

**А.В. Мангушев****РЕАЛИЗАЦИЯ MQTT КЛИЕНТА НА БАЗЕ ОДНОКРИСТАЛЬНОГО  
МИКРОКОНТРОЛЛЕРА ДЛЯ ЗАДАЧ УДАЛЕННОГО УПРАВЛЕНИЯ  
ОБОРУДОВАНИЕМ**

*Описывается устройство, осуществляющее передачу данных и обработку команд по протоколу MQTT в сети Интернет. Изготовлена небольшая партия для оценки работоспособности в реальной практике. Программное обеспечение для микроконтроллера написано на языке Си с применением операционной системы реального времени (ОСРВ), позволяющей упорядочить разработку программного обеспечения за счет псевдопараллельного исполнения кода, механизмов синхронизации задач (семафоров, мьютексов), механизма очередей. В работе представлен пример алгоритма взаимодействия задач FreeRTOS: задачи ожидания команды от сервера, задачи измерения температуры, основного потока, выполняющего пересылку сообщения на сервер в ответ на появление строки в очереди сообщений. Ядро разрабатываемого MQTT-клиента составляют микроконтроллер STM32 и GSM/GPRS-модуль SIM800. На устройстве выведено большое число интерфейсов, что позволяет подключать самую разнообразную периферию (датчики температуры, влажности, давления, сервоприводы и т.д.). Устройство способно самостоятельно восстанавливать подключение к сети в случае его обрыва. Используется собственная реализация протокола MQTT версии 3.1 QoS 0 в программном обеспечении управляющего микроконтроллера. Сообщения MQTT передаются в рамках TCP-подключения, устанавливаемого средствами GSM-модуля. Управляющий микроконтроллер общается с GSM-модулем посредством AT-команд. Реализована возможность двустороннего обмена сообщениями в реальном времени, что позволяет использовать разработанную микропроцессорную систему как в режиме автономного управляющего устройства, так и в интерактивном режиме выполнения команд, поступающих по беспроводной сети, и отправки статусов исполнения этих команд. Главной особенностью MQTT-клиента является возможность удаленного обновления программного обеспечения микроконтроллера (OTA), реализованное с использованием вспомогательного микроконтроллера и микросхемы flash-памяти. Разработанный MQTT-клиент предназначен для использования в качестве основы микропроцессорных систем – клиентов Интернета вещей.*

*Интернет вещей; программно-аппаратное решение; mqtt; микроконтроллер; gsm; операционная система реального времени; rtos; ota.*

**A. V. Mangushev****MQTT CLIENT IMPLEMENTATION BASED ON A SINGLE-CHIP  
MICROCONTROLLER FOR REMOTE EQUIPMENT MANAGEMENT TASKS**

*The article describes a device that transmits data and processes commands over the MQTT protocol on the Internet. A small batch was made to assess the performance in real practice. The software for the microcontroller is written in C using a real-time operating system (RTOS), which allows you to streamline software development through pseudo-parallel code execution, task synchronization mechanisms (semaphores, mutexes), and a queue mechanism. The paper presents an example of an algorithm for the interaction of FreeRTOS tasks: the tasks of waiting for a command from the server, the tasks of measuring temperature, the main thread that sends a message to the server in response to the appearance of a line in the message queue. The core of the MQTT client being developed is the STM32 microcontroller and the GSM/GPRS SIM800 module. A large number of interfaces are displayed on the device, which allows you to connect a wide variety of peripherals (temperature, humidity, pressure sensors, servos, etc.). The device is able to independently restore the connection to the network in case of its interruption. The proprietary implementation of the MQTT protocol version 3.1 QoS 0 is used in the software of the control microcontroller. MQTT messages are transmitted as part of a TCP connection established by means of a*

*GSM module. The control microcontroller communicates with the GSM module via AT commands. The possibility of two-way real-time messaging is implemented, which allows using the developed microprocessor system both in the mode of an autonomous control device and in the interactive mode of executing commands received wirelessly and sending the execution statuses of these commands. The main feature of the MQTT client is the ability to remotely update the microcontroller software (OTA), implemented using an auxiliary microcontroller and a flash memory chip. The developed MQTT client is intended to be used as the basis of microprocessor systems - clients of the Internet of Things.*

*Internet of things; mqtt protocol; microcontroller; gsm; real-time operating system; rtos; ota.*

