

ЗАГРУЗЧИК U-BOOT ДЛЯ 1892ВМ14Я. РУКОВОДСТВО ПРОГРАММИСТА

**Версия
11.03.2022**

ОГЛАВЛЕНИЕ

1	О документе	3
2	Назначение и основные особенности загрузчика	4
3	Сборка загрузчика	6
3.1	Исходные коды	6
3.2	Конфигурация	7
3.3	Сборка	7
3.4	Состав образа	8
4	Запуск загрузчика	9
4.1	Обзор загрузки	9
4.2	Инициализация U-Boot SPL	10
4.3	Карта распределения ОЗУ	11
4.4	Переменные окружения	11
4.5	Команды	15
5	Загрузка целевого приложения	18
5.1	Схемы загрузки Legacy и Distro	18
5.2	Загрузка ELF-файла из SPI флеш-памяти	19
5.3	Загрузка Linux с SD/MMC/USB-носителя	19
5.4	Загрузка Linux с NAND-носителя	19
5.5	Загрузка Linux по TFTP и NFS	20
5.6	Передача управления из U-Boot в baremetal-приложение	21

1. О ДОКУМЕНТЕ

Данный документ описывает особенности работы загрузчика U-Boot 2019.01.0.9 для следующих модулей на базе СнК 1892ВМ14Я (далее МСом-02):

- Салют-ЭЛ24Д1 r1.3;
- Салют-ЭЛ24Д1 r1.4;
- Салют-ЭЛ24Д1 r1.5;
- Салют-ЭЛ24Д1 r1.5 с установленным навигационным радиомодулем RF2Chan v2;
- Салют-ЭЛ24Д2 r1.1;
- Салют-ЭЛ24ОМ1 r1.1 с установленным Салют-ЭЛ24ПМ1 r1.1 или Салют-ЭЛ24ПМ1 r1.2;
- Салют-ЭЛ24ОМ1 r1.2 с установленным Салют-ЭЛ24ПМ1 r1.2, Салют-ЭЛ24ПМ2 r1.0 или Салют-ЭЛ24ПМ2 r1.1.

2. НАЗНАЧЕНИЕ И ОСНОВНЫЕ ОСОБЕННОСТИ ЗАГРУЗЧИКА

Основное назначение загрузчика:

- начальная инициализация аппаратуры;
- загрузка Device Tree Blob (DTB) из SPI флеш-памяти или с SD/MMC/USB/NAND-носителя¹;
- загрузка образа Linux с SD/MMC/USB/NAND-носителя;
- загрузка образа Linux по TFTP;
- загрузка baremetal приложений из SPI флеш-памяти или с SD/MMC/USB/NAND-носителя;

Основные особенности загрузчика:

- поддержка схем загрузки Linux Distro и Legacy;
- передача параметров запуска Linux;
- инициализация контроллеров памяти DDR;
- загрузка и редактирование DTB;
- поддержка переменных окружения;
- поддержка монитора U-Boot по терминалу UART;
- поддержка сторожевого таймера;
- поддержка программной перезагрузки;
- поддержка GPIO;
- поддержка I2C;
- поддержка USB на встроенном контроллере USBIC в режиме Host/Device²;
- поддержка MMC;
- поддержка SPI флеш-памяти;
- поддержка NAND флеш-памяти³;
- поддержка Ethernet;
- поддержка файловых систем FAT, ext2, ext4 (только чтение), UBIFS;

¹ Загрузчик не поддерживает USB на модулях Салют-ЭЛ24ПМ1.

² Для работы USB в режиме Device необходимо, чтобы USB на плате был разведён в режиме OTG/Device. На модулях Салют-ЭЛ24ОМ1, Салют-ЭЛ24Д1 и Салют-ЭЛ24Д2 USB разведён только в режиме Host.

³ Термином NAND флеш-память в U-Boot обозначается флеш-память параллельного NAND, подключенного к контроллеру NFC.

- поддержка заводских настроек;
- вывод статуса последнего сброса микросхемы.

Загрузчик реализует обходы для следующих ограничений согласно документу «Микросхема интегральная 1892ВМ14Я. Перечень выявленных ограничений»:

- #867;
- #971;
- #972;
- #1160;
- #1969;
- #3346.

3. СБОРКА ЗАГРУЗЧИКА

3.1 Исходные коды

Исходные коды основаны на U-Boot 2019.01.

Имена файлов и директорий указаны относительно корневой директории исходных кодов загрузчика.

Описание общей структуры исходных кодов доступно в файле README.

Список файлов для поддержки модулей на базе MCom-02:

- Файлы начальной инициализации:
 - arch/arm/cpu/armv7/mcom/*.c
 - arch/arm/cpu/armv7/mcom/*.S
 - arch/arm/include/asm/arch-mcom/*.h
 - board/elvees/common/*.c
 - board/elvees/salute/*.c
 - board/elvees/salute-pm/*.c
- Файлы Device Tree Source (DTS):
 - arch/arm/dts/mcom*.dts
 - arch/arm/dts/mcom*.dtsi
- Файлы конфигурации Kconfig:
 - arch/arm/cpu/armv7/mcom/Kconfig
 - board/elvees/Kconfig
 - board/elvees/salute/Kconfig
 - board/elvees/salute-pm/Kconfig
 - configs/saluted1_defconfig
 - configs/saluted2_defconfig
 - configs/salutepm_defconfig
- Файлы конфигурации для поддерживаемых модулей:
 - include/configs/mcom.h

3.2 Конфигурация

Подробное описание параметров конфигурации загрузчика содержится в файле README.

Дополнительные параметры конфигурации для модулей на базе MCom-02:

- **DDR_CALIBRATION**
Включение режима калибровки памяти DDR.
Значение по умолчанию: не задано.
- **DDR_CALIBRATION_DDRMC_ID**
Номер контроллера DDR для режима калибровки.
Значение по умолчанию: 0.
- **HW_WATCHDOG**
Включение аппаратного сторожевого таймера.
Значение по умолчанию: не задано.
- **BOOT_ELF_FROM_SPI**
Включение загрузки ELF-файла из SPI флеш-памяти.
Значение по умолчанию: не задано.

3.3 Сборка

Результатом сборки исходных кодов загрузчика является образ `u-boot.mcom`, предназначенный для прошивки SPI флеш-памяти модуля или записи на SD/MMC-карту. Образ доступен в корневой директории загрузчика после завершения сборки.

Для сборки загрузчика на ПЭВМ должно быть установлено следующее программное обеспечение:

- `arm-linux-gnueabi toolchain` для кросс-компиляции;

Переменные окружения, влияющие на сборку загрузчика:

- `ARCH` — целевая архитектура;
- `CROSS_COMPILE` — префикс кросс-компилятора;
- `DEVICE_TREE` — имя файла DTS (без расширения) для целевого модуля;
- `PATH` — пути для поиска используемых приложений.

Соответствие модулей, конфигураций и имён DTS файлов приведено в таблице 3.3.

Таблица 3.1. Конфигурации и DTS файлы для модулей Салют-ЭЛ24

Модуль	Конфигурация	DTS
Салют-ЭЛ24Д1 r1.3	saluted1_defconfig	mcom02-saluteel24d1-r1.3
Салют-ЭЛ24Д1 r1.4	saluted1_defconfig	mcom02-saluteel24d1-r1.4
Салют-ЭЛ24Д1 r1.5	saluted1_defconfig	mcom02-saluteel24d1-r1.5
Салют-ЭЛ24Д1 r1.5 с RF2Chan	saluted1_defconfig	mcom02-saluteel24d1-r1.5-rf2chan
Салют-ЭЛ24Д2 r1.1	saluted2_defconfig	mcom02-saluteel24d2-r1.1
Салют-ЭЛ24ПМ1 с OM1	salutepm_defconfig	mcom02-saluteel24pm1-r1.1-1.2-om1-r1.1-1.2
Салют-ЭЛ24ПМ2 с OM1	salutepm_defconfig	mcom02-saluteel24pm2-r1.0-1.1-om1-r1.2

Пример сборки загрузчика для модуля Салют-ЭЛ24Д1 r1.3:

```
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabi-
export DEVICE_TREE=mcom02-salute-el24d1-r1.3
make saluted1_defconfig
make
```

3.4 Состав образа

Образ загрузчика `u-boot.mcom` состоит из образа U-Boot, прикрепленного к образу U-Boot SPL. Образ U-Boot включает DTB для настройки устройств и передачи в Linux.

Образы U-Boot SPL и U-Boot имеют формат uImage и создаются приложением `tools/mkimage`, входящим в состав загрузчика.

Для обхода ограничения #867 в заголовке образа U-Boot SPL значению поля `Target Operating System` присваивается значение U-Boot.

Для обхода ограничения #3346 размер образа U-Boot SPL выравнивается до ближайшего четного значения.

Схема разбиения образа `u-boot.mcom` представлена в таблице 3.4.

