Release Notes for the QNX Neutrino 6.4.0 BSP for Texas Instruments DM644X EVM 1.0.0#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: ti dm644x (Davinci) evm
- 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

System layout#

The table below depicts the memory layout for the image and for the flash.

Memory layout

Item	Address
OS image loaded at:	0x80010000
NOR Flash base address	0x02000000
NAND base address	0x02000000

Getting Started#

Step 1: Connect your hardware#

1. Connect the serial cable to the first serial port of the DM644x EVM and 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 between the ethernet port on the DM644x EVM and your local network.

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 #

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 (Dec  4 2006 - 12:05:38)

U-Boot code: 81080000 -> 81097628  BSS: -> 810A0018
RAM Configuration:
Bank #0: 80000000 256 MB
Flash: 16 MB
In:    serial
Out:   serial
Err:   serial
ARM Clock :- 297MHz
DDR Clock :- 162MHz
Hit any key to stop autoboot:  0
```

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

```
DaVinci EVM #

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

DaVinci EVM # setenv netmask 255.255.240.0
DaVinci EVM # setenv ipaddr 10.42.101.242
DaVinci EVM # setenv gatewayip 10.42.96.1
DaVinci EVM # setenv serverip 10.42.98.211
DaVinci EVM # setenv 'bootcmd tftpboot 80010000;go 80010000'
DaVinci EVM # saveenv
```

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

```
Hit any key to stop autoboot:  0
TFTP from server 10.42.98.211; our IP address is 10.42.101.242
Filename 'ifs-dm6446.raw'.
TFTP from server 10.42.98.211; our IP address is 10.42.101.242
Load address: 0x80010000
Loading: #################################################################
         #################################################################
         #################################################################
         ###########################################################
done
Bytes transferred = 1303956 (13e594 hex)
DaVinci EVM # go 0x80010000
```

```
## Starting application at 0x80010000 ...
CPU0: Dcache: 256x32 WB
CPU0: Icache: 512x32
CPU0: 41069265: arm926 rev 5 283MHz

System page at phys:802ad000 user:fc404000 kern:fc404000
Starting next program at vfe03da20
cpu_startnext: cpu0 -> fe03da20
Welcome to QNX Neutrino 6.4.0 on a TI Davinci-DM644x Platform
#
```

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 “Driver Command Summary” section.

Creating a flash partition#

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

```
devf-generic -s0x02000000,32M
```

2. Unlock the entire flash, except for the first megabyte:

Note:Because the ROM monitor is in the first megabyte of flash, you may not want to erase it. Use the -l (length) and -o (offset) options to avoid this area.

```
flashctl -p/dev/fs0 -o1M -l15M -U
```

3. Erase the flash, except for the first megabyte:

```
flashctl -p/dev/fs0 -o1M -l15M -ve
```

4. Format the partition:

```
flashctl -p/dev/fs0p0 -o1M -l15M -vf
```

5. Slay, then restart the driver to mount the new partition:

```
slay devf-generic devf-generic -s0x02000000,32M
```

You should now have /fs0p1 and /fs0p3 directories automounted.

Driver Command Summary#

The driver command lines below are specific to the TI DM644x EVM 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.

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	`startup-dm6446`	.	.	`src/hardware/startup/boards/dm6446`
Serial	`devc-ser8250 -e -F -S -b115200 -c27000000/16 0x01c20000^2,0x28`	devc-ser8250	.	`src/hardware/devc/ser8250`
Network	`io-pkt-v4 -ddm644x mac=005647564757 -ptcpip`	io-pkt-v4-hc ifconfig	devn-dm644x.so devnp-shim.so libsocket.so	`src/hardware/devn/dm644x`
ETFS	`fs-etfs-dm644x -m /fs/etfs`	fs-etfs-dm644x etfsctl	.	`src/hardware/etfs/nand512/dm644x`
Flash	`devf-generic -s0x02000000,32M`	devf-generic	.	*QNX SPD 6.4.x* (Binary Only)
I2C	`i2c-dm6446`	i2c-dm6446	.	`src/hardware/i2c/dm6446`
EIDE	`devb-eide-dm644x eide noslave,master=blk cache=12M`	devb-eide-dm644x `udma=4`	.	`src/hardware/devb/eide/arm/dm644x.le`
Audio	`io-audio -d dm644x-davinci`	io-audio	deva-ctrl-dm644x-davinci.so libasound.so deva-mixer-ac97.so	`src/hardware/deva/ctrl/dm644x`
Graphics	`io-display -dvid=0,did=0`	io-display	devg-davinci.so libm.so libGLES_CL.so.1 libffb.so.2 libgf.so.1 libdisputil.so	`src/hardware/devg/davinci`
SPI	`spi-master -d dm644x base=0x01c66800,irq=43,edmairq=0xc111,edmachannel=17`	spi-master	spi-dm644x.so	`hardware/spi/dm644x`
USB	`io-usb -djacinto ioport=0x01c6400,irq=12`	io-usb	devu-jacinto.so libusbdi.so class drivers	*prebuilt only*

Startup:#

Some modules aren't enabled after boot up, so you need to use command-line options to startup to enable them. You must pass these options before any other startup options.

To enable the EIDE driver, use the -E option at startup:

