|  |  |  |
| --- | --- | --- |
|  |  |  |

**ПАКЕТ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РАСШИРЕННОЙ ПОДДЕРЖКИ СБИС 1888BC048. АДАПТИРОВАННЫЙ ЗАГРУЗЧИК U-BOOT ДЛЯ ПЛАТЫ MT143.05 НА КРИСТАЛЛЕ СБИС 1888BC048**

Руководство оператора

**ЮФКВ.30174-01 34 01**

**Листов 24**

|  |  |  |
| --- | --- | --- |
|  |  |  |

Аннотация

Настоящее руководство распространяется на адаптированный загрузчик U-Boot для платы МT143.05 на кристалле СБИС 1888BC048 (далее – Загрузчик U-Boot, программа).

Документ содержит общие сведения о программе, условия её выполнения, описание действий оператора при выполнении программы, описание сообщений оператору.

Документ составлен с учётом требований ГОСТ 19.101-77, ГОСТ 19.105-78,   
ГОСТ 19.106-78.

Содержание

[1. Назначение программы 4](#_Toc26965701)

[1.1 Общие сведения 4](#_Toc26965702)

[2. Условия выполнения программы 6](#_Toc26965703)

[2.1. Требования к среде функционирования 6](#_Toc26965704)

[2.2. Установка программы 6](#_Toc26965705)

[2.3. Подготовка к запуску программы 6](#_Toc26965706)

[3. Выполнение программы 7](#_Toc26965707)

[3.1. Запуск программы 7](#_Toc26965708)

[3.2. Загрузка образа ядра 7](#_Toc26965709)

[3.3. Настройка переменных окружения загрузчика 9](#_Toc26965710)

[3.4. Работа с памятью 11](#_Toc26965711)

[3.5. Работа с Flash памятью 14](#_Toc26965712)

[3.6.Работа с I2C 15](#_Toc26965713)

[4. Сообщения оператору 17](#_Toc26965715)

[Перечень принятых сокращений 18](#_Toc26965716)

[Приложение А. Список доступных команд загрузчика U-Boot 20](#_Toc26965717)

# Назначение программы

## Общие сведения

### Адаптированный загрузчик U-Boot для платы MT143.05 на кристалле СБИС 1888BC048 входит в состав пакета программного обеспечения (далее по тексту ППО) расширенной поддержки СБИС 1888BC048, разработки НТЦ «Модуль», предназначенного для выполнения на аппаратных ресурсах платы MT143.05 на кристалле СБИС 1888BC048.

### Загрузчик U-Boot предназначен для загрузки U-Boot, образа ядра и device tree с:

* I2C EEPROM;
* SD карты;
* SPI Flash;
* Ethernet (TFTP, EDCL).

### Загрузчик U-Boot обеспечивает следующие основные возможности:

* Конфигурация DDR памяти на базе данных в EEPROM;
* Поддержка драйверов:
  + I2C;
  + UART;
  + Ethernet;
  + SDIO;
  + GPIO;
  + SPI;
  + PINMUX;
  + USB.
* Поддержка протокола TFTP на канале Ethernet;
* Управление через командную строку на порту UART;
* Тестирование фиксированного блока памяти DDR встроенным тестом;
* Запись и чтение SD карты;
* Запись и чтение файлов на томе FAT16/FAT32/EXT3/EXT4 на SD карте;
* Сохранение рабочих переменных на SD карте, I2C EEPROM или SPI flash памяти (в зависимости от параметров, указанных при компиляции);

### Используемая версия исходных кодов загрузчика U-Boot − 2020.11.

# Условия выполнения программы

## Требования к среде функционирования

### Загрузчик U-Boot должен исполняться на аппаратных ресурсах платы МT143.05 на СБИС 1888BC048.

### Условия эксплуатации Загрузчик U-Boot должны соответствовать условиям эксплуатации платы МT143.05.

## Установка программы

Для корректной работы с SPI Flash носителем необходимо выполнить прошивку загрузчика U-Boot на носитель. Подробнее см. документ «Пакет программного обеспечения расширенной поддержки СБИС 1888BC048. Адаптированный загрузчик U-Boot для платы MT143.05 на кристалле СБИС 1888BC048. Инструкция по применению исходных кодов».

