Release Notes for the QNX Neutrino 6.3.2 BSP for Freescale MPC8313E RDB 1.0.0#

System requirements#

Target system#

- QNX Neutrino RTOS 6.3.2
- Board version: Freescale mpc8313e Rev A3
- ROM Monitor version UBoot v 1.1.6 (Apr 28 2007 - 14:48:50) MPC83XX
- EON EN29LV640 (8 MB) NOR Flash
- Samsung K9F5608UOD (32 MB) NAND Flash

Host development system#

- QNX Momentics 6.3.2, one of the following host systems:
  - QNX Neutrino 6.3.2
  - 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 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.

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: Loading the Firmware#

The MPC8313E-RDB board contains a Vitesse VSC7385 switch connected to TSEC0 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 TSEC0. This firmware comes factory preloaded on MPC8313E-RDB board's NOR flash.

```
=> 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
=>
or
=> run run_vscld1
## Starting application at 0x00040004 ...
VSC7385 Version = 0x273850e9
Value in IRAM matches binary, total bytes 0x2000 loaded
## Application terminated, rc = 0x0
=>
```

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 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... 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: 1000, full duplex
Using TSEC1 device
TFTP from server 172.18.80.174; our IP address is 172.18.80.142
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 6.3.2 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 -o5M -ve

- Format the partition:

flashctl -p/dev/fs0p0 -l1M -o5M -vf

- Slay, and then restart the driver:

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

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

Summary of driver commands#

The driver command lines below are specific to the Freescale MPC8313E RDB board. See the online docs for each driver for additional command-line options and other details.

Note: 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-mpc8313e-rdb -vvvv
```

Serial:#

Command:

```
devc-ser8250 -e -F -c166000000 -b115200 0xe0004500,0x09 0xe0004600,0x0A
```

Required binaries:

- devc-ser8250

Flash(NOR):#

Command:

```
devf-generic -s 0xFE000000,8M
```

Required binaries:

- devf-generic
- flashctl

PCI:#

Command:

```
pci-mpc83xx
```

Required binaries:

- pci
- pci-mpc8313

SPI:#

Command:

```
spi-master -d mpc8349
```

Required binaries:

- spi-master

Required libraries:

- spi-mpc8349.so

> **Note**: SPI doesn't support DMA mode.

I2C:#

Command:

```
i2c-mpc5200 -p0xe0003000 -c88000000 -i14 --u1
i2c-mpc5200 -p0xe0003100 -c88000000 -i15 --u0
```

Required binaries:

- i2c-mpc5200

Required libraries:

- libi2c-master.a

USB:#

Command:

```
io-usb -d ehci-8313 ioport=0xe0023100,irq=38 &
```

Required binaries:

- io-usb
- usb
- devb-umass
- devu-ehci-8313.so
- libusbdi.so
- io-blk.so
- libcam.so
- fs-dos.so
- fs-qnx4.so
- fs-ext2.so
- cam-disk.so

Ethernet:#

Command:

```
io-net -dmpc85xx syspage -ptcpip &
```

Required binaries:

- io-net
- ifconfig

Required libraries:

- devn-mpc85xx.so
- libsocket.so
- npm-tcpip.so

ETFS NAND flash#

Command:

```
fs-etfs-mpc8313erdb512 -D cs=1
```

Required binaries:

- fs-etfs-mpc8313erdb512
- etfsctl

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

Known issues for this BSP#

- U-Boot has a known bug on this board (as documented in U-Boot/board documentation), that device TSEC0 fails to load the image in the first run, as evidenced in the following log:

```
=> tftpboot 200000 image
    Speed: 1000, full duplex
    Using TSEC0 device
    TFTP from server 172.18.80.174; our IP address is 172.18.80.142
    Filename 'ifs-mpc8313rdb.raw'.
    Load address: 0x200000
    Loading: Got error 4

    Abort
    => tftpboot 200000 ifs-mpc8313rdb.raw
    Speed: 1000, full duplex
    Using TSEC0 device
    TFTP from server 172.18.80.174; our IP address is 172.18.80.142
    Filename 'ifs-mpc8313rdb.raw'.
    Load address: 0x200000
    Loading: ###################################################################
    ###################################################################
    ###################################################################
    ###################################################################
    ##############################
    done
    Bytes transferred = 1483860 (16a454 hex)
    => go 200000
        ## Starting application at 0x00200000 ...
```

Workaround: Either TSEC1 can be used for loading image in the first run, or TSEC0 can be used to load image successfully second run onwards.

- MPC8313E RDB reboot fails with "procnto-600-instr" (instrumented kernel) in QNX 6.3.2, but works fine in QNX 6.4.0.

Workaround: "procnto-600-smp-instr" kernel can be used to solve this issue.

- Nicinfo may show "Current Operation Rate" as "Unknown" for en0 only, but the Ethernet driver and en0 interface works fine.

- mpc85xx.so driver will not support the lan=<#> option.