```
startup-dm6446 -L 0x84000000,0x4000000 -E -v
```

Flash/NAND Flash:#

The Flash, NAND Flash, or SRAM are mapped into the same memory space and selectable via J4. Only one device can be selected at a time. For information about the hardware or firmware, refer to the documentation for the TI DM644x board.

Audio:#

This driver must be started after the I2C driver is active.

Graphics:#

To start the drive:

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

Note: Starting io-display simply starts the display server. Nothing will appear on the display until a graphical application is run (i.e. vsync or egl-gears).

Required config files:

- /etc/system/config/display.conf
- /etc/system/config/jacinto.conf

The `/etc/system/config/display.conf` configuration file describes the graphics memory interface settings. It should contain the following for a 720x480 resolution:

```
device {
   drivername=davinci
   vid=0
   did=0
   deviceindex=0
   modeopts=/etc/system/config/davinci.conf
   display {
      xres=720
      yres=480
      refresh=60
      pixel_format=rgb565
      }
   }
```

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

About graphics#

The driver supports only the following analog outputs:

- NTSC 720x480 interlaced output (composite, s-video, component)
- PAL 720x576 interlaced output (composite, s-video, component)

The VPSS controller supports 4 layers / "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 and vertical zoom (2X, 4X)

The OSD windows can support the following 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)

Note Caveats:

- Although the driver supports NTSC and PAL composite, s-video, and component outputs, due to hardware availability, we could verify only the NTSC format and composite and s-video outputs at this time.
- The OSD window restrictions (from the TI documentation) include:
  - Video Window 1 and all OSD windows must be fully contained within the Video Window.
  - 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 start X position must be offset a multiple of 16 pixels from the left edge of Video Window 0.
- There is no hardware-accelerated 2D or 3D rendering.
- Transparency of the OSD windows appears to work only with a pixel value of 0, due to an apparent hardware limitation.
- We don't support digital (LCD) output at this point, due to hardware limitations.
- The VPSS display controller on the DM644x can't display memory addresses above 128 MB (from TI Errata). If your system has more than 128MB of memory, you will need to reserve memory for the controller in the bottom 128MB. You can do this by using the -r option to startup. The -r option takes two parameters: physical address,size (in bytes) For example, to reserved 8 MB of memory for the VPSS controller, specify:

```
startup-dm6446 -r0x83800000,0x800000
```

To tell the graphics driver to use the memory, specify the following options in davinci.conf:

- vbase=address (where address matches that used with the -r option to startup)
- vsize=size (where size is in bytes and matches that used with the -r option to startup)

Known issues for this BSP#

- You can't access flash and the hard drive at the same time. The default version of startup supports the flash. To enable EIDE support, specify the -E option to startup-dm6446 in your buildfile.
- The on-board flash is AM29LV256M, which has 32 MB, but only 16 MB are addressable on the DM644x EVM. The highest address A23 is connected to Vcc, so you will see a mirror partition when creating a flash partition.
- The audio driver works only when I2C is running in low-speed mode (10 kbps). You should start I2C before starting the audio driver, using the following command: **`i2c-dm6446 --b10000`**
- If you use a surface format of GF_FORMAT_BGR888, your application may terminate with a SIGBUS signal with the Advanced Graphics TDK 2.0. This problem will be fixed in the next version of the TDK. (Ref# 54335)
- In those instances where the the ROM monitor's MAC address is different from the one you pass in when running io-net, 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-net, 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 usb drive(devu-jacinto.so) has some issues,and will be fixed for a future release . (Ref#63292)