

**ДИСТРИБУТИВ ОС GNU/LINUX НА
БАЗЕ BUILDROOT ДЛЯ 1892ВМ14Я.
РУКОВОДСТВО СИСТЕМНОГО
ПРОГРАММИСТА**

**Версия v2.3
06.06.2017**

ОГЛАВЛЕНИЕ

1	О документе	3
2	Общие сведения о дистрибутиве ОС	4
3	Состав дистрибутива ОС	5
4	Состав образа SD-карты	6
5	Сборка образа SD-карты и образа загрузчика	7
6	Запись образа SD-карты	9
7	Прошивка SPI флеш-памяти	10
8	Настройка ОС	11
8.1	Увеличение объема ОЗУ	11
8.2	Включение драйвера framebuffer voutfb	11
8.3	Настройка сети	11
8.4	Настройка 6LoWPAN	11
8.5	Параметры запуска Linux	12
8.6	Добавление программ в образ SD-карты	12
8.7	Управление выводами GPIO из пространства пользователя	12
8.8	Разметка свободной области SD-карты и монтирование в /data	13
9	Запуск отладочного модуля	14
10	Сообщения системному программисту	15

1. О ДОКУМЕНТЕ

Документ описывает дистрибутив операционной системы GNU/Linux на базе Buildroot для микросхемы 1892ВМ14Я (МСom-02), процедуру сборки и прошивки образа SD-карты с операционной системой и настройки операционной системы (далее — “ОС”).

Документ описывает дистрибутив версии 2.3.

2. ОБЩИЕ СВЕДЕНИЯ О ДИСТРИБУТИВЕ ОС

Дистрибутив ОС GNU/Linux предназначен для распространения исходных кодов ОС GNU/Linux, инструментального ПО и скриптов сборки ОС GNU/Linux.

Дистрибутив ОС поддерживает следующие отладочные модули на базе микросхемы 1892ВМ14Я:

- 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.

Дистрибутив ОС распространяется в виде архива исходных кодов (подробнее см. “Состав дистрибутива ОС”).

Результатом сборки исходных кодов дистрибутива являются:

- Образ SD-карты, содержащий ОС GNU/Linux (подробнее см. “Состав образа SD-карты”). Образ SD-карты является унифицированным и совместим со всеми поддерживаемыми модулями.
- Образы загрузчика U-Boot для всех поддерживаемых отладочных модулей (подробнее см. “Загрузчик U-Boot для 1892ВМ14Я. Руководство пользователя”).

Дистрибутив ОС построен на базе Buildroot версии 2015.08.1¹.

В состав ОС входит набор демо-тестов для проверки работы блоков и интерфейсов СнК в составе отладочного модуля.

В качестве ядра ОС используется Linux v4.1.39.1. Ядро построено на базе Linux версии 4.1.39².

В качестве загрузчика ОС используется U-Boot v2017.01.0.2 (подробнее см. “Загрузчик U-Boot для 1892ВМ14Я. Руководство пользователя”).

¹ <https://git.buildroot.net/buildroot/commit/?h=2015.08.1&id=e009e7d8615eec10d2c0c2676ef5a276f0a6a5e2>

² <https://git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/commit/?h=v4.1.39>

3. СОСТАВ ДИСТРИБУТИВА ОС

Состав архива дистрибутива представлен в таблице 3.1.

Таблица 3.1. Состав архива дистрибутива

Директория/файл	Описание
build.sh	Скрипт для сборки ОС, разметки и создания образа SD-карты
baremetal-src/tests	baremetal тесты для функционального контроля плат
buildroot-script	Рецепты и скрипты для сборки корневой файловой системы на базе Buildroot
u-boot	Исходные коды загрузчика U-Boot
tools	Дополнительные утилиты
tools/flash-spi.py	Утилита для прошивки SPI флеш-памяти версии 2.0, или выше, управляет монитором BootROM СнК по интерфейсу UART
toolchain-arm-cs-bare-2013.11	Пакет программ от CodeSourcery, необходимых для компиляции и генерации выполняемого кода из исходных текстов baremetal-приложений на ARM

4. СОСТАВ ОБРАЗА SD-КАРТЫ

Схема разбиения образа SD-карты представлена в таблице 4.1.

