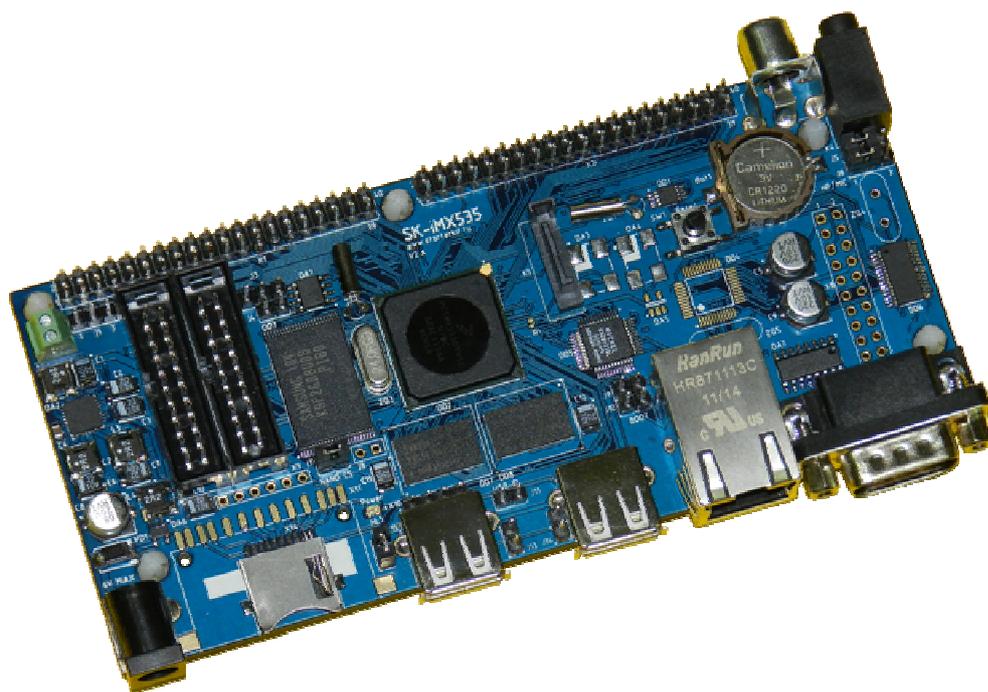


Отладочная плата SK-iMX53

Инструкция пользователя



SK-iMX53:

FreeScale iMX535 (ARM Cortex-A8 1000МГц)
DDR2 256Мбайт (64Мx32)
SLC NAND Flash 256Мбайт
Ethernet 100/10М
Audio CODEC вход/выход
Audio SPDIF выход
uSD разъем
SATA разъем
USB Host/Device
USB Host
RS232
CAN PHY
RTC (часы реального времени), держатель батареи 1220
Разъемы расширения
Система питания

Возможность прямого подключения:

SK-MI0430FT-Plug или аналог – плата расширения LCD TFT 4,3” панелей
SK-ATM0700D4-Plug или аналог – плата расширения LCD TFT 7” панелей
SK-HDMI-Plug – плата расширения HDMI выхода
SK-SIMCOM-Plug – плата расширения GSM/GPS/3G модулей

Комплект поставки: отладочная плата SK-iMX53, USB-A-USB-A кабель, RS232 кабель, ссылка для скачивания необходимых материалов

1. Общие характеристики

- Напряжение питания: 5-6В (центральная жила разъема), 6В максимум в следующих случаях:
 - 1) при использовании USB
 - 2) при совместном использовании SK-ATM0700D4-Plug
 - 3) при питании внешнего SATA HDD от разъема платы
- Потребляемый ток платы - до 1А, следует учитывать потребление подключаемой внешней периферии.
- Габариты 138x64x20мм

2. Назначение джамперов

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

- J1 позволяет выбрать какой из сигналов (VS или FIELD) будет использован модулем CSI в качестве синхросигнала кадровой развертки
- J2 разъем для подключения питания SATA HDD
- J3 штыревой разъем для подключения линии CAN интерфейса
- J4 позволяет подключить согласующий резистор к CAN линии
- J5,J6 определяют функцию разъема X4, положение 1-2 – выход на наушники, 2-3 – микрофонный вход
- J7 позволяет исключить NAND Flash из системы, актуально для загрузки по USB
- J11 позволяет управлять уровнем ID сигнала для USB-OTG
- J12 позволяет подключить питание к USB разъему X15 минуя транзисторные каскады управления
- J13 позволяет подключить питание к USB разъему X16 минуя транзисторные каскады управления, следует учесть дальнейшее наличие в цепи J14
- J14 управляет подачей питающего напряжения к X16, **ВНИМАНИЕ!!!** В режиме работы порта как Device (например, при USB загрузке) должен быть разомкнут
По умолчанию замкнуты перемычки: J4, J5 и J6 положение 1-2, J7

