

## КРАТКИЕ СООБЩЕНИЯ

УДК 004.422.8

### РАЗРАБОТКА СРЕДСТВА ДЛЯ ДИСТАНЦИОННОГО ОБНОВЛЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Н.И. ЛИСТОПАД, М.О. ПИКУЗА

*Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь*

*Поступила в редакцию 21 сентября 2018*

**Аннотация.** Рассмотрена возможность реализации средства дистанционного обновления программного обеспечения устройств под управлением микроконтроллеров, объединенных в общую сеть. Предъявлены требования к средству. Разработана структурная схема этого средства для микроконтроллеров ARM-архитектуры с использованием CAN-протокола.

**Ключевые слова:** загрузчик, программное обеспечение, микроконтроллер, ARM-архитектура, CAN-протокол.

**Abstract.** The possibility of implementing a remote software update mean for connected to a common network devices controlled by microcontrollers is considered. Requirements for the mean are presented. The structural scheme of this mean for ARM microcontrollers using the CAN-protocol is developed.

**Keywords:** bootloader, software, microcontroller, ARM-architecture, CAN-protocol.

**Doklady BGUIR. 2018, Vol. 118, No. 8, pp. 116-118**

**Development of the mean for remote software update**

**N.I. Listopad, M.O. Pikuza**

#### Введение

Современные радиоэлектронные средства (РЭС) не обходятся без управляющего устройства, в роли которого чаще всего выступает микроконтроллер (МК). МК в РЭС может выполнять различные задачи, такие как цифровая обработка сигналов (ЦОС), управление и контроль устройств, входящих в РЭС, обмен информацией как с устройствами, входящими в РЭС, так и с другими РЭС. РЭС может состоять из нескольких устройств, содержащих МК и объединенных в общую сеть. Эти устройства могут обмениваться информацией между собой по средствам какого-либо протокола. На основе общей сети возникает возможность дистанционной смены программного обеспечения (ПО) МК любого из устройств, подключенных к этой сети.

#### Описание загрузчика

Дистанционная смена ПО осуществляется при помощи загрузчика (бутлоадера). Загрузчик – это приложение, которое позволяет обновить ПО МК без использования специализированного оборудования, такого как JTAG-программатор. Загрузчик также позволяет прочитать и скопировать текущую версию прошивки, которая в данный момент записана в памяти МК. В памяти МК должно сосуществовать как минимум два образа программ: основная программа и загрузчик, в который должен быть включен код ветвления, производящий проверку, выполняется ли попытка обновления ПО. Обновление ПО может осуществляться по средствам различных протоколов, таких как USART, CAN, I2C, Ethernet, USB и многих других.

## Требования к загрузчику

Каждый загрузчик имеет свой собственный уникальный набор требований, основанный на типе приложения; однако есть несколько главных требований, которые являются общими для всех загрузчиков. Эти требования можно разбить на семь основных групп. Ниже представлены эти семь групп с примерами:

1. Возможность переключения или выбора режима работы (приложение или загрузчик).

Вход порта ввода-вывода, удерживаемый в течение 10 с, должен привести к тому, что система войдет в режим загрузчика. Байт EEPROM должен использоваться для переключения режимов. Сообщение загрузчика, принятое в течение 100 мс после запуска, должно привести к тому, что система войдет в режим загрузчика.

2. Требования к интерфейсу связи (USB, CAN, I2C, USART и т. д.).

Образы приложений должны передаваться системе через UART.

3. Требование к формату файла для записи (S-Record, hex, intel, toeff и т. д.).

Образ приложения должен быть отправлен в систему в формате S-Record.

4. Требования к Flash-памяти (удаление, запись, чтение, обращение к ячейкам памяти).

Загрузчик должен иметь возможность стирать только разделы Flash-памяти, содержащие приложение. Загрузчик должен записывать образы приложений во Flash-память. Загрузчик должен иметь возможность считывать с указанного адреса в Flash-памяти по команде. Загрузчик должен быть расположен в первых 4 Кб Flash-памяти.

5. Требования к EEPROM (разделение, стирание, чтение, запись).

Загрузчик должен по команде разбивать EEPROM. Загрузчик должен быть способен стереть EEPROM. Загрузчик должен быть способен считывать с указанного места в EEPROM. Загрузчик должен быть способен записывать данные в указанные места EEPROM.

6. Контрольная сумма приложения (проверка на отсутствие повреждения приложения).

Загрузчик должен вычислять контрольную сумму образа и сохранять ее в Flash-памяти. Загрузчик должен до выполнения приложения убедиться, что контрольная сумма приложения действительна.

7. Безопасность кода (защита загрузчика и приложения).

Загрузчик должен находиться в защищенной Flash-памяти для предотвращения его обновления или модификации.

