Release Notes for the QNX Neutrino 6.4.1 BSP for RMI XLR 732 ATX-IB EK Trunk#

System requirements#

Target system

- QNX Neutrino RTOS 6.4.1
- Board version: RMI ATX-I-B Evaluation Board
- XRL732 processor
- 16 MB Intel Strata NOR flash

Host development system

- QNX Momentics 6.4.1
- Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
- RS-232 serial port and serial cable
- Ethernet link

Getting Started#

Starting Neutrino#

Step 1: Connect your hardware

1. Connect one end of a NULL-modem cable to Serial Port A (Top DB9 Connector) on the target board. (For details, see your hardware documentation.)
2. Connect the other end of the NULL-modem cable to a free serial port on your host development system. For the purposes of this doc, we'll assume it's ser1 (e.g. Com1 in Windows).
3. You may have five Ethernet links on the system. Four Gigabit Ethernet Ports are located on the ATX-I-B main board. The other one is located on a PCI NIC card shipped from the manufacturer. For the purposes of this docs, we require you to connect your target board to your network using Gigabit Ethernet Port 0 (gmac0) on the ATX-I-B main board. Please check the ATX-I-B XLR Processor Evaluation Kit Users Manual "Table 1".

Note: We recommend you use the -o option while connecting the target with minicom, i.e. minicom -o xlr732

Step 2: 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 3: Transfer the OS image to the board

On your host machine, start your favorite terminal program with these settings:

- Baud: 38400
- Bits: 8
- Stop bits: 1
- Parity: none
- Flow control: none

Step 4: Transfer the OS image to the board

1. Apply power to the board. You should see output similar to the following:

```
        Compiled @ 0x0000000045303B41 seconds from epoch
Power On reset config = 0xFFFFFFFFB466551F
XLR Junction temperature is 55 Celsius
Type any key to enter board setup 0
board major 1, board minor 1
DRAM SPD Info: Detected XLR version B
Compile-time presumed DRAM channel width:72
----- Configuring DRAM Channel 0 -----
Rank allocation table:
SPD address:0x54 Exists:1 CSid:0 ODTid:0 Logical rank ID:0
SPD address:0x54 Exists:1 CSid:1 ODTid:1 Logical rank ID:1
SPD address:0x55 Exists:1 CSid:2 ODTid:2 Logical rank ID:2
 SPD address:0x55 Exists:1 CSid:3 ODTid:3 Logical rank ID:3
 Memory channel operating at 533 MHz data rate.
 Info: chosen AL: 3
 Trying configuration: CMD:6 SLV_CMD:6 tBOARD:1
 Pass count:2906
 Chosen DRAM strobe settings:
 C:6 SC:6 tBOARD:1
 RX0:65 TX0:8 RX1:65 TX1:8 RX2:65 TX2:8 RX3:65 TX3:8
 RX4:65 TX4:7 RX5:65 TX5:7 RX6:65 TX6:7 RX7:65 TX7:8
 Amount of memory available on channel 0: 2048 MB
 ----- Configuring DRAM Channel 2 -----
 Rank allocation table:
 SPD address:0x56 Exists:1 CSid:0 ODTid:0 Logical rank ID:0
 SPD address:0x56 Exists:1 CSid:1 ODTid:1 Logical rank ID:1
 SPD address:0x57 Exists:1 CSid:2 ODTid:2 Logical rank ID:2
 SPD address:0x57 Exists:1 CSid:3 ODTid:3 Logical rank ID:3
 Memory channel operating at 533 MHz data rate.
 Info: chosen AL: 3
 Trying configuration: CMD:6 SLV_CMD:5 tBOARD:1
 Pass count:3571
 Chosen DRAM strobe settings:
 C:6 SC:5 tBOARD:1
 RX0:65 TX0:8 RX1:65 TX1:8 RX2:65 TX2:8 RX3:65 TX3:8
 RX4:65 TX4:8 RX5:65 TX5:8 RX6:65 TX6:8 RX7:65 TX7:8
 Amount of memory available on channel 2: 2048 MB
 Programming Bridge BARS and MTRS
 Running DRAM mats
 DRAM mats Passed
MAC ADDRESS FROM BOARD PROM 0000002121212121
ARIZONA Board Major Version:     0000000000000001
ARIZONA Board Minor Version:     0000000000000001
cpu_online_map FFFFFFFFFFFFFFFF
CPU Revision: c0b04
Processor Info XLR732 (Revision = B2) (iobase = 0xbef00000)
Bridge Device Mask: 0
Power on reset config: b466551f
Initializing message ring for cpu_0
Running On Board "ATX_I_B"
CPU Frequency = 1200.000000MHz, mips_counter_frequency = 0x47868c00
Total Available Ram = 0x3f00000, Total Heap Size = 0x3d9f400, stackbase = 0x8c1e5fe0
Master cpu id = 0, boot1_cpu_online_map = ffffffff
Detected 32 online CPU(s), map = 0xffffffff
Initializing message ring for cpu_3
```

```
        Initializing message ring for cpu_5
        Initializing message ring for cpu_1
        Initializing message ring for cpu_7
        Initializing message ring for cpu_2
        Initializing message ring for cpu_6
        Initializing message ring for cpu_4
        All cpus successfully started
        [Detected iobus]
        [Detected cfiflash on iobus]
        [Detected pcmcia on iobus]
        [Detected spi4_0 on iobus]
        [Detected spi4_1 on iobus]
        [Detected cpld on iobus]
        [Detected cpld_i2c on iobus]
        [Initializing cpld]
        [Initializing pcmcia inferface in IO mode]
        Disc Controller pcmcia registered.
        [Initializing cfiflash]

        Intel Flash. [16 MB]
        PCI BUS is in PCI mode (DLL settings: 0x12)
        PCI XLR is configured in Host mode
        PCI Host mode Status = a1c
        pci Controller registered.
        [Initializing qdr]
        List of PCI devices found:
           PCI device 10ec:8139    00:01 0
           resource[0] 0x10000000:0x100000ff:0x21
           resource[1] 0xd0000000:0xd00000ff:0x20
        Scanning pcmcia Bus 0
        Found pcmcia_1 disk
        Model: Hitachi XX.V.3.7.0.0
        Firmware: Rev 0.00 Serial#: X0401 20060221061329
        Capacity: 244.3 MB = 0.2 GB (500400 x 512)
        configuring gmac0 in nibble mode @ 25MHz (100Mbps): full duplex mode
        configuring gmac1 in nibble mode @ 2.5MHz (10Mbps): full duplex mode
        configuring gmac2 in nibble mode @ 2.5MHz (10Mbps): full duplex mode
        configuring gmac3 in nibble mode @ 2.5MHz (10Mbps): full duplex mode
        Using flash for environment storage.

=============================================================================================
        RMI Bootloader [Version: 1.00.9] for XLR732 on ATX_I_B Reference Board
             (type h for help)

=============================================================================================
        B2_XLR732 @ ATX_I_B $
```

