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

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/mx51pdk/build`**

Getting Started#

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-mx51pdk/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 1985024KB
Redboot uses 1985024KB
... Read from 0x1fee0000-0x1ff00000 at 0x00040000: .
... Read from 0x1fed3000-0x1fed4000 at 0x0005f000: .
PMIC ID: 0x000045d0 [Rev: 2.0a]
Initializing SPI-NOR flash...
FEC LAN8700 PHY: ID=7c0c4
FEC: [ HALF_DUPLEX ] [ disconnected ] [ 10M bps ]:
Ethernet mxc_fec: MAC address 00:04:9f:00:ff:f8
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 200933 - built 07:49:01, Aug 17 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 - 0x79280000, 15508 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 -v -r ifs-mx51pdk.raw -m tftp -b 0x90100000`**

Replace **`192.168.1.15`** with the IP address of your TFTP server and **`ifs-mx51pdk.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 0x90100000-0x9030db4f, assumed entry at 0x90100000
```

Type **`go 0x90100000`** to jump to startup and boot the IFS image. You 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: 412fc085: Cortex A8 rev 5 800MHz

System page at phys:90011000 user:fc404000 kern:fc404000
Starting next program at vfe040e74
cpu_startnext: cpu0 -> fe040e74
VFPv3: fpsid=410330c2
coproc_attach(10): replacing fe062c84 with fe062444
coproc_attach(11): replacing fe062c84 with fe062444
Welcome to QNX Neutrino trunk on the i.MX51PDK (ARM Cortex-A8 core) Board
Starting I2C1 driver...
Starting I2C2 driver...
Starting watchdog
Starting USB Host driver...
starting Input Drivers...
#
```

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

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-mx51pdk`	.	.	`src/hardware/startup/boards/mx51pdk`

Serial	`devc-sermx1 -e -F -c66500000 0x73FBC000,31`	devc-sermx1	.	`src/hardware/devc/sermx1`
I2C	`i2c-mx35 -p0x83FC8000 -i62 --u0` `i2c-mx35 -p0x83FC4000 -i63 --u1`	i2c-mx35	.	`src/hardware/i2c/mx35`
FEC Network	`io-pkt-v4-hc -d mx51 mac=xxxxxxxxxxxx` `syspage`	io-pkt-v4 ifconfig	devnp-mx51.so libsocket.so	`src/hardware/devnp/mx51`
USB Host	`io-usb -d ehci-mx31 ioport=0x73F80300,irq=14`	io-usb usb* class drivers	devu-ehci-mx31.so libusbdi.so	*prebuilt only*

SPI	`spi-master -d mx51cspi base=0x83FC0000,irq=38,waitstate=2,loopback=1`
	`spi-master -d mx51ecspi base=0x70010000,irq=36,waitstate=2`
	`spi-master -d mx51ecspi base=0x70010000,irq=36,waitstate=2,loopback=1`
		spi-master	spi-mx51cspi.so
			spi-mx51ecspi.so
				`src/hardware/spi/mx51cspi`

WDT	`wdtkick-mx51`	wdtkick-mx51	.	`src/hardware/support/wdtkick-mx51`

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 and Ethernet have additional details:

Startup#

`startup-mx51pdk [`*`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	

CSPI and Enhanced CSPI#

```
spi-master -d mx51cspi base=0x83FC0000,irq=38,waitstate=2,loopback=1
```

```
spi-master -d mx51ecspi base=0x70010000,irq=36,waitstate=2,loopback=1
```

Note: There is no peripheral available on CSPI interface and Second Enhanced CSPI Interface on the i.mx51 EVK board.

Ethernet#

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

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

Known Issues for This BSP#

- SD card works only on SD1 Port. SD card insertion on SD2 Port hangs the board.