

ПРОТОТИП АППАРАТНО–ПРОГРАММНОГО КОМПЛЕКСА УПРАВЛЕНИЯ ПРИЛОЖЕНИЯМИ «УМНОГО ДОМА»

А.А. Андрушевич¹, А.С. Вардак², И.С. Войтешенко²

¹ООО «Интелиджент семантик системс»,
ул. Сурганова, 48А, комната 16, Минск, 220013, Беларусь
E-mail: andrushevich@bsu.by

²Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь

В докладе рассматривается разработанный авторами прототип аппаратно-программного комплекса для применения в сфере интернета вещей. Обоснован выбор аппаратных средств и инструментария разработки: одноплатный компьютер Raspberry Pi; визуальная среда Node-RED, облачная платформа AWS IoT Core. Обоснован выбор сетевого протокола, брокера сообщений, программных библиотек. Сфера применения: «умный дом».

Ключевые слова: Raspberry Pi; Node-RED; AWS IoT Core; протокол MQTT; «умный дом»; освещенность жилища; энергопотребление и энергоэффективность жилища.

PROTOTYPE OF HARDWARE AND SOFTWARE COMPLEX FOR CONTROL OF "SMART HOME" APPLICATIONS

A.A. Andrushevich^a, A.S. Vardak^b, I.S. Vojteshenko^b

^aIntelligent Semantic Systems LLC,
st. Surganova, 48A, office 16, Minsk, 220013, Belarus
E-mail: andrushevich@bsu.by

^bBelarusian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus

The paper discusses the prototype of the hardware-software complex developed by the authors for use in the field of the Internet of Things. The choice of hardware and development tools is justified in the paper: single-board computer Raspberry Pi; Node-RED visual flow programming environment, AWS IoT Core cloud platform. The choice of a network protocol, a message broker, software libraries is also justified. Scope of application: "smart home".

Keywords: Raspberry Pi; Node-RED; AWS IoT Core; MQTT protocol; smart home; home lighting; power consumption and home energy efficiency.

Введение

Интернет вещей активно развивается. Исследования и разработки в этой сфере проводятся как организациями-разработчиками и консорциумами разработчиков, так и университетскими учеными и коллективами, включающими студентов, магистрантов и аспирантов. Университетскими научными коллективами Республики Беларусь тоже проводятся такие работы, укажем несколько последних по времени публикаций [1, 2].

Целью настоящей работы является создание прототипа аппаратно-программного комплекса для управления важнейшими приложениями в «умного дома». Прототип должен базироваться на недорогом компьютере и включать следующие приложения:

- приложение для управления освещенностью жилища;
- приложение для управления входом в помещение ("умный замок");
- приложение для управления энергопотреблением и энергоэффективностью жилища.

Предложенное решение должно допускать добавление других приложений.

1. Обоснование выбора аппаратных средств и программного инструментария

Состав как аппаратной, так и программной части комплекса определяется решаемыми задачами.

Состав аппаратной части. Аппаратная часть аппаратно-программного комплекса включает¹:

1. Одноплатный компьютер Raspberry Pi без монитора. Его выбор обусловлен, в первую очередь, невысокой стоимостью. Этими же соображениями обосновывают выбор компьютера и авторы схожих по тематике разработок [5,6].

Основные характеристики используемого компьютера:

- 64-битный четырехъядерный процессор ARMv8 Cortex-A72 с тактовой частотой 1.5 ГГц;
- Графический сопроцессор VideoCore VI®;
- Память LPDDR4 SDRAM объемом 4 ГБ;

¹ В настоящем докладе аппаратная и программные части аппаратно-программного комплекса рассматривается только применительно к приложениям для управления освещенностью и входом в помещение. Имитационное моделирование с использованием разработанного авторами доклада приложения для управления энергопотреблением и энергоэффективностью жилища рассмотрены в [2].

- Gigabit Ethernet;
- USB 3.0;
- 2 x micro-HDMI;
- 2.4 ГГц и 5 ГГц IEEE 802.11.b/g/n/ac WI-FI + Bluetooth 5.0 Low Energy (BLE).

2. Карта памяти 8 ГБ.

3. Веб камера 4 Мпикс.

