Release Notes for the QNX Neutrino 6.4.1 BSP for Freescale i.MX51 EVK 1.0.0#

Board Level Documentation#

http://cache.freescale.com/files/32bit/doc/user_guide/evk_imx51_Hardware_UG.pdf

System requirements#

Target system

- QNX Neutrino RTOS 6.4.1

Host development system

- QNX Momentics 6.4.1
- Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
- RS-232 serial port or a USB-to-serial adapter, and a straight-through serial cable
- Ethernet link

System Layout#

The tables below depict the memory layout for the image.

Memory layout

Item	Address
OS image loaded at	0x90100000

The interrupt vector table can be found in the buildfile located at **`src/hardware/startup/boards/mx51evk/build`**

Getting Started#

Preparing an SD Card#

The i.MX51 EVK can boot from USB, UART, SPI-NOR, MMC-1 (on bottom side of board), or MMC-2 (on top side of board). For example, to boot from the bottom SD slot, set dipswitch SW1 like to 0-0-0-0-0-0-1-1-0-0, see i.MX51 EVK board documentation for more boot switch details.

An SD card can be configured to by dd'ing the redboot image to the beginning of the SD card:

dd if=mx51_babbage_redboot.bin of=/dev/sdX

'dmesg' can be used on a Linux host to determine the SD device - e.g. /dev/sdd. The redboot image can be downloaded at the same Foundry27 location where the QNX BSP was downloaded.

Starting Neutrino#

Step 1: Build the BSP
You can build a BSP OS image from the source code. For instructions about building a BSP OS image, please refer to the chapter Working with a BSP in the Building Embedded Systems manual.

Copy or transfer the IFS image into your tftp server's directory.

- When compiling using the command line, the ifs image is in the **`images`** directory.
- When compiling using the IDE, the IFS image is by default at **`/Workspace_root_dir/bsp-freescale-mx51evk-src/images`**.

Step 2: Connect your hardware

1. Set up the i.mx51 EVK board in boot mode. Refer to the manual for the correct default jumper settings to use the CPU board.
2. Connect one end of the serial cable to the P5A1 (UART) serial port on the CPU board.
3. Connect the other end of the serial cable to the first available serial port of your host machine (e.g. ser1 on a Neutrino host).
4. Connect one end of RJ-45 Ethernet cable to the FEC Ethernet RJ45 Connector (P3 on the CPU board).
5. Connect the other end of the Ethernet cable to the Ethernet network where a TFTP server (which you'll use to transfer the boot image) exists.

On your host machine, start your favorite terminal program with these settings:

- Baud: 115200
- Bits: 8
- Stop bits: 1
- Parity: none

Then, apply power to the target. You should see output similar to the following:

```
++Booting from SDHC0
Bus Width:    1
Card initialization successful!
Actual capacity of the card is 3866624KB
Redboot uses 2097152KB
... Read from 0x1fee0000-0x1ff00000 at 0x00040000: .
... Read from 0x1fed3000-0x1fed4000 at 0x0005f000: .
PMIC ID: 0x000041d0 [Rev: 2.0]
Initializing SPI-NOR flash...
FEC LAN8700 PHY: ID=7c0c4
FEC: [ HALF_DUPLEX ] [ disconnected ] [ 10M bps ]:
Ethernet mxc_fec: MAC address 00:04:9f:00:ea:c2
IP: 172.18.74.117/255.255.255.0, Gateway: 172.18.74.1
Default server: 172.18.80.127

Reset reason: Power-on reset
fis/fconfig from MMC
Boot switch: INTERNAL
       EXPANSION: SD/MMC-0

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version FSL 200938 - built 11:47:45, Sep 15 2009

Platform: MX51 Babbage (Freescale i.MX51 based) PASS 3.0 [x32 DDR]. Board Rev 2.
5
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
Copyright (C) 2003, 2004, 2005, 2006 eCosCentric Limited

RAM: 0x00000000-0x1ff00000, [0x000953e0-0x1fed1000] available
FLASH: 0x00000000 - 0x80000000, 16384 blocks of 0x00020000 bytes each.
RedBoot>
```

Step 3: Setup the environment

At the RedBoot prompt, issue the **`fconfig`** command to change the current environment.

