ВЕСТНИК НАЦИОНАЛЬНОГО ИССЛЕДОВАТЕЛЬСКОГО ЯДЕРНОГО УНИВЕРСИТЕТА “МИФИ”, 2022, том 11, № 3, с. 248–253

ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

УДК 004.627

РАЗРАБОТКА ПРОГРАММНОГО ИНТЕРФЕЙСА ВЗАИМОДЕЙСТВИЯ ПЛАТФОРМЫ HOME ASSISTANT И ГРАФИЧЕСКОГО ФРЕЙМВОРКА TOUCHGFX НА БАЗЕ МИКРОКОНТРОЛЛЕРА STM32

© 2022 г. И. А. Митяков[1,*], А. М. Жариков[2,], Д. М. Козин[2,***], П. В. Некрасов[1,****]**

[1]Национальный исследовательский ядерный университет “МИФИ”, Москва, 115409, Россия

[2]АО “ЭНПО СПЭЛС”, Москва, 115409, Россия

**e-mail: mityakov_ia@mail.ru*

***e-mail: AMZharikov@mephi.ru*

****e-mail: driver3@list.ru*

*****e-mail: pvnek@spels.ru*

Поступила в редакцию 29.07.2022 г.
После доработки 11.08.2022 г.
Принята к публикации 23.08.2022 г.

Технологии интернета вещей в последнее десятилетие активно развиваются, что приводит к появлению на рынке новых сопутствующих программных продуктов. Широкое распространение получила платформа Home assistant (HA) – программное обеспечение (ПО), представляющее собой систему управления умным домом. Данная статья посвящена разработке интерфейса взаимодействия между Home assistant и фреймворком TouchGFX, который используется при разработке устройств с графическим интерфейсом на базе микроконтроллеров STM32. Протокол MQTT, использующийся для подключения устройств к центральному серверу в описанной системе, позволяет взаимодействовать с заранее известными устройствами посредством использования уникального идентификатора, который формируется при подключении устройства в систему. Это исключает возможность динамического изменения состава системы (добавления/удаления устройств) без перенастройки всех взаимодействующих узлов. Поэтому было предложено разработать программное расширение стандартной конфигурации HA, которое позволило бы устройствам получать полную информацию о текущем состоянии системы. Для передачи информации об устройствах из динамического массива в графический интерфейс были использованы механизмы очередей операционной системы реального времени, а также шаблон проектирования Модель-Вид-Представитель. Автором предлагается метод, позволяющий оптимизировать работу системы “Умный дом” под управлением ПО Home assistant.

Ключевые слова: STM32, IoT, Home assistant, TouchGFX, умный дом, MVP, RTOS

DOI: 10.56304/S2304487X22030087

I. ВВЕДЕНИЕ

Главной частью системы интернета вещей (IoT), как правило, является управляющее ПО, которое отвечает за сбор, обработку, передачу информации, поступающей с устройств, входящих в состав системы, а также осуществление сценариев автоматизации и функций машинного обучения [1]. При создании собственной системы IoT возникает необходимость выбора программной платформы. На рынке представлено большое количество готовых управляющих программных систем [2], также возможна разработка собственной. Недостатком коммерческих систем являются их цена и ориентированность на собственную инфраструктуру. Создание собственного ПО ведет к значительным временным и экономическим затратам. Поэтому в качестве центральной системы в данной работе было выбрано ПО Home assistant. Данная платформа обладает всеми необходимыми свойствами: активное состояние разработки, открытый исходный код, поддержка большого числа протоколов взаимодействия с подключаемыми устройствами, возможность работы на многих платформах, включая Windows и Linux [3]. Задачей данной статьи являлась разработка программного интерфейса для интеграции HA в собственную аппаратную платформу, созданную на базе микроконтроллера (МК).

