ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
СПбГУТ))) Сентябрь 2017 Том 5 № 3 ISSN 2307-1303 http://www.itt.sut.ru
TELECOM IT
September 2017 Vol. 5 Iss. 3 ISSN 2307-1303 http://www.itt.sut.ru

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАБОТЫ С МИКРОСХЕМАМИ ПАМЯТИ С SPI-ИНТЕРФЕЙСОМ ПРИ ИССЛЕДОВАНИИ ИДЕНТИФИКАЦИИ СЕТЕВЫХ УСТРОЙСТВ

С. С. Владимиров[1*], А. К. Янковский[1]

[1] СПбГУТ, Санкт-Петербург, 193232, Российская Федерация
* Адрес для переписки: vladimirov.opds@gmail.com

Аннотация
Предмет исследования. Статья представляет разработанное авторами программное обеспечение, предназначенное для исследования микросхемам флеш-памяти с интерфейсом SPI на предмет пригодности их для однозначной идентификации сетевых устройств в сетях IoT. В статье рассматривается функционал, обеспечивающий работу с USB-SPI устройствами, и вопросы построения интерфейса пользователя. **Метод.** На основе необходимого функционала определены механизмы работы с микросхемами памяти и сформулированы требования к программе управления. Проведен сравнительный анализ существующего программного обеспечения с точки зрения привлечения его к реализации функций программы управления. **Основные результаты.** Реализованы базовые функции обеспечения параллельной работы с несколькими USB-SPI устройствами, функции стирания блоков памяти произвольного размера и интерфейс задаваемых пользователем команд управления микросхемой флеш-памяти. Разработан графический интерфейс пользователя. Выполнена оценка быстродействия разработанного ПО в сравнении с существующими аналогами. **Практическая значимость.** Представленная программа обеспечивает одновременную работу нескольких устройств USB-SPI на основе чипа CH341a при проведении исследования микросхем флеш-памяти для решения задачи однозначной идентификации сетевых устройств.

Ключевые слова
интерфейс SPI, флеш-память, устройства USB-SPI, микросхема CH341a.

Информация о статье
УДК 004.428, 004.072, 004.087.2
Язык статьи – русский.
Поступила в редакцию 20.06.17, принята к печати 01.09.17.

Ссылка для цитирования: Владимиров С. С., Янковский А. К. Программное обеспечение для работы с микросхемами памяти с SPI-интерфейсом при исследовании идентификации сетевых устройств // Информационные технологии и телекоммуникации. 2017. Том 5. № 3. С. 74–83.

THE SOFTWARE FOR WORKING WITH SPI FLASH MEMORY CHIPS IN THE STUDY OF NETWORK DEVICES IDENTIFICATION

S. Vladimirov[1*], A. Yankovskiy[1]

[1] SPbSUT, St. Petersburg, 193232, Russian Federation
* Corresponding author: vladimirov.opds@gmail.com

Abstract—Research subject. The article presents the software developed by the authors designed to investigate flash memory chips with the SPI interface for their suitability for unambiguous network devices identification in IoT networks. The article deals with the functional that provides operation with USB-SPI devices, and the construction of the user interface. **Method.** On the basis of the necessary functional, mechanisms for working with memory chips are defined and requirements for the control program are formulated. A comparative analysis of the existing software in terms of its involvement in the management program functions implementation is carried out. **Core results.** Basic functions for parallel operation with several USB-SPI devices, erase of random-size memory blocks and interface of user-defined commands for controlling the flash memory chip are implemented. A graphical user interface was developed. The performance of the developed software was compared with existing analogs. **Practical relevance.** The presented program provides simultaneous operation of several USB-SPI devices based on the chip CH341a during the research of flash memory chips for solving the problem of unambiguous identification of network devices.

Keywords—SPI interface, flash memory, USB-SPI devices, chip CH341a.

Article info
Article in Russian.
Received 20.06.17, accepted 01.09.17.

For citation: Vladimirov S., Yankovskiy A.: The Software for Working with SPI Flash Memory Chips in the Study of Network Devices Identification // Telecom IT. 2017. Vol. 5. Iss. 3. pp. 74–83 (in Russian).