The current configurations will be displayed; change the configuration if you want.

```
Run script at boot: false
Use BOOTP for network configuration: false
Gateway IP address: 192.168.1.1
Local IP address: 192.168.1.202
Local IP address mask: 255.255.255.0
Default server IP address: 192.168.1.15
Board specifics: 0
Console baud rate: 115200
Set eth0 network hardware address [MAC]: false
Set FEC network hardware address [MAC]: false
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Default network device: mxc_fec
```

Step 4: Boot the IFS image

Once the above setup is complete, reset the board and you can run the load command at the RedBoot prompt to download the image: **`load -r -b 0x100000 ifs-mx51evk.raw; go 0x100000`**

Replace **`192.168.1.15`** with the IP address of your TFTP server and **`ifs-mx51evk.raw`** with the path of the image on the TFTP server.

RedBoot will display the follow message and start downloading the boot image:

```
Using default protocol (TFTP)
```

If the image is successfully loaded RedBoot will display:

```
Raw file loaded 0x00100000-0x00389fb7, assumed entry at 0x00100000
```

You also should see QNX Neutrino boot, followed by the welcome message on your terminal screen:

```
CPU0: L1 Icache: 512x64
CPU0: L1 Dcache: 512x64 WB
CPU0: L2 Dcache: 4096x64 WB
CPU0: VFP 410330c2
CPU0: 412fc081: Cortex A8 rev 1 800MHz

System page at phys:90011000 user:fc404000 kern:fc404000
Starting next program at vfe043df0
cpu_startnext: cpu0 -> fe043df0
VFPv3: fpsid=410330c2
coproc_attach(10): replacing fe0659e0 with fe0651a0
coproc_attach(11): replacing fe0659e0 with fe0651a0
Welcome to QNX Neutrino 6.4.1 on the i.MX51 EVK (ARM Cortex-A8 core) Board
Starting on-board ethernet with TCP/IP stack...
Starting I2C1 driver (/dev/i2c0)...
Starting I2C2 driver (/dev/i2c1)...
Starting SD1 memory card driver...
Starting SD2 memory card driver...
Starting watchdog
Starting SPI driver for CSPI (dev/spi0)...
Starting Enhanced SPI driver for ECSPI1(dev/spi1) ...
Starting Enhanced SPI driver for ECSPI2(dev/spi2) ...
Starting SSI Audio driver for SGTL5000...
```

```
"Mic In",0 - Capture Group
      Capabilities -  Volume Jointly-Capture Exclusive-Capture
      Channels - Front-Left Front-Right
      Volume Range - minimum=0, maximum=3
      Channel 0 Front-Left   -   0 (  0%)  Capture
      Channel 1 Front-Right  -   0 (  0%)  Capture
Starting USB Host driver...
Starting Graphics driver...
# Path=0 - FREESCALE MX35 e
 target=0 lun=0     Direct-Access(0) - SD:2 SA04G Rev: 0.4
Path=0 - FREESCALE MX35 e
 target=0 lun=0     Direct-Access(0) - SD:2 SD02G Rev: 4.1
#
```

You can test the OS simply by executing any shell builtin command or any command residing within the OS image (e.g. **`ls`**).

Auto Display Detection#

The i.MX51 EVM supports multiple output displays. This BSP's default build file produces an IFS that will auto detect which display is connected and configure video output for that display. The BSP checks for attached displays in the following order:

WVGA LCD Display Panel
DVI-D Monitor
VGA Monitor (default)

VGA is the default and is selected if neither the LCD nor the DVI-D monitor are detected. Note that if both the DVI-D and LCD panel are attached, the display will be configured for the LCD. Also, to detect a DVI-D monitor it must be connected to the DVI connector on the i.MX51 EVM and the monitor must be turned on.

Driver Command Summary#

The following table summarizes the commands to launch the various drivers.

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	`startup-mx51evk -d1 -d2`	.	.	`src/hardware/startup/boards/mx51evk`
Serial	`devc-sermx1 -e -F -c66500000 0x73FBC000,31` `devc-sermx1 -e -F`	devc-sermx1	.	`src/hardware/devc/sermx1`
FEC Network	`io-pkt-v4 -d mx51 mac=xxxxxxxxxxxx`	io-pkt-v4 ifconfig	devnp-mx51.so libsocket.so	`src/hardware/devnp/mx51`
I2C	`i2c-mx35 --u0,0` `i2c-mx35 --u1,1`	i2c-mx35	.	`src/hardware/i2c/mx35`
CSPI	`spi-master -d mx35 base=0x83FC0000,irq=38,waitstate=2`	spi-master	spi-mx35.so	`src/hardware/spi/mx35`