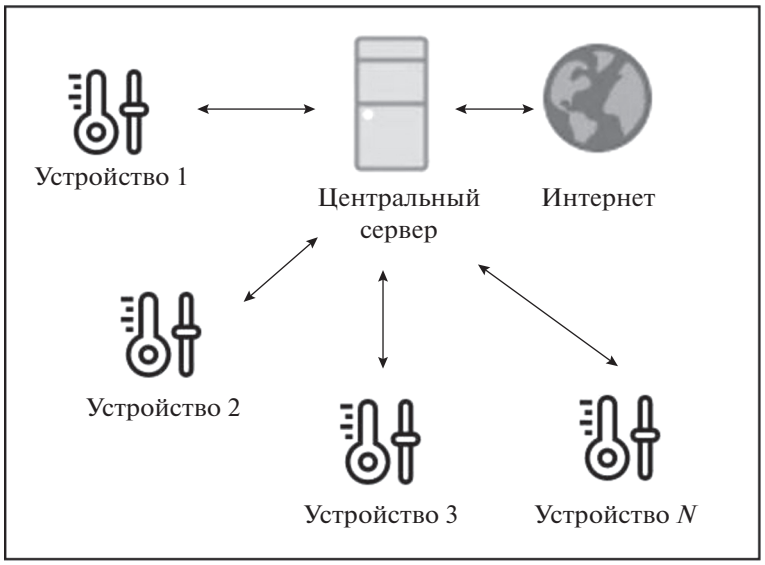

Рис. 1. Структурная схема системы умного дома.

II. СТРУКТУРА СИСТЕМЫ

A. Общая схема

Топология системы умного дома обычно включает в себя совокупность датчиков и исполнительных устройств, а также центральный сервер, на котором выполняется обозначенное выше управляющее ПО. Структурная схема приведена на рис. 1.

Двусторонний обмен информацией между центральным сервером и устройствами осуществляется по беспроводным каналам связи посредством протокола MQTT [4]. Этот протокол поддерживается системой Home assistant и позволяет подключать как существующие устройства, так и модели собственной разработки. В состав разработанной системы входят следующие устройства:

– панель контроля и управления (STM32H7);
– управляемая розетка (ESP32);
– диммер освещения (ESP32);
– универсальный автономный датчик (STM32L1);
– и др.

Одно из устройств с самыми большими функциональными возможностями, включая графический интерфейс, является панель контроля и управления. Именно поэтому это устройство будет использовано в качестве примера интеграции с HA.

B. Описание устройства панели контроля и управления

Панель контроля и управления системой (рис. 2) визуализирует состояние всех объектов, подключенных к Home assistant, получает информацию о состоянии датчиков и обеспечивает возможность управления исполнительными устройствами. Аппаратная часть панели реализована на базе высокопроизводительного микроконтроллера серии STM32H7, с подключенным к нему сенсорным LCD дисплеем для визуализации информации и управления. В качестве беспроводного интерфейса в устройстве установлен модуль Wi-Fi на базе SoC ESP8266, связанный с основным МК посредством интерфейса UART. Для создания графического интерфейса пользователя с поддержкой сенсорного управления была выбрана библиотека (фреймворк) TouchGFX, поскольку она содержит все современные элементы построения интерфейсов и поддерживает упрощенную интеграцию с МК фирмы STM32 [5].

C. Постановка проблемы

Протокол MQTT, использующийся для подключения устройств к центральному серверу в описанной системе, позволяет взаимодействовать с заранее известными устройствами посредством использования уникального идентификатора, который формируется при подключении устройства в систему. Это исключает возможность динамического изменения состава системы (добавления/удаления устройств) без перенастройки всех взаимодействующих узлов. Поэтому было предложено разработать программное расширение стандартной конфигурации HA, которое позволило бы устройствам получать полную информацию о текущем состоянии системы.