Введение

Задача однозначной идентификации устройств является одним из основных вопросов, относящихся к информационной безопасности сетей передачи данных и Интернета вещей. В сетях передачи данных должна быть обеспечена невозможность подмены устройств ложными, что могло бы привести к осуществлению угроз безопасности, и невозможность незаконного использования сетевых устройств, например, мобильных терминалов сотовой связи. В последние годы проводятся исследования методик однозначной идентификации сетевых устройств, в том числе и в сетях IoT, по свойствам встроенного в них запоминающего устройства. Например, предлагается использовать встроенную в устройства флеш-память в качестве аппаратного генератора случайных чисел и источника уникальных идентификаторов устройства [1, 2]. На кафедре сетей связи и передачи данных Санкт-Петербургского государственного университета теле-

коммуникаций им. проф. М. А. Бонч-Бруевича было предложено [3, 4] использовать встраиваемую в устройства NOR флеш-память для формирования уникального идентификатора, предназначенного для однозначного определения устройства IoT в процессе сетевого взаимодействия. В основе предлагаемого метода используется свойство деградации ячеек флеш-памяти при проведении большого числа циклов перезаписи и стирания, когда отдельные ячейки, перестав изменять свое состояние, формируют так называемые «бэд-блоки», рисунок-образ которых уникален для каждого чипа памяти вследствие особенностей производства [1, 2, 5]. Для подтверждения принципиальной возможности данной процедуры было проведено исследование чипов NOR флеш-памяти нескольких производителей, отдельные секторы которых были принудительно деградированы для формирования образов-идентификаторов. Для принудительного деградирования секторов памяти использовался программно-аппаратный модуль на основе микрокомпьютера Raspberry Pi, имеющего встроенный интерфейс SPI для работы с чипами памяти [3]. Тем не менее, такой подход к подготовке микросхем памяти мало применим для исследования больших выборок микросхем по причине недостаточного быстродействия. Исходя из этого, решено использовать метод параллельной обработки чипов памяти посредством программно-аппаратного комплекса на основе персонального компьютера и устройств USB-SPI. В качестве базовой модели выбран USB-программатор на основе чипа CH341a компании WCH, предназначенный для работы с микросхемами памяти по интерфейсам I2C, SPI и UART и являющийся на сегодня самым бюджетным решением на рынке. Однако существующее для работы с ним ПО имеет ряд ограничений и непригодно для проведения необходимых исследований. В связи с этим, перед авторами возникла задача разработки программного обеспечения для эффективной работы в рамках разрабатываемого программно-аппаратного комплекса, предназначенного для исследования микросхем NOR флеш-памяти, работающих через интерфейс SPI.

Для ПО были сформулированы следующие требования:

1. Полное взаимодействие с чипами памяти по протоколу SPI: чтение, запись, стирание как всего чипа, так и отдельных участков памяти.

2. Возможность параллельной работы нескольких модулей USB-SPI в рамках одной операционной системы.

3. Режим отправки произвольных набираемых пользователем команд на SPI устройство.

Анализ существующего ПО для управления устройствами USB-SPI на основе чипа CH341a

Авторами протестировано и проанализировано следующее ПО, предназначенное для работы с выбранными устройствами USB-SPI на чипе CH341a:

1. Программа «P1119 CH341a Programmer» под ОС Windows для работы с чипами памяти, поставляемая с большинством программаторов данного типа.

2. Проект Flashrom.

3. Официальный драйвер CH341a от компании WCH под ОС Linux.

4. Проект Ch341Prog (открытая библиотека ch341a).

В результате было определено, что ни один из рассмотренных программных продуктов не удовлетворяет сформулированным требованиям. В частности, перечисленные программы и библиотеки обеспечивают лишь работу со всем объемом памяти чипа и не предполагают параллельную работу нескольких устройств USB-SPI, будучи предназначенными для чтения и загрузки прошивок сетевых устройств. Тем не менее, библиотека ch341a из проекта Ch341Prog была взята за основу разрабатываемого ПО, поскольку она содержит необходимые базовые функции, компактна и написана легко читаемым кодом.

