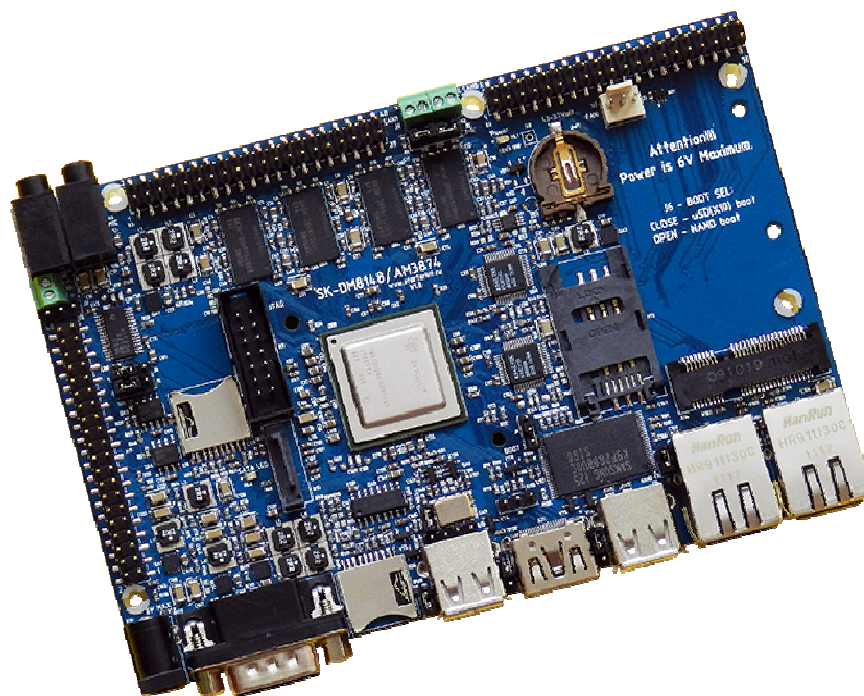


Отладочная плата SK-DM8148

Инструкция пользователя



SK-DM8148:

Texas Instruments TMS320DM8148 (ARM Cortex-A8 1000МГц + DSP 700МГц)
DDR3-800 1024Мбайт (128Мx64)
SLC NAND Flash 256Мбайт
2 x Ethernet 1000/100М, два отдельных порта или один порт на коммутатор
Mini PCIe разъем, SIM card держатель
SATA разъем
HDMI выход
Audio CODEC вход/выход
2 x uSD разъема
USB Host
RS232
CAN PHY
RTC (часы реального времени), держатель батареи 1220
Разъемы расширения
Система питания

Возможность прямого подключения:

SK-TFT1024x768TP-Plug или аналог – модуль расширения 8” 1024x768 LCD TFT панели с сенсорным экраном
SK-TFT1024x768-Plug или аналог – модуль расширения 8” 1024x768 LCD TFT панели
SK-ATM0700D4-Plug или аналог – модуль расширения 7” 800x480 LCD TFT панели с сенсорным экраном
SK-MI0430FT-Plug или аналог – модуль расширения 4” 480x272 LCD TFT панели с сенсорным экраном
SK-HDMI-Plug – плата расширения HDMI выхода
SK-SIMCOM-Plug – плата расширения GSM/GPS/3G модулей

Комплект поставки: отладочная плата SK-DM8148, RS232 кабель, ссылка для скачивания необходимых материалов

1. Общие характеристики

- Напряжение питания: 5-6В (центральная жила разъема), 6В максимум.
- Потребляемый ток платы - до 2А, рекомендуется блок питания с выходом 3А, следует учитывать потребление подключаемой внешней периферии.
- Габариты 144x96x20мм

2. Назначение джамперов

1-ый вывод перемычек и переключающих перемычек помечен квадратной контактной площадкой.

- J1, J2 позволяет подключить согласующий резистор к CAN линии
 - J3 линейный вход аудиокодека
 - J3 штыревой разъем для подключения линии CAN интерфейса
 - J4,J5 определяют функцию разъема X5, положение 1-2 – выход на наушники, 2-3 – линейный выход
 - J6 определяет источник загрузки процессора:
 - замкнут – загрузка с uSD карты (разъем X10)
 - разомкнут – загрузка с NAND Flash (J9 должен быть замкнут)
 - J7 позволяет управлять уровнем ID сигнала USB0
 - J8 композитный видеовыход
 - J9 управляет CS сигналом NAND flash, функция дублируется джампером J6, в последующих ревизиях платы J9 будет удален
 - J10 управляет подачей питающего напряжения к X18, **ВНИМАНИЕ!!!** В режиме работы порта как Device (например, при USB загрузке) должен быть разомкнут
 - J11 позволяет подключить питание к USB разъему X18 (USB1) минуя транзисторные каскады управления, следует учесть дальнейшее наличие в цепи J10
 - J11 позволяет подключить питание к USB разъему X19 (USB0) минуя транзисторные каскады управления
- По умолчанию замкнуты перемычки: J2, J7,J9,J10 и J4,J5 - положение 1-2

