Release Notes for the QNX Neutrino 6.4.0 BSP for Texas Instruments DRX45X EVM (J1DDR) 1.0.1#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: TI DRX45X EVM RevB
- ROM Monitor version UBoot v 1.1.3
- 64 MB AMD flash
- 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#

Connect the serial cable to the UART0 serial port (P2) of the DRX45X board and to the first serial port on the 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.

Step 2: Build the BSP#

You can build a BSP OS image from the source code and 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

Boot OS images by ROM monitor#

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

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

2. Start your target. You should see output similar to the following:

```
U-Boot 1.1.3 (Apr 29 2009 - 11:26:47)

U-Boot code: 81080000 -> 810978DC  BSS: -> 8109C5D4
RAM Configuration:
Bank #0: 80000000 128 MB

******************** J1DDR version 0.03 *******************
qjwang@ti.com

ARM Clock   : 408MHz            DSP Clock   : 552MHz
SDRAM Clock : 110MHz            DDR Clock   : 192MHz
SPANSION NOR Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  3
```

Press any key to stop autobooting, and then you'll see the J1DDR prompt:

```
J1DDR #
```

On your target, type the following, filling in the appropriate IP addresses and ifs file:

```
J1DDR # setenv netmask 255.255.240.0
J1DDR # setenv ipaddr 10.42.101.242
J1DDR # setenv gatewayip 10.42.96.1
J1DDR # setenv serverip 10.42.98.211
J1DDR # setenv 'bootcmd tftpboot 80100000 ifs-drx45x.raw;go 80100000'
J1DDR # saveenv
```

Restart your target. You should see output similar to the following:

```
U-Boot 1.1.3 (Apr 29 2009 - 11:26:47)

U-Boot code: 81080000 -> 810978DC  BSS: -> 8109C5D4
RAM Configuration:
Bank #0: 80000000 128 MB

******************** J1DDR version 0.03 *******************
qjwang@ti.com

ARM Clock   : 408MHz            DSP Clock   : 552MHz
SDRAM Clock : 110MHz            DDR Clock   : 192MHz
SPANSION NOR Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  0
Using MAC Address 00:0E:99:02:60:E9
TFTP from server 10.42.98.211; our IP address is 10.42.101.242
Filename 'ifs-drx45x.raw'.
Load address: 0x80100000
Loading: ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
```

```
        ##################################################################
        ########################################
done
Bytes transferred = 2194176 (217b00 hex)
## Starting application at 0x80100000 ...
Welcome to QNX Neutrino trunk on a TI DRx45x EVM
#
```

Boot OS images by IPL#

Note: Before burning the IPL, you might want to save the first 1 MB of flash in order to save the ROM monitor.

1. Modify the build file for IPL to get the binary OS image and run `make clean all` in the BSP's images directory to create a combined IPL/OS image to use when burning the flash.

 # make clean all

2. Download the ipl-ifs-drx45x.bin image to your target board, /dev/shmem/ipl-ifs-drx45x.bin.

3. Make sure the target can read the image you want to copy, and then execute the following commands in the Neutrino shell to burn the IPL-IFS images to flash:

 # devf-generic -s0x20000000,64M
 # flashctl -p/dev/fs0 -l6M -ve
 # cp -V /dev/shmem/ipl-ifs-drx45x.bin /dev/fs0

4. Reboot the target. You should see output similar to the following:

```
Welcome to QNX Neutrino on the Texas Instruments EVMDRX45X board.
Commands:
Press 'S' for serial download, using the 'sendnto' utility
Press 'F' to boot an OS image in flash
```

5. When IPL prompt come up, Either press **s** to transfer the OS image serially using sendnto, or press **f** to scan the flash for an os image.

Note: When booting serially, images larger than 6 MB won't load unless you first modify the IPL source.

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.

Known issues for this BSP#

Note: Please check the version of these release notes on the website for the most up-to-date information.

- If you use the SPI with DMA enabled, the ADE (and hence the MME) locks up. This conflict occurs because the prebuilt DSP/BIOS and ADE image from TI was compiled to use EDMA channel 1. (Ref# 54350)

    - **Workaround:** When you start spi-master, don't specify the **edma** option. If you have acquired the ADE code from TI, you might be able to to resolve the conflict by modifying the EDMA channel numbers on both the DSP and ARM systems.