3. Начало работы

Перед началом работы убедитесь в положении джамперов (см. выше), так же следует ознакомиться со всеми материалами имеющих статус «Важная тема» или «Объявление» на форуме starterkit.ru в разделе “Отладочные платы > SK-iMX53”

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.

Подключите питание, в терминальной программе появятся аналогичные сообщения:

```
U-Boot 2009.08 (Nov 23 2011 - 19:11:25)
CPU: Freescale i.MX53 family 2.1V at 1000 MHz
mx53 pll1: 1000MHz
mx53 pll2: 400MHz
mx53 pll3: 216MHz
mx53 pll4: 455MHz
ipg clock : 66666666Hz
```

```
ipg per clock : 33333333Hz
uart clock    : 66666666Hz
cspi clock    : 54000000Hz
ahb clock     : 133333333Hz
axi_a clock   : 400000000Hz
axi_b clock   : 200000000Hz
emi_slow clock: 133333333Hz
ddr clock     : 400000000Hz
esdhc1 clock  : 80000000Hz
esdhc2 clock  : 80000000Hz
esdhc3 clock  : 80000000Hz
esdhc4 clock  : 80000000Hz
nfc clock     : 33333333Hz
Board: MX53-SK Rev. A
Boot Reason: [unknown]
Boot Device: NAND
DRAM: 256 MB
NAND: Manufacturer      : Samsung (0xec)
Device Code           : 0xda
Cell Technology       : SLC
Chip Size             : 256 MiB
Pages per Block      : 64
Page Geometry        : 2048+64
ECC Strength         : 4 bits
ECC Size             : 512 B
Data Setup Time      : 20 ns
Data Hold Time       : 10 ns
Address Setup Time   : 20 ns
GPMI Sample Delay    : 6 ns
tREA                 : Unknown
tRLOH                : Unknown
tRHOH                : Unknown
Description           : K9F2G08U0A
Bad block table found at page 131008, version 0x01
Bad block table found at page 130944, version 0x01
nand_read_bbt: Bad block at 0x00000bae0000
256 MiB
MMC: FSL_ESDHC: 0
Using default environment
```

```
In: serial
Out: serial
Err: serial
Net: FEC0 [PRIME]
Hit any key to stop autoboot: 0
```

```
NAND read: device 0 offset 0x1000000, size 0xa00000
10485760 bytes read: OK
## Booting kernel from Legacy Image at 70800000 ...
Image Name: linux-2.6
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 2455412 Bytes = 2.3 MB
Load Address: 70008000
Entry Point: 70008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
OK
```

