

# Release Notes for the QNX Neutrino 6.5.0 BSP for OMAP3530 Beagle Board#

## System requirements#

### Target system#

- QNX Neutrino RTOS 6.5.0
- Board version: OMAP3530 Beagle board Rev B, rev C3 and rev C4
- ROM Monitor version: Texas Instruments X-Loader 1.4.2, U-Boot 2009.01
- Micron Package-On-Package MT29C2G48MAKLCJI-6 IT: Comprises of 2Gb NAND x 16 (256MB) and 2Gb MDDR SDRAM x32 (256MB @ 166MHz)

### Host development system#

- QNX Momentics 6.5.0
- 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/beagle.build`

## Getting Started#

### Step 1: Connect your hardware#

- Connect the serial cable to the serial port on OMAP3530 Beagle Board 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 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 1.4.2 (Feb 19 2009 - 12:01:24)
```

```
Reading boot sector
```

```
Loading u-boot.bin from nand
```

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

```
U-Boot 2009.01-dirty (Feb 19 2009 - 12:22:31)
```

```
I2C: ready
```

```
OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
```

```
OMAP3 Beagle board + LPDDR/NAND
```

```
DRAM: 256 MB
```

```
NAND: 256 MiB
```

```
MUSB: using high speed
```

```
In: serial usbtty
```

```
Out: serial usbtty
```

```
Err: serial usbtty
```

```
Board revision C
```

```
Serial #7842000300000000040323090b017014
```

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

```
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:

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

```
6198048 bytes read
```

```
## Starting application at 0x80100000 ...
```

```
Beagle Board Revision: c3
```

```
CPU0: L1 Icache: 256x64
```

```
CPU0: L1 Dcache: 256x64 WB
```

```
CPU0: L2 Dcache: 4096x64 WB
```

```
CPU0: VFP-d32 FPSID=410330c1
```

```
CPU0: NEON MVFR0=11110222 MVFR1=00011111
```

```
CPU0: 411fc083: Cortex A8 rev 3 600MHz
```

```
Loading IFS...decompressing...done
```

```
System page at phys:80010000 user:fc404000 kern:fc404000
```

```
Starting next program at vfe04e604
```

```
cpu_startnext: cpu0 -> fe04e604
```

```
VFPv3: fpsid=410330c1
```

```
coproc_attach(10): replacing fe07e01c with fe07d8bc
```

```
coproc_attach(11): replacing fe07e01c with fe07d8bc
```

```
Welcome to QNX Neutrino 6.5.0 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).

## Flashing the IPL on to the target#

### Step A: Create the IPL image#

Run the mkflashimage script, inside the /images directory of the BSP. The output file from this script is a binary IPL image called **ipl-omap3530beagle.bin** and **nand-ipl-omap3530beagle.bin**.

First, you'll download **ipl-omap3530beagle.bin** file to the board's memory using the pserial tools, and then you can burn the ifs image **ifs-omap3530-beagle.bin** and NAND IPL image **nand-ipl-omap3530beagle.bin** into the board's NAND flash. The IPL is padded to 24K.

The following steps describes how to generate the ipl-omap3530evm.bin and nand-ipl-omap3530beagle.bin:

- generates a bin format OS image called ifs-omap3530-beagle.bin using the beagle.build file
- Convert IPL NAND boot header into Binary format
- Convert IPL into Binary format
- Cat NAND boot header and ipl together
- Pad Binary IPL to 24K image, this is the image used by boot from UART
- Pad Binary IPL with Header to 24K image, this is the image to put on NAND and boot from NAND

Here is the mkflashimage script:

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

```
# script to build a binary IPL and boot image for the OMAP3530 Beagle board
```

```
set -v
```

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

```
${QNX_HOST}/usr/bin/ntoarmv7-objcopy --input-format=elf32-littlearm --output-format=binary ../src/hardware/ipl/bo
```

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

```
${QNX_HOST}/usr/bin/ntoarmv7-objcopy --input-format=elf32-littlearm --output-format=binary ../src/hardware/ipl/bo
```

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

```
cat ./tmp-boot-header.bin ./tmp-ipl-omap3530beagle.bin > ./tmp-header-ipl-omap3530beagle.bin
```

```
# Pad Binary IPL to 24K image, this is the image used by boot from UART
```

```
mkrec -s24k -ffull -r ./tmp-ipl-omap3530beagle.bin > ./ipl-omap3530beagle.bin
```

```
# Pad Binary IPL with Header to 24K image, this is the image to put on NAND and boot from NAND
```

```
mkrec -s24k -ffull -r ./tmp-header-ipl-omap3530beagle.bin > ./nand-ipl-omap3530beagle.bin
```

```
# clean up temporary files
```

```
rm -f tmp*.bin
```

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

## Step B: Booting an IPL image#

Refer to the [AM/OMAP Boot Resource Pages](#) for options for installing the IPL to flash.

## 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 OMAP3530 Beagle board #

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	startup-omap3530 -L 0x87E00000,0x200000 -v - D8250.0x49020000^2.0.48000000.16 beagle	.	.	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	i2c-omap35xx	.	src/hardware/i2c/omap35xx
USB OTG Host	io-usb -domap3530-mg ioport=0x480ab000,irq=92	io-usb usb*	libusbdi.so devu-omap3530-mg.so	<i>prebuilt only</i>
USB EHCI Host(For Rev C board)	io-usb -dehci-omap3 ioport=0x48064800,irq=77	io-usb usb*	libusbdi.so devu-ehci-omap3.so	<i>prebuilt only</i>
SD card	devb-mmcsd-beagle cam quiet blk cache=2m mmcsd noacl2	devb-mmcsd-beagle	libcam.so fs-dos.so cam-disk.so	src/hardware/devb/mmcsd

Audio	io-audio - domap35xx- twl4030	resource_seed 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 display.conf io-display vsync egl-gears-lite vsync	libGLES_CL.so libGLES_CM.so libffb.so libgf.so devg-omap35xx.so devg-soft3d-fixed.so libFF-T2K.so.3 libm.so.2 devg-soft3d.so	src/hardware/ devg/omap35xx
WIFI	io-pkt-v4 -d /proc/ boot/devnp- mv8686.so dir=/root/io- pkt,verbose=1,p -p tcpip mclbytes=4096,stacksize=65000 random	io-pkt-v4 helper_sd.bin sd8686.bin wpa_supplicant wpa_cli wpa_supplicant.conf	libcrypto.so.1 libssl.so.1 libsdio.so.1 libsocket.so devnp-mv8686.so	<i>prebuilt only</i>
SGX Graphics Accelerator driver	pvrsvrd	pvrsvrd graphics.conf gles1-egl-gears	libsrv_um.so libglslcompiler.so libIMGegl.so libImgGLESv1_CM.so libImgGLESv2.so libImgOpenVG.so libpvr2d.so libsrv_um.so libWFDdevg.so pvrsvrinit.so wsegl-gf.so libEGLdevg.so libWFDdevg.so libiow.so.1 libGLESv1_CM.so.1 libGLESv1_CL.so.1 libEGL.so.1 libGLESv1_CLdevg.so libGLESv1_CMdevg.so	<i>prebuilt only</i>

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.

## Startup #

Command:

```
startup-omap3530 -L 0x87E00000,0x200000 -v -D8250.0x49020000^2.0.48000000.16 beagle
```

## **SD card #**

Command:

```
resource_seed  
i2c-omap35xx  
pmic_tw4030_cfg  
devb-mmcsd-beagle cam quiet blk cache=2m
```

**Note:** To start the SD driver, you'll need to execute `resource_seed`, the I2C driver and `pmic_tw4030_cfg` utility first

## **Audio#**

Command:

```
resource_seed  
i2c-omap35xx  
pmic_tw4030_cfg  
io-audio -domap35xx-twl4030  
wave testfile.wav
```

**Note:** To start the audio driver, you'll need to execute `resource_seed`, the I2C driver and `pmic_tw4030_cfg` utility first

## **Graphics#**

Command:

```
i2c-omap35xx  
pmic_tw4030_cfg  
io-display -dvid=0,did=0
```

**Note:** To start the graphics driver, you'll need to execute the I2C driver and `pmic_tw4030_cfg` utility first

## **USB OTG Host Controller driver#**

Command:

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

**Note:**

1. Min A to min B usb cable is required.
2. Connect to the micro usb adapter as we do not support the larger one yet.
3. Press and hold the S1 (User) button at power up or reset until board starts booting (~1 sec).
4. May need to power cycle board to get S1 (User) button working depending on how/if the power management chip has been reconfigured.
5. Start `io-usb` with the stock options from the bsp build file.

## **SGX Graphics Accelerator driver#**

Command:

In build file:

```
#GRAPHICS_ROOT=/usr/lib/graphics/omap3530
#/usr/lib/graphics/omap3530/pvrsvd
#gles1-egl-gears &
```

In command line:

```
#export GRAPHICS_ROOT=/usr/lib/graphics/omap3530
#cd /usr/lib/graphics/omap3530/
#pvrsvd
## validate application
#gles1-egl-gears &
```

### Note:

1. io-display have to run first
2. The Composition\_Manager patch-650-2258-[CompMgr.tar](#) need be installed first.
  - You can get the composition manager patch : patch-650-2258-[CompMgr.tar](#) [here](#).
  - To install the patch-650-2258-[CompMgr.tar](#) in the Beagle board BSP's prebuilt directory:

```
# cd beagle_workdir
# tar -xvf patch-650-2258-CompMgr.tar
# cd patches/650-2258/target/qnx6
# cp -r armle-v7 ../../../../prebuilt/
# cd beagle_workdir
# make clean all
```

### WiFi driver #

Command:

```
## WIFI start ram filesystem
devb-ram ram capacity=16384,nodinit,cache=0m
waitfor /dev/hd0
fdisk /dev/hd0 add -t 6
mount -e /dev/hd0
mkdosfs /dev/hd0t6
mount -t dos /dev/hd0t6 /fsram

## SDIO WIFI driver
## Note: please insert SDIO WIFI card. and then run ". /root/wifi.sh"
```

/root/wifi.sh

```
io-pkt-v4 -d /proc/boot/devnp-mv8686.so dir=/root/io-pkt,verbose=1,poll=0 -p tcpip mclbytes=4096,stacksize=65000 random
ifconfig mv0 up
wpa_supplicant -Dwext -imv0 -C /fsram/tmpfs &
wpa_cli -i mv0 -p /fsram/tmpfs
```

### Note:

1. Because SD and SDIO wifi cards using the same slot, please slay the SD driver first before running the SDIO wifi driver.

2. Please refer to `WIFI_HOW_TO_RUN.txt` which is included in the BSP to setup the wifi connection in `wpa_cli` command line.

3. Please refer to [QNX document:WiFi Configuration Using WPA and WEP](#). Edit `/etc/wpa_supplicant.conf` file in build file to configure the wifi connection.

### **Known issues for this BSP#**

- 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.
- SGX can lock up the board when multiple accelerated applications are run concurrently, run one at a time.
- USB-OTG: Some SanDisk Cruzer devices do not operate reliably on the OTG interface (ref #76047).
- USB-OTG: A D-Link DUB E-100 USB to Ethernet adapter did not operate reliably on the OTG interface (ref #76107). Use the EHCI interface when possible.
- Audio: The right/left channels swapped (ref #10937)