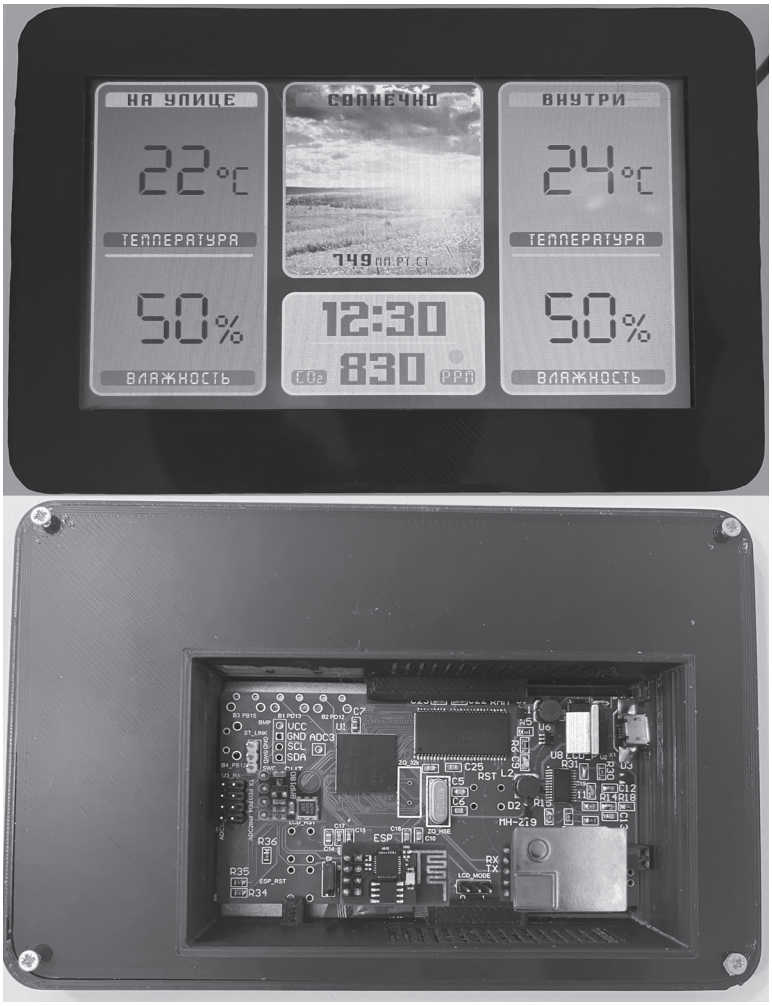

Рис. 2. Внешний вид устройства: передняя панель с дисплеем и печатная плата.

III. РАЗРАБОТКА ПРОГРАММНОГО ИНТЕРФЕЙСА

Общая структурная схема программного интерфейса приведена на рис. 3.

Основная идея, заложенная в создание описываемого метода – использование программного интерфейса (REST API) системы Home assistant для получения информации обо всех объектах и их свойствах, а также для реализации управления системой. Получаемый ответ от HA закодирован посредством формата JSON, однако данный способ сериализации данных существенно уступает ряду аналогов по таким параметрам, как объем занимаемой сообщением памяти, времени кодирования и декодирования [6]. Кроме того, формат JSON не детерминирован по объему сообщения, что может привести к переполнению оперативной памяти микроконтроллера при большом количестве объектов в системе умного дома.

Исходя из вышеописанного, в качестве протокола кодирования данных для передачи между HA, ESP8266, STM32 в разрабатываемой системе был выбран формат protobuf, который не содержит обозначенных недостатков и рекомендуется к применению в системах интернета вещей рядом авторов [6]. Для реализации данного метода был разработан формат сообщения, содержащий такие поля как: уникальный идентификатор устройства, отображаемое имя устройства, состояние устройства, единица измерения (опциональное поле для датчиков).

SoC ESP8266 инициирует запрос списка устройств, обращаясь к разработанному для HA дополнению (аддону) транслятора сообщений. Дополнение обращается к API Home assistant, перекодирует полученные данные из формата JSON в protobuf, после чего отправляет ответ обратно на ESP8266. Поскольку аппаратная платформа центрального сервера обладает значительно большими вычислительными мощностями, чем микроконтроллер, перекодирование гарантированно будет выполняться успешно. После приема сооб-

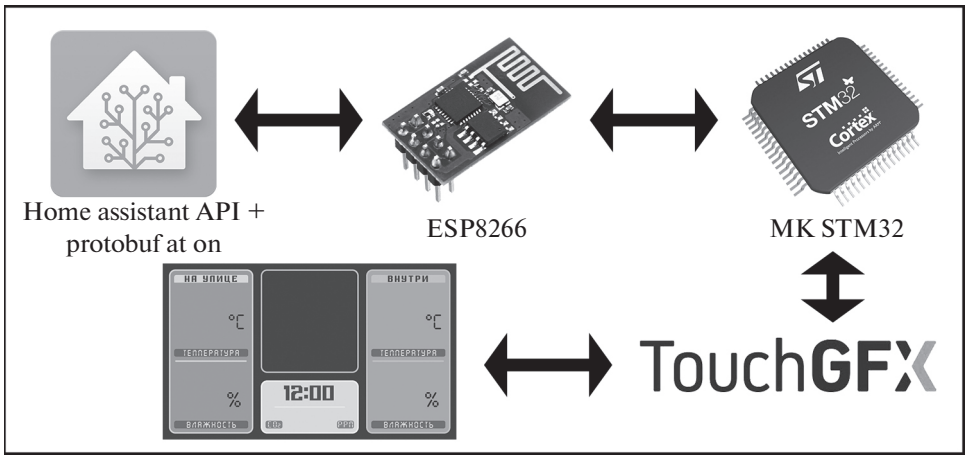

Рис. 3. Общая схема разработанного интерфейса.

щения ESP8266, не декодируя его, пересылает через последовательный интерфейс UART в микроконтроллер. STM32 принимает сообщение и, при необходимости, обновляет динамический список устройств и их параметров.

При изменении пользователем состояния какого-либо устройства с панели управления, информация об этом кодируется и передается в транслятор сообщений, который переформатирует посылку в формат JSON и возвращает ее в HA.

Дальнейшая визуализация устройств производится с использованием средств библиотеки TouchGFX. Данный метод, во-первых, ускоряет разработку интерфейсов, поскольку в комплекте поставки TouchGFX содержится программа-редактор интерфейсов, а также встроенный симулятор, дающий возможность отладить ПО без необходимости загрузки в микроконтроллер. Во-вторых, TouchGFX использует методы сжатия графических примитивов, что позволяет экономить до 70% объема оперативной памяти МК без потери качества [7].

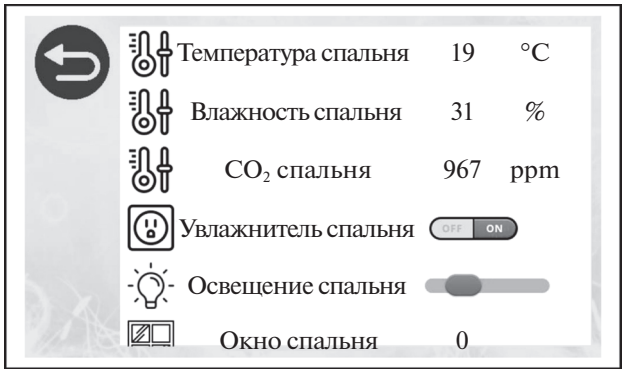

Рис. 4. Пример визуализации объектов.

В редакторе интерфейсов был создан универсальный шаблон для отображения карточки устройства, содержащий изображение, название, а также в зависимости от типа устройства органы управления: переключатель, ползунок, либо индикатор текущего значения и единицы измерения (для датчиков).

Для передачи принятой информации об устройствах из динамического массива в графический интерфейс используются механизмы очередей операционной системы реального времени [8], а также шаблон проектирования Модель-Вид-Представитель (MVP). Данный шаблон позволяет улучшить масштабируемость программного кода и упростить внедрение модульного тестирования [9].