4. Замок «Умный дверной замок Volibel Intelligent Lock Tuya X9s» со считывателем RFID-меток

5. Ethernet кабель для подключения замка и компьютера Raspberry Pi к wi-fi.

Часть устройств аппаратной части в настоящей версии прототипа эмулируется программным способом:

1. эмулятор детектора движения, на его основе можно включить/отключить свет;
2. эмулятор “умной” лампы со светорегулятором («диммером»), включение которой зависит от детектора движения;
3. эмулятор датчика освещенности в зависимости от времени суток и яркости умной лампы.

Программное обеспечение / используемые программные библиотеки. Основным инструментом разработки является визуальное средство разработки Node-RED, интегрированное с сервисами управляемой облачной платформы AWS IoT Core. Для передачи сообщений внутри системы используется протокол MQTT.

Node-RED² — это инструмент потокового программирования для соединения аппаратных устройств, API и онлайн сервисов. 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.

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

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

² Node-RED [Electronic resource] — Mode of access: <https://nodered.org/> — Date of access: 12.05.2022.

потока. Преимущества использования Node-RED для исследования и проектирования IoT систем:

- Сценарии, которые разрабатываются на Node-Red и правила их реализации не зависят от физической системы, поскольку он предоставляет возможности управления абстрактными устройствами. В то же время абстрактные устройства могут быть связаны с физическими устройствами.
- Визуальное программирование упрощает процесс разработки и создания прототипов.
- Возможности запуска на большом количестве программных платформ «из коробки», так как есть его образ для запуска под Docker.

Поэтому Node-Red уже нашел свое применение при разработке приложений Интернета Вещей, в том числе, относящихся к тематике «умного дома» [4, 5, 8].

Использование приложениями облачных сервисов, к которым относится AWS IoT Core³, позволяет значительно уменьшить затраты на инфраструктуру системы, при этом обеспечивая лучшую масштабируемость и отказоустойчивость. Облачные технологии предлагают вычислительные ресурсы, ресурсы для хранения данных и ресурсы коммуникации.

MQTT⁴ — сетевой протокол, использующий архитектуру издатель-подписчик и работающий поверх TCP/IP с использованием очередей (Message Queuing Telemetry Transport). Он поддерживается и при невысокой пропускной способности сети. Кроме минимизации количества передаваемой информации и требований к устройствам, данный протокол также старается обеспечить надежность доставки. Эти свойства сделали протокол MQTT подходящим для применения в сфере интернета Интернета Вещей (см., например, [7]).

RabbitMQ⁵ – брокер сообщений с открытым исходным кодом. Для передачи данных в низкоскоростных средах с высокой задержкой используется плагин MQTT Plugin RabbitMQ⁶, являющегося реализацией протокола MQTT.

При разработке приложения для управления входом в жилище использовались возможности доступных программных библиотек.

³ AWS IoT Core. — Режим доступа: <https://aws.amazon.com/ru/iot-core>. — Дата доступа: 12.05.2022

⁴ MQTT - The Standard for IoT Messaging. — Режим доступа: <https://mqtt.org/>. — Дата доступа: 4.05.2022

⁵ RabbitMQ. [Electronic resource] — Mode of access: <https://www.rabbitmq.com/> — Date of access: 12.05.2022

⁶ RabbitMQ. MQTT Plugin. [Electronic resource] — Mode of access: <https://www.rabbitmq.com/mqtt.html> / — Date of access: 12.05.2022

2. Результаты и их обсуждение

Общая схема разработанного прототипа аппаратно–программного комплекса управления приложениями «умного дома» приведена на рисунке.