	spi-master -d mx35 base=0x83FC0000,irq=38,waitstate=2,loopback=1			
Enhanced SPI	spi-master -u1 -d mx51ecspi base=0x70010000,irq=36,waitstate=2 spi-master -u2 -d mx51ecspi base=0x83FAC000,irq=37,waitstate=2 spi-master -u1 -d mx51ecspi base=0x70010000,irq=36,waitstate=2,loopback=1 spi-master -u2 -d mx51ecspi base=0x83FAC000,irq=37,waitstate=2,loopback=1	spi-master	spi-mx51ecspi.so	src/hardware/spi/mx51ecspi
USB Host	io-usb -d ehci-mx31 ioport=0x73F80300,irq=14	io-usb usb* class drivers	devu-ehci-mx31.so libusbdi.so	*prebuilt only*
Audio	io-audio -d mx-mx51evk ssibase=0x83FCC000,tevt=29,tchd=1,rlvt=28,rchn=2,rate=48000,mixe	io-audio wave wave,rec mix_ctl	deva-ctrl-mx-mx51evk.so deva-util-restore.so libdma-sdma-imx51.so libasound.so.2	src/hardware/deva/ctrl/mx
SD card	devb-mmcsd-mx51 mmcsd ioport=0x70004000,irq=1 devb-mmcsd-mx51 mmcsd ioport=0x70008000,irq=2	devb-mmcsd-mx51	libcam.so fs-dos.so cam-disk.so	src/hardware/devb/mmcsd
WDT	wdtkick	wdtkick	.	src/hardware/support/wdtkick
Graphics	PATH=/proc/boot:/sbin:/bin:/usr/sbin:/usr/bin display-detect io-display -dvid=0,did=0,deviceindex=#	io-display display-detect imx51.conf imx51-1.conf imx51-2.conf	devg-imx51.so libgf.so.1 libGLES_CM.so.1 libffb.so.2 libm.so.2	src/hardware/devg/imx51

Some of the drivers are commented out in the default buildfile. To use the drivers in the target hardware, you'll need to uncomment them in your buildfile, rebuild the image, and load the image into the board.

Startup, Watchdog, Configurable SPI, Enhanced Configurable SPI, Ethernet, and Graphics have additional details:

Startup#

`startup-mx51evk [`*`startup-options`*`]`

Some modules aren't enabled after boot up, so you need to use command line options to startup to enable them. These options must be passed before any other startup options.

To enable	Use this option	Comment
WDT	-W	Enable WatchDogTimer
DVI mode	-d1	Support using the DVI mode to display
VGA mode	-d2	Support using the VGA mode to display

Note that both -d1 and -d2 must be specified to support auto display detection using the `display-detect` program.

Ethernet#

```
io-pkt-v4-hc -d mx51 mac=xxxxxxxxxxxx
```

MAC address labelled on i.mx51 evk board must be provided with "mac" command line option.

Audio#

```
io-audio -d mx-mx51evk
ssibase=0x83FCC000,tevt=29,tchn=1,revt=28,rchn=2,rate=48000,mixer=i2cdev=1:adr0

mix_ctl group "Mic In" capture=on
```

Note:
1.To start the audio system, you'll need to execute the I2C driver "i2c-mx35 --u1,1" first. 2.This audio driver sets "Line in " as the capture default device. i.MX51 EVK uses "Mic in " capture device.Please run "mix_ctl group "Mic In" capture=on" to set the capture device after the audio driver running.

Graphics#

```
PATH=/proc/boot:/sbin:/bin:/usr/sbin:/usr/bin display-detect io-display -
dvid=0,did=0,deviceindex=#
```

The `display-detect` program uses the i.MX51 EVM display hardware to detect the attached display and then replaces the '#' in the io-display command line with the selected displayindex using the following mapping:

Display	Index
WVGA LCD	1
DVI-D	0
VGA	2

The `display.conf` config file then maps these indexes 0-2 to driver config files imx51.conf, imx51-1.conf, and imx51-2.conf respectively to get the appropriate output display timing and configuration settings.

Accelerated Graphics#

An archive containing an experimental driver for the GPU (OpenVG & OpenGLES) can be found on the Foundry 27 graphics project:

http://community.qnx.com/sf/go/projects.graphics/frs.imx51_gpu_driver

Please obtain the latest version and follow the corresponding setup instructions.

Known Issues for This BSP#

- The audio driver `deva-ctrl-mx-mx51evk.so` doesn't support Syncronous data playback ability since the DMA library doesn't have position information. (Ref# 75488)
- The i.MX35 (ARM1136) processor doesn't support unaligned accesses in hardware. If an application (e.g. pwmopts) tries to access data that isn't aligned on a 32-bit boundary, a bus error will occur. To avoid this memory fault, you can enable the software emulation of unaligned accesses by starting procnto with the -ae option. (Ref# 71252)