Release Notes for the QNX Neutrino 6.4.0 BSP for Freescale i.MX31 PDK 1.0.0#

System requirements#

Target system

- QNX Neutrino RTOS 6.4.0
- Board version: i.MX31 PDK
- ARM1136 processor
- 128 MB DDR SDRAM
- 256 MB NAND flash

Host development system

- QNX Momentics 6.4.0
- Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
- RS-232 serial port and serial cable, or a USB-to-serial cable
- Ethernet link

System Layout#

The tables below depict the memory layout for the image and for the flash.

Memory layout

Item	Address
OS image loaded at:	0x00100000
Ethernet base address	0xB6000000

The interrupt vector table can be found in the buildfile located at `src/hardware/startup/boards/mx31pdk/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 is ifs image is in the `images` directory.
- When compiling using the IDE the IFS image is by default at `/Workspace_root_dir/bsp-freescale-mx31pdk/Images`.

Step 2: Connect your hardware

1. Set up the board to three-board mode. Refer to the manual for the correct default jumper settings to use.
2. Connect one end of the serial cable to the CON4 serial port on the debug 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 an RJ-45 Ethernet cable to J1 10/100 Base T Ethernet RJ45 Connector.
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:

```
... Read from 0x07ee0000-0x07f00000 at 0xed100000: .
... Read from 0x07ed3000-0x07ed4000 at 0xed11f000: .
PMIC ID: 0x0000009b [Rev: 3.3]

Clock input is 26 MHz
LAN92xx Driver version 1.0
SMSC LAN9217: ID = 0x117a0000
Ethernet eth0: MAC address 00:04:9f:00:89:cb
IP: 10.42.103.136/255.255.255.0, Gateway: 10.42.96.1
Default server: 10.42.101.3

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version FSL 200740 - built 10:58:22, Oct 29 2007

Platform: Freescale (i.MX) MX31 3DS (Freescale i.MX31 based) PASS 2.0 [x32 DDR]
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.

RAM: 0x00000000-0x07f00000, [0x00013ee0-0x07ed1000] available
FLASH: 0xe0000000 - 0xed120000, 1673 blocks of 0x00020000 bytes each.
== Executing boot script in 1.000 seconds - enter ^C to abort
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: 10.42.121.2
Local IP address: 10.42.121.185
Local IP address mask: 255.255.255.0
Default server IP address: 10.42.121.2
Board specifics: 0
Console baud rate: 115200
Set eth0 network hardware address [MAC]: false
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
```

Step 4: Boot the IFS image

Once the above setup is complete, you can run the load command at the RedBoot prompt to download the image: **`load -r -b 0x00100000 –h <TFTP server IP> /<TFTP server directory>/ mx31pdk.raw`**

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 0x00100000-0x002300d3, assumed entry at 0x00100000
```

Type **`run`** to jump to startup and boot the IFS image. You should see the QNX Neutrino welcome message on your terminal screen:

```
Welcome to Neutrino on the i.MX31PDK (ARM 1136 core) Board
```

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-mx31pdk`	.	.	`src/hardware/startup/boards/mx31pdk`
Serial	`devc-sermx1 -e -F -c66500000 0x43F90000,45`	devc-sermx1	.	`src/hardware/devc/sermx1`
SPI	`spi-master -d mx31 base=0x50010000,irq=13,waitstate=2`	spi-master	spi-mx31.so	`src/hardware/spi/mx31`
SDMA	`mx31_dma_cfg -c`	mx31_dma_cfg}	libdma-imx31v2.so	`src/utils/m/mx31_dma_cfg`
Network	`io-pkt-v4 -dsmc9118 ioport=0xb6000000,irq=28,verbose=0 -ptcpip`	io-pkt-v4 ifconfig nicinit* ping*	devn-smc9118.so libsocket.so devn-shim.so	`src/hardware/devn/smc9118`
USB	`io-usb -d ehci-mx31 ioport=0x43f88500,irq=36` (host 2)	io-ubs usb*	devu-ehci-mx31.so libusbdi.so class drivers	*prebuilt only*
I2C	`i2c-imx31ads`	i2c-imx31ads	.	`src/hardware/i2c/imx31ads`
Audio	`mx31_dma_cfg -c` `io-audio -d imx31pdk`	io-audio mx31_dma_cfg	deva-ctrl-imx31pdk.so libasound.so libdma-imx31v2.so	`src/hardware/deva/ctrl/imx31pdk`
Graphics	`mx31pdk_lcd_init` `io-display -dvid=0,did=0`	io-display mx31pdk_lcd_init	devg-imx31.so libgf.so.1 libGLES_CM.so.1	*prebuilt only*

			libffb.so.2 libm.so.2	

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.

USB, Graphics and Audio have addtional details:

USB#

USB can be run on the USB HOST ULPI Interface (HS).

- Start io-usb as **`io-usb -d ehci-mx31 ioport=0x43f88500,irq=36`**

The USB host port is `j4`.

You will want to include class drivers, such as **`devb-umass`** in order to make use of attached USB devices.

Graphics:#

The startup must be started with the following options:

```
startup-mx31pdk -v -r0x86000000,0x02000000,1
```

To start the driver. `mx31pdk_lcd_init` `io-display -dvid=0,did=0`

Note:

- To start graphic driver, first launch spi driver for CSPI 1 on /dev/spi0, then launch LCD init application `mx31pdk_lcd_init`, then start io-display.
- Starting io-display simply starts the display server. Nothing will appear on the display until a graphical application is run (i.e. vsync or egl-gears).

Required Config Files:

- /etc/system/config/display.conf
- /etc/system/config/imx31.conf

The `/etc/system/config/display.conf` configuration file describes the graphics memory interface settings. It should contain the following for a 480x640 resolution:

```
device {
        drivername=imx31
        vid=0
        did=0
        modeopts=/etc/system/config/imx31.conf
        display {
                xres=480
                yres=640
                refresh=60
                pixel_format=rgb565
        }
    }
}
```

Audio #

> **Note**:
> Only playback is supported

The sdma configuration utility, **`mx31_dma_cfg -c`**, should be launched before starting **`io-audio`**.

The stereo output jack is located at `J19` on the Personality Board.

Known Issues for This BSP#

- The resolution of the playback/capture positional information returned by the audio driver **deva-ctrl-imx31pdk.so** to the client is limited to the fragment size since we are unable to get the transfer count of the current DMA operation from the SDMA microcode. (Hardware limitation, no Ref# available)
- Running the Open GLES conformance tests will cause the devg-imx31.so to lockup. This behaviour has not appeared with other 3D applications and will be investigated for a future release. (Ref # 56923)
- The serial driver: devc-sermx1 doesn't support hardware flow control, and will be investigated for a future release . (Ref# 57988)