstract. The article considers the architecture of a typical Internet of Things system, identifies a set of requirements for components of a typical Internet of Things system, and based on these requirements, the requirements for the development and maintenance of systems implementing the Internet of Things paradigm are formulated.

Keywords: IoT, development tools, architecture, requirements

Литература

1. Oracle: What is IoT?, https://www.oracle.com/internet-of-things/what-is-iot/
2. Deshmukh S., Deshmukh C. Study of Internet of Things and development tools and technology, International Journal of Advanced Innovative Technology in Engineering (IJAITE), Vol. 4, Issue 3, 2019, pp. 20-26.
3. 4 Key IoT Protocols – Learn In Great Detail, https://data-flair.training/blogs/iot-protocols/
4. What is RFID? The Beginner's Guide to How RFID Systems Work, https://www.atlasrfidstore.com/rfid-beginners-guide/
5. NFC Specifications, https://nfc-forum.org/build/specifications
6. Z-Wave Specifications, https://z-wavealliance.org/development-resources-overview/specification-for-developers/
7. Sigfox Device Radio Specifications, https://build.sigfox.com/sigfox-device-radio-specifications
8. Long Term Evolution for Machines: LTE-M, https://www.gsma.com/iot/long-term-evolution-machine-type-communication-lte-mtc-cat-m1/
9. Narrowband Internet of Things Whitepaper, https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1ma266/1MA266_0e_NB_IoT.pdf
10. Zigbee alliance. – Zigbee Specification, Revision 22 1.0, https://csa-iot.org/wp-content/uploads/2022/01/docs-05-3474-22-0csg-zigbee-specification-1.pdf
11. LoRa alliance, LoRaWAN Specification v1.1, https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-1
12. Motor Industry Software Reliability Association. – MISRA C Coding Standard, https://misra.org.uk/
13. Бетелин В.Б., Галатенко В.А., Костюхин К.А. Основные понятия контролируемого выполнения сложных систем, Информационные технологии, 2013, стр. 1-32.