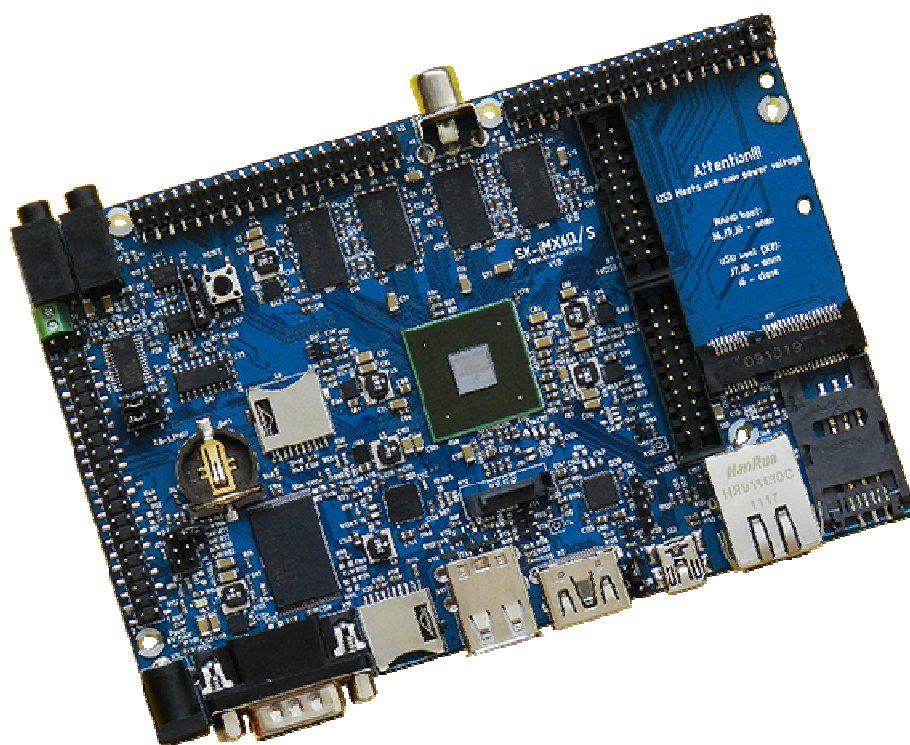


Отладочная плата SK-iMX6Q

Инструкция пользователя



SK-iMX6Q:

Freescall iMX6Q (4 ядра ARM Cortex-A9 1000МГц)
DDR3-1066 1024Мбайт (128Мx64)
NAND Flash 2Гбайт
Ethernet 1000/100/10М
Mini PCIe разъем, SIM card держатель
SATA разъем
HDMI выход
2 x LVDS с возможностью подключения сенсорных экранов
Audio CODEC вход/выход
SPDIF вход/выход
2 x uSD разъема
4 x USB Host (3 порта на разъемы, 1 на разъем PCIe), USB OTG
RS232
2 x CAN
RTC (часы реального времени), держатель батареи 1220
Разъемы расширения

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-iMX6Q, 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

Первый вывод перемычек и переключающих перемычек помечен квадратной контактной площадкой.

- J1, J3 позволяет подключить согласующий резистор к CAN линии
- J2 линейный вход аудиокодека
- J3 штыревой разъем для подключения линии CAN интерфейса
- J4,J5 определяют функцию разъема X4, положение 1-2 – выход на наушники, 2-3 – линейный выход
- J6,J7,J8 определяют источник загрузки процессора:
 - замкнут J6 – загрузка с uSD карты (разъем X8)
 - разомкнуты все – загрузка с NAND Flash
- J9 управляет подачей питающего напряжения к X22 (USB OTG)
По умолчанию замкнуты перемычки: J1,J3 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-iMX6Q”

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 2009.08 (Apr 30 2013 - 20:38:18)

CPU: Freescale i.MX6 family T01.2 at 792 MHz
Thermal sensor with ratio = 187
Temperature: 45 C, calibration data 0x5a24f069
mx6q pll1: 792MHz
mx6q pll2: 528MHz
mx6q pll3: 480MHz
mx6q pll8: 50MHz
ipg clock      : 660000000Hz
ipg per clock : 660000000Hz
uart clock    : 800000000Hz
cspi clock    : 600000000Hz
ahb clock     : 1320000000Hz
axi clock     : 2640000000Hz
emi_slow clock: 1320000000Hz
ddr clock     : 5280000000Hz
usdhc1 clock  : 1980000000Hz
usdhc2 clock  : 1980000000Hz
usdhc3 clock  : 1980000000Hz
usdhc4 clock  : 1980000000Hz
nfc clock     : 1980000000Hz
Board: SK-iMX6Q:[ POR]
Boot Device: NAND
I2C:  ready
DRAM:  1 GB
```