Таблица 4.1. Схема разбиения образа SD-карты на области

Область	Начало (байт)	Размер (байт)	Примечание
MBR	0	512	
Раздел boot	1 МиБ	128 МиБ	Раздел с файловой системой FAT32
Раздел root	129 МиБ	1 ГиБ	Раздел с файловой системой EXT4 с корневой файловой системой rootfs

Состав раздела boot:

- `zImage` — скомпилированное ядро Linux
- `test-ddr.bin` — параметризуемый baremetal тест для проверки DDR-памяти;
- `test-nand.bin` — baremetal тест для проверки NAND-памяти;
- `test-bist.bin` — параметризуемый baremetal тест для проверки памяти посредством BIST;
- `test-mfbsp-gpio.bin` — параметризуемый baremetal тест для проверки MF BSP GPIO в режиме loopback;
- `u-boot.env` — переменные окружения загрузчика U-Boot.

5. СБОРКА ОБРАЗА SD-КАРТЫ И ОБРАЗА ЗАГРУЗЧИКА

Сборка образа SD-карты и образа загрузчика выполняется на ПЭВМ. ПЭВМ должна удовлетворять требованиям:

1. не менее 4 ГиБ ОЗУ, 8 ГиБ свободного места на НЖМД или твердотельном накопителе;
2. на ПЭВМ должен быть установлен кард-ридер для подключения SD-карт;
3. на ПЭВМ должен быть установлен дистрибутив GNU/Linux CentOS 7.2 архитектуры x86-64;
4. на ПЭВМ должен быть настроен доступ в интернет. Если доступ в интернет осуществляется через прокси-сервер, то должно быть установлены переменные окружения `http_proxy`, `https_proxy`, `ftp_proxy`;
5. на ПЭВМ должны быть установлены следующие приложения (пути до исполняемых файлов должны быть прописаны в переменной окружения `PATH`):
 - `bash` версии 4.1.2;
 - `cmake` версии 2.8 или выше;
 - `GNU make` версии 3.81;
 - `parted` версии 2.1;
 - `texinfo` версии 4.13;
 - `glibc.i686`;
 - `ncurses-devel` версии 5.9 или выше.

Для сборки образов необходимо:

1. разархивировать архив для сборки образа SD-карты (`<package-name>` - имя архива `tar.bz2`, но без расширения `tar.bz2`):

```
tar xf <package-name>.tar.bz2
```

Предупреждение: Полный путь к архиву и имя архива не должны содержать пробелов.

2. перейти в распакованную директорию:

```
cd <package-name>
```

3. выполнить команду по запуску сборки Linux, загрузчика и приложений:

```
./build.sh build
```

Длительность сборки составляет около 45 минут и зависит от производительности CPU ПЭВМ.

4. выполнить команду для подготовки образа SD-карты:

```
./build.sh mk_image
```

При запуске появится запрос пароля для sudo (sudo требуется для монтирования образа и записи корневой файловой системы).

После завершения процедуры сборки в директории output будут доступны следующие файлы образов:

- mcom02-buildroot-sdcard.img

Образ SD-карты, содержащий ОС GNU/Linux.

- mcom02-*-uboot.img

Образы загрузчика для всех поддерживаемых отладочных модулей.

