Release Notes for the QNX Neutrino 6.4.0 BSP for Freescale MPC8572 DS 1.0.0#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: Freescale mpc8572 DS
- ROM Monitor version UBoot v 1.2.0
- K9NBG08U5A-P (4G x 8 Bit) 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

System layout#

The table below depicts the memory layout for the image and for the flash.

Memory layout

Item	Address
OS image loaded at:	0x00100000
OS image begins execution at:	0x00101e38
Flash base address	0xff000000
Monitor flash offset	0xff800000
TSEC1 base address	0xe0002400 (IRQ: 13, 14, 18)
TSEC2 base address	0xe0002500 (IRQ: 19, 20, 24)
TSEC3 base address	0xe0002600 (IRQ: 15, 16, 17)
TSEC4 base address	0xe0002700 (IRQ: 21, 22, 23)
Ethernet base address	0xB4020000 (IRQ: 168)
Serial base address	0xffe04500 (IRQ: 26)

Getting Started#

Step 1: Connect your hardware#

Connect the serial cable to the first serial port of the MPC8572DS 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 ipaddr 192.168.200.2
=> setenv serverip 192.168.200.1
=> setenv bootfile ifs-mpc8572ds.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...
flash erase done
Erased 1 sectors
Writing to Flash... 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:

```
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-mpc8572ds.raw'
Load address: 0x100000
Loading: ################################################################
   ################################################################
   ################################################################
   ################################################################
   ################################################################
   ################################################################
   ################################################################
```

```
   #################################################################
   #################################################################
   #################################################################
   #################################################################
   ###############################################################
done
Bytes transferred = 3956736 (3c6000 hex)
## Starting application at 0x00100000 ...
```

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

```
Welcome to QNX Neutrino 6.4 on the PowerPC 8572DS 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).

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-mpc8572ds.srec /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

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

```
System page at phys:0000c000 user:0000c000 kern:0000c000
Starting next program at v00133af4
Welcome to QNX Neutrino 6.4 on the PowerPC 8572DS 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 " Driver Command Summary" section.

Creating a flash partition#

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

fs-etfs-mpc8572ds2048 -r65536 -D cs=2

- Stop the filesystem on the device:

etfsctl -d /dev/etfs2 -s

- Format the filesystem:

etfsctl -d /dev/etfs2 -f

- Make the filesystem continue:

etfsctl -d /dev/etfs2 -c

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

Driver Command Summary#

The driver command lines below are specific to the Freescale MPC8572DS board. See the online docs for each driver for additional command-line options and other details.

NOTE: Some of these 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.

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	`startup-mpc8572ds`	.	.	`src/hardware/startup/boards/mpc8572ds`
Serial	`devc-ser8250-mpc8540 -e -c600000000 -b115200 0xffe04500,26`	devc-ser8250-mpc8540	.	`src/hardware/devc/ser8250`
PCI	`pci-mpc8572 pex=3 (PEX controller 3 ) pci-mpc8572 pex=2 (PEX controller 2 )`	pci-mpc8572 pci	.	`src/hardware/pci/mpc8572`

	pci-mpc8572 pex=1 (PEX controller 1) pci-mpc8572 pex=0 (PEX controller 0)			
Network	io-pkt-v4-hc -d mpc85xx mac=001100111111 -ptcpip	io-pkt-v4-hc ifconfig 1,syspage	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=001100111111	io-pkt-v4-hc ifconfig	devnp-mpc85xx.so devnp-mpcsec.so libsocket.so	"Binary form only:" prebuilt/ppcbe/lib/dll/devnp-mpcsec.so
ETFS	fs-etfs-mpc8572ds2048 -r65536 -D cs=2 -m /fs/etfs	fs-etfs-mpc8572ds2048 etfsctl	.	src/hardware/etfs/nand2048/mpc8572ds2048
I2C	i2c-mpc8572 -i27 -p0xe0003000 (Interface 1) i2c-mpc8572 -i27 -p0xe0003100 (Interface 2)	i2c-mpc8572	.	src/hardware/i2c/mpc8572

Serial#

The value passed to the -c option depends on the CCB clock frequency as displayed by U-Boot at boot time, which may change depending upon board dip switch settings. For instance, CCB:600Mhz translates to -c600000000.

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 (for MPC8572E):
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.

I2C:#

Run both I2C channels:

```
i2c-mpc8572 --u0 -i27 -p0xe0003000
i2c-mpc8572 --u1 -i27 -p0xe0003100
```

Note:
For more information about these commands, see the Neutrino Utilities Reference.

Known issues for this BSP#

- MPC8572 DS boards may ship with different revisions of U-Boot. Depending on the U-boot revision, CCSRBAR may be configured differently. . By default, the startup module assumes a CCSRBAR=0xe0000000. If U-boot configures the CCSRBAR to something other than the default, then the build file must be modified such that the CCSRBAR is passed to the startup via the -c flag. E.G. `startup-mpc8572ds -v -c0xffe00000 .` A simple way to verify the CCSBAR(i.e. immr_base) within U-Boot is to issue the `bdinfo` command.
- Some MPC8572 DS boards may come equipped with a non-default system clock. By default, the startup module assumes a system clock of 66MHz. If the system clock is non standard, then the build file must be changed such that the actual system clock frequency is passed to the startup as a parameter via the -t option. E.G. Assuming a 100MHz system clock, the buildfile should be modified as follows: `startup-mpc8572ds -v -t100000000`
- This BSP works only on the Freescale MPC8572DS development system. Since this BSP is based on the Freescale CDS MPC85xx BSP, it may contain generic code related to the MPC85xx series of processors.
- Unplugging the network cable results in an "Unknown" media speed on reinsertion. The driver is still fully functional in this state. (Ref# 21992)
- 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.
- PCI server is still work in progress and might fail to launch properly. The problem have been observed on few graphic cards.
- The PCI server is not able to allocate prefetchable memory to graphics adapters on the PCI or PCI Express buses.
- A "SIGSEGV" signal can be generated when terminating the pci server via "slay" command.