Release Notes for the QNX Neutrino 6.4.0 BSP for TI OMAP-L137 EVM#

1.System Requirements#

Target Requirements

1. QNX Neutrino RTOS 6.4.0
2. Board version: TI OMAP-L137 EVM
3. L137 processor
4. 64MB DDR SDRAM
5. 4 MB SPI flash

Host Requirements

1. QNX Momentics 6.4.0
2. Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
3. RS-232 serial port and serial cable, or a USB-to-serial cable
4. Ethernet link

2.Getting Started#

2.1 Building 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.

2.2 Connect your Hardware#

Connect the serial cable to the serial port of the OMAP L137 EVM board to the first serial port of your host machine. There is 1 serial port on OMAP L137 EVM board. Use the one which is near the boundary of the board. Usually you should see some Uboot output on the console when you connect cable to the correct port. 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.

The correct terminal settings of the program handling serial connection should be:

baudrate	115200
data	8 bit
parity	none
stop	1bit
flow control	none

2.3 Setup you environment#

1. Power on your target. You should see the u-boot output on your console. 2. Conenct an ethernet cable to any of the 2 PHY port available on the back side of the board.

3. Boot the IFS image#

You can use TFTP download (the default) or serial download to transfer an OS image to the board, as described below.

3.1 Boot via tftp#

This method requires that you put the raw image generated by BSP (by default at $BSP_ROOT/images/ifs-omapl137.raw) to a TFTP server. This server must be reachable via board and preferably should be on the same LAN. As soon as u-boot starts, press any key so that u-boot stops and doesnt boot the prebuild linux kernel. Configure u-boot parameters as follows:

```
=> setenv ipaddr 10.90.74.214
=> setenv serverip 10.90.74.42
=> setenv netmask 255.255.255.0
=> setenv gatewayip 10.90.74.1
=> setenv bootfile ifs-omapl137.raw
=> setenv loadaddr 0xC0008000
=> setenv bootcmd 'tftpboot $loadaddr $bootfile; go $loadaddr'
=> setenv bootdelay 2
=> saveenv
Saving Environment to SPI Flash...
Erasing SPI flash...Writing to SPI flash...done
=> boot
```

Note: You should set the ipaddr, serverip, netmask, gatewayip accordingly to your network.

At this point, you should see output similar to this when it finishes downloading:

```
## Starting application at 0xC0008000 ...
Welcome to QNX Neutrino 6.4.0 on the TI OMAPL137EVM Board
#
```

Congratulations! QNX 6.4.0 kernel is running on your system. 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. Still the network and flash drivers need to be supported for the TI OMAP L137 EVM board.

4. Driver Command Summary#

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	startup-omapl137 -wt -L 0xC3E00000,0x200000 -vvv	startup-omapl137	libstartup.a	src/hardware/startup/boards/omapl137
Serial	devc-ser8250 -e -F -S -b115200 -c150000000/16 0x01d0d000^2,61 waitfor /dev/ser1 4 reopen /dev/ser1	devc-ser8250	none	src/hardware/devc

I2C	i2c-dm6446 -a0x1 -i15 -p0x01c22000 -f24000000 -l0x1000 --b10000 --u0 (for controller 1) i2c-dm6446 -a0x2 -i51 -p0x01e28000 -f24000000 -l0x1000 --b10000 --u1 (for controller 2)	i2c-dm6446	libi2c-master.a	src/hardware/i2c/dm6446
USB	io-usb -dohci-omapl1xx ioport=0x01E25000,irq=59 -domapl1xx-mg ioport=0x01E00400,irq=58	devu-ohci-omapl1xx.so devu-omapl1xx-mg.so	libusbdi.so io-blk.so io-usb usb devb-umass libcam.so fs-dos.so fs-qnx4.so fs-ext2.so cam-disk.so	"prebuiltonly"
Network	io-pkt-v4 -ddm644x-omapl1xx phy=2 -ptcpip	io-pkt-v4 ifconfig	devn-dm644x-omapl1xx.so	/src/hardware/devn
SPI	spi-master -d dm644x-omapl1xx base=0x01c41000,irq=20,clock=150000000,edma=0,en0def=1,spi spi-master -u1 -d dm644x-omapl1xx base=0x01E12000,irq=56,clock=150000000,edma=0,en0def=1,spi	spi-master	spi-dm644x-omapl1xx.so libdma-omapl1xx.so libdma-omapl1xxS.a	/src/hardware/spi/dm644x cntrlr=omapl1xx cntrlr=omapl1xx
RTC	rtc -s -v omapl1xx rtc -v omapl1xx	rtc date	libutil.a	src/utils/r/rtc
DMA			libdma-omapl1xx.so libdma-omapl1xxS.a	src/lib/dma/omapl1xx
DSPLink			libdsplink160-clientS.a	"prebuiltonly"
MMCSD	devb-mmcsd-omapl1xx cam mmcsd bs="wp_gpio_bank=2 wp_gpio=1 ins_gpio_bank=2 ins_gpio=2 ins_gpio_irq=44" verbose=3 blk cache=2M	devb-mmcsd-omapl1xx	libcam.so cam-disk.so io-blk.so fs-dos.so	src/hardware/devb/mmcsd