Таблица 3.2. Схема разбиения образа загрузчика

Область	Смещение (КБ)	Максимальный размер (КБ)
Образ U-Boot SPL	0	56
Переменные окружения ⁴	64	64
Образ U-Boot	128	

⁴ Размер и смещение области переменных окружения выровнены на границу сектора SPI флеш-памяти, установленной на модулях на базе MCom-02. При сохранении переменных окружения загрузчик стирает соответствующую область SPI флеш-памяти.

4. ЗАПУСК ЗАГРУЗЧИКА

4.1 Обзор загрузки

При включении питания модуля выполняется:

1. Первичный загрузчик BootROM копирует U-Boot Secondary Program Loader (U-Boot SPL) из SPI флеш-памяти или с SD/MMC-карты во внутреннюю память RAM и передает ему управление. Источник копирования определяется значением регистра BOOT контроллера SMCTR СнК 1892ВМ14Я.
2. U-Boot SPL копирует основной загрузчик (U-Boot) из SPI флеш-памяти или с SD/MMC-карты в память DDR и передает ему управление. Источник копирования определяется значением регистра BOOT контроллера SMCTR СнК 1892ВМ14Я. Подробнее см *Инициализация U-Boot SPL*.
3. U-Boot выполняет:
 1. Чтение заводских настроек в соответствии с документом «Спецификация заводских настроек модулей на базе 1892ВМ14Я». Подробнее см. описание переменных окружения *factory_eth_mac* и *factory_serial*.
 2. Выключение контроллера DDR, заданного переменной *ddrctl_cid*, если значение переменной *ddrctl_cmd* установлено в *disable*. По умолчанию выключается контроллер DDR1 для модулей Салют-ЭЛ24Д1 и Салют-ЭЛ24Д2.
 3. Исполнение сценария, заданного переменной окружения *bootcmd*. Выполнение сценария может быть прервано отсылкой любого символа в терминал UART модуля. В этом случае запускается монитор U-Boot для выполнения команд загрузчика (подробнее см. *Команды*).

В зависимости от значения переменной окружения *bootcmd* возможны сценарии работы (подробнее см. *Загрузка целевого приложения*):

- загрузка Linux с SDMMC0-носителя (установлен по умолчанию);
- загрузка Linux с SDMMC1-носителя;
- загрузка Linux с USB-носителя;
- загрузка Linux с NAND-носителя;
- загрузка Linux по TFTP и NFS;
- загрузка ELF-файла из SPI флеш-памяти.

Для загрузки Linux с SD/MMC/USB/NAND-носителя носитель должен содержать загрузочный раздел, соответствующий требованиям:

- быть основным (первичным);
- содержать в корневой директории файл образа Linux (см. *bootfile*).

4.2 Инициализация U-Boot SPL

При инициализации U-Boot SPL выполняется:

- включение L1-кэша инструкций для CPU0, CPU1 (L1-кэш данных включается для CPU0 в основном загрузчике);
- минимальная начальная настройка аппаратуры (см. функцию `board_init_f()` в файле `arch/arm/cpu/armv7/mcom/board.c`):
 - выключение ядра CPU1 (подробнее см. *Передача управления из U-Boot в baremetal-приложение*);
 - выключение DDR retention (ограничение #1160);
 - запись адреса функции холодного сброса BootROM в регистр ALWAYS_MISC0;
 - включение сторожевого таймера, если задано конфигурацией;
 - включение отображения загрузочных областей по умолчанию (ограничение #971);
 - включение конфигурации по умолчанию для коммутатора Accelerator Coherency Port (ограничение #972);
 - настройка APLL, CPLL и SPLL;
 - настройка терминала UART0;
 - включение питания модулей памяти DDR0 и DDR1 (для модулей Салют-ЭЛ24ПМ);
 - настройка контроллеров памяти DDR0 и DDR1;
 - настройка контроллеров SDMMC0 и SDMMC1 (включение резистивных подтяжек и т.д.);
- загрузка U-Boot (см. функцию `board_init_r()` в файле `common/spl/spl.c`):
 - загрузка драйвера GPIO;
 - загрузка драйвера SPI для контроллера SPI0;
 - загрузка драйвера SPI флеш-памяти;
 - загрузка драйвера контроллера SDMMC;
 - загрузка драйвера контроллера USBIC;
 - копирование из SPI флеш-памяти или с SD/MMC-карты в память DDR основного загрузчика;
 - передача управления основному загрузчику.

4.3 Карта распределения ОЗУ

После запуска загрузчик располагается в верхней области памяти, подключенной к контроллеру DDR0. Свободные области DDR0/DDR1 могут использоваться для загрузки Linux и автономных приложений.