**Введение.** Со все большим распространением Интернета становится достаточно легко подключить разработанное устройство к сети, что позволяет с меньшим вмешательством человека автоматизировать различные процессы. При построении сетей Интернета вещей на первый план выходят протоколы машинно-машинного взаимодействия, среди которых наибольшее распространение получил протокол MQTT (англ. Message Query Telemetry Transport), реализующий обмен сообщениями в режиме «издатель-подписчик». Среди систем, использующих этот протокол, встречаются системы мониторинга и удаленного управления. Например, в [1] рассматривается разработка системы мониторинга для обнаружения частиц пыли, опасных газов, контроля температуры и влажности в окружающей среде, в [2] разработана система мониторинга загрязнения воздуха с использованием технологии Интернета вещей. В [3] предлагается прикладной протокол, позволяющий в рамках одного MQTT-подключения работать сразу с несколькими темами. В [4] представляется разработка и внедрение недорогой SCADA-системы с открытым исходным кодом. В [5] представлена гибридная носимая сенсорная сетевая система для мониторинга здоровья и повышения безопасности на рабочем месте. В [6] рассматривается разработка системы для отслеживания показаний энергопотребления в системе «умного дома». В [7] предлагается получать дополнительную энергию для носимых гаджетов из радиоволн и нагрева человеческого тела. В [8] был произведен анализ технологий Интернета вещей, используемых при уходе за пожилыми людьми. Поэтому разработка устройств – микропроцессорных систем, получающих и отправляющих в Интернет данные в режиме реального времени, обеспечивающих возможность оперативного отслеживания показаний датчиков и возможность удаленного управления различными приборами, при этом обладающих высокой надежностью и легко конфигурируемых программно, представляется актуальной [9].

#### Аппаратная часть MQTT-клиента.

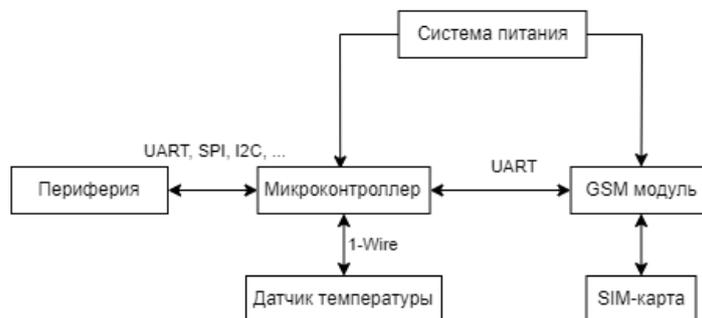


Рис. 1. Структурная схема устройства

На структурной схеме (рис. 1) представлены основные блоки устройства.

В качестве центрального микроконтроллера был применен STM32F446RE [10], поскольку имеет достаточную память и быстродействие, аппаратную поддержку вычислений с плавающей точкой, необходимых в перспективе использования разрабатываемого устройства в приложениях, требующих широкого использования цифровой обработки сигналов.

Для выхода в Интернет используется сеть GPRS, имеющая широкое покрытие и достаточную для передачи управляющих сообщений пропускную способность.

Используется широкофункциональный GSM/GPRS модуль SIM800 [11], поддерживающий подключение к сети при помощи стека TCP/IP. Управляющий микроконтроллер реализует прикладной протокол обмена данными в рамках сообщений, передаваемых по протоколу MQTT[12], который, в свою очередь, работает поверх протокола TCP. Само TCP-соединение осуществляется средствами GSM-модуля. Управление GSM-передатчиком производится через интерфейс UART посредством передачи строковых AT-команд. Также следует отметить, что у производителя данных модулей имеются также линейки для более современных стандартов связи (в первую очередь, NB IoT [13]) с практически полной сохранностью протокола общения, что в будущем позволит без особых сложностей перенести проект на другой класс беспроводных сетей.

В качестве протокола общения между пользователем и устройством был применен MQTT (англ. Message Query Telemetry Transport). К его особенностям можно отнести отсутствие высоких требований к производительности устройства и возможность широкоэвещательной передачи по системе издатель-подписчик. Сообщения передаются от издателя в конкретную тему (топик). Подписчики могут получать данные от множества издателей, в зависимости от подписки на соответствующие темы. Одно устройство может одновременно выступать и в качестве издателя, и в качестве подписчика. Реализуется данный протокол поверх TCP/IP [14].

Питание устройства осуществляется от напряжения 12В, что позволяет подключать соответствующую периферию. Микроконтроллер подключается через линейный преобразователь, но сначала напряжение понижается импульсным преобразователем для снижения нагрузки на линейный.

В текущей версии модуля выведены все основные аппаратные интерфейсы контроллера: I2C, UART, SPI, а также некоторое количество портов ввода вывода, что позволяет без труда подключать дополнительное оборудование. Во время отладки программы в качестве внешнего сенсора применялся датчик температуры DS18B20.

**Алгоритм работы устройства.** Программа для микроконтроллера основана на операционной системе реального времени (ОСРВ) FreeRTOS [15, 16], работа с периферийными устройствами осуществляется при помощи предоставляемой производителем микроконтроллеров библиотеки HAL, обеспечивающей переносимость кода между микроконтроллерами STM32 различных серий. Такое решение позволяет в будущем поддерживать код и добавлять новую функциональность.

