

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

Версия v2.9

11.10.2018

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
5.1	Сборка в среде ОС ПЭВМ	7
5.2	Сборка в среде контейнера Docker	8
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 vputfb	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 модуля	14
10	Сообщения системному программисту	15
11	Приложение А. Демонстрация передачи управления из U-Boot в baremetal-приложение	16

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 2.9.

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 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 или Салют-ЭЛ24ПМ1 r1.2;
- Салют-ЭЛ24ОМ1 r1.2 с установленным Салют-ЭЛ24ПМ1 r1.2, Салют-ЭЛ24ПМ2 r1.0 или Салют-ЭЛ24ПМ2 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 модулей (подробнее см. “Загрузчик U-Boot для 1892ВМ14Я. Руководство пользователя”).

Дистрибутив ОС построен на базе Buildroot версии 2018.02.3¹.

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

В качестве ядра ОС используется Linux 4.4.111.8².

В качестве загрузчика ОС используется U-Boot 2017.07.0.14 (подробнее см. “Загрузчик U-Boot для 1892ВМ14Я. Руководство пользователя”).

¹ <https://git.buildroot.net/buildroot/commit/?h=2018.02.3>

² <https://github.com/elvees/linux/tree/mcom02/v4.4.111.8>

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
baremetal-src/tests/sample-cpu01	приложение для демонстрации передачи управления из U-Boot (подробнее см. Приложение А. Демонстрация передачи управления из U-Boot в baremetal-приложение)
buildroot-script	Рецепты и скрипты для сборки корневой файловой системы на базе Buildroot
u-boot	Исходные коды загрузчика U-Boot
tools	Дополнительные утилиты
tools/mcom02-flash	Пакет MCom-02 flash tools версии 2.1.1. Состоит из утилит для прошивки SPI флеш-памяти и SD/MMC-карты.
toolchain-arm-cs-bare-2013.11	Пакет программ от CodeSourcery, необходимых для компиляции и генерации выполняемого кода из исходных текстов baremetal-приложений на ARM
Dockerfile	Конфигурация <i>Docker-образа</i> ³ для среды сборки дистрибутива
Makefile	Скрипт сборки Docker-образа и сборки дистрибутива в контейнере Docker

³ <https://docs.docker.com/>

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
- `sample-cpu01.bin` — приложение для демонстрации передачи управления из U-Boot;
- `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;

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 7.5 архитектуры x86-64;
4. на ПЭВМ должен быть настроен доступ в интернет. Если доступ в интернет осуществляется через прокси-сервер, то должны быть установлены переменные окружения `http_proxy`, `https_proxy`, `ftp_proxy`;

Перед сборкой образов необходимо:

1. разархивировать архив дистрибутива (`<package-name>` — имя архива 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>
```

Сборка дистрибутива выполняется в среде ОС ПЭВМ или в среде контейнера Docker.

5.1 Сборка в среде ОС ПЭВМ

1. Установить RPM-пакеты, перечисленные в файле `Dockerfile`:

```
sudo yum install <packages-from-dockerfile> -y
```

2. Выполнить команду по запуску сборки Linux, загрузчика и приложений:

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

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

3. 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 модулей.

5.2 Сборка в среде контейнера Docker

1. Установить и настроить сервис Docker:

1. Установить Docker версии 17.07 или выше на ПЭВМ согласно инструкции [Get Docker CE for CentOS](#)⁴.
2. Добавить текущего пользователя в группу `docker` согласно инструкции [Post-installation steps for Linux](#)⁵.
3. Настроить прокси для сервиса Docker (при необходимости) согласно инструкции [Control Docker with systemd](#)⁶.
4. Настроить прокси для клиента Docker (при необходимости) согласно инструкции [Configure the Docker client](#)⁷.

2. Запустить сборку образа Docker:

```
make checkenv docker-image
```

Проверить, что образ создан командой `docker images | grep buildroot`. Пример вывода:

```
elvees mcom02-buildroot-centos-v1.0 ad261c2c728c 26 hours ago 533MB
```

3. Запустить контейнер Docker и сборку образов Buildroot:

```
make build-in-docker
```

После завершения процедуры сборки в директории `output` будут доступны файлы образов. Имена файлов образов соответствуют именам файлов образов перечисленных в главе *Сборка в среде ОС ПЭВМ*.

⁴ <https://docs.docker.com/install/linux/docker-ce/centos/>

⁵ <https://docs.docker.com/install/linux/linux-postinstall/>

⁶ <https://docs.docker.com/config/daemon/systemd/>

⁷ <https://docs.docker.com/network/proxy/#configure-the-docker-client>

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 модуля образом загрузчика (см. *“Сборка образа SD-карты и образа загрузчика”*) выполняются утилитами из пакета MCom-02 flash tools, входящего в состав дистрибутива, согласно документу *“Инструкция по прошивке SPI флеш-памяти модулей на базе 1892BM14Я”*.

8. НАСТРОЙКА ОС

8.1 Увеличение объема ОЗУ

Доступный объем ОЗУ по умолчанию равен 1 ГиБ для модулей Салют-ЭЛ24Д1 и Салют-ЭЛ24Д2, 2 ГиБ — для модулей Салют-ЭЛ24ОМ1. 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 1892ВМ14Я. 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 модулей Салют-ЭЛ24Д1 и Салют-ЭЛ24Д2. При необходимости включение произвести путем удаления строки `modprobe.blacklist=voutfb` из переменной окружения `cmdline` в режиме монитора загрузчика (подробнее см. “Загрузчик U-Boot для 1892ВМ14Я. Руководство пользователя”)

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”).

Совет: При работе с Салют-ЭЛ24ОМ1 для изменения загрузочного устройства на карту памяти microSD необходимо установить переменную окружения загрузчика `mtddev` в значение 1.

Дополнительные параметры запуска Linux необходимо передавать через переменную окружения `cmdline` загрузчика.

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`.

Возможно добавление программ на базе стандарта компьютерного зрения OpenVX (см. “ELVEES OpenVX SDK для 1892BM14Я. Руководство пользователя”).

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 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 МОДУЛЯ

Для запуска модуля необходимо выполнить следующие действия:

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

```
Linux mcom 4.1.41.3 #1 SMP Fri Sep 1 17:08:44 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 2017.07.0.3 (Sep 01 2017 - 17:12:01)
DDR controllers init started
DDR controller #0 init done
DDR controller #1 init done
Trying to boot from SPI

U-Boot 2017.07.0.3 (Sep 01 2017 - 17:12:01 +0300), Build: v2.5-2017-09-01

CPU:   MCom-compatible
Model: Salute-EL240M1 r1.1
I2C:   ready
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(part 0) is current device
reading zImage
3310424 bytes read in 211 ms (15 MiB/s)
## Flattened Device Tree blob at 7f768140
   Booting using the fdt blob at 0x7f768140
   Loading Device Tree to 4fff8000, end 4ffffbcd ... OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
[ 0.000000] Booting Linux on physical CPU 0x0
[ 0.000000] Linux version 4.1.41.3 (jenkins_drap@leo-pc.elvees.com)
(gcc version 5.2.0 (Buildroot 2015.08.1) ) #1 SMP Fri Sep 1 17:08:44 MSK 2017
...

Welcome!
mcom login:
```

11. ПРИЛОЖЕНИЕ А. ДЕМОНСТРАЦИЯ ПЕРЕДАЧИ УПРАВЛЕНИЯ ИЗ U-BOOT В BAREMETAL-ПРИЛОЖЕНИЕ

Для демонстрации передачи управления из U-Boot в baremetal-приложение используется приложение `sample-cpu01`. Приложение выполняет:

1. CPU0 включает домен питания для CPU1 и передает управление CPU1 в функцию `cpu1_entry()`;
2. CPU0, CPU1 печатают в терминал сообщение:

```
Hello from CPUx
```

3. CPU0, CPU1 переходят в бесконечный цикл ожидания.

Для запуска приложения `sample-cpu01` необходимо выполнить:

1. Перевести U-Boot в режим монитора.
2. Загрузить файл приложения в память ОЗУ модуля. Например, для модуля Салют-ЭЛ24ОМ1 с установленным модулем Салют-ЭЛ24ПМ1 и установленной SD-картой необходимо выполнить команду `load mmc 1:1 0x20001000 sample-cpu01.bin`.
3. Передать управление из U-Boot в приложение командой `go 0x20001000 0 0`.

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

С

cmdline, 11, 12

Д

ddrctl_cmd, 11

М

mmcdev, 12

□

переменная окружения

cmdline, 11, 12

ddrctl_cmd, 11

mmcdev, 12