Starting kernel ...

```
Linux version 2.6.35.3-1129-g691c08a (user@imx535-bld) (gcc version 4.4.1 (Sourcery G++ Lite
2009q3-67) ) #253 PREEMPT Sun Nov 27 15:40:20 UTC 2011
CPU: ARMv7 Processor [412fc085] revision 5 (ARMv7), cr=10c53c7f
CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
Machine: Freescale MX53 LOCO Board
Ignoring unrecognised tag 0x54410009
Memory policy: ECC disabled, Data cache writeback
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 56896
Kernel command line: noinitrd console=ttymx0,115200 ubi.mtd=1 root=ubi0:nandfs
rw rootfstype=ubifs video=mxcdi0fb:RGB888,SK-800x480-LVDS video=mxcdi1fb:RGB888,SK-800x480-
LVDS di0_primary ldb=di0
PID hash table entries: 1024 (order: 0, 4096 bytes)
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 224MB = 224MB total
Memory: 222376k/222376k available, 7000k reserved, 0K highmem
Virtual kernel memory layout:
vector : 0xffff0000 - 0xffff1000 ( 4 kB)
fixmap : 0xffff0000 - 0xffffe000 ( 896 kB)
DMA : 0xf9e00000 - 0xffe00000 ( 96 MB)
vmalloc : 0x8e800000 - 0xf4000000 (1624 MB)
lowmem : 0x80000000 - 0x8e000000 ( 224 MB)
pkmap : 0x7fe00000 - 0x80000000 ( 2 MB)
```

```

modules : 0x7f000000 - 0x7fe00000 ( 14 MB)
 .init : 0x80008000 - 0x80028000 ( 128 kB)
 .text : 0x80028000 - 0x80463000 (4332 kB)
 .data : 0x80480000 - 0x804bcc60 ( 244 kB)
SLUB: Genslabs=11, HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
Hierarchical RCU implementation.
 RCU-based detection of stalled CPUs is disabled.
 Verbose stalled-CPU detection is disabled.
NR_IRQS:368
MXC GPIO hardware
MXC IRQ initialized
MXC_Early serial console at MMIO 0x53fbc000 (options '115200')
bootconsole [ttymx0] enabled
Console: colour dummy device 80x30
Calibrating delay loop... 999.42 BogoMIPS (lpj=4997120)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
regulator: core version 0.5
regulator: dummy:
NET: Registered protocol family 16
i.MX IRAM pool: 128 KB@0x8e840000
IRAM READY
CPU is i.MX53 Revision 2.1
Using SDMA I.API
MXC DMA API initialized
IMX usb wakeup probe
IMX usb wakeup probe
bio: create slab <bio-0> at 0
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
IPU DMFC NORMAL mode: 1(0~1), 5B(4,5), 5F(6,7)
Advanced Linux Sound Architecture Driver Version 1.0.23.
Switching to clocksource mxc_timer1
NET: Registered protocol family 2
IP route cache hash table entries: 2048 (order: 1, 8192 bytes)
TCP established hash table entries: 8192 (order: 4, 65536 bytes)
TCP bind hash table entries: 8192 (order: 3, 32768 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
TCP reno registered
UDP hash table entries: 256 (order: 0, 4096 bytes)
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
NET: Registered protocol family 1
LPMODE driver module loaded
Static Power Management for Freescale i.MX5
PM driver module loaded
sdram autogating driver module loaded
Bus freq driver module loaded
DIO is primary
mxc_dvfs_core_probe
regulator: get() with no identifier
mxc_dvfs_core_probe: failed to get gp regulator
DVFS driver module loaded
i.MXC CPU frequency driver
regulator: get() with no identifier
mxc_cpufreq_driver_init: failed to get gp regulator
DVFS PER driver module loaded
msgmni has been set to 434
alg: No test for stdrng (krng)
cryptodev: driver loaded.
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
regulator: get() with no identifier
mxc_ipu mxc_ipu: Channel already disabled 9
mxc_ipu mxc_ipu: Channel already uninitialized 9
Console: switching to colour frame buffer device 100x30
mxc_ipu mxc_ipu: Channel already disabled 7
mxc_ipu mxc_ipu: Channel already uninitialized 7
mxc_ipu mxc_ipu: Channel already disabled 10
mxc_ipu mxc_ipu: Channel already uninitialized 10
Serial: MXC Internal UART driver
mxcintuart.0: ttymx0 at MMIO 0x53fbc000 (irq = 31) is a Freescale i.MX
console [ttymx0] enabled, bootconsole disabled
console [ttymx0] enabled, bootconsole disabled
mxcintuart.1: ttymx1 at MMIO 0x53fc0000 (irq = 32) is a Freescale i.MX
mxcintuart.2: ttymx2 at MMIO 0x5000c000 (irq = 33) is a Freescale i.MX
mxcintuart.3: ttymx3 at MMIO 0x53ff0000 (irq = 13) is a Freescale i.MX
mxcintuart.4: ttymx4 at MMIO 0x63f90000 (irq = 86) is a Freescale i.MX
loop: module loaded
ahci: SSS flag set, parallel bus scan disabled