6. ЗАПИСЬ ОБРАЗА SD-КАРТЫ

Для записи образа на SD-карту необходимо:

1. извлечь SD-карту из кард-ридера ПЭВМ и считать список устройств командой:

```
ls -la /dev/sd*
```

2. вставить SD-карту в кард-ридер ПЭВМ и повторно считать список устройств командой `ls -la /dev/sd*`. Вычесть из списка устройств после установки SD-карты список устройств до установки карты и получить устройство `/dev/sdX` и/или список устройств `/dev/sdX1`, `/dev/sdX2`... (где 1, 2, ... номера разделов SD-карты). В случае, если получен список устройств, то получить устройство `/dev/sdX` отбрасыванием последней цифры из устройства соответствующего первому разделу SD-карты `/dev/sdX1`.

3. записать образ на SD-карту:

```
sudo dd if=output/mcom02-buildroot-sdcard.img of=/dev/sdX bs=4M  
sudo sync
```

4. извлечь SD-карту из кард-ридера ПЭВМ.

7. ПРОШИВКА SPI ФЛЕШ-ПАМЯТИ

Прошивка SPI флеш-памяти отладочного модуля образом загрузчика (см. “Сборка образа SD-карты и образа загрузчика”) выполняется утилитой `flash-spi.py`, входящей в состав дистрибутива, согласно документу “Инструкция по прошивке SPI флеш-памяти отладочных модулей на базе 1892ВМ14Я”.

8. НАСТРОЙКА ОС

8.1 Увеличение объема ОЗУ

По умолчанию доступный объем ОЗУ равен 1 ГиБ. Для увеличения объема ОЗУ до 2 ГиБ необходимо активировать второй контроллер DDR (если применимо для данного модуля).

Включение контроллера DDR выполняется установкой значения `enable` для переменной окружения `ddrctl_cmd` в режиме монитора загрузчика (подробнее см. “Загрузчик U-Boot для 1892BM14Я. Руководство пользователя”):

```
setenv ddrctl_cmd enable
saveenv
```

8.2 Включение драйвера `framebuffer voutfb`

По умолчанию образ SD-карты собирается с выключенным модулем ядра `voutfb`. Для включения необходимо удалить фразу `modprobe.blacklist=voutfb` из файла `u-boot.env`.

8.3 Настройка сети

По умолчанию ОС настроена на получение сетевого адреса по DHCP. Настройка параметров сети задаётся в конфигурационном файле `/etc/network/interfaces` на корневой файловой системе. Полная документация по настройке сети доступна на странице <https://wiki.debian.org/NetworkConfiguration>.

Имя хоста по умолчанию — `mcom`. Для изменения имени хоста необходимо отредактировать конфигурационные файлы `/etc/hostname` и `/etc/hosts` на корневой файловой системе.

8.4 Настройка 6LoWPAN

Сетевой интерфейс 6LoWPAN обеспечивает взаимодействие по протоколу IPv6 через сеть стандарта IEEE 802.15.4.

Параметры сетевого интерфейса задаются в файле `/etc/network/interfaces` корневой файловой системы.

Пример настройки параметров интерфейса `lowpan0` с заданием статического IPv6-адреса `fc00::1`:

```
auto lowpan0
iface lowpan0 inet6 static
```

```
address fc00::1
netmask 8
```

8.5 Параметры запуска Linux

Параметры запуска Linux задаются с помощью переменных окружения загрузчика (подробнее см. “Загрузчик U-Boot для 1892BM14Я. Руководство пользователя”).

Для установки переменных окружения загрузчика дополнительно используется файл `u-boot.env`, импортируемый перед загрузкой Linux.

Для задания дополнительных параметров запуска Linux предназначена переменная окружения `cmdline`.

Список параметров запуска Linux находится в `linux/Documentation/kernel-parameters.txt`.

8.6 Добавление программ в образ SD-карты

Система сборки Buildroot поддерживает добавление в сборку программ и библиотек пользователя. Подробная документация находится в директории `buildroot-script/buildroot/docs`.

8.7 Управление выводами GPIO из пространства пользователя

Управление выводами GPIO SnK осуществляется с помощью служебных файлов в `/sys/class/gpio` (см. <https://www.kernel.org/doc/Documentation/gpio/sysfs.txt>).