Для выполнения поставленных перед разрабатываемым ПО требований необходимо реализовать следующие функции, в той или иной мере отсутствующие в исходной библиотеке:

1. Функция захвата и инициализации устройств USB-SPI.
2. Функция очистки модуля памяти блоками заданного размера.
3. Функция отправки заданных пользователем команд на модуль SPI.

Далее рассмотрим особенности реализации данных функций.

Захват и инициализация устройств USB-SPI

Идентификация USB устройств в операционной системе Linux использует следующие параметры:

- идентификатор устройства, состоящий из идентификатора производителя VID[1] и идентификатора самого устройства PID[2];
- номер шины интерфейса USB, к которой подключено устройство, и номер устройства, подключенного к шине.

Существующее ПО и, в частности, библиотека ch341a используют метод идентификации и последующего захвата USB-SPI устройств на чипе Ch341a исключительно на основе идентификаторов VID и PID. Учитывая, что все устройства, основанные на модуле Ch341a, имеют одинаковые идентификаторы, существующее ПО способно одновременно работать лишь с одним таким устройством и из-за этого неспособно обеспечить параллельную обработку нескольких чипов памяти.

Для обеспечения одновременной работы с несколькими USB-SPI устройствами на чипе Ch341a авторы использовали комбинацию двух вышеприведенных параметров идентификации. Устройства определяются и по идентификаторам VID/PID, и по номерам USB шины и подключенного к ней устройства. Это позволяет определять устройства по времени подключения: устройства, подключенные в более позднее время, имеют номер устройства на шине больший, чем у ранее подключенных устройств.

Блок-схема итогового алгоритма идентификации устройства с его последующими захватом и инициализацией представлена на рис. 1.

Подпрограмма, реализующая этот алгоритм, получает на вход номер шины и подключенного к ней устройства. Далее определяется список доступных устройств с VID и PID чипа Ch341a, которые в цикле проверяются на соответствие заданным номерам шины и устройства. После этого подпрограмма производит захват и инициализацию устройства. В случае если в списке не окажется нужного

[1] Vendor IDentificator.
[2] Product IDentificator.

устройства, или все возможные устройства будут заняты, подпрограмма вернет соответствующий код ошибки.