Для корректной работы с SD-картой необходимо выполнить прошивку загрузчика U-Boot на карту. Подробнее см. документ «Пакет программного обеспечения расширенной поддержки СБИС 1888BC048. Адаптированный загрузчик U-Boot для платы MT143.05 на кристалле СБИС 1888BC048. Инструкция по применению исходных кодов».

## Подготовка к запуску программы

Для корректной работы загрузки по сети по протоколу TFTP оператор должен настроить перед работой с программой сервер TFTP.

# Выполнение программы

## Запуск программы

Программа запускается автоматически при подключении платы.

Управление программой выполняется через командную строку. Список всех доступных команд приведён в приложении 1 (см. Приложение А. Список доступных команд загрузчика U-Boot).

Для вывода полного списка доступных команд используется команда help.

Для получения справки по отдельной команде после команды help следует указать название интересующей команды. Например, при вводе help base оператору будет выведена следующая информация:

base - print or set address offset

Usage:

base

- print address offset for memory commands

base off

- set address offset for memory commands to 'off'=>

## Загрузка образа ядра

Загрузка образа ядра может выполняться с SPI Flash, SD-карты или по сети автоматически при запуске работы загрузчика, если это задано в переменных окружения (подробнее см. п. 3.3), или по соответствующей команде.

К основным командам загрузки относятся:

* boot – выполнение команды загрузки по умолчанию, то есть выполнение команды из переменной окружения bootcmd;
* bootd – выполнение команды загрузки по умолчанию, то есть выполнение команды из переменной окружения bootcmd;
* bootelf –загрузка образа ELF из памяти;
* bootm – загрузка образа приложения из памяти;

bootm [addr [arg ...]]

* bootp – загрузка образа по сети при помощи BOOTP протокола;

bootp [loadAddress] [[hostIPaddr:]bootfilename]

где [loadAddress] − адрес, откуда выполняется загрузка;

[hostIPaddr:] − IP-адрес хост-сервера;

[bootfilename] − имя загружаемого файла.

* bootvx – загрузка образа приложения VxWorks;
* tftp – загрузка образа по сети при помощи протокола TFTP;
* dhcp – загрузка образа по сети при помощи протокола DHCP;

bootp [loadAddress] [[hostIPaddr:]bootfilename]

где [loadAddress] − адрес, откуда выполняется загрузка;

[hostIPaddr:] − IP-адрес хост-сервера;

[bootfilename] − имя загружаемого файла.

* loadb – загрузка бинарного файла через терминал по протоколу Kermit;

loadb [ off ] [ baud ]

где [off] − смещение;

[baud:] − скорость передачи.

* loads – загрузка файла в формате S-Record через терминал;

loads [ off ] [ baud ]

где [off] − смещение;

[baud:] − скорость передачи.

### Загрузка образа по протоколу TFTP

Для корректной загрузки по протоколу TFTP должен быть настроен сервер TFTP и переменные окружения ipaddr, serverip и bootfile.

Для загрузки образа ядра через сеть по протоколу TFTP используется команда: tftpboot.

tftpboot [loadAddress] [[hostIPaddr:]bootfilename]

где [loadAddress] − адрес, откуда выполняется загрузка;

[hostIPaddr:] −IP-адрес хост-сервера;

[bootfilename] − имя загружаемого файла.

Если в строке не указан [loadAddress], то значение берётся из переменной окружения loadaddr.

## Настройка переменных окружения загрузчика

Переменные окружения используются для настройки поведения загрузчика по желанию оператора.

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

При успешном обнаружении устанавливаются переменные окружения, сохраненные в SPI Flash-памяти;

В случае ошибки устанавливаются переменные окружения, заданные по умолчанию.

Для настройки значений переменных окружения используется команда setenv.

Новое значение задаётся по следующей схеме:

setenv name value

где name − имя переменной окружения, а value − значение переменной.

### К переменным окружения загрузчика, которые может задать оператор, относятся:

* ipaddr - IP-адрес, необходимый для выполнения команды tftp. Подробнее см. п. 3.4;
* serverip - IP-адрес TFTP-сервера, необходимый для выполнения команды tftp. Подробнее см. п. 3.2.2;
* netmask – маска подсети;
* loadaddr − адрес буфера для копирования образа с SD/MMC-карты;
* bootdelay – время в секундах до выполнения команды, заданной в переменной bootcmd после перезагрузки загрузчика U-Boot. При этом на экране отображается обратный отсчёт, оператор может прервать выполнение команды, нажав любую кнопку на клавиатуре. Если значение переменной - «0», то загрузка выполняется сразу же;
* bootfile - имя файла, содержащего образ для загрузки образа по протоколу TFTP;
* bootcmd – команда, выполняемая при перезагрузке загрузчика после обратного отсчёта, заданного в переменной bootdelay;
* bootm\_low − нижняя граница области памяти, используемой для загрузки образа;
* bootm\_size − размер области памяти, используемой для загрузки образа;
* fdt\_addr\_r – адрес, где хранится device tree.

