

Release Notes for the QNX Neutrino 6.4.0 BSP for Freescale MPC8313E RDB 1.0.0#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: Freescale MPC8313E RDB REVC
- ROM Monitor version U-Boot 1.3.0 (Dec 22 2008 - 11:19:29) MPC83XX
- Spansion S29GL054A9QTFIR3 (8 MB) NOR Flash
- Samsung K9F5608UOD (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#

1. Connect the serial cable to the first serial port (UART1: upper port of P11) of the MPC8313E RDB 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.

2. Connect the network cable to the lower-right-most TSEC2 port on the MPC8313E RDB board

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.

The MPC8313E-RDB REVC board contains eTSEC1: five 10/100/1000 BaseT RJ-45 interface using Vitesse VSC7385 L2 switch and eTSEC2: one 10/100/1000 BaseT RJ-45 interface using Marvell 88E1111 PHY(lower-right-most is TSEC2, other five are eTSEC1).

Step 4A: Loading the Firmware for Vitesse VSC7385 L2 switch#

The MPC8313E-RDB board contains a Vitesse VSC7385 L2 switch connected to TSEC1 device. This switch essentially requires to load/execute a Vitesse firmware before booting any OS on the MPC8313E-RDB platform, in order to be able to use TSEC1. This firmware comes factory preloaded on MPC8313E-RDB board's NOR flash.

```
U-Boot 1.3.0 (Dec 22 2008 - 11:19:29) MPC83XX
```

```
Reset Status:
```

```
CPU: e300c3, MPC8313E, Rev: 21 at 333.333 MHz, CSB: 166 MHz
```

```
Board: Freescale MPC8313ERDB
```

```
I2C: ready
```

```
DRAM: 128 MB
```

```
FLASH: 8 MB
```

```
NAND: 32 MiB
```

```
In: serial
```

```
Out: serial
```

```
Err: serial
```

```
Net: TSEC0, TSEC1 [PRIME]
```

```
Hit any key to stop autoboot: 0
```

```
=>
```

```
=> cp.b fe710000 40000 11000; go 40004
```

```
## Starting application at 0x00040004 ...
```

```
VSC7385 Version = 0x273850e9
```

```
Value in IRAM matches binary, total bytes 0x2000 loaded
```

```
## Application terminated, rc = 0x0
```

```
=>
```

Note:

There are 2 Vitesse VSC7385 L2 switch firmware in flash memory from 0x70000000 to 0x7ffffff. If they have been erased by accident, you could restore them in flash as following. Please be sure to use the actual file size in hex that is returned from the tftp command.

Firmware Loader Program:

```
=> tftp 1000000 /xfer/vsc7385_load.bin
```

```
=> protect off fe710000 fe72ffff
```

```
=> erase fe710000 fe72ffff
```

```
=> cp.b 1000000 fe710000 10408 <- This is the hex size of vsc7385_load.bin returned by tftp
```

```
=> protect on all
```

Firmware Image

```
=> tftp 1000000 /xfer/vsc2bin
```

```
=> protect off fe7f0000 fe7f3fff
```

```
=> erase fe7f0000 fe7f3fff
```

```
=> cp.b 1000000 fe7f0000 1ff8 <- This is the hex size of vsc2bin returned by tftp
```

=> protect on all

Step 4B: 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 TSEC1
=> setenv ipaddr 172.18.80.142
=> setenv serverip 172.18.80.174
=> setenv bootfile ifs-mpc8313rdb.raw
=> setenv loadaddr 0x200000
=> setenv ldswitch 'cp.b fe710000 40000 11000; go 40004'
=> setenv bootcmd 'tftpboot $loadaddr $bootfile; run ldswitch; go $loadaddr'
=> setenv bootdelay 2
=> saveenv
Saving Environment to Flash...
Un-Protected 1 sectors
Erasing Flash...
. done
Erased 1 sectors
Writing to Flash... done
Protected 1 sectors
=> boot
```

Note:

There are some conflict name of eTSEC interface between UBoot and MPC8313E Reference Design Board User's Guide. If connect eTSEC1 to download, please use "setenv ethact TSEC0". If connect eTSEC2 to download, please use "setenv ethact TSEC1" .

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 TSEC1 device
TFTP from server 10.42.97.136; our IP address is 10.42.104.42
Filename 'ifs-mpc8313rdb.raw'.
Load address: 0x200000
Loading: #####
#####
#####
#####
#####
done
Bytes transferred = 1483860 (16a454 hex)
## Starting application at 0x00200000 ...
```

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

```
Welcome to QNX Neutrino 6.4.0 on the Freescale MPC8313E RDB 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 -s0xFE000000,8M
```

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

```
flashctl -p/dev/fs0 -l1M -o6M -ve
```

- Format the partition:

```
flashctl -p/dev/fs0p0 -l1M -o6M -vf
```

- Slay, and then restart the driver:

```
slay devf-generic &  
devf-generic -s0xFE000000,8M &
```

You should now have a /fs0p1 directory which you can copy files to.