3. Начало работы

Перед началом работы убедитесь в положении перемычек (см. выше), так же следует ознакомиться со всеми материалами имеющих статус «Важная тема» или «Объявление» на форуме starterkit.ru в разделе “Отладочные платы > SK-DM8148”

Подключите RS232 кабель, идущий в комплекте, к COM порту PC (или USB-COM преобразователю), настройте терминальную программу на используемый COM порт с параметрами 115200 без управления потоком.

Подключите сетевой (Ethernet) кабель к разъему T1, настройте IP адрес сетевой карты PC в диапазоне 192.168.0.XXX.

Подключите питание, в терминальной программе появятся аналогичные сообщения:

```
U-Boot 2010.06 (Apr 06 2013 - 09:05:41)
```

```
TI8148-GP rev 2.1
```

```
ARM clk: 1000MHz
```

```
DDR clk: 400MHz
```

```
I2C: ready
```

DRAM: 2 GiB
NAND: HW ECC BCH8 Selected
256 MiB
MMC: OMAP SD/MMC: 0
*** Warning - bad CRC or NAND, using default environment

```
.;rrr;;.  
5#####2,  
A@hi;;r5;;r;rrSG@A,  
r@#i;;s222hG;rrsrrrrr;ri#@@r  
:@hr:r;SG3ssrr2r;rrsrsrsr;rh@:  
B@H;;rr;3Hs;rrr;sr;;rrsrsrsr;H@B  
@s:rrs;5#;;rrrr;r#@H:;rrsrsrsr:s@  
@;srs&X#9;r;r;;2@rrr;;rrsrsr;@  
@;rrsrs@MB#####r;rrsrsr;@  
G@r;rrsrsr;#X;SX25Ss#@#M@#9H9rrsrsrsr;r@G  
@9:srsrsr;2@;::.X@H:;rrsrsrsr:3@  
X@;rrsrsr;XAI;;&@#Bs:rrsrsrsr;@X  
@#;rsrsrsr;r2ir@##:rrsrsrsr:@  
@A:rrsrsr;:2@29@M@;:rrrsrsrsrsrs;H@  
@&;rsrsrsr;A@#####s:;;rrsrsrsr;G@  
@#:rrsrsr;G@5Hr25@#####9XG9s:rrrsrsrsr:#@  
M@;rsrsrsr;r@&;:S@M@Grr:rsrsr;@#  
:s;rsrsr:M#Msrr;#&#####H@5;rsrsr;s@,  
@:rrsrsr;S@rrsr;;r3MH@#M5,S@irsr:@  
@A:rrsrsr;rrrsrsr;:;@#r;rH@h;sr:H@  
;@9;rrsrsr;rrrsrsr;S@Hi@i;s;MX;rr:h@;  
r@B:rrrsrsr;sa@#i,i@h;r;S5;r:H@r  
,@@r;rrrsrsr;2BM3r;r:G@:rrr;r@e,  
B@Mr;rrrsrsr@S;;rrr;5M;rr;rM@H  
.@@i;rrrsrs2i;rrrr;r@M:i@e.  
.A@#5r;;r;;rrr;r:#AsM@H.  
&@MhXS5i5SX9B@G;  
ihM#####hs,
```

Net: <ethaddr> not set. Reading from E-fuse
Detected MACID:0:17:eb:82:79:ea
cpsw
Hit any key to stop autoboot: 3 2 1 0
Card did not respond to voltage select!
** Can't read from device 0 **