- The spi-dm644x.so shared object doesn't support the SPIENA pin. The current driver supports only the first two chip selects.

- In those instances where the the ROM monitor's MAC address is different from the one you pass in when running io-pkt-v4, 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-v4, 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 hardware of the J1 DDR board doesn't support audio capture.
    - **Workaround:** The following hardware modification is required to make the capture work: On the EXP_P3 CONNECTOR on CPU board, use a wire to connect Pin 80 and Pin 54 and use a wire to connect Pin 64 and Pin 56.

- The number of simultaneously connected USB peripherals is limited, as the host controller can only support four out standing transactions. For example, attaching a network dongle and a mass storage device will result in one or both devices not operating correctly. (Ref#61212)

- The SMC9000 chipsets seem to have known issues for packet sizes larger than 5K.
    - **Workaround:** Use packet sizes smaller than 5K.

Summary of driver commands#

The driver command lines below are specific to the TI DRX45X EVM board.

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-drx45x -L 0x87e00000,0x200000 -v
```

Serial:#

Command:

```
devc-ser8250-jacinto -e -F -S -b 115200 -c24000000/16 0x01c20000^2,43
```

Required binaries:

- devc-ser8250-jacinto

Flash(NOR):#

Command:

```
devf-generic -s0x20000000,64M
```

Required binaries:

- devf-generic
- flashctl

SPI:#

Command:

```
spi-master -d dra446 base=0x01C4cc00,irq=31,edma=-1,clock=24000000
```

Required binaries:

- spi-master

Required libraries:

- spi-dm644x.so

I2C:#

Command:

```
i2c-dm6446 -p0x01C21000 -i39 --u0
```

Required binaries:

- i2c-dm6446

USB:#

Command:

```
io-usb -d jacinto ioport=0x01c42000,irq=2
```

Required binaries:

- io-usb
- usb

Required libraries:

- devu-jacinto.so
- libusbdi.so

Ethernet:#

Command:

```
io-pkt-v4 -dsmc9000-jacinto ioport=0x2a000000,irq=108
```

Required binaries:

- io-pkt-v4
- nicinfo
- ifconfig

Required libraries:

- devn-smc9000-jacinto.so

- libsocket.so
- devnp-shim.so

Block:#

Command:

```
devb-eide-dm644x blk cache=12M
```

Required binaries:

- devb-eide-dm644x

Required libraries:

- libcam.so
- cam-disk.so
- cam-cdrom.so
- io-blk.so
- fs-qnx4.so
- fs-dos.so
- fs-cd.so

SD:#

Command:

```
devb-mmcsd-drx45x cam quiet mmcsd ioport=0x01C50000,irq=16,dma=20,dma=21 blk cache=2m (SD0 on the  Jacinto base
devb-mmcsd-drx45x cam quiet mmcsd ioport=0x01C41800,irq=54,dma=27,dma=28 blk cache=2m (SD1)
devb-mmcsd-drx45x cam quiet mmcsd ioport=0x01c41c00,irq=42,dma=29,dma=30 blk cache=2m (SD2)
```

Required binaries:

- devb-mmcsd-drx45x

Required libraries:

- libcam.so
- cam-disk.so
- cam-cdrom.so
- io-blk.so
- fs-qnx4.so
- fs-dos.so
- fs-cd.so

Note: the sd device can be mounted with "mount" command or "automount" options. For example:

Using "mount" command:

```
devb-mmcsd-drx45x cam quiet mmcsd ioport=0x01C50000,irq=16,dma=20,dma=21 blk cache=2m
```

- For dos FAT16/FAT32 filesystem: `mount -t dos /dev/hd0t11 /fs/sd0`
- For QNX4 filesystem: `mount /dev/hd0t79 /fs/sd0`

Using "automount" options

- For dos FAT16/FAT32 filesystem:`devb-mmcsd-drx45x cam quiet mmcsd ioport=0x01C50000,irq=16,dma=20,dma=21 blk cache=2m,automount=hd0t6:/fs/sd0 dos exe=all`
- For QNX4 filesystem: `devb-mmcsd-drx45x cam quiet mmcsd ioport=0x01C50000,irq=16,dma=20,dma=21 blk cache=2m,automount=hd0t79:/fs/sd0 qnx4`

Audio:#

Command:

```
io-audio -d dra44x-jacinto
```

Note: This driver must be started after the I2C interface 0 is active on i2c0 (i.e. i2c-dm6446 -p0x01C21000 -i39 --u0).

Required binaries:

- io-audio
- mix_ctl

Required libraries:

- deva-ctrl-dra44x-jacinto.so
- libasound.so.2

ETFS NAND flash#

Command when using 16-bit nand chips:

```
fs-etfs-jacinto -Dwidth=16 -m /fs/etfs
```

Command when using 8-bit nand chips:

```
fs-etfs-jacinto -Dwidth=8 -m /fs/etfs
```

You should then see /dev/etfs1 and /dev/etfs2 partitions. Required binaries:

- fs-etfs-jacinto
- etfsctl

To erase and format the NAND flash partition:

- etfsctl -d/dev/etfs2 -S -e
- etfsctl -d/dev/etfs2 -S -f -c

You should now see the mountpoint /fs/etfs/ which you can use to copy files to.

Note: You have to change the S1 config switch on the base board and start the driver with the correct command line options depending on the width of the chip.

Graphics:#

Command:

```
io-display -dvid=0,did=0
```

Required binaries:

- io-display

Required libraries:

- devg-drx45x.so
- devg-soft3d-fixed.so
- libGLES_CL.so.1
- libffb.so.2
- libgf.so.1

Required config files:

- display.conf
- drx45x.conf

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

About graphics#

The graphics driver only supports the secondary digital display interface (VPSS1) with a single display output. The driver was tested with the LG Philips LB080WV3-B1 digital LCD. Using other displays might result in undesired behavior. The LCD should be connected to the DC5 connector found at the bottom of the base board.

The VPSS controller supports the following layers or “OSD windows”:

- Video Window 0, supporting YUV422 or RGB888 formats
- Video Window 1, supporting YUV422 or RGB888 formats
- OSD Window 0, supporting RGB565 format
- OSD Window 1, supporting RGB565 format

You can control the following features for all layers:

- window start position and size
- horizontal & vertical zoom (2X, 4X)

The OSD windows can support the follow features:

- alpha blending (16 levels)
- transparency (when a pixel matches the transparent color, the pixel will be transparent and the underlying video pixels will be displayed)

You can enable or configure OSD Rectangular Cursor support using drx45x.conf.

Note: Caveats:

- The OSD window restrictions (from the TI documentation) include:
  - Video Window 1 and all OSD windows must be fully contained within Video Window 0.
  - Only one video layer can be set to RGB888 at one time.
  - Only one OSD window can be set to display 16-bit RGB data (but we've been able to run 2 RGB windows).
  - The OSD RGB565 window shouldn't overlap Video Window 1. ** Video Window 1's start X position must be offset a multiple of 16 pixels from the left edge of Video Window 0.
- Transparency of the OSD windows appears to work only with a pixel value of 0, due to an apparent HW limitation.

Optional graphics memory management#

The DRX45X EVM (J1DDR) has two memory buses (EMIFA, EMIFB). We've found that for certain graphics operations, performance is improved if the source and destination surfaces are on separate EMIFs.

Creating a flash partition#

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

devf-generic -s0x20000000,64M

2. Erase the flash, except for the first 6 MB:

Note: Because the ROM monitor or IPL/OS image are in the first 6 megabyte of flash , you may not want to erase them. Use the **-l** (length) and **-o** (offset) options to avoid these areas.

flashctl -p/dev/fs0 -o6M -l58M -ve

3. Format the partition:

flashctl -p/dev/fs0p0 -o6M -l58M -vf

4. Slay and then restart the driver to mount the new partition:

slay devf-generic

devf-generic -s0x20000000,64M

You should now have a /fs0p0 and /fs0p1 directory automounted.