

# Рекомендации по установке ОС Linux на модуль TE-AM1705v2

## Оглавление

Вступление.....	2
Первичная загрузка процессора AM1705.....	2
Источник загрузки : UART1 .....	3
Начальная загрузка TE-AM1705.....	4
Загрузка ОС Linux с использованием интерфейса Ethernet .....	6
Запись ОС Linux в энергонезависимую память .....	7
Запись ОС Linux в SPI dataflash + SDIO .....	7
Запись ОС Linux в NAND flash память. ....	9
Приложение 1 – Лог запуска утилиты slh_OMAP-L137.exe .....	12
Приложение 2 – Лог загрузки с использованием интерфейса Ethernet.....	16
Приложение 3 – лог загрузки SPI0 .....	20
Приложение 4 – лог загрузки из NAND flash .....	24

## Вступление.

После включения питания и выхода процессора из состояния «RESET», процессор AM1705 исполняет код загрузчика первого уровня D800K003 из ROM памяти или исполняет код напрямую из NOR flash памяти. Загрузчик первого уровня D800K003 требует чтобы загрузчик второго уровня или пользовательское приложение было в формате AIS. AIS - формат загрузочного образа, проприетарный загрузочный скрипт фирмы Texas Instruments, Inc. широко используемый в её продукции. AIS образ начинается с кодового слова (0x41504954), содержит набор AIS команд последовательно исполняемых загрузчиком и бинарный код загрузчика второго уровня или пользовательское приложение. Команда Jump & Close (J&C) помечает окончание AIS скрипта. Загрузчик первого уровня исполняет команды AIS образа (инициализация PLL, EMIFA, EMIFB, LPSC, PINMUX), загружает загрузчик второго уровня или пользовательское приложение в память и передает ему управление. Для генерации AIS образа из загрузчика второго уровня или из пользовательского приложения можно воспользоваться GUI приложением для операционной системы Windows “AISGen for D800K003” или консольным приложением HexAIS\_OMAP-L137.exe входящим в состав утилит “OMAP-L137\_FlashAndBootUtils\_2\_20” написанным на Mono (что позволяет выполнять его в ОС Windows или в ОС Linux).

## Первичная загрузка процессора AM1705

Процессор AM1705 определяет источник загрузки для загрузчика первого уровня D800K003 на основе состояния boot pins во время выхода из состояния “reset”, установленного на разъеме X4. Возможные конфигурации boot pins показаны в таблице 1.

Режим загрузки \ boot pins	UART1	SPI0 flash	NAND8	ED
Boot0	1	1	1	1
Boot1	1	0	1	1
Boot2	0	1	1	1
Boot3	1	X	X	0
Boot7	1	0	0	1
1 – джампер установлен, 0 – джампер снят, X – не учитывается при загрузке (джампер может быть снят или установлен)				

Таблица 1 Конфигурация boot pins в разъеме X4

Для отладочной платы TE-AM1705 доступны следующие источники : UART1, SPI0 flash, NAND8, возможные варианты загрузки показаны на рисунке 1.

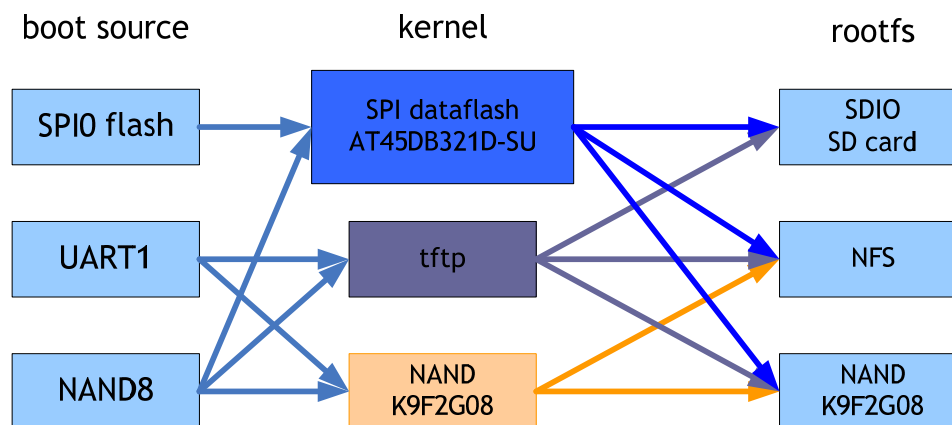


Рисунок 1 Возможные конфигурации загрузки платы TE-AM1705

Для успешной загрузки из SPI0 flash необходимо чтобы SPI dataflash AT45DB321 содержала AIS образу ядра Linux. Для успешной загрузки из NAND8 необходимо чтобы NAND память k9f2g08 содержала AIS образ загрузчика второго уровня u-boot или AIS образ ядра Linux. Для успешной загрузки по интерфейсу UART1 необходимо передать процессору AM1705 корректный AIS образ.

## Источник загрузки : UART1

Для восстановления или первичной загрузке платы TE-AM1705 используется интерфейс UART1 выведенный на разъем X3 USB mini-B, используется мост CP2102-GMR. Для включения загрузки через интерфейс UART1 необходимо выставить состояния boot pins на разъеме X4 как показано на рисунке 2.

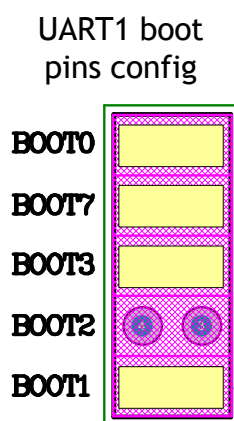


Рисунок 2 Выбор режима загрузки : UART1

Плата TE-AM1705 допускает питание через разъем X3 (USB mini-B) или X1 (штыревой разъем, 2.5 мм). При подключении платы TE-AM1705 к компьютеру и прохождения эnumерации в ОС Windows и в ОС Linux должно создаваться устройство Bridge USB-COM, пример показан на рисунке 3.

## OC Linux

```
[ 216.113381] usb 2-2: new full speed USB device using ohci_hcd and address 3
[ 216.701935] usbcore: registered new interface driver usbserial
[ 216.701960] USB Serial support registered for generic
[ 216.703639] usbcore: registered new interface driver usbserial_generic
[ 216.703639] usbserial: USB Serial Driver core
[ 216.716629] USB Serial support registered for cp210x
[ 216.717450] cp210x 2-2:1.0: cp210x converter detected
[ 217.112147] usb 2-2: reset full speed USB device using ohci_hcd and address 3
[ 217.607474] usb 2-2: cp210x converter now attached to ttyUSB0
[ 217.609277] usbcore: registered new interface driver cp210x
[ 217.609281] cp210x: v0.09:Silicon Labs CP210x RS232 serial adaptor driver
```

## OC Windows

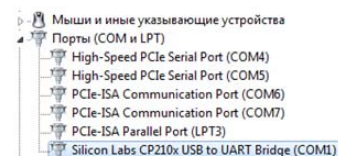
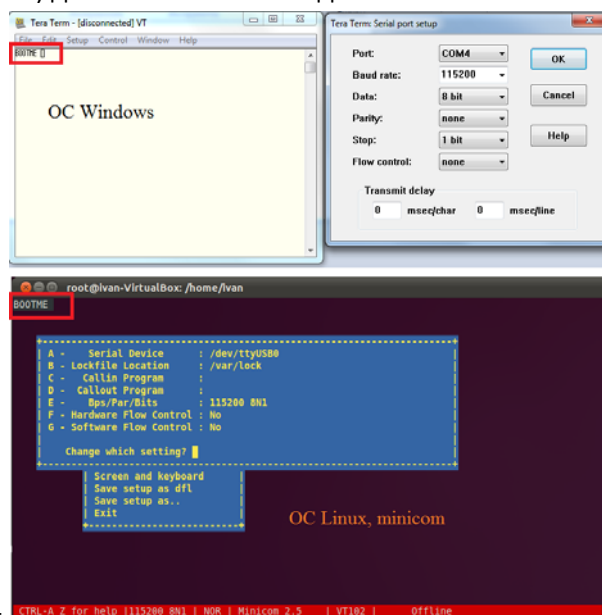