Serial#

To start the Serial driver, run the following command:

devc-ser8250 -e -F -S -b115200 -c150000000/16 0x01d0d000^2,61

Network#

To start network driver, run:

```
io-pkt-v4 -ddm644x-omapl1xx phy=2 -ptcpip
```

Note : Here the phy=2 indicates which ethernet port on the OMAPL137 board you are connecting to, the ethernet port next to the serial interface on the board corresponds to phy = 2, and the next to that one corresponds to phy = 1.

you should see following output when you run ifconfig

```
# ifconfig en0
en0: flags=80008802<BROADCAST,SIMPLEX,MULTICAST,SHIM> mtu 1500
        address: 00:0e:99:02:f9:4e
        media: Ethernet 100baseTX
        status: active
```

To bring the network interface up, type following command

```
ifconfig en0 10.90.74.246/24 up
```

Here 10.90.74.246 is the IP address assigned to your target and 24 is the netmask bits (i.e. 255.255.255.0)
The LEDs of the RJ45 port corresponding to the enabled interface shall glow periodically. This can be used to identify the correct port, port 1 (phy 1) or port 2 (phy 2).

RTC#

To set the RTC time as that of the System time, run:

```
rtc -s -v omapl1xx
```

To set the System time as that of the RTC time, run:

```
rtc -v omapl1xx
```

SPI#

To start spi driver for spi0 interface, run:

```
spi-master -d dm644x-omapl1xx
base=0x01c41000,irq=20,clock=150000000,edma=0,en0def=1,spicntrlr=omapl1xx
```

you can see a file /dev/spi0 after the above step.

To start spi driver for spi1 interface, run:

```
spi-master -u1 -d dm644x-omapl1xx
base=0x01E12000,irq=56,clock=150000000,edma=0,en0def=1,spicntrlr=omapl1xx
```

you can see a file /dev/spi1 after the above step.

For reading the on-board spi-flash you need to use /dev/spi0.

USB#

To start both USB OHCI (1.1) and USB 2.0 drivers, run:

```
io-usb -dohci-omapl1xx ioport=0x01E25000,irq=59 -domapl1xx-mg
ioport=0x01E00400,irq=58
```

wait for /dev/io-usb/io-usb

To list and mount mass storage device, run:

`devb-umass&` (This will exit if it doesn't find any mass storage device, so start this only after plugging in a device)

```
mount -t <file_system_name> /dev/<device_name> <path to be mounted at>
```

MMCSD#

To start mmcsd driver, run

devb-mmcsd-omapl1xx cam mmcsd bs="wp_gpio_bank=2 wp_gpio=1 ins_gpio_bank=2 ins_gpio=2 ins_gpio_irq=44" verbose=3 blk cache=2M

After the the card is inserted you can see /dev/hdX

5. Burning the QNX IFS image to the SPI flash#

If the Board has U-Boot, the commands posted below are to be followed, to burn the QNX IFS image to the SPI flash.

The SPI flash on the OMAPL137 board is of size 4MB. Hence the IFS image should be less than 4MB, in order to burn it to the SPI flash.

The IFS image built from the BSP package from the Foundry is an Uncompressed QNX IFS image (which is more than 4MB). To compress it, add the compress flag in the file src/hardware/startup/boards/omapl137/build as follows

```
[image=0xc0008000]
[+compress]
[virtual=armle,raw] .bootstrap = {
```

and rebuild the QNX IFS image.

Now you get a Compressed QNX IFS image which is less than 4MB. To burn this Compressed QNX IFS image to the SPI flash, the following commands should be used, after the U-Boot boots up:

```
tftp 0xc0008000 ifs-omapl137.raw

sf probe 0

sf erase 0x80000 0x200000

sf write 0xc0008000 0x80000 size_of_the_qnx_ifs

setenv bootcmd 'sf probe 0; sf read 0xc0008000 0x80000
size_of_the_qnx_ifs; go 0xc0008000;'

saveenv
```

Remove the Network cable now and Reset the Board. The Board now boots the U-Boot and after the boot delay, it boots up the QNX IFS image.

6. Known Issues for This BSP#

1. The network driver will not behave properly if issued with a flood ping of more than 50000 bytes (refer PR : 74266)
2. The USB 2.0 driver gives a sloginfo warning " devu-omapl1xx-mg.so : MENTOR_ProcessArgs - Unknown option irq=58 " because of incorrect parsing by io-usb.