

# Release Notes for QNX Neutrino 6.4.0 BSP for Renesas SH7774 GoldAle 1.0.0#

## System requirements#

### Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: renesas sh7774 GoldAle board
- 64M RAM

### 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 tables below depict the memory layout for the image and for the flash.

| Item                | Address    |
|---------------------|------------|
| OS image loaded at: | 0x88010000 |

## Getting Started#

### Step 1: Connect your hardware#

Connect the serial cable to the Debug0 serial port of the Renesas SH7774 GoldAle and to a free 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#

- unzip the bsp package archive to the desired location on the host machine
- cd to the root of the BSP and type 'make'. The resulting OS image will be located in the images directory

### Step 3: Transfer the OS image to the target#

#### Step 3A: Transfer the OS image to the target using the ROM monitor#

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

- Baud: 38400
- Bits: 8
- Stop bits: 1
- Parity: none
- Flow control: none

To boot using the ROM monitor, you must first modify the image format in your build file.

1. Modify the following line in your build (images/goldale.build): **[virtual=shle/binary +compress] .bootstrap = {**

to this:

**[virtual=shle/srec +compress] .bootstrap = {**

Note: The name of the resultant image will remain ifs-goldale.bin.

2. Once you've made the change, you must regenerate the image by typing **make** in the images directory:

**# make**

3. Connect to your board, and at the boot monitor prompt, type "h" to get the help information:

```
SH7774/little/map0 (c)Renesas Technology Corp. $Date: 2006/02/07 01:11:40 $
jh
SH monitor commands
D {sadr {eaddr}}      memory dump (DM sets dump size)
F [sadr] [eaddr] [data] fill memory
FL [sadr] [eaddr] [data] fill memory(LONG)
G {start_adr}         go program
H                     help
M [adr]               set memory(BYTE)
MW [adr]              set memory(WORD)
ML [adr]              set memory(LONG)
X {reg {data}}        change register
R                     display registers
L                     load program
DM {B|W|L}            set&disp dump mode
MV [sadr] [dadr] [len] move memory
CF                    LAN/ATA-flash
TM {sadr eaddr}       memory test
]
```

4. Type "L" or "l" to load image to target.

```
]l
please send ! ( '.' & CR stop load)
```

5. Copy the image to a serial port (you may need to connect your terminal session): **cp -v ifs-goldale.bin /dev/ser1**

6. Jump to the image

You can find this address from the output of the the mkifs utility: For example:

```
# make
make
mkifs -r./install -v goldale.build ifs-goldale.bin
Offset Size Entry Ramoff Target=Host
88010000 100 ---- --- Startup-header
88010100 c008 88011fc0 --- /tmp/FAA891574
8801c108 5c ---- --- Image-header
8801c164 72c ---- --- Image-directory
      ---- --- ---- --- tmp=/dev/shmem
      ---- --- ---- --- dev/console=/dev/ser1
...
```

Note: 88011fc0 is the address to use.

You should now see the Neutrino welcome message on your terminal screen:

```
Welcome to QNX Neutrino 6.4.0 on the Renesas Goldale SH7774
```

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

### Step 3B: Transfer the OS image to the target using IPL#

You can write the IPL to the flash. This may be desirable once you've tweaked the OS image exactly the way you want it, and you want the board to boot the image automatically, immediately on power up.

- 1.Modify your buildfile to generate a binary image.
- 2.Run the mkimage script, inside the /images directory of the BSP. The output file from this script is a combined IPL/OS image called ipl-ifs-goldale.bin. You'll download this file to the board's memory using the bootloader, and then burn the image into the board's flash.

Here is the mkimage script:

```
#!/bin/sh
#
# Generate a binary image from IPL code for the goldale
# board and combine with the binary OS image
set -v
# Convert IPL into BINARY format
${QNX_HOST}/usr/bin/ntosh-objcopy --input-format=elf32-shl --output-format=binary -R.data ../install/shle/boot/sys/ipl
# Pad BINARY IPL
mkrec -s256k -ffull -r ipl-tmp-goldale.bin > ipl-tmp2-goldale.bin
# Combine the BINARY IPL with the BINARY OS Image
cat ipl-tmp2-goldale.bin ifs-goldale.raw > ipl-ifs-goldale.bin
# Cleaning up temporary files
rm -f *tmp*
```

- 3.Boot the board as described above, using Rom Monitor to download the ifs-goldale.bin boot image.

- 4.You should make the ipl-ifs-goldale.bin available to your target. In this example we copied it into /dev/shmem .

- Enter the following command to start the flash filesystem driver: # **devf-generic -s0x4000000,64M**
- Erase the beginning of flash: # **flashctl -p/dev/fs0 -14M -ev**

- Copy the combined IPL and OS image to flash: **# cp -v /dev/shmem/ipl-ifs-goldale.bin /dev/fs0**

When the copy is complete, you can reboot; it should now boot from the native QNX IPL. You should see output as follows:

QNX Neutrino IPL for the Goldale :

Commands:

d: download image to RAM

f: scan flash for image

ipl>

5. If you wish to download the OS image serially, press the d key and use sendnto to download the image:

**sendnto -b115200 -d/dev/ser1 ifs-goldale.bin**

## Creating a flash partition#

We'll create a filesystem partition at 4.0 MB offset from the beginning of flash to reserve room for a Neutrino image plus IPL:

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

**# devf-generic -s0x4000000,64M**

2. Erase the flash at offset 4 MB for a length of 60 MB:

**# flashctl -p/dev/fs0 -o4M -l60M -ve**

3. Format the partition:

**# flashctl -p/dev/fs0p0 -o4M -l60M -f**

This command will format the 60 MB partition.

4. Slay the flash filesystem driver:

**# slay devf-generic**

5. Restart the driver:

**# devf-generic -s0x4000000,64M**

You should now have a read/write partition mounted as /fs0p1 off the root directory.

## 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-goldale   | .                 | .                  | src/hardware/startup/boards/goldale |

|             |                                                                                                                                       |                                        |                                                                                                         |                                           |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|---------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Serial      | devc-sersci -e<br>-F -b115200 -x<br>-c 14745600/16<br>scif0&<br>devc-sersci<br>-e -F -u2 -<br>b115200 -x -<br>c 14745600/16<br>scif1& | devc-sersci                            | .                                                                                                       | src/hardware/<br>devc/sersci              |
| Flash (NOR) | devf-generic -<br>s0x4000000,64M                                                                                                      | devf-generic<br>flashctl               | .                                                                                                       | src/hardware/<br>flash/boards/<br>generic |
| Network     | io-pkt-v4<br>-dsmc9000<br>ioport=0x18180000,irq=0x804<br>-ptcpip                                                                      | io-pkt-v4<br>ifconfig                  | devn-smc9000.so<br>libsocket.so                                                                         | src/hardware/<br>devn/smc9000             |
| Graphics    | io-display -<br>dvid=0,did=0                                                                                                          | io-display<br>goldale.conf<br>img.conf | devg-goldale.so<br>devg-soft3d-fixed.so<br>libGLS_CL.so.1<br>libffb.so.2<br>libm.so.2<br>libdisputil.so | src/hardware/<br>devg/goldale             |

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.

## 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-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-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.
- Since the default baud rate of the ROM Monitor is 38400. this BSP uses 38400 as the default value for the debug serial device in starup (main.c) and build file. QNX IPL uses 115200 as the default value.when you use QNX IPL, please make sure to match the baud rate in statup(main.c) and build file.