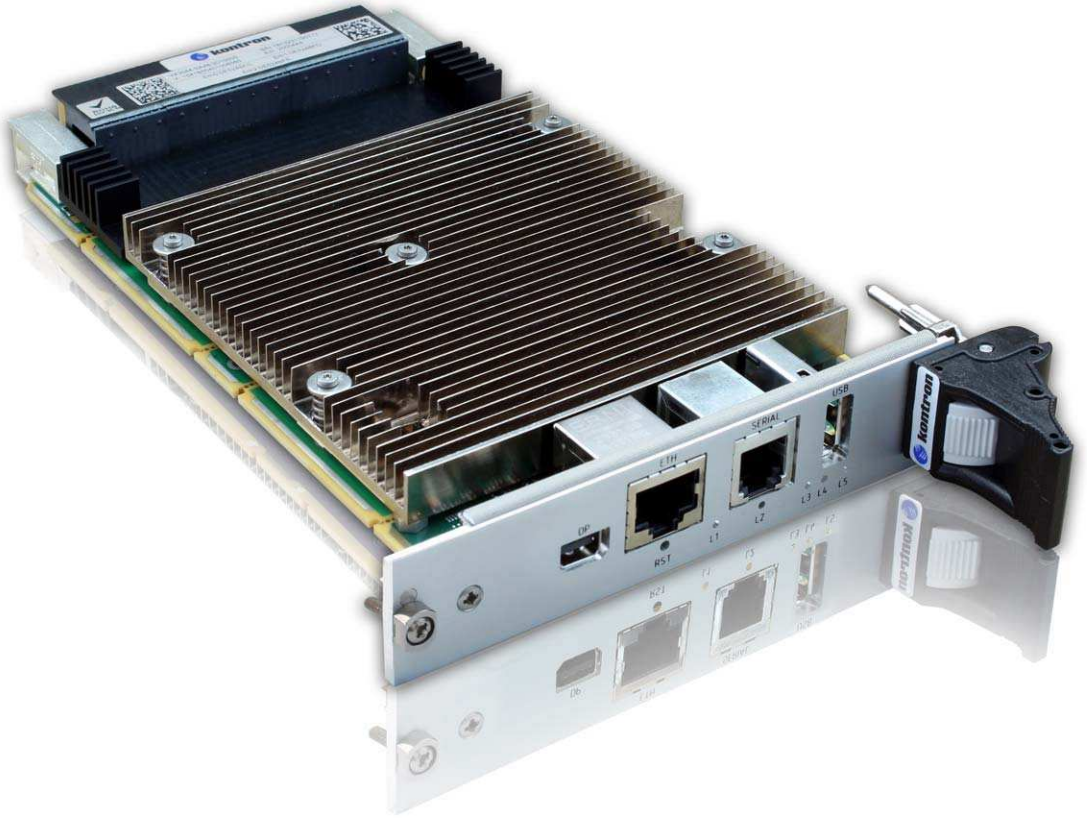
» VX304x «



Release Notes VxWorks 6.9 64 bits on VX304x

SD.DT.G30-0e - September 2013

Revision History

Publication Title:	VxWorks 6.9 64 bits on VX304x	
Doc. ID:	SD.DT.G30-0e	
Rev.	Brief Description of Changes	Date of Issue
0e	Initial Issue	09-2013

Proprietary Note

This document contains information proprietary to Kontron. It may not be copied or transmitted by any means, disclosed to others, or stored in any retrieval system or media without the prior written consent of Kontron or one of its authorized agents.

The information contained in this document is, to the best of our knowledge, entirely correct. However, Kontron cannot accept liability for any inaccuracies or the consequences thereof, or for any liability arising from the use or application of any circuit, product, or example shown in this document.

Kontron reserves the right to change, modify, or improve this document or the product described herein, as seen fit by Kontron without further notice.

Trademarks

This document may include names, company logos and trademarks, which are registered trademarks and, therefore, proprietary to their respective owners.