Расположение загрузчика в памяти для поддерживаемых модулей на базе MCom-02 показано в таблице 4.3.

Таблица 4.1. Карта распределения ОЗУ

Область	Начальный адрес	Конечный адрес
Свободная память	0x40000000	0x7EFFFFFF
U-Boot ⁵	0x7F000000	0x7FFFFFFF
Свободная память	0xA0000000	0xDFFFFFFF

Подробную информацию об использовании памяти загрузчиком можно получить с помощью команды монитора `bdinfo`.

4.4 Переменные окружения

Загрузчик поддерживает возможность настройки через переменные окружения.

Во время запуска загрузчик выполняет поиск переменных окружения, сохраненных в SPI флеш-памяти:

- при успешном обнаружении устанавливаются переменные окружения, сохраненные в SPI флеш-памяти;
- в случае ошибки устанавливаются переменные окружения, заданные по умолчанию.

Переменные окружения загрузчика, заданные по умолчанию (см. файл `include/configs/mcom.h`):

bootfile

Имя файла, содержащего образ Linux для загрузки.

Значение по умолчанию: `zImage`.

loadaddr

Адрес буфера для копирования образа Linux с SD/MMC/USB/NAND-носителя.

Значение по умолчанию: `0x40000000`.

bootm_low

Нижняя граница области памяти, используемой для загрузки Linux.

Значение по умолчанию: не задано.

bootm_size

Размер области памяти, используемой для загрузки Linux.

⁵ Начальный адрес области памяти, используемой загрузчиком, указан для справки и изменяется в зависимости от конфигурации загрузчика.

Значение по умолчанию: 0x10000000.

stdin, stdout, stderr

Имена стандартных устройств ввода/вывода, используемых загрузчиком.

Значение по умолчанию: `serial`.

ddrctl_cmd

Команда управления контроллерами DDR.

Значение по умолчанию: `disable`.

ddrctl_cid

Номер контроллера DDR для команды `ddrctl_cmd`.

Значение по умолчанию: 1.

bootcmd

Последовательность действий, выполняемых при команде `boot`.

Значение по умолчанию соответствует последовательности действий, описанной ниже.

Если загрузчик собран с включенной опцией `BOOT_ELF_FROM_SPI`, то выполняется:

1. Копирование ELF-файла приложения из SPI флеш-памяти в ОЗУ. Источник размещения в SPI флеш-памяти определяется переменными окружения `bootelf_elfsize`, `bootelf_spibus`, `bootelf_spioffset`. Адрес в ОЗУ определяется переменной `bootelf_addr`.
2. Передача управления в скопированный ELF-файл.

Если загрузчик собран с выключенной опцией `BOOT_ELF_FROM_SPI`, то выполняется последовательность команд, указанная в переменной `boot_targets`. Если в названии команды присутствует префикс `legacy`, будет использована схема загрузки Legacy, в противном случае — Distro. Если команда завершилась не успешно, загрузчик переходит к выполнению следующей команды из списка `boot_targets`. Например, при `boot_targets="legacy_mmc0 mmc0"` будет выполнено:

- команда загрузки Linux с SDMMC0 с использованием схемы Legacy;
- если `legacy_mmc0` завершится не успешно, команда загрузки Linux с SDMMC0 с использованием схемы Distro.

Последовательность действий, выполняемая при загрузке Linux с использованием схемы Legacy:

1. Выполнение команд, заданных переменной `bootenvcmd`.
2. Чтение образа Linux с загрузочного раздела SD/MMC/USB/NAND-носителя в память по адресу, заданному переменной `loadaddr`. Устройство и файл, содержащие образ Linux, номер загрузочного раздела задаются переменными `boot_targets`, `bootfile`, `bootpartnum`.
3. Запуск образа Linux по адресу, заданному переменной `loadaddr`. При запуске загрузчик:

- устанавливает параметры загрузки Linux, заданные переменной `set_bootargs`;
- копирует и модифицирует DTB (подробнее см. *Состав образа*), передает Linux адрес модифицированного DTB.

Последовательность действий, выполняемая при загрузке Linux с использованием схемы Distro:

1. Поиск файла `extlinux/extlinux.conf` на загрузочном разделе SD/MMC/USB/NAND-носителя.
2. Чтение и обработка опций, заданных в `extlinux/extlinux.conf`.
3. Загрузка образа Linux по адресу, указанному в переменной `kernel_addr_r`.
4. Запуск образа Linux с помощью команды `sysboot`.

bootenvcmd

Дополнительная команда, выполняемая перед загрузкой Linux.