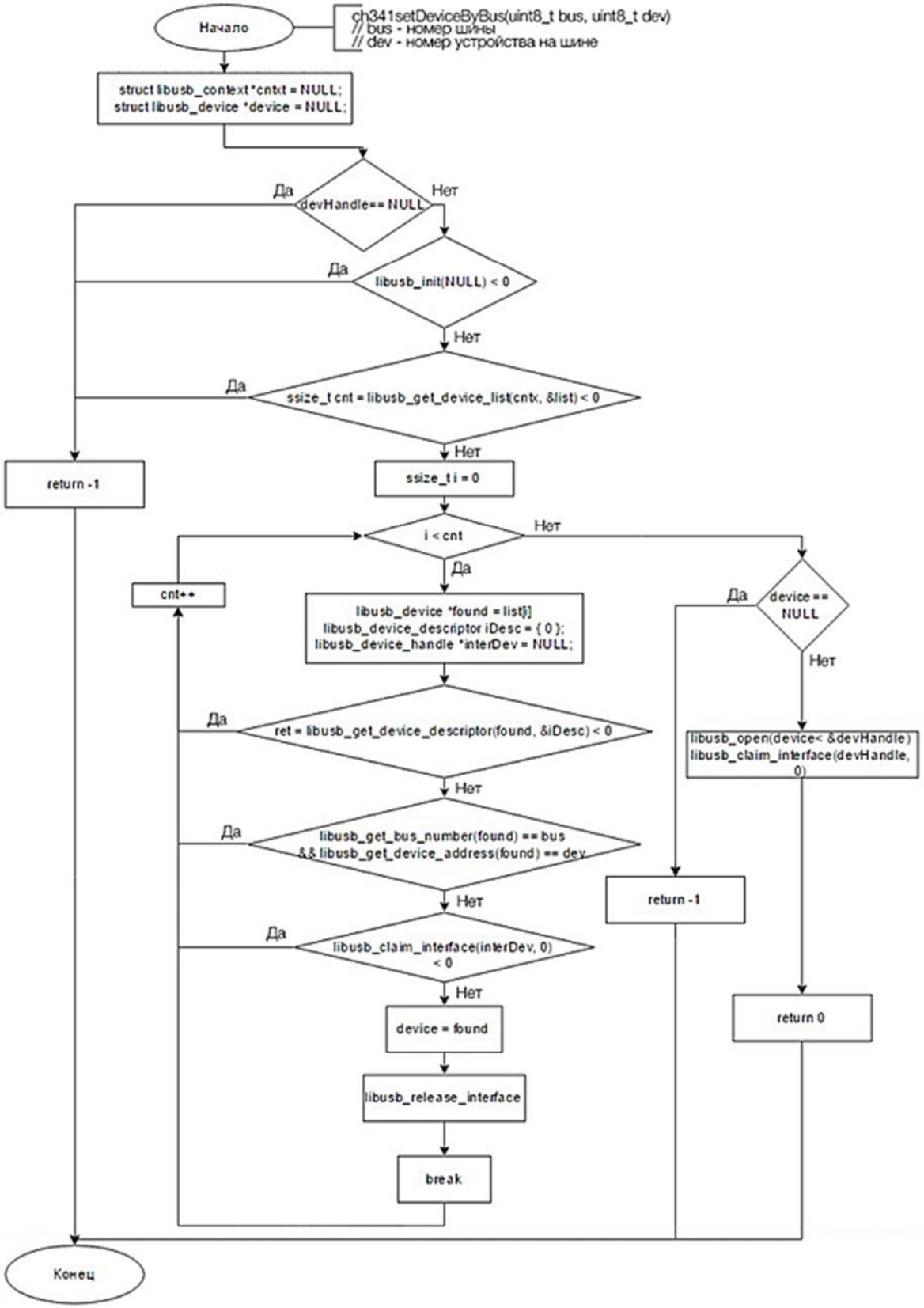

Рис. 1. Блок-схема алгоритма идентификации, захвата и инициализации USB устройства