Рассмотрим алгоритм работы системы с помощью диаграммы активностей (рис. 1, 2). Каждый из потоков (задач) обозначен в виде законченного алгоритма с бесконечным циклом. После подачи питания начинается выполнение инициализация GSM модуля, происходящая в задаче с самым высоким приоритетом (StartSIM800Task). В случае ошибки на каком-либо этапе инициализация начинается с начала. После её завершения эта задача удаляется, освобождая мьютекс (UARTMutex), отвечающий за доступ потоков к интерфейсу общения с GSM-модулем. Далее начинается свое выполнение задача, инициализирующая подключение к серверу и подписку на соответствующий топик (StartMQTTConnectTask). В случае нескольких неудачных попыток подключения происходит создание пер-

вой задачи (StartSIM800Task) и процесс повторяется. Далее этот поток также уничтожается, и в фоне начинают работать несколько потоков: задача по умолчанию (default\_task), задача отправки пинг-пакета (PINGStartTask), задача отправки данных на сервер (StartSIM800SendTask), задача обработки входящих команд (MessHandlerTask) и задача измерения температуры (StartGetTempTask). Для синхронизации работы потоков применяются счетные семафоры, очереди и мьютексы. Задача StartGetTempTask ожидает выставления семафора SemGetTemp, задача StartSIM800SendTask – освобождения мьютекса UARTMutex и наличия сообщений в очереди на передачу, задача MessHandlerTask – наличия новой строки в очереди команд. Задача PINGStartTask отправляет раз в несколько секунд пинг-пакет, поддерживающий соединение открытым. Замер времени между посылками пинг-пакета происходит при помощи таймера, обнуляемого при передаче какого-либо сообщения по соединению.

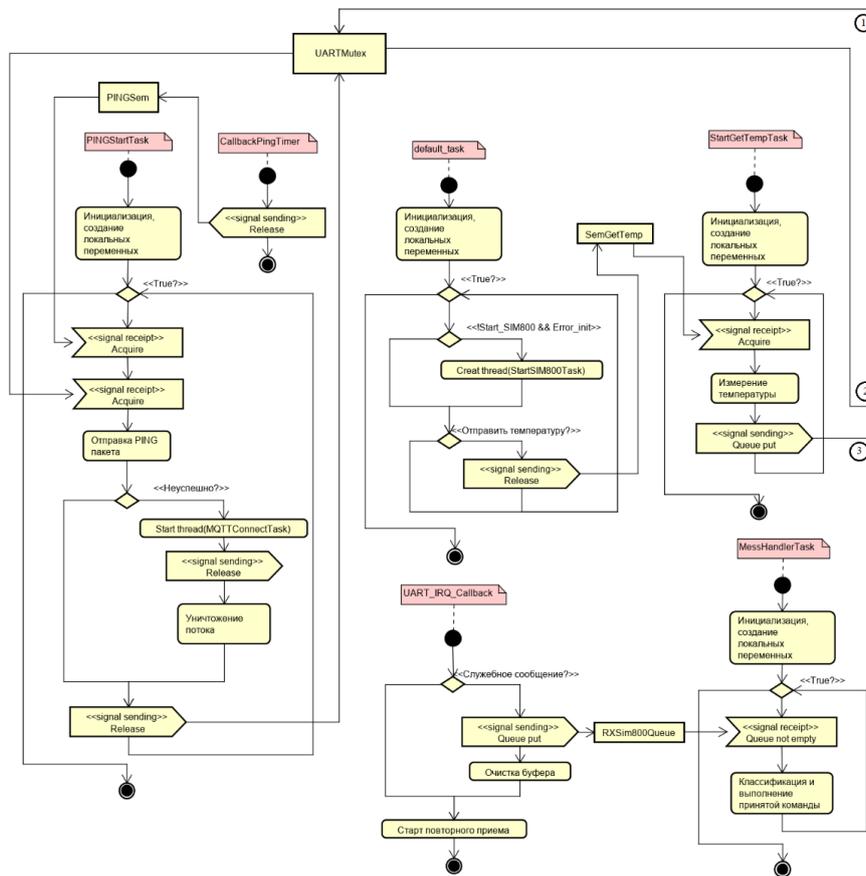


Рис. 2. Диаграмма активностей программы для микроконтроллера, часть 1

Задача по умолчанию переключает состояние светодиода, демонстрируя отсутствие зависания системы. В случае обнаружения ошибок в работе модуля происходит создание потока инициализации StartSIM800Task. Разбор сообщений от модуля происходит в задаче (MessHandlerTask): принятые от GSM-модуля строки, воспринимаемые как полезные данные, направляются в обработчике прерывания в соответствующую очередь (RXSim800Queue).