Значение по умолчанию: не задано.

console

Имя и параметры устройства Linux, используемого в качестве консоли.

Значение по умолчанию: `ttyS0,115200`.

cmdline

Дополнительные параметры загрузки Linux.

Значение по умолчанию: не задано.

boot_targets

Последовательность команд загрузки Linux (подробнее см. *bootcmd*).

Значение по умолчанию: `mmc0, legacy_mmc0, mmc1, legacy_mmc1, usb0, legacy_usb0, ubifs0, legacy_ubifs0`.

bootpartnum

Номер раздела SD/MMC/USB/NAND-носителя, используемого для загрузки Linux и файла переменных

окружения. Значение по умолчанию: 1.

rootpartnum

Номер раздела SD/MMC/USB/NAND-носителя, содержащего корневую файловую систему.

Значение по умолчанию: 2.

set_bootargs

Параметры загрузки Linux для режима загрузки с SD/MMC/USB/NAND-носителя.

Значение по умолчанию: `setenv bootargs console=${console} root=${rootfsdev} rootfstype=${rootfstype} rw rootwait ${cmdline}`.

rootfstype

Тип корневой файловой системы Linux для режима загрузки с SD/MMC/USB/NAND-носителя.

Значение по умолчанию: `ext4`.

bootelf_addr

Адрес памяти ОЗУ для загрузки ELF-файла.

Значение по умолчанию: `0x50000000`.

bootelf_elfsize

Размер ELF-файла в байтах.

Значение по умолчанию: `0x200000`.

bootelf_spibus

Номер шины SPI к которой подключен чип SPI флеш-памяти.

Значение по умолчанию: `0`.

bootelf_spioffset

Смещение ELF-файла в SPI флеш-памяти в байтах.

Значение по умолчанию: `0x100000`.

ethaddr

MAC-адрес контроллера Ethernet. При наличии переменной, U-Boot использует указанный адрес для инициализации контроллера Ethernet и добавляет свойство `local-mac-address` в узел контроллера внутри Device Tree. Поиск узла осуществляется посредством псевдонима `ethernet` узла `aliases`. Таким образом, U-Boot и Linux используют один MAC-адрес во время работы. При отсутствии переменной U-Boot и Linux генерируют произвольные MAC-адреса независимо.

При загрузке модуля выполняется чтение заводских настроек и анализ наличия переменной `factory_eth_mac`. Если переменная `factory_eth_mac` существует, а `ethaddr` не существует, то `ethaddr` создаётся и заполняется значением `factory_eth_mac`. В противном случае `ethaddr` не изменяется.

factoryoffset

Смещение в байтах последнего сектора SPI флеш-памяти, где хранятся заводские настройки. Переменная создаётся динамически и не должна изменяться пользователем.

factorysize

Размер в байтах сектора SPI флеш-памяти, где хранятся заводские настройки. Переменная создаётся динамически и не должна изменяться пользователем.

factory_eth_mac

MAC-адрес Ethernet из заводских настроек. Переменная создаётся динамически и не должна изменяться пользователем. Переменная не создаётся в случае отсутствия заводских настроек. См. также `ethaddr`.

factory_serial

Серийный номер модуля из заводских настроек. Переменная создаётся динамиче-

ски и не должна изменяться пользователем. Переменная не создаётся в случае отсутствия заводских настроек.

Значение переменной записывается в свойство `serial-number` в DTB (свойство доступно в Linux).

kernel_addr_r

Адрес буфера для копирования образа Linux с SD/MMC/USB/NAND-носителя. Используется при загрузке с использованием схемы загрузки Distro.

Значение по умолчанию: `0x40000000`.

fdt_addr_r

Адрес буфера для копирования DTB-файла с SD/MMC/USB/NAND-носителя. Используется при загрузке с использованием схемы загрузки Distro, если в `extlinux/extlinux.conf` заданы опции `fdtdir` или `fdt`.

Значение по умолчанию: `0x48000000`.

fdt_addr

Адрес буфера, содержащего DTB-файл. Используется при загрузке с использованием схемы загрузки Distro, если в `extlinux/extlinux.conf` не заданы опции `fdtdir` или `fdt`.

Значение по умолчанию: `0x41000000`.

scriptaddr

Адрес буфера для копирования и обработки файла `extlinux/extlinux.conf`. Используется при загрузке с использованием схемы загрузки Distro.

Значение по умолчанию: `0x49000000`.

ramdisk_addr_r

Адрес буфера для копирования ramdisk с SD/MMC/USB/NAND-носителя. Используется при загрузке с использованием схемы загрузки Distro.