** Unable to use mmc 0:1 for fatload **

NAND read: device 0 offset 0xa0000, size 0x440000
4456448 bytes read: OK
Booting kernel from Legacy Image at 81000000 ...
Image Name: Linux-2.6.37
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 2957220 Bytes = 2.8 MiB
Load Address: 80008000
Entry Point: 80008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
Linux version 2.6.37 (user@DM8148-bld) (gcc version 4.3.3 (Sourcery G++ Lite 2009q1-203))
#101 Sat Apr 6 15:44:23 UTC 2013
CPU: ARMv7 Processor [413fc082] revision 2 (ARMv7), cr=10c53c7f
CPU: VIPT nonaliasing data cache, VIPT aliasing instruction cache
Machine: ti8148evm
reserved size = 52428800 at 0x0
FB: Reserving 52428800 bytes SDRAM for VRAM
Memory policy: ECC disabled, Data cache writeback
OMAP chip is TI8148 2.1
SRAM: Mapped pa 0x402f1000 to va 0xfe400000 size: 0xf000
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 79656
Kernel command line: console=ttyO0,115200n8 ubi.mtd=5 root=ubi0:nandfs rw rootfstype=ubifs
mem=364M@0x80000000 324M@0x9F900000 notifyk.vpssm3_sva=0xBF900000 noinitrd
PID hash table entries: 2048 (order: 1, 8192 bytes)
Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
Memory: 314MB = 314MB total
Memory: 311924k/311924k available, 60812k reserved, 0K highmem
Virtual kernel memory layout:
vector : 0xffff0000 - 0xffff1000 (4 kB)
fixmap : 0xffff0000 - 0xffffe000 (896 kB)
DMA : 0xffc00000 - 0xffe00000 (2 MB)
vmalloc : 0xd7000000 - 0xf8000000 (528 MB)
lowmem : 0xc0000000 - 0xd6c00000 (364 MB)

```

pkmap      : 0xbfe00000 - 0xc0000000   (  2 MB)
modules    : 0xbf000000 - 0xbfe00000   ( 14 MB)
  .init    : 0xc0008000 - 0xc003d000   ( 212 kB)
  .text    : 0xc003d000 - 0xc0592000   (5460 kB)
  .data    : 0xc0592000 - 0xc05de440   ( 306 kB)
SLUB: Genslabs=11, HWalig=64, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
NR_IRQS:407
IRQ: Found an INTC at 0xfa200000 (revision 5.0) with 128 interrupts
Total of 128 interrupts on 1 active controller
GPMC revision 6.0
Trying to install interrupt handler for IRQ400
Trying to install interrupt handler for IRQ401
Trying to install interrupt handler for IRQ402
Trying to install interrupt handler for IRQ403
Trying to install interrupt handler for IRQ404
Trying to install interrupt handler for IRQ405
Trying to install interrupt handler for IRQ406
Trying to install type control for IRQ407
Trying to set irq flags for IRQ407
OMAP clockevent source: GPTIMER1 at 20000000 Hz
Console: colour dummy device 80x30
Calibrating delay loop... 999.42 BogoMIPS (lpj=4997120)
pid_max: default: 32768 minimum: 301
Security Framework initialized
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
devtmpfs: initialized
omap_voltage_early_init: voltage driver support not added
regulator: core version 0.5
regulator: dummy:
NET: Registered protocol family 16
omap_voltage_domain_lookup: Voltage driver init not yet happened.Faulting!
omap_voltage_add_dev: VDD specified does not exist!
OMAP GPIO hardware version 0.1
OMAP GPIO hardware version 0.1
OMAP GPIO hardware version 0.1
OMAP GPIO hardware version 0.1
omap_mux_init: Add partition: #1: core, flags: 4
Debugfs: Only enabling/disabling deep sleep and wakeup timer is supported now
registered ti81xx_vpss device
registered ti81xx_vidout device
registered ti81xx_on-chip HDMI device
registered ti81xx_fb device
registered ti81xx_vin device
ti81xx_pcie: Invoking PCI BIOS...
ti81xx_pcie: Setting up Host Controller...
ti81xx_pcie: Register base mapped @0xd7020000
ti81xx_pcie: Starting PCI scan...
PCI: bus0: Fast back to back transfers enabled
bio: create slab <bio-0> at 0
vgaarb: loaded
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
USBSS revision 4ea2080b
registerd cppl-dma Intr @ IRQ 17
Cppi41 Init Done
omap_i2c omap_i2c.1: bus 1 rev4.0 at 100 kHz
Advanced Linux Sound Architecture Driver Version 1.0.23.
cfg80211: Calling CRDA to update world regulatory domain
Switching to clocksource gp timer
musb-hdrc: version 6.0, host, debug=0
musb-hdrc musb-hdrc.0: dma type: dma-cppi41
MUSB controller-0 revision 4ea20800
musb-hdrc musb-hdrc.0: MUSB HDRC host driver
musb-hdrc musb-hdrc.0: new USB bus registered, assigned bus number 1
usb usb1: New USB device found, idVendor=1d6b, idProduct=0002
usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
usb usb1: Product: MUSB HDRC host driver
usb usb1: Manufacturer: Linux 2.6.37 musb-hcd
usb usb1: SerialNumber: musb-hdrc.0
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
musb-hdrc musb-hdrc.0: USB Host mode controller at d701e000 using DMA, IRQ 18
musb-hdrc musb-hdrc.1: dma type: dma-cppi41
MUSB controller-1 revision 4ea20800
musb-hdrc musb-hdrc.1: MUSB HDRC host driver
musb-hdrc musb-hdrc.1: new USB bus registered, assigned bus number 2
usb usb2: New USB device found, idVendor=1d6b, idProduct=0002
usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
usb usb2: Product: MUSB HDRC host driver
usb usb2: Manufacturer: Linux 2.6.37 musb-hcd