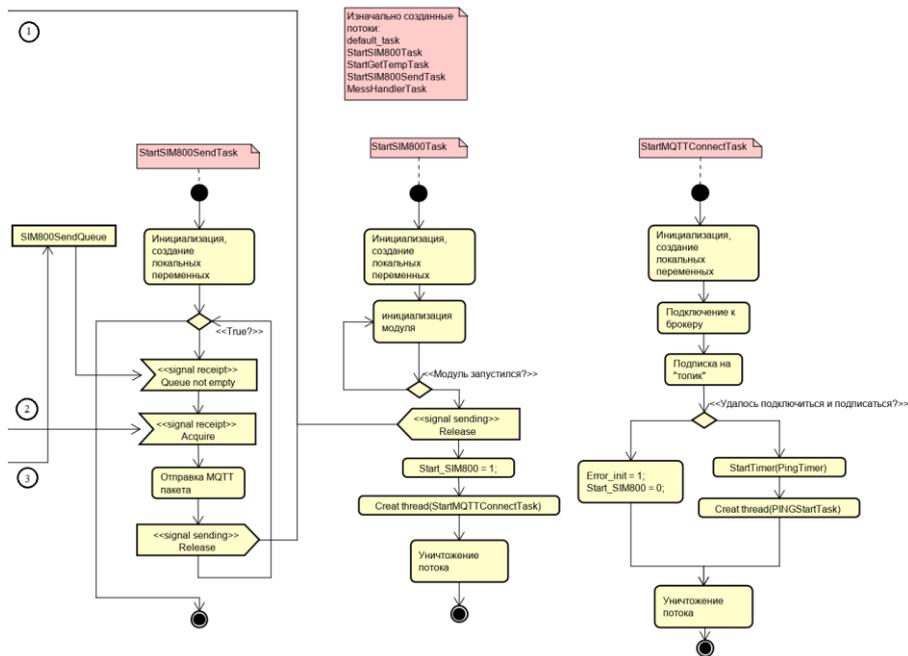


Рис. 3. Диаграмма активностей программы для микроконтроллера, часть 2

**Удаленное обновление программы микроконтроллера.** Одной из ключевых особенностей системы является возможность удаленно обновить программу микроконтроллера через Интернет (англ. over-the-air, OTA). Рассмотрим, каким образом данная система реализована в проекте, для чего обратимся к диаграмме последовательностей (рис. 4).

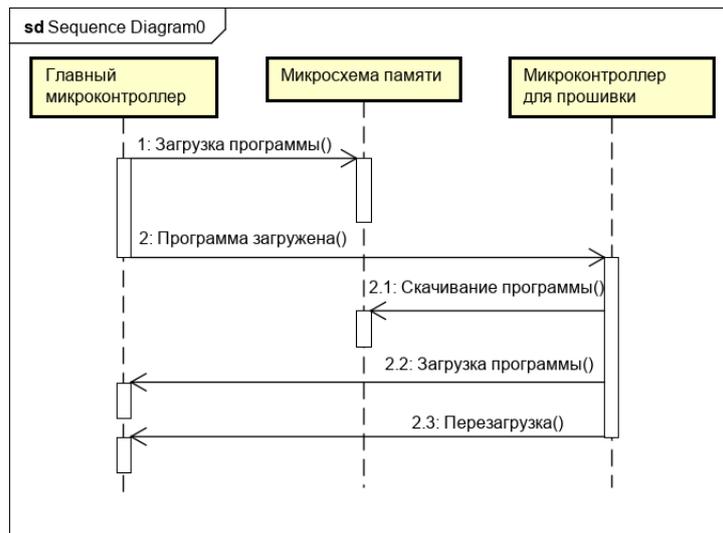


Рис. 4. Диаграмма последовательностей удаленного обновления программы для микроконтроллера

Для того чтобы обновить программу, мы предварительно сохраняем файл с ней на сервере и отправляем устройству команду обновления. После этого начнется загрузка файла с сервера по протоколу FTP. Инициализация FTP-подключения выполняется самим GSM-модулем, при этом управляющий микроконтроллер посылает GSM-модулю соответствующие AT-команды. Загруженный файл сохраняется на внешней микросхеме flash-памяти. Далее управление передается вспомогательному микроконтроллеру, который загрузит новую программу в основной микроконтроллер. В случае успешной загрузки вспомогательный микроконтроллер возвращает управление основному микроконтроллеру, и устройство переходит в штатный режим работы.

На рис. 5 в качестве главного микроконтроллера выступает F446, а в качестве вспомогательного – F103.