Значение по умолчанию: `0x50000000`.

В мониторе U-Boot доступны функции управления переменными окружения.

Для возврата переменной окружения значения по умолчанию — в мониторе U-Boot выполнить команды:

```
env default <var_name>
saveenv
```

Описание некоторых переменных окружения содержится в файле README.

4.5 Команды

Некоторые команды, поддерживаемые монитором загрузчика:

- `base` — установка смещения для команд обращения к памяти;
- `bdinfo` — печать информации о модуле;

- `bootd` — выполнение команды загрузки по умолчанию;
- `bootelf` — загрузка образа ELF из памяти;
- `bootm` — загрузка образа приложения из памяти;
- `bootz` — загрузка образа zImage из памяти;
- `cmp` — сравнение содержимого памяти;
- `coninfo` — печать информации о консольных устройствах;
- `cp` — копирование содержимого памяти;
- `crc32` — вычисление контрольной суммы;
- `dm` — печать информации о драйверах устройств;
- `echo` — печать аргументов;
- `editenv` — редактирование переменных окружения;
- `env` — управление переменными окружения;
- `fdt` — управление Flattened Device Tree (FDT);
- `go` — запуск приложения по указанному адресу;
- `help` — печать справки и полного списка команд монитора;
- `iminfo` — печать информации об образе приложения;
- `load` — загрузка файла из файловой системы;
- `loadb` — загрузка файла через терминал по протоколу Kermit;
- `loads` — загрузка файла в формате S-Record через терминал;
- `loadx` — загрузка файла через терминал по протоколу XMODEM;
- `loady` — загрузка файла через терминал по протоколу YMODEM;
- `loop` — бесконечный цикл по диапазону адресов;
- `md` — отображение содержимого памяти;
- `meminfo` — отображение информации о памяти;
- `mm` — изменение содержимого памяти с автоматическим увеличением адреса;
- `mmc` — функции для работы с подсистемой MMC;
- `mmcinfo` — отображение информации о MMC;
- `mw` — заполнение памяти;
- `mw` — изменение содержимого памяти по постоянному адресу;
- `printenv` — печать переменных окружения;
- `reset` — выполнение «теплого» сброса микросхемы;
- `run` — выполнение команд из указанной переменной окружения;

- `save` — сохранение файла в файловой системе;
- `saveenv` — сохранение переменных окружения;
- `setenv` — установка переменных окружения;
- `sf` — функции для работы с подсистемой SPI флеш-памяти;
- `version` — печать версий монитора, компилятора и компоновщика.

Дополнительные команды монитора для модулей на базе MCom-02:

- `ddrctl disable <0|1>`

Выключение контроллеров памяти DDR. Команда отключает тактовую частоту для указанного контроллера.

Полный список команд доступен по команде монитора `help`.

5. ЗАГРУЗКА ЦЕЛЕВОГО ПРИЛОЖЕНИЯ

5.1 Схемы загрузки Legacy и Distro

Загрузчик поддерживает загрузку Linux с SD/MMC/USB/NAND-носителя с использованием схем:

- Distro — схема загрузки с использованием файла `extlinux/extlinux.conf`;
- Legacy — схема загрузки без использования файла `extlinux/extlinux.conf`.

Преимущества использования схемы Distro описаны в файле [README.distro](#)⁶.

Формат файла `extlinux/extlinux.conf` описан на [wiki.syslinux.org](#)⁷.

Последовательность действий при загрузке Linux описана в `bootcmd`, список команд загрузки — в `boot_targets`.

Например, для загрузки с SDMMC1-носителя на модуле Салют-ЭЛ24ПМ с установленным Салют-ЭЛ24ОМ1 с использованием схемы Distro выполнить:

1. Создать файл `extlinux/extlinux.conf` на загрузочном разделе SD-карты. Содержимое файла:

```
menu title Linux Boot Options

default Buildroot
timeout 20

label Buildroot
    kernel ../zImage
    append root=/dev/mmcblk0p2 rw rootwait console=ttyS0,115200
    ↪video=HDMI:1920x1080
```

2. В мониторе U-Boot установить переменную `boot_targets` в значение `mmc1`:

```
setenv boot_targets mmc1
saveenv
```

Примечание: По умолчанию Buildroot 3.1 собирается в режиме Legacy. Дальнейшие инструкции по загрузке Linux приводятся для схемы загрузки Legacy, если не указано иное.

⁶ <https://gitlab.denx.de/u-boot/u-boot/blob/master/doc/README.distro>

⁷ <https://wiki.syslinux.org/wiki/index.php?title=Config>

5.2 Загрузка ELF-файла из SPI флеш-памяти

Для загрузки ELF-файла из SPI флеш-памяти необходимо собрать загрузчик с опцией `BOOT_ELF_FROM_SPI` (подробнее см. *bootcmd*).

5.3 Загрузка Linux с SD/MMC/USB-носителя

Для загрузки Linux с SD/MMC/USB-носителя необходимо выполнить:

1. Прошить образ SD-карты на SD/MMC/USB-носитель. Процедура описана в документе «Дистрибутив ОС GNU/Linux на базе Buildroot для 1892BM14Я. Руководство программиста».
2. Перезагрузить модуль и в мониторе U-Boot установить переменную окружения *boot_targets* в значение:
 - `legacy_usb0` для загрузки с USB-носителя;
 - `legacy_mmc0` для загрузки с SDMMC0-носителя;
 - `legacy_mmc1` для загрузки с SDMMC1-носителя.

Например, для загрузки с SDMMC1-носителя:

```
setenv boot_targets legacy_mmc1
saveenv
```

3. Перезагрузить модуль или выполнить команду загрузки в мониторе U-Boot:

```
boot
```

5.4 Загрузка Linux с NAND-носителя

Для загрузки Linux с NAND-носителя необходимо выполнить:

1. На модуле подготовить файловую систему на NAND-носителе. Нижеприведённые команды выполнять в терминале ОС Linux модуля:
 1. Определить устройство NAND-носителя:

```
mtddinfo -a
```

В выведенном списке выбрать устройство с полем `Type: nand`.

2. Подготовить разделы NAND и примонтировать их:

```
ubiformat /dev/<устройство NAND>
ubiattach -p /dev/<устройство NAND>
ubimkvol /dev/ubi0 -N boot -s 128MiB
ubimkvol /dev/ubi0 -N root -s 1024MiB
mkdir -p /mnt/boot
mkdir -p /mnt/root
```

(continues on next page)

(продолжение с предыдущей страницы)

```
mount -t ubifs ubi0:boot /mnt/boot
mount -t ubifs ubi0:root /mnt/root
```

3. Скопировать с ПЭВМ файл образа Linux zImage, расположенный в директории buildroot/output/images дистрибутива Buildroot, в директорию /mnt/boot.
4. Скопировать с ПЭВМ и распаковать архив rootfs.tar, расположенный в директории buildroot/output/images дистрибутива Buildroot, в директорию /mnt/root.
5. Отмонтировать разделы:

```
sync
umount /mnt/boot
umount /mnt/root
ubidetach -d 0
```

2. Перезагрузить модуль и установить переменную окружения *boot_targets* в значение *legacy_ubifs0*.
3. Выполнить команду загрузки в мониторе U-Boot:

```
boot
```

5.5 Загрузка Linux по TFTP и NFS

Для загрузки Linux по TFTP и корневой файловой системы по NFS со статическим назначением IP-адреса модулю в мониторе U-Boot необходимо выполнить:

1. Установить IP-адрес модуля:

```
setenv ipaddr <ip_address>
```

2. Установить IP-адрес TFTP-сервера:

```
setenv serverip <tftp_server_ip_address>
```

3. Установить параметры запуска Linux:

```
setenv bootargs console=ttyS0,115200 root=/dev/nfs rw nfsroot=<nfs_server_ip>
↪:/<nfs_root_directory>,vers=3 ip=${ipaddr}
```

4. Установить команду загрузки Linux:

```
setenv bootcmd 'tftpboot; bootz ${loadaddr} - ${fdtcontroladdr}'
```

5. Сохранить переменные окружения:

```
saveenv
```

6. Перезагрузить модуль или выполнить команду загрузки:

```
boot
```

Для загрузки Linux по TFTP и корневой файловой системы по NFS с динамическим назначением IP-адреса модулю в мониторе U-Boot необходимо выполнить:

1. Установить IP-адрес TFTP-сервера:

```
setenv serverip <tftp_server_ip_address>
```

2. Установить параметры запуска Linux:

```
setenv bootargs console=ttyS0,115200 root=/dev/nfs rw nfsroot=<nfs_server_ip>  
↪: /<nfs_root_directory>, vers=3 ip=dhcp
```

3. Установить команду загрузки Linux:

```
setenv bootcmd 'tftpboot; dhcp; bootz ${loadaddr} - ${fdtcontroladdr}'
```

4. Сохранить переменные окружения:

```
saveenv
```

5. Перезагрузить модуль или выполнить команду загрузки:

```
boot
```

5.6 Передача управления из U-Boot в baremetal-приложение

Варианты передачи управления в baremetal-приложение:

- с возвратом в U-Boot,
- без возврата в U-Boot.

Для передачи управления без возврата в U-Boot необходимо:

1. разработать исходный код приложения. Требования к исходному коду:
 - секции кода, стека и данных не должны использовать используемые загрузчиком U-Boot адреса (подробнее см. *Карта распределения ОЗУ*);
 - для включения CPU1 необходимо:
 - переключить отображение нулевых адресов CPU1 в BootROM (записать значение 0x3 по физическому адресу 0x38096004);
 - записать адрес начала кода, исполняемого CPU1, в регистр ALWAYS_MISC0;

- включить домен питания CPU1 (записать значение 4 по физическому адресу 0x38095000).
2. скомпилировать ELF-файл приложения;
 3. загрузить ELF-файл приложения с ПЭВМ в память ОЗУ модуля, например:
 - через интерфейс Ethernet:
 - загрузить ELF-файл на сервер TFTP на ПЭВМ;
 - загрузить ELF-файл приложения в память ОЗУ модуля командой `tftpboot`.
 - через интерфейс SD/eMMC/USB:
 - скопировать ELF-файл приложения в раздел Boot на SD/eMMC/USB-носителе;
 - загрузить ELF-файл приложения в память ОЗУ модуля. Например для модуля Салют-ЭЛ24ОМ1 с установленным модулем Салют-ЭЛ24ПМ1 и установленной SD-картой необходимо выполнить команду `fatload mmc 1 <addr> <file.elf>`, где `<addr>` — адрес ОЗУ для загрузки ELF-файла, `<file.elf>` — имя ELF-файла на SD-карте.
 - через интерфейс NAND:
 - скопировать ELF-файл приложения в раздел Boot NAND флеш-памяти;
 - загрузить ELF-файл приложения в память ОЗУ модуля, выполнив команду `ubifsload <addr> <file.elf>`, где `<addr>` — адрес ОЗУ для загрузки ELF-файла, `<file.elf>` — имя ELF-файла в разделе Boot NAND флеш-памяти.
 - через интерфейс SPI:
 - загрузить ELF-файл приложения в SPI флеш-память:
 - * загрузить ELF-файл приложения в память ОЗУ модуля через интерфейсы UART, SD/eMMC, Ethernet;
 - * выполнить запись командой `sf write`.
 - загрузить ELF-файл приложения в память ОЗУ модуля командой `sf read`.
 - через интерфейс UART:
 - выполнить команду `loady` на модуле;
 - передать ELF-файл приложения с помощью программы `minicom` с использованием протокола YMODEM⁸ на ПЭВМ.
 4. передать управление из U-Boot в приложение командой `bootelf <addr>`, где `<addr>` — адрес загруженного в памяти ELF-файла.

⁸ <https://en.wikipedia.org/wiki/YMODEM>

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

В

boot_targets, 12, 18–20
bootcmd, 9, 13, 18, 19
bootelf_addr, 12
bootelf_elfsize, 12
bootelf_spibus, 12
bootelf_spioffset, 12
bootenvcmd, 12
bootfile, 9, 12
bootpartnum, 12

D

ddrctl_cid, 9
ddrctl_cmd, 9, 12

E

ethaddr, 14

F

factory_eth_mac, 9, 14
factory_serial, 9

K

kernel_addr_r, 13

L

loadaddr, 12

S

set_bootargs, 13

□

переменная окружения

boot_targets, 12, 13, 18–20
bootcmd, 9, 12, 13, 18, 19
bootelf_addr, 12, 14
bootelf_elfsize, 12, 14
bootelf_spibus, 12, 14
bootelf_spioffset, 12, 14
bootenvcmd, 12, 13
bootfile, 9, 11, 12
bootm_low, 11
bootm_size, 11
bootpartnum, 12, 13
cmdline, 13

console, 13
ddrctl_cid, 9, 12
ddrctl_cmd, 9, 12
ethaddr, 14
factory_eth_mac, 9, 14
factory_serial, 9, 14
factoryoffset, 14
factorysize, 14
fdt_addr, 15
fdt_addr_r, 15
kernel_addr_r, 13, 15
loadaddr, 11, 12
ramdisk_addr_r, 15
rootfstype, 13
rootpartnum, 13
scriptaddr, 15
set_bootargs, 13
stdin, stdout, stderr, 12