```

```

usb usb2: SerialNumber: musb-hdrc.1
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
musb-hdrc musb-hdrc.1: USB Host mode controller at d7026800 using DMA, IRQ 19
NET: Registered protocol family 2
IP route cache hash table entries: 4096 (order: 2, 16384 bytes)
TCP established hash table entries: 16384 (order: 5, 131072 bytes)
TCP bind hash table entries: 16384 (order: 4, 65536 bytes)
TCP: Hash tables configured (established 16384 bind 16384)
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 udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
NetWinder Floating Point Emulator V0.97 (double precision)
PMU: registered new PMU device of type 0
omap-iommu omap-iommu.0: ducati registered
omap-iommu omap-iommu.1: sys registered
JFFS2 version 2.2. (NAND) B© 2001-2006 Red Hat, Inc.
msgmni has been set to 609
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
Serial: 8250/16550 driver, 4 ports, IRQ sharing enabled
omap_uart.0: ttyO0 at MMIO 0x48020000 (irq = 72) is a OMAP UART0
console [ttyO0] enabled
omap_uart.1: ttyO1 at MMIO 0x48022000 (irq = 73) is a OMAP UART1
omap_uart.2: ttyO2 at MMIO 0x48024000 (irq = 74) is a OMAP UART2
omap_uart.3: ttyO3 at MMIO 0x481a6000 (irq = 44) is a OMAP UART3
omap_uart.4: ttyO4 at MMIO 0x481a8000 (irq = 45) is a OMAP UART4
omap_uart.5: ttyO5 at MMIO 0x481aa000 (irq = 46) is a OMAP UART5
brd: module loaded
loop: module loaded
ahci ahci.0: forcing PORTS_IMPL to 0x1
ahci ahci.0: AHCI 0001.0300 32 slots 1 ports 3 Gbps 0x1 impl platform mode
ahci ahci.0: flags: ncq sntf pm led clo only pmp pio slum part ccc apst
scsi0 : ahci_platform
ata1: SATA max UDMA/133 mmio [mem 0x4a140000-0x4a150fff] port 0x100 irq 16
omap2-nand driver initializing
NAND device: Manufacturer ID: 0xec, Chip ID: 0xda (Samsung )
Creating 6 MTD partitions on "omap2-nand.0":
0x0000000000000-0x000000020000 : "U-Boot-min"
0x0000000200000-0x000000080000 : "U-Boot"
0x0000000800000-0x0000000a0000 : "U-Boot Env"
0x0000000a00000-0x00000004e0000 : "Kernel"
0x00000004e0000-0x000001000000 : "Safe kernel (initramfs)"
0x0000010000000-0x0000010000000 : "File System"
UBI: attaching mtd5 to ubi0
UBI: physical eraseblock size: 131072 bytes (128 KiB)
UBI: logical eraseblock size: 129024 bytes
UBI: smallest flash I/O unit: 2048
UBI: sub-page size: 512
UBI: VID header offset: 512 (aligned 512)
UBI: data offset: 2048
ata1: SATA link down (SStatus 0 SControl 300)
UBI: max. sequence number: 839
UBI: attached mtd5 to ubi0
UBI: MTD device name: "File System"
UBI: MTD device size: 240 MiB
UBI: number of good PEBs: 1916
UBI: number of bad PEBs: 4
UBI: number of corrupted PEBs: 0
UBI: max. allowed volumes: 128
UBI: wear-leveling threshold: 4096
UBI: number of internal volumes: 1
UBI: number of user volumes: 1
UBI: available PEBs: 267
UBI: total number of reserved PEBs: 1649
UBI: number of PEBs reserved for bad PEB handling: 19
UBI: max/mean erase counter: 1/0
UBI: image sequence number: 380229539
UBI: background thread "ubi_bgt0d" started, PID 46
davinci_mdio davinci_mdio.0: davinci mdio revision 1.6
davinci_mdio davinci_mdio.0: detected phy mask ffffffff5
davinci_mdio.0: probed
davinci_mdio davinci_mdio.0: phy[1]: device 0:01, driver RTL821x Gigabit Ethernet
davinci_mdio davinci_mdio.0: phy[3]: device 0:03, driver RTL821x Gigabit Ethernet
CAN device driver interface
CAN bus driver for Bosch D_CAN controller 1.0
d_can d_can.0: d_can device registered (irq=52, irq_obj=53)
usbcore: registered new interface driver cdc_ether