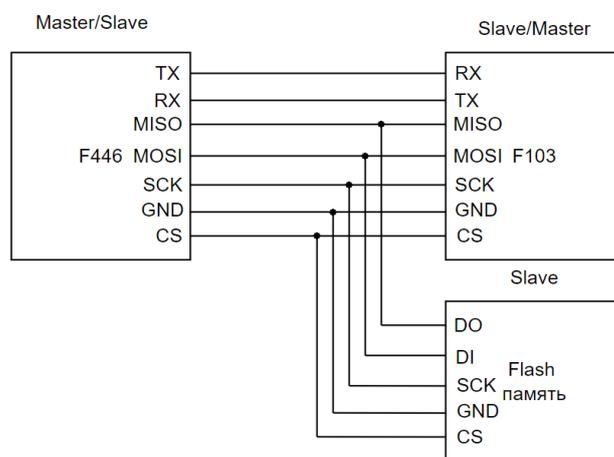


Рис. 5. Схема соединения микроконтроллеров и flash-памяти

Микросхема памяти подключается через интерфейс SPI, и возникает проблема того, что один микроконтроллер должен записать в нее данные, а другой считать, то есть на одной шине должны будут два устройства, способные быть ведущими на шине. Согласно протоколу SPI [17] можно подключать на одну шину более одного устройства, но в каждый момент времени только одно из них может быть ведущим на шине (master). Микросхема памяти всегда является ведомым устройством (slave), поэтому необходимо передавать полномочия ведущего между двумя микроконтроллерами. Для этого контроллеры соединены через интерфейс UART. Изначально оба устройства являются ведомыми. После начала процесса обновления основной контроллер переходит в режим ведущего и записывает данные в микросхему памяти, после чего становится ведомым и отправляет команду второму микроконтроллеру на разрешение стать ведущим на шине SPI. После загрузки программы в главный контроллер вспомогательный микроконтроллер вновь становится ведомым, и мы вновь имеем три ведомых устройства на шине. Такая конфигурация реализуема вследствие того, что для каждого из контроллеров можно настраивать его состояние на шине программным способом без изменения назначения выводов.

Однако в ходе испытаний выяснилось, что данное решение не применимо в описанном выше виде. Когда оба контроллера находятся в своем обычном состоянии все функционирует должным образом, однако при переводе основного микро-

контроллера в режим прошивки, состояние его ножек становится неопределенным и не позволяет подавать необходимые для работы с flash-памятью сигналы. Для решения данной проблемы необходимо отключать основной микроконтроллер от шины SPI, для чего была предложена следующая схема (рис. 6).

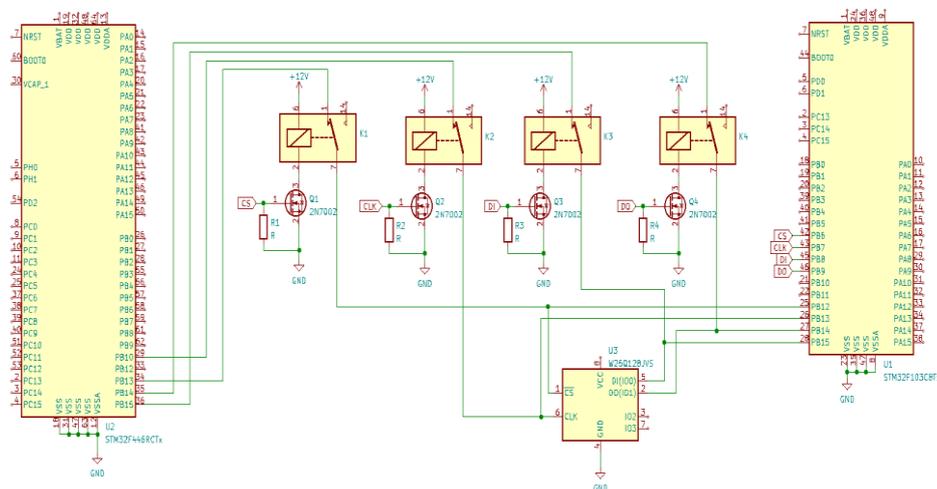


Рис. 6. Обновленная схема соединения микроконтроллеров и flash-памяти

По умолчанию на обмотки всех реле не поступает никакого напряжения и схема соединения эквивалента рис. 5, но после получения сигнала для начала обновления прошивки вспомогательный микроконтроллер подает напряжение на все обмотки реле и основной микроконтроллер отсоединяется от шины. После завершения обновления реле переходят в изначальное состояние.