```

```

ahci ahci.0: AHCI 0001.0100 32 slots 1 ports 3 Gbps 0x1 impl platform mode
ahci ahci.0: flags: ncq sntf stag pm led clo only pmp pio slum part ccc
scsi0 : ahci
ata1: SATA max UDMA/133 irq_stat 0x00000040, connection status changed irq 28
MXC MTD NAND Driver 3.0
NAND device: Manufacturer ID: 0xec, Chip ID: 0xda (Samsung NAND 256MiB 3,3V 8-bit)
RedBoot partition parsing not available
Creating 2 MTD partitions on "NAND 256MiB 3,3V 8-bit":
0x000000000000-0x000002000000 : "bootloader and kernel"
0x000002000000-0x000010000000 : "nand.rootfs"
UBI: attaching mtd1 to ubi0
UBI: physical eraseblock size: 131072 bytes (128 KiB)
UBI: logical eraseblock size: 129024 bytes
UBI: smallest flash I/O unit: 2048
UBI: sub-page size: 512
UBI: VID header offset: 512 (aligned 512)
UBI: data offset: 2048
UBI: attached mtd1 to ubi0
UBI: MTD device name: "nand.rootfs"
UBI: MTD device size: 224 MiB
UBI: number of good PEBs: 1787
UBI: number of bad PEBs: 5
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: 140
UBI: total number of reserved PEBs: 1647
UBI: number of PEBs reserved for bad PEB handling: 17
UBI: max/mean erase counter: 1/0
UBI: image sequence number: 696306421
UBI: background thread "ubi_bgt0d" started, PID 467
vcan: Virtual CAN interface driver
Freescale FlexCAN Driver
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 18, io base 0x53f80000
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
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 14, io base 0x53f80200
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
usbcore: registered new interface driver cdc_acm
cdc_acm: v0.26:USB Abstract Control Model driver for USB modems and ISDN adapters
Initializing USB Mass Storage driver...
ata1: SATA link up 1.5 Gbps (SStatus 113 SControl 300)
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
mice: PS/2 mouse device common for all mice
spi0.0 supply vcc not found, using dummy regulator
ads7846 spi0.0: touchscreen, irq 214
input: ADS7843 Touchscreen as /devices/platform/spi_gpio.0/spi0.0/input/input0
spi1.0 supply vcc not found, using dummy regulator
ads7846 spi1.0: touchscreen, irq 211
input: ADS7843 Touchscreen as /devices/platform/spi_gpio.1/spi1.0/input/input1
spi2.0 supply vcc not found, using dummy regulator
ads7846 spi2.0: touchscreen, irq 166
input: ADS7843 Touchscreen as /devices/platform/spi_gpio.2/spi2.0/input/input2
rtc-ds1307 1-0068: rtc core: registered ds1338 as rtc0
rtc-ds1307 1-0068: 56 bytes nvram
mxc_rtc mxc_rtc.0: rtc core: registered mxc_rtc as rtc1
i2c /dev entries driver
Linux video capture interface: v2.00
mxc_v4l2_output mxc_v4l2_output.0: Registered device video0
usbcore: registered new interface driver uvcvideo
USB Video Class driver (v0.1.0)
MXC WatchDog Driver 2.0
MXC Watchdog # 0 Timer: initial timeout 60 sec
VPU initialized
mxc_asrc registered
gpu mmu enabled
mxsdhci: MXC Secure Digital Host Controller Interface driver
mxsdhci: MXC SDHCI Controller Driver.
mmc0: SDHCI detect irq 206 irq 1 INTERNAL DMA
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver

```

```

No device for DAI tlv320aic23
No device for DAI imx-ssi-1-0
No device for DAI imx-ssi-1-1
No device for DAI imx-ssi-2-0
No device for DAI imx-ssi-2-1
AIC23 Audio Codec 0.1
DMA Sound Buffer Allocated: Playback UseIram=1 ext_ram=0 buf->addr=f8016000 buf-
>area=8e856000 size=24576
DMA Sound Buffer Allocated: Capture UseIram=1 ext_ram=1 buf->addr=7d240000 buf-
>area=fa8c3000 size=24576
asoc: tlv320aic23 <-> imx-ssi-2-0 mapping ok
ALSA device list:
 #0: imx-3stack (tlv320aic23)
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)
VFP support v0.3: implementor 41 architecture 3 part 30 variant c rev 2
rtc-ds1307 1-0068: setting system clock to 2011-11-30 14:49:33 UTC (1322664573)ata1.00: ATA-
8: ST9160314AS, 0010LVM1, max UDMA/100
ata1.00: 312581808 sectors, multi 16: LBA48 NCQ (depth 31/32)
ata1.00: configured for UDMA/100
scsi 0:0:0:0: Direct-Access ATA ST9160314AS 0010 PQ: 0 ANSI: 5
sd 0:0:0:0: [sda] 312581808 512-byte logical blocks: (160 GB/149 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI disk
UBIFS: mounted UBI device 0, volume 0, name "nandfs"
UBIFS: file system size: 208373760 bytes (203490 KiB, 198 MiB, 1615 LEBs)
UBIFS: journal size: 10450944 bytes (10206 KiB, 9 MiB, 81 LEBs)
UBIFS: media format: w4/r0 (latest is w4/r0)
UBIFS: default compressor: lzo
UBIFS: reserved for root: 4952683 bytes (4836 KiB)
VFS: Mounted root (ubifs filesystem) on device 0:10.
Freeing init memory: 128K
Init: EXT3-fs: barriers not enabled
EXT3-fs (sda1): warning: maximal mount count reached, running e2fsck is recommended
kjournald starting. Commit interval 5 seconds
EXT3-fs (sda1): using internal journal
EXT3-fs (sda1): mounted filesystem with writeback data mode
Starting logging: OK
Initializing random number generator... done.
Starting network...
eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy_addr=0:01, irq=-1)
PHY: 0:01 - Link is Up - 100/Full

Welcome to SK-iMX53

buildroot login:

```

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

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

Если был подключен SK-ATM0700D4-Plug (к любому или обоим LVDS разъемам), на экране появится графическое изображение и сообщение о старте системы, при первой загрузке, необходимо откалибровать сенсорную панель.

4. Состав ОС Linux

Ядро 2.6.35.3, включая драйвера:

- Ethernet
- NAND flash
- SATA
- SD-card

- USB-host
- USB-gadget
- LVDS
- I2C
- SPI
- UART
- RTC
- CAN
- WatchDog
- Frame Buffer
- TP ADS7843
- ...

5. Способы загрузки и содержимое корневой файловой системы

iMX53X подразумевает различные возможные источники загрузки, на плате предусмотрено два - NAND flash и USB

5.1. NAND flash

NAND flash разбита на две части:

1) 32M – для хранения загрузчиков, ядра системы и системы загрузки «safe mode»

0-0x1000000 – область загрузчика

0x1000000 – 0x1A00000 – область хранения Safe системы (ядро со встроенной корневой ФС)

0x1A00000 – 0x2000000 – область хранения ядра системы

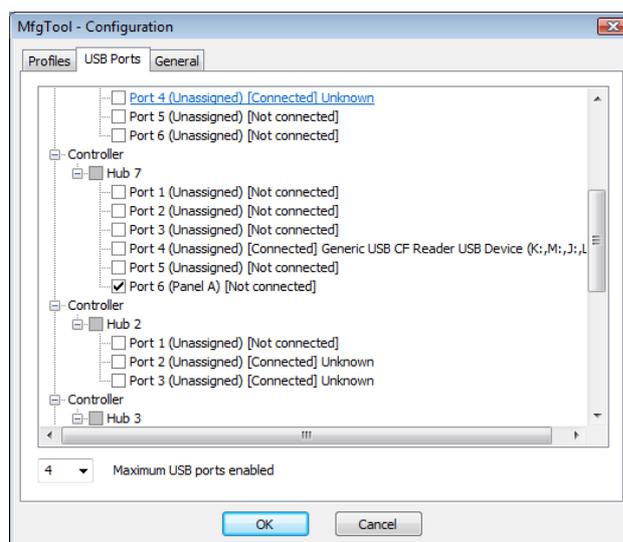
2) 224M – раздел UBI файловой системы, используется в качестве корневой файловой системы

5.2. USB

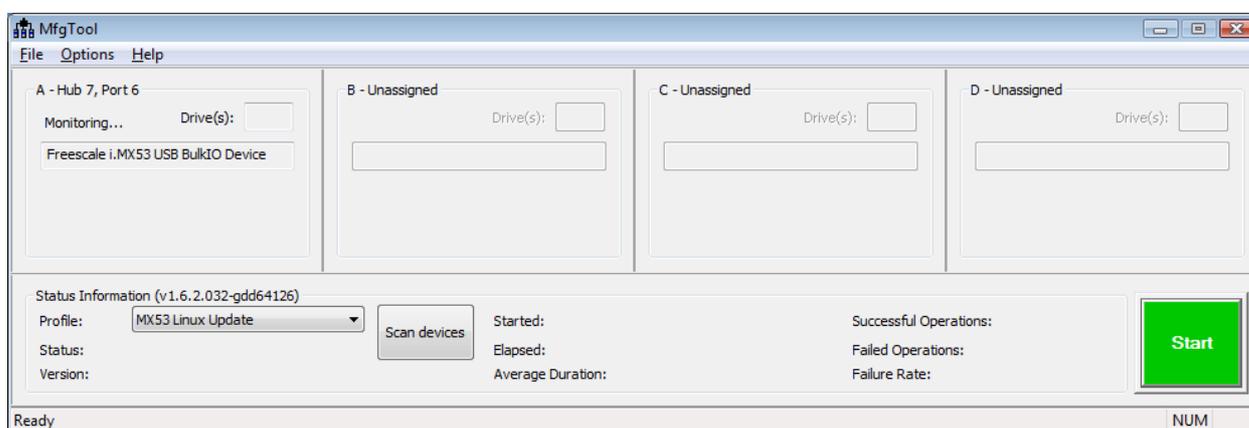
Загрузчик (нестираемый, располагаемый в самом процессоре), в случае отсутствия приложения на NAND flash (или разомкнут джампер «NAND CS»), переходит в режим загрузки по USB, для взаимодействия используется утилита MfgTool (из комплекта материалов к плате).