```

```

usbcore: registered new interface driver dm9601
usbcore: registered new interface driver cdc_acm
cdc_acm: v0.26:USB Abstract Control Model driver for USB modems and ISDN adapters
usbcore: registered new interface driver uas
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
mice: PS/2 mouse device common for all mice
ads7846 spi1.0: touchscreen, irq 207
input: ADS7846 Touchscreen as /devices/platform/omap2_mcspi.1/spi1.0/input/input0
qt602240_ts 1-004a: __qt602240_read_reg: i2c transfer failed
qt602240_ts: probe of 1-004a failed with error -5
omap_rtc omap_rtc: rtc core: registered omap_rtc as rtc0
rtc-pcf8563 1-0051: chip found, driver version 0.4.3
rtc-pcf8563 1-0051: rtc core: registered rtc-pcf8563 as rtc1
i2c /dev entries driver
Linux video capture interface: v2.00
usbcore: registered new interface driver uvcvideo
USB Video Class driver (v1.0.0)
OMAP Watchdog Timer Rev 0x00: initial timeout 60 sec
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
notify_init : notify drivercreated for remote proc id 2 at physical Address 0xbf900000
usbcore: registered new interface driver snd-usb-audio
AIC23 Audio Codec 0.1
asoc: tlv320aic23-hifi <-> davinci-mcasp.0 mapping ok
asoc: HDMI-DAI-CODEC <-> hdmi-dai mapping ok
ALSA device list:
  #0: TI81XX EVM
TCP cubic registered
NET: Registered protocol family 17
can: controller area network core (rev 20090105 abi 8)
NET: Registered protocol family 29
can: raw protocol (rev 20090105)
can: broadcast manager protocol (rev 20090105 t)
lib80211: common routines for IEEE802.11 drivers
Registering the dns_resolver key type
VFP support v0.3: implementor 41 architecture 3 part 30 variant c rev 3
omap_voltage_late_init: Voltage driver support not added
Power Management for TI81XX.
Detected MACID=0:17:eb:82:79:ea
rtc-pcf8563 1-0051: low voltage detected, date/time is not reliable.
rtc-pcf8563 1-0051: retrieved date/time is not valid.
rtc-pcf8563 1-0051: hctosys: invalid date/time
UBIFS: mounted UBI device 0, volume 0, name "nandfs"
UBIFS: file system size: 208373760 bytes (203490 KiB, 198 MiB, 1615 LEBs)
UBIFS: journal size: 10450944 bytes (10206 KiB, 9 MiB, 81 LEBs)
UBIFS: media format: w4/r0 (latest is w4/r0)
UBIFS: default compressor: lzo
UBIFS: reserved for root: 4952683 bytes (4836 KiB)
VFS: Mounted root (ubifs filesystem) on device 0:14.
devtmpfs: mounted
Freeing init memory: 212K
Init: rm: can't remove '/bin/ip': No such file or directory
Starting logging: OK
Initializing random number generator... done.
Starting network...
CPSW phy found : id is : 0x1cc912
CPSW phy found : id is : 0x1cc912
d_can d_can.0: can0: setting CAN BT = 0x1c09
Starting dropbear sshd: NET: Registered protocol family 10
ADDRCONF(NETDEV_UP): eth0: link is not ready
OK
Starting sshd: OK
Starting wi-fi network ...
Error for wireless request "Set Mode" (8B06) :
  SET failed on device wlan0 ; No such device.
Error for wireless request "Set ESSID" (8B1A) :
  SET failed on device wlan0 ; No such device.
ifconfig: SIOCSIFADDR: No such device
Loading HDVICP2 Firmware
DM814X prcm_config_app version: 2.0.0.1
Doing PRCM settings...
PRCM for IVHD0 is in Progress, Please wait.....
BW Phy Addr : 0x48180600 Data : 0x00000001
AW Phy Addr : 0x48180600 Data : 0x00000002
Phy Addr : 0x48180c04 Data : 0x00000037
BW Phy Addr : 0x48180620 Data : 0x00070000
AW Phy Addr : 0x48180620 Data : 0x00050002
BW Phy Addr : 0x48180624 Data : 0x00030000
AW Phy Addr : 0x48180624 Data : 0x00010002
Phy Addr : 0x48180600 Data : 0x00000102
BW Phy Addr : 0x48180c10 Data : 0x00000007

