Release Notes for the QNX Neutrino 6.4 BSP for Texas Instruments Mistral OMAP3530 EVM / Beagle Board#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4.0/6.4.1
- Board version: Mistral OMAP3530 EVM and OMAP3530 Beagle board
- ROM Monitor version: X-Loader 1.41-Mistral, UBoot
- Samsung Package-On-Package K5W1G1GACM: Comprises of 128 MB OneNAND flash (interface to OMAP via GPMC CS0 16-bit wide)
- 128 Mobile DDR SDRAM 166Mhz (32MX32 bit, interface to OMAP via SDRC CS0)

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:	0x80100000

The interrupt vector table can be found in the buildfile located at `src/hardware/startup/boards/omap3530/build`

Getting Started#

Step 1: Connect your hardware#

- Connect the serial cable to the first serial port UART1 of the Mistral OMAP3530 EVM board or OMAP3530 Beagle 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.

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#

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

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

Setting up the environment#

For the Mistral OMAP3530 EVM Board#

Apply power to the target board. You should see output on your terminal console, similar to the following:

```
X-Loader 1.41-Mistral
Detected Samsung MuxOneNAND1G Flash  Starting OS Bootloader...

U-Boot 1.1.4 (Oct 17 2008 - 18:12:55)

OMAP3-GP rev 2, CPU-OPP2 L3-165MHz
OMAP3EVM 1.0 Version + mPOP (Boot NAND)
DRAM:  128 MB
NAND:256 MiB

Reading data from 0x25f800 -- 100% complete.
In:    serial
Out:   serial
Err:   serial
Reseting CHIP... Done
LAN9x18 (0x01150002) detected.
Setting mac address: aa:bb:cc:dd:ee:ff
start Auto negotiation... (take ~2sec)
Auto negotiation complete, 100BaseTX, full duplex
Hit any key to stop autoboot:  0
```

Use the **`printenv`** command to show the current settings for **`ipaddr, ethaddr, serverip`** etc.

TFTP download#

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.

Next, use the `setenv` command to configure the following parameters:

serverip
gatewayip
netmask
bootfile
ipaddr
ethaddr

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

After U-boot is configured, boot the `ifs-omap3530-mistral.bin` image as follows (we'll assume it's in a directory called /xfer/) from the OMAP3EVM # prompt: **`tftpboot 0x80100000 /xfer/ifs-omap3530-mistral.bin`**

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:

```
Reseting CHIP... Done
LAN9x18 (0x01150002) detected.
Setting mac address: aa:bb:cc:dd:ee:ff
start Auto negotiation... (take ~2sec)
Auto negotiation complete, 100BaseTX, full duplex
TFTP from server 10.42.97.136; our IP address is 10.42.104.42
Filename '/root/ifs-omap3530-mistral.bin'.
Load address: 0x80100000
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         ######
done
Bytes transferred = 2355312 (23f070 hex)
OMAP3EVM #
```

Now, to run the image, enter:

`go 0x80100000`

You should see output similar to the following, with the QNX Neutrino welcome message on your terminal screen:

```
## Starting application at 0x80100000 ...
CPU0: L1 Icache: 256x64
CPU0: L1 Dcache: 256x64 WB
CPU0: L2 Dcache: 4096x64 WB
CPU0: VFP 410330c1
CPU0: 411fc082: Cortex A8 rev 2 500MHz

System page at phys:80010000 user:fc404000 kern:fc404000
Starting next program at vfe047df0
cpu_startnext: cpu0 -> fe047df0
VFPv3: fpsid=410330c1
coproc_attach(10): replacing fe0699e0 with fe0691a0
coproc_attach(11): replacing fe0699e0 with fe0691a0
Welcome to QNX Neutrino trunk on the Texas Instruments OMAP3530 (ARMv7 Cortex-A8 core) EVM Board
Starting on-board ethernet with TCP/IP stack...
#
```

You can now 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.