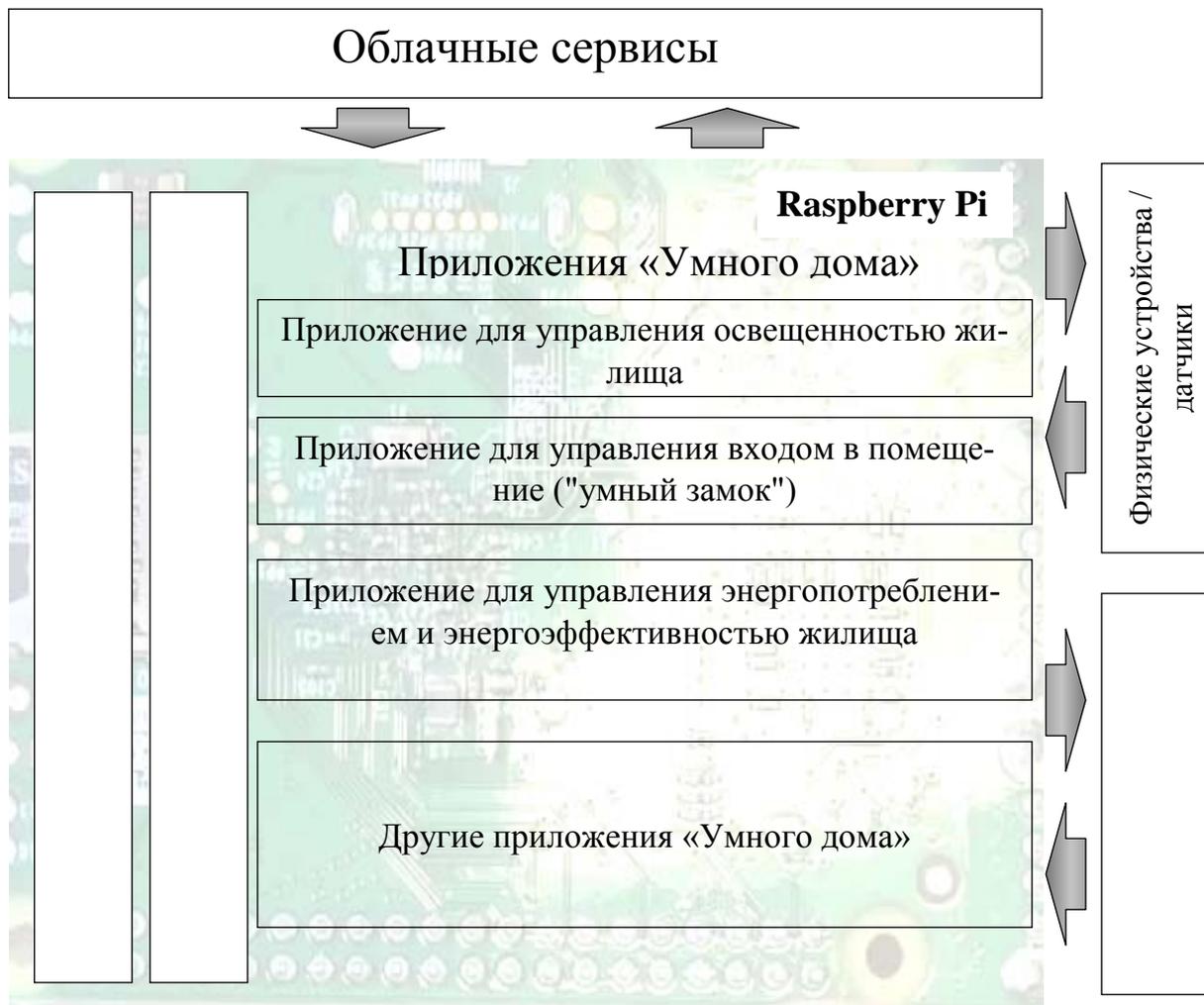


Рис. – Схема аппаратно-программного комплекса управления приложениями «умного дома»

Установка операционной системы и инициализация периферийных устройств. Установка операционной системы Raspberry Pi OS производится на основе сборки его образа на microSD карту. Операционную систему можно скачать с официального сайта [3]. Однако, к операционной системе нельзя подключиться без монитора. Мы использовали обходной путь посредством автоматической настройки сети на основе использования ssh — сетевого протокола прикладного уровня, позволяющего производить удалённое управление операционной системой и туннелирование TCP-соединений.

Процесс инициализации периферийных устройств работает на основе поэтапной проверки: если устройство не доступно, то процесс прерывается и происходит попытка автоматического исправления. Если попытка оказывается неудачной, то выдается сообщение об ошибке.

Установка Node-RED под Docker. При установке стандартного образа без пользовательской конфигурации и настройки, будет установлен весь пакет библиотек и зависимостей, что приведет к большой нагрузке на систему. Поэтому мы собрали свою модифицированную версию, потребляющую меньше ресурсов. Изменения вносились в конфигурационный докер-файл после построения образа. Для этого сначала устанавливали docker и docker-compose. Для нормального функционирования и критических важных зависимостей в образ был установлен python и библиотеки разработки.

Разработка приложений

Приложение для управления освещенностью жилища

Функциональные требования к приложению: при входе в помещение свет загорается, среагировав на движение, а через 10 секунд после выхода из помещения свет отключается. Яркость освещения должна корректироваться с учетом уровня уличного освещения, проникающего в жилище. Кроме того, освещение с 23-00 до 6-00 утра должно работать в режиме ночника, с пониженной яркостью.

Разработка приложения проведена приблизительно в соответствии со схемой, изложенной в [4] и здесь не излагается подробно.

Приложение для управления входом в помещение ("умный замок").

Основной частью приложения является созданная в рамках настоящей работы экспериментальная система распознавания лиц. Она использует машинное обучение, разработана и функционирует на основе библиотек dlib, cmake, OpenCV, face-recognition и streamlit. Для управления средой применен кроссплатформенный менеджер пакетов conda.

Текущий вариант приложения следует стандартным процедурам и алгоритмам биометрической аутентификации и не обеспечивает в полной мере защиту от взлома через подмену изображением считываемого с камеры лица пользователя. Для усиления процедуры биометрической аутентификации предполагается применение одного из методов Face Liveness Detection [9, 10], обеспечивающего проверку предъявленного идентификатора (изображения лица) на принадлежность «живому» пользователю.

Заключение

Разработанный прототип базируется на недорогом аппаратном обеспечении и, в основном, для своей разработки использует свободно рас-

пространяемое программное обеспечение, в том числе с открытым исходным кодом.

Прототип прошел первичное тестирование и показал свою начальную работоспособность.

Разработка как отдельных приложений «умного» дома, так и аппаратно-программных комплексов, кроме своей непосредственной ценности, является полигоном для тестирования инструментальных средств разработки, коммуникационных протоколов и подходов к разработке стандартов, применимых в других сферах интернета вещей.

Библиографические ссылки

1. Vishniakou U.A., Shaya V.H. Modeling the internet of things network for monitoring audio information on the Amazon platform = Моделирование сети интернета вещей для мониторинга звуковой информации на платформе компании Амазон // Системный анализ и прикладная информатика. 2021. № 2. С. 28–33.
2. Андрушевич А.А., Войтешенко И.С., Емельянова О.Ю. О среде имитационного моделирования компонентов и приложений в Интернете вещей на основе инструментов визуального потокового программирования и облачных сервисов // Информатика. 2022. № 19(2). С. 100-116. doi.org/10.37661/1816-0301-2022-19-2-100-116.
3. Raspberry Pi OS. URL: [https://www.raspberrypi.com /software/](https://www.raspberrypi.com/software/). (дата обращения: 4.05.2022.)
4. Програмируем управление освещением по датчикам движения и освещения на Node-RED/ [Electronic resource]. URL: <https://habr.com/ru/post/396985/> (date of access: 12.05.2022.)
5. Zare, Atefeh & Iqbal, Tariq. Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System. 2020. № 1–5. DOI: 10.1109/IEMTRONICS51293.2020.9216412
6. LOW COST WIRELESS HOME SECURITY SYSTEM USING RASPBERRY PI/Adarsh Sawant, IJPRET, 2015. Vol. 3 (9). P. 814–821.
7. Froiz-Míguez, I.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on MQTT and ZigBee-WiFi Sensor Nodes. Sensors 2018. 18, 2660. doi.org/10.3390/s18082660
8. Kodali, Ravi & Anjum, Arshiya. (2018). IoT Based HOME AUTOMATION Using Node-RED. 2018. 386–390. 10.1109/ICGCIoT.2018.8753085
9. Shweta Policepatil, Sanjeevakumar M. Hatture Face Liveness Detection: An Overview // International Journal of Scientific Research in Science and Technology. 2021. URL: <https://ijsrst.com/paper/8266.pdf>.
10. Золотарев В.В., Поважнюк А.О., Маро Е.А. Методы усиления // Известия ЮФУ. Технические науки. 2022. № 2. С. 212–225. DOI 10.18522/2311-3103-2022-2-212-225.