Release Notes for the AMCC PPC460EX BSP#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4
- Processor: AMCC PPC460EX
- Board version: AMCC Canyonlands - EmbeddedPlanet EP460C 1.1 (DES0225)
- ROM Monitor version: U-Boot 1.3.3-00249-ga524e11 (Jun 30 2008 - 16:05:51)
- 512M DDR2 SDRAM
- 64M Spansion NOR flash (S29GL512N)
- 128M STMicroelectronics NAND flash (NAND01GW3B2B)

Host development system#

- QNX Momentics 6.4, one of the following host systems:
  - QNX Neutrino 6.4
  - 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 (provided with the board)
- Ethernet link

Getting Started#

Starting Neutrino#

Step 1: 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 2: Connect your hardware

- Connect one end of the serial cable to the first serial port UART0 of the EP460C board and the other end of the serial cable 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.
- Connect the Ethernet cable to the upper RJ45 port on the EP460C board and plug in the other end of the Ethernet cable to your local network.

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

- Baud: 115200
- Bits: 8
- Stop bits: 1
- Parity: none
- Flow control: none

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

```
U-Boot 1.3.3-00249-ga524e11 (Jun 30 2008 - 16:05:51)

CPU:   AMCC PowerPC 460EX Rev. A at 800 MHz (PLB=200, OPB=100, EBC=100 MHz)
      Security/Kasumi support
      Bootstrap Option H - Boot ROM Location I2C (Addr 0x52)
      Internal PCI arbiter disabled
      32 kB I-Cache 32 kB D-Cache
Board: Canyonlands - AMCC PPC460EX Evaluation Board, 2*PCIe, Rev. 16
I2C:   ready
DTT:   1 is 25 C
DRAM:  512 MB (ECC not enabled, 400 MHz, CL3)
FLASH: 64 MB
NAND:  128 MiB
PCI:   Bus Dev VenId DevId Class Int
PCIE0: link is not up.
PCIE0: initialization as root-complex failed
PCIE1: link is not up.
PCIE1: initialization as root-complex failed
Net:   ppc_4xx_eth0, ppc_4xx_eth1

Type run flash_nfs to mount root filesystem over NFS

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

Step 3: Setup the environment

Use the **`printenv`** command to show the current settings for **`ipaddr`**, **`ethaddr`**, **`serverip`** etc. On your target, type the following, filling in the appropriate IP addresses and ifs file:

```
=> setenv ipaddr 192.168.1.62
=> setenv serverip 192.168.1.253
=> setenv bootfile /tftpboot/ifs-ep460c.raw
=> setenv loadaddr 0x200000
=> setenv bootcmd 'tftpboot $loadaddr $bootfile; go $loadaddr'
```

Once these parameters are configured, you can use the **`saveenv`** command to store your changes. Refer to the U-Boot documentation for more information.

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.

This method requires a raw image, which is simply a binary image, with a header on the beginning, that allows the bootloader to jump to the very beginning of the image (the raw header), where it executes another jump to the first instruction of the image.

After U-boot is configured, boot the `ifs-ep460c.raw` image as follows (we'll assume it's in a directory called `/tftpboot/`) from the U-boot prompt:

```
=> tftpboot 200000 /tftpboot/ifs-ep460c.raw
```

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:

```
Waiting for PHY auto negotiation to complete.. done
ENET Speed is 100 Mbps - FULL duplex connection (EMAC0)
```

```
Using ppc_4xx_eth0 device
TFTP from server 192.168.1.253; our IP address is 192.168.1.62
Filename '/tftpboot/ifs-ep460c.raw'.
Load address: 0x200000
Loading: #################################################################
         #################################################################
         ##################
done
Bytes transferred = 757600 (b8f60 hex)
=>
```

Now, to run the image, enter:

=> go 200000

You should see output similar to the following, with 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 v00245ea4
Welcome to QNX Neutrino 6.4 on the AMCC PPC460EX Reference 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 5: Replace the U-Boot bootloader with a native QNX IPL and OS image in flash

At some point, you may wish to replace the U-Boot bootloader with the native QNX IPL code. This may be desirable once you've tweaked the OS image exactly the way you want it, and you want the board to boot the image automatically, immediately on power up.

To replace the U-Boot bootloader:

1.Modify your buildfile to generate a binary image.

2.Boot the board as described above, using U-Boot to TFTP-download the raw OS boot image.

3.Ensure that you can "see" the IPL image file `ipl-ep460c` and the binary OS image file `ifs-ep460c.bin` from the image. You can get them on the target using QCONN if you are using the Momentics IDE, or access them remotely over a NFS mount that you have configured on your network.

4.Start the flash filesystem driver and erase the last 128KB of NOR flash as follows:

devf-generic -s0x4CC000000,64M
flashctl -p /dev/fs0 -o 65408K -l 128K -ve

5.Copy `ipl-ep460c` to the flash, as follows:

dd if=ipl-ep460c of=/dev/fs0 bs=131072 seek=511

6.To flash the OS image, erase the blocks of NOR flash starting from last 16 MB region (the following commands assume a maximum OS image size of 3 MB):

flashctl -p /dev/fs0 -o 48M -l 3M -ve
dd if=ifs-ep460c.bin of=/dev/fs0 bs=131072 seek=384

When the copy is complete, you can reboot; it should now boot from the native QNX IPL. You should see output as follows:

```
QNX/Neutrino IPL for AMCC 460EX based Canyonlands Board:
Commands:
 d: download image to RAM
 f: scan flash for image
IPL>
```

7.Enter f or F, and the board should boot from the OS image in flash. You'll see Neutrino boot:

```
System page at phys:0000b000 user:0000b000 kern:0000b000
Starting next program at v00245ea4
Welcome to QNX Neutrino 6.4 on the AMCC PPC460EX Reference Board
#
```

If desired, the IPL code can be modified to eliminate the prompt and automatically boot from the flash without user intervention. To do this, modify the <main.c> file of the IPL source, located under:

$BSP_PATH/src/hardware/ipl/boards/ep460c/

Creating a flash partition#

NOR flash

1. Start the NOR flash filesystem driver by issuing the following command:

devf-generic -s0x4cc000000,64M

2. Prepare the area for the partition. Because the Linux images are in the first 22 MB of flash and U-Boot is in the last 640KB of flash, you may not want to erase them. Use the -l (length) and -o (offset) options to avoid these areas. Assuming that we want to create a 20 MB partition:

flashctl -p/dev/fs0 -o30M -l20M -ve

3. Format the partition:

flashctl -p/dev/fs0p0 -o30M -l20M -vf

4. Slay, then restart the driver:

slay devf-generic
devf-generic -s0x4cc000000,64M

In this example, you have a 20 MB flash partition starting at an offset of 30 MB from the start of flash. You should now have a `/fs0p1` mount on the target to which you can copy files.

NAND flash

1. Enter the following command to start the ETFS driver:

fs-etfs-ep460c_2048 -m/fs/etfs

2. Stop the filesystem on the device, format and continue:

etfsctl -d /dev/etfs2 -s -f -c

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

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-ep460c`	.	.	`src/hardware/startup/boards/ep460c`
Serial	`devc-ser8250 -e -F -c11059200 -b115200 0x4EF600300,33 0x4EF600400,1`	devc-ser8250	.	`src/hardware/devc/ser8250`
Flash (NOR)	`devf-generic -s0x4cc000000,64M`	devf-generic flashctl	.	`src/hardware/flash/boards/generic`
Flash (NAND)	`fs-etfs-ep460c_2048 -m /fs/etfs`	fs-etfs-ep460c_2048 etfsctl	.	`src/hardware/etfs/nand2048/ep460c_2048`
PCI and PCI Express	`pci-ppc460ex`	pci-ppc460ex pci	.	`src/hardware/pci/ppc460ex`
Network	`io-pkt-v4-hc -dppc405-ep460c mac=001122334455,rgmii -ptcpip`	io-pkt-v4-hc ifconfig ping	devn-ppc405-ep460c.so libsocket.so devnp-shim.so	`src/hardware/devn/ppc405`
I2C	`i2c-ppc405 -p0x4EF600700 -i2`	i2c-ppc405	.	`src/hardware/i2c/ppc405`
SPI	`spi-master -dppc405`	spi-master	spi-ppc405.so	`src/hardware/spi/ppc405`
USB	`io-usb -d ehci ioport=0x4bffd0400,irq=93` `io-usb -d ohci ioport=0x4bffd0000,irq=94`	io-usb usb	devu-ohci.so devu-ehci.so libusbdi.so	*prebuilt only* `prebuilt/ppcbe/lib/dll/`
RTC	`rtc hw`	rtc date	.	`src/utils/r/rtc`

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.

RTC#

The on-board M41T62 RTC communicates over I2C, so the `rtc` utility requires the I2C driver to be running.

Known issues for this BSP#

- USB host controller drivers for Open Host Controller Interface (OHCI) and Enhanced Host Controller Interface (EHCI) are operational when started standalone. However, due to a known USB port contention

problem between OHCI and EHCI on certain AMCC processors (including PPC460EX), these drivers currently cannot be used at the same time. You should, therefore, make sure to enable only one of OHCI and EHCI at a time. OHCI can support low, full and high speed devices (with high speed devices operating at full speed). EHCI supports only high speed devices by default. If both high and low/full speed devices are required to run simultaneously with EHCI, then the EHCI driver should be used in conjunction with a high speed hub connected to the root port.

- The PCI Express interface may not work correctly with U-Boot as boot loader. You may have to replace U-Boot with the QNX IPL in order to have PCI Express functioning correctly.
- In Microsoft Windows, certain programs (e.g. Norton Ghost) add directories inside double quotation marks (e.g. ...;`"c:\Program Files\Norton Ghost\";`...) to your **PATH** environment variable. This causes the Cygwin `spawn()` function to fail, which in turn causes `cp` to fail when called by `ln-w`. (Ref# 20046) **Workaround:** Modify your **PATH** environment variable and remove any quotation marks.
- This BSP includes the latest USB stack (io-usb) in the prebuilt directory to fix a bug with 64-bit physical addressing. This fix will be included in future releases of QNX Momentics.