Flashing the IPL on to the target#

Step A: Creat the IPL image#

> **Note**: For further information on how to use the combination of UART ~Downloader tool and U-Boot to program the IPL/IFS images to OneNAND, please refer to "OMAP35x EVM Getting Started Guide V0.9.7".

Run the mkflashimage script, inside the /images directory of the BSP. The output file from this script is a binary IPL image called **`ipl-omap3530evm.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-omap3530evm.bin:

- generates a bin format OS image called ifs-omap3530.bin using the omap3530.build file
- Convert IPL into Binary format
- pads the binary IPL to 64K image

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

# Pad Binary IPL to 64K image
   mkrec -s64k -ffull -r ./tmp-ipl-omap3530evm.bin > ./ipl-omap3530evm.bin

# clean up temporary files
rm -f tmp-ipl-omap3530evm.bin

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

Step B: Flashing an IPL image using the combination of UART Downloader utility and U-Boot#

> OneNAND on the OMAP35xEVM board is default partitioned in the following manner:
>
> *0x0000000 - 0x0080000 X-Loader
> *0x0080000 - 0x0260000 U-boot Image
> *0x0260000 - 0x0280000 U-Boot Env Data (X-loader doesn't care)
> *0x0280000 - 0x0780000 Kernel Image
> *0x0780000 - 0x8000000 depends on application

The following procedure describes how to erase the U-Boot image and flash an IPL image using the combination of UART Downloader utility and U-Boot bootloader. The UART Downloader(DownloadUtil) program should be present under OMAP35x_SDK_0.9.7/board_utilities/windows_host/Utilities on the host. The working directory is C:/omap3530evm/.

1. Turn OFF the EVM by unplugging the power adapter, and close any open terminal programs. Connect a RS232 null modem cable between the host and the RS232 connector (UART3 Debug/Boot) on the OMAP35x EVM base board.

2. Ensure SW4 is set as described in Section 2.1 “Main Board SW4” of "OMAP35x EVM Getting Started Guide V0.9.7".

3. Start the download process from the Windows host with the DownloadUtil program, using the following steps. Select the correct COM port, press the “open” button and select the following file **`u-boot.bin`**. Next press the Download button, Press OK on the dialog that says “Press OK and reset the target”. Power ON the EVM. If the download does not immediately start, Press S2 button on the EVM. The download utility should show progress and eventually will show “Main Image has been sent”. Close Download Utility and restart your terminal program.

> **Note**: Do not power cycle the EVM at this stage.

Move the serial cable to UART1 (Please ensure Jumper J8 is set appropriately) and press enter to get a U-boot prompt.

4. Now that U-boot is running on the target, you can use it to update any or all of the software in flash. The steps provided here will use the Ethernet connection with the TFTP protocol to transfer the files. Please ensure that a TFTP server using a Windows host is already setup, with its root directory same as our working directory.

> **Note**: The SDK comes with the PumpKIN TFTP server. Alternatively, tftpd32 can also be used.

5. Ensure that the terminal program is configured with 100 ms of delay after sending each line as described in Section 3.1 “Setup Terminal Program” of "OMAP35x EVM Getting Started Guide V0.9.7".

6. Ensure the environment is set as appropriate for your EVM and network setup. You may use the `setenv` command to configure the following parameters:

ethaddr (see sticker on EVM)
ipaddr (as appropriate)
serverip (PumpKIN PC’s addr)
netmask (as appropriate)
gatewayip (if needed)

Check the IP settings using `ping $(serverip)` expecting “x.x.x.x is alive”

7. Use the terminal program to send the text file reflash.txt file to the EVM

- a. Make sure you know if your board has Samsung or Micron memory parts (See Section 1.2 “Identifying Board Variations”)
- b. path: OMAP35x_SDK_0.9.7/board_utilities/scripts/reflash-samsung.txt (for Samsung memory boards)
- c. For HyperTerminal use Transfer / Send Text File …
- d. For TeraTerm use File / Send file …
- e. For minicom use the paste file command (Ctrl-A Y)

8. Run the following command sequence to replace U-Boot with QNX IPL in the OneNAND flash

```
run rf_unlock
run rf_blank_ram
run rf_er_uboot
tftpboot 0x81600000 ipl-omap3530evm.bin
run rf_wr_uboot
```

9. Run the following command sequence to place the QNX image in the OneNAND flash

```
run rf_unlock
run rf_blank_ram
run rf_er_fs
tftpboot 0x81600000 ifs-omap3530.bin
```

run rf_wr_fs

10. Reset the board and it should boot into the X-Loader which then loads the QNX IPL. The QNX IPL offers two options to the user wherein we can either transfer the QNX image over serial, or boot the QNX image resident in OneNAND flash, burnt as per the procedure above.

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

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 UART1 on the OMAP35xEVM board.

4. Start the target, assuming an IPL has been successfully flashed to the board following the steps in the section above.

5. When you see the following message:

```
The data ''

Texas Instruments X-Loader 1.41
Detected Samsung MuxOneNAND1G Flash  Starting OS Bootloader...0x1B

QNX Neutrino Initial Program Loader for Texas Instruments OMAP35xEVM
Commands:
Press 'D' for serial download, using the 'sendnto' utility
Press 'N' to boot an OS image in ~OneNAND flash

'' is not legal for a JDOM character content: 0x1b is not a legal XML character.
```

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 proctocol. Click OK to start the download on the host side.

or using the QNX 'sendnto' utility: **`sendnto -d/dev/ser1 -b115200 ./ifs-omap3530-mistral.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: L1 Icache: 256x64
CPU0: L1 Dcache: 256x64 WB
CPU0: L2 Dcache: 4096x64 WB
CPU0: VFP 410330c1
CPU0: 411fc082: Cortex A8 rev 2 500MHz

System page at phys:80391000 user:fc404000 kern:fc404000
Starting next program at vfe03cdf0
cpu_startnext: cpu0 -> fe03cdf0
```

```
coproc_attach(10): replacing fe05e044 with fe05d928
coproc_attach(11): replacing fe05e044 with fe05d928
Welcome to QNX Neutrino 6.4 on the Texas Instruments OMAP3530 (ARMv7 Cortex-A8 core) EVM Board
#
```

For OMAP3530 Beagle Board#

Apply power to the target board. You should see output on your terminal console, similar to the following:

```
Texas Instruments X-Loader 1.41
Starting OS Bootloader...



U-Boot 1.3.3 (Jul 10 2008 - 16:33:09)

OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
OMAP3 Beagle Board + LPDDR/NAND
DRAM:  128 MB
NAND:  256 MiB
In:    serial
Out:   serial
Err:   serial
Audio Tone on Speakers  ... complete
OMAP3 beagleboard.org # printenv
filesize=AF13C
bootargs=console=ttyS2,115200n8 ramdisk_size=8192 root=/dev/ram0 rw rootfstype=ext2 initrd=0x81600000,8M nohz=0ff
stdin=serial
stdout=serial
stderr=serial

Environment size: 170/131068 bytes
OMAP3 beagleboard.org #
```

SD card download#

```
OMAP3 beagleboard.org # mmcinit; fatload mmc 0 0x80100000 ifs-omap3530-beagle.bin; go 80100000
```

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:

```
reading ifs-omap3530-beagle.bin

1291448 bytes read
## Starting application at 0x80100000 ...
CPU0: L1 Icache: 256x64
CPU0: L1 Dcache: 256x64 WB
CPU0: L2 Dcache: 4096x64 WB
CPU0: VFP 410330c1
CPU0: 411fc083: Cortex A8 rev 3 500MHz

System page at phys:80010000 user:fc404000 kern:fc404000
Starting next program at vfe04427c
cpu_startnext: cpu0 -> fe04427c
VFPv3: fpsid=410330c1
coproc_attach(10): replacing fe071518 with fe070d6c
coproc_attach(11): replacing fe071518 with fe070d6c
Welcome to QNX Neutrino trunk on the Texas Instruments OMAP3530 (ARMv7 Cortex-A8 core) - Beagle 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).

Creating a new flash partition#

- Enter the following command to start the flash filesystem driver: `fs-etfs-omap3530 -r65536 -m /fs/etfs`
- Stop the filesystem on the device: `etfsctl -d /dev/etfs2 -s`
- Erase the etfs2 partition: `etfsctl -d /dev/etfs2 -e`
- Format the filesystem: `etfsctl -d /dev/etfs2 -f`
- Make the filesystem continue: `etfsctl -d /dev/etfs2 -c`

You should now have a /fs/etfs directory which you can copy files to.

Summary of driver commands#

The following tables summarize the commands to launch the various drivers.

For the Mistral OMAP3530 EVM board #

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	`startup-omap3530`	.	.	`src/hardware/startup/boards/omap3530`
Serial	`devc-seromap -e -F -b115200 -c48000000/16 0x4806A000^2,72`	devc-seromap	.	`src/hardware/devc/seromap`
Nand Flash	`fs-etfs-omap3530 -r65536 -m /fs/etfs`	fs-etfs-omap3530 etfsctl	.	`src/hardware/etfs/nand2048/omap3530`
Network	`io-pkt-v4 -dsmc9118 ioport=0x2C000000,irq=176,mac=001122334455 -ptcpip`	io-pkt-v4 ifconfig	devn-smc9118.so libsocket.so devnp-shim.so	`src/hardware/devn/smc9118`
I2C	`i2c-omap35xx For interface 1: i2c-omap35xx -i56 -p0x48070000 For interface 2: i2c-omap35xx -i57 -p0x48072000`	i2c-omap35xx	.	`src/hardware/i2c/omap35xx`
SPI	`spi-master -d omap3530 For interface 1: spi-master`	spi-master	spi-mpc8536.so	`src/hardware/spi/omap3530`

	-d omap3530 base=0x48098000,irq=65 For interface 2: spi-master -d omap3530 base=0x4809A000,irq=66			
USB OTG Host	io-usb -domap3530-mg ioport=0x480ab000,irq=92	io-usb usb*	libusbdi.so devu-ohci.so devu-ehci.so devu-uhci.so devu-omap3530-mg.so	*prebuilt only*
SD card	devb-mmcsd-omap3 cam quiet blk cache=2m,automount=hd0t11:/fs/sd0 mmcnetwork noac12 dos exe=all	devb-mmcsd	libcam.so fs-dos.so cam-disk.so	src/hardware/devb/mmcsd
Audio	io-audio -domap35xx-twl4030	io-audio wave waverec mix_ctl	deva-ctrl-omap35xx-twl4030.so libasound.so.2	src/hardware/deva/ctrl/omap35xx
Graphics	io-display -dvid=0,did=0	omap35xx-evm.conf io-didplay egl-gears-lite vsync	libGLES_CL.so libffb.so libgf.so devg-omap35xx.so devg-soft3d-fixed.so libdisputil.so	src/hardware/devg/omap35xx

For the OMAP3530 Beagle board #

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	startup-omap3530 -L 0x87E00000,0x200000 -v -p -D8250.0x49020000^2.0.48000000.16	.	.	src/hardware/startup/boards/omap3530
Serial	devc-seromap -e -F -b115200 -c48000000/16 0x49020000^2,74	devc-seromap	.	src/hardware/devc/seromap
Nand Flash	fs-etfs-omap3530_micron -r65536 -m/fs/etfs	fs-etfs-omap3530_micron etfsctl	.	src/hardware/etfs/nand2048/omap3530_micron
I2C	i2c-omap35xx For interface 1: i2c-omap35xx -i56 -p0x48070000 For interface 2: i2c-	i2c-omap35xx	.	src/hardware/i2c/omap35xx

	omap35xx -i57 -p0x48072000			
SPI	spi-master -d omap3530 For interface 1: spi-master -d omap3530 base=0x48098000,irq=65 For interface 2: spi-master -d omap3530 base=0x4809A000,irq=66	spi-master	spi-mpc8536.so	src/hardware/spi/omap3530
USB OTG Host	io-usb -domap3530-mg ioport=0x480ab000,irq=92	io-usb usb*	libusbdi.so devu-ohci.so devu-ehci.so devu-uhci.so devu-omap3530-mg.so	*prebuilt only*
USB EHCI Host(For Rev C board)	io-usb -dehci-omap3 ioport=0x48064800,irq=77,verbose=5	io-usb usb*	libusbdi.so devu-ehci-omap3.so	*prebuilt only*
SD card	devb-mmcsd-omap3 cam quiet blk cache=2m,automount=hd0t11:/fs/sd0 mmcsd noac12 dos exe=all	devb-mmcsd	libcam.so fs-dos.so cam-disk.so	src/hardware/devb/mmcsd
Audio	io-audio -domap35xx-twl4030	io-audio wave waverec mix_ctl	deva-ctrl-omap35xx-twl4030.so libasound.so.2	src/hardware/deva/ctrl/omap35xx
Graphics	io-display -dvid=0,did=0	omap35xx.conf io-didplay egl-gears-lite vsync	libGLES_CL.so libffb.so libgf.so devg-omap35xx.so devg-soft3d-fixed.so libdisputil.so	src/hardware/devg/omap35xx

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.

Audio#

Command:

i2c-omap35xx
io-audio -d omap35xevm
wave testfile.wav

Note: To start the audio system, you'll need to execute the Audio driver and commands in the order indicated.

Graphics#

Note: The config file **omap35xx.conf** is for Beagle board, omap35xx-evm.conf for Mistral board.

NAND Flash driver#

For OMAP3530 Mistral board: OneNAND

Note: The NAND ETFS driver supports only the 128 Mbytes Samsung OneNAND currently. Reserve the first 64M for bootloader, and now as an ETFS filesystem is no longer mounted by default; you can use the `-m` option or `mount -tetfs /dev/etfs2 my_mountpoint`. (Ref# 41841, 57498; Ticket ID 74912)

For OMAP3530 Beagle Board: Micron 2Gb NAND (256MB)

Command:

```
fs-etfs-omap3530_micron -r65536  -m/fs/etfs
```

Note: Provide proper "-r" value for reserveing the bootloader area on your board.

-r kbytes: Set size of raw partition /dev/etfs1. Default: 0.

USB OTG Host Controller driver#

Note: For OMAP3530 Beagle Board:

1. Connect to the micro usb adapter as we do not support the larger one yet.

2. Press the S1 button until the beagle logo is displayed

3.Start io-usb with the stock options from the bsp build file.

USB EHCI Host Controller driver#

Note: Only for the OMAP3530 Beagle Board (Rev C).

To run USB OTG Host and EHCI Host both:

Command:

```
io-usb -domap3530-mg ioport=0x480ab000,irq=92 -dehci-omap3 ioport=0x48064800,irq=77,verbose=5
```

Known issues for this BSP#

- The NAND ETFS driver supports only the 128 Mbytes Samsung OneNAND currently.
- The graphics driver `devg-omap35xx.so` does not link against the graphics libraries when built in the IDE. **Workaround:** Modify the QNX C/C++ Project Properties of the graphics driver: in the Linker tab, under the Extra libraries category, add two entries **ffb** and **disputilS** as type Dynamic. Save and rebuild the graphics driver.
- Temporary override: Use an updated version of `armv_chip_a8.c` file which has a FIXME that disables the branch target cache, as this seems to induce application crashes.

- USB Host - the USB-OTG port on the EVM does not provide a sufficient vbus to power attached devices. A powered USB hub is preferred.
- The latest IDE (4.6.0) can only import one build file. But this has two *.build files - one for beagle and one for mistral. When this BSP is imported into IDE, only the Mistral board image is produced. **Workaround:** Currently you can only import second build file manually, I.e. create another System Builder project, from existing build file, then modify search path for this project to include install directory of the bsp.