

# Процессорный модуль SK-iMX6S-OEM(-WiFi)

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



### **SK-iMX6S-OEM:**

Freescalе iMX6 Solo (ARM Cortex-A9 1000МГц)  
DDR3 512Мбайт  
NAND Flash 2Гбайт  
100/10М Ethernet  
Micro SD держатель  
Разъемы расширения  
Совместимость с SK-iMX53-OEM

### **SK-iMX6S-OEM-WiFi:**

Freescalе iMX6 Solo (ARM Cortex-A9 1000МГц)  
DDR3 512Мбайт  
NAND Flash 2Гбайт  
100/10М Ethernet  
WiFi, управление питанием  
Micro SD держатель  
Разъемы расширения  
Совместимость с SK-iMX53-OEM

Модуль SK-iMX6S-OEM(-WiFi) совместим по выводам и его можно использовать в материнских платах предназначенных для модулей SK-iMX53-OEM. Отличием в интерфейсах является:

- 1) Отсутствие SATA интерфейса, вместо него выведен Mini PCIe интерфейс
- 2) Отсутствие аналогового видеовыхода, на его места подключены GPIO линии
- 3) Исключен GPIO\_7\_11, заменен питанием батареи для встроенных часов реального времени

SK-iMX6S-OEM-WiFi отличается наличием в своем составе USB-WiFi модуля вписанного в периметр печатной платы, при этом использован USB HOST порт процессора, который не может быть использован пользователем.

### **SK-iMX6S-OEM(-WiFi), возможность прямого подключения:**

SK-iMX6-MB – материнская плата для демонстрации возможностей процессорного модуля

### **SK-iMX6-MB, 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-OEM(-WiFi):

- Напряжение питания: 5В
- Потребляемый ток до 0,5А
- Габариты 77x57мм

### SK-iMX6-MB:

- Напряжение питания: 5-6В (питающее напряжение – центральный штырь разъема), при использовании USB-host 6В максимум, рекомендуемое напряжение 5В
- Потребляемый ток (зависит от подключения внешних модулей) до 2А
- Габариты 124x109мм

## 1. Назначение джамперов

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

### SK-iMX53-OEM:

- J1 определяет источник загрузки системы: разомкнут – NAND flash, замкнут - SD карта, разъем X1

### SK-iMX53-MB:

- J1 позволяет подключать согласующий резистор для CAN шины, интерфейс CAN1
- J2 позволяет подключать согласующий резистор для CAN шины, интерфейс CAN2
- J3-J4 позволяет выбирать подключение разъема X24 к микрофонному или линейному входу звукового кодека
- J5-J6 позволяет выбирать подключение разъема X25 к выходу на наушники или линейному выходу звукового кодека
- J8 позволяет использовать питание шины USB

**По умолчанию замкнуты перемычки: J1, J2, J3-J4 – положение 1-2, J5-J6 – положение 1-2;**

## 2. 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 “ОЕМ модули > SK-iMX6S-OEM”.

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, настройте IP адрес сетевой карты PC в диапазоне 192.168.0.XXX.

При необходимости, подключите SK-ATM0700D4-Plug к разъему X1, SK-TFT1024x768TP-Plug к разъему X2.

Подключите питание, в терминальной программе появятся следующие сообщения:

```
U-Boot 2009.08 (Aug 28 2013 - 13:53:14)
CPU: Freescale i.MX6 family TO1.1 at 792 MHz
Thermal sensor with ratio = 179
Temperature: 44 C, calibration data 0x5704e169
mx6q pll1: 792MHz
mx6q pll2: 528MHz
mx6q pll3: 480MHz
mx6q pll8: 50MHz
ipg clock : 66000000Hz
```

```

ipg per clock : 66000000Hz
uart clock    : 80000000Hz
cspi clock    : 60000000Hz
ahb clock     : 132000000Hz
axi clock     : 198000000Hz
emi_slow clock: 99000000Hz
ddr clock     : 396000000Hz
usdhc1 clock  : 198000000Hz
usdhc2 clock  : 198000000Hz
usdhc3 clock  : 198000000Hz
usdhc4 clock  : 198000000Hz
nfc clock     : 198000000Hz
Board: SK-iMX6-OEM:[ WDOG]
Boot Device: NAND
I2C:  ready
DRAM:  512 MB
NAND:  Manufacturer ID: 0xec, Chip ID: 0xd5 (Samsung NAND 2GiB 3,3V 8-bit), page size: 8192, OOB
size: 436
2048 MiB
MMC:  FSL_ESDHC: 0,FSL_ESDHC: 1
*** Warning - bad CRC or NAND, using default environmen
In:  serial
Out: serial
Err: serial
Net:  FEC0 [PRIME]
Hit any key to stop autoboot:  3  2  1  0
NAND read: device 0 offset 0x1100000, size 0x500000
5242880 bytes read: OK
## Booting kernel from Legacy Image at 10800000 ...
Image Name:   Linux-3.0.35
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:   3716260 Bytes = 3.5 MB
Load Address: 10008000
Entry Point: 10008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
OK
Starting kernel ...
Linux version 3.0.35 (user@iMX6-bld) (gcc version 4.7.1 20120402 (prerelease) (crosstool-NG
linaro-1.13.1-2012.04-20120426 - Linaro GCC 2012.04) ) #107 SMP PREEMPT Thu Aug 22 20:35:48 UTC
2013
CPU: ARMv7 Processor [412fc09a] revision 10 (ARMv7), cr=10c53c7d
CPU: VIPT nonaliasing data cache, VIPT aliasing instruction cache
Machine: SK-iMX6-OEM module, www.starterkit.ru
Memory policy: ECC disabled, Data cache writealloc
CPU identified as i.MX6DL/SOLO, silicon rev 1.1
PERCPU: Embedded 7 pages/cpu @8bc06000 s5440 r8192 d15040 u32768
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 97280
Kernel command line: console=ttymx0,115200 ubi.mtd=4 root=ubi0:nandfs rootwait rw
rootfstype=ubifs video=mxcfb0:dev=ldb,SK-ATM0704,if=RGB24 video=mxcfb1:dev=ldb,LDB-XGA,if=RGB24
ldb=sepl
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: 384MB = 384MB total
Memory: 380992k/380992k available, 143296k 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 - 0x8003c000 ( 208 kB)
 .text : 0x8003c000 - 0x806cd8dc (6727 kB)
 .data : 0x806ce000 - 0x80730c00 ( 395 kB)
 .bss : 0x80730c24 - 0x8077ea4c ( 312 kB)
SLUB: Genslabs=13, HWalign=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 [ttymx0] 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
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).
print_constraints: dummy:
NET: Registered protocol family 16
print_constraints: vddpu: 725 <--> 1300 mV at 700 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
l2x0: 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.0: probed
spi_imx imx6q-ecspi.3: probed
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
imx-ipuv3 imx-ipuv3.0: IPU DMFC NORMAL mode: 1(0~1), 5B(4,5), 5F(6,7)
mxc_mipi_csi2 mxc_mipi_csi2: i.MX MIPI CSI2 driver probed
mxc_mipi_csi2 mxc_mipi_csi2: i.MX MIPI CSI2 dphy version is 0x3130302a
MIPI CSI2 driver module loaded
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=a08b8000
PM driver module loaded
IMX usb wakeup probe
IMX usb wakeup probe
cpu regulator mode:ldo_enable
i.MXC CPU frequency driver
JFFS2 version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
msgmni has been set to 744
alg: No test for stdrng (krng)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
MIPI DSI driver module loaded
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
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
loop: module loaded
NAND device: Manufacturer ID: 0xec, Chip ID: 0xd5 (Samsung NAND 2GiB 3,3V 8-bit)
Creating 5 MTD partitions on "gpmi-nand":
```



```

0x000000000000-0x000001000000 : "U-boot partition"
0x000001000000-0x000001100000 : "U-boot environment partition"
0x000001100000-0x000001800000 : "Linux kernel partition"
0x000001800000-0x000002400000 : "Linux safe kernel partition (rootfs in initramfs)"
0x000002400000-0x000008000000 : "NAND rootfs partition"
GPMI NAND driver registered. (IMX)
UBI: attaching mtd4 to ubi0
UBI: physical eraseblock size: 1048576 bytes (1024 KiB)
UBI: logical eraseblock size: 1032192 bytes
UBI: smallest flash I/O unit: 8192
UBI: VID header offset: 8192 (aligned 8192)
UBI: data offset: 16384
UBI: max. sequence number: 107
UBI: attached mtd4 to ubi0
UBI: MTD device name: "NAND rootfs partition"
UBI: MTD device size: 2012 MiB
UBI: number of good PEBs: 2005
UBI: number of bad PEBs: 7
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: 152
UBI: total number of reserved PEBs: 1853
UBI: number of PEBs reserved for bad PEB handling: 20
UBI: max/mean erase counter: 1/0
UBI: image sequence number: 1168332990
UBI: background thread "ubi_bgt0d" started, PID 1216
vcan: Virtual CAN interface driver
CAN device driver interface
flexcan netdevice driver
flexcan imx6q-flexcan.0: device registered (reg_base=a0cd0000, irq=142)
flexcan imx6q-flexcan.1: device registered (reg_base=a0cd8000, irq=143)
FEC Ethernet Driver
fec_enet_mii_bus: probed
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
add wake up source irq 75
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)
udc: request mem region for fsl-usb2-udc failed
fsl-usb2-udc: probe of fsl-usb2-udc failed with error -16
mousedev: PS/2 mouse device common for all mice
_regulator_get: spi0.0 supply vcc not found, using dummy regulator
ads7846 spi0.0: touchscreen, irq 367
input: ADS7846 Touchscreen as /devices/platform/imx6q-ecspi.0/spi_master/spi0/spi0.0/input/input0
_regulator_get: spi3.0 supply vcc not found, using dummy regulator
ads7846 spi3.0: touchscreen, irq 265
input: ADS7846 Touchscreen as /devices/platform/imx6q-ecspi.3/spi_master/spi3/spi3.0/input/input1
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 video16
mxc_v4l2_output mxc_v4l2_output.0: V4L2 device registered as video17
mxc_v4l2_output mxc_v4l2_output.0: V4L2 device registered as video18
i2c-core: driver [mag3110] using legacy suspend method
i2c-core: driver [mag3110] using legacy resume method
imx2-wdt imx2-wdt.0: IMX2+ Watchdog Timer enabled. timeout=60s (nowayout=1)
Bluetooth: Virtual HCI driver ver 1.3
Bluetooth: HCI UART driver ver 2.2
Bluetooth: HCIATH3K protocol initialized
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
mmc0: SDHCI controller on platform [sdhci-esdhc-imx.0] using ADMA
sdhci sdhci-esdhc-imx.1: no write-protect pin available!
mmc1: SDHCI controller on platform [sdhci-esdhc-imx.1] using ADMA

```

```

mxc_vdoa mxc_vdoa: i.MX Video Data Order Adapter(VDOA) driver probed
VPU initialized
mxc_asrc registered
Thermal calibration data is 0x5704e169
Thermal sensor with ratio = 179
Anatop Thermal registered as thermal_zone0
anatop_thermal_probe: default cooling device is cpufreq!
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
Cirrus Logic CS42888 ALSA SoC Codec Driver
i2c-core: driver [cs42888] using legacy suspend method
i2c-core: driver [cs42888] using legacy resume method
usb 2-1: new high speed USB device number 2 using fsl-ehci
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
imx_3stack asoc driver
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)
can: broadcast manager protocol (rev 20090105 t)
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
Bluetooth: HIDP (Human Interface Emulation) ver 1.2
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
Bus freq driver Enabled
mxc_dvfs_core_probe
DVFS driver module loaded
snvs_rtc snvs_rtc.0: setting system clock to 1970-01-01 01:50:12 UTC (6612)
UBIFS: mounted UBI device 0, volume 0, name "nandfs"
UBIFS: file system size: 1878589440 bytes (1834560 KiB, 1791 MiB, 1820 LEBs)
UBIFS: journal size: 33030144 bytes (32256 KiB, 31 MiB, 32 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 0:12.
Freeing init memory: 208K
can't open /dev/null: No such file or directory
Init: rm: can't remove '/bin/ip': No such file or directory
imx-ipuv3 imx-ipuv3.0: IPU DMFC DC HIGH RESOLUTION: 1(0~3), 5B(4,5), 5F(6,7)
Starting logging: OK
Initializing random number generator... done.
Starting network...
eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy_addr=1:00, irq=-1)
flexcan imx6q-flexcan.0: writing ctrl=0x0e312085
flexcan imx6q-flexcan.1: writing ctrl=0x0e312085
Starting dropbear sshd: OK
Starting sshd: OK
Starting wi-fi network ...
usb 2-1: USB disconnect, device number 2
usb 2-1: new high speed USB device number 3 using fsl-ehci
PHY: 1:00 - Link is Up - 100/Full
ifconfig: SIOCSIFADDR: No such device
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.

Welcome to SK-iMX6-OEM module!

SK-iMX6-OEM login:

```

Это 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 (для SSH потребуется

создать пользователя в системе), сетевой адрес платы 192.168.0.136. При подключении-отключении USB, SD/MMC карт памяти, они будут автоматически монтироваться-размонтироваться в системе.

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

```
# date -s 2012.06.05-15:24:10
Tue Jun  5 15:24:10 MSD 2012
# hwclock -w
```

## 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 (X1), разрешение 800x480, консоль /dev/tty1

/dev/fb1 – прозрачное наложение на fb0

/dev/fb2 – LVDS1 выход (X2), разрешение 1024x768

SK-ATM0700D4-Plug – разъем X1

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\_ev1\_test», во время работы которого сначала выполнится калибровка сенсорного экрана, а затем запустится пример работы Qt приложения «Affine».

SK-TFT1024x768TP-Plug – разъем X2

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\_ev0\_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:

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



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

```
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 «run vout\_modeX» выполнить команду «saveenv».

#### vout\_mode0

Режим настроек по умолчанию, описан выше.

#### vout\_mode1

Конфигурирует оба LVDS порта процессора на использование SK-ATM0700D4-Plug.

#### vout\_mode2

Конфигурирует LVDS1 (X2) порта процессора на использование SK-TFT1024x768TP-Plug - /dev/fb0. LCD порт /dev/fb2 (X8,X9) на использование SK-MI0430FT-Plug.

Для использования сенсорного экрана при подключении модулей через разъем X8,X9 необходимо модифицировать ядро Linux:

- 1) Внести изменения в файл `home/user/src/kernel/linux_rel_imx_3.0.35_4.0.0-sk/arch/arm/mach-mx6/ board-mx6q_sabrelite.c` – закоментировать строку `#define TS_TO_LVDS0`
- 2) Пересобрать ядро – выполнить скрипт `build.sh`
- 3) Загрузить или обновить новое ядро системы

#### vout\_mode3

Конфигурирует LVDS1 (X2) порта процессора на использование SK-TFT1024x768TP-Plug - /dev/fb0. LCD порт /dev/fb2 (X8,X9) 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 (X2) порта процессора на использование SK-ATM0700D4-Plug - /dev/fb0. LCD порт /dev/fb2 (X8,X9) 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.

При использовании сенсорного экрана совместно с модулем подключенным через разъемы X8,X9 необходимо проделать манипуляции описанные для режима `vout_mode2`.

#### vout\_mode5

Конфигурирует LVDS1 (X2) порта процессора на использование SK-TFT1024x768TP-Plug - /dev/fb2. HDMI интерфейс процессора конфигурируется на устройство /dev/fb0.

#### SK-VideoADC-Plug – разъем X6

В штатной поставке драйвер CSI включен в ядро, но на данный момент никаких утилит для его использования не предусмотрено.

#### SK-SIMCOM-Plug – разъем X10

Линии процессора с функциями UART портов выводятся на соответствующие контакты данного модуля расширения, но на данный момент Linux BSP не использует эти функции.

### 3. Состав ОС Linux

Ядро 3.0.35, включая драйвера:

- Ethernet
- NAND flash
- USB-host
- USB-gadget
- HDMI
- LCD/LVDS
- SD/MMC
- CSI
- I2C
- ISI
- SPI
- UART
- RTC
- Frame Buffer
- TP ADS7843
- ...

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

iMX6 подразумевает различные возможные источники загрузки, на модуле предусмотрено три - NAND flash, Micro SD карта, USB порт

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

NAND flash разбита на две части:

- 1) /dev/mtd0 - 16М, содержит загрузчик u-boot
- 2) /dev/mtd1 – 1М, содержит переменные окружения u-boot
- 3) /dev/mtd2 – 7М, содержит ядро Linux
- 4) /dev/mtd3 – 12.5М, содержит ядро Linux со встроенной корневой ФС (для «аварийной» загрузки)
- 5) /dev/mtd4 - 2100М – раздел UBI файловой системы, используется в качестве корневой файловой 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

- ...

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 пятый раздел NAND flash, скопирован с TFTP сервера виртуальной машины и распакован архив корневой ФС.

#### 4.1. Загрузка с Micro SD карты, разъем X1

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

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

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

Подготовленную карту необходимо вставить в разъем X1, замкнуть J1, после чего включить питание.

#### 4.2. 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 осуществить режим загрузки с uSD карты при отсутствии самой карты. В штатном виде, основное назначение режима загрузки через USB – 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-OEM\_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 взаимодействие 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 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 того, как сетевые интерфейсы настроены, можно запускать виртуальную машину, после загрузки 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 два пользователя:

- root, пароль 123456
- user, пароль 123456 (настоятельно рекомендую работать под этим пользователем, или создать нового, но не вести всю работу под root)

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/nand\_fs – содержит корневую систему штатной загрузки собираемую с помощью buildroot, скрипт **build.sh** 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
- u-boot – содержит загрузчик системы, в корневой директории лежат скрипты:  
**build.sh** – собирает u-boot для загрузки системы с NAND flash и копирует исполняемый файл в /home/user/tftp папку

## 5.1. Примеры

Обновление ядра Linux, для этого необходимо:

- запускаем виртуальную машину
- запускаем скрипт /home/user/src/kernel/linux-3.0.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 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 для 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, по какому адресу NAND flash следует прочитать образ ядра, куда этот образ памяти скопировать и по какому адресу запустить. Следующие сообщения консоли иллюстрируют этот процесс:

```
NAND read: device 0 offset 0x1100000, size 0x500000
5242880 bytes read: OK
```

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

```
## Booting kernel from Legacy Image at 10800000 ...
Image Name:   Linux-3.0.35
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    3716260 Bytes = 3.5 MB
Load Address: 10008000
Entry Point:  10008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
OK
Starting kernel ...
```

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

## 7. SK-iMX6S-OEM-WiFi

Скрипт запуска беспроводного сетевого соединения располагается в файле /etc/init.d/S90wlan. По умолчанию, настраивается режим ad-hoc (компьютер-компьютер), присваивается IP адрес 192.168.3.136, SSID: SK. Так же на модуле предусмотрено управление питанием WiFi модуля, реализовано в скрипте запуска беспроводной сети S90wlan.

WiFi модуль использует USB Host порт процессора, поэтому выводы модуля USB1\_N и USB1\_P не будут доступны пользователю.



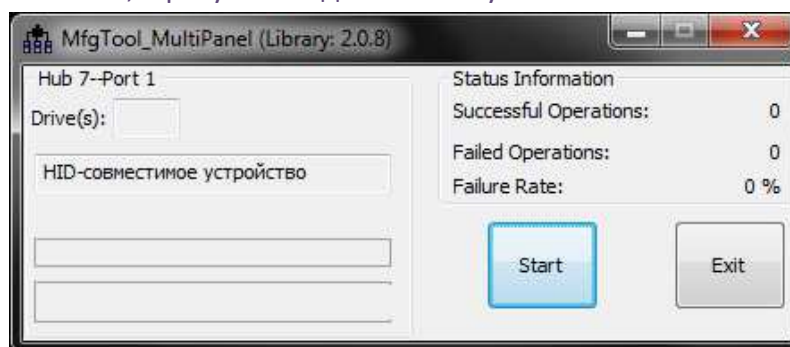
## 8. Восстановление системы

Предварительно необходимо распаковать MFG-tool утилиту (архив Mfgtools-Rel-1.1.0\_121218\_MX6S\_OEM\_UPDATER\_SK.rar). Подключить USB кабель к разъему X20, подключить RS232 кабель и запустить терминальную программу.

Произвести манипуляции в соответствии с пунктом 4.2. (загрузка через USB)

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

Запустить MfgTool2.exe, в результате должно получиться:

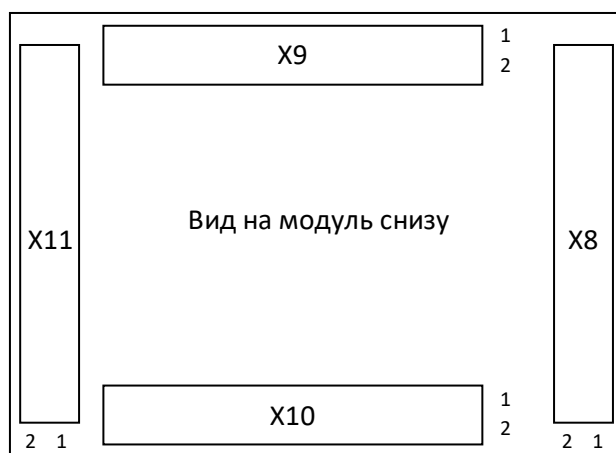


Нажать кнопку «Start», в терминальной программе будет отображаться рабочий процесс.

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/system\_prepare\_imx6\_oem», после копирует и распакует файловую систему во второй раздел NAND flash.

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

## 9. Назначение контактов модуля SK-iMX6S-OEM



Ниже перечислены названия выводов процессора, соответствие можно уточнить в «Reference Manual». Жирным шрифтом выделена функция используемая для данного вывода в BSP Linux. Желтым фоном помечены выводы, функциональное назначение которых отличается от выводов модуля SK-iMX53-OEM.

Выводы портов общего назначения имеют уровни 3.3В.

### 9.1 Разъем X9

N	Наименование вывода	Номер вывода процессора, дополнительные функции вывода, примечание
1	GND	0B
2	LCD_DCLK	N19: <b>DISP0_CLK</b> , GPIO_4_16 ...
3	LCD_B0	P24: <b>DISP0_DAT0</b> , GPIO_4_21, ECSP13 SCLK ...
4	LCD_B1	P22: <b>DISP0_DAT1</b> , GPIO_4_22, ECSP13 MOSI ...
5	LCD_B2	P23: <b>DISP0_DAT2</b> , GPIO_4_23, ECSP13 MISO ...
6	LCD_B3	P21: <b>DISP0_DAT3</b> , GPIO_4_24, ECSP13 SS0 ...
7	LCD_B4	P20: <b>DISP0_DAT4</b> , GPIO_4_25, ECSP13 SS1 ...
8	LCD_B5	R25: <b>DISP0_DAT5</b> , GPIO_4_26, ECSP13 SS2 ...
9	LCD_B6	R23: <b>DISP0_DAT6</b> , GPIO_4_27, ECSP13 SS3 ...
10	LCD_B7	R24: <b>DISP0_DAT7</b> , GPIO_4_28, ECSP13 RDY ...
11	LCD_G0	R22: <b>DISP0_DAT8</b> , GPIO_4_29, PWM1 PWMO, WDOG1_B ...
12	LCD_G1	T25: <b>DISP0_DAT9</b> , GPIO_4_30, PWM2 PWMO, WDOG2_B ...
13	LCD_G2	R21: <b>DISP0_DAT10</b> , GPIO_4_31 ...
14	LCD_G3	T23: <b>DISP0_DAT11</b> , GPIO_5_5 ...
15	LCD_G4	T24: <b>DISP0_DAT12</b> , GPIO_5_6 ...
16	LCD_G5	R20: <b>DISP0_DAT13</b> , GPIO_5_7, AUD5_RXFS ...
17	LCD_G6	U25: <b>DISP0_DAT14</b> , GPIO_5_8, AUD5_RXC ...
18	LCD_G7	T22: <b>DISP0_DAT15</b> , GPIO_5_9, ECSP11 SS1, ECSP12 SS1 ...
19	LCD_R0	T21: <b>DISP0_DAT16</b> , GPIO_5_10, ECSP12 MOSI, AUD5_TXC ...
20	LCD_R1	U24: <b>DISP0_DAT17</b> , GPIO_5_11, ECSP12 MISO, AUD5_TXD ...
21	LCD_R2	V25: <b>DISP0_DAT18</b> , GPIO_5_12, ECSP12 SS0, AUD5_TXFS, AUD5_RXFS ...

22	LCD_R3	U23: <b>DISP0_DAT19</b> , GPIO_5_13, ECSP12 SCLK, AUD5_RXD, AUD5_RXC, WEIM_CS3 ...
23	LCD_R4	U22: <b>DISP0_DAT20</b> , GPIO_5_14, ECSP11 SCLK, AUD4_TXC ...
24	LCD_R5	T20: <b>DISP0_DAT21</b> , GPIO_5_15, ECSP11 MOSI, AUD4_TXD ...
25	LCD_R6	V24: <b>DISP0_DAT22</b> , GPIO_5_16, ECSP11 MISO, AUD4_TXFS ...
26	LCD_R7	W24: <b>DISP0_DAT23</b> , GPIO_5_17, ECSP11 SS0, AUD4_RXD ...
27	LCD_DE	N21: <b>DIO_PIN15</b> , GPIO_4_17, AUD6_TXC ...
28	LCD_VS	N20: <b>DIO_PIN3</b> , GPIO_4_19, AUD6_TXFS ...
29	SPIO_NCS	G20: EIM_DATA20, ECSP14_SS0,GPIO_3_20...
30	LCD_HS	N25: <b>DIO_PIN2</b> , GPIO_4_18, AUD6_TXD ...
31	SPIO_MISO	E23: ECSP14_MISO, GPIO_3_22, SPDIF_OUT, USB_OTG_PWR ...
32	SPIO_CLK	H20: ECSP14_SCLK, GPIO_3_21, I2C1_SCL, SPDIF_IN ...
33	GPIO7_6	A15: GPIO_7_6 ...
34	SPIO_MOSI	G23: ECSP14_MOSI, GPIO_3_28 ...
35	CAN1_TX	B13: <b>FLEXCAN1_TX</b> , GPIO_7_2 ...
36	CAN1_RX	D14: <b>FLEXCAN1_RX</b> , GPIO_7_3 ...
37	I2C3_SCL	F21: I2C3_SCL, GPIO_3_17, ECSP11_MISO ...
38	I2C3_SDA	D24: I2C3_SDA, GPIO_3_18, ECSP11_MOSI ...
39	GND	0B
40	PWM2	T2: PWM1_OUT, WDOG1_B ...

## 9.2 Разъем X11

N	Наименование вывода	Номер вывода процессора, дополнительные функции вывода, примечание
1	USB0_ID	T4: USB_OTG_ID, GPIO_1_1, PWM2_OUT...
2	GNG	0B
3	USB0_N	USB OTG порт DN
4	USB1_N	USB Host порт DN
5	USB0_P	USB OTG порт DP
6	USB1_P	USB Host порт DP
7	USB0_VBUS	USB OTG порт VBUS
8	SD1_DET	B19: <b>GPIO_2_9</b> , PWM3_OUT...
9	SD1_CMD	B21: SD1_CMD, GPIO_1_8, PWM4_OUT ...
10	SD1_CLK	D20: <b>SD1_CLK</b> , GPIO_1_20 ...
11	SD1_D0	A21: <b>SD1_DAT0</b> , GPIO_1_16, GPT_CAPTURE1 ...
12	SD1_D2	E19: <b>SD1_DAT2</b> , GPIO_1_19, PWM2_OUT, GPT_COMPARE2 ...
13	SD1_D1	C20: <b>SD1_DAT1</b> , GPIO_1_17, PWM3_OUT, GPT_CAPTURE2 ...
14	SD1_D3	F18: <b>SD1_DAT3</b> , GPIO_1_21, PWM1_OUT, WDOG2_B ...
15	PCIE_TXP	(SK-iMX53-OEM – SATA)
16	PCIE_RXN	(SK-iMX53-OEM – SATA)
17	PCIE_TXN	(SK-iMX53-OEM – SATA)
18	PCIE_RXP	(SK-iMX53-OEM – SATA)
19	3.3V	«Средняя точка» Ethernet трансформатора, см. схему SK-iMX6-MB
20	GND	0B

21	ETH_RXN	Вывод Ethernet PHY RXM
22	ETH_TXN	Вывод Ethernet PHY TXM
23	ETH_RXP	Вывод Ethernet PHY RXP
24	ETH_TXP	Вывод Ethernet PHY TXM
25	AUD_TXC	W5: KPP COL0, GPIO_4_6, <b>AUD5_TXC</b> , UART4 TXD_MUX, ECSPI1 SCLK ...
26	AUD_MCLK	A17: <b>CCM_CLKO2</b> , GPIO_6_15 ...
27	AUD_TXFS	U7: KPP COL1, GPIO_4_8, <b>AUD5_TXFS</b> , UART5 TXD_MUX, ECSPI1 MISO ...
28	AUD_TXD	V6: KPP ROW0, GPIO_4_7, <b>AUD5_TXD</b> , UART4 RXD_MUX, ECSPI1 MOSI ...
29	AUD_RXD	U6: KPP ROW1, GPIO_4_9, <b>AUD5_RXD</b> , UART5 RXD_MUX, ECSPI1 SS0 ...
30	UART2_TX	R3: GPIO_1_7, UART2_TXD, FLEXCAN1_TX, I2C4_SCL ...
31	UART2_RX	R5: GPIO_1_8, UART2_RXD, FLEXCAN1_RX, I2C4_SDA ...
32	UART3_TX	F22: EMI D24, GPIO_3_24, UART3_TXD, ECSPI1 SS2, ECSPI2 SS2, AUD5_RXFS, UART1 DTR ...
33	UART3_RX	G22: EMI D25, GPIO_3_25, UART3_RXD, ECSPI1 SS3, ECSPI SS3, AUD5_RXC, UART1 DSR ...
34	UART_TX	M1: GPIO_5_28, <b>UART1_TX</b> , ECSPI2_MISO ...
35	UART_RX	M3: GPIO_5_29, <b>UART1_RX</b> , ECSPI2_SS0 ...
36	I2C2_SCL	E22: EMI EB2, GPIO_2_30, ECSPI1 SS0, <b>I2C2_SCL</b> ...
37	I2C2_SDA	C25: GPIO_3_16, <b>I2C2_SDA</b> , ECSPI1_SCLK ...
38	CAN2_TX	E14: GPIO_7_4, UART1 CTS, <b>FLEXCAN2_TX</b> ...
39	CAN2_RX	F14: GPIO_7_5, UART1 RTS, <b>FLEXCAN2_RX</b> ...
40	GND	0B

### 9.3 Разъем X8

N	Наименование вывода	Номер вывода процессора, дополнительные функции вывода, примечание
1	GND	0B
2	PWM1	F17: <b>GPIO_2_10</b> , PWM4_OUT ...
3	LVDS1_0_TXP	LVDS1
4	LVDS1_1_TXP	LVDS1
5	LVDS1_0_TXN	LVDS1
6	LVDS1_1_TXN	LVDS1
7	LVDS1_CLKP	LVDS1
8	LVDS1_2_TXP	LVDS1
9	LVDS1_CLKN	LVDS1
10	LVDS1_2_TXN	LVDS1
11	LVDS1_3_TXP	LVDS1
12	LVDS0_0_TXP	LVDS0
13	LVDS1_3_TXN	LVDS1
14	LVDS0_0_TXN	LVDS0
15	LVDS0_1_TXP	LVDS0
16	LVDS0_2_TXP	LVDS0
17	LVDS0_1_TXN	LVDS0
18	LVDS0_2_TXN	LVDS0
19	LVDS0_CLKP	LVDS0

20	LVDS0_3_TXP	LVDS0
21	LVDS0_CLKN	LVDS0
22	LVDS0_3_TXN	LVDS0
23	CSI_D0	M2: <b>CSIO D12</b> , GPIO_5_30, UART4 TXD_MUX ...
24	CSI_D1	L1: <b>CSIO D13</b> , GPIO_5_31, UART4 RXD_MUX ...
25	CSI_D2	M4: <b>CSIO D14</b> , GPIO_6_0, UART5 TXD_MUX ...
26	CSI_D3	M5: <b>CSIO D15</b> , GPIO_6_1, UART5 RXD_MUX ...
27	CSI_D4	L4: <b>CSIO D16</b> , GPIO_6_2, UART4 RTS ...
28	CSI_D5	L3: <b>CSIO D17</b> , GPIO_6_3, UART4 CTS ...
29	CSI_D6	M6: <b>CSIO D18</b> , GPIO_6_4, UART5 RTS ...
30	CSI_D7	L6: <b>CSIO D19</b> , GPIO_6_5, UART4 CTS ...
31	CSI_HS	P4: <b>CSIO HSYNC</b> , GPIO_5_19 ...
32	CSI_VS	N2: <b>CSIO VSYNC</b> , GPIO_5_21 ...
33	CSI_PCK	P1: <b>CSIO PIXCLK</b> , GPIO_5_18 ...
34	CSI_DE	P3: <b>CSIO DATA EN</b> , GPIO_5_20 ...
35	SPI1_NCS	N3: CSIO D7, GPIO_5_25, <b>ECSPI1_SS0</b> , AUD3_RXD ...
36	GPIO_4_15	V5: <b>GPIO_4_15</b> , FLEXCAN2_RX, USB_OTG_PWR, UART5_CTS ...
37	SPI1_CLK	N1: CSIO D4, GPIO_5_22, <b>ECSPI1_SCLK</b> , AUD3_TXC ...
38	SPI1_MOSI	P2: CSIO D5, GPIO_5_23, <b>ECSPI1_MOSI</b> , AUD3_TXD ...
39	GND	0B
40	SPI1_MISO	N4: CSIO D6, GPIO_5_24, <b>ECSPI1_MISO</b> , AUD3_TXFS ...

#### 9.4 Разъем X10

N	Наименование вывода	Номер вывода процессора, дополнительные функции вывода, примечание
1	+5V	Питающее напряжение 5В
2	+5V	Питающее напряжение 5В
3	+5V	Питающее напряжение 5В
4	+5V	Питающее напряжение 5В
5	GND	0B
6	GND	0B
7	EIM_DA0	L20: EIM_DA0, GPIO_3_0 ...
8	EIM_DA8	L24: EIM_DA8, GPIO_3_8 ...
9	EIM_DA1	J25: EIM_DA1, GPIO_3_1 ...
10	EIM_DA9	M21: EIM_DA9, GPIO_3_9 ...
11	EIM_DA2	L21: EIM_DA2, GPIO_3_2 ...
12	EIM_DA10	M22: EIM_DA10, GPIO_3_10 ...
13	EIM_DA3	K24: EIM_DA3, GPIO_3_3 ...
14	EIM_DA11	M20: EIM_DA11, GPIO_3_11 ...
15	EIM_DA4	L22: EIM_DA4, GPIO_3_4 ...
16	EIM_DA12	M24: EIM_DA12, GPIO_3_12 ...
17	EIM_DA5	L23: EIM_DA5, GPIO_3_5 ...
18	EIM_DA13	M23: EIM_DA13, GPIO_3_13 ...



19	EIM_DA6	K25: EIM_DA6, GPIO_3_6 ...
20	EIM_DA14	N23: EIM_DA14, GPIO_3_14 ...
21	EIM_DA7	L25: EIM_DA7, GPIO_3_7 ...
22	EIM_DA15	N24: EIM_DA15, GPIO_3_15 ...
23	EIM_OE	J24: EIM_OE, GPIO_2_25, ECSPI2 MISO, DI1_PIN7 ...
24	EIM_WAIT	M25: EIM_WAIT, GPIO_5_0, EIM DTACK_B ...
25	EIM_RW	K20: EIM_RW, GPIO_2_26, ECSPI2 SS0, DI1_PIN8 ...
26	EIM_BCLK	N22: EIM_BCLK
27	EIM_EB0	K21: EIM_EB0, GPIO_2_28 ...
28	EIM_LBA	K22: EIM_LBA, GPIO_2_27, ECSPI2 SS1 ...
29	EIM_EB1	K23: EIM_EB1, GPIO_2_29 ...
30	EIM_CS0	H24: EIM_CS0, GPIO_2_23, ECSPI2 SDCLK ...
31	RESET	Логический уровень 3,3В, активировать «открытым коллектором»
32	EIM_CS1	J23: EIM_CS1, GPIO_2_24, ECSPI2 MOSI ...
33	P_L1	Управление светодиодной индикацией Ethernet, см. схему SK-iMX6-MB
34	GPIO3_26	E24: GPIO_3_26, UART2_TX ... (SK-iMX53-OEM - Аналоговый выход RED)
35	P_L2	Управление светодиодной индикацией Ethernet, см. схему SK-iMX6-MB
36	GPIO3_27	E25: GPIO_3_27, UART2_RX ... (SK-iMX53-OEM - Аналоговый выход GREEN)
37	VBAT	Питание внутренних часов реального времени 2.8-3.0В (SK-iMX53-OEM - GPIO_7_11, SPDIF IN1, I2C3 SDA )
38	GPIO2_16	F24: GPIO_2_16 ... (SK-iMX53-OEM - Аналоговый выход BLUE)
39	SPDIF_OUT	R1: GPIO_7_12, SPDIF OUT ...
40	GND	0В

## 9.5. HDMI

На верхней стороне модуля предусмотрено посадочное место для разъема гибкого шлейфа X2 (HDMI) с шагом выводов 1мм, на его место можно установить разъем или припаять кабель один из концов которого предназначен под пайку.

## 9.6. Дополнительные разъемы

На верхней стороне модуля имеется четыре контакта:

X3 – PCIE\_RP (CPU D7) – тактовый сигнал для Mini PCIe интерфейса

X6 – PCIE\_RM (CPU C7) – тактовый сигнал для Mini PCIe интерфейса

X5 – подключен к выводу процессора D12 “ONOFF”

X7 – подключен к выводу процессора D11 “PMIC\_ON\_REQ”

## 10. Дополнительные материалы

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