Единый формат хранения ПО модулей системы БЛОК

ВЕРСИЯ 1

Предпосылки

# Цели

1. Иметь единый унифицированный формат хранения и распространения ПО всех разновидностей модулей системы БЛОК. Этот формат должен быть достаточно гибким, чтобы быть пригодным как для ПО, помещаемого в различные микроконтроллеры, так и для ПО, представляющего собой программу для ОС.
2. В файл с ПО внедрить дополнительную информацию. Например, номер версии и указание, для какого модуля это ПО предназначено.
3. Объединить в одном файле ПО для всех полукомплектов и модификаций ячейки.
4. Обеспечить защиту целостности ПО от умышленных и неумышленных модификаций.\*

# Несколько программных модулей в одной ячейке

Для ясности всегда в дальнейшем используются эти понятия:

Ячейка – аппаратно обособленная часть комплекса БЛОК.

Модуль (Программный модуль) – логическая единица, выполняющая функционально обособленную часть алгоритма комплекса БЛОК.

В одной классической ячейке без ОС зачастую находится один программный модуль, поэтому понятия ячейки и модуля были тождественны. Но даже, когда это не так (например, в одном БС‑ДПС как минимум три функционально обособленных алгоритма: измеритель скорости, МПХ, БС‑АЛС) технически невозможно обновить только один модуль (например, МПХ), соответственно не имело смысла присваивать этому модулю номер версии и т.д.

С появлением ячеек под управлением ОС стало возможным на одном процессоре одновременно запустить несколько независимых программ. И обновлять их независимо. Отсюда появилась необходимость в разделении понятий ячейка и модуль.

Таким образом, ПО существует только для модуля, но не для ячейки.

# Файл ПО модуля

ПО модуля распространяется в виде SFP‑файла.

Один **SFP-файл это:**

* Одна версия
* Один модуль
* Все полукомплекты
* Все модификации ячейки

Один SFP‑файл содержит одну версию программы для одного модуля в вариантах для каждого полукомплекта и каждой модификации ячейки этого модуля.

Помимо файлов программы SFP‑файл содержит описание:

* Номер версии, подверсии и литерную метку
* Дату выпуска версии
* Текстовые комментарии разработчика[[1]](#footnote-1)\*
* Защитную хэш-сумму\*

# Загрузчик

В модуле находится загрузчик и программа. Роль загрузчика в том, чтобы обеспечить возможность обновления программы по линии CAN. Программа выполняет возложенные на неё алгоритмы БЛОК и имеет минимальные представления о загрузчике и процессе её обновления.

**Модуль**

**=**

**Загрузчик**

**+**

**Программа**

Если программа поддерживает переход в режим программирования, то начать обновление ПО можно не перезагружая ячейку. В противном случае потребуется перезагрузка, которая при старте на короткое время даст управление загрузчику.

# Программы для работы

* **FmPack** – создание SFP-файла
* **CanProg** – обновление по CAN
* **FmBurn** – первоначальная прошивка программатором

### Для разработчика: Выпуск версии при помощи FmPack

Разработчик формирует SFP-файл на этапе выпуска версии ПО модуля при помощи программы FmPack.

FmPack позволяет:

* Выбрать из каких файлов состоит программа, или в каком месте памяти она должна быть расположена;
* При необходимости задать разные программы для разных полукомплектов или разных модификаций ячейки;
* Присвоить версию и подверсию, добавить литерное обозначение и текстовый комментарий;
* Необходимо указать для модуля какой ячейки предназначено данное ПО.

### Для пусконаладчика: Обновление по CAN при помощи CanProg

После выбора SFP-файла программа проверяет его целостность по хэш-сумме и отображает, для какого модуля он предназначен и какую версию содержит.

При программировании CanProg:

* Находит все подключенные модули, для которых предназначен выбранный SFP‑файл, и переводит их в режим программирования по очереди.
* Проверяет версию установленного в модуле ПО и обновляет ПО при несовпадении версий.
* Контролирует целостность записанного в модуль ПО и применяет изменения.

Если модуль поддерживает безопасное обновление, то при перебое питания в процессе обновления в модуле остаётся предыдущее ПО. В противном случае гарантированно остаётся возможность обновления по CAN, а повреждённое ПО в модуле не загружается.

### Для завода: Первоначальная прошивка с помощью FmBurn

Первоначальная прошивка подразумевает программирование пустой ячейки. На этом этапе устанавливается загрузчик и, опционально, некоторая версия основной программы. Важным этапом первоначальной прошивки является конфигурирование загрузчика. В процессе конфигурирования заполняется информация, необходимая для дальнейшей идентификации ячейки: её тип, модификация, серийный номер и дата изготовления. Изменить эту информацию возможно только во время конфигурирования загрузчика.

Файл создается разработчиком

Что внутри файла?

Компоненты и цели

Структура файлов для устройств с ФС

Структура файлов для устройство без ОС (папочки e/, f/, имя файла = адрес начала), картинка с примером (можно сделать скриншот дерева в проводнике)

1. \* Не реализовано [↑](#footnote-ref-1)