Release Notes for the QNX Neutrino 6.4.0 BSP for Freescale i.MX25 3DS PDK 1.0.0#

System requirements#

Target system

- QNX Neutrino RTOS 6.4.0
- Board version: i.MX25 PDK
- ARM926 processor
- 64 MB DDR SDRAM
- 2 GB NAND flash

Host development system

- QNX Momentics 6.4.0
- IDE 4.6.0
- Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
- RS-232 serial port and a straight-through 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

The interrupt vector table can be found in the buildfile located at **`src/hardware/startup/boards/3dsmx25/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-3dsmx35/Images`**.

Step 2: Connect your hardware

1. Set up the board to 3-Stack mode. Refer to the manual for the correct default jumper settings to use for the Personality and Debug boards.

2. Connect one end of the serial cable to the CON4 (UART-DCE) 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 one end of RJ-45 Ethernet cable to the FEC Ethernet RJ45 Connector (J16 on the Personality 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:

```
++NAND: RCSR=64000150
Searching for BBT table in the flash ...
.
Found version 1 Bbt0 at block 8191 (0x7ffc0000)
Total bad blocks: 0
... Read from 0x03ec0000-0x03f00000 at 0x000c0000: ..
... Read from 0x03eb3000-0x03eb4000 at 0x000ff000: .
Turning on PMIC regulators: 1,2,3,4,5

LAN92xx Driver version 1.1
SMSC LAN9217: ID = 0x117a REV = 0x0
[Warning] FEC not connect right PHY: ID=5c8000
FEC: [ HALF_DUPLEX ] [ disconnected ] [ 10M bps ]:
Ethernet eth0: MAC address 00:04:9f:00:a5:16
IP: 10.42.104.42/255.255.240.0, Gateway: 10.42.96.1
Default server: 10.42.97.136
hardware reset by POR

Clock input is 24 MHz
Booting from [NAND flash]
[0x80000000 bytes]: 8192 blocks of 128 pages of 2048 bytes each.


RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version FSL 200910 - built 08:59:52, Mar  2 2009

Platform: MX25 3-Stack (Freescale i.MX25 based) PASS 1.1 [x32 DDR]
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
Copyright (C) 2003, 2004, 2005, 2006 eCosCentric Limited

RAM: 0x00000000-0x03f00000, [0x000953e8-0x03eb1000] available
FLASH: 0x00000000 - 0x80000000, 8192 blocks of 0x00040000 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
```

Note: If you set the default network device as lan92xx_eth0 by **`ifconfig`**, then you should connect the Ethenet cable to J1 Ethernet Connector on the debug board to download the OS image. Once the OS image is running, you have to switch the Ethernet cable to J16 FEC Ethernet Connector on the personality board for the network connection, because the QNX Ethernet driver only support the FEC interface on the personality board.

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 0x00100000 -h 192.168.1.15 /tftpboot/ifs-3dsmx25.bin`**

Replace **`192.168.1.15`** with the IP address of your TFTP server and **`/tftpboot/ifs-3dsmx25.bin`** 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 0x00100000-0x002aae8b, assumed entry at 0x00100000
```

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

```
load entry_address=0x100000
virt_addr=0x100000
phys_addr=0x80100000
CPU0: Dcache: 512x32 WB
CPU0: Icache: 512x32
CPU0: 41069264: arm926 rev 4 396MHz

System page at phys:80010000 user:fc404000 kern:fc404000
Starting next program at vfe03c830
cpu_startnext: cpu0 -> fe03c830
Welcome to QNX Neutrino 6.4.0 on the Freescale MX25 3DS Board (ARM 926 core)
#
```

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

Step 5: Replace the ~Redboot bootloader with a native QNX IPL and OS image in flash

At some point, you may wish to replace the ~Redboot bootloader with the native QNX IPL code. This may be desirable once you've tweaked the OS image exactly the way you want it, and you want the board to boot the image automatically, immediately on power up.

To replace the ~Redboot bootloader:

1. Modify your buildfile to generate a binary image.
2. Run the **`mkflashimage.sh`** script, inside the **`images`** directory of the BSP. The output file from this script is a binary ROM image of the IPL called **`image.rom`**.

The **`mkflashimage.sh`** script will convert the IPL image to binary and prefix it with a 16k pad.

Here is the **`mkflashimage.sh`** script:

```
#!/bin/sh
# Script to build a binary IPL for the i.MX25 3DS Board that can be programmed
# into NAND flash at the beginning of block 0
#
# Example usage: ./mkflashimage.sh arm/le/ipl-3dsmx25

if [ "$1" == "" ]; then
   filename="../install/armle/boot/sys/ipl-3dsmx25"
else
   filename=$1
fi

echo "Converting $filename"

# Convert the IPL into binary format
${QNX_HOST}/usr/bin/ntoarm-objcopy --input-format=elf32-littlearm --output-format=binary $filename ./tmp-ipl.bin

# Pad the result to 16K (this will ensure the assumptions within the IPL are correct)
mkrec -s16k -ffull -r tmp-ipl.bin > image.rom

# Clean up after ourselves
rm -f tmp-*

echo "You can now program image.rom to NAND block 0"
```

3. Burn **`image.rom`** to the board's NAND using the Advanced Toolkit. The Advanced Toolkit can be found on Freescale's i.MX25 Product Development Kit download page

Set the Personality Board switches to the following settings:

Switch	1	2	3	4	5	6	7	8
SW21	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
SW22	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Set the Debug Board to the following settings:

SW5	SW6	SW7	SW8	SW9	SW10
0	0	0	0	1	1

Switch	1	2	3	4	5	6	7	8
SW4	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON

Install the tool on a Windows host and connect it to the i.MX25 via the USB OTG connector on the Personality Board. Power up the board and install the Windows drivers as specified in the Advanced Toolkit installation notes.

Launch Advanced Toolkit and select the following options:

i.MX CPU: i.MX25_TO1.1
Device memory: DDR2
Communication Channel: USB

Click Next, then click *Flash Tool*. You can now program the **`image.rom`** file to NAND block 0. You should also program it to NAND block 1 when done development as it will be used by the boot ROM as the backup boot block.

To boot from NAND, first power off the board, disconnect the USB cable from the OTG port and set the Personality Board switches to the following settings:

Switch	1	2	3	4	5	6	7	8
SW21	ON	OFF	OFF	ON	ON	OFF	OFF	OFF
SW22	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF

Set the Debug Board to the following settings:

SW5	SW6	SW7	SW8	SW9	SW10
0	1	1	0	0	0

Switch	1	2	3	4	5	6	7	8
SW4	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON

Powering on will now boot the IPL.

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-3dsmx25`	.	.	`src/hardware/startup/boards/3dsmx25`
Serial	`devc-sermx1 -e -F -b115200 -c66500000 0x43f90000,45`	devc-sermx1	.	`src/hardware/devc/sermx1`
I2C	`i2c-mx21 -p0x43f80000, -i3`	i2c-mx21	.	`src/hardware/i2c/mx21`
SPI	`spi-master -d mx35`	spi-master	spi-mx35.so	`src/hardware/spi/mx35`

FEC Network	`io-pkt-v4 -d mcimx25 ioport=0x50038000,irq=57,mac=0000456789ab,verbose`	io-pkt-v4 ifconfig mcinfo ping	devn-mcimx25.so libsocket.so devnp-shim.so	`src/hardware/devn/mcimx25`
USB Host	`io-usb -d ehci-mx31 ioport=0x53ff4500,irq=35,verbose=4` `io-usb -d ehci-mx31 ioport=0x53ff4100,irq=37,verbose=4`	io-usb usb* class-drivers	devu-ehci-mx31.so libusbdi.so	*prebuilt only*

USB Device	`devb-ram mkdosfs /dev/hdX`
	`io-usb-dcd -d usbumass-mx25ads-ci ioport=0x53ff4000,irq=37`
	`devu-umass_client-block -l lun=0,fname=/dev/hdX`
	`ulink_ctrl -l1`
		io-usb-dcd
		devu-umass_client-block
		ulink_ctrl
			devu-usbumass-mx25ads-ci.so
			libusbdci.so
				prebuilt only

NAND	`fs-etfs-3dsmx35_2048 -m /fs/etfs`	fs-etfs-3dsmx35_2048 etfsctl	.	`src/hardware/etfs/nand2048/3dsmx35_20`
Audio	`i2c-mx21 -p0x43f80000, -i3` `io-audio -d mx-3dsmx25 ssibase=0x50034000,tevt=29,tchn=3,revt=28,rchn=4`	io-audio i2c-mx21	deva-ctrl-mx-3dsmx25.so deva-util-restore.so libasound.so.2	`src/hardware/deva/ctrl/mx`
Graphics	`io-display -dvid=0,did=0`	io-display	devg-imx25.so devg-soft3d-fixed.so libGLES_CL.so.1 libgf.so.1 libffb.so.2 libdisputil.so	`src/hardware/devg/imx25`
TSC (Touch Screen Controller)	`devi-mxSenna -P touch -i46 -a0x50030000 -c0x53f80000 -e abs -S150`	devi-mxSenna		`src/hardware/devi/mxSenna`
CAN	`dev-can-mx35 can1`	dev-can-mx35 canctl*	.	`src/hardware/can/mx35`
RTC	`rtc hw`	rtc date*	.	`src/utils/r/rtc`
WDT	`wdtkick`	wdtkick	.	`src/hardware/suppot/wdtkick`

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, USB, and Graphics have additional details:

Startup#

```
startup-3dsmx25 [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	
USB Host	-U	Configure USB host pins	

CAN #

```
dev-can-mx35 can1
```

Note: there is only one CAN interface on the i.mx25 PDK/ADS board.

About Graphics#

The graphics driver provides support for the Liquid Crystal Display Controller (LCDC). The default LCD is the Chunghwa Picture Tubes CLAA057VA01CW TFT LCD that comes with the MX25 3DS reference board. Other LCDs may be supported through the driver configuration file - imx25.conf.

The LCDC has a requirement that a displayable surface not cross a 4MB memory boundary. To prevent this it is recommended that memory be reserved for graphics on a 4MB from start-up (using -r option) and then this memory area provided to the driver using the vbase and vsize options in imx25.conf. (see imx25.conf for details on these options)

Known Issues for This BSP#

- The speed and duplex options for io-pkt may not work in all cases. We suggest you do not use those options and allow io-pkt to auto negotiate the link settings.
  - Workaround: use auto-negotiation.
- The i.MX25 (ARM926) 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)
- When this BSP is built by IDE, there is a build warning "Unable to find devi-mxSenna in search paths". This IDE 4.6.0 known issue will be fixed in IDE 4.7.0.(Ref# 72902)
  - WORKAROUND: Remove the devi-mxSenna driver from the project.bld file, and re-add using browse and selecting the "Use file's absolute path" option.
- The left/right channels of headphone output of the audio driver are reversed for the old version of mx25 3ds board before March 31, 2009 due to a hardware bug.

- Hardware Issue: Some boards are missing R232 resistor(Ref#71995). If the R232 resister has not been populated on the board, you must add a 0 ohm resistor on R232 on the board. This fix bypasses the internal LDO on the SGTL5000. Without this resistor you may see may issues including:
  - the I2C1 bus hangs and an "Error in I2C transaction" error message is displayed in Redboot.
  - graphics, USB, and networking may not work, or will work intermittently.