IV. ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА

В рамках апробации предложенного метода была создана тестовая система умного дома на базе Home assistant, содержащая 100 устройств различных типов. Все устройства и соответствующие им органы управления были корректно визуализированы (рис. 4).

Было проведено измерение и сравнение размеров сообщения, содержащих информацию о 100 тестовых объектах, а также время декодирования данного сообщения микроконтроллером STM32 (ARM compiler 6, оптимизации отключены, частота ядра 200 МГц. Использованы библиотеки: nanopb-0.4.5, cJSON-1.7.15). Результаты сравнения приведены в таблице 1.

Таким образом, создание транслятора сообщений и использование формата сериализации данных protobuf позволили сократить размер передаваемого сообщения примерно на 90%, уменьшить время кодирования и декодирования сообщений на 70%.

Таблица 1. Сравнение производительности для двух типов сообщений

Параметр	Тип сообщения	
	JSON	protobuf
Размер, байт	31400	3300
Время кодирования, мс	10.63	3.52

V. ЗАКЛЮЧЕНИЕ

В данной статье был предложен метод, позволяющий оптимизировать работу системы "Умный дом" под управлением ПО Home assistant. Результат достигается путем создания дополнения – транслятора сообщений JSON – protobuf, и дальнейшей визуализации объектов, входящих в систему, при помощи библиотеки TouchGFX. Предложенный метод позволяет сократить количество используемых ресурсов микроконтроллера до 80%, исключить возможность возникновения сбоев в ПО, вызванных переполнением памяти во время декодирования сообщения, снизить нагрузку на каналы передачи информации (Wi-Fi, UART) до 10 раз.

СПИСОК ЛИТЕРАТУРЫ

1. *Humayed A., Lin J., Li F., Luo B.* Cyber-Physical Systems Security – A Survey // IEEE Internet of Things Journal, 2017. V. 4. № 6. P. 1802–1831. https://doi.org/10.1109/JIOT.2017.2703172
2. *Tanganelli G., Vallati C., Mingozzi E.* Rapid Prototyping of IoT Solutions: A Developer's Perspective // IEEE Internet Computing, 2019. V. 23. № 4. P. 43–52. https://doi.org/10.1109/MIC.2019.2927202
3. Home assistant. [Электронный ресурс]. URL: https://www.home-assistant.io. [дата обращения 20.02.2022].
4. The MQTT Protocol. [Электронный ресурс]. URL: http://www.mqtt.org. [дата обращения 20.02.2022].
5. *İşnas G., Şenyer N.* Comparison of TouchGFX and LVGL Embedded Hardware GUI Libraries // Gazi University Journal of Science Part C: Design and Technology, 2021. V. 9. № 3. P. 373–384. https://doi.org/10.29109/gujsc.915163
6. *Popić S., Pezer D., Mrazovac B., Teslić N.* Performance evaluation of using Protocol Buffers in the Internet of Things communication // International Conference on Smart Systems and Technologies (SST), 2016. P. 261–265. https://doi.org/10.1109/SST.2016.7765670.
7. The ST blog. [Электронный ресурс]. URL: https://blog.st.com/touchgfx/ [дата обращения 20.02.2022].
8. *Loskutov I.O. et al.* Investigation of Operating System Influence on Single Event Functional Interrupts Using Fault Injection and Hardware Error Detection in ARM Microcontroller // International Siberian Conference on Control and Communications (SIBCON), 2021. P. 1–4. https://doi.org/10.1109/SIBCON50419.2021.9438916.
9. *Esbai R., Erramdani M.* Model-to-model transformation in approach by modeling: From UML model to Model-View-Presenter and Dependency Injection patterns // 5th World Congress on Information and Communication Technologies (WICT), 2015. P. 1–6. https://doi.org/10.1109/WICT.2015.7489648.

Vestnik Natsional'nogo Issledovatel'skogo Yadernogo Universiteta "MIFI", 2022, vol. 11, no. 3, pp. 248–253

Design and Implementation of the Interaction Interface between Home Assistant and TOUCHGFX Based on an STM32 Microcontroller

I. A. Mityakov[a,#], A. M. Zharikov[a,b,##], D. A. Kozin[a,b,###], and P. V. Nekrasov[a,####]

[a] *National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, 115409 Russia*

[b] *Experimental Scientific Production Association Specialized Electronic Systems, Moscow, 115409 Russia*

[#]*e-mail: mityakov_ia@mail.ru*

[##]*e-mail: AMZharikov@mephi.ru*

[###]*e-mail: driver3@list.ru*

[####]*e-mail: pvnek@spels.ru*

Received July 29, 2022; revised August 11, 2022; accepted August 23, 2022

Abstract—Internet of things technologies has been rapidly developed in the past decade, stimulating the development of related software products. Home Assistant is a popular home automation software. The interaction interface between Home Assistant and TouchGFX framework, which is used in the development of

devices with a graphical user interface, which is based on an STM32 microcontroller, is presented in this work. The MQTT protocol, which is used to connect devices to the central server in the described system, allows the server to interact with previously known devices by using a unique identifier, which is generated when the device is connected to the system. This excludes the possibility of dynamically changing the composition of the system (adding/removing devices) without reconfiguring all interacting nodes. For this reason, it has been proposed to develop an addon for Home Assistant, which would allow devices to receive complete information on the current state of the system. To transfer received information on devices from a dynamic array to the graphical interface, real-time operating system queuing mechanisms, as well as the Model-View-Presenter design pattern, have been used. This pattern improves the scalability of the program code and simplifies the implementation of unit testing. A method has been proposed to optimize the operation of the Smart Home system under the control of the Home Assistant software.

Keywords: STM32 microcontroller, Internet of things, Home assistant, TouchGFX, smart home system, Model-View-Presenter, real-time operating system

DOI: 10.56304/S2304487X22030087

REFERENCES

1. Humayed A., Lin J., Li F., Luo B. Cyber-Physical Systems Security-A Survey. *IEEE Internet of Things Journal*, 2017, vol. 4, no. 6, pp. 1802–1831. https://doi.org/10.1109/JIOT.2017.2703172.
2. Tanganelli G., Vallati C., Mingozzi E. Rapid Prototyping of IoT Solutions: A Developer's Perspective. *IEEE Internet Computing*, 2019, vol. 23, no. 4, pp. 43–52. https://doi.org/10.1109/MIC.2019.2927202.
3. *Home assistant*. Available at: https://www.home-assistant.io. [accessed: February 20, 2022].
4. *The MQTT Protocol*. Available at: http://www.mqtt.org. [accessed: February 20, 2022].
5. İşnas G., Şenyer N. Comparison of TouchGFX and LVGL Embedded Hardware GUI Libraries. *Gazi University Journal of Science Part C: Design and Technology*, 2021, vol. 9, no. 3, pp. 373–384. https://doi.org/10.29109/gujsc.915163
6. Popić S., Pezer D., Mrazovac B., Teslić N. Performance evaluation of using Protocol Buffers in the Internet of Things communication. *International Conference on Smart Systems and Technologies (SST)*, 2016, pp. 261–265. https://doi.org/10.1109/SST.2016.7765670.
7. *The ST blog*. Available at: https://blog.st.com/touchgfx/ [accessed: February 20, 2022].
8. Loskutov I.O. et al. Investigation of Operating System Influence on Single Event Functional Interrupts Using Fault Injection and Hardware Error Detection in ARM Microcontroller. *International Siberian Conference on Control and Communications (SIBCON)*, 2021. pp. 1–4. https://doi.org/10.1109/SIBCON50419.2021.9438916.
9. Esbai R., Erramdani M. Model-to-model transformation in approach by modeling: From UML model to Model-View-Presenter and Dependency Injection patterns. *5th World Congress on Information and Communication Technologies (WICT)*, 2015, pp. 1–6. https://doi.org/10.1109/WICT.2015.7489648.