

ЗАГРУЗЧИК U-BOOT ДЛЯ 1892ВМ14Я. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

**Версия v2.10
30.01.2019**

ОГЛАВЛЕНИЕ

1	О документе	3
2	Назначение и основные возможности загрузчика	4
3	Краткое описание исходных кодов загрузчика	6
4	Параметры конфигурации загрузчика	7
5	Особенности работы загрузчика	8
5.1	Переменные окружения загрузчика	9
5.2	Режим загрузки ELF-файла из SPI флеш-памяти	11
5.3	Режим загрузки Linux с SD/MMC/USB/NAND-носителя	11
5.4	Режим загрузки Linux по TFTP	13
5.5	Режим монитора	14
5.6	Передача управления из U-Boot в baremetal-приложение	15
6	Сборка загрузчика	17
7	Образ загрузчика	18
8	Расположение загрузчика в памяти	19

1. О ДОКУМЕНТЕ

Данный документ описывает особенности работы загрузчика U-Boot 2017.07.0.19 для следующих модулей на базе СнК 1892ВМ14Я (далее МСom-02):

- Салют-ЭЛ24Д1 r1.3;
- Салют-ЭЛ24Д1 r1.4;
- Салют-ЭЛ24Д1 r1.5;
- 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 ВОЗМОЖНОСТИ 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;

Загрузчик обеспечивает следующие основные возможности:

- передача параметров запуска Linux;
- инициализация контроллеров памяти DDR;
- загрузка и редактирование DTB;
- переменные окружения;
- терминал UART;
- режим монитора по терминалу UART;
- поддержка сторожевого таймера;
- поддержка GPIO;
- поддержка I2C;
- поддержка USB на встроенном контроллере USBIC в режиме Host/Device²;
- команды доступа к памяти;
- подсистема MMC;
- подсистема SPI флеш-памяти;
- подсистема NAND флеш-памяти³;
- поддержка Ethernet;
- поддержка файловых систем FAT, ext2, ext4 (только чтение), UBIFS.

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

- #867;

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.

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

3. КРАТКОЕ ОПИСАНИЕ ИСХОДНЫХ КОДОВ ЗАГРУЗЧИКА

Исходные коды основаны на U-Boot 2017.07⁴.

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

⁴ <http://git.denx.de/?p=u-boot.git;a=commit;h=v2017.07>

4. ПАРАМЕТРЫ КОНФИГУРАЦИИ ЗАГРУЗЧИКА

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.

5. ОСОБЕННОСТИ РАБОТЫ ЗАГРУЗЧИКА

Загрузка U-Boot происходит в два этапа:

- в режиме загрузки из SPI флеш-памяти или с SD/MMC-карты первичный загрузчик BootROM копирует U-Boot Secondary Program Loader (U-Boot SPL) во внутреннюю память RAM и передает ему управление;
- U-Boot SPL копирует основной загрузчик (U-Boot) из SPI флеш-памяти или с SD/MMC-карты в память DDR и передает ему управление.

Устройство для загрузки U-Boot SPL и U-Boot определяется значением регистра BOOT контроллера SMCTR СнК 1892ВМ14Я.

Основные функции U-Boot SPL:

- 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-ЭЛ24ПМ1);
 - настройка контроллеров памяти DDR0 и DDR1;
 - установка параметров `t_rfc_min` и `t_ras_min` для памяти DDR (ограничение #1969);
 - настройка контроллеров SDMMC0 и SDMMC1 (включение резисторных 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.

Параметры настройки аппаратуры для драйверов устройств определяются файлом DTS, соответствующим типу модуля (указывается при сборке).

Основной загрузчик обеспечивает следующие режимы работы:

- режим загрузки Linux с SD/MMC/USB/NAND-носителя;
- режим загрузки Linux по TFTP;
- режим монитора.

5.1 Переменные окружения загрузчика

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.

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`.

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.

bootsource

Источник загрузки Linux. Значение для SD/MMC-карты: `mmc`, для USB-носителя: `usb`, для NAND флеш-памяти: `nand`.

Значение по умолчанию: `mmc`.

mmcdev

Номер контроллера SD/MMC, используемого для загрузки Linux и файла переменных

окружения. Значение по умолчанию: `0`.

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.

В режиме монитора доступны функции управления переменными окружения.

Описание некоторых переменных окружения содержится в файле README.

5.2 Режим загрузки ELF-файла из SPI флеш-памяти

Для включения режима загрузки ELF-файла из SPI флеш-памяти необходимо собрать загрузчик с опцией CONFIG_BOOT_ELF_FROM_SPI.

В режиме загрузки ELF-файла из SPI флеш-памяти выполняется следующая последовательность действий:

1. ELF-файл приложения загружается из SPI флеш-памяти в ОЗУ. Адрес памяти для загрузки ELF-файла, смещение в SPI флеш-памяти, размер ELF-файла задаются с помощью переменных окружения (подробнее см. *Переменные окружения загрузчика*).
2. Приложение из ELF файла загружается на исполнение автоматически.

5.3 Режим загрузки Linux с SD/MMC/USB/NAND-носителя

Последовательность действий, выполняемых в режиме загрузки Linux с SD/MMC/USB/NAND-носителя, задана командой загрузки `bootcmd`, определенной в конфигурации загрузчика (см. файл `include/configs/mcom.h`):

- выключение контроллера DDR, заданного переменной `ddrctl_cid`, если значение переменной `ddrctl_cmd` установлено в `disable` (по умолчанию выключается контроллер DDR1);
- выполнение дополнительных команд, заданных переменной `bootenvcmd`;
- чтение образа Linux из файла, заданного переменной `bootfile`, в память по адресу, заданному переменной `loadaddr`;
- запуск образа Linux по адресу, заданному переменной `loadaddr`.

