

# Release Notes for the QNX Neutrino 6.4.0 BSP for Texas Instruments OMAP2420 SDP Trunk#

## System requirements#

### Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: Texas Instruments OMAP2420 SDP
- ROM Monitor version UBoot 1.0.0
- NAND: 64M Samsung K9K121

### Host development system#

- QNX Momentics 6.4.0
- Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
- RS-232 serial port
- NULL-modem 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:	0x80010000

The interrupt vector table can be found in the buildfile located at `src/hardware/startup/boards/omap2420sdp/build`

## Getting Started#

### Step 1: Connect your hardware#

Connect the serial cable to the serial port of the OMAP2420 SDP 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 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 using the 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

Note: The OMAP2420 SDP board from the factory doesn't have a Boot Loader or ROM Monitor installed. You'll need to initially install the U-Boot boot loader onto the board, using Texas Instruments' Code Composer Studio software, and a suitable JTAG emulator, such as the Spectrum Digital XDS510PP.

Once you've installed the bootloader, you need to configure it with the TFTP server, etc., which will provide the boot image. At this point, the OMAP2420 SDP board is ready to receive a QNX boot image, in raw format.

A raw-format image 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. Later, when the QNX boot image is combined with the native QNX IPL, the boot image will remain in raw format, and the IPL will simply scan past the raw boot header and locate the QNX boot image signature.

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

```
U-Boot 1.0.0 (Jan 30 2004 - 07:49:02)
```

```
U-Boot code: 11000000 -> 11015970 BSS: -> 11019674
```

```
DRAM Configuration:
```

```
Bank #0: 10000000 32 MB
```

```
Micron StrataFlash MT28F128J3 device initialized
```

```
Flash: 32 MB
```

```
In: serial
```

```
Out: serial
```

```
Err: serial
```

Use the **printenv** command to show the current settings for ipaddr, ethaddr, serverip, etc., use the **setenv** command to configure the following parameters:

- serverip
- gatewayip
- netmask
- bootfile
- ipaddr
- ethaddr

3. Once these parameters are configured, use the saveenv command to store your changes. Refer to the U-Boot documentation for more information: [www.sourceforge.net/projects/u-boot](http://www.sourceforge.net/projects/u-boot)

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

```
tftpboot 10010000 /xfer/ifs-omap2420sdp.raw
```

You should see output similar to the following:

```
OMAP2420 SDP# tftpboot 10010000
```

```
TFTP from server 10.0.0.5; our IP address is 10.1.0.28; sending through  
gateway 10.0.0.5
```

```
Filename '/xfer/ifs-omap2420sdp.raw'.
```

```
Load address: 0x80010000
```

```
Loading: #####
```

```
#####  
#####  
done  
Bytes transferred = 799464 (c32e8 hex)  
OMAP2420 SDP #
```

Now, to run the image, enter:

```
go 80010000
```

You should see output similar to the following:

```
CPU0: Dcache: 1024x32 WB  
CPU0: Icache: 1024x32  
CPU0: VFP 410120b2  
CPU0: 4107b362: arm1136 rev 2 300MHz  
  
System page at phys:802db000 user:fc404000 kern:fc404000  
Starting next program at vfe03c03c  
cpu_startnext: cpu0 -> fe03c03c  
Welcome to QNX Neutrino Trunk on the Texas Instruments OMAP2420SDP Board  
  
#  
#
```

4. Once you've established the U-Boot commands to download the image and boot the board, you can also configure U-Boot to save these commands; type the following at the OMAP5912 OSK prompt:

```
setenv bootcmd tftpbboot 80010000 /xfer/ifs-omap2420sdp.raw\; go 80010000  
saveenv
```

To boot the board, enter:

```
bootd
```

#### Step 4: Flashing the IPL on to the target#

##### Step 4A: Creat the IPL image#

**Note:** For further information on how to use the TI CSST tools to program the IPL/IFS images to NAND, please refer to "CSST Quick Start Guide for SDP242x" and "OSTBoot Version 1.7 Release OMAP242x Release Notes".

Run the mkflashimage script, inside the /images directory of the BSP. The output file from this script is a combined IPL/OS image called **ipl-ifs-omap2420sdp.bin**. You'll download this file to the board's memory using the bootloader, and then burn the image into the board's flash. The IPL is padded to 64K.

The following steps describes how to generate the ipl-ifs-omap2420sdp.bin:

- generates a bin format OS image called ifs-omap2420sdp.bin using the omap2420sdp.build file
- Convert IPL header into Binary format
- Convert IPL into Binary format
- Cat boot header and ipl together
- pads the binary IPL to 64K image
- Combine the IPL with the OS Image creating a file called ipl-ifs-omap2420sdp.bin