Команда setenv используется также для удаления заданного значения у переменной. Уже заданное значение удаляется по следующей схеме:

setenv name

где name − имя переменной окружения.

Для изменения уже заданных значений переменных окружения используется команда editenv.

Новое значение задаётся по следующей схеме:

edit name value

где name − имя переменной окружения, а value − новое значение переменной.

Все описанные выше изменения, выполняемые с переменными окружения, сохраняются только в RAM. При перезагрузке все настройки будут утеряны. Для того чтобы сохранить изменения, используется команда saveenv.

## Работа с памятью

При частом обращении к определенному участку памяти устанавливают смещение для команд обращения при помощи команды base. Пример:

=> base 0x100000

Base Address: 0x00100000

При вводе команды без указания смещения выводится текущее значение смещения. Значение по умолчанию: 0.

=> base

Base Address: 0x00000000

Для отключения смещения используется команда base off.

Для подсчёта контрольных сумм CRC32 используется команда crc32 с указанием диапазона памяти в качестве аргумента:

=> crc 0x100004 0x3FC

CRC32 for 00100004 ... 001003ff ==> 8083764e

При вводе трёх аргументов произведённый расчёт сохраняется в указанном участке памяти:

=> crc 0x100004 0x3FC 0x100000

CRC32 for 00100004 ... 001003ff ==> 8083764e

=> md 0x100000 4

00100000: 8083764e bd86200a 60a19054 2c12c402 ..vN.. .`..T,...

=>

Для сравнения содержимого памяти двух участков памяти используется команда cmp:

cmp [.b, .w, .l] addr1 addr2 count

Команда выполнит либо полную проверку участка, заданного в третьем аргументе (длина), либо остановится при первом обнаруженном различии:

=> cmp 0x100000 0x200000 0x400

word at 0x00100000 (0x8083764e) != word at 0x00200000 (0x27051956)

Total of 0 words were the same

Команда может получить доступ к участкам памяти с разными размерами информации:

cmp.l - 32 бита (longword). По умолчанию сравнение выполняет для 32-битной памяти;

cmp.w - 16 бита (word),

cmp.b - 8 байт (byte).

Аргумент count задаёт количество обрабатываемых единиц данных, т.е. размер машинных слов и символов (32 бита, 16 бит или 8 байт).

Команда cp используется для копирования участков памяти:

cp [.b, .w, .l] source target count

где [.b, .w, .l] – разрядность копируемой информации в битах или байтах,

source – участок, откуда копируются данные;

target – участок, куда копируются данные;

count – количество копируемых данных в указанной величине.

Например:

=> cp.l 0x200000 0x100000 0x10000

=> cp.w 0x200000 0x100000 0x20000

=> cp.b 0x200000 0x100000 0x40000

Команда md используется для отображения содержимого памяти:

md [.b, .w, .l] address [# of objects]

где [.b, .w, .l] – разрядность отображаемой информации в битах или байтах,

address – адрес участка памяти;

[# of objects] – количество объектов.

Отображение содержимого памяти выполняется и в шестнадцатеричной системе счисления, и в ASCII:

=> md 0x100000

00100000: 8083764e bd86200a 60a19054 2c12c402 ..vN.. .`..T,...

00100010: c101d028 00438198 7ab01239 62406128 ...(.C..z..9b@a(

00100020: 0c900d05 320b4581 1ca3d0a2 c498293a ....2.E.......):

Загрузчик запоминает последний указанный адрес и аргумент количества, поэтому при вводе команды без аргументов, будет автоматически подставлен следующий адрес и прежний аргумент количества.