Note:
DO NOT erase the first 1M and last 1M memory in NOR flash memory. They are used for UBoot and other applications(e.g. The Vitesse switch firmware).

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-mpc8313e-rdb	.	.	src/hardware/startup/boards/mpc8313e-rdb
Serial	devc-ser8250 -e -F -c166000000 -b115200 0xe0004500,0x090xe0004600,0x0A	devc-ser8250	.	src/hardware/devc/ser8250
Flash (NOR)	devf-generic -s0xFE000000,8M	devf-generic flashctl	.	src/hardware/flash/boards/generic
Flash (Nand)	fs-etfs-mpc8313erdb512 -D cs=1 -m /fs/etfs	mpc8313erdb512 etfsctl	.	src/hardware/etfs/nand512/mpc8313erdb512
PCI	pci-mpc8313	pci-mpc8313 pci	.	src/hardware/pci/mpc8313
Network	eTSEC1: io-pkt-v4 -d mpc85xx pci=0,syspage,emu_phy=0,irq_tx=32,irq_rx=33,irq_err=34 eTSEC2: io-pkt-v4 -d mpc85xx pci=1,ioport=0xe0025000,irq_tx=35,irq_rx=36,irq_err=37	io-pkt-v4-hc ifconfig	devnp-mpc85xx.so libsocket.so	prebuilt binary only: prebuilt/ mpc8313e-rdb/ dll/devnp-mpc85xx.so

	Both: io-pkt-v4 -dm <code>pc85xx pci=0,syspage,emu_phy=0,irq_tx=32,irq_rx=33,irq_err=34 -d mpc85xx pci=1,ioport=0xe0025000,irq_tx=35,irq_rx=36,irq_err=37</code>			
I2C	For I2C interface 1: <code>i2c-mpc5200 -p0xe0003000 -c88000000 -i14 --u0</code> For I2C interface 2: <code>i2c-mpc5200 -p0xe0003100 -c88000000 -i15 --u1</code>	<code>i2c-mpc5200</code>	.	<code>src/hardware/i2c/mpc5200</code>
SPI	<code>spi-master -d mpc8349</code>	<code>spi-master</code>	<code>spi-mpc8349.so</code>	<code>src/hardware/spi/mpc8349</code>
USB	<code>io-usb -d ehci-8313 ioport=0xe0023100</code>	<code>io-ubs usb* irq=38</code>	<code>devu-ehci-8313.so libusbdi.so</code> class drivers	prebuilt binary only: <code>prebuilt/ppcbe/lib/dll/devu-ehci-8313.so</code>
RTC	Set OS clock from RTC: <code>rtc hw /dev/i2c0</code> Set RTC from OS clock: <code>rtc -s hw /dev/i2c0</code>	<code>rtc date</code>	.	QNX640 prebuilt binary only: NOTE: requires I2C driver to be running first
WDT	<code>wtdkick-mpc83xx</code>	<code>wtdkick-mpc83xx</code>	.	<code>src/hardware/support/wtdkick-mpc83xx</code> NOTE: To enable the WatchDog Timer , Please pass -w option to startup: <code>startup-mpc8313e-rdb -w</code>

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.

Ethernet:#

For eTSEC1:five 10/100/1000 BaseT RJ-45 interface using Vitesse VSC7385 L2 switch:

```
io-pkt-v4 -dmpc85xx pci=0,syspage,emu_phy=0,irq_tx=32,irq_rx=33,irq_err=34
```

Note:

The Vitesse switch firmware must first be loaded before the QNX image is run. This must be done after every reset of the board. Add the following to the U-boot environment:

```
setenv ldswitch 'cp.b fe710000 40000 11000; go 40004'
```

```
setenv bootcmd 'tftpboot $loadaddr $bootfile; run lds witch; go
```

For eTSEC2: one 10/100/1000 BaseT RJ-45 interface using Marvell 88E1111 PHY:

```
io-pkt-v4 -d mpc85xx pci=1,ioport=0xe0025000,irq_tx=35,irq_rx=36,irq_err=37
```

For Both eTSEC1 and eTSEC2:

```
io-pkt-v4 -dm pc85xx pci=0,syspage,emu_phy=0,irq_tx=32,irq_rx=33,irq_err=34 -d mpc85xx pci=1,ioport=0xe0025000,irq_tx=
```

Note:

The latest sources for devnp-mpc85xx .so is available from the [networking project](#).

Known issues for this BSP#

None