Here is the mkflashimage script:

```
#!/bin/sh
```

```
# script to build a binary IPL and boot image for the OMAP 2420 SDP board
```

```
set -v
```

```
# Convert IPL header into Binary format
```

```
 ${QNX_HOST}/usr/bin/ntoarm-objcopy --input-format=elf32-littlearm --output-format=binary ../src/hardware/ipl/bo
```

```
# Convert IPL into Binary format
```

```
 ${QNX_HOST}/usr/bin/ntoarm-objcopy --input-format=elf32-littlearm --output-format=binary ../install/armle/boot/sy
```

```
# Cat boot header and ipl together
```

```
 cat ./tmp-boot-header.bin ./tmp-ipl-omap2420sdp.bin > ./tmp-header-ipl-omap2420sdp.bin
```

```
# Pad Binary IPL to 64K image
```

```
 mkrec -s64k -ffull -r ./tmp-header-ipl-omap2420sdp.bin > ./ipl-omap2420sdp.bin
```

```
# Combine the IPL with the OS Image
```

```
 cat ./ipl-omap2420sdp.bin ./ifs-omap2420sdp.bin > ./ipl-ifs-omap2420sdp.bin
```

```
# clean up temporary files
```

```
rm -f tmp-boot-header.bin tmp-ipl-omap2420sdp.bin tmp-header-ipl-omap2420sdp.bin ipl-omap2420sdp.bin
```

```
echo "done!!!!!!!!"
```

#### **Step 4B: Flashing an IPL image using the TI OSTBOOT utility#**

The following procedure describes how to erase the first 4MB and flash an IPL image using the TI OSTBOOT utility. The TI OMAP2420 OSTBOOT utility should have been successfully installed on the host. These commands assume that the OS image, ifs-omap2420sdp.bin, is less than (4M - 64K) bytes. The working directory is C:/omap2420sdp/.

1. Connect a RS232 null modem cable between the host and the RS232 connector (J400) on the OMAP2420 SDP base board.
2. Set Switch SW1 to SW5 according to the 2nd table on page 10 of "CSST Quick Start Guide for SDP242x"
3. Start the download process from the Windows host with the TI OMAP242x OSTBOOT utility. Using CSST\_OMAP242x\_V1.7, Click "download" in the "target" pane on the left side. Select the "Erase" tab in the "SDP2420-GP" pane on the right, and erase the entire NAND flash. Select the "Download" tab; click "Add File" and select ipl-ifs- omap2420sdp.bin. Click the "Download" button to start the download process.
4. Reset the target board when CSST\_OMAP242x\_V1.2 prompts to do so.
5. After the download process finishes, connect the RS232 cable to J9 on the daughter board.
6. Set switch SW1 to SW5 according to the 3rd table on page 14 of the "CSST Quick Start Guide for SDP242x".
7. Reset the board.

#### **Step 5: Download an OS image using the IPL and sendnto#**

Due to issues with TI OSTBOOT utility, an OS image burnt to NAND flash with OSTBOOT doesn't work. Here is the procedure to download an OS image to memory using sendnto.

1. Start the QNX Momentics IDE on the host.
2. Open a terminal server window. See the IDE documentation for details.

3. Connect the RS232 null modem cable between the host and J9 on the OMAP2420sdp board.
4. Start the target, assuming an IPL has been successfully flashed to the board following the steps in section IV above.
5. When you see the following message:

QNX Neutrino Initial Program Loader for Texas Instruments OMAP2420SDP

Commands:

Press 'D' for serial download, using the 'sendnto' utility

Press 'N' to boot an OS image in NAND flash

Type **D** to start download process on the target end.

6. In the terminal server window, click the "send file" button in the upper left corner. A popup window will open. Type or browse the correct image file name in the "Filename" box and sendto as the protocol. Click OK to start the download on the host side.

or using the QNX 'sendnto' utility: **sendnto -d/dev/ser1 -b115200 ./ifs-omap2420sdp.bin** to download image

7. After the download process finishes, the target will start the OS automatically. You should see output similar to the following:

```
send image now...
download OK...
Found image      @ 0x81000000
Jumping to startup @ 0x80011A60

CPU0: Dcache: 1024x32 WB
CPU0: Icache: 1024x32
CPU0: VFP 410120b2
CPU0: 4107b362: arm1136 rev 2 300MHz

System page at phys:802db000 user:fc404000 kern:fc404000
Starting next program at vfe03c03c
cpu_startnext: cpu0 -> fe03c03c
Welcome to QNX Neutrino Trunk on the Texas Instruments OMAP2420SDP Board

#
```

Now you can test the OS, simply by executing any shell builtin command or any command residing within the OS image (e.g. ls).

## 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-omap2420sdp	.	.	src/hardware/startup/boards/omap2420sdp

Serial	devc-seromap - e -F -b115200 -c48000000/16 0x4806A000^2,72	devc-seromap	.	src/hardware/ devc/seromap
Network	io-pkt-v4 -dsmc9000 ioport=0x08000300,irq=188 -ptcpip	io-pkt-v4 ifconfig ioport=0x08000300,irq=188	devn-smc9000.so libsocket.so devnp-shim.so	src/hardware/ devn/smc9000
USB	omap2420sdp_usb io-usb -dohci ioport=0x4805e000,irq=188	io-usb usb* omap2420sdp_usb_init	devu-ohci.so libusbdi.so	<i>QNX SPD 6.4.x</i> (Binary Only)
I2C	i2c-omap59xx - p0x48070000 - i56	i2c-omap59xx	.	src/hardware/ i2c/omap59xx
NAND	fs-etfs- omap2420 - r4096	fs-etfs- omap2420 etfsctl	.	src/hardware/ etfs/nand512/ omap2420

## Known Issues for this BSP#

- In those instances where the the ROM monitor's MAC address is different from the one you pass in when running **io-net** and/or **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-net** and/or **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.
- USB Host - the usbotg port on the does not provide a sufficient vbus to power attached devices. A powered USB hub is required.
- The OS image can not be flashed to the NAND flash using TI OMAP2420 OSTBOOT utility.