Environmental Protection Statement

This product has been manufactured to satisfy environmental protection requirements where possible. Many of the components used (structural parts, printed circuit boards, connectors, batteries, etc.) are capable of being recycled.

Final disposition of this product after its service life must be accomplished in accordance with applicable country, state, or local laws or regulations.

Environmental protection is a high priority with Kontron.

Kontron follows the DEEE/WEEE directive.

You are encouraged to return our products for proper disposal.

The Waste Electrical and Electronic Equipment (WEEE) Directive aims to:

- reduce waste arising from electrical and electronic equipment (EEE)
- make producers of EEE responsible for the environmental impact of their products, especially when they become waste
- encourage separate collection and subsequent treatment, reuse, recovery, recycling and sound environmental disposal of EEE
- improve the environmental performance of all those involved during the lifecycle of EEE

Conventions

This guide uses several types of notice: Note, Caution, ESD.

Note: this notice calls attention to important features or instructions.


Caution: this notice alert you to system damage, loss of data, or risk of personal injury.


ESD: This banner indicates an Electrostatic Sensitive Device.

All numbers are expressed in decimal, except addresses and memory or register data, which are expressed in hexadecimal. The prefix `0x' shows a hexadecimal number, following the `C' programming language convention.

The multipliers `k', `M' and `G' have their conventional scientific and engineering meanings of $*10^3$, $*10^6$ and $*10^9$ respectively. The only exception to this is in the description of the size of memory areas, when `K', `M' and `G' mean $*2^{10}$, $*2^{20}$ and $*2^{30}$ respectively.

When describing transfer rates, `k' `M' and `G' mean $*10^3$, $*10^6$ and $*10^9$ *not $*2^{10}$ $*2^{20}$ and $*2^{30}$.*

In PowerPC terminology, multiple bit fields are numbered from 0 to n, where 0 is the MSB and n is the LSB. PCI and CompactPCI terminology follows the more familiar convention that bit 0 is the LSB and n is the MSB.

Signal names ending with an asterisk (*) or a hash (#) denote active low signals; all other signals are active high.

Signal names follow the PICMG 2.0 R3.0 CompactPCI Specification and the PCI Local Bus 2.3 Specification.

For Your Safety

Your new Kontron product was developed and tested carefully to provide all features necessary to ensure its compliance with electrical safety requirements. It was also designed for a long fault-free life. However, the life expectancy of your product can be drastically reduced by improper treatment during unpacking and installation. Therefore, in the interest of your own safety and of the correct operation of your new Kontron product, you are requested to conform with the following guidelines.

High Voltage Safety Instructions


Warning!
All operations on this device must be carried out by sufficiently skilled personnel only.


Caution, Electric Shock!
Before installing a not hot-swappable Kontron product into a system always ensure that your mains power is switched off. This applies also to the installation of piggybacks. Serious electrical shock hazards can exist during all installation, repair and maintenance operations with this product. Therefore, always unplug the power cable and any other cables which provide external voltages before performing work.

Special Handling and Unpacking Instructions

ESD Sensitive Device!
Electronic boards and their components are sensitive to static electricity. Therefore, care must be taken during all handling operations and inspections of this product, in order to ensure product integrity at all times

Do not handle this product out of its protective enclosure while it is not used for operational purposes unless it is otherwise protected.

Whenever possible, unpack or pack this product only at EOS/ESD safe work stations. Where a safe work station is not guaranteed, it is important for the user to be electrically discharged before touching the product with his/her hands or tools. This is most easily done by touching a metal part of your system housing.

It is particularly important to observe standard anti-static precautions when changing piggybacks, ROM devices, jumper settings etc. If the product contains batteries for RTC or memory backup, ensure that the board is not placed on conductive surfaces, including anti-static plastics or sponges. They can cause short circuits and damage the batteries or conductive circuits on the board.

General Instructions on Usage

In order to maintain