Release Notes of the QNX 6.4.0 BSP for Freescale MPC8548 CDS Trunk#

System requirements#

Target system

- QNX Neutrino RTOS 6.4.0
- Board version: Freescale MPC8548 CDS
- 16 MB NOR flash
- ROM Monitor version U-Boot 1.1.3

Host development system

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

System Layout#

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

Item	Address
OS image loaded at:	0x00100000
OS image begins execution at:	0x00101e38
Flash base address	0xff000000
Monitor Flash offset	0xff800000
TSEC1 base address	0xe0024000 (IRQ: 13,14,18)
TSEC2 base address	0xe0025000 (IRQ: 19,20,24)
TSEC3 base address	0xe0026000 (IRQ: 15,16,17)
TSEC4 base address	0xe0027000 (IRQ: 21,22,23)
Serial base address	0xe0004600 (IRQ: 26)

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.

Step 2: Connect your hardware
1. Set up the board.

Some PCI cards may not operate at a PCI frequency of 66 MHz.If this is the case for your PCI cards,we recommend that you cor
For the Freescale CDS MPC8548 CPU card, the following configuration (i.e. PCI frequency of 33 MHz) has been validated.

SW1:[1-6],8 ON, SW1:7 OFF.
SW2:[2-6] ON, SW2:1,[7-8] OFF.
SW3:[1-2],[4-6] ON, SW3:3,[7-8] OFF.
SW4:1,[5-8] ON, SW4:[2-4] OFF.

Refer to the hardware manual for a complete list of jumper and dip switch settings to use and compile the ROM monitor for the a

Note: To change the PCI frequency for the Freescale MPC8540 ADS board to 66 MHz:

1. Change the jumper settings as described in the hardware manual.
2. Compile the ROM monitor to use a frequency of 66 MHz.
3. Add the -t66000000 option to the startup command line in your buildfile and rebuild your OS image.

---

2. Connect one end of the serial cable to the P35 serial port 1.

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).

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

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

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

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

```
U-Boot 1.1.3 (FSL Development) (Dec  4 2006 - 13:13:55)

CPU:   8548_E, Version: 2.1, (0x80390021)
Core:  E500, Version: 2.2, (0x80210022)
Clock Configuration:
      CPU: 999 MHz, CCB: 399 MHz,
      DDR: 199 MHz, LBC:  49 MHz
L1:    D-cache 32 kB enabled
      I-cache 32 kB enabled
Board: CDS Version 0x13
CPU Board Revision 0.0 (0x0000)
   PCI1: 64 bit, 33 MHz, async
   PCI2: disabled
I2C:   ready
DRAM:  Initializing
   SDRAM: 64 MB
   DDR: 256 MB
FLASH: 16 MB
L2 cache 512KB: enabled
In:    serial
Out:   serial
Err:   serial
Net:   eTSEC0: PHY is Marvell 88E1145 (1410cd4)
eTSEC1: PHY is Marvell 88E1145 (1410cd4)
eTSEC2: PHY is Marvell 88E1145 (1410cd4)
eTSEC3: PHY is Marvell 88E1145 (1410cd4)
eTSEC0, eTSEC1, eTSEC2, eTSEC3
The IP address of the board is currently set to 10.42.104.42
```

```
The MAC address is 00:04:9f:00:aa:ab
If they don't match your network environment, please change them in U-Boot and kernel manually.
Hit any key to stop autoboot: 0
=> setenv ipaddr 192.168.200.2
```

Step 3: Setup the environment

On your target, type the following, filling in the appropriate IP addresses and ifs file:

```
=> setenv ipaddr 192.168.200.2
=> setenv serverip 192.168.200.1
=> setenv bootfile ifs-8548cds.raw
=> setenv loadaddr 0x100000
=> setenv bootcmd 'tftpboot $loadaddr $bootfile; go $loadaddr'
=> setenv bootdelay 2
=> saveenv
Saving Environment to Flash...
Un-Protected 4 sectors
Erasing Flash...
.... done
Erased 4 sectors
Writing to Flash...
256 KBdone
Protected 4 sectors
```

Step 4: Boot the IFS image

You can use TFTP download (the default) or serial download to transfer the image from your host to the target:

Step 4A: TFTP download

This method requires a raw image, which the buildfile creates by default.

Once the above setup is complete, you can run the load command at the => prompt to download the image:

`=> boot`

At this point you should see the ROM monitor download the boot image, indicated by a series of number signs. You'll also see output similar to this when it completes downloading:

```
Speed: 100, full duplex
Using MOTO ENET0 device
TFTP from server 192.168.200.1; our IP address is 192.168.200.2
Filename 'ifs-8548cds.raw'
Load address: 0x100000
Loading: ###################################################################
   ###################################################################
   ###################################################################
   ###################################################################
   ###################################################################
   ###################################################################
   ###################################################################
   ###################################################################
   ###################################################################
   ###################################################################
   ###################################################################
   ###########################################################
done
```

```
done
Bytes transferred = 3361576 (334b28 hex)
## Starting application at 0x00100000 ...
```

Step 4B: Serial download

This method requires an SREC image. You have to modify the buildfile to create this format. Change this:

[virtual=ppcbe,raw]

to this:

[virtual=ppcbe,srec]

Rebuild the image. On your target, type:

```
=> setenv loads_echo 0
=> saveenv
=> loads
```

On your host, copy the image to the serial port that's connected to the board. For example, on a Neutrino host:

```
cp ifs-8548cds.raw /dev/ser1
```

On a Windows host, you can use Hyperterminal's transfer feature to copy the image as a text file.

Note: The serial line shouldn't already be in use.

At this point, you should see the ROM monitor download the boot image, indicated by a series of dots. You'll also see output similar to this when it finishes downloading:

```
## First Load Addr = 0x00100000
## Last  Load Addr = 0x0023955B
## Total Size      = 0x0013955C = 1283420 Bytes
## Start Addr      = 0x00101E38
=>
```

Type: **`go start_addr`**

Note: The start_addr is the startup entry point address. You can find this address from the mkifs utility (you'll need to use the __

For example:

```
#mkifs -v -r../install  85x0ads.8548cds.build ifs-8548cds.raw
 Offset       Size     Entry     Ramoff Target=Host
 100000       100         0     --- /usr/qnx640/target/qnx6/ppcbe/boot/sys/raw.boot
 100100       100       ----      --- Startup-header
 100200     11108    102bb4     --- /tmp/DAA003706
              ...
```

In this example, 102bb4 is the address to use.

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

```
System page at phys:0000b000 user:0000b000 kern:0000b000
Starting next program at v0014415c
Welcome to QNX Neutrino 6.4.0 on the PowerPC 8548CDS 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 flash filesystem driver:

```
devf-generic -s0xff000000,16M
```

2. To prepare the area for the partition.

Caution: Do not erase the last 8M -- it contains the ROM monitor. Use the -l (length) option to avoid these areas. To create a 1 MB partition, enter the following command:

flashctl -p/dev/fs0 -l1M -ve

3. Format the partition:

flashctl -p/dev/fs0p0 -l1M -vf

4. Slay, then restart the driver:

slay devf-generic &
devf-generic -s0xff000000,16M &

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-85x0ads`	.	.	`src/hardware/startup/boards/85x0ads`
Serial for 8548 CDS	`devc-ser8250 -e -c396000000 -b115200 0xe0004600,26`	devc-ser8250	.	`src/hardware/devc/ser8250`
Flash (NOR)	`devf-generic -s0xff000000,16M`	devf-generic flashctl	.	`src/hardware/flash/boards/generic`
PCI	`pci-mpc85xx -dmpc85xx x3`	pci-mpc85xx pci	.	`src/hardware/pci/mpc85xx`
Network	`io-pkt-v4-hc -dmpc85xx mac=xxxxxxxxxxxx,verbose -ptcpip`	io-pkt-v4-hc ifconfig	devnp-mpc85xx.so libsocket.so	"Binary form only:" `prebuilt/ppcbe/lib/dll/devnp-mpc85xx.so`

Network:MPC Security Engine (AKA SEC)	`io-pkt-v4-hc -dmpcsec -p tcpip-v6 ipsec -dmpc85xx mac=00112233AABB`	io-pkt-v4-hc ifconfig	devnp-mpc85xx.so devnp-mpcsec.so libsocket.so	"Binary form only:" `prebuilt/ppcbe/lib/dll/devnp-mpcsec.so`

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.

Network:#

without encryption:
`io-pkt-v4-hc -dmpc85xx mac=xxxxxxxxxxxx,verbose -ptcpip`

with encryption in software:
`io-pkt-v4-hc -p tcpip-v6 ipsec -dmpc85xx mac=00112233AABB`

with encryption in hardware:
`io-pkt-v4-hc -dmpcsec.so -p tcpip-v6 ipsec -dmpc85xx.so mac=00112233AABB`

Note:
The latest sources for `devnp-mpc85xx.so` and `devnp-mpcsec.so` are available from the networking project.

Known Issues#

- 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-v4`** , 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.

Appendix: CPM memory layout#

The following information applies to the Freescale 8548 processors, which contain the Communications Processor Module (CPM). Neutrino supports various devices found within this processors, namely the SCC and FCC channels found on both the 8548 CPM. The CPM module contains 32K of dual-port RAM, which is divided up as follows:

- The first 16K is divided into eight banks, each 2K in size, that can be used for buffer descriptors or data buffers for the various CPM peripherals.
- The next 4K is reserved for parameter RAM, used to store operating parameters for the various CPM devices.

The Buffer Descriptor Tables and Data Buffers for the SCC channels are allocated as follows:

- For the SCC channels, bank 8 of the dual port RAM BD/Data area is used for Buffer Descriptor Tables and Data buffers. The SCC channels have their own dedicated area for parameter RAM, within the Parameter RAM area. Bank 8 is divided as follows:

Item	Space
SCC1 Buffer Descriptor Table	0x100 bytes
SCC1 Data Buffers	0x100 bytes
SCC2 Buffer Descriptor Table	0x100 bytes
SCC2 Data Buffers	0x100 bytes
SCC3 Buffer Descriptor Table	0x100 bytes
SCC3 Data Buffers	0x100 bytes
SCC4 Buffer Descriptor Table	0x100 bytes
SCC4 Data Buffers	0x100 bytes