Управление выводом GPIO может быть недоступно, если он используется драйвером.

В соответствии с руководством пользователя на SnK выводы GPIO делятся на 4 группы: GPIOA, GPIOB, GPIOC, GPIOD. В Linux выводы GPIO обозначаются номерами. Соответствие базовых номеров для каждой группы приведено в таблице 8.1. Внутри групп номера идут по порядку. Например, вывод GPIOA5 соответствует номеру $480 + 5 = 485$ в Linux.

Таблица 8.1. Соответствие обозначений выводов GPIO номерам в Linux

Группа	Базовый номер в Linux
GPIOA	480
GPIOB	448
GPIOC	416
GPIOD	384

8.8 Разметка свободной области SD-карты и монтирование в /data

Для создания раздела из свободной области SD-карты и монтирования в директорию /data необходимо:

1. Запустить отладочный модуль.
2. Выполнить команду:

```
create-data-partition && echo "Partition successfully created"
```

Разметку считать завершённой успешно в случае вывода в терминал скриптом сообщения:

```
Partition successfully created
```

9. ЗАПУСК ОТЛАДЧНОГО МОДУЛЯ

Для запуска отладочного модуля необходимо выполнить следующие действия:

1. *Собрать образ SD-карты и образ загрузчика для отладочного модуля.*
2. *Записать образ SD-карты.*
3. *Прошить SPI флеш-память отладочного модуля образом загрузчика.*
4. *Настроить ОС.*
5. *Установить SD-карту в слот MicroSD отладочного модуля.*
6. *Установить переключатель BOOT отладочного модуля в положение, соответствующее загрузке из SPI флеш-памяти (подробнее см. Руководство пользователя на отладочный модуль).*
7. *Подключить отладочный модуль к источнику питания (подробнее см. Руководство пользователя на отладочный модуль).*
8. *Открыть терминал UART отладочного модуля, или установить соединение по протоколу SSH (логин: root, пароль: root).*
9. *Выполнить команду uname -a. Считать отладочный модуль готовым к использованию при выводе в терминал сообщения:*

```
Linux mcom 4.1.39.1 #1 SMP Wed May 3 15:03:28 MSK 2017 armv7l GNU/Linux
```

10. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

Отладочный модуль выводит в терминал UART сообщения о ходе загрузки. Пример вывода в терминал при успешной загрузке:

```
DDR retention disabled

U-Boot SPL 2017.01.0.2 (May 03 2017 - 15:06:15)
DDR controllers init started
DDR controller #0 init done
DDR controller #1 init done
Trying to boot from SPI

U-Boot 2017.01.0.2 (May 03 2017 - 15:06:15 +0300), Build: v2.3-2017-05-03

CPU:   MCom-compatible
Model: Salute-EL24D1 r1.3
DRAM:  2 GiB
MMC:   sdhci0@3800b000: 0, sdhci1@3800d000: 1
SF: Detected m25p32 with page size 256 Bytes, erase size 64 KiB, total 4 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
DDR controller #1 disabled
Hit any key to stop autoboot:  0
switch to partitions #0, OK
mmc0 is current device
reading u-boot.env
35 bytes read in 10 ms (2.9 KiB/s)
reading zImage
3290544 bytes read in 164 ms (19.1 MiB/s)
## Flattened Device Tree blob at 7f770610
   Booting using the fdt blob at 0x7f770610
   Loading Device Tree to 4fff8000, end 4ffff8cd ... OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
[ 0.000000] Booting Linux on physical CPU 0x0
[ 0.000000] Linux version 4.1.39.1 (jenkins_drap@leo-pc.elvees.com)
(gcc version 5.2.0 (Buildroot 2015.08.1) ) #1 SMP Wed May 3 15:03:28 MSK 2017
...

Welcome!
mcom login:
```

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

С

cmdline, 12

Д

ddrctl_cmd, 11

□

переменная окружения

cmdline, 12

ddrctl_cmd, 11