В [18] рассматривается создание аналогичного устройства, однако оно имеет ряд недостатков, и в представленном виде не подходит под контрактное производство и промышленное применение. В качестве GSM/GPRS модуля применен SIM800L, являющийся не полнофункциональным клоном линейки модулей SIM800, не поддерживающим часть необходимых AT-команд, позволяющих упростить реализацию некоторых аспектов. Также устройство состоит из отдельных готовых модулей, что не совсем пригодно для контрактного производства и промышленного применения. Кроме того, программа для микроконтроллера в [18] составлена с использованием уже устаревшей библиотеки SPL для работы со встроенной периферией микроконтроллеров STM32 серии F1, что затрудняет перенос функциональности на микроконтроллеры других семейств. Разработанный прототип сводит наличие указанных недостатков к минимуму. В [19–20] рассматривается проблема уязвимости подобных систем к различным сетевым атакам, поэтому в следующей версии устройства особое внимание будет уделено улучшению сетевой безопасности, в частности, использование TLS-шифрования TCP-подключения, создаваемого GSM/GPRS-модулем.

**Приложение для взаимодействия с устройством.** Для более удобного взаимодействия с устройством было разработано приложение на языке C#. Для реализации графического интерфейса был применен фреймворк WPF (англ. Windows Presentation Foundation). На рис. 7 представлен скриншот окна разработанного приложения.

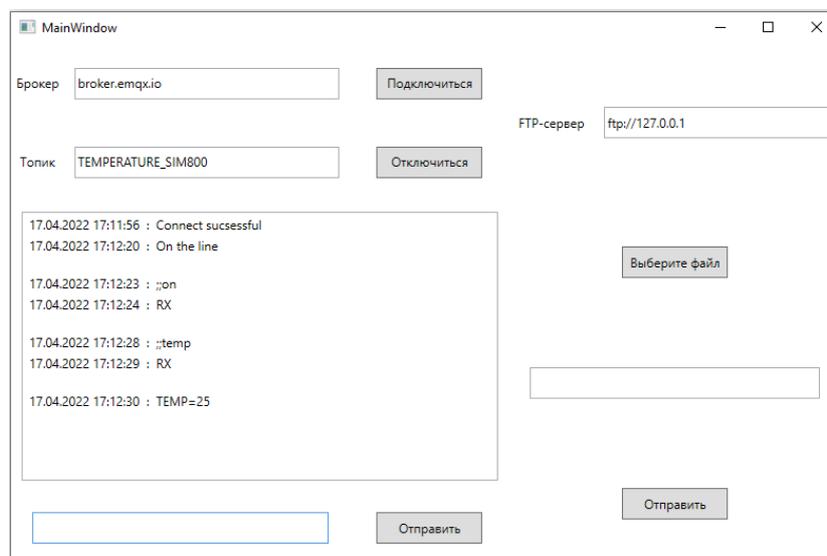


Рис. 7. Окно клиентского приложения

На данный момент приложение обладает следующим функционалом:

1. Подключение к MQTT брокеру.
2. Подписка на топик и отправка сообщений в него.
3. Подключение к FTP серверу с возможностью загрузки файла обновляемой программы.

**Заключение.** В результате проделанной работы была разработана и изготовлена первая версия устройства, обеспечивающая подключение к MQTT серверу и обработку некоторого количества команд. Кроме того, была реализована возможность удаленно обновить программу микроконтроллера для дальнейшего совершенствования устройства. В дальнейшем планируется учесть конструктивные недоработки первой версии и изготовить вторую, доработанную версию с целью отладки программного обеспечения в реальных условиях. Планируется также реализовать шифрование трафика с целью предотвращения несанкционированного доступа к управляемым устройствам.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Saidatul Norlyana Azemi *et al.* An IoT-Based Alarm Air Quality Monitoring System // J. Phys.: Conf. Ser. – 2021. – Vol. 1755. – 012035.
2. Hussein M., Galal A.I., Abd-Elrahman E., Zorkany M. Internet of Things (IoT) Platform for Multi-Topic Messaging // Energies. – 2020. – No. 13. – 3346. – <https://doi.org/10.3390/en13133346>.
3. Lawrence O., Aghenta M. Tariq Iqbal. Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, Things Board and MQTT protocol // J. AIMS Electronics and Electrical Engineering. – 2020. – No. 4 (1). – P. 57-86. – DOI: 10.3934/ElectrEng.2020.1.57.
4. Wu F., Wu T., Yuce M.R. An Internet-of-Things (IoT) Network System for Connected Safety and Health Monitoring Applications // Sensors. – 2019. – No. 19. – P. 21. – <https://doi.org/10.3390/s19010021>.
5. Saraereh O.A., Alsaraira A., Khan I., Choi B.J. A Hybrid Energy Harvesting Design for On-Body Internet-of-Things (IoT) Networks // Sensors. – 2020. – No. 20. – P. 407. – <https://doi.org/10.3390/s20020407>.
6. Sudarsono Novianta M.A., & Parwati, C.I. Application of Internet of Things (IoT) on air pollution monitoring database system // E3S Web of Conferences. – 2019.