```

NAND: Manufacturer ID: 0xec, Chip ID: 0xd5 (Samsung NAND 2GiB 3,3V 8-bit), page size: 8192,
OOB size: 436 2048 MiB
MMC: FSL_USDHC: 0,FSL_USDHC: 1
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: 4118116 Bytes = 3.9 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) ) #73 SMP PREEMPT Wed May 1 02:01:10
UTC 2013
CPU: ARMv7 Processor [412fc09a] revision 10 (ARMv7), cr=10c53c7d
CPU: VIPT nonaliasing data cache, VIPT aliasing instruction cache
Machine: Freescale i.MX 6Quad Sabre-Lite Board
Memory policy: ECC disabled, Data cache writealloc
CPU identified as i.MX6Q, silicon rev 1.2
PERCPU: Embedded 7 pages/cpu @8c008000 s5440 r8192 d15040 u32768
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 227328
Kernel command line: console=ttyMXC0,115200 ubi.mtd=4 root=ubi0:nandfs rootwait rw
rootfstype=ubifs video=mxcfb3:dev=ldb,SK-ATM0704,if=RGB24 video=mxcfb2:dev=ldb,LDB-
XGA,if=RGB24 video=mxcfb1:dev=lcd,SK-MI0430,if=RGB24 video=mxcfb0:dev=hdmi,1920x1080M@60,
if=RGB24
PID hash table entries: 4096 (order: 2, 16384 bytes)
Dentry cache hash table entries: 131072 (order: 7, 524288 bytes)
Inode-cache hash table entries: 65536 (order: 6, 262144 bytes)
Memory: 640MB 256MB = 896MB total
Memory: 896416k/896416k available, 152160k reserved, 0K highmem
Virtual kernel memory layout:
vector : 0xffff0000 - 0xffff1000 ( 4 kB)
fixmap : 0xffff0000 - 0xffffe000 ( 896 kB)
DMA : 0xf4600000 - 0xffe00000 ( 184 MB)
vmalloc : 0xc0800000 - 0xf2000000 ( 792 MB)
lowmem : 0x80000000 - 0xc0000000 (1024 MB)
pkmap : 0x7fe00000 - 0x80000000 ( 2 MB)
modules : 0x7f000000 - 0x7fe00000 ( 14 MB)
 .init : 0x80008000 - 0x8003c000 ( 208 kB)
 .text : 0x8003c000 - 0x80afc2f8 (11009 kB)
 .data : 0x80afe000 - 0x80b5c820 ( 379 kB)
 .bss : 0x80b5c844 - 0x80baaf9c ( 314 kB)
SLUB: Genslabs=13, HWalign=32, Order=0-3, MinObjects=0, CPUs=4, Nodes=1
Preemptible hierarchical RCU implementation.
NR_IRQS:496
MXC GPIO hardware
sched_clock: 32 bits at 3000kHz, resolution 333ns, wraps every 1431655ms
arm_max_freq=2
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
CPU: Testing write buffer coherency: ok
hw perfevents: enabled with ARMv7 Cortex-A9 PMU driver, 7 counters available
CPU1: Booted secondary processor
CPU2: Booted secondary processor
CPU3: Booted secondary processor
Brought up 4 CPUs
SMP: Total of 4 processors activated (6324.22 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 1100 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: vdd1p1: 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 0x410000c7, AUX_CTRL 0x02070000, Cache size: 1048576 B
bio: create slab <bio-0> at 0
mxs-dma mxs-dma-apbh: initialized
print_constraints: vmmc: 3300 mV
vgaarb: loaded
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
Freescale USB OTG Driver loaded, $Revision: 1.55 $
imx-ipuv3 imx-ipuv3.0: IPU DMFC NORMAL mode: 1(0~1), 5B(4,5), 5F(6,7)
imx-ipuv3 imx-ipuv3.1: 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
cfg80211: Calling CRDA to update world regulatory domain
i2c-core: driver [max17135] using legacy suspend method
i2c-core: driver [max17135] using legacy resume method
Switching to clocksource mxc_timer1
NET: Registered protocol family 2
IP route cache hash table entries: 32768 (order: 5, 131072 bytes)
TCP established hash table entries: 131072 (order: 8, 1048576 bytes)
TCP bind hash table entries: 65536 (order: 7, 786432 bytes)
TCP: Hash tables configured (established 131072 bind 65536)
TCP reno registered
UDP hash table entries: 512 (order: 2, 16384 bytes)
UDP-Lite hash table entries: 512 (order: 2, 16384 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
_regulator_get: etm.1 supply vcore not found, using dummy regulator
_regulator_get: etm.2 supply vcore not found, using dummy regulator
_regulator_get: etm.3 supply vcore not found, using dummy regulator
Static Power Management for Freescale i.MX6
wait mode is enabled for i.MX6
cpaddr = c0880000 suspend_iram_base=c0934000
PM driver module loaded
IMX PCIe port: link down!
IMX usb wakeup probe
add wake up source irq 75
IMX usb wakeup probe
cpu regulator init ldo=0
i.MXC CPU frequency driver
JFFS2 version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
msgmni has been set to 1750
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 hdmi
mxc_hdmi mxc_hdmi: Detected HDMI controller 0x13:0xa:0xa0:0xc1
fbcvt: 1920x1080@60: CVT Name - 2.073M9
imx-ipuv3 imx-ipuv3.0: IPU DMFC DP HIGH RESOLUTION: 1(0,1), 5B(2~5), 5F(6,7)
Console: switching to colour frame buffer device 240x67
mxc_sdc_fb mxc_sdc_fb.1: register mxc display driver lcd
mxc_sdc_fb mxc_sdc_fb.2: register mxc display driver ldb
_regulator_get: get() with no identifier
mxc_sdc_fb mxc_sdc_fb.3: register mxc display driver ldb
fbcvt: Invalid input parameters
mxc_sdc_fb mxc_sdc_fb.4: register mxc display driver ldb
fbcvt: Invalid input parameters
mxc_sdc_fb mxc_sdc_fb.4: ipu1-di1 already in use
mxc_sdc_fb: probe of mxc_sdc_fb.4 failed with error -16
mxc_sdc_fb mxc_sdc_fb.5: register mxc display driver lcd
mxc_sdc_fb mxc_sdc_fb.5: ipu0-di0 already in use
mxc_sdc_fb: probe of mxc_sdc_fb.5 failed with error -16
mxc_sdc_fb mxc_sdc_fb.6: register mxc display driver lcd
fbcvt: Invalid input parameters
mxc_sdc_fb mxc_sdc_fb.6: ipu0-di0 already in use

```

