Release Notes of the QNX 6.4.0 BSP for Freescale i.MX27 ADS Trunk#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: Freescale i.mx27ads reference board (REV2.8)
- Board version: i.MX27 ADS
- ARM926EJ-S processor
- 64 MB SDRAM
- 32 MB NOR flash (S71WS256NDO)
- 128 MB Saumsung NAND flash (K9K1G08U0B-JIB0)

Host development system#

- QNX Momentics 6.4.0
- Latest version of the QNX Momentics Character Driver Development Kit (DDK)
- Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
- RS-232 serial port
- NULL-modem 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	0xd4040300 (IRQ: 26)
Burst flash base address	0xC0000000

The interrupt vector table can be found in the buildfile located at `src/hardware/startup/boards/mx27ads/build`

Getting Started#

Step 1: Connect your hardware#

1. Set up the board. Refer to the manual for the correct default jumper settings to use.
2. Connect the serial cable to the first serial port of the i.MX27 ADS and the first serial port of your host machine (e.g. ser1 on a Neutrino host).

- If you have a Neutrino host with a serial mouse,you may have to move it to the second serial port on your host, because some terminal programs require the first serial port.

3. Connect an RJ-45 ethernet cable between the ethernet port on the i.MX27 ADS and your local network.

Step 2: Build the BSP#

You can build a BSP OS image from the source code or the binary components contained in a BSP package. For instructions about building a BSP OS image, please refer to the chapter Working with a BSP in the Building Embedded Systems manual.

Step 3: Transfer the OS image to the target using the ROM monitor#

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

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

1. Make sure the host machine is capable of handling TFTP requests (see the inetd documentation for details).
2. Verify the serial link between the host machine and the target.
3. Verify that an Ethernet link exists between the host machine and the target.
4. Apply power to the target board.

You should see output similar to the following:

```
RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version FSL 200749 - built 17:00:12, Dec 13 2007

Platform: MX27 ADS/EVB (Freescale i.MX27 based) PASS 2.1 [x32 SDR]
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.

RAM: 0x00000000-0x07f00000, [0x00025260-0x07ed1000] available
FLASH: 0xc0000000 - 0xc2000000, 256 blocks of 0x00020000 bytes each.
RedBoot>
```

Step 4: Setting up the environment#

1. At the RedBoot prompt, issue the following command to change the current environment:

`fconfig`

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: 10.42.96.1
Local IP address: 10.42.103.53
Local IP address mask: 255.255.240.0
Default server IP address: 10.42.98.230
Board specifics: 0
Console baud rate: 115200
Set eth0 network hardware address [MAC]: true
eth0 network hardware address [MAC]: 0x40:0x00:0x00:0x00:0xEA:0x18:0xF0
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: cs8900a_eth0
alias/bsp_tag: BSP_VER=rel3
Update RedBoot non-volatile configuration - continue (y/n)?
```

Typey to accept the new configuration; RedBoot will write the new configuration to the flash.

2. Once the above setup is complete, you can run the load command at the RedBoot prompt to download the image:

```
load -r -b 0x00100000 /images/ifs-mx27.raw
```

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

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

If the image is successfully loaded, you will see the following message from the RedBoot prompt:

```
Raw file loaded 0x00100000-0x002cbef7, assumed entry at 0x00100000
```

3. Type:

```
run 0x00100000
```

You should now see the QNX Neutrino welcome message on your terminal screen:

```
Welcome to Neutrino on the MX27ADS (ARM 926 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). Once the initial image is running, you can update the OS image using the network and flash drivers. For sample command lines, please see the "Summary of driver commands" section.

Creating a flash partition#

1. Enter the following command to start the NOR flash filesystem driver:

```
devf-mx31ads -s0xc0000000,32M
```

2. Erase the flash: Because the ROM monitor and the QNX boot image are in the flash, you may not want to erase them. Use the -l (length) and -o (offset) options to avoid these areas.
For example, if the OS image has a maximum size of 3 MB, and you want to create a 16 MB partition,

```
flashctl -p/dev/fs0 -o4M -l16M -ve
```

3. Format the partition:

```
flashctl -p/dev/fs0p0 -o4M -l16M -vf
```

4. Slay, then restart the driver:

```
slay devf-mx31ads
devf-mx31ads -s0xc0000000,32M
```

You should now have a /fs0p0 directory which you can copy files to.

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-mx21ads`	.	.	`src/hardware/startup/boards/mx21ads`
Serial	`devc-sermx1 -e -F -b115200 -c13300000 0x1000a000,20`	devc-sermx1	.	`src/hardware/devc/sermx1`
Flash (NOR)	`devf-mx31ads -s0xc0000000,32M`	devf-mx31ads flashctl	.	`src/hardware/flash/boards/mx31ads`
ETFS Flash (NAND)	`fs-etfs-imx31ads_512 -D addr=0xd8000000 -m /fs/etfs`	fs-etfs-imx31ads etfsctl	.	`src/hardware/etfs/nand512/imx31ads_512`
Ethernet	`io-pkt-v4 -dcrys8900 ioport=0xd4040300,irq=26,mac=00e03991234e -ptcpip`	io-pkt-v4 ifconfig nicinfo* ping* cat*	devn-crys8900.so libsocket.so devnp-shim.so	`src/hardware/devn/crys8900`
FEC	`io-pkt-v4 -dmcimx27 irq=50 -p tcpip`	io-pkt-v4 ifconfig nicinfo* ping* cat*	devn-mcimx27.so libsocket.so devnp-shim.so	`src/hardware/devn/mcimx27`
USB	`io-usb -dehci-mx31 ioport=0x10024300,irq=54,verbose=4` (host 1) `io-usb -dehci-mx31 ioport=0x10024500,irq=55,verbose=4` (host 2)	io-ubs usb*	devu-ehci-mx31.so libusbdi.so class drivers	*prebuilt only*
I2C	`i2c-imx31ads -p0x10012000, -i12`	i2c-imx31ads	.	`src/hardware/i2c/imx31ads`

Some of the following 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#

Command:

startup-mx21ads [startup-options]

Some modules are not enabled by default on power 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	
I2C	-c	.	
USB Host 1 FS	-U1	enable USB Host 1 FS	

USB Host 2 HS	-U2	enable USB Host 2 HS	

ETFS flash (NAND):#

Command to run the ETFS NAND flash driver (64 MB):
```
fs-etfs-imx31ads_512 -m /fs/etfs -D addr=0xd8000000
```

You need to use the -e option the first time you start NAND flash driver.
```
fs-etfs-imx31ads_512 -e -m /fs/etfs -D addr=0xd8000000
```

To erase and format the NAND flash partition:

```
etfsctl -d /dev/etfs2 -S -e
etfsctl -d /dev/etfs2 -S -f -c
```

You should slay and restart the driver, after which you should have the mountpoint /fs/etfs/, which you can use to copy files to.

USB host :#

USB can be run either on the USB HOST Serial Interface (FS/LS) or the USB HOST ULPI Interface (HS), which are referred to as Host 1 and Host 2 respectively.

To use Host 1 FS:

- Start **`startup-mx21ads`** with the **`-U1`** option.
- Start io-usb as **`io-usb -d ehci-mx31 ioport=0x10024300,irq=54,verbose=4`**

The USB host port is `J8`.

To use Host 2 HS:

- Start **`startup-mx21ads`** with the **`-U2`** option.
- Start io-usb as **`io-usb -d ehci-mx31 ioport=0x10024500,irq=55,verbose=4`**

The USB host port is `J7`.

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

Known Issues for this BSP#

- Because of a known issue with the crys8900 chipset, the Ethernet driver, devn-crys8900.so, can't ping with large packets on any platform. (Ref#58001)
- The serial driver: devc-sermx1 doesn't support hardware flow control, and will be investigated for a future release . (Ref# 57988)
- In those instances where the the ROM monitor's MAC address is different from the one you pass in when running **`io-net`** and/or **`io-pkt`** , the host can cache the ROM monitor's address. This can result in a loss of connectivity. **`Workaround`**: If you need to specify a MAC address to **`io-net`** and/or **`io-pkt`**, we recommend that you use the same MAC address that the ROM monitor uses. This will ensure that if the host caches the ROM monitor's MAC address, you'll still be able to communicate with the target. Otherwise you might need to delete the target's arp entry on your host.