7. *Syafaah L. et al.* Distance Measurement of ESP8266 for Control and Monitoring in Smart Home Application // *J. Phys.: Conf. Ser.* – 2019. – Vol. 1381. – 012058.
8. *Hongan Mu et al.* Discovering and Analyzing Internet of Things (IoT) Technologies and Applications for Aged Care // *J. Phys.: Conf. Ser.* – 2021. – Vol. 1948. – 012141.
9. *Ядгарова Ю.В.* Модель и алгоритм выбора программной архитектуры для систем Интернета вещей // Программные продукты и системы. – 2019. – № 4. – С. 682-689. – EDN QLQDSZ.
10. Микроконтроллеры линейки STM32F446. – URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32f446.html> (дата обращения: 29.05.2022).
11. Модуль SIM800. – URL: <https://www.simcom.com/> (дата обращения: 29.05.2022).
12. *Ли П.* Архитектура интернет вещей: пер. с англ. М.А. Райтмана. – М.: ДМК Пресс, 2019. – С. 282-293.
13. Стандарт NB-IoT: применение и перспективы // Беспроводные технологии. – 2019. – № 3 (56). – С. 27-31. – EDN SUHNQH.
14. *Батыев Б.* Работа со встроенным TCP/IP-стеком модулей GSM/GPRS серии SIM800 компании SIMCom Wireless Solutions // Беспроводные технологии. – 2016. – № 2 (43). – С. 38-43. – EDN XRLINH.
15. Операционная система реального времени FreeRTOS. – URL: <https://www.freertos.org/features.html> (дата обращения: 29.05.2022).
16. *Курниц А.* FreeRTOS - операционная система для микроконтроллеров // Компоненты и технологии. – 2011. – № 2 (115). – С. 96-100. – EDN NCDJHZ.
17. Цифровой синтез: практический курс / под общ. ред. А.Ю. Романова, Ю.В. Панчула. – М.: ДМК Пресс, 2020. – 556 с.
18. *Конченков В.И., Скакунов В.Н.* Реализация MQTT-клиента на основе микроконтроллера: учеб.-метод. пособие. – Волгоград: ВолгГТУ, 2021. – 64 с.
19. *Lee In.* Internet of Things (IoT) Cybersecurity: Literature Review and IoT Cyber Risk Management // *Future Internet.* – 2020. – Vol. 12. – P. 157.
20. *Spielvogel K., Pöhls H.C., Posegga J.* TLS Beyond the Broker: Enforcing Fine-Grained Security and Trust in Publish/Subscribe Environments for IoT. In: Roman R., Zhou J. (eds) // *Security and Trust Management. STM 2021. Lecture Notes in Computer Science.* – 2021.
21. – Vol. 13075. – Springer, Cham. – [https://doi.org/10.1007/978-3-030-91859-0\\_8](https://doi.org/10.1007/978-3-030-91859-0_8).

#### REFERENCE

1. *Saidatul Norlyana Azemi et al.* An IoT-Based Alarm Air Quality Monitoring System, *J. Phys.: Conf. Ser.*, 2021, Vol. 1755, 012035.
2. *Hussein M., Galal A.I., Abd-Elrahman E., Zorkany M.* Internet of Things (IoT) Platform for Multi-Topic Messaging, *Energies*, 2020, No. 13, 3346. Available at: <https://doi.org/10.3390/en13133346>.
3. *Lawrence O., Aghenta M. Tariq Iqbal.* Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, Things Board and MQTT protocol, *J. AIMS Electronics and Electrical Engineering*, 2020, No. 4 (1), pp. 57-86. DOI: 10.3934/ElectrEng.2020.1.57.
4. *Wu F., Wu T., Yuce M.R.* An Internet-of-Things (IoT) Network System for Connected Safety and Health Monitoring Applications, *Sensors*, 2019, No. 19, pp. 21. Available at: <https://doi.org/10.3390/s19010021>.
5. *Saraereh O.A., Alsaraira A., Khan I., Choi B.J.* A Hybrid Energy Harvesting Design for On-Body Internet-of-Things (IoT) Networks, *Sensors*, 2020, No. 20, pp. 407. Available at: <https://doi.org/10.3390/s20020407>.
6. *Sudarsono Novianta M.A., & Parwati, C.I.* Application of Internet of Things (IoT) on air pollution monitoring database system, *E3S Web of Conferences*, 2019.
7. *Syafaah L. et al.* Distance Measurement of ESP8266 for Control and Monitoring in Smart Home Application, *J. Phys.: Conf. Ser.*, 2019, Vol. 1381, 012058.
8. *Hongan Mu et al.* Discovering and Analyzing Internet of Things (IoT) Technologies and Applications for Aged Care, *J. Phys.: Conf. Ser.*, 2021, Vol. 1948, 012141.
9. *Yadgarova Yu.V.* Model' i algoritm vybora programmnoy arkhitektury dlya sistem Interneta veshchey [A model and algorithm for choosing a software architecture for Internet of Things systems], *Programmnye produkty i sistemy* [Software products and systems], 2019, No. 4, pp. 682-689. EDN QLQDSZ.

10. Mikrokontrollery lineyki STM32F446 [Microcontrollers of the STM32F446 line]. Available at: <https://www.st.com/en/microcontrollers-microprocessors/stm32f446.html> (accessed 29 May 2022).
11. Modul' SIM800 [SIM800 module]. Available at: <https://www.simcom.com/> (accessed 29 May 2022).
12. Li P. Arkhitektura internet veshchey [Architecture of the Internet of Things]: transl. from engl. by M.A. Raytmana. Мшчысцц: DMK Press, 2019б зз. 282-293.
13. Standart NB-IoT: primenenie i perspektivy [NB-IoT standard: application and prospects], *Besprovodnye tekhnologii* [Wireless technologies], 2019, No. 3 (56), pp. 27-31. EDN SUHNQH.
14. Batuev B. Rabota so vstroennym TCP/IP-stekom moduley GSM/GPRS serii SIM800 kompanii SIMCom Wireless Solutions [Working with the built-in TCP/IP stack of GSM/GPRS modules of the SIM800 series of SIMCom Wireless Solutions], *Besprovodnye tekhnologii* [Wireless Technologies], 2016, No. 2 (43), pp. 38-43. EDN XRLINH.
15. Operatsionnaya sistema real'nogo vremeni FreeRTOS [FreeRTOS real-time operating system]. Available at: <https://www.freertos.org/features.html> (accessed 29 May 2022).
16. Kurnits A. FreeRTOS - operatsionnaya sistema dlya mikrokontrollerov [FreeRTOS - operating system for microcontrollers], *Komponenty i tekhnologii* [Components and Technologies], 2011, No. 2 (115), pp. 96-100. EDN NCDJHZ.
17. Tsifrovoy sintez: prakticheskiy kurs [Digital synthesis: a practical course], under the general ed. A.Yu. Romanova, Yu.V. Panchula. Moscow: DMK Press, 2020, 556 p.
18. Konchenkov V.I., Skakunov V.N. Realizatsiya MQTT-klienta na osnove mikrokontrollera: ucheb.-metod. posobie [Implementation of an MQTT client based on a microcontroller: educational and methodical manual]. Volgograd: VolgGTU, 2021, 64 p.
19. Lee In. Internet of Things (IoT) Cybersecurity: Literature Review and IoT Cyber Risk Management, *Future Internet*, 2020, Vol. 12, pp. 157.
20. Spielvogel K., Pöhls H.C., Posegga J. TLS Beyond the Broker: Enforcing Fine-Grained Security and Trust in Publish/Subscribe Environments for IoT. In: Roman R., Zhou J. (eds), *Security and Trust in Management. STM 2021. Lecture Notes in Computer Science*, 2021, Vol. 13075. Springer, Cham. Available at: [https://doi.org/10.1007/978-3-030-91859-0\\_8](https://doi.org/10.1007/978-3-030-91859-0_8).

Статью рекомендовал к опубликованию д.т.н., профессор Н.Е. Сергеев.

**Мангушев Александр Вячеславович** – Волгоградский государственный технический университет; e-mail: [mangushev2001@yandex.ru](mailto:mangushev2001@yandex.ru); г. Волгоград, Россия; тел.: +79880522090; кафедра электронно-вычислительные машины и системы; студент.

**Mangushev Aleksandr Vyacheslavovich** – Volgograd State Technical University; e-mail: [mangushev2001@yandex.ru](mailto:mangushev2001@yandex.ru); Volgograd, Russia; phone: +79880522090; the department of electronic computing machines and systems; student.

УДК 621.315.611

DOI 10.18522/2311-3103-2022-3-84-103

**Н.К. Полуянович, Н.В. Азаров, М.Н. Дубяго**

### **НЕЙРОКОМПЬЮТЕРНОЕ УПРАВЛЕНИЕ ПРОПУСКНОЙ СПОСОБНОСТЬЮ КАБЕЛЬНЫХ СЕТЕЙ ПОСРЕДСТВОМ УЧЕТА И КОНТРОЛЯ ИХ ПАРАМЕТРОВ\***

*Рассматривается нейροкомпьютерная система прогнозирования ресурса силовой кабельной линии (СКЛ) с использованием нейросетевых технологий. Выбрана аппаратная модульная реализация нейροкомпьютера (НК) реализованной на базе ПЛИС. Для решения задачи прогнозирования термических процессов СКЛ принято решение использовать цифровой нейрончип NeuroMatrix NM6404 с переменной структурой из-за их высокой производительности по сравнению с потребляемой мощностью, высокой степенью универсально-*

\* Работа выполнена при поддержке гранта РНФ №22-29-00370.