If you see "Booting in 2 units. Press anykey to halt ...", you should press any key promprly to abort autobooting.

2. At the B2_XLR732 @ ATX_I_B $ prompt, you have 2 options:

either configure with **DHCP** or without **DHCP** (manually):

With **DHCP:**

B2_XLR732 @ ATX_I_B $ ifconfig –i gmac0
B2_XLR732 @ ATX_I_B $ tftpc

This command sets the board to **DHCP** mode so it will get an IP address from a **DHCP** server.

Without **DHCP** or with custom settings:

B2_XLR732 @ ATX_I_B $ ifconfig –i gmac0 – a <static IP address for ATX-I-B>

This command manually sets the IP of the board.

3. Now enter the following command at the B2_XLR732 @ ATX_I_B $ prompt:

B2_XLR732 @ ATX_I_B $ tftpc -s <IP address of TFTP server> -f ifs_phoenix.elf
B2_XLR732 @ ATX_I_B $ run

The run command tells the bootloader to boot the recently ELF-format OS image transfer from tftpc command.

When Neutrino starts, you'll see a message like this:

```
Welcome to QNX Neutrino on the RMI ATX-I-B Board
```

Step 4: Start working with Neutrino OS#

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 "Summary of driver commands" section.

Creating a flash partition#

1. Start the NOR flash filesystem driver by issuing the **`devf-phoenix`** command at the `ksh` prompt, or in the startup script.

2. Prepare the area for the partition. Because the Boot loader is in the first 4MB of flash, you will not want to erase them. Use the -l (length) and -o (offset) options to avoid these areas.

flashctl -p/dev/fs0 -o 4M -l 12M –ev

This command will erase 12 MB of flash. The offset (4MB) ensures that you won't erase the bootloader(< 4 MB) area.

3. Format the partition:

flashctl -p/dev/fs0p0 -o 4M -l 12M –fv

4. Slay, then restart the driver:

slay devf-phoenix
devf-phoenix

You should now have a `/fs0p1` mount on the target to which you can copy files.

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-phoenix -vvvv -P8`	.	.	`src/hardware/startup/boards/phoenix`
Serial	`devc-serphoenix -e -c66000000/16 -b38400 0x1ef14000^2,0x80000011 0x1ef15000^2,0x80000012`	devc-serphoenix	.	`src/hardware/devc/serphoenix`
PCI	`pci-phoenix`	pci-phoenix pci	.	`src/hardware/pci/phoenix`
Flash (NOR)	`devf-phoenix`	devf-phoenix flashctl	.	`src/hardware/flash/boards/phoenix`
Network	`io-pkt-v4 -v -ptcpip -dphoenix`	io-pkt-v4 ifconfig nicinfo* ping*	devn-phoenix.so libsocket.so devnp-shim.so	`src/hardware/devn/phoenix`
I2C	`i2c-phoenix --u0 -m0 -p0x1EF16000 i2c-phoenix --u1 -m1 -p0x1EF17000`	i2c-phoenix	.	`src/hardware/i2c/phoenix`

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.

Known Issues for This BSP#

- I2C driver does not work in 6.4.1.