

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

Версия v1.3

29.03.2016

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	7
6	Настройка ОС	9
6.1	Настройка типа платы	9
6.2	Увеличение объема ОЗУ	9
6.3	Настройка сети	9
6.4	Параметры ядра Linux	9
6.5	Добавление программ в образ SD-карты	10
6.6	Управление выводами GPIO из пространства пользователя	10
6.7	Разметка свободной области SD-карты и монтирование в /data	10
7	Проверка ОС	12
8	Сообщения системному программисту	13

1 О ДОКУМЕНТЕ

Документ описывает состав дистрибутива операционной системы GNU/Linux для микросхемы 1892ВМ14Я (МСom-02), процедуру сборки и прошивки образа SD-карты с операционной системой и настройки операционной системы (далее - "ОС").

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.

ОС предназначена для работы в составе модулей на базе микросхемы 1892ВМ14Я.

ОС построена на базе ядра Linux версии 4.1.0.

Загрузчик ОС построен на базе загрузчика U-Boot версии [d77447fdb122dab290fb1ad184a62456011e6e06](http://git.denx.de/?p=u-boot.git;a=commit;h=d77447fdb122dab290fb1ad184a62456011e6e06)¹.

Дистрибутив ОС построен на базе Buildroot версии 2015.08.

Дистрибутив ОС распространяется в виде архивированного пакета (подробнее см. “Состав дистрибутива ОС”).

Результатом сборки исходных кодов ОС является образ SD-карты соответствующего модуля (подробнее см. “Состав образа SD-карты”).

¹<http://git.denx.de/?p=u-boot.git;a=commit;h=d77447fdb122dab290fb1ad184a62456011e6e06>

3 СОСТАВ ДИСТРИБУТИВА ОС

Состав разархивированного пакета дистрибутива представлен в Таблица 3.1.

Таблица 3.1. Состав пакета

Директория/файл	Описание
build.sh	Скрипт для сборки ОС, разметки и создания образа SD-карты
baremetal-src/fixed-bootrom	Исходные коды загрузчика с 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: <code>flash-spi.py</code> - утилита для прошивки SPI-флеш установленной на плате (при наличии). Утилита управляет по интерфейсу UART монитором Bootrom СнК.
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 Таблица 4.1.

Таблица 4.1. Схема разбиения образа SD-карты на области

Область	Начало (байт)	Размер (байт)	Примечание
MBR	0	512	
–резерв–	512	512	
u-boot SPL	1024	22800	
–резерв–	23824	42736	
u-boot	66560	217432	
–резерв–	283860	764584	
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;
- `*.dtb` — скомпилированные бинарные файлы Device Tree Blob с описанием плат для ядра Linux;
- `u-boot.env` — переменные окружения загрузчика U-Boot.

5 СБОРКА ОБРАЗА SD-КАРТЫ

Сборка образа SD-карты 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 6.3 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.

Для сборки образа SD-карты необходимо:

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

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

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_<board> ,
```

При запуске появится запрос пароля для sudo (sudo требуется для монтирования образа и записи корневой файловой системы). После запуска этой команды будет доступен файл `sdcard-<board>.img`.

Для просмотра доступных опций `<board>` необходимо выполнить команду:

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

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` соответствующее установленной SD-карте;

3. записать образ на SD-карту:

```
dd if=sdcard-<board>.img of=/dev/sdX bs=4M  
sync
```

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

6 НАСТРОЙКА ОС

6.1 Настройка типа платы

При загрузке ядра Linux U-Boot должен передавать файл Device Tree Blob соответствующий типу платы. Для указания типа платы необходимо в файле `u-boot.env` (находится в корне раздела boot SD-карты) изменить строку:

```
fdtfile=<board-type>.dtb
```

где `<board-type>` - требуемый тип платы.

6.2 Увеличение объема ОЗУ

По умолчанию объем доступной ОЗУ равен 1 ГиБ. Для увеличения объема ОЗУ до 2 ГиБ необходимо активировать второй DDR-контроллер (если применимо для данного модуля). Для этого в файл `u-boot.env` необходимо добавить строку:

```
ddrctl_cmd=enable
```

6.3 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.

6.4 Параметры ядра Linux

Для передачи параметров ядру Linux используется переменная `bootargs` в файле `u-boot.env` в корне раздела boot. Загрузчик U-Boot считывает файл `u-boot.env` при загрузке, значение переменной `bootargs` используется в качестве параметров ядра.

Основные используемые параметры ядра:

1. `console=ttyS0,115200` - включает вывод сообщений ядра на UART0;

2. `earlyprintk` - включает вывод сообщений ядра на ранних этапах загрузки.

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

6.5 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`.

6.6 Управление выводами GPIO из пространства пользователя

Управление выводами GPIO СнК 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 СнК выводы GPIO делятся на 4 группы: GPIOA, GPIOB, GPIOC, GPIOD. В Linux выводы GPIO обозначаются номерами. Соответствие базовых номеров для каждой группы приведено в Таблица 6.1. Внутри групп номера идут по порядку. Например, вывод GPIOA5 соответствует номеру $480 + 5 = 485$ в Linux.

Таблица 6.1. Соответствие обозначений выводов GPIO номерам в Linux

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

6.7 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 ОМ и выполнить логин, как описано в главе “Проверка ОС”.
2. Выполнить команду:

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

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

```
Partition successfully created
```

7 ПРОВЕРКА ОС

Для проверки корректности работы ОС необходимо:

1. *собрать и прошить SD-карту;*
2. *настроить ОС при необходимости;*
3. зайти на устройство по протоколу SSH, логин: root, пароль: root;
4. выполнить команду `uname -a`. ОС считать работающей корректно в случае вывода в терминал сообщения:

```
Linux <hostname> 4.1.0 #1 SMP Fri Jun 26 21:23:43 MSK 2015  
armv7l GNU/Linux.
```

8 СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

1. ОС выводит сообщения через последовательный интерфейс UART0. В случае успешной загрузки выводятся следующие сообщения:

```
Start SD loader
DDR controller #0 init ... done

U-Boot SPL 2015.04 (Jun 16 2015 - 14:21:40)

U-Boot 2015.04 (Jun 16 2015 - 14:21:40)

CPU:   MCom-02
DRAM:  1 GiB
MMC:   sdhci: 0
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  0
Loading Linux...
13973256 bytes read in 1298 ms (10.3 MiB/s)
10171 bytes read in 45 ms (220.7 KiB/s)
75 bytes read in 31 ms (2 KiB/s)
## Info: input data size = 540 = 0x21C
Kernel image @ 0x40008000 [ 0x000000 - 0xd53708 ]
## Flattened Device Tree blob at 51000000
   Booting using the fdt blob at 0x51000000
   Loading Device Tree to 4eff9000, end 4effffff ... OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.1.0 (<username>@<build-hostname>)
(gcc version 4.8.3 20140320 (prerelease) (Sourcery CodeBench
Lite 2014.05-29) )
#20 SMP Fri Jun 26 21:23:43 MSK 2015
...

Welcome!
mcom login:
```

2. Сообщения U-boot описаны в `u-boot/doc`.
3. Сообщения ядра ОС Linux описаны в `linux/Documentation`.

4. Сообщения об ошибках и действия по их устранению описаны в Таблица 8.1.

Таблица 8.1. Сообщения об ошибках при загрузке ОС

Сообщение об ошибке	Описание ошибки	Действия по устранению
Error: No card in SDMMC	В устройстве отсутствует SD-карта	Установить прошитую SD-карту в Модуль BM
** File not found u-boot.env **	На SD-карте отсутствует файл u-boot.env. Загрузка возможна, но сеть настроена не будет	Создать файл u-boot.env в корне SD-карты
** File not found <board-type>.dtb **	На SD-карте отсутствует файл <board-type>.dtb	Пересобрать образ загрузочной SD-карты, перепрошить SD-карту
libfdt fdt_check_header(): FDT_ERR_BADMAGIC	Файл <board-type>.dtb повреждён	Пересобрать образ загрузочной SD-карты, перепрошить SD-карту
** File not found zImage **	На карте отсутствует файл zImage	Пересобрать образ загрузочной SD-карты, перепрошить SD-карту
Bad Linux ARM zImage magic!	Файл zImage повреждён	Пересобрать образ загрузочной SD-карты, перепрошить SD-карту
Failed to mount ext2 filesystem...	Повреждена файловая система на первом разделе	Пересобрать образ загрузочной SD-карты, перепрошить SD-карту
Error: Header is invalid	Повреждён загрузчик U-boot	Пересобрать образ загрузочной SD-карты, перепрошить SD-карту
add_sdhci: mmc create fail! MMC Device 0 not found spl: mmc device not found!!	Повреждён загрузчик U-boot	Пересобрать образ загрузочной SD-карты, перепрошить SD-карту
spl: mmc init failed: err - -18	Повреждена SD-карта	Заменить SD-карту, прошить SD-карту