Команда mm используется для изменения содержимого памяти:

mm [.b, .w, .l] address

где [.b, .w, .l] – разрядность изменяемой информации в битах или байтах,

address – адрес участка памяти;

После ввода команды на экран выводится текущее содержимое участка памяти и знак вопроса для ввода новых данных в шестнадцатеричной системе:

=> mm 0x100000

00100000: 8083764e ? 0

00100004: bd86200a ? 0xaabbccdd

00100008: 60a19054 ? 0x01234567

0010000c: 2c12c402 ? .

Для проведения теста RAM используется команда mtest:

mtest [start [end [pattern [iterations]]]]

Для записи разных данных несколько раз на один и тот же адрес используется команда nm:

nm [.b, .w, .l] address

где [.b, .w, .l] – разрядность изменяемой информации в битах или байтах,

address – адрес участка памяти.

Для запуска бесконечного цикла по диапазону адресов используется команда loop

loop [.b, .w, .l] address number\_of\_objects

где [.b, .w, .l] – разрядность информации в битах или байтах,

address – адрес участка памяти;

number\_of\_objects – количество объектов.

## Работа с Flash-памятью

Команда cp используется для копирования участков памяти. Подробнее см. п. 3.4.4.

Команда flinfo используется для вывода информации по Flash-памяти.

Команда erase используется для вывода информации по Flash.

Для удаления от одного участка памяти до конца сектора:

erase start end

Для удаления всех участков памяти:

erase all

Команда protect используется для включения или отключения защиты Flash-носителя от записи. Защиту можно поставить на отдельные сектора памяти или на всю память. Подробнее см. справку загрузчика U-boot в программе.

## Работа с I2C

### Команда i2c dev задает текущую шину I2C, например, для доступа к микросхеме D5 – i2c dev 0.

### Команда i2c write позволяет записать последовательность байтов из ОЗУ в устройство, начиная с заданного адреса, например, для записи 0x80 байтов в микросхему D5:

i2c write 60000 50 0.2 80

где 60000 – адрес в ОЗУ, 50 – адрес EEPROM, 0.2 – стартовый адрес EEPROM   
(2-х байтовый), 0x80 – число байтов.

### Команда i2c read позволяет прочитать последовательность байтов из устройства в ОЗУ, начиная с заданного адреса, например, для чтения 0x80 байтов из микросхемы D5:

i2c read 50 0.2 80 60080

где 60080 – адрес в ОЗУ, 50 – адрес EEPROM, 0.2 – стартовый адрес EEPROM  
(2-х байтовый), 0x80 – число байтов.

# Сообщения оператору

В процессе своей работы Загрузчик U-Boot формирует лишь один типа сообщений – информационные сообщения.

Информационные сообщения выводятся в строке после запуска команды, оповещая о статусе выполнения команды или полученных результатах.