## Процесс обновления ПО с помощью загрузчика

Опишем процесс обновления ПО с помощью загрузчика для системы из нескольких устройств с МК, объединенных в общую сеть. Для подключения к общей сети МК должен содержать соответствующий интерфейс. В качестве управляющего устройства возьмем МК ARM-архитектуры, так как в настоящее время они пользуются популярностью из-за небольшой стоимости, широкой периферии и высокой производительности. Общая сеть будет реализована на основе CAN-протокола, так как он является простым в реализации и имеет надежный контроль ошибок передачи и приема, что объясняет его частое использование в промышленных сетях и транспортных средствах. Структурная схема системы представлена на рис. 1. Все устройства системы (У1, У2, У3) подключены к общей CAN-шине. Подключение непосредственно МК к шине осуществляется с помощью входящего в состав устройства CAN-драйвера, который является приемопередатчиком и осуществляет переход от канального уровня протокола CAN к физическому уровню. К CAN-шине также подключено устройство, которое является мостом USB-CAN. Это устройство в режиме реального времени преобразует данные из одного интерфейса в другой, что позволяет подключить персональный компьютер (ПК) к общей сети.

Процесс смены ПО инициируется пользователем по средствам ПК. Вначале все устройства специальной командой переводятся в режим загрузчика. В этом режиме устройства ожидают последующих команд от ПК. После перевода в режим загрузчика пользователь выбирает конкретное устройство на шине, в котором необходимо сменить ПО, а также непосредственно само ПО. Далее происходит стирание секторов памяти МК устройства, в которых содержится предыдущая версия ПО. После стирания старого ПО происходит запись нового. Заключительной стадией обновления ПО является вывод специальной командой всех устройств из режима загрузчика. После этого устройства работают в штатном режиме.

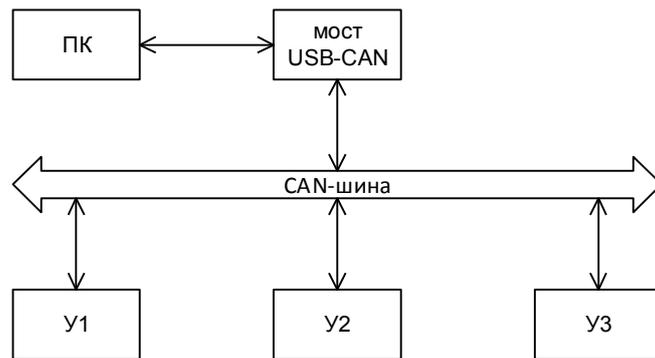


Рис. 1. Структурная схема системы

### Заключение

Разработана и реализована структурная схема средства дистанционного обновления ПО, которое значительно повышает эффективность процесса смены ПО в устройствах, подключенных к общей сети. Преимуществами данной схемы являются отсутствие необходимости в специализированном оборудовании и прямого доступа к МК устройства. Таким образом, для обновления ПО достаточно подключить ПК к системе с помощью кабеля. Загрузчик может использоваться для МК различных архитектур и производителей. Основным требованием к МК является возможность работы с внутренней FLASH-памятью и наличие интерфейса, позволяющего подключиться к общей сети.

### Список литературы

1. Bootloader Design for Microcontrollers in Embedded Systems [Электронный ресурс]. URL: [https://www.beningo.com/wp-content/uploads/images/Papers/bootloader\\_design\\_for\\_microcontrollers\\_in\\_embedded\\_systems%20.pdf](https://www.beningo.com/wp-content/uploads/images/Papers/bootloader_design_for_microcontrollers_in_embedded_systems%20.pdf). (дата обращения: 21.09.2018).

### References

1. Bootloader Design for Microcontrollers in Embedded Systems [Electronic resource]. URL: [https://www.beningo.com/wp-content/uploads/images/Papers/bootloader\\_design\\_for\\_microcontrollers\\_in\\_embedded\\_systems%20.pdf](https://www.beningo.com/wp-content/uploads/images/Papers/bootloader_design_for_microcontrollers_in_embedded_systems%20.pdf). (date of access: 21.09.2018).

### Сведения об авторах

Листопад Н.И., д.т.н, профессор, заведующий кафедрой информационных радиотехнологий Белорусского государственного университета информатики и радиоэлектроники.

Пикюза М.О., магистрант кафедры информационных радиотехнологий Белорусского государственного университета информатики и радиоэлектроники.

### Information about the authors

Listopad N.I., D.Sci., professor, head of information radioengineering department of Belarusian state university of informatics and radioelectronics.

Pikuza M.O., master student of information radioengineering department of Belarusian state university of informatics and radioelectronics.

### Адрес для корреспонденции

220013, Республика Беларусь,  
г. Минск, ул. П. Бровки, 6,  
Белорусский государственный университет  
информатики и радиоэлектроники  
тел. +375-33-650-31-78;  
e-mail: maksimpikuza@gmail.com  
Пикюза Максим Олегович

### Address for correspondence

220013, Republic of Belarus,  
Minsk, P. Brovka st., 6,  
Belarusian state university of  
informatics and radioelectronics  
tel. +375-33-650-31-78;  
e-mail: maksimpikuza@gmail.com  
Pikuza Maksim Olegovich