Рисунок 3 Подключение к компьютеру по интерфейсу UART1

При успешном подключении к компьютеру и правильной конфигурации boot pins в окне терминала после нажатия кнопки B1( расположена на плате рядом с кварцевым резонатором процессора AM1705) должна появиться надпись BOOTME как показано на рисунке



4. CTRL-A Z for help | 115200 Bn1 | NOR | Minicom 2.5 | VT102 | Offline

Рисунок 4 Проверка подключения по интерфейсу UART1

## Начальная загрузка ТЕ-AM1705

Для восстановления или начальной установки программного обеспечения используется AIS образ модифицированного u-boot (u-boot-03.20.00.12) входящего в состав BSP linux-03.20.00.12.tar.gz от Texas Instruments. Для создания AIS образа из u-boot в формате elf используются настройки утилиты AISGen показанные на рисунке 5.

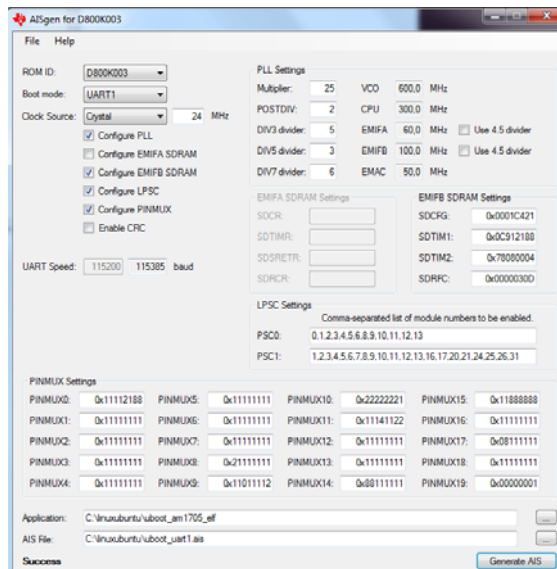


Рисунок 5 Настройки AISGen для первоначальной загрузки

Использование загрузчика второго уровня u-boot позволяет загрузить ОС Linux и выполнить запись ядра и корневой файловой системы в энергонезависимую память.

Этапы загрузка платы te-am1705 показаны на рисунке 6.

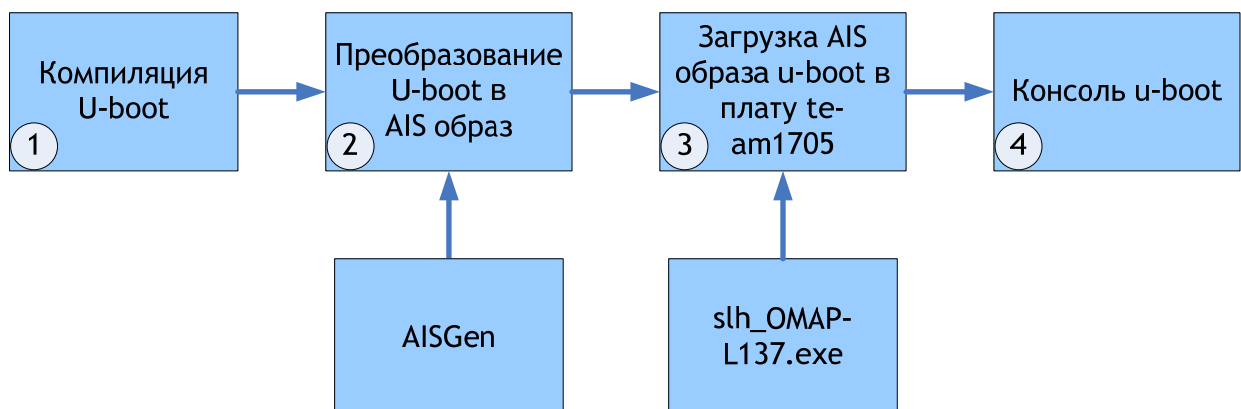


Рисунок 6 Порядок загрузки платы te-am1705 по интерфейсу UART1

Описание загрузки :

1. Конфигурирование и компиляция u-boot. Выполняется в ОС Linux.

#### Сброс предыдущих сборок

```
make distclean CROSS_COMPILE=/home/ivan/buildroot/buildroot/output/host/usr/bin/arm-unknown-linux-uclibcgnueabi-
```

#### Конфигурирование

```
make da830evm_config CROSS_COMPILE=/home/ivan/buildroot/buildroot/output/host/usr/bin/arm-unknown-linux-uclibcgnueabi-
```

#### Компиляция

```
make all CROSS_COMPILE=/home/ivan/buildroot/buildroot/output/host/usr/bin/arm-unknown-linux-uclibcgnueabi-
```

При успешной компиляции в директории uboot-03.20.00.12 появляется файл u-boot в формате elf :  
root@ivan-VirtualBox:/home/ivan/terra/1705# file uboot-03.20.00.12/u-boot

u-boot-03.20.00.12/u-boot: ELF 32-bit LSB executable, ARM, version 1 (SYSV), statically linked, not stripped

2. Преобразование u-boot из формата elf в формат AIS образа осуществляется при помощи GUI программы AISGen for D800K003 в ОС Windows, используются настройки показанные на рисунке 5.
3. Полученный AIS образ u-boot загружается в плату через интерфейс UART1 при помощи утилиты slh\_OMAP-L137.exe. Полный лог загрузки приведен в приложении 1.

**Командная строка для ОС Windows:**

```
slh_OMAP-L137.exe -v -waitForDevice -p COM1 u-boot_1705_test.bin
```

**Командная строка для ОС Linux:**

```
mono slh_OMAP-L137.exe -v -waitForDevice -p /dev/ttyUSB0 u-boot_1705_test.bin
```

4. При успешной загрузке можно открыть терминал и проверить правильность загрузки u-boot :

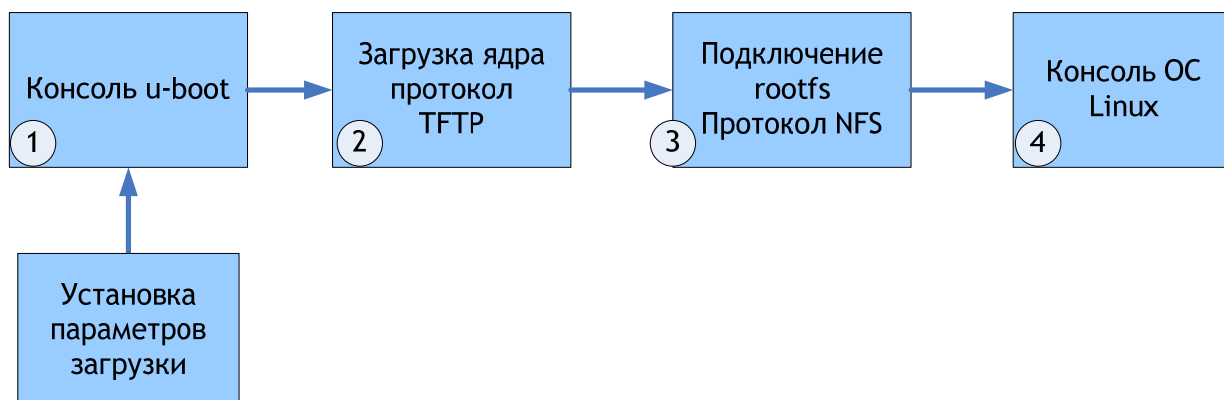
```
U-Boot > version
```

```
U-Boot 2009.11 (Mar 06 2012 - 00:42:53)
```

```
U-Boot >
```

## Загрузка ОС Linux с использованием интерфейса Ethernet

Варианты загрузки и запуска на плате te-am1705 ОС Linux показаны на рисунке 1, частный случай загрузки и запуска с использованием интерфейса Ethernet показан на рисунке 7.



**Рисунок 7 Порядок загрузки ОС Linux через интерфейс Ethernet**

Пояснения к рисунку 7:

1. В консоли u-boot необходимо корректно указать параметры сетевого интерфейса и командную строку ядра, пример:

Установка командной строки, которую передаст загрузчик ядру :

```
U-Boot >setenv bootargs consoleblank=0 mem=32M console=ttyS1,115200n8 rw rootwait earlyprintk  
root=/dev/nfs ip=192.168.0.10:192.168.0.5:192.168.0.1:255.255.255.0:uspd:eth0:bootp  
nfsroot=192.168.0.5:/home/ivan/nfs
```

Установка параметров сетевого интерфейса

```
U-Boot >setenv netmask 255.0.0.0
```

```
U-Boot >setenv ipaddr 192.168.0.10
```

```
U-Boot >setenv serverip 192.168.0.5
```

```
U-Boot >tftp ul_te_am1705
```

Запуск ядра Linux

```
U-Boot >bootm
```

- После успешной загрузки в консоли должно появиться приглашение входа в систему:

Полный лог загрузки приведен в приложении 2.

## Запись ОС Linux в энергонезависимую память

Для платы te-am1705 есть две возможности размещения ядра: SPI0 dataflash и NAND. Так как процессор am1705 не может одновременно работать с SD картой и NAND памятью для ядра требуются разные конфигурации для отдельной поддержки этих интерфейсов. Возможные конфигурации загрузки показаны на рисунке 1, но будут рассмотрены только два целесообразных варианта : SPI0 dataflash + SDIO и NAND flash.

### Запись ОС Linux в SPI dataflash + SDIO

В данном варианте загрузчик второго уровня uboot и ядро ОС Linux хранится в SPI dataflash, rootfs хранится на SD карте. Порядок записи показан на рисунке 8.

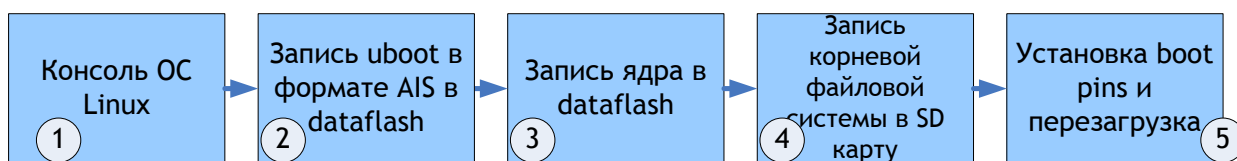


Рисунок 8 Ядро SPI dataflash, rootfs на внешней SD карте

Пояснения к рисунку 8

- Начальные условия : плата te-am1705 загружена с ядром поддерживающим spi dataflash и SD карту. Корневая файловая система подключена по протоколу NFS и содержит бинарный образ ядра, AIS образ u-boot и архив с корневой файловой системой. Пример бинарного образа находится в папке binaries\ul\_te\_am1705\_sdio BSP пакета платы te-am1705. Пример AIS образа u-boot находится в папке ais\_images\uboot\_spi0\_sdio.ais BSP пакета платы te-am1705. Пример архива содержащего корневую файловую систему для платы te-am1705 находится папке binaries/rootfs.tar BSP пакета платы te-am1705. Исходные коды загрузчика второго уровня u-boot находятся в папке BSP uboot-03.20.00.12\_sdio. Исходные коды ядра находятся в папке linux-3.2.1\_sdio.

Пример конфигурации утилиты AISGen для генерации AIS образа uboot показанные на рисунке 9.

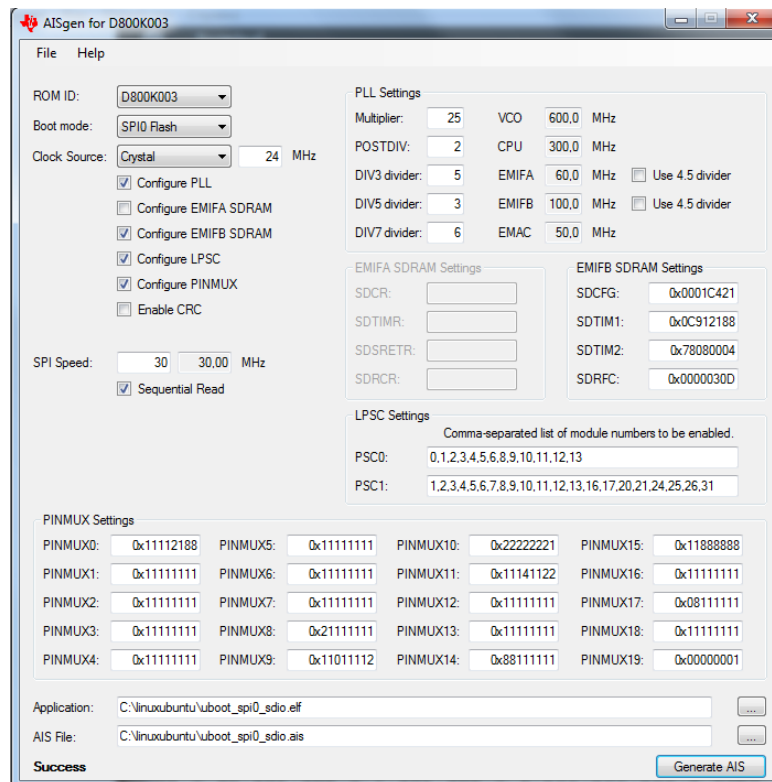


Рисунок 9 Настройки AISGen для загрузки через интерфейс SPI0

Пример командной строки для ядра ОС Linux, корневая файловая система подключается по протоколу NFS :

```
consoleblank=0 mem=32M console=ttyS1,115200n8 rw rootwait earlyprintk root=/dev/nfs
ip=192.168.0.10:192.168.0.5:192.168.0.1:255.255.255.0:uspd:eth0:bootp
nfsroot=192.168.0.5:/home/ivan/nfs
```

Проверить, что ядро корректно определило SPI dataflash и SD карту можно выполнив команду:

```
[root@buildroot ~]# cat /proc/partitions
major minor #blocks name
```

```
31 0 257 mtdblock0
31 1 16 mtdblock1
31 2 3949 mtdblock2
179 0 994816 mmcblk0
179 1 994691 mmcblk0p1
```

Кроме того в логе работы ОС должны присутствовать события :

```
mtd_dataflash spi0.0: AT45DB321x (4224 KBytes) pagesize 528 bytes (OTP)
Creating 3 MTD partitions on "at45db321d":
0x000000000000-0x000000040740 : "U-Boot"
0x000000040740-0x000000044940 : "U-Boot-Environment"
0x000000044940-0x0000000420000 : "Kernel"
```

```
mmc0: host does not support reading read-only switch. assuming write-enable.
mmc0: new high speed SD card at address 0002
mmcblk0: mmc0:0002 00000 971 MiB
mmcblk0: p1
```

2. Для записи AIS образа u-boot в SPI dataflash достаточно выполнить одну команду:

```
[root@buildroot ~]# flashcp -v /uboot_spi0_sdio.ais /dev/mtd0
Erasing blocks: 353/353 (100%)
```



Writing data: 181k/0k (100%)  
Verifying data: 181k/0k (100%)

3. Для записи ядра в SPI dataflash достаточно выполнить одну команду:  
[root@buildroot ~]# flashcp -v /ul\_te\_am1705\_sdio /dev/mtd2  
Erasing blocks: 3796/3796 (100%)  
Writing data: 1956k/0k (100%)  
Verifying data: 1956k/0k (100%)
4. Для записи корневой файловой системы в SD карту достаточно выполнить четыре команды:
  - a. На SD карте необходимо создать файловую систему командой mkfs:  
[root@buildroot /]# mkfs.ext3 /dev/mmcblk0p1
  - b. Вновь созданную файловую систему необходимо примонтировать выполнив команду:  
[root@buildroot /]# mount -t ext3 /dev/mmcblk0p1 /mnt
  - c. На примонтированную файловую систему нужно развернуть архив, содержащий rootfs:  
[root@buildroot /]# tar xvf /rootfs.tar -C /mnt
  - d. По окончании распаковки архива необходимо отмонтировать файловую систему:  
[root@buildroot /]# umount /mnt
5. На рисунке 10 показана конфигурация boot pins для загрузки через интерфейс SPI0, лог загрузки системы приведен в приложении 3.

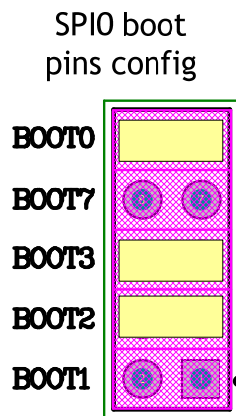


Рисунок 10 Выбор режима загрузки : SPI0

### Запись ОС Linux в NAND flash память.

В данном варианте uboot, ядро и корневая файловая система хранятся в NAND flash памяти. Общая схема записи ОС Linux в NAND flash память показана на рисунке 11.

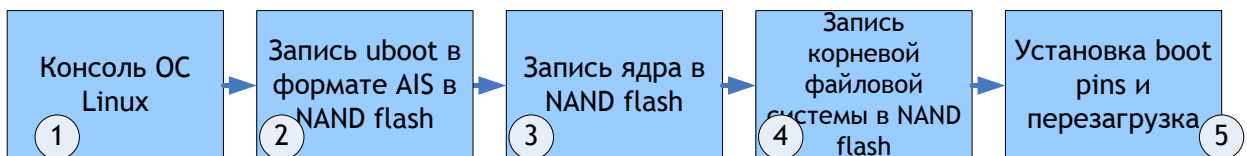


Рисунок 11 Запись ОС Linux в NAND flash память

Пояснения к рисунку 11:

1. Начальные условия : плата te-am1705 загружена с ядром поддерживающим NAND flash. Корневая файловая система подключена по протоколу NFS и содержит бинарный образ ядра, AIS образ u-boot и архив с корневой файловой системой. Пример бинарного образа ядра находится в папке binaries\

ul\_te\_am1705\_nand BSP пакета платы te-am1705. Пример AIS образа u-boot находится в папке ais\_images\ u-boot\_nand.ais BSP пакета платы te-am1705. Пример архива содержащего корневую файловую систему для платы te-am1705 находится папке binaries\rootfs.tar BSP пакета платы te-am1705. Исходные коды загрузчика второго уровня u-boot находятся в папке BSP uboot-03.20.00.12\_nand. Исходные коды ядра находятся в папке linux-3.2.1\_nand.

Пример конфигурации утилиты AISGen для генерации AIS образа uboot показан на рисунке 12.

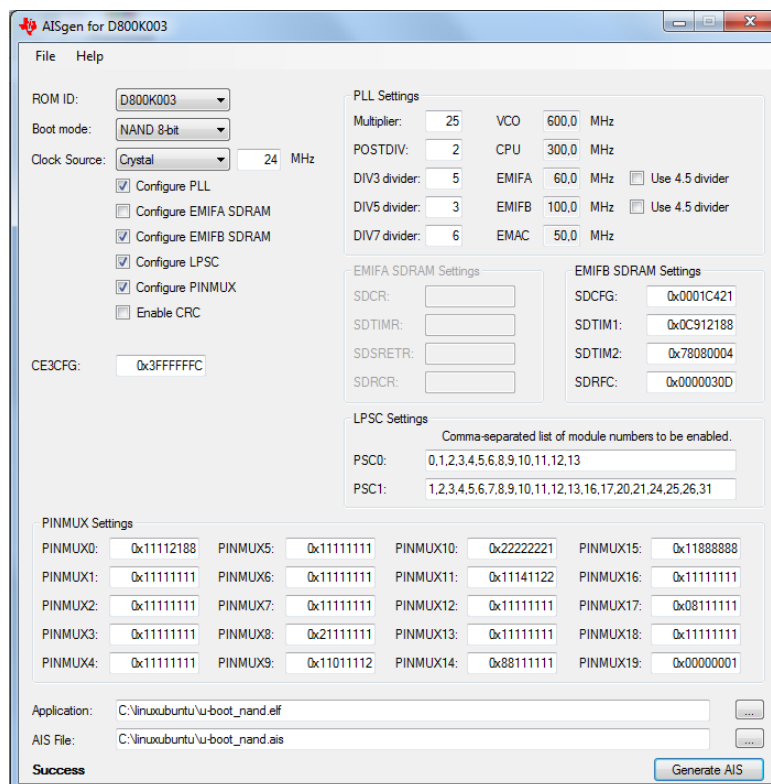


Рисунок 12 Настройки AISGen для загрузки из NAND flash

Пример командной строки для ядра ОС Linux, корневая файловая система подключается по протоколу NFS :

```
consoleblank=0 mem=32M console=ttyS1,115200n8 rw rootwait earlyprintk root=/dev/nfs
ip=192.168.0.10:192.168.0.5:192.168.0.1:255.255.255.0:uspd:eth0:bootp
nfsroot=192.168.0.5:/home/ivan/nfs
```

Проверить, что ядро корректно определило NAND flash можно выполнив команду:

```
[root@buildroot ~]# cat /proc/mtd
dev: size erasesize name
mtd0: 00020000 00020000 "bootloader"
mtd1: 00040000 00020000 "uboot"
mtd2: 00020000 00020000 "params"
mtd3: 00400000 00020000 "kernel"
mtd4: 0fa00000 00020000 "filesystem"
mtd5: 00180000 00020000 "bbt"
mtd6: 00420000 00000210 "spi0.0-AT45DB321x"
```

2. Для записи AIS образа u-boot в nand flash достаточно выполнить две команды:

```
[root@buildroot ~]# flash_erase /dev/mtd1 0 2
Erasing 128 Kibyte @ 20000 -- 100 % complete
[root@buildroot ~]# nandwrite -p /dev/mtd1 /u-boot_nand.ais
Writing data to block 0 at offset 0x0
Writing data to block 1 at offset 0x20000
```

```
[root@buildroot ~]#
```

3. Для записи ядра в NAND память достаточно выполнить две команды:

```
[root@buildroot ~]# flash_erase /dev/mtd3 0 32
```

Erasing 128 Kibyte @ 3e0000 -- 100 % complete

```
[root@buildroot ~]# nandwrite -p /dev/mtd3 /ul_te_am1705_nand
```

Writing data to block 0 at offset 0x0

.....

Writing data to block 16 at offset 0x200000

```
[root@buildroot ~]#
```

4. Для записи корневой файловой системы в NAND flash достаточно выполнить четыре команды:

- a. Подготовить NAND flash память стиранием содержимого раздела:

```
[root@buildroot ~]# flash_erase /dev/mtd4 0 2000
```

- b. Смонтировать раздел, тип файловой системы jffs2

```
[root@buildroot ~]# mount -t jffs2 -rw /dev/mtdblock4 /mnt
```

- c. Распаковать в точку монтирования архив, содержащий корневую файловую систему

```
[root@buildroot ~]# tar xvf /rootfs.tar -C /mnt
```

- d. Отмонтировать файловую систему

```
[root@buildroot ~]# umount /mnt
```

5. На рисунке 13 показана конфигурация boot pins для загрузки через интерфейс SPI0, лог загрузки системы приведен в приложении 4.

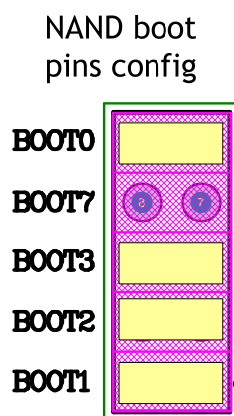


Рисунок 13 Конфигурация boot pins, источник NAND flash

## Приложение 1 – Лог запуска утилиты slh\_OMAP-L137.exe

mono slh\_OMAP-L137.exe -v -waitForDevice -p /dev/ttyUSB0 uboot\_uart1.ais

-----  
TI Serial Loader Host Program for OMAP-L137  
(C) 2012, Texas Instruments, Inc.  
Ver. 1.65  
-----

Attempting to connect to device /dev/ttyUSB0...  
Press any key to end this program at any time.

Entering AIS Parser

Waiting for the OMAP-L137...

(AIS Parse): Read magic word 0x41504954.  
(AIS Parse): Waiting for BOOTME... (power on or reset target now)  
(AIS Parse): BOOTME received!  
(AIS Parse): Performing Start-Word Sync...  
(AIS Parse): Performing Ping Opcode Sync...  
(AIS Parse): Processing command 0: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 1: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 2: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 3: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 4: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 5: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 6: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 7: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 8: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 9: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 10: 0x5853590D.

[illegible]

[illegible]

(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 47: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 48: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 49: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 50: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 51: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 52: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 53: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 54: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 55: 0x5853590D.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Executing function...  
(AIS Parse): Processing command 56: 0x58535901.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Loading section...  
(AIS Parse): Loaded 142080-Byte section to address 0xC0E80000.  
(AIS Parse): Processing command 57: 0x58535901.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Loading section...  
(AIS Parse): Loaded 29596-Byte section to address 0xC0EA2B00.  
(AIS Parse): Processing command 58: 0x58535901.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Loading section...  
(AIS Parse): Loaded 7124-Byte section to address 0xC0EA9E9C.  
(AIS Parse): Processing command 59: 0x58535901.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Loading section...  
(AIS Parse): Loaded 1568-Byte section to address 0xC0EABA70.  
(AIS Parse): Processing command 60: 0x58535906.  
(AIS Parse): Performing Opcode Sync...  
(AIS Parse): Performing jump and close...  
(AIS Parse): AIS complete. Jump to address 0xC0E80000.  
(AIS Parse): Waiting for DONE...  
(AIS Parse): Boot completed successfully.

## Приложение 2 – Лог загрузки с использованием интерфейса Ethernet

U-Boot 2009.11 (May 04 2012 - 22:17:23)

DRAM: 32 MB

In: serial

Out: serial

Err: serial

ARM Clock : 300000000 Hz

Net: Ethernet PHY: GENERIC @ 0x07

Hit any key to stop autoboot: 0

U-Boot > setenv bootargs consoleblank=0 mem=32M console=ttyS1,115200n8 rw rootwait earlyprintk

root=/dev/nfs ip=192.168.0.10:192.168.0.5:192.168.0.1:255.255.255.0:uspd:eth0:bootp

nfsroot=192.168.0.5:/home/ivan/nfs

U-Boot > setenv ipaddr 192.168.0.10

U-Boot > setenv serverip 192.168.0.5

U-Boot > tftp ul\_te\_am1705\_sdio

Using device

TFTP from server 192.168.0.5; our IP address is 192.168.0.10

Filename 'ul\_te\_am1705\_sdio'.

Load address: 0xc0700000

Loading: #####T #####T ##  
#####  
#####  
#####T #####T #####  
#####T #####  
#####T #####  
#####T #####  
##

done

Bytes transferred = 2003840 (1e9380 hex)

U-Boot > bootm

## Booting kernel from Legacy Image at c0700000 ...

Image Name: Linux-3.2.1

Image Type: ARM Linux Kernel Image (uncompressed)

Data Size: 2003776 Bytes = 1.9 MB

Load Address: c0008000

Entry Point: c0008000

Verifying Checksum ... OK

Loading Kernel Image ... OK

OK

Starting kernel ...

Linux version 3.2.1 (root@ivan-VirtualBox) (gcc version 4.5.3 (Buildroot 2012.02-git) ) #158 PREEMPT Fri

May 4 22:53:15 MSD 2012

CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00053177

CPU: VIVT data cache, VIVT instruction cache

Machine: DaVinci DA830/OMAP-L137/AM17x EVM

Memory policy: ECC disabled, Data cache writethrough



DaVinci da830/omap-l137 rev2.0 variant 0x9

Built 1 zonelists in Zone order, mobility grouping on. Total pages: 8128

Kernel command line: consoleblank=0 mem=32M console=ttyS1,115200n8 rw rootwait earlyprintk  
root=/dev/nfs ip=192.168.0.10:192.168.0.5:192.168.0.1:255.255.255.0:uspd:eth0:bootp  
nfsroot=192.168.0.5:/home/ivan/nfs

PID hash table entries: 128 (order: -3, 512 bytes)

Dentry cache hash table entries: 4096 (order: 2, 16384 bytes)

Inode-cache hash table entries: 2048 (order: 1, 8192 bytes)

Memory: 32MB = 32MB total

Memory: 28484k/28484k available, 4284k reserved, 0K highmem

Virtual kernel memory layout:

vector : 0xffff0000 - 0xffff1000 ( 4 kB)

fixmap : 0xfff00000 - 0xfffe0000 ( 896 kB)

vmalloc : 0xc2800000 - 0xfea00000 ( 962 MB)

lowmem : 0xc0000000 - 0xc2000000 ( 32 MB)

modules : 0xbf000000 - 0xc0000000 ( 16 MB)

.text : 0xc0008000 - 0xc0378c30 (3524 kB)

.init : 0xc0379000 - 0xc039c000 ( 140 kB)

.data : 0xc039c000 - 0xc03c0f60 ( 148 kB)

.bss : 0xc03c0f84 - 0xc03dddc0 ( 116 kB)

SLUB: Genslabs=13, HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1

NR\_IRQS:245

Console: colour dummy device 80x30

Calibrating delay loop... 148.88 BogoMIPS (lpj=744448)

pid\_max: default: 32768 minimum: 301

Mount-cache hash table entries: 512

CPU: Testing write buffer coherency: ok

devtmpfs: initialized

DaVinci: 128 gpio irqs

print\_constraints: dummy:

NET: Registered protocol family 16

MUX: initialized GPIO3\_11

MUX: Setting register GPIO3\_11

PINMUX11 (0x0000002c) = 0x11141122 -> 0x11181122

MUX: initialized MMCSD\_DAT\_0

MUX: Setting register MMCSD\_DAT\_0

PINMUX13 (0x00000034) = 0x11111111 -> 0x12111111

MUX: initialized MMCSD\_DAT\_1

MUX: Setting register MMCSD\_DAT\_1

PINMUX13 (0x00000034) = 0x12111111 -> 0x22111111

MUX: initialized MMCSD\_DAT\_2

MUX: Setting register MMCSD\_DAT\_2

PINMUX14 (0x00000038) = 0x88111111 -> 0x88111112

MUX: initialized MMCSD\_DAT\_3

MUX: Setting register MMCSD\_DAT\_3

PINMUX14 (0x00000038) = 0x88111112 -> 0x88111122

MUX: initialized MMCSD\_DAT\_4

MUX: Setting register MMCSD\_DAT\_4

PINMUX14 (0x00000038) = 0x88111122 -> 0x88112222

MUX: initialized MMCSD\_DAT\_5

MUX: Setting register MMCSD\_DAT\_5

PINMUX14 (0x00000038) = 0x88112222 -> 0x88112222

MUX: initialized MMCSD\_DAT\_6

MUX: Setting register MMCSD\_DAT\_6

PINMUX14 (0x00000038) = 0x88112222 -> 0x88122222  
MUX: initialized MMCSO\_DAT\_7  
MUX: Setting register MMCSO\_DAT\_7  
PINMUX14 (0x00000038) = 0x88122222 -> 0x88222222  
MUX: initialized MMCSO\_CLK  
MUX: Setting register MMCSO\_CLK  
PINMUX15 (0x0000003c) = 0x11888888 -> 0x21888888  
MUX: initialized MMCSO\_CMD  
MUX: Setting register MMCSO\_CMD  
PINMUX16 (0x00000040) = 0x11111111 -> 0x11111112  
MUX: initialized GPIO2\_1  
MUX: Setting register GPIO2\_1  
PINMUX17 (0x00000044) = 0x08111111 -> 0x88111111  
MUX: initialized GPIO4\_1  
MUX: Setting register GPIO4\_1  
PINMUX12 (0x00000030) = 0x11111111 -> 0x11811111  
MUX: initialized GPIO4\_0  
MUX: Setting register GPIO4\_0  
PINMUX12 (0x00000030) = 0x11811111 -> 0x11881111  
bio: create slab <bio-0> at 0  
SCSI subsystem initialized  
usbcore: registered new interface driver usbfs  
usbcore: registered new interface driver hub  
usbcore: registered new device driver usb  
Switching to clocksource timer0\_0  
musb-hdrc: version 6.0, ?dma?, otg (peripheral+host)  
Waiting for USB PHY clock good...  
musb-hdrc musb-hdrc: MUSB HDRC host driver  
musb-hdrc musb-hdrc: new USB bus registered, assigned bus number 1  
hub 1-0:1.0: USB hub found  
hub 1-0:1.0: 1 port detected  
musb-hdrc musb-hdrc: USB OTG mode controller at fee00000 using PIO, IRQ 58  
NET: Registered protocol family 2  
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)  
TCP established hash table entries: 1024 (order: 1, 8192 bytes)  
TCP bind hash table entries: 1024 (order: 0, 4096 bytes)  
TCP: Hash tables configured (established 1024 bind 1024)  
TCP reno registered  
UDP hash table entries: 256 (order: 0, 4096 bytes)  
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)  
NET: Registered protocol family 1  
RPC: Registered named UNIX socket transport module.  
RPC: Registered udp transport module.  
RPC: Registered tcp transport module.  
RPC: Registered tcp NFSv4.1 backchannel transport module.  
msgmni has been set to 55  
io scheduler noop registered (default)  
start plist test  
end plist test  
Serial: 8250/16550 driver, 3 ports, IRQ sharing disabled  
serial8250.0: ttyS0 at MMIO 0x1c42000 (irq = 25) is a 16550A  
serial8250.0: ttyS1 at MMIO 0x1d0c000 (irq = 53) is a 16550A  
console [ttyS1] enabled  
serial8250.0: ttyS2 at MMIO 0x1d0d000 (irq = 61) is a 16550A

brd: module loaded  
spi\_davinci spi\_davinci.0: DMA: supported  
spi\_davinci spi\_davinci.0: DMA: RX channel: 14, TX channel: 15, event queue: 0  
mtd\_dataflash spi0.0: AT45DB321x (4224 KBytes) pagesize 528 bytes (OTP)  
Creating 3 MTD partitions on "at45db321d":  
0x000000000000-0x0000000040740 : "U-Boot"  
0x0000000040740-0x0000000044940 : "U-Boot-Environment"  
0x0000000044940-0x00000000420000 : "Kernel"  
spi\_davinci spi\_davinci.0: Controller at 0xfec41000  
davinci\_mdio davinci\_mdio.0: davinci mdio revision 1.5  
davinci\_mdio davinci\_mdio.0: detected phy mask fffffff7  
davinci\_mdio.0: probed  
davinci\_mdio davinci\_mdio.0: phy[7]: device 0:07, driver Micrel KS8001 or KS8721  
ohci\_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver  
Initializing USB Mass Storage driver...  
usbcore: registered new interface driver usb-storage  
USB Mass Storage support registered.  
i2c /dev entries driver  
cpuidle: using governor ladder  
cpuidle: using governor menu  
davinci\_mmc davinci\_mmc.0: Using DMA, 4-bit mode  
TCP cubic registered  
NET: Registered protocol family 17  
console [netcon0] enabled  
netconsole: network logging started  
davinci\_emac davinci\_emac.1: using random MAC addr: a6:87:6d:76:52:10  
net eth0: no phy, defaulting to 100/full  
IP-Config: Complete:  
    device=eth0, addr=192.168.0.10, mask=255.255.255.0, gw=192.168.0.1,  
    host=uspd, domain=, nis-domain=(none),  
    bootserver=192.168.0.5, rootserver=192.168.0.5, rootpath=  
VFS: Mounted root (nfs filesystem) on device 0:14.  
devtmpfs: mounted  
Freeing init memory: 140K  
mmc0: host does not support reading read-only switch. assuming write-enable.  
mmc0: new high speed SD card at address 0002  
mmcblk0: mmc0:0002 00000 971 MiB  
    mmcblk0: p1  
Starting logging: OK  
Initializing random number generator... done.  
Starting network...  
ip: RTNETLINK answers: File exists  
Starting dropbear sshd: OK  
Starting atd: OK  
  
Welcome to Buildroot  
buildroot login:

## Приложение 3 – лог загрузки SPI0

U-Boot 2009.11 (May 04 2012 - 23:04:19)

DRAM: 32 MB

In: serial

Out: serial

Err: serial

ARM Clock : 300000000 Hz

Net: Ethernet PHY: GENERIC @ 0x07

Hit any key to stop autoboot: 0

16896 KiB AT45DB321D at 0:0 is now current device

## Booting kernel from Legacy Image at c0700000 ...

Image Name: Linux-3.2.1

Image Type: ARM Linux Kernel Image (uncompressed)

Data Size: 2003776 Bytes = 1.9 MB

Load Address: c0008000

Entry Point: c0008000

Verifying Checksum ... OK

Loading Kernel Image ... OK

OK

Starting kernel ...

Linux version 3.2.1 (root@ivan-VirtualBox) (gcc version 4.5.3 (Buildroot 2012.02-git) ) #158 PREEMPT Fri May 4 22:53:15 MSD 2012

CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00053177

CPU: VIVT data cache, VIVT instruction cache

Machine: DaVinci DA830/OMAP-L137/AM17x EVM

Memory policy: ECC disabled, Data cache writethrough

DaVinci da830/omap-l137 rev2.0 variant 0x9

Built 1 zonelists in Zone order, mobility grouping on. Total pages: 8128

Kernel command line: mem=32M console=ttyS1,115200n8 root=/dev/mmcblk0p1 rw rootwait ip=off

PID hash table entries: 128 (order: -3, 512 bytes)

Dentry cache hash table entries: 4096 (order: 2, 16384 bytes)

Inode-cache hash table entries: 2048 (order: 1, 8192 bytes)

Memory: 32MB = 32MB total

Memory: 28484k/28484k available, 4284k reserved, 0K highmem

Virtual kernel memory layout:

vector : 0xffff0000 - 0xffff1000 ( 4 kB)

fixmap : 0xfff00000 - 0xfffe0000 ( 896 kB)

vmalloc : 0xc2800000 - 0xfea00000 ( 962 MB)

lowmem : 0xc0000000 - 0xc2000000 ( 32 MB)

modules : 0xbf000000 - 0xc0000000 ( 16 MB)

.text : 0xc0008000 - 0xc0378c30 (3524 kB)

.init : 0xc0379000 - 0xc039c000 ( 140 kB)

.data : 0xc039c000 - 0xc03c0f60 ( 148 kB)

.bss : 0xc03c0f84 - 0xc03dddc0 ( 116 kB)

SLUB: Genslabs=13, HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1

NR\_IRQS:245

Console: colour dummy device 80x30

Calibrating delay loop... 148.88 BogoMIPS (lpj=744448)  
pid\_max: default: 32768 minimum: 301  
Mount-cache hash table entries: 512  
CPU: Testing write buffer coherency: ok  
devtmpfs: initialized  
DaVinci: 128 gpio irqs  
print\_constraints: dummy:  
NET: Registered protocol family 16  
MUX: initialized GPIO3\_11  
MUX: Setting register GPIO3\_11  
PINMUX11 (0x0000002c) = 0x11141122 -> 0x11181122  
MUX: initialized MMCSD\_DAT\_0  
MUX: Setting register MMCSD\_DAT\_0  
PINMUX13 (0x00000034) = 0x11111111 -> 0x12111111  
MUX: initialized MMCSD\_DAT\_1  
MUX: Setting register MMCSD\_DAT\_1  
PINMUX13 (0x00000034) = 0x12111111 -> 0x22111111  
MUX: initialized MMCSD\_DAT\_2  
MUX: Setting register MMCSD\_DAT\_2  
PINMUX14 (0x00000038) = 0x88111111 -> 0x88111112  
MUX: initialized MMCSD\_DAT\_3  
MUX: Setting register MMCSD\_DAT\_3  
PINMUX14 (0x00000038) = 0x88111112 -> 0x88111122  
MUX: initialized MMCSD\_DAT\_4  
MUX: Setting register MMCSD\_DAT\_4  
PINMUX14 (0x00000038) = 0x88111122 -> 0x88111222  
MUX: initialized MMCSD\_DAT\_5  
MUX: Setting register MMCSD\_DAT\_5  
PINMUX14 (0x00000038) = 0x88111222 -> 0x88112222  
MUX: initialized MMCSD\_DAT\_6  
MUX: Setting register MMCSD\_DAT\_6  
PINMUX14 (0x00000038) = 0x88112222 -> 0x88122222  
MUX: initialized MMCSD\_DAT\_7  
MUX: Setting register MMCSD\_DAT\_7  
PINMUX14 (0x00000038) = 0x88122222 -> 0x88222222  
MUX: initialized MMCSD\_CLK  
MUX: Setting register MMCSD\_CLK  
PINMUX15 (0x0000003c) = 0x11888888 -> 0x21888888  
MUX: initialized MMCSD\_CMD  
MUX: Setting register MMCSD\_CMD  
PINMUX16 (0x00000040) = 0x11111111 -> 0x11111112  
MUX: initialized GPIO2\_1  
MUX: Setting register GPIO2\_1  
PINMUX17 (0x00000044) = 0x08111111 -> 0x88111111  
MUX: initialized GPIO4\_1  
MUX: Setting register GPIO4\_1  
PINMUX12 (0x00000030) = 0x11111111 -> 0x11811111  
MUX: initialized GPIO4\_0  
MUX: Setting register GPIO4\_0  
PINMUX12 (0x00000030) = 0x11811111 -> 0x11881111  
bio: create slab <bio-0> at 0  
SCSI subsystem initialized  
usbcore: registered new interface driver usbfs  
usbcore: registered new interface driver hub

usbcore: registered new device driver usb  
Switching to clocksource timer0\_0  
musb-hdrc: version 6.0, ?dma?, otg (peripheral+host)  
Waiting for USB PHY clock good...  
musb-hdrc musb-hdrc: MUSB HDRC host driver  
musb-hdrc musb-hdrc: new USB bus registered, assigned bus number 1  
hub 1-0:1.0: USB hub found  
hub 1-0:1.0: 1 port detected  
musb-hdrc musb-hdrc: USB OTG mode controller at fee00000 using PIO, IRQ 58  
NET: Registered protocol family 2  
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)  
TCP established hash table entries: 1024 (order: 1, 8192 bytes)  
TCP bind hash table entries: 1024 (order: 0, 4096 bytes)  
TCP: Hash tables configured (established 1024 bind 1024)  
TCP reno registered  
UDP hash table entries: 256 (order: 0, 4096 bytes)  
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)  
NET: Registered protocol family 1  
RPC: Registered named UNIX socket transport module.  
RPC: Registered udp transport module.  
RPC: Registered tcp transport module.  
RPC: Registered tcp NFSv4.1 backchannel transport module.  
msgmni has been set to 55  
io scheduler noop registered (default)  
start plist test  
end plist test  
Serial: 8250/16550 driver, 3 ports, IRQ sharing disabled  
serial8250.0: ttyS0 at MMIO 0x1c42000 (irq = 25) is a 16550A  
serial8250.0: ttyS1 at MMIO 0x1d0c000 (irq = 53) is a 16550A  
console [ttyS1] enabled  
serial8250.0: ttyS2 at MMIO 0x1d0d000 (irq = 61) is a 16550A  
brd: module loaded  
spi\_davinci spi\_davinci.0: DMA: supported  
spi\_davinci spi\_davinci.0: DMA: RX channel: 14, TX channel: 15, event queue: 0  
mtd\_dataflash spi0.0: AT45DB321x (4224 KBytes) pagesize 528 bytes (OTP)  
Creating 3 MTD partitions on "at45db321d":  
0x000000000000-0x0000000040740 : "U-Boot"  
0x0000000040740-0x0000000044940 : "U-Boot-Environment"  
0x0000000044940-0x00000000420000 : "Kernel"  
spi\_davinci spi\_davinci.0: Controller at 0xfec41000  
davinci\_mdio davinci\_mdio.0: davinci mdio revision 1.5  
davinci\_mdio davinci\_mdio.0: detected phy mask fffffff7  
davinci\_mdio.0: probed  
davinci\_mdio davinci\_mdio.0: phy[7]: device 0:07, driver Micrel KS8001 or KS8721  
ohci\_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver  
Initializing USB Mass Storage driver...  
usbcore: registered new interface driver usb-storage  
USB Mass Storage support registered.  
i2c /dev entries driver  
cpuidle: using governor ladder  
cpuidle: using governor menu  
davinci\_mmc davinci\_mmc.0: Using DMA, 4-bit mode  
TCP cubic registered  
NET: Registered protocol family 17

console [netcon0] enabled  
netconsole: network logging started  
davinci\_emac davinci\_emac.1: using random MAC addr: 4a:ba:f2:41:97:e7  
Waiting for root device /dev/mmcblk0p1...  
mmc0: host does not support reading read-only switch. assuming write-enable.  
mmc0: new high speed SD card at address 0002  
mmcblk0: mmc0:0002 00000 971 MiB  
mmcblk0: p1  
kjournald starting. Commit interval 5 seconds  
EXT3-fs (mmcblk0p1): using internal journal  
EXT3-fs (mmcblk0p1): mounted filesystem with ordered data mode  
VFS: Mounted root (ext3 filesystem) on device 179:1.  
devtmpfs: mounted  
Freeing init memory: 140K  
Starting logging: OK  
Initializing random number generator... done.  
Starting network...  
Starting dropbear sshd: OK  
Starting atd: OK

Welcome to Buildroot  
buildroot login:

## Приложение 4 – лог загрузки из NAND flash

U-Boot 2009.11 (May 05 2012 - 00:34:00)

DRAM: 32 MB  
NAND: 256 MiB  
Bad block table not found for chip 0  
Bad block table found at page 130944, version 0x01  
nand\_bbt: ECC error while reading bad block table  
nand\_read\_bbt: Bad block at 0x0000049e0000  
nand\_read\_bbt: Bad block at 0x000006b40000  
nand\_read\_bbt: Bad block at 0x000007320000  
nand\_read\_bbt: Bad block at 0x00000dd60000  
nand\_read\_bbt: Bad block at 0x00000f5a0000  
Bad block table written to 0x00000ffe0000, version 0x01  
\*\*\* Warning - bad CRC or NAND, using default environment

In: serial  
Out: serial  
Err: serial  
ARM Clock : 300000000 Hz  
Net: Ethernet PHY: GENERIC @ 0x07

Hit any key to stop autoboot: 0

Loading from NAND 256MiB 3,3V 8-bit, offset 0x80000  
Image Name: Linux-3.2.1  
Image Type: ARM Linux Kernel Image (uncompressed)  
Data Size: 2193664 Bytes = 2.1 MB  
Load Address: c0008000  
Entry Point: c0008000

## Booting kernel from Legacy Image at c0700000 ...

Image Name: Linux-3.2.1  
Image Type: ARM Linux Kernel Image (uncompressed)  
Data Size: 2193664 Bytes = 2.1 MB  
Load Address: c0008000  
Entry Point: c0008000  
Verifying Checksum ... OK  
Loading Kernel Image ... OK

OK

Starting kernel ...

Linux version 3.2.1 (root@ivan-VirtualBox) (gcc version 4.5.3 (Buildroot 2012.02-git) ) #152 PREEMPT Sat  
May 5 02:02:34 MSD 2012  
CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00053177  
CPU: VIVT data cache, VIVT instruction cache  
Machine: DaVinci DA830/OMAP-L137/AM17x EVM  
Memory policy: ECC disabled, Data cache writethrough  
DaVinci da830/omap-l137 rev2.0 variant 0x9  
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 8128



Kernel command line: mem=32M consoleblank=0 console=ttyS1,115200n8 rw rootwait earlyprintk  
root=/dev/mtdblock4 rw rootfstype=jffs2 ip=off  
PID hash table entries: 128 (order: -3, 512 bytes)  
Dentry cache hash table entries: 4096 (order: 2, 16384 bytes)  
Inode-cache hash table entries: 2048 (order: 1, 8192 bytes)  
Memory: 32MB = 32MB total  
Memory: 27964k/27964k available, 4804k reserved, 0K highmem  
Virtual kernel memory layout:  
vector : 0xffff0000 - 0xffff1000 ( 4 kB)  
fixmap : 0xfff00000 - 0xfffe0000 ( 896 kB)  
vmalloc : 0xc2800000 - 0xfea00000 ( 962 MB)  
lowmem : 0xc0000000 - 0xc2000000 ( 32 MB)  
modules : 0xbf000000 - 0xc0000000 ( 16 MB)  
.text : 0xc0008000 - 0xc03c90a8 (3845 kB)  
.init : 0xc03ca000 - 0xc03f1000 ( 156 kB)  
.data : 0xc03f2000 - 0xc041a540 ( 162 kB)  
.bss : 0xc041a564 - 0xc045fd40 ( 278 kB)  
SLUB: Genslabs=13, HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1  
NR\_IRQS:245  
Console: colour dummy device 80x30  
Calibrating delay loop... 148.88 BogoMIPS (lpj=744448)  
pid\_max: default: 32768 minimum: 301  
Mount-cache hash table entries: 512  
CPU: Testing write buffer coherency: ok  
devtmpfs: initialized  
DaVinci: 128 gpio irqs  
print\_constraints: dummy:  
NET: Registered protocol family 16  
MUX: initialized GPIO3\_11  
MUX: Setting register GPIO3\_11  
PINMUX11 (0x0000002c) = 0x11141122 -> 0x11181122  
bio: create slab <bio-0> at 0  
SCSI subsystem initialized  
usbcore: registered new interface driver usbfs  
usbcore: registered new interface driver hub  
usbcore: registered new device driver usb  
Switching to clocksource timer0\_0  
musb-hdrc: version 6.0, ?dma?, otg (peripheral+host)  
Waiting for USB PHY clock good...  
musb-hdrc musb-hdrc: MUSB HDRC host driver  
musb-hdrc musb-hdrc: new USB bus registered, assigned bus number 1  
hub 1-0:1.0: USB hub found  
hub 1-0:1.0: 1 port detected  
musb-hdrc musb-hdrc: USB OTG mode controller at fee00000 using PIO, IRQ 58  
NET: Registered protocol family 2  
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)  
TCP established hash table entries: 1024 (order: 1, 8192 bytes)  
TCP bind hash table entries: 1024 (order: 0, 4096 bytes)  
TCP: Hash tables configured (established 1024 bind 1024)  
TCP reno registered  
UDP hash table entries: 256 (order: 0, 4096 bytes)  
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)  
NET: Registered protocol family 1  
RPC: Registered named UNIX socket transport module.

RPC: Registered udp transport module.  
RPC: Registered tcp transport module.  
RPC: Registered tcp NFSv4.1 backchannel transport module.  
JFFS2 version 2.2. (NAND) © 2001-2006 Red Hat, Inc.  
GFS2 installed  
msgmni has been set to 54  
io scheduler noop registered (default)  
start plist test  
end plist test  
Serial: 8250/16550 driver, 3 ports, IRQ sharing disabled  
serial8250.0: ttyS0 at MMIO 0x1c42000 (irq = 25) is a 16550A  
serial8250.0: ttyS1 at MMIO 0x1d0c000 (irq = 53) is a 16550A  
console [ttyS1] enabled  
serial8250.0: ttyS2 at MMIO 0x1d0d000 (irq = 61) is a 16550A  
brd: module loaded  
NAND device: Manufacturer ID: 0xec, Chip ID: 0xda (Samsung NAND 256MiB 3,3V 8-bit)  
Bad block table not found for chip 0  
Bad block table not found for chip 0  
Scanning device for bad blocks  
Bad eraseblock 591 at 0x0000049e0000  
Bad eraseblock 858 at 0x000006b40000  
Bad eraseblock 921 at 0x000007320000  
Bad eraseblock 1771 at 0x00000dd60000  
Bad eraseblock 1965 at 0x00000f5a0000  
Creating 6 MTD partitions on "davinci\_nand.1":  
0x000000000000-0x000000020000 : "bootloader"  
0x000000020000-0x000000060000 : "uboot"  
0x000000060000-0x000000080000 : "params"  
0x000000080000-0x000000480000 : "kernel"  
0x000000480000-0x00000fe80000 : "filesystem"  
0x00000fe80000-0x000010000000 : "bbt"  
davinci\_nand davinci\_nand.1: controller rev. 2.5  
spi\_davinci spi\_davinci.0: DMA: supported  
spi\_davinci spi\_davinci.0: DMA: RX channel: 14, TX channel: 15, event queue: 0  
mtd\_dataflash spi0.0: AT45DB321x (4224 KBytes) pagesize 528 bytes (OTP)  
spi\_davinci spi\_davinci.0: Controller at 0xfec41000  
davinci\_mdio davinci\_mdio.0: davinci mdio revision 1.5  
davinci\_mdio davinci\_mdio.0: detected phy mask fffffff7  
davinci\_mdio.0: probed  
davinci\_mdio davinci\_mdio.0: phy[7]: device 0:07, driver Micrel KS8001 or KS8721  
ohci\_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver  
Initializing USB Mass Storage driver...  
usbcore: registered new interface driver usb-storage  
USB Mass Storage support registered.  
i2c /dev entries driver  
cpuidle: using governor ladder  
cpuidle: using governor menu  
TCP cubic registered  
NET: Registered protocol family 17  
console [netcon0] enabled  
netconsole: network logging started  
davinci\_emac davinci\_emac.1: using random MAC addr: 32:0d:b9:77:ef:bf  
VFS: Mounted root (jffs2 filesystem) on device 31:4.  
devtmpfs: mounted

Freeing init memory: 156K  
Starting logging: OK  
Initializing random number generator... done.  
Starting network...  
Starting dropbear sshd: OK  
Starting atd: OK

Welcome to Buildroot  
buildroot login: