

## Процессорный модуль SK-iMX6S-SODIMM, SK-iMX6D-SODIMM

Инструкция пользователя при совместном  
использовании с платой SK-iMX6-MB-SODIMM



**SK-iMX6S-SODIMM:**

Freescale iMX6 Solo (ARM Cortex-A9 1000МГц)

DDR3-800 512Мбайт

eMMC Flash 4Гбайт

100/10M Ethernet

I/O: HDMI, LVDS, PCIe, Ethernet, USB, SD/MMC, SPDIF, CSI, Ext BUS, LCD, SPI, UART, CAN, PWM, I2C, GPIO

**SK-iMX6D-SODIMM:**

Freescale iMX6 **Dual** (ARM Dual Core Cortex-A9 1000МГц)

DDR3-**1066 1024**Мбайт

eMMC Flash 4Гбайт

100/10M Ethernet

I/O: HDMI, LVDS, PCIe, **SATA**, Ethernet, USB, SD/MMC, SPDIF, CSI, Ext BUS, LCD, SPI, UART, CAN, PWM, I2C, GPIO

**SK-iMX6-MB-SODIMM, возможность прямого подключения:**

SK-MI0430FT-Plug или аналог – модуль расширения LCD TFT 4,3” панелей

EV-ATM5HD-Plug или аналог – модуль расширения LCD TFT 5” панелей

SK-ATM0700D4-Plug или аналог – модуль расширения LCD TFT 7” панелей

SK-TFT1024x768TP-Plug или аналог – модуль расширения LCD TFT 8” панелей

SK-SIMCOM-Plug – модуль расширения GSM/GPS/3G модулей

SK-VideoADC-Plug – модуль расширения видео захвата

## Общие характеристики

### SK-iMX6S/D-SODIMM:

- Напряжение питания: 5В
- Потребляемый ток до 1А
- Габариты 67.6x58мм

### SK-iMX6-MB-SODIMM:

- Напряжение питания: 5В (питающее напряжение – центральный штырь разъема)
- Потребляемый ток (зависит от подключения внешних модулей) до 2А
- Габариты 117x124мм

## 1. Назначение джамперов

### SK-iMX6-MB-SODIMM:

- J1 определяет источник загрузки модуля, замкнут – USB OTG, разомкнут – eMMC
- J2 линейный вход звукового кодека
- J3 позволяет подключать согласующий резистор для CAN шины
- J4-J5 позволяет выбирать подключение разъема X14 к выходу на наушники или линейному выходу звукового кодека
- J6 позволяет подключать питание к USB OTG (X19) интерфейсу, **может быть замкнут только в случае подключения Device устройства к этому разъему!!!**

Первый контакт разъемов и переключающих переключателей отмечен квадратной контактной площадкой.

**По умолчанию замкнуты переключатели: J3, J4-J5 – положение 1-2**

## 2. Начало работы

Перед началом работы убедитесь в положении переключателей (см. выше), так же следует ознакомиться со всеми материалами имеющих статус «Важная тема» или «Объявление» на форуме starterkit.ru в разделе “ОЕМ модули > SK-iMX6S/D-SODIMM”.

Подключите RS232 кабель, настройте терминальную программу на используемый COM порт с параметрами 115200 без управления потоком.

Подключите сетевой (Ethernet) кабель, настройте IP адрес сетевой карты PC в диапазоне 192.168.0.XXX (любой кроме 192.168.0.136).

При необходимости, подключите SK-ATM0700D4-Plug к разъему X12, USB WiFi RTL8192 модуль к любому из USB разъемов.

Подключите питание, в терминальной программе появятся следующие сообщения:

```
U-Boot 2013.10 (Dec 05 2013 - 01:37:28)
CPU: Freescale i.MX6SOLO rev1.1 at 792 MHz
Reset cause: POR
Board: Starterkit
DRAM: 512 MiB
Setup enet clock OK
MMC: FSL_SDHC: 0, FSL_SDHC: 1
*** Warning - bad CRC, using default environment
In: serial
Out: serial
Err: serial
Net: FEC
Hit any key to stop autoboot: 0
mmc0(part 0) is current device
8451860 bytes read in 461 ms (17.5 MiB/s)
## Booting kernel from Legacy Image at 12000000 ...
Image Name: Linux
Image Type: ARM Linux Kernel Image (uncompressed)
```

```

Data Size:      8451796 Bytes = 8.1 MiB
Load Address: 10008000
Entry Point:   10008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
Starting kernel ...
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 3.1.2 (user@iMX6-bld) (gcc version 4.7.1 20120402 (prerelease) (crosstool-NG
linaro-1.13.1-2012.04-20120426 - Linaro GCC 2012.04) ) #128 SMP PREEMPT Sun Dec 8 09:02:21 UTC
2013
CPU: ARMv7 Processor [412fc09a] revision 10 (ARMv7), cr=10c53c7d
CPU: VIPT nonaliasing data cache, VIPT aliasing instruction cache
Machine: Starterkit i.MX 6Solo Board
Memory policy: ECC disabled, Data cache writealloc
CPU identified as i.MX6DL/SOLO, silicon rev 1.1
PERCPU: Embedded 7 pages/cpu @8bc06000 s6080 r8192 d14400 u32768
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 89088
Kernel command line: console=ttyMXC0,115200 consoleblank=0 dmfc=3 root=/dev/mmchblk0p1 rootwait
video=mxcfb0:dev=ldb,SK-ATM0704,if=RGB24 video=mxcfb1:dev=ldb,SK-ATM0704,if=RGB24 ldb=sep0
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)
allocated 2097152 bytes of page_cgroup
please try 'cgroup_disable=memory' option if you don't want memory cgroups
Memory: 352MB = 352MB total
Memory: 345116k/345116k available, 179172k reserved, 0K highmem
Virtual kernel memory layout:
   vector : 0xffff0000 - 0xffff1000   (   4 kB)
   fixmap : 0xffff0000 - 0xffffe000   ( 896 kB)
   DMA    : 0xf4600000 - 0xffe00000   ( 184 MB)
   vmalloc : 0xa0800000 - 0xf2000000 (1304 MB)
   lowmem : 0x80000000 - 0xa0000000   ( 512 MB)
   pkmap  : 0x7fe00000 - 0x80000000   (   2 MB)
   modules : 0x7f000000 - 0x7fe00000   (  14 MB)
     .init : 0x80008000 - 0x8003e000   ( 216 kB)
     .text : 0x8003e000 - 0x807abdb4   (7608 kB)
     .data : 0x807ac000 - 0x808176b0   ( 430 kB)
     .bss : 0x808176d4 - 0x80887364   ( 448 kB)
SLUB: Genslabs=13, HWalig=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
Preemptible hierarchical RCU implementation.
NR_IRQS:624
MXC GPIO hardware
sched clock: 32 bits at 3000kHz, resolution 333ns, wraps every 1431655ms
Set periph_clk's parent to pll2_pfd_400M!
arm_max_freq=1GHz
MXC_Early serial console at MMIO 0x2020000 (options '115200')
bootconsole [ttyMXC0] enabled
Console: colour dummy device 80x30
Calibrating delay loop... 1581.05 BogoMIPS (lpj=7905280)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 512
Initializing cgroup subsys cpuacct
Initializing cgroup subsys memory
Initializing cgroup subsys devices
Initializing cgroup subsys freezer
Initializing cgroup subsys blkio
Initializing cgroup subsys perf_event
CPU: Testing write buffer coherency: ok
hw perfevents: enabled with ARMv7 Cortex-A9 PMU driver, 7 counters available
Brought up 1 CPUs
SMP: Total of 1 processors activated (1581.05 BogoMIPS).
devtmpfs: initialized
print_constraints: dummy:
NET: Registered protocol family 16
print_constraints: vddpu: 725 <--> 1300 mV at 1150 mV fast normal
print_constraints: vddcore: 725 <--> 1300 mV at 1150 mV fast normal
print_constraints: vddsoc: 725 <--> 1300 mV at 1200 mV fast normal
print_constraints: vdd2p5: 2000 <--> 2775 mV at 2400 mV fast normal
print_constraints: vddlpl: 800 <--> 1400 mV at 1100 mV fast normal
print_constraints: vdd3p0: 2625 <--> 3400 mV at 3000 mV fast normal
hw-breakpoint: found 6 breakpoint and 1 watchpoint registers.
hw-breakpoint: 1 breakpoint(s) reserved for watchpoint single-step.
hw-breakpoint: maximum watchpoint size is 4 bytes.
L310 cache controller enabled
12x0: 16 ways, CACHE_ID 0x410000c8, AUX_CTRL 0x02050000, Cache size: 524288 B
bio: create slab <bio-0> at 0
mxs-dma mxs-dma-apbh: initialized
print_constraints: vmmc: 3300 mV
SCSI subsystem initialized
spi_imx imx6q-ecspi.3: probed
spi_imx imx6q-ecspi.0: probed
usbcore: registered new interface driver usbfs

```

```

usbcore: registered new interface driver hub
usbcore: registered new device driver usb
Freescale USB OTG Driver loaded, $Revision: 1.55 $
imx-ipuv3 imx-ipuv3.0: IPU DMFC ONLY-DP HIGH RESOLUTION: 5B(0~3), 5F(4~7)
Advanced Linux Sound Architecture Driver Version 1.0.24.
Bluetooth: Core ver 2.16
NET: Registered protocol family 31
Bluetooth: HCI device and connection manager initialized
Bluetooth: HCI socket layer initialized
Bluetooth: L2CAP socket layer initialized
Bluetooth: SCO socket layer initialized
i2c-core: driver [max17135] using legacy suspend method
i2c-core: driver [max17135] using legacy resume method
Switching to clocksource mxc_timer1
cfg80211: Calling CRDA to update world regulatory domain
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: 5, 196608 bytes)
TCP: Hash tables configured (established 16384 bind 16384)
TCP reno registered
UDP hash table entries: 256 (order: 1, 8192 bytes)
UDP-Lite hash table entries: 256 (order: 1, 8192 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.
_regulator_get: etb supply vcore not found, using dummy regulator
_regulator_get: etm.0 supply vcore not found, using dummy regulator
Static Power Management for Freescale i.MX6
wait mode is enabled for i.MX6
cpaddr = a0820000 suspend_iram_base=a08b4000
PM driver module loaded
iMX6 PCIe PCIe RC mode imx_pcie_pltfrm_probe entering.
PCIe: imx_pcie_pltfrm_probe start link up.
link up failed, DB_R0:0x00cbee00, DB_R1:0x08200000!
IMX PCIe port: link down!
IMX usb wakeup probe
add wake up source irq 75
IMX usb wakeup probe
cpu regulator mode:ldo_enable
i.MXC CPU frequency driver
nfs4filelayout init: NFSv4 File Layout Driver Registering...
NTFS driver 2.1.30 [Flags: R/W].
fuse init (API version 7.16)
msgmni has been set to 674
alg: No test for stdrng (krng)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
mxc_sdc_fb mxc_sdc_fb.0: register mxc display driver ldb
_regulator_get: get() with no identifier
fbcv: Invalid input parameters
Console: switching to colour frame buffer device 100x30
mxc_sdc_fb mxc_sdc_fb.1: register mxc display driver ldb
fbcv: Invalid input parameters
imx-sdma imx-sdma: loaded firmware 1.1
imx-sdma imx-sdma: initialized
Serial: IMX driver
imx-uart.0: ttyMXC0 at MMIO 0x2020000 (irq = 58) is a IMX
console [ttyMXC0] enabled, bootconsole disabled
console [ttyMXC0] enabled, bootconsole disabled
imx-uart.1: ttyMXC1 at MMIO 0x21e8000 (irq = 59) is a IMX
imx-uart.2: ttyMXC2 at MMIO 0x21ec000 (irq = 60) is a IMX
loop: module loaded
GPMI NAND driver registered. (IMX)
vcan: Virtual CAN interface driver
CAN device driver interface
flexcan netdevice driver
flexcan imx6q-flexcan.1: device registered (reg_base=a0958000, irq=143)
flexcan imx6q-flexcan.0: device registered (reg_base=a0960000, irq=142)
FEC Ethernet Driver
fec_enet_mii bus: probed
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
fsl-ehci fsl-ehci.0: Freescale On-Chip EHCI Host Controller
fsl-ehci fsl-ehci.0: new USB bus registered, assigned bus number 1
fsl-ehci fsl-ehci.0: irq 75, io base 0x02184000
fsl-ehci fsl-ehci.0: USB 2.0 started, EHCI 1.00
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
add wake up source irq 72
fsl-ehci fsl-ehci.1: Freescale On-Chip EHCI Host Controller

```

```

fsl-ehci fsl-ehci.1: new USB bus registered, assigned bus number 2
fsl-ehci fsl-ehci.1: irq 72, io base 0x02184200
fsl-ehci fsl-ehci.1: USB 2.0 started, EHCI 1.00
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
ARC USBOTG Device Controller driver (1 August 2005)
mousedev: PS/2 mouse device common for all mice
_regulator_get: spi3.0 supply vcc not found, using dummy regulator
ads7846 spi3.0: touchscreen, irq 362
input: ADS7846 Touchscreen as /devices/platform/imx6q-ecspi.3/spi_master/spi3/spi3.0/input/input0
_regulator_get: spi0.0 supply vcc not found, using dummy regulator
ads7846 spi0.0: touchscreen, irq 347
input: ADS7846 Touchscreen as /devices/platform/imx6q-ecspi.0/spi_master/spi0/spi0.0/input/input1
i2c-core: driver [isl29023] using legacy suspend method
i2c-core: driver [isl29023] using legacy resume method
snvs rtc snvs_rtc.0: rtc core: registered snvs_rtc as rtc0
i2c /dev entries driver
Linux video capture interface: v2.00
mxc_v4l2_output mxc_v4l2_output.0: V4L2 device registered as videol6
mxc_v4l2_output mxc_v4l2_output.0: V4L2 device registered as videol7
mxc_v4l2_output mxc_v4l2_output.0: V4L2 device registered as videol8
imx2-wdt imx2-wdt.0: IMX2+ Watchdog Timer enabled. timeout=60s (nowayout=1)
Bluetooth: Generic Bluetooth USB driver ver 0.6
usbcore: registered new interface driver btusb
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
_regulator_get: sdhci-esdhc-imx.2 supply vmmc not found, using dummy regulator
mmc0: SDHCI controller on platform [sdhci-esdhc-imx.2] using ADMA
mmc1: SDHCI controller on platform [sdhci-esdhc-imx.0] using ADMA
mxc_vdoa mxc_vdoa: i.MX Video Data Order Adapter (VDOA) driver probed
VPU initialized
mxc_asrc registered
Galcore version 4.6.9.6622
Thermal calibration data is 0x58f50d5f
Thermal sensor with ratio = 184
Anatop Thermal registered as thermal_zone0
anatop_thermal_probe: default cooling device is cpufreq!
HDMI CEC initialized
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
usbcore: registered new interface driver snd-usb-audio
mxc_hdmi_soc mxc_hdmi_soc.0: MXC HDMI Audio
imx-hdmi-soc-dai imx-hdmi-soc-dai.0: Failed: Load HDMI-video first.
AIC23 Audio Codec 0.1
asoc: tlv320aic23-hifi <-> imx-ssi.1 mapping ok
Initialize HDMI-audio failed. Load HDMI-video first!
ALSA device list:
  #0: sgtl5000-audio
NET: Registered protocol family 26
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)
usb 2-1: new high speed USB device number 2 using fsl-ehci
mmc0: new high speed DDR MMC card at address 0001
mmcblk0: mmc0:0001 M4G1FB 3.64 GiB
mmcblk0boot0: mmc0:0001 M4G1FB partition 1 1.00 MiB
can: broadcast manager protocol (rev 20090105 t)
mmcblk0boot1: mmc0:0001 M4G1FB partition 2 1.00 MiB
Bluetooth: RFCOMM TTY layer initialized
Bluetooth: RFCOMM socket layer initialized
Bluetooth: RFCOMM ver 1.11
Bluetooth: BNEP (Ethernet Emulation) ver 1.3
Bluetooth: BNEP filters: protocol multicast
mmcblk0: p1
Bluetooth: HIDP (Human Interface Emulation) ver 1.2
  mmcblk0boot1: unknown partition table
lib80211: common routines for IEEE802.11 drivers
VFP support v0.3: implementor 41 architecture 3 part 30 variant 9 rev 4
Bus freq driver module loaded
  mmcblk0boot0: unknown partition table
Bus freq driver Enabled
mxc_dvfs_core_probe
DVFS driver module loaded
snvs_rtc snvs_rtc.0: setting system clock to 1970-01-01 00:00:00 UTC (0)
EXT4-fs (mmcblk0p1): mounting ext3 file system using the ext4 subsystem
hub 2-1:1.0: USB hub found
hub 2-1:1.0: 4 ports detected
mmc1: new high speed SDHC card at address 0001

```

```
mmcblk1: mmc1:0001 00000 14.6 GiB
mmcblk1: p1 p2
EXT4-fs (mmcblk0p1): mounted filesystem with ordered data mode. Opts: (null)
VFS: Mounted root (ext3 filesystem) on device 179:1.
devtmpfs: mounted
Freeing init memory: 216K
EXT4-fs (mmcblk0p1): re-mounted. Opts: user_xattr,barrier=1,nodelalloc,data=ordered
usb 2-1.1: new high speed USB device number 3 using fsl-ehci
Starting logging: OK
Populating /dev using udev: udevd[79]: starting version 182
done
Starting watchdog...
Initializing random number generator... done.
Starting system message bus: DBG sensor data is at 7f001c88
done
Starting network...
eth0: Freescale FEC PHY driver [SMSC LAN8710/LAN8720] (mii_bus:phy_addr=1:00, irq=-1)
flexcan imx6q-flexcan.1: writing ctrl=0x0e312085
Starting dropbear sshd: OK
Starting lighttpd: OK
Starting sshd: OK
Starting vsftpd: OK
Starting SMB services: usbcore: registered new interface driver rtl8192cu
ip_tables: (C) 2000-2006 Netfilter Core Team
nf_contrack version 0.5.0 (5395 buckets, 21580 max)
PHY: 1:00 - Link is Up - 100/Full
done
Starting NMB services: done
Simple mixer control 'Mic',0
  Capabilities: cswitch cswitch-joined cswitch-exclusive
  Capture exclusive group: 0
  Capture channels: Mono
  Mono: Capture [on]
Simple mixer control 'Mic Booster',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
  Mono: Playback [on]

Welcome to Buildroot
SK-iMX6S-SODIMM login:
```

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

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

Если был подключен USB WiFi RTL8188/8192cu модуль, активируется беспроводная точка доступа iMX6\_AP, пароль для доступа 87654321.

Для настройки часов реального времени необходимо настроить дату-время и сохранить настройки:

```
# date -s 2013.12.09-22:24:10
Mon Dec  9 22:24:10 UTC 2013
# hwclock -w
```

В системе настроен сторожевой таймер (watchdog) на интервал 20с.

## 2.1. Подключение модулей расширения

Процессор iMX6S позволяет подключить два независимых устройства отображения, что продемонстрировано в штатной настройке системы. Настройки передаются через аргументы запуска ядра и хранятся в переменных окружения загрузчика u-boot.

Следует учитывать, что для большей производительности системы необходимо отключать не используемые источники.

Видеосистема штатного ядра:

```
/dev/fb0 – LVDS0 выход (X12), разрешение 800x480, консоль /dev/tty1
/dev/fb1 – прозрачное наложение на fb0
```

/dev/fb2 – LVDS1 выход (X11), разрешение 800x480

SK-ATM0700D4-Plug – разъем X12

Модуль расширения предварительно необходимо настроить на использование LVDS интерфейса – замкнуть J10, подключить модуль через разъем X10.

В штатной поставке ядро сконфигурировано на использование данного модуля расширения – /dev/fb0, в качестве контроллера TP включен ADS7843 (или аналог). Для демонстрации необходимо выполнить скрипт «qt\_fb0\_ev0\_test», во время работы которого сначала выполнится калибровка сенсорного экрана, а затем запустится пример работы Qt приложения «Affine».

SK-ATM0700D4-Plug – разъем X11

Модуль расширения предварительно необходимо настроить на использование LVDS интерфейса – разомкнуть все переключки располагаемые на X3, подключить модуль через разъем X4.

В штатной поставке ядро сконфигурировано на использование данного модуля расширения – /dev/fb2, в качестве контроллера TP включен ADS7843 (или аналог). Для демонстрации необходимо выполнить скрипт «qt\_fb2\_ev1\_test», во время работы которого сначала выполнится калибровка сенсорного экрана, а затем запустится пример работы Qt приложения «Affine».

**Возможные режимы конфигурации устройств отображения**

Настройки и режимы работы передаются через аргументы запуска ядра, которые хранятся в переменных окружения загрузчика u-boot. Предусмотрено 6 вариантов конфигурации:

```
select_vout=run vout_mode0
vout_mode0=setenv select_vout 'run vout_mode0'; setenv bootargs_vout 'setenv bootargs
${bootargs} video=mxcfb0:dev=ldb,SK-ATM0704,if=RGB24 video=mxcfb1:dev=ldb,SK-ATM0704,if=RGB24
ldb=sep0'
vout_mode1=setenv select_vout 'run vout_mode1'; setenv bootargs_vout 'setenv bootargs
${bootargs} video=mxcfb0:dev=ldb,LDB-XGA,if=RGB24 ldb=dul0'
vout_mode2=setenv select_vout 'run vout_mode2'; setenv bootargs_vout 'setenv bootargs
${bootargs} video=mxcfb1:dev=ldb,SK-ATM0704,if=RGB24 video=mxcfb0:dev=lcd,SK-MI0430,if=RGB24
ldb=dull1'
vout_mode3=setenv select_vout 'run vout_mode3'; setenv bootargs_vout 'setenv bootargs
${bootargs} video=mxcfb0:dev=lcd,SK-ATM0704,if=RGB24 video=mxcfb1:dev=ldb,SK-ATM0704,if=RGB24
ldb=dull1'
vout_mode4=setenv select_vout 'run vout_mode4'; setenv bootargs_vout 'setenv bootargs
${bootargs} video=mxcfb1:dev=ldb,SK-ATM0704,if=RGB24 video=mxcfb0:dev=lcd,LDB-XGA,if=RGB24
ldb=dull1'
vout_mode5=setenv select_vout 'run vout_mode5'; setenv bootargs_vout 'setenv bootargs
${bootargs} video=mxcfb0:dev=hdmi,1920x1080M@60,if=RGB24 video=mxcfb1:dev=ldb,SK-ATM0704,if=RGB24
ldb=dull1'
```

Для активации необходимого режима необходимо прервать загрузку системы в загрузчике u-boot нажатием любой клавиши и ввести команду:

```
U-Boot > run vout_modeX <- указать режим работы
U-Boot > boot
```

После чего система загрузится с новой конфигурацией режимов отображения. Если необходимо сделать конфигурацию видеорежимов загружаемой по умолчанию, необходимо после команды «setenv select\_vout 'run vout\_modeX'» выполнить команду «saveenv».

**vout\_mode0**

Режим настроек по умолчанию, описан выше.

**vout\_mode1**

Конфигурирует LVDS0 порт процессора на использование SK-TFT1024x768TP-Plug.

**vout\_mode2**

Конфигурирует LVDS1 (X11) порт процессора на использование SK-ATM0700D4-Plug - /dev/fb2. LCD порт /dev/fb0 (X5) на использование SK-MI0430FT-Plug.

**vout\_mode3**

Конфигурирует LVDS1 (X11) порта процессора на использование SK-ATM0700D4-Plug - /dev/fb2. LCD порт /dev/fb0 (X5) на использование SK-ATM0700D4-Plug или EV-

ATM5HD-Plug. SK-ATM0700D4-Plug предварительно необходимо настроить на использование параллельной RGB шины - разомкнуть J10, подключиться к разъему X2.

При использовании сенсорного экрана совместно с модулями подключаемыми через разъемы X8,X9 необходимо проделать манипуляции описанные для режима vout\_mode2.

#### vout\_mode4

Конфигурирует LVDS1 (X11) порта процессора на использование SK-ATM0700D4-Plug - /dev/fb0. LCD порт /dev/fb2 (X5) на использование SK-TFT1024x768TP-Plug, который предварительно необходимо настроить на использование параллельной RGB шины - замкнуть все перемычки на разъеме X3, разомкнуть EDGESEL, подключиться к разъему X1.

#### vout\_mode5

Конфигурирует LVDS1 (X11) порта процессора на использование SK-TFT1024x768TP-Plug - /dev/fb2. HDMI интерфейс процессора конфигурируется на устройство /dev/fb0.

#### SK-VideoADC-Plug – разъем X9

В штатной поставке имеется два примера взаимодействия с этим модулем расширения:

videoadc\_plug\_record – осуществляет захват-сжатие и запись в MKV файл

videoadc\_plug\_stream – осуществляет трансляцию в видеопотока в сеть, для просмотра можно воспользоваться плеером VLC, tcp://192.168.0.136:8080

#### SK-SIMCOM-Plug – разъем X4

В штатной поставке, для взаимодействия с данным модулем могут быть использованы /dev/ttymx1 и /dev/ttymx2 порты.

### 3. Состав ОС Linux

Ядро 3.0.35, включая драйвера:

- Ethernet
- PCIe
- USB-host
- USB-gadget
- HDMI
- LCD/LVDS
- SD/MMC/eMMC
- CSI
- I2C
- ISI
- SPI
- UART
- RTC
- TP ADS7843
- ...

### 4. Способы загрузки и содержимое корневой файловой системы

iMX6 подразумевает различные возможные источники загрузки, загрузка модуля осуществляется с eMMC.

**Внимание!** Далее описывается содержимое носителя при розничных отгрузках модулей (до 5шт), в случае оптовых и мелкооптовых поставок модули содержат только ядро с интегрированной файловой системой.

В штатной поставке, eMMC содержит загрузчик, ядро, ядро с интегрированной ФС (для «аварийной» загрузки) и корневую ФС.

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

- Qt
- Gstreamer
- Samba – сервер Microsoft сети
- HTTPD – сервер HTTP
- VSFTPD – сервер FTP
- Dropbear – сервер SSH
- 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
- Mplayer – медиа-проигрыватель
- MC – файловый менеджер
- ...

На случай аварии корневой файловой системы (или для случая обновления корневой ФС), 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 отформатирована eMMC flash, скопирован с TFTP сервера виртуальной машины и распакован архив корневой ФС.

#### 4.1. Загрузка через USB порт

В случае, если процессор не находит исполняемого кода во внешних носителях, он переходит в режим загрузки через USB. Для активации этого режима, необходимо замкнуть J1 перед включением питания, а после подачи питания разомкнуть. Основное назначение данного режима загрузки – восстановление системы, см. далее.

### 5. Виртуальная машина VMware

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

Разархивируйте файл “ SK-iMX6-SODIMM\_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, предусмотрен 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) 192.168.0.10

Eth1 (Host-only) с адресом 192.168.2.2, задуман для взаимодействия с PC (т.к. Bridget интерфейс отключается при физически выключенном кабеле), в частности, для возможности копирования файлов из виртуальной системы по FTP. 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, виртуальную машину 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 системным 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

После входа, переключаемся на консоль (Ctrl+Alt+F(1-6)) (потребуется в опциях VMware освободить сочетание клавиш Ctrl+Alt - по умолчанию это выход из окна виртуальной машины), запускаем 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/main\_fs/emmc\_fs – содержит корневую систему штатной загрузки собираемую с помощью buildroot, «make» собирает корневую ФС и копирует ее архив в /home/user/tftp папку. Для конфигурирования содержимого необходимо выполнить «make menuconfig»
- rootfs/safe\_fs – содержит корневую систему для safe загрузки, Для конфигурирования содержимого необходимо выполнить «make menuconfig», для сборки достаточно выполнить скрипт **build.sh**
- rootfs/linaro\_fs/ubuntu\_desktop – содержит Ubuntu вариант корневой ФС (вариант системы с графическим столом), для его подготовки необходимо запустить скрипт **prepare\_dt.sh**
- rootfs/linaro\_fs/ubuntu\_server – содержит Ubuntu вариант корневой ФС, для его подготовки необходимо запустить скрипт **prepare\_server.sh**
- u-boot – содержит загрузчик u-boot  
**build.sh** – собирает u-boot для загрузки системы с NAND flash и копирует исполняемый файл в /home/user/tftp папку
- app/hello\_world – содержит пример вывода текста в консоль
- app/gpio\_mmap – содержит пример взаимодействия с регистрами процессора из области приложения пользователя

На заметку, пример gpio\_mmap создан в первую очередь с целью показать, как можно читать-писать в регистры процессора из собственного приложения. Управлять состоянием GPIO гораздо проще через стандартные средства ядра см. /etc/init.d/S92gpio.

## 5.1. Примеры

Обновление ядра Linux, для этого необходимо:

- запускаем виртуальную машину
- запускаем скрипт /home/user/src/kernel/linux\_rel\_imx\_3.0.35\_4.1.0-sk/build.sh

- включаем/перезагружаем плату с подключенным Ethernet (разъем T1) и RS232 кабелями
- после загрузки, выполняем «system\_update && reboot»  
Загрузка ядра Linux с TFTP сервера, для этого необходимо:
- запускаем виртуальную машину
- включаем/перезагружаем плату с подключенным Ethernet (разъем T1) и RS232 кабелями
- прерываем в u-boot процесс загрузки нажатием любой клавиши
- выполняем “run tftp\_boot”  
Обновление u-boot, для этого необходимо:
- запускаем виртуальную машину
- включаем/перезагружаем плату с подключенным Ethernet (разъем T1) и RS232 кабелями
- после загрузки системы, выполняем “uboot\_update”

## 6. 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 (загрузчик u-boot) имеет заголовок с настройками внешней DDR3 памяти. Загрузчик u-boot обладает обширными возможностями, например, он умеет копировать файлы с TFTP, SD и т.п., поддерживает целый набор команд и режимов. В переменных окружения u-boot есть команда запуска, в которой указано, откуда следует считать и запустить ядро. Следующие сообщения консоли иллюстрируют этот процесс:

```
mmc0(part 0) is current device
8451860 bytes read in 461 ms (17.5 MiB/s)
## Booting kernel from Legacy Image at 12000000 ...
```

Перед запуском ядра Linux, первым делом проверяется контрольная сумма собственного архива и передается управление ядру (в случае safe загрузки, ядро включает в себя еще корневую ФС), иллюстрация:

```
## Booting kernel from Legacy Image at 12000000 ...
Image Name:      Linux
Image Type:      ARM Linux Kernel Image (uncompressed)
Data Size:       8451796 Bytes = 8.1 MiB
Load Address:    10008000
Entry Point:     10008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
Starting kernel ...
```

Далее идет инициализация всей системы, драйверов, файловых систем, после чего управление передается скриптам запуска /etc/init.d.

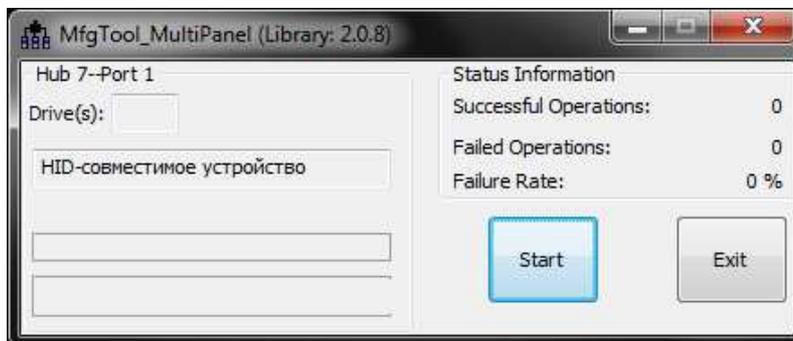
## 7. Восстановление системы

Предварительно необходимо распаковать MFG-tool утилиту (архив Mfgtools-Rel-1.1.0\_121218\_MX6S\_SODIMM\_UPDATER\_SK.rar). Подключить USB кабель к разъему X19, подключить RS232 кабель и запустить терминальную программу.

Произвести манипуляции в соответствии с пунктом 4.1. (Загрузка через USB порт)

Включить питание платы, при первом запуске, в системе появится новое HID устройство.

Запустить MfgTool2.exe, в результате должно получиться:



Нажать кнопку «Start», в терминальной программе будет отображаться рабочий процесс.

Утилита копирует в память загрузчик и ядро со встроенной ФС, далее передаст управление загрузчику. После запуска ядра, утилита копирует необходимые файлы и запустил скрипт «/bin/emmc\_format», который отформатирует носитель, копирует и распакует файловую систему.

После завершения работы, необходимо отключить питание и настроить режим загрузки с eMMC flash.

## 8. Назначение контактов модуля SK-iMX6S/D-SODIMM

Ниже перечислены названия выводов процессора, соответствие можно уточнить в «Reference Manual». Жирным шрифтом выделена функция используемая для данного вывода в BSP Linux.

Выводы портов общего назначения имеют логические уровни 3.3В.

Ножевой разъем X6 (возможные варианты разъемов: “Тусо” – 1433005-4, “Foxconn” AS0A426-N2SN ... )

N	Наименование вывода	Номер вывода процессора, дополнительные функции вывода, примечание
1	5V	Питание 5В
3	5V	Питание 5В
5	5V	Питание 5В
7	GND	0В
9	SPI2_CLK	N1: CSIO D4, GPIO_5_22, <b>ECSPI1_SCLK</b> , AUD3_TXC ...
11	SPI2_CS	N3: CSIO D7, GPIO_5_25, <b>ECSPI1_SS0</b> , AUD3_RXD ...
13	SPI2_MOSI	P2: CSIO D5, GPIO_5_23, <b>ECSPI1 MOSI</b> , AUD3_TXD ...
15	SPI2_MISO	N4: CSIO D6, GPIO_5_24, <b>ECSPI1 MISO</b> , AUD3_TXFS ...
17	RESET	CPU Reset
19	VBAT	Питание часов реального времени (RTC), 3.0V
21	SD1_D0	A21: <b>SD1_DAT0</b> , GPIO_1_16, GPT_CAPTURE1 ...
23	SD1_D1	C20: <b>SD1_DAT1</b> , GPIO_1_17, PWM3_OUT, GPT_CAPTURE2 ...
25	SD1_D2	E19: <b>SD1_DAT2</b> , GPIO_1_19, PWM2_OUT, GPT_COMPARE2 ...
27	SD1_D3	F18: <b>SD1_DAT3</b> , GPIO_1_21, PWM1_OUT, WDOG2_B ...
29	SD1_CMD	B21: <b>SD1_CMD</b> , GPIO_1_8, PWM4_OUT ...
31	SD1_CLK	D20: <b>SD1_CLK</b> , GPIO_1_20 ...
33	3.3V_OUT	Выход питания 3.3В, максимальный ток нагрузки по всем контактам 3.3V_OUT должен быть не более 0,5А
35	SD2_CLK	C21: SD2_CLK, KEY_COL5, AUD4_RXFS, <b>GPIO1_IO10</b>
37	SD2_CMD	F19: SD2_CMD, KEY_ROW5, AUD4_RXC, <b>GPIO1_IO11</b>
39	SD2_D0	A22: SD2_DAT0, AUD4_RXD, KEY_ROW7, <b>GPIO1_IO15</b>
41	SD2_D1	E20: SD2_DAT1, EIM_CS2, AUD4_TXFS, KEY_COL7, <b>GPIO1_IO14</b>
43	SD2_D2	A23: SD2_DAT2, EIM_CS3, AUD4_TXD, KEY_ROW6, <b>GPIO1_IO13</b>
45	SD2_D3	B22: SD2_DAT3, KEY_COL6, AUD4_TXC, <b>GPIO1_IO12</b>
47	GND	0В
49	SPDIF_OUT	R1: SPDIF_OUT, ESAI_TX0, GPIO7_IO12 ...
51	SPDIF_IN	U5: SPDIF_IN, KEY_COL3, ECSPI1_SS3, HDMI_TX_DDC_SCL, I2C2_SCL, GPIO4_IO12
53	HDMI_HPD	K1: HDMI_HPD
55	ON_OFF	D12: ONOFF
57	ON_REQ	D11: PMIC_ON_REQ
59		
61		
63		
65		

67	LVDS2_0_TXN	CPU LVDS1
69	LVDS2_0_TXP	CPU LVDS1
71	LVDS2_1_TXN	CPU LVDS1
73	LVDS2_1_TXP	CPU LVDS1
75	LVDS2_2_TXN	CPU LVDS1
77	LVDS2_2_TXP	CPU LVDS1
79	LVDS2_CLKN	CPU LVDS1
81	LVDS2_CLKP	CPU LVDS1
83	LVDS2_3_TXN	CPU LVDS1
85	LVDS2_3_TXP	CPU LVDS1
87	GND	0B
89	SATA_TXP	CPU SATA, доступен только у SK-iMX6D-SODIMM
91	SATA_TXN	CPU SATA, доступен только у SK-iMX6D-SODIMM
93	SATA_RXN	CPU SATA, доступен только у SK-iMX6D-SODIMM
95	SATA_RXP	CPU SATA, доступен только у SK-iMX6D-SODIMM
97	GND	0B
99	HDMI_CKN	CPU HDMI
101	HDMI_CKP	CPU HDMI
103	HDMI_N0	CPU HDMI
105	HDMI_P0	CPU HDMI
107	HDMI_N1	CPU HDMI
109	HDMI_P1	CPU HDMI
111	HDMI_N2	CPU HDMI
113	HDMI_P2	CPU HDMI
115	GND	0B
117	LVDS1_0_TXN	CPU LVDS0
119	LVDS1_0_TXP	CPU LVDS0
121	LVDS1_1_TXN	CPU LVDS0
123	LVDS1_1_TXP	CPU LVDS0
125	LVDS1_2_TXN	CPU LVDS0
127	LVDS1_2_TXP	CPU LVDS0
129	LVDS1_CLKN	CPU LVDS0
131	LVDS1_CLKP	CPU LVDS0
133	LVDS1_3_TXN	CPU LVDS0
135	LVDS1_3_TXP	CPU LVDS0
137	GND	0B
139		
141		
143		
145		
147	USB2_N	CPU USB Host
149	USB2_P	CPU USB Host
151	USB1_OTG_N	CPU USB OTG
153	USB1_OTG_P	CPU USB OTG

155	USB1_OTG_VBUS	CPU USB OTG
157	USB1_OTG_ID	CPU USB OTG
159	3.3V_OUT	Выход питания 3.3В, максимальный ток нагрузки по всем контактам 3.3V_OUT должен быть не более 0,5А
161	ETH_LED1	Индикация состояния Ethernet
163	ETH_LED2	Индикация состояния Ethernet
165	ETH_CT	«Средняя точка» для трансформатора Ethernet
167	TXP	Ethernet TXP
169	TXN	Ethernet TXN
171	RXP	Ethernet RXP
173	RXN	Ethernet RXN
175		
177		
179		
181		
183	GND	0В
185	PCIE_TXP	CPU PCIe
187	PCIE_TXN	CPU PCIe
189	PCIE_RXP	CPU PCIe
191	PCIE_RXN	CPU PCIe
193	PCIE_REFP	CPU PCIe
195	PCIE_REFN	CPU PCIe
197	GND	0В
199	3.3V_OUT	Выход питания 3.3В, максимальный ток нагрузки по всем контактам 3.3V_OUT должен быть не более 0,5А
2	5V	Питание 5В
4	5V	Питание 5В
6	5V	Питание 5В
8	GND	0В
10	SPI1_CLK	H20: ECSPi4_SCLK, GPIO_3_21, I2C1_SCL, SPDIF_IN ...
12	SPI1_CS	G20: EIM_DATA20, ECSPi4_SS0, GPIO_3_20...
14	SPI1_MOSI	G23: ECSPi4_MOSI, GPIO_3_28 ...
16	SPI1_MISO	E23: ECSPi4_MISO, GPIO_3_22, SPDIF_OUT, USB_OTG_PWR ...
18	UART1_TX_CON	M1: GPIO_5_28, UART1_TX, ECSPi2_MISO ...
20	UART1_RX_CON	M3: GPIO_5_29, UART1_RX, ECSPi2_SS0 ...
22	UART2_TX	R3: GPIO_1_7, UART2_TXD, FLEXCAN1_TX, I2C4_SCL ...
24	UART2_RX	R5: GPIO_1_8, UART2_RXD, FLEXCAN1_RX, I2C4_SDA ...
26	UART3_TX	F22: EMI D24, GPIO_3_24, UART3_TXD, ECSPi1 SS2, ECSPi2 SS2, AUD5_RXFS, UART1 DTR ...
28	UART3_RX	G22: EMI D25, GPIO_3_25, UART3_RXD, ECSPi1 SS3, ECSPi2 SS3, AUD5_RXC, UART1 DSR ...
30	UART4_TX	F24: EIM_ADDR22, GPIO2_IO16 ...
32	UART4_RX	E25: EIM_D27, UART2_RX, GPIO3_IO27 ...
34	CAN1_TX	W6: ECSPi1_SS1, FLEXCAN1_TX, GPIO4_IO10 ...
36	CAN1_RX	W4: ECSPi1_SS2, FLEXCAN1_RX, GPIO4_IO11 ...
38	CAN2_TX	T6: FLEXCAN2_TX, USB_OTG_OC, UART5_RTS_B, GPIO4_IO14 ...
40	CAN2_RX	V5: FLEXCAN2_RX, USB_OTG_PWR, UART5_CTS, GPIO4_IO15 ...

42	I2C1_SDA	C25: GPIO_3_16, <b>I2C2_SDA</b> , ECSP11_SCLK ...
44	I2C1_SCL	E22: EMI EB2, GPIO_2_30, ECSP11_SSO, <b>I2C2_SCL</b> ...
46	I2C2_SDA	D24: <b>I2C3_SDA</b> , GPIO_3_18, ECSP11_MOSI ...
48	I2C2_SCL	F21: <b>I2C3_SCL</b> , GPIO_3_17, ECSP11_MISO ...
50	PWM2	T2: WDOG1_B, PWM1_OUT, GPIO1_IO09 ...
52	PWM1	F17: PWM4_OUT, GPIO2_IO10, SD4_DATA2
54	GND	0B
56	CSI_PCK	P1: <b>CSIO_PIXCLK</b> , GPIO_5_18 ...
58	CSI_HS	P4: <b>CSIO_HSYNC</b> , GPIO_5_19 ...
60	CSI_VS	N2: <b>CSIO_VSYNC</b> , GPIO_5_21 ...
62	CSI_D7	L6: <b>CSIO D19</b> , GPIO_6_5, UART4_CTS ...
64	CSI_D6	M6: <b>CSIO D18</b> , GPIO_6_4, UART5_RTS ...
66	CSI_D5	L3: <b>CSIO D17</b> , GPIO_6_3, UART4_CTS ...
68	CSI_D4	L4: <b>CSIO D16</b> , GPIO_6_2, UART4_RTS ...
70	CSI_D3	M5: <b>CSIO D15</b> , GPIO_6_1, UART5_RXD_MUX ...
72	CSI_D2	M4: <b>CSIO D14</b> , GPIO_6_0, UART5_TXD_MUX ...
74	CSI_D1	L1: <b>CSIO D13</b> , GPIO_5_31, UART4_RXD_MUX ...
76	CSI_D0	M2: <b>CSIO D12</b> , GPIO_5_30, UART4_TXD_MUX ...
78	I2S_TXFS	U7: KPP_COL1, GPIO_4_8, <b>AUD5_TXFS</b> , UART5_TXD_MUX, ECSP11_MISO ...
80	I2S_RXD	U6: KPP_ROW1, GPIO_4_9, <b>AUD5_RXD</b> , UART5_RXD_MUX, ECSP11_SSO ...
82	I2S_TXD	V6: KPP_ROW0, GPIO_4_7, <b>AUD5_TXD</b> , UART4_RXD_MUX, ECSP11_MOSI ...
84	I2S_TXC	W5: KPP_COL0, GPIO_4_6, <b>AUD5_TXC</b> , UART4_TXD_MUX, ECSP11_SCLK ...
86	I2S_MCLK	A17: <b>CCM_CLKO2</b> , GPIO_6_15 ...
88	GND	0B
90	BUS_BCLK	N22: <b>EIM_BCLK</b>
92	BUS_CS1	J23: <b>EIM_CS1</b> , GPIO_2_24, ECSP12_MOSI ...
94	BUS_CS0	H24: <b>EIM_CS0</b> , GPIO_2_23, ECSP12_SDCLK ...
96	BUS_OE	J24: <b>EIM_OE</b> , GPIO_2_25, ECSP12_MISO, DI1_PIN7 ...
98	BUS_WAIT	M25: <b>EIM_WAIT</b> , GPIO_5_0, EIM_DTACK_B ...
100	BUS_RW	K20: <b>EIM_RW</b> , GPIO_2_26, ECSP12_SSO, DI1_PIN8 ...
102	BUS_LBA	K22: <b>EIM_LBA</b> , GPIO_2_27, ECSP12_SS1 ...
104	BUS_EB1	K23: <b>EIM_EB1</b> , GPIO_2_29 ...
106	BUS_EB0	K21: <b>EIM_EB0</b> , GPIO_2_28 ...
108	BUS_DA15	N24: <b>EIM_DA15</b> , GPIO_3_15 ...
110	BUS_DA14	N23: <b>EIM_DA14</b> , GPIO_3_14 ...
112	BUS_DA13	M23: <b>EIM_DA13</b> , GPIO_3_13 ...
114	BUS_DA12	M24: <b>EIM_DA12</b> , GPIO_3_12 ...
116	BUS_DA11	M20: <b>EIM_DA11</b> , GPIO_3_11 ...
118	BUS_DA10	M22: <b>EIM_DA10</b> , GPIO_3_10 ...
120	BUS_DA9	M21: <b>EIM_DA9</b> , GPIO_3_9 ...
122	BUS_DA8	L24: <b>EIM_DA8</b> , GPIO_3_8 ...
124	BUS_DA7	L25: <b>EIM_DA7</b> , GPIO_3_7 ...
126	BUS_DA6	K25: <b>EIM_DA6</b> , GPIO_3_6 ...
128	BUS_DA5	L23: <b>EIM_DA5</b> , GPIO_3_5 ...

130	BUS_DA4	L22: EIM_DA4, GPIO_3_4 ...
132	BUS_DA3	K24: EIM_DA3, GPIO_3_3 ...
134	BUS_DA2	L21: EIM_DA2, GPIO_3_2 ...
136	BUS_DA1	J25: EIM_DA1, GPIO_3_1 ...
138	BUS_DA0	L20: EIM_DA0, GPIO_3_0 ...
140	GND	0B
142	LCD_DE	N21: DIO_PIN15, GPIO_4_17, AUD6_TXC ...
144	LCD_VS	N20: DIO_PIN3, GPIO_4_19, AUD6_TXFS ...
146	LCD_HS	N25: DIO_PIN2, GPIO_4_18, AUD6_TXD ...
148	LCD_B7	R24: DISPO_DAT7, GPIO_4_28, ECSPI3 RDY ...
150	LCD_B6	R23: DISPO_DAT6, GPIO_4_27, ECSPI3 SS3 ...
152	LCD_B5	R25: DISPO_DAT5, GPIO_4_26, ECSPI3 SS2 ...
154	LCD_B4	P20: DISPO_DAT4, GPIO_4_25, ECSPI3 SS1 ...
156	LCD_B3	P21: DISPO_DAT3, GPIO_4_24, ECSPI3 SS0 ...
158	LCD_B2	P23: DISPO_DAT2, GPIO_4_23, ECSPI3 MISO ...
160	LCD_B1	P22: DISPO_DAT1, GPIO_4_22, ECSPI3 MOSI ...
162	LCD_B0	P24: DISPO_DAT0, GPIO_4_21, ECSPI3 SCLK ...
164	LCD_G7	T22: DISPO_DAT15, GPIO_5_9, ECSPI1 SS1, ECSPI2 SS1 ...
166	LCD_G6	U25: DISPO_DAT14, GPIO_5_8, AUD5_RXC ...
168	LCD_G5	R20: DISPO_DAT13, GPIO_5_7, AUD5_RXFS ...
170	LCD_G4	T24: DISPO_DAT12, GPIO_5_6 ...
172	LCD_G3	T23: DISPO_DAT11, GPIO_5_5 ...
174	LCD_G2	R21: DISPO_DAT10, GPIO_4_31 ...
176	LCD_G1	T25: DISPO_DAT9, GPIO_4_30, PWM2 PWMO, WDOG2_B ...
178	LCD_G0	R22: DISPO_DAT8, GPIO_4_29, PWM1 PWMO, WDOG1_B ...
180	LCD_R7	W24: DISPO_DAT23, GPIO_5_17, ECSPI1 SS0, AUD4_RXD ...
182	LCD_R6	V24: DISPO_DAT22, GPIO_5_16, ECSPI1 MISO, AUD4_TXFS ...
184	LCD_R5	T20: DISPO_DAT21, GPIO_5_15, ECSPI1 MOSI, AUD4_TXD ...
186	LCD_R4	U22: DISPO_DAT20, GPIO_5_14, ECSPI1 SCLK, AUD4_TXC ...
188	LCD_R3	U23: DISPO_DAT19, GPIO_5_13, ECSPI2 SCLK, AUD5_RXD, AUD5_RXC, WEIM_CS3 ...
190	LCD_R2	V25: DISPO_DAT18, GPIO_5_12, ECSPI2 SS0, AUD5_TXFS, AUD5_RXFS ...
192	LCD_R1	U24: DISPO_DAT17, GPIO_5_11, ECSPI2 MISO, AUD5_TXD ...
194	LCD_R0	T21: DISPO_DAT16, GPIO_5_10, ECSPI2 MOSI, AUD5_TXC ...
196	LCD_DCLK	N19: DIO_DISP_CLK, GPIO_4_16 ...
198	GND	0B
200	GND	0B

## 9. Дополнительные материалы

К модулю прилагаются материалы: габаритный чертеж модуля (в формате DXF), структурная схема модуля, схема электрическая принципиальная материнской платы SK-iMX6-MB-SODIMM, проектные файлы материнской платы SK-iMX6-MB-SODIMM (схема, файл печатной платы).