Release Notes for the QNX Neutrino 6.4.0 BSP for Freescale MPC8536 DS (Development System a.k.a Calamari) 1.0.0#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: Freescale mpc8536 ds (Development System a.k.a Calamari)
- U-Boot 1.3.3-rc3-00334-gf214729-dirty (Jun 2 2008 - 14:07:04)
- Spansion S29GL01GP (128 MB) NOR Flash
- Samsung K9NBG08U5A (32 MB) NAND Flash

Host development system#

- QNX Momentics 6.4.0, one of the following host systems:
  - QNX Neutrino 6.4.0
  - Microsoft Windows Vista, XP SP2 or SP3, 2000 SP4
  - Linux Red Hat Enterprise Workstation 4 or 5, Red Hat Fedora Core 6 or 7, Ubuntu 6.0.6 LTS or 7, or SUSE 10
- Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
- RS-232 serial port
- NULL-modem serial cable
- Ethernet link

Getting Started#

Step 1: Connect your hardware#

Connect the serial cable to the first serial port of the MPC8536DS board to 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 the mouse to the second serial port on your host, because some terminal programs require the first serial port.

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
- Flow control: none

Step 4: Setting up the environment#

- Apply power to the target board. You should see the U-Boot ROM monitor output.
- You can use TFTP download (the default) or serial download to transfer an OS image to the board, as described below.

Step 4A: TFTP download#

This method requires a raw image, which the buildfile creates by default. On your target, type the following, filling in the appropriate IP addresses and `ifs` file:

```
=> setenv ethact eTSEC1
=> setenv ipaddr 172.18.80.231
=> setenv serverip 172.18.80.174
=> setenv bootfile ifs-mpc8536ds.raw
=> setenv loadaddr 0x100000
=> setenv bootcmd 'tftpboot $loadaddr $bootfile; go $loadaddr'
=> setenv bootdelay 2
=> saveenv
Saving Environment to Flash...
Un-Protected 1 sectors
Erasing Flash...
. done
Erased 1 sectors
Writing to Flash... 9....8....7....6....5....4....3....2....1....done
Protected 1 sectors
=> 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:

```
eTSEC1: No link.
Speed: 100, full duplex
Using eTSEC3 device
TFTP from server 172.18.80.174; our IP address is 172.18.80.231
Filename 'ifs-mpc8536ds.raw'.
Load address: 0x100000
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         ############################################################
done
Bytes transferred = 4652048 (46fc10 hex)
## Starting application at 0x00100000 ...
```

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

```
Welcome to QNX Neutrino 6.4.0 on the Freescale MPC8536DS board
```

You can now 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 " Driver Command Summary" section.

Creating a flash partition#

- Enter the following command to start the flash filesystem driver:

devf-generic -s 0xE8000000,128M,,,128k,2,1 -r

- To prepare the area for the partition, enter the following command:

flashctl -p/dev/fs0 -l2M -o16M -ve

- Format the partition:

flashctl -p/dev/fs0p0 -l2M -o16M -vf

- Slay, and then restart the driver:

slay devf-generic &
devf-generic -s 0xE8000000,128M,,,128k,2,1 -r &

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

Summary of driver commands#

The following table summarizes the commands to launch the various drivers.

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	`startup-mpc8536ds`	.	.	`src/hardware/startup/boards/mpc8536ds`
Serial	`devc-ser8250-mpc8540 -e -c396000000 -b115200 0xFFE04500,26 0xFFE04600,26`	devc-ser8250-mpc8540	.	`src/hardware/devc/ser8250`
Flash (NOR)	`devf-generic -s 0xE8000000,128M,,,128k,2,1 -r`	devf-generic flashctl	.	`src/hardware/flash/boards/generic`
Flash (Nand)	`fs-etfs-mpc8536ds2048 -D addr=ffe05000,cs=2,bfaddr=0xffff00000,syspage -m /fs/etfs`	fs-etfs-mpc8536ds2048 etfsctl	.	`src/hardware/etfs/nand2048/mpc8536ds2048`
PCI	`pci-mpc8536 -dmpc8536 pex=1`	pci-mpc8536 pci	.	`src/hardware/pci/mpc8536`
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`
I2C	`For I2C interface 1: i2c-mpc8536 -i27 -p0xFFE03000`	i2c-mpc8536	.	`src/hardware/i2c/mpc8536`

	`For I2C interface 2: i2c-mpc8536 -i27 -p0xFFE03100`			
SPI	`spi-master -dmpc8536`	spi-master	spi-mpc8536.so	`src/hardware/spi/mpc8536`
USB	`io-usb -d ehci-8349 ioport=0xFFE22100,irq=12,ioport=0xFFE23100,irq=30`	io-ubs usb*	devu-ehci-8349.so libusbdi.so [illegible]-drivers	*prebuilt only*
SD card	`devb-mmcsd mmcsd bw=4`	devb-mmcsd	libcam.so fs-dos.so cam-disk.so	`src/hardware/devb/mmcsd`

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:#

Note:
1) There is a primary clock input SYSCLK (also referred to as the system clock) to the processor, value of which can range from 33MHz - 166 MHz. The actual value is determined by a switch setting on the board (SW3 (3:5) - default 66 MHz).

2) A CCB clock signal is derived from the above SYSCLK using a ratio defined by another switch setting on the board (SW2 (1:4) - default 6:1)

3) The CPU core clock is derived from the CCB clock using other switch settings on board (SW10 (1:3) and SW7 (5)), which detrmine the ratio (Core_clock: CCB - default 5:2)

Now in startup, "clock_rate" variable is used to get the command line option "-t" you are referring as platform_clock. You need to determine the switch setting first (i.e. SW2 (1:4)) and then calculate SYSCLK using CCB printed in Uboot.

E.g, if switch setting is (SW2 (1:4) is default 6:1) , And CCB printed in Uboot is 400Mhz, it derives SYSCLK as 66MHz i.e. 400MHz/6.

Ethernet:#

Note:
For more information about io-pkt use the following link, http://community.qnx.com/sf/projects/networking

Known issues for this BSP#

- MPC8536 PCI driver only handles one PCI bus at a time.