```

```

AW Phy Addr : 0x48180c10 Data : 0x00000003
Phy Addr : 0x48180c14 Data : 0x00000004
BW Phy Addr : 0x58088000 Data : 0xb356bf06
AW Phy Addr : 0x58088000 Data : 0xeaffffff
BW Phy Addr : 0x58098000 Data : 0xbe8a28cc
AW Phy Addr : 0x58098000 Data : 0xeaffffff
BW Phy Addr : 0x48180c10 Data : 0x00000003
AW Phy Addr : 0x48180c10 Data : 0x00000000
Phy Addr : 0x48180c14 Data : 0x00000007
PRCM for IVHD0 is Done Successfully
PRCM Initialization completed
SysLink version : 2.20.02.20
SysLink module created on Date:Mar 29 2013 Time:15:54:10
PHY: 0:01 - Link is Up - 0/Half
PHY: 0:01 - Link is Down
PHY: 0:03 - Link is Up - 1000/Full
ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
FIRMWARE: I2cInit will be done by M3
FIRMWARE: Memory map bin file not passed
Usage : firmware_loader <Processor Id> <Location of Firmware> <start|stop> [-mmap
<memory_map_file>] [-i2c <0|1>]
===Mandatory arguments===
<Processor Id>          0: DSP, 1: Video-M3, 2: Vpss-M3
<Location of Firmware> firmware binary file
<start|stop>           to start/stop the firmware
===Optional arguments===
-mmap                  input memory map bin file name
-i2c                   0: i2c init not done by M3, 1(default): i2c init done by M3
FIRMWARE: isI2cInitRequiredOnM3: 1
FIRMWARE: Default memory configuration is used
Firmware Loader debugging not configured
Default FL_DEBUG: warning
Allowed FL_DEBUG levels: error, warning, info, debug, log
MemCfg: DCMM (Dynamically Configurable Memory Map) Version : 2.1.2.1
FIRMWARE: 1 start Successful
Loading HDVPSS Firmware
FIRMWARE: I2cInit will be done by M3
FIRMWARE: Memory map bin file not passed
Usage : firmware_loader <Processor Id> <Location of Firmware> <start|stop> [-mmap
<memory_map_file>] [-i2c <0|1>]
===Mandatory arguments===
<Processor Id>          0: DSP, 1: Video-M3, 2: Vpss-M3
<Location of Firmware> firmware binary file
<start|stop>           to start/stop the firmware
===Optional arguments===
-mmap                  input memory map bin file name
-i2c                   0: i2c init not done by M3, 1(default): i2c init done by M3
FIRMWARE: isI2cInitRequiredOnM3: 1
FIRMWARE: Default memory configuration is used
Firmware Loader debugging not configured
Default FL_DEBUG: warning
Allowed FL_DEBUG levels: error, warning, info, debug, log
MemCfg: DCMM (Dynamically Configurable Memory Map) Version : 2.1.2.1
FIRMWARE: 2 start Successful
VPSS_FVID2: M3 firmware version 0x1000145 is newer,driver may not work properly.
Configuring fb0 to LCD
HDMI W1 rev 4.0
HDMI CEC Spec version 1.2
I2C No Ack
Console: switching to colour frame buffer device 240x67

Welcome to SK-DM8148 development board!
SK-DM8148 login:

```

Что означает, что система успешно загрузилась и готова к работе.

Для входа в консоль введите имя пользователя root, пароль не требуется (других пользователей в системе нет), после чего имеете полный консольный доступ к системе. Так же можно подключиться с помощью Telnet, FTP, HTTP, SSH, сетевой адрес платы 192.168.0.136. При подключении-отключении USB, uSD карт памяти, они будут автоматически монтироваться-размонтироваться в системе.

3.1 Подключение модулей расширения

Модули расширения необходимо подключать при отсутствующем напряжении питания.

Видеосистема штатного ядра настроена:

/dev/fb0 – HDMI выход, разрешение 1920x1080, консоль tty1

/dev/fb1 – LCD выход (X4), разрешение 800x480, сенсорный ввод

Внимание! При старте системы анализируется наличие HDMI подключения, при его отсутствии не подгрузятся необходимые драйвера, поэтому HDMI соединение необходимо устанавливать перед запуском системы.

3.1.1 Подключение SK-ATM0700D4-Plug к разъему X4

Штатное ядро настроено на этот модуль расширения, никаких дополнительных настроек не требуется. Перед началом работы необходимо откалибровать сенсорный экран – запустите `ts_calibrate`.

3.1.2 Подключение SK-TFT1024x768TP-Plug к разъему X4

Для конфигурирования ядра необходимо выполнить команду «`config_lcd 1024x768`» и перезагрузить систему. Перед началом работы необходимо откалибровать сенсорный экран – запустите `ts_calibrate`.

3.1.3 Подключение SK-TFT1024x768-Plug к разъему X4

Для конфигурирования ядра необходимо выполнить команду «`config_lcd 1024x768`» и перезагрузить систему.

3.1.4 Подключение SK-MI0430FT-Plug к разъему X4

Для конфигурирования ядра необходимо выполнить команду «`config_lcd 480x272`» и перезагрузить систему. Перед началом работы необходимо откалибровать сенсорный экран – запустите `ts_calibrate`.

4. Состав ОС Linux

Ядро 2.6.37, включая драйвера:

- Ethernet
- NAND flash
- SATA
- PCIe
- SD-card
- USB-host
- USB-gadget
- I2C
- SPI
- UART
- RTC
- CAN
- WatchDog
- Graphic system
- TP ADS7846
- ...

5. Способы загрузки и содержимое корневой файловой системы

TMS320DM8148 подразумевает различные возможные источники загрузки, на плате предусмотрено два - NAND flash и uSD

5.1. NAND flash

В штатной поставке система сконфигурирована на загрузку из NAND flash. Перемычка J6 должна быть разомкнута

NAND flash разбита на 6 частей:

- 1) 0-0x20000 – первичный загрузчик
- 2) 0x20000-0x80000 – вторичный загрузчик u-boot
- 3) 0x80000-0xa0000 – область хранения переменных окружения вторичного загрузчика
- 4) 0xa0000-0x4e0000 – ядро Linux
- 5) 0x4e0000-0x1000000 – ядро Linux со встроенной корневой файловой системой
- 6) 0x1000000-0x10000000 – область корневой файловой системы

5.2. uSD

Предварительно карта памяти должна быть подготовлена скриптом «/home/user/src/scripts/sd_prepare.sh» находящимся на виртуальной машине, который создаст требуемую таблицу разделов, отформатирует разделы и скопирует необходимые файлы. **Внимание!** Скрипт sd_prepare.sh уничтожит все имеющиеся до этого на карте данные.

Первый раздел «BOOT» (FAT) содержит исполняемые файлы загрузчиков и ядро Linux, а так же скрипт подготовки системы.

Второй раздел «ROOTFS» (EXT3) содержит корневую файловую систему.

Подготовленную карту необходимо вставить в разъем X10, замкнуть J6, после чего включить питание. Файлы копируемые скриптом sd_prepare.sh в первую очередь предназначены для подготовки системы к начальному состоянию, ядро запускаемое с uSD карты запускает скрипт «system_prepare_dm8148», который стрирует и форматирует разделы NAND flash, а потом копирует файлы.

6. Корневая файловая система

Корневая файловая система (ФС), в поставляемом варианте платы, хранится на NAND flash и монтируется во время загрузки, поэтому, следует внимательней относиться к изменениям в скриптах загрузки системы.

Корневая ФС содержит набор базовых приложений (большинство из которых являются реализацией мультифункционального приложения BusyBox), содержит:

- HTTPD – сервер HTTP
- FTPD – сервер FTP
- Telnetd – сервер Telnet
- TFTP – утилита приема-передачи файлов по TFTP протоколу
- Z-modem утилиты (для обмена файлами через COM порт)
- Microcom – терминальная программа
- TS-lib – набор утилит для операций с сенсорной панелью
- Memtester – тест памяти

- Mplayer – медиа-проигрыватель
- MC – файловый менеджер
- Qt
- ...

На случай аварии корневой файловой системы, предусмотрен режим аварийной загрузки, для его активации необходимо прервать загрузку в U-boot (нажав на любую клавишу) и выполнить команду «run safe_boot». Загрузится образ системы, в котором корневая ФС расположена в памяти и можно будет приступить к ремонту основной корневой ФС, например, запустить скрипт «rootfs_update_tftp», в результате работы которого будет заново отформатирован второй раздел NAND flash, скопирован с TFTP сервера и распакован архив корневой ФС.

7. Виртуальная машина VMware

Для сборки ядра и корневой ФС используется виртуальная машина VMware с установленной ОС Ubuntu, в состав которой входят все исходные тексты, компилятор и утилиты для сборки (toolchain), скрипты. Так же в виртуальной машине установлены и настроены сервисы для удобства взаимодействия с «материнской» ОС и отладочной платой: SSH, FTP, TFTP, Samba (доступ к файлам сети Microsoft).

Разархивируйте файл “ SK-DM8148_linux_build_machine.rar”, установите VMware-player или VMware, откройте и проект виртуальной машины.