Стирание блоков памяти заданного размера

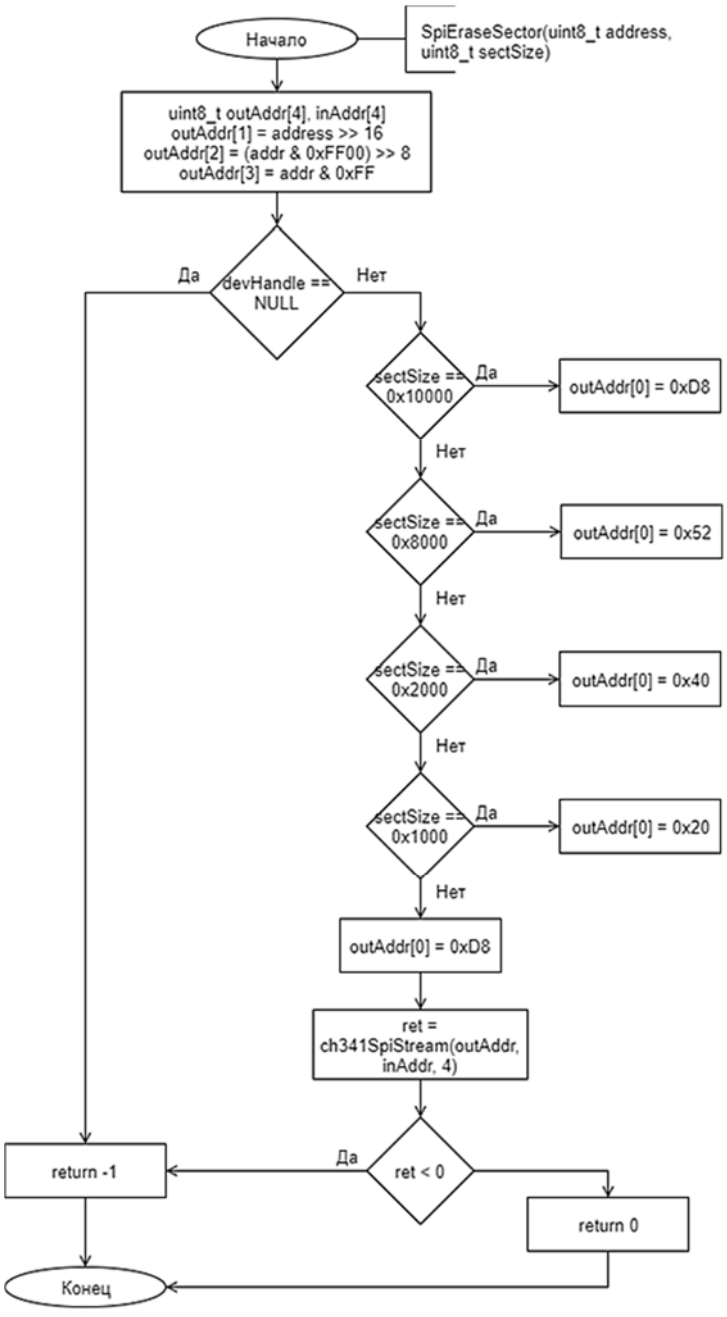

Рис. 2. Блок-схема функции стирания блока памяти

В имеющемся ПО реализован механизм стирания всего чипа памяти целиком. При этом само стирание производится либо поблочно, либо в потоковом режиме в зависимости от программы и вида чипа памяти. Поскольку при исследовании идентификации сетевого устройства предполагается проводить обработку участков памяти определенного размера (например, одного сектора памяти) [3, 4], возникла необходимость в реализации соответствующей функции.

В большинстве чипов NOR флеш-памяти процедура очистки реализуется отправкой серии команд: активация режима записи, команда очистки, деактивация режима записи. Данные команды являются общими для большинства таких микросхем. При использовании команд стирания статус завершения необходимо отслеживать отдельно посредством команд чтения статуса. Реализованная функция стирания блока памяти, принимает в качестве аргументов адрес стираемого блока и его размер. Ее блок-схема представлена на рис. 2.

В соответствии с заданным в качестве аргумента размером блока, выбирается соответствующая команда для модуля памяти. Если передаваемый размер не соответствует ни одному из типовых размеров, то используется стандартный размер стирания блока для модуля NOR флеш-памяти, равный 64 килобайтам. Передаваемый адрес разделяется на 3 отдельных байта, которые записываются в соответствующие поля отправляемой команды.

Интерфейс передачи пользовательских команд

При проведении исследований механизма идентификации возникает необходимость постранично перезаписывать участки памяти определенного размера по указанному адресу и проверять статус выполнения операций. Для этого необходимо использовать соответствующие команды, имеющие различный формат в зависимости от их назначения и от объема чипа флеш-памяти.

В существующем ПО возможность отправки произвольных команд отсутствует, как и возможность постраничной записи информации произвольного объема в память по заданному адресу. Реализована лишь запись всего чипа памяти, начиная с нулевого адреса. Для отправки всех команд в исходной библиотеке ch341a используется функция потоковой передачи данных, в качестве аргументов принимающая массив выходных и входных данных, а также размер данных массивов, которые должны совпадать. Отправка данных на устройство производится USB пакетами длиной 32 байта.

Расширенный функционал, реализованный в авторской библиотеке, обеспечивает различные сценарии ввода команд, а именно:

1. Отправка команды на устройство путем ее непосредственного ввода, включая необходимые параметры.
2. Отправка команды на устройство из внешнего файла.
3. Вывод результата исполнения команды на экран.
4. Вывод результата исполнения команды в файл.

Данный интерфейс позволил осуществить проверку возможностей устройства USB-SPI на непрерывную отправку серии команд для чтения всего чипа, используя встроенный в модуль памяти механизм инкрементации адреса. Испытание проведено на 2 МБ модуле памяти W25P16 производства компании Winbond при использовании USB пакетов длины 32 байта. Получение полных данных на ПК было выполнено за 21 секунду, что соответствует скорости передачи данных 779 кбит/с. Данный эксперимент подтвердил реальную возможность использования программатора CH341a для потоковой передачи байт на SPI устройства без прерываний — ранее в существующем программном обеспечении для модуля CH341a данный функционал реализован не был.

Сравнительный анализ быстродействия программного обеспечения для управления USB-SPI устройствами на основе чипа Ch341a

На рис. 3 приведены графики скорости выполнения операций чтения, записи и стирания при использовании авторской программы для чипов памяти Winbond W25P16 емкостью 2 Мбайта и Spansion S25FL032 емкостью 4 Мбайта в сравнении с ПО «P1119 CH341a Programmer» под ОС MS Windows и свободным проектом Flashrom. На графиках приведены средние значения скорости по десяти экспериментам.

Из графиков видно, что скорость выполнения операции чтения примерно одинакова для всех трех программ с незначительным превышением для авторской программы. Скорости записи и стирания у авторской программы и ПО P1119 для ОС MS Windows также примерно одинаковы с незначительным превышением у P1119. Программа Flashrom обеспечивает очень низкую скорость стирания в сравнении другими программами, что объясняется тем, что она производит ис-

ключительно поблочное стирание с контролем ошибок и последовательным перебором адресов, а разработанное авторами ПО использует потоковый режим стирания. Необходимо отметить, что программное обеспечение P1119 не позволило использовать потоковый режим стирания для чипа S25FL032 и соответствующие результаты приведены для поблочного стирания блоками размера 64 кбайт без контроля ошибок.

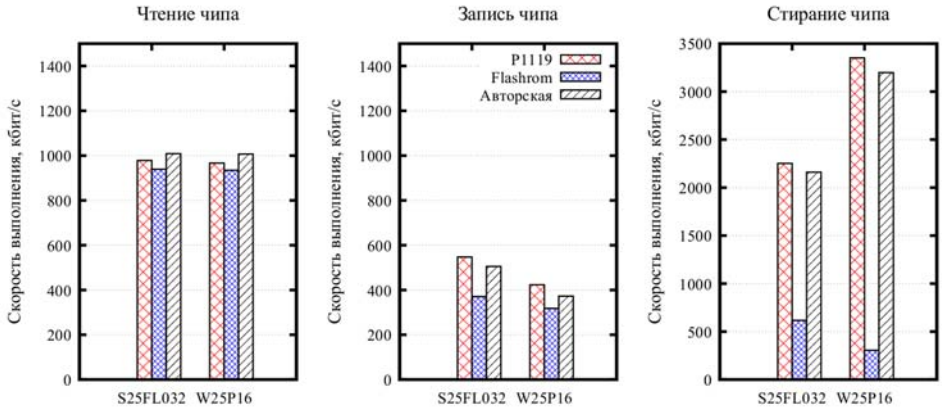

Рис. 3. Графики скорости выполнения операций чтения, записи и стирания

Графический интерфейс пользователя

В качестве интерфейса для управления USB-SPI устройствами на чипе Ch341a было разработано приложение с графическим интерфейсом пользователя в концепции многодокументного интерфейса, для формирования которого был использован фреймворк Qt5. На рис. 4 приведен один из примеров работы графического интерфейса пользователя. Показана одновременная работа с двумя чипами памяти. В левом окне производится считывание содержимого чипа памяти, а в правом окне показана отправка команд пользователя и получение ответов от контроллера.

Возможность одновременной неблокируемой работы с несколькими окнами обеспечивается использованием механизма многопоточности — работа с каждым чипом производится в отдельном потоке, не оказывая влияния друг на друга.

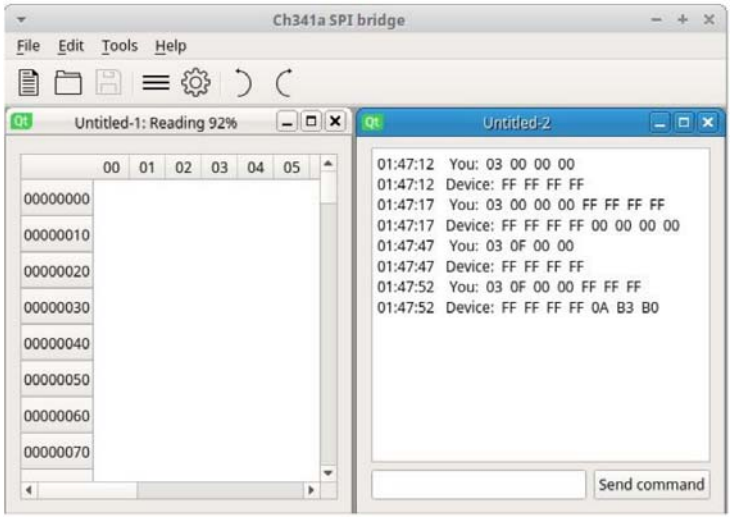

Рис. 4. Графический интерфейс пользователя

Заключение

В статье рассмотрена реализация разработки программного обеспечения для управления микросхемами памяти с SPI интерфейсом, подключаемыми к персональным ЭВМ посредством устройств USB-SPI на основе чипа CH341a компании WCH, для проведения исследований на пригодность микросхем флеш-памяти к идентификации сетевых устройств IoT. Рассмотрены общие вопросы взаимодействия с устройством, принцип одновременной работы с несколькими USB-SPI устройствами, имеющими в том числе одинаковый USB-идентификатор, а также реализован отсутствующий в существующем программном обеспечении для работы с устройствами CH341a функционал инициализации устройства на основе номера USB шины и номера подключенного к ней устройства, стирания блока памяти произвольного размера и отсутствующий интерфейс передачи пользовательских команд. Выполнено сравнение разработанного ПО с существующими решениями по скорости выполнения операций чтения, записи и стирания содержимого чипа флеш-памяти.

Литература

1. Wang Y., Yu W., Wu S., Malysa G., Suh G.E., Kan E.C. Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints // Proceedings of the 2012 IEEE Symposium on Security and Privacy. 2012. PP. 33–47. DOI: 10.1109/SP.2012.12.

2. Jia S., Xia L., Wang Z., Lin J., Zhang G., Ji Y. Extracting Robust Keys from NAND Flash Physical Unclonable Functions // Lecture Notes in Computer Science. 2015. Vol. 9290. pp. 437–454.

3. Владимиров С. С., Киричёк Р. В. Методика идентификации устройств интернета вещей на основе принудительной деградации участка флеш-памяти // Электросвязь. 2017. № 2. С. 32–35.

4. Vladimirov S., Kirichek R. The IoT Identification Procedure Based on the Degraded Flash Memory Sector // Lecture Notes in Computer Science. 2017. Vol. 10531. pp. 66–74.

5. Jakobsson M., Johansson K.-A. Unspoofable Device Identity Using NAND Flash Memory. URL: http://www.securityweek.com/unspoofable-device-identity-using-nand-flash-memory

References

1. Wang Y., Yu W., Wu S., Malysa G., Suh G.E., Kan E.C. Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints // Proceedings of the 2012 IEEE Symposium on Security and Privacy. 2012. PP. 33–47. DOI: 10.1109/SP.2012.12.

2. Jia S., Xia L., Wang Z., Lin J., Zhang G., Ji Y. Extracting Robust Keys from NAND Flash Physical Unclonable Functions // Lecture Notes in Computer Science. 2015. Vol. 9290. pp. 437–454.

3. Vladimirov S., Kirichek R. The IoT Devices Identification Procedure based on Forced Degrading of Flash-Memory Sector // Electrosvyaz'. 2017. No. 2. pp. 32–35.

4. Vladimirov S., Kirichek R. The IoT Identification Procedure Based on the Degraded Flash Memory Sector // Lecture Notes in Computer Science. 2017. Vol. 10531. pp. 66–74.

5. Jakobsson M., Johansson K.-A. Unspoofable Device Identity Using NAND Flash Memory. URL: http://www.securityweek.com/unspoofable-device-identity-using-nand-flash-memory

Владимиров Сергей Сергеевич – кандидат технических наук, доцент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, vladimirov.opds@gmail.com

Янковский Антон Константинович – магистрант, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, ostnipx@gmail.com

Vladimirov Sergey – Candidate of Engineering Sciences, Associate Professor, SPbSUT, St. Petersburg, 193232, Russian Federation, vladimirov.opds@gmail.com

Yankovskiy Anton – Undergraduate, SPbSUT, St. Petersburg, 193232, Russian Federation, ostnipx@gmail.com