Release Notes of the QNX 6.4.0 BSP for Freescale MPC8323E RDB Trunk#

System requirements#

Target system

- QNX Neutrino RTOS 6.4.0
- Board version: Freescale MPC8323E RDB
- 16 MB flash
- 64 MB DDR RAM
- ROM Monitor version U-Boot 1.1.6

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:	0x200000
Flash base address	0xFE000000
U-BOOT base address in flash	0xFE000000
IPL-OS image base address in flash	0xFE100000
Flash file system base address in flash	0xFE300000

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.Connect the serial cable to the first serial port of the MPC8439E MDS CPU card to the first 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.

2.Connect an RJ-45 Ethernet cable to P3-Enet4 on your target.

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:

```
Clock configuration:
  Coherent System Bus:  133 MHz
  Core:               333 MHz
  QE:                 200 MHz
  BRG:                 100 MHz
  Local Bus Controller: 133 MHz
  Local Bus:            66 MHz
  DDR:                 266 MHz
  SEC:                133 MHz
  I2C1:               133 MHz
CPU: MPC8323E, Rev: 11 at 333.333 MHz
Board: Freescale MPC8323ERDB
I2C:   ready
DRAM:
   DDR RAM: 64 MB
FLASH: 16 MB
PCI clock is 66MHz
In:    serial
Out:   serial
Err:   serial
Net:   UEC: PHY is Generic MII (2430d80)
FSL UEC0: Full Duplex
FSL UEC0: Speed 100BT
FSL UEC0: Link is up
UEC: PHY is Generic MII (2430d80)
FSL UEC1: Full Duplex
FSL UEC1: Speed 100BT
FSL UEC1: Link is up
FSL UEC0, FSL UEC1
Hit any key to stop autoboot:  0
=>
```

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-mpc8323erdb.raw
=> setenv loadaddr 0x200000
=> setenv bootcmd 'tftpboot $loadaddr $bootfile; go $loadaddr'
=> setenv bootdelay 2
=> saveenv
Saving Environment to Flash...
Un-Protected 2 sectors
Erasing Flash...
.. done
Erased 2 sectors
Writing to Flash... done
Protected 2 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:

```
Using FSL UEC0 device
TFTP from server 10.42.97.136; our IP address is 10.42.104.42
Filename '/root/ifs-mpc8323erdb.raw'.
Load address: 0x200000
Loading: ##################################################################
         ##################################################################
         ##################################################################
         ##########################################
done
Bytes transferred = 1208216 (126f98 hex)
## Starting application at 0x00200000 ...

System page at phys:0000b000 user:0000b000 kern:0000b000
Starting next program at v00244860
Welcome to QNX Neutrino Trunk on the Freescale MPC8323E RDB board
#
#
```

Step 4B: Using the U-Boot ROM monitor via Serial port download

1.Apply power to the target board and press any key to stop autobooting. You should see the U-Boot ROM monitor output.

2.You can use the serial port to transfer an OS image to the board. This method requires a raw image, which the buildfile creates by default.

3.Use loadb to download the OS image through the serial port. You should see the ROM monitor waiting for transfer:

```
=> loadb 0x200000 115200
  ## Ready for binary (kermit) download to 0x00200000 at 115200 bps..
```

4.On your host PC, run kermit to send the appropriate binary file. If you are using hyperterm; go to the pull-down menu Transfer and select **`Send File...`**; you will get the Send File pop-up window. In the Send file window, select the Kermit Protocol and the correct OS image filename, and then click Send to download the image. This download method will take several minutes to complete. Once complete, you sould see the following message:

```
## Total Size      = 0x00121368 = 1184616 Bytes
## Start Addr      = 0x00200000
=>
```

5.Start the OS image:

```
=> go 0x200000
```

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

```
## Starting application at 0x00200000 ...

System page at phys:0000b000 user:0000b000 kern:0000b000
Starting next program at v00244860
Welcome to QNX Neutrino Trunk on the Freescale MPC8323E 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 "Summary of driver commands" section.

Creating a flash partition#

Note: Be aware that the ROM Monitor is housed in flash. Check the target hardware manual for the position of the ROM Monitor in flash. Use extreme caution if you attempt to modify the contents on the flash. If critical contents (e.g. the U-BOOT code at the bottom of the memory area starting from address 0xFE000000) become damaged, QNX Neutrino may no longer boot from the flash.

You may be able to repair a flash by reprogramming it using an ICE or a flash programmer.

The commands below will destroy any data sitting in flash at an offset of 3 MB for a length of 13 MB.

The mkflash script creates a 2 MB flash image (in binary format) that you can download to the MPC8323E RDB board. This 2 MB image contains the following:

- IPL image
- OS image

Note:

- This leaves 13 MB of free space for a flash filesystem partition
- The first 1 MB of flash is reserved for U-BOOT.
- The next 2 MB of flash space are reserved for a simplified IPL and a small OS image that starts the flash filesystem driver in order to pull the other drivers from flash.
- The simplified IPL does not replace U-BOOT because it does not initialize the board, it only transfers the OS image from flash to RAM and starts the kernel image in RAM.

Flash partitions#

To reformat the flash partition, do the following (for an 16 MB flash part):

1. Enter the following command to start the flash filesystem driver:

```
devf-generic -s0xfe000000,16M
```

2. To prepare the area for the partition, enter the following command:

```
flashctl -p/dev/fs0 -l13M -o3M -ve
```

In order to avoid erasing the ROM monitor and IPL-OS images (which are located in the first 3 MB of flash), use the -o option to erase starting at offset 3MB (i.e. -o3M) and use the -l option to specify the length of flash to erase(i.e. -l13M). This leaves the top part of the flash intact with the U-BOOT, the IPL, and the OS image. Both the -o and -l options can be used at the same time.

3. Format the partition:

```
flashctl -p/dev/fs0p0 -l13M -o3M -vf
```

4. Slay, and then restart the driver: `slay devf-generic &`

devf-generic -s0xfe000000,16M & }

You should now have a /fs0p0 directory that you can copy files to.

Programming OS flash image into target board flash#

On your target board, you can use U-BOOT to transfer an IPL-OS flash image flash_img to the target board RAM, and then program it to the flash.

1.Use tftp to download the image:

```
=> tftp 0x200000 flash_img
Using FSL UEC0 device
TFTP from server 10.42.104.16; our IP address is 10.42.104.17
Filename 'flash_img'.
Load address: 0x200000
Loading: ##################################################################
         ##################################################################
         ##################################################################
         ####################################
done

done
Bytes transferred = 1184768 (121400 hex)
```

2.Use protect to turn off the flash protection:

```
=> protect off all
Un-Protect Flash Bank # 1
```

3.Use erase to erase 2 MB of flash (from 0xfe100000 to 0xfe2fffff):

```
=> erase 0xfe100000 0xfe2fffff

................ done
```

4.Use cp.b to program the IPL-OS image to flash:

```
=> cp.b 0x200000 0xfe100000 0x121400
Copy to Flash... done
```

5. Use the setenv and saveenv commands to configure booting from flash and save the environment:

```
=> setenv bootcmd 'go 0xfe100000'
=> saveenv
Saving Environment to Flash...
Un-Protected 1 sectors
Erasing Flash...
flash erase done
Erased 1 sectors
Writing to Flash... done
Protected 1 sectors
```

6. Reset the target, and the U-BOOT will autoboot the IPL-OS flash image at address 0xfe100000.

```
## Starting application at 0xFE100000 ...

QNX Neutrino IPL for MPC8323E RDB Board
Scanning for image        @ 0xFE101000
Found image               @ 0xFE101100
Jumping to startup        @ 0x0020221C

System page at phys:0000c000 user:0000c000 kern:0000c000
Starting next program at v00233d54
   Welcome to QNX Neutrino Trunk on the Freescale MPC8323E RDB board
#
```

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-mpc8323e-rdb`	.	.	`src/hardware/startup/boards/mpc8323e-rdb`
Serial	`devc-ser8250 -e -F -c133000000 -b115200 0xe0004500,0x09 0xe0004600,0x0A`	devc-ser8250	.	`src/hardware/devc/ser8250`
Flash (NOR)	`devf-generic -s0xfe000000,16M`	devf-generic flashctl	.	`src/hardware/flash/boards/generic`
PCI	`pci-mpc83xx`	pci-mpc83xx pci	.	`src/hardware/pci/mpc83xx`
Network	`io-pkt-v4 -ducc_8323 mac=00049fef0301,phy=4,mode=MII,speed=100,channel=3 -ptcpip`	io-pkt-v4 ifconfig	devn-ucc_8323.so libsocket.so	`src/hardware/devn/ucc_8323`
I2C	`For I2C interface 1: i2c-mpc5200 -p0xe0003000 -c88000000 -i14 --u1`	i2c-mpc5200	.	`src/hardware/i2c/mpc8520`

	For I2C interface 2: i2c-mpc5200 -p0xe0003100 -c88000000 -i15 --u0			

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:#

Configuration options:

- phy - PHY address. The default is to scan for PHY.
- mode - PHY mode: MII, RMII, GMII, RGMII, TBI, RTBI. The default is RGMII.
- speed - transfer rate in Mbps: 10, 100, 1000. The default is 1000 Mbps.
- channel - UCC channel: 1,2,3,4,5. The default is UCC1.

PCI:#

> **Note**:
> This driver must be started after the I2C driver for channel 2 is active.

Known Issues#

- In those instances where the the ROM monitor's MAC address is different from the one you pass in when running 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-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.

- The network drive only works on P3-Enet4.This issue will be fixed in the future release .