```

mxs_sdc_fb: probe of mxs_sdc_fb.6 failed with error -16
mxs_sdc_fb mxs_sdc_fb.7: register mxs display driver lcd
mxs_sdc_fb mxs_sdc_fb.7: ipu0-di0 already in use
mxs_sdc_fb: probe of mxs_sdc_fb.7 failed with error -16
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
Can't recover from RESET HBA!
No sata disk.
NAND device: Manufacturer ID: 0xec, Chip ID: 0xd5 (Samsung NAND 2GiB 3,3V 8-bit)
Creating 5 MTD partitions on "gpmi-nand":
0x0000000000000-0x000001000000 : "U-boot partition"
0x0000010000000-0x000001100000 : "U-boot environment partition"
0x0000011000000-0x000001800000 : "Linux kernel partition"
0x0000018000000-0x000002400000 : "Linux safe kernel partition (rootfs in initramfs)"
0x0000024000000-0x0000080000000 : "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: 83
UBI: attached mtd4 to ubi0
UBI: MTD device name: "NAND rootfs partition"
UBI: MTD device size: 2012 MiB
UBI: number of good PEBs: 2007
UBI: number of bad PEBs: 5
UBI: number of corrupted PEBs: 0
UBI: max. allowed volumes: 128
UBI: wear-leveling threshold: 256
UBI: number of internal volumes: 1
UBI: number of user volumes: 1
UBI: available PEBs: 134
UBI: total number of reserved PEBs: 1873
UBI: number of PEBs reserved for bad PEB handling: 40
UBI: max/mean erase counter: 1/0
UBI: image sequence number: 25825458
UBI: background thread "ubi_bgt0d" started, PID 1263
vcan: Virtual CAN interface driver
CAN device driver interface
flexcan netdevice driver
flexcan imx6q-flexcan.0: device registered (reg_base=c0e78000, irq=142)
flexcan imx6q-flexcan.1: device registered (reg_base=c0e80000, irq=143)
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: spi0.0 supply vcc not found, using dummy regulator
ads7846 spi0.0: touchscreen, irq 263
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 362
input: ADS7846 Touchscreen as /devices/platform/imx6q-ecspi.3/spi_master/spi3/spi3.0/input/input1
_regulator_get: spi4.0 supply vcc not found, using dummy regulator
ads7846 spi4.0: touchscreen, irq 264
input: ADS7846 Touchscreen as /devices/platform/spi_gpio.4/spi_master/spi4/spi4.0/input/input2
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
DBG sensor data is at 80b7e010
mxv_v4l2_output mxv_v4l2_output.0: V4L2 device registered as video16
mxv_v4l2_output mxv_v4l2_output.0: V4L2 device registered as video17
mxv_v4l2_output mxv_v4l2_output.0: V4L2 device registered as video18
mxv_v4l2_output mxv_v4l2_output.0: V4L2 device registered as video19
mxv_v4l2_output mxv_v4l2_output.0: V4L2 device registered as video20
mxv_v4l2_output mxv_v4l2_output.0: V4L2 device registered as video21
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 DMA
sdhci sdhci-esdhc-imx.1: no write-protect pin available!
mmc1: SDHCI controller on platform [sdhci-esdhc-imx.1] using DMA
mxv_vdoa mxv_vdoa: i.MX Video Data Order Adapter(VDOA) driver probed
VPU initialized
mxv_asrc registered
usb 2-1: new high speed USB device number 2 using fsl-ehci
Thermal calibration data is 0x5a24f069
Thermal sensor with ratio = 187
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
mxv_hdmi_soc mxv_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
AIC23 Audio Codec 0.1
asoc: tlv320aic23-hifi <-> imx-ssi.1 mapping ok
imx_3stack asoc driver
asoc: mxv-hdmi-soc <-> imx-hdmi-soc-dai.0 mapping ok
ALSA device list:
  #0: sgtl5000-audio
  #1: imx-hdmi-soc
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
hub 2-1:1.0: USB hub found
Bus freq driver module loaded
Bus freq driver Enabled
mxv_dvfs_core_probe
DVFS driver module loaded
hub 2-1:1.0: 4 ports detected
snvs_rtc snvs_rtc.0: setting system clock to 1970-01-01 00:00:00 UTC (0)
mmc1: host does not support reading read-only switch. assuming write-enable.
mmc1: new high speed SDHC card at address 0007
mmcblk0: mmc1:0007 SD4GB 3.70 GiB
  mmcblk0: p1 p2
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: mkdir: can't create directory '/mnt/mmcblk0p1': File exists
mkdir: can't create directory '/mnt/mmcblk0p2': File exists

```



```
EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
rm: can't remove '/bin/ip': No such file or directory
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 ...
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

Welcome to SK-iMX6Q development board!
PHY: 1:00 - Link is Up - 1000/Full
SK-iMX6Q 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, 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

Процессор iMX6Q позволяет подключить до четырех отдельных устройств отображения графической информации, 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 – HDMI выход, разрешение 1920x1080, консоль /dev/tty1

/dev/fb1 – прозрачное наложение на fb0

/dev/fb2 – LCD выход (X1), разрешение 480x272

/dev/fb3 – LVDS0 выход (X10), разрешение 1024x768

/dev/fb4 – прозрачное наложение на fb0

/dev/fb5 – LVDS0 выход (X7), разрешение 800x480

Модули расширения необходимо подключать при отсутствующем напряжении питания.

3.1.1 Подключение SK-ATM0700D4-Plug к разъему X7

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 X10

Для использования сенсорного экрана, необходимо внести изменения в файле /etc/profile в соответствии с комментариями. Перед началом работы необходимо откалибровать сенсорный экран – запустите `ts_calibrate`.

3.1.3 Подключение SK-MI0430FT-Plug к разъему X1

Для использования сенсорного экрана, необходимо внести изменения в файле /etc/profile в соответствии с комментариями. Перед началом работы необходимо откалибровать сенсорный экран – запустите `ts_calibrate`.

4. Состав ОС Linux

Ядро 3.0.35, включая драйвера:

- Ethernet
- NAND flash
- SATA
- PCIe
- SD-card
- USB-host
- USB-OTG
- I2C
- SPI
- UART
- RTC
- CAN
- WatchDog
- fb
- TP ADS7846
- ...

5. Способы загрузки и содержимое корневой файловой системы

iMX6Q подразумевает различные возможные источники загрузки, на плате предусмотрено четыре варианта: NAND flash, uSD карта, SATA HDD, USB

5.1. NAND flash

В штатной поставке система сконфигурирована на загрузку из NAND flash, перемычки J6-J8 должны быть разомкнуты.

NAND flash разбита на 5 частей:

- 1) 0-0x1000000 – загрузчик u-boot
- 2) 0x1000000-0x1100000 – область хранения переменных окружения загрузчика
- 3) 0x1100000-0x1800000 – ядро Linux
- 4) 0x1800000-2400000 – ядро Linux со встроенной корневой файловой системой
- 5) 0x2400000-0x80000000 – область корневой файловой системы

5.2. uSD

Предварительно карта памяти должна быть подготовлена скриптом «/home/user/src/rootfs/linaro_fs/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 Ubuntu.

Подготовленную карту необходимо вставить в разъем X8, замкнуть J6 (разомкнуть J7 и J8), после чего включить питание. По умолчанию, загрузчик начнет копировать и запускать файлы с NAND flash, для загрузки с uSD карты, необходимо прервать загрузчик и выполнить команду «run mmc_boot».

5.3. SATA

Аппаратная возможность загрузки с SATA HDD предусмотрена, но в настоящий момент она не проверена.

5.4. USB

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.

6. Корневая файловая система

6.1. Buildroot 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 виртуальной машины и распакован архив корневой ФС.

Следует учесть, ядро запускаемое командой «safe_boot», при старте анализирует присутствие и запускает скрип «system_prepare_imx6q» в первом разделе uSD карты, который в свою очередь восстанавливает систему.

6.2. Ubuntu корневая файловая система

После подготовки uSD карты скриптом sd_prepare.sh, на нее копируется Ubuntu ФС. Для ее загрузки, необходимо прервать загрузчик u-boot и выполнить команду «run mmc_boot». Рабочий стол системы настроен на HDMI выход, при первом запуске необходимо настроить сетевой интерфейс под реалии вашего сетевого окружения (настроить сетевой адрес и gateway), после чего можно будет необходимые пакеты в систему. В штатном виде настроен: SSH, FTP ...

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-iMX6Q_linux_build_machine.rar”, установите VMware-player или VMware, откройте и проект виртуальной машины (на вопрос, скопирована система или перемещена необходимо ответить - перемещена).

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.XXX

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 (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 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

Суперпользователя root в виртуальной машине нет, для действий с его привилегиями необходимо 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.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

7.1. 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
- прерываем в u-boot процесс загрузки нажатием любой клавиши
- выполняем “run safe_boot”
- после загрузки системы, выполняем “uboot_update”

9. Восстановление системы

На случай необходимости восстановления системы предусмотрено два варианта: загрузка по USB с последующей подготовкой системы, загрузка с uSD карты с последующей подготовкой системы.

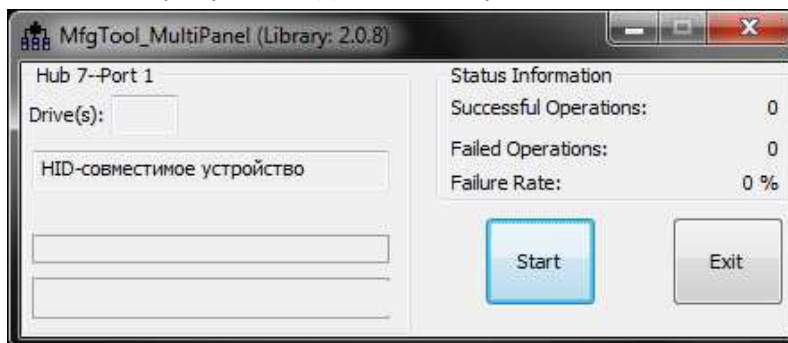
9.1. Восстановление через USB OTG интерфейс

Предварительно необходимо распаковать Mfg-tool утилиту (архив Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER_SK.rar). Подключить USB кабель к разъему X22, подключить RS232 кабель и запустить терминальную программу.

Произвести манипуляции в соответствии с пунктом 5.4. (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_imx6q», после копирует и распакует файловую систему во второй раздел NAND flash.

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

9.2. Восстановление с uSD

Предварительно необходимо подготовить uSD карту скриптом «/home/user/src/rootfs/linaro_fs/sd_prepare.sh», произвести манипуляции в соответствии с пунктом 5.2 (загрузка с uSD карты), подключить RS232 кабель.

Вставить uSD карту в разъем X8, включить питание.

Прервать работу загрузчика u-boot нажатием любой клавиши.

Выполнить команду «run safe_boot»

Во время старта системы запускается скрипт system_prepare_imx6q с первого раздела uSD карты, который копирует исполняемые файлы в NAND flash, а так же форматирует и распаковывает корневую файловую систему.

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

10. 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
## Booting kernel from Legacy Image at 10800000 ...
```

Аргументы запуска ядра хранятся в переменных окружения загрузчика.

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 10800000 ...
Image Name:   Linux-3.0.35
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    4118116 Bytes = 3.9 MB
Load Address: 10008000
Entry Point:  10008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
OK
```

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