Для работы необходимо настроить сетевые интерфейсы (появляющиеся после установки VMware), присвоив им описываемые ниже IP адреса:

Eth0 (Bridget) с адресом 192.168.0.2, задуман для взаимодействия с платой, для загрузки образов по TFTP ... Т.е. для нормальной работы, потребуется присвоить IP адрес PC сетевой карты (к которой подключается отладочная плата) 192.168.0.1

Eth1 (Host-only) с адресом 192.168.2.2, задуман для взаимодействия с PC (т.к. Bridget интерфейс отключается при физически отключенном сетевом кабеле, в случае с прямым подключением платы к PC), в частности, для возможности копирования файлов из виртуальной системы. В свойствах сетевых устройств, этому виртуальному адаптеру нужно присвоить IP 192.168.2.1

После правильной настройки (и с подключенной платой) должны успешно проходить PING с PC по адресам 192.168.2.2, 192.168.0.2, 192.168.0.136.

После загрузки виртуальной машины ее не обязательно выключать, достаточно будет нажать кнопку паузы и во время следующего сеанса работы не придется ждать загрузки виртуальной ОС, но при этом, в некоторых случаях, нужно следить за системными временем, особенно при копировании новых файлов (имеющих более позднюю дату создания относительно системы) для сборки.

По умолчанию, в системе присутствует один пользователь:

- user, пароль 123456

Суперпользователя root в виртуальной машине нет, для действий с его привилегиями можно пользоваться командами su или sudo.

После входа можно переключиться на консоль (Ctrl+Alt+F(1-6)) и запустить MidnightComander (mc).

Основная рабочая папка /home/user/src, содержимое:

- kernel – содержит ядро системы, в корневой директории ядра лежат скрипты:
menuconfig.sh – служит для конфигурирования ядра системы штатной загрузки
build.sh – служит для сборки ядра
- rootfs/nand_fs – содержит корневую систему штатной загрузки собираемую с помощью buildroot, скрипт **build_system** собирает корневую ФС и копирует ее архив в /home/user/tftp папку. Для конфигурирования содержимого необходимо выполнить «make menuconfig», для сборки достаточно выполнить скрипт build.sh
- rootfs/safe_fs – содержит корневую систему для safe загрузки, Для конфигурирования содержимого необходимо выполнить «make menuconfig», для сборки достаточно выполнить скрипт build.sh
- u-boot – содержит загрузчик системы, в корневой директории лежат скрипты:
build.sh – собирает u-boot для загрузки системы с NAND flash и копирует исполняемые файлы в /home/user/tftp папку

7.1.1. Примеры

Обновление ядра Linux, для этого:

- запускаем скрипт /home/user/src/kernel/linux-2.6.XX/build.sh
- включаем/перезагружаем плату с подключенным Ethernet (разъем T1) и RS232 кабелями
- прерываем в u-boot процесс загрузки нажатием любой клавиши
- выполняем “run system_update”

Загрузка ядра Linux с TFTP сервера, для этого:

- включаем/перезагружаем плату с подключенным Ethernet (разъем T1) и RS232 кабелями
- прерываем в u-boot процесс загрузки нажатием любой клавиши
- выполняем “run tftp_boot”

8. Общий принцип работы системы

После подачи питания (перезагрузки), процессор запускает загрузчик (находится во внутренней не перепрограммируемой ROM), который анализируя уровни конфигурационных сигналов, определяет источник загрузки.

Поскольку внешняя DDR3 (или любая другая память - не инициализирована), первое запускаемое приложение должно быть загрузчиком. Это приложение (первичный загрузчик u-boot) в первую очередь должен проинициализировать внешнюю память (например, правильно настроить параметры DDR3), скопировать исполняемое приложение из внешней Flash памяти во внешнюю DDR3 память и передать ему управление.

Вторичный загрузчик u-boot обладает обширными возможностями, например, он умеет копировать файлы по TFTP, SD и т.п., поддерживает целый набор команд и режимов. В переменных окружения u-boot есть команда запуска, в которой указано, по какому адресу NAND flash следует прочитать образ ядра, куда этот образ памяти записать и по какому адресу запустить. Следующие сообщения консоли иллюстрируют этот процесс:

```
NAND read: device 0 offset 0xa0000, size 0x440000
4456448 bytes read: OK
## Booting kernel from Legacy Image at 81000000 ...
```

Перед запуском ядра Linux, оно первым делом проверяет контрольную сумму собственного архива и распаковывает себя (в случае safe загрузки, ядро включает в себя еще корневую ФС), иллюстрация:

```
## Booting kernel from Legacy Image at 81000000 ...
Image Name:   Linux-2.6.37
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    2957220 Bytes = 2.8 MiB
Load Address: 80008000
Entry Point:  80008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
Starting kernel ...
```

Далее идет инициализация всей системы, драйверов, файловых систем, после чего управление передается скриптам начального запуска.