5.2.1 MFgTool

Распакуйте архив с программой, подключите плату USB кабелем к разъему X16 (J14 разомкнут!) с разомкнутым джампером «NAND CS» (J7) , включите питание. Далее необходимо установить драйвер из папки Drivers. Затем необходимо определить USB порт через который программа должна взаимодействовать с платой и явно его указать, для этого можно сравнить содержимое Configuration-USB Ports с подключенной и отключенной платой:



В результате манипуляций, при подключении платы, утилита MfgTool должна определять «FreeScale iMX53 USB BulkIO Device»:



Для загрузки достаточно будет выбрать профиль «MX53 Linux Update» и нажать кнопку «Start»

6. Корневая файловая система

Корневая файловая система (ФС), в поставляемом варианте платы, хранится в 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
- X11
- ...

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

7. Виртуальная машина VMware

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

Разархивируйте файл “ SK-iM53_linux_build_machine.exe”, установите VMware-player или VMware, откройте и проект виртуальной машины.

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

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

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

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

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

По умолчанию, в системе присутствует два пользователя:

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

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

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_safe – служит для конфигурирования ядра Safe системы
menuconfig_nand – служит для конфигурирования ядра системы штатной загрузки
build_safe_system – служит для сборки Safe системы
build_with_nand_rootfs – служит для сборки ядра штатной загрузки
- rootfs/nand_fs – содержит корневую систему штатной загрузки собираемую с помощью buildroot, скрипт **build_system** собирает корневую ФС и копирует ее архив в /home/user/tftp папку. Для конфигурирования содержимого необходимо выполнить «make menuconfig», для сборки достаточно выполнить make.
- rootfs/safe_fs – содержит корневую систему для safe загрузки, Для конфигурирования содержимого необходимо выполнить «make menuconfig», для сборки достаточно выполнить make.
- u-boot – содержит загрузчик системы, в корневой директории лежат скрипты:
build.sh – собирает u-boot для загрузки системы с NAND flash и копирует бинарный образ в /home/user/tftp папку
build_mfg.sh – собирает u-boot для загрузки системы через USB и копирует бинарный образ в /home/user/tftp папку
- utils – содержит дополнительные утилиты и скрипты

Например, необходимо обновить ядро Linux, для этого:

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

8. Общий принцип работы системы

После подачи питания (перезагрузки), процессор запускает первичный загрузчик (находится во внутренней не перепрограммируемой ROM) и по определенному алгоритму определяет наличие исполняемого кода во внешних носителях. Если приложение не найдено, процессор остается в режиме, который подразумевает взаимодействие с ним утилиты MfgTool.

Поскольку внешняя DDR2 (или любая другая память не инициализирована), первое запускаемое приложение должно быть загрузчиком. Это приложение (загрузчик u-boot) в первую очередь должен проинициализировать внешнюю память (например, правильно настроить параметры DDR2), скопировать исполняемое приложение из внешней Flash памяти во внешнюю DDR2 память и передать ему управление.

Загрузчик u-boot обладает обширными возможностями, например, он умеет копировать файлы по TFTP, SD или SATA, поддерживает целый набор команд и режимов.

В переменных окружения u-boot есть команда запуска, в которой указано, по какому адресу NAND flash следует прочитать образ ядра, куда этот образ памяти записать и по какому адресу запустить. Следующие сообщения консоли иллюстрируют этот процесс:

```
NAND read: device 0 offset 0x1000000, size 0xa00000
10485760 bytes read: OK
## Booting kernel from Legacy Image at 70800000 ...
```

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 70800000 ...
Image Name:   linux-2.6
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    2455412 Bytes = 2.3 MB
Load Address: 70008000
Entry Point:  70008000
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.