# Приложение А. Список доступных команд загрузчика U-Boot

|  |  |
| --- | --- |
| Команда | Описание |
| base | установка смещения для команд обращения к памяти |
| bdinfo | печать информации о модуле |
| boot | выполнение команды загрузки |
| bootd | выполнение команды загрузки по умолчанию |
| bootelf | загрузка образа ELF из памяти |
| bootm | загрузка образа приложения из памяти |
| bootp | загрузка образа по сети при помощи BOOTP-протокола |
| bootvx | загрузка образа приложения VxWorks |
| cmp | сравнение содержимого памяти |
| coninfo | печать информации о консольных устройствах |
| cp | копирование содержимого памяти |
| crc32 | вычисление контрольной суммы |
| dhcp | загрузка образа по сети при помощи DHCP протокола |
| dm | печать информации о драйверах устройств |
| echo | печать аргументов |
| editenv | редактирование переменных окружения |
| env | управление переменными окружения |
| erase | удаление данных с Flash-памяти |
| exit | завершение выполнения сценария |
| ext4load | загрузка бинарного файла из файловой системы EXT4 |
| ext4ls | вывод списка всех файлов в папке из файловой системы EXT4 |
| ext4size | установка размера файла из файловой системы EXT4 |
| false | ничего не выполняет, неуспешное выполнение |
| fatinfo | печать информации о файловой системе |
| fatload | загрузка бинарного файла из файловой системы DOS |
| fatls | вывод списка всех файлов в папке из файловой системы DOS |
| fatsize | установка размера файла из файловой системы DOS |
| fdt | управление Flattened Device Tree (FDT) |
| flinfo | печать информации по Flash-памяти |
| fstype | просмотр типа файловой системы |
| go | запуск приложения по указанному адресу |
| gpio | запрос и управление интерфейса GPIO |
| help | печать справки и полного списка команд монитора |
| i2c | операции с i2c устройствами |
| iminfo | печать информации об образе приложения |
| imxtract | излечение образа из файла с несколькими образами |
| itest | возвращение значения true/false по целому числу |
| 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 |
| ls | вывод списка файлов в папке |
| md | отображение содержимого памяти |
| mm | изменение содержимого памяти с автоматическим увеличением адреса |
| mmap | маппинг физической памяти |
| mmc | функции для работы с подсистемой MMC |
| mmcinfo | отображение информации о MMC |
| mtest | простой тест RAM памяти на чтение/запись |
| mw | заполнение памяти |
| nfs | загрузка образа по сети при помощи NFS-протокола |
| nm | изменение памяти (постоянный адрес) |
| part | команды для разбиения диска |
| ping | отправка запроса ICMP ECHO\_REQUEST на хост-сервер в сети |
| printenv | печать переменных окружения |
| protect | включение/отключение защиты от записи для Flash-памяти |
| reginfo | печать регистрационных данных |
| reset | перезагрузка процессора |
| run | выполнение команд из указанной переменной окружения |
| save | сохранение файла в файловой системе |
| saveenv | сохранение переменных окружения |
| setenv | установка переменных окружения |
| setexpr | установка переменных окружения как результат оценочного выражения |
| showvar | печать локальных переменных hushshell |
| size | установка размера файла |
| sleep | отложенное выполнение команды |
| source | выполнение сценария из памяти |
| test | выполнение минимального теста |
| tftpboot | загрузка образа по сети при помощи протокола TFTP |
| true | ничего не выполняет, успешное выполнение |
| ums | использование UMS |
| version | печать версий монитора, компилятора и компоновщика |

# Перечень принятых сокращений

|  |  |
| --- | --- |
| ASCII | − American Standard Code for Information Interchange (Американский стандартный код для обмена информацией) |
| BOOTP | − Bootstrap protocol (Cетевой протокол, используемый для автоматического получения клиентом IP-адреса) |
| CRC | − Cyclic Redundancy Check (Циклический избыточный код) |
| DDR | − Double Data Rate (Удвоенная скорость передачи данных) |
| DHCP | − Dynamic Host Configuration Protocol (Протокол динамической настройки узла) |
| DOS | − Disk Operating System (Дисковая операционная система) |
| EDCL | − Ethernet Debug Communication Link (Отладочная коммуникационная ссылка Ethernet) |
| ELF | − Executable and Linkable Format (Формат исполнимых и компонуемых файлов) |
| EEPROM | − Electrically Erasable Programmable Read-Only Memory (Электрически стираемое перепрограммируемое ПЗУ) |
| EXT | − Extended File System (Расширенная файловая система) |
| FAT | − File Allocation Table (Таблица размещения файлов) |
| FDT | − Flattened Device Tree (Формат файла с упрощенным описание дерева устройств) |
| GPIO | − General-Purpose Input/Output (Интерфейс ввода/вывода общего назначения) |
| IP | − Internet Protocol (Межсетевой протокол) |
| MMC | − MultiMedia Card (Портативная флеш-карта памяти, использующаяся для многократной записи и хранения информации в портативных электронных устройствах) |
| NFS | − Network File System (ПUротокол сетевого доступа к файловым системам) |
| RAM | − Random Access Memory (Память с произвольной выборкой) |
| SD | − Secure Digital (Формат карт памяти для использования в портативных устройствах) |
| SDIO | − Secure Digital Input Output (Стандарт, поддержка которого позволяет использовать со слотом расширения формата SD/MMS соответствующую периферию) |
| SPI | − Serial Peripheral Interface (Последовательный периферийный интерфейс) |
| TFTP | − Trivial File Transfer Protocol (Простой протокол передачи файлов) |
| UART | − Universal Asynchronous Receiver-Transmitter (Универсальный асинхронный приёмопередатчик) |
| USB | − Universal Serial Bus (Универсальная последовательная шина) |
| НТЦ | − научно-технический центр |
| ППО | − пакет программного обеспечения |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Лист регистрации изменений** | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц) в докум. | №  документа | Входящий № сопрово-дительно- го докум. и дата | Подп. | Дата |
| Изм. | изменен­ных | заменен­ных | новых | аннули­рован-ных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |