Release Notes for the QNX Neutrino 6.4.0 BSP for Fujitsu Jade-D EVB 1.0.0#

System requirements#

Target system#

- QNX Neutrino RTOS 6.4.0
- Board version: Fujitsu Jade-D Evaluation Board (Interface Board PA.4 and base board Rev.PA5)
- ROM Monitor version UBoot v 1.3.0
- 32 MB flash

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 table below depicts the memory layout for the image and for the flash.

Memory layout

Item	Address
OS image loaded at:	0x40100000
Flash base address	0x10000000

Getting Started#

Step 1: Connect your hardware#

1. Connect the serial cable to the first (upper) serial port of the Fujitsu Jade-D EVB and the first serial port of your host machine (e.g. ser1 on a Neutrino host).

2. Connect an RJ-45 ethernet cable between the ethernet port on the Fujitsu Jade-D EVB and your local network.

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 #

Boot OS images by 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

2. Make sure the host machine is capable of handling TFTP requests (see the inetd documentation for details).

3. Verify the serial link between the host machine and the target.

4. Verify that an Ethernet link exists between the host machine and the target.

5. Apply power to the target board. You should see output similar to the following:

```
U-Boot 1.3.0-jadeevalkit-0.8.3 (Sep  9 2009 - 14:34:17)

DRAM:  128 MB
Flash: 32 MB
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  0
```

Press any key to stop autobooting, and then you'll see the Fujitsu Jade-D EVB prompt:

```
xxsvideo>
```

On your target, type the following, filling in the appropriate IP addresses and ifs file:

```
xxsvideo> setenv netmask 255.255.240.0
xxsvideo> setenv ipaddr 10.42.101.242
xxsvideo> setenv gatewayip 10.42.96.1
xxsvideo> setenv serverip 10.42.98.211
xxsvideo> setenv 'bootcmd tftpboot 40100000;go 40100000'
xxsvideo> saveenv
```

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

```
Hit any key to stop autoboot:  0
LAN9x18 (0x118a0000) detected.
start Auto negotiation... (take ~2sec)
Auto negotiation complete, 100BaseTX, full duplex
TFTP from server 10.42.98.211; our IP address is 10.42.101.242
Filename 'ifs-jade-d.bin'.
Load address: 0x40100000
Loading: ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         #######################
```

```
done
Bytes transferred = 2111004 (20361c hex)
## Starting application at 0x40100000 ...
Multiplex mode (CMUX_MD): 0x00090000
CPU0: Dcache: 512x32 WB
CPU0: Icache: 512x32
CPU0: 41069265: arm926 rev 5 333MHz

System page at phys:40010000 user:fc404000 kern:fc404000
Starting next program at vfe03d838
cpu_startnext: cpu0 -> fe03d838
Welcome to QNX Neutrino 6.4.0 on the Fujitsu JADE-D Board
Starting on-board ethernet with TCP/IP stack...
Starting I2C driver...
Starting Display driver...
Starting audio driver with cs4245 codec
Starting CAN driver...
#
```

Programming IPL and boot OS images by IPL#

1. Make a backup copy of the ifs-jade.bin image used for U-Boot.

2. Make sure the jade.build is for IPL image (according to the previous note), and then rebuild your image in order to produce a valid IPL file.

3. Once U-Boot is up and running, load an U-Boot ifs-jade.bin image (backed-up previously) onto the board.

4. Set up a shared network drive on your workstation.

In this example we are using a Windows Share folder to accomplish our work. You have to replace the following names with ones that match your configuration.

- WORKSTATION : Windows Computer Name
- WS_IP_ADDR : Computer IP Address
- SHARE_FOLDER : Name use to share folder
- MOUNT_FOLDER : Folder name under which Neutrino will access your share folder
- USER_NAME : User name that has read access to the share folder
- PASSWORD : User's password

5. Ensure that the following utilities are present in your build file.

- fs-cifs
- cp
- slay
- flashctl
- dd

6. Execute the following commands in the Neutrino shell.

```
fs-cifs -l //WORKSTATION:WS_IP_ADDR:/SHARE_FOLDER/MOUNT_FOLDER USER_NAME
PASSWORD
```

```
cp /MOUNT_FOLDER/ipl-ifs-jade-d.bin /dev/shmem/jaded.bin
```

```
slay io-pkt
```

```
flashctl -p /dev/fs0 -o 0m -l 4m -ve
```

```
dd if=/dev/shmem/jaded.bin of=/dev/fs0 bs=1k seek=0k
```

Note:

- Erase the flash only if you've copied a valid IPL image into /dev/shmem.
- Before programming the IPL, you should have proper tools to reprogram the on-board flash. Any error in flash programming will result in an unusable board.

7. Boot OS images by IPL

- Once you've programmed ipl-ifs-jade-d.bin to flash, you can use the IPL to download the image through the serial port or run the image from the flash directly.
- Power on the target board, The terminal emulator will show the following output:

```
QNX Neutrino Initial Program Loader for Fujitsu JADE
Commands:

Press 'D' for serial download, using the sendnto utility
Press 'F' to boot an OS image in flash
```

If you press 'd' or 'D', the terminal emulator will prompt you to download the image through the serial port, as follows:

```
send image now...
```

Use the sendnto utility to download the image.

Step 4: Start working with Neutrino OS#

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. For sample command lines, please see the “Driver Command Summary” section.

Creating a flash partition#

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

```
devf-generic -s0x10000000,32M
```

2. Unlock the entire flash, except for the first megabyte:

Note:
Because the ROM monitor is in the first megabyte of flash, you may not want to erase it. Use the -l (length) and -o (offset) options to avoid this area.

```
flashctl -p/dev/fs0 -o1M -l31M -U
```

3. Erase the flash, except for the first megabyte:

```
flashctl -p/dev/fs0 -o1M -l31M -ve
```

4. Format the partition:

```
flashctl -p/dev/fs0p0 -o1M -l31M -vf
```

5. Slay, then restart the driver to mount the new partition:

```
slay devf-generic
```

```
devf-generic -s0x10000000,32M
```

You should now have /fs0p1 directory automounted.

Driver Command Summary#

The driver command lines below are specific to the Fujitsu Jade-D Evaluation Board. See the online docs for each driver for additional command-line options and other details.

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	`startup-jade-d`	.	.	`src/hardware/startup/boards/jade-d`
Serial	`devc-ser8250 -e -F -S -b115200 -c41500000 0xfffe1000^2,24`	devc-ser8250	.	`src/hardware/devc/ser8250`
Network	`io-pkt-v4 -dsmc9118 ioport=0x02000000,irq=10,mac=6622000021605,verbose -pqnet`	io-pkt-v4 ifconfig	devn-smc9118.so devnp-shim.so libsocket.so lsm-qnet.so	`src/hardware/devn/smc9118`
Flash	`devf-generic -s0x10000000,32M`	devf-generic	.	`/src/hardware/flash/boards/generic`
I2C	`Channel 0: i2c-mb8x -p 0xfff56000 Channel 1: i2c-mb8x -p 0xfff57000 -i 44 --u 1`	i2c-mb8x	.	`src/hardware/i2c/mb8x`
Audio	`io-audio -v -d jade i2schn=0,noioexp,i2cdev=1,codecaddr=0x4e`	io-audio	deva-ctrl-jade.so libasound.so deva-mixer-ac97.so deva-util-restore.so	`src/hardware/deva/ctrl/jade`
Graphics	`io-display -dvid=0,did=0`	io-display	devg-jade.so libm.so libGLES_CL.so.1 libffb.so.2 libgf.so.1 libdisputil.so	`src/hardware/devg/jade`
SPI	`spi-mb8x -k -x`	spi-mb8x	.	`src/hardware/spi/mb8x`
CAN	`Channel 0: dev-can-jade can0`	dev-can-jade canctl	.	`src/hardware/can/jade`

	Channel 1: dev-can-jade can1			

Note:
Some of these drivers can be 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:#

Some modules aren't enabled after boot up, so you need to use command-line options to startup to enable them. You must pass these options before any other startup options.

Command :

```
startup-jade-d options
```

A command-line option is required to startup to enable some modules.

`-m 0xXXXX`: Groups 2-4, 6-11 multiplex I/O configuration. Value can range from 0-0x1fff (default=0x0009).

Note:
For video capture use option `-m 0x0008`.
To enable RGB video input instead of ITU656 use option `-m 0x0019`.

Audio:#

The I2C channel 1 driver should be running before start the audio driver:
`i2c-mb8x -p 0xfff57000 -i 44 --u 1`
Then start the audio driver:
`io-audio -vv -d jade i2schn=0,noioexp,i2cdev=1,codecaddr=0x4e`

Graphics:#

To start the drive:

```
io-display -dvid=0,did=0
```

Note:
Starting io-display simply starts the display server. Nothing will appear on the display until a graphical application is run (i.e. vsync or egl-gears).

Required config files:

- /etc/system/config/display.conf
- /etc/system/config/jade.conf

The `/etc/system/config/display.conf` configuration file describes the graphics memory interface settings. It should contain the following for a 640x480 resolution:

```
device {
   drivername=jade
   vid=0
   did=0
```

```
deviceindex=0
modeopts=/etc/system/config/jade.conf
display {
   xres=640
   yres=480
   refresh=60
   pixel_format=argb1555
   }
}
```

> **Note**:
> For more information about these commands, see the Neutrino Utilities Reference.

Known issues for this BSP#

- For usages of peripherals (such as I2S/SPI/GPIO/PWM) correct multiplex IO configuration must be set. Please refer to the Jade-D Evaluation board documentation. For more information on how to resolve possible peripheral multiplex conflicts contact Fujitsu.
- The frame synchronization of I2S_WSx signal has some issues which may cause channel-swapping when you capture/playback audio simultaneously. We've added a workaround to avoid this issue. You can disable this workaround by commenting out the line #define AUDIO_WORKAROUND in the file src \hardware\deva\ctrl\jade\jade_dll.c.
- If you want to run the CRT monitor at the highest resolution(1024X768), you need to modify the default jade.conf file to increase the vsize from 0x400000 to 0x1000000. The display.conf file be used to configure the resolutions.
- MediaLB functionality is not included in this BSP due to SMSC license restrictions. QNX MediaLB functionality for Jade is available as a separate archive for use with "NetService library v2.1.1", which you must obtain from SMSC directly. Please contact your QNX sales representative to request the QNX MediaLB Jade archive.
- 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.
- The graphics driver devg-jade.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.
- Graphics driver does work with multiple displays (Ref #69850).