Загрузчик передает Linux соответствующий типу модуля DTB, который входит в состав образа U-Boot.

Загрузочный раздел должен соответствовать следующим требованиям:

- быть основным (первичным);
- иметь файловую систему FAT;
- содержать в корневой директории файл образа Linux.

Переменными окружения загрузчика задаются:

- источник загрузки Linux;
- номер контроллера SD/MMC для загрузки Linux;
- параметры запуска Linux.

5.3.1 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:

```
mtdinfo -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  
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-script/buildroot/output/images дистрибутива Buildroot, в директорию /mnt/boot.
4. Скопировать с ПЭВМ и распаковать архив rootfs.tar, расположенный в директории buildroot-script/buildroot/output/images дистрибутива Buildroot, в директорию /mnt/root.
5. Отмонтировать разделы:

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

2. В переменных окружения загрузчика установить источник загрузки из NAND (подробнее см. *bootsource*) — перезагрузить модуль и в режиме монитора U-Boot выполнить:

```
setenv bootsource nand
saveenv
```

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

```
boot
```

5.3.2 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 подробно описана в документе “Дистрибутив ОС GNU/Linux на базе Buildroot для 1892BM14Я. Руководство системного программиста”.
2. Выбрать источник загрузки. Для этого перезагрузить модуль и в режиме монитора U-Boot установить переменную окружения *bootsource* в значение *usb* для загрузки с USB-носителя или в *mmc* для загрузки с SD/MMC-носителя:

```
setenv bootsource usb/mmc
saveenv
```

3. Выбрать номер SD/MMC-контроллера для загрузки. Для этого установить переменную окружения *mmcdev* в значения 0 или 1:

```
setenv mmcdev 0/1
saveenv
```

На модулях Салют-ЭЛ24ОМ1 с установленным Салют-ЭЛ24ПМ1/ПМ2 значение 0 соответствует загрузке с eMMC-карты, значение 1 - с SD-карты.

4. Перезагрузить модуль или выполнить команду загрузки в режиме монитора U-Boot:

```
boot
```

5.4 Режим загрузки Linux по TFTP

Включение режима загрузки Linux по TFTP выполняется установкой переменных окружения загрузчика.

Например, для загрузки Linux по TFTP с использованием файловой системы, расположенной на SD/MMC-карте, необходимо выполнить следующие действия в мониторе загрузчика:

- Установить IP-адрес модуля:

```
setenv ipaddr <ip_address>
```

- Установить IP-адрес TFTP-сервера:

```
setenv serverip <server_ip_address>
```

- Задать параметры запуска Linux:

```
setenv bootargs 'console=ttyS0,115200 root=/dev/mmcblk0p2 rootfstype=ext4 rw  
↵rootwait'
```

- Задать команду загрузки Linux:

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

5.5 Режим монитора

После запуска загрузчик в течение 2 секунд ожидает от пользователя ввода любого символа через терминал UART0 для перехода в режим монитора.

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;
- `nm` — изменение содержимого памяти по постоянному адресу;
- `printenv` — печать переменных окружения;
- `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`.

5.6 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. *Расположение загрузчика в памяти*);
 - 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-носитель;
 - 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 `loadu` на модуле;
 - передать ELF-файл приложения с помощью программы `minicom` с использованием протокола YMODEM⁵ на ПЭВМ.
4. передать управление из U-Boot в приложение командой `bootelf <addr>`, где `<addr>` — адрес загруженного в памяти ELF-файла.

⁵ <https://en.wikipedia.org/wiki/YMODEM>

6. СБОРКА ЗАГРУЗЧИКА

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.

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
```

7. ОБРАЗ ЗАГРУЗЧИКА

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 7.1.

Таблица 7.1. Схема разбиения образа загрузчика

Область	Смещение (КБ)	Максимальный размер (КБ)
Образ U-Boot SPL	0	56
Переменные окружения ⁶	64	64
Образ U-Boot	128	

⁶ 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.

8. РАСПОЛОЖЕНИЕ ЗАГРУЗЧИКА В ПАМЯТИ

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 8.1.

Таблица 8.1. Расположение загрузчика в памяти

Область	Начальный адрес	Конечный адрес
Свободная память	0x40000000	0x7EFFFFFF
U-Boot ⁷	0x7F000000	0x7FFFFFFF
Свободная память	0xA0000000	0xDFFFFFFF

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

⁷ Начальный адрес области памяти, используемой загрузчиком, указан для справки и изменяется в зависимости от конфигурации загрузчика.

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

В

bootcmd, 11
bootenvcmd, 11
bootfile, 11
bootsource, 13

D

ddrctl_cid, 11
ddrctl_cmd, 10, 11

L

loadaddr, 11

M

mmcdev, 13

□

переменная окружения

- bootcmd, 11
- bootelf_addr, 10
- bootelf_elfsize, 11
- bootelf_spibus, 11
- bootelf_spioffset, 11
- bootenvcmd, 10, 11
- bootfile, 9, 11
- bootm_low, 9
- bootm_size, 9
- bootpartnum, 10
- bootsource, 10, 13
- cmdline, 10
- console, 10
- ddrctl_cid, 10, 11
- ddrctl_cmd, 9–11
- loadaddr, 9, 11
- mmcdev, 10, 13
- rootfstype, 10
- rootpartnum, 10
- set_bootargs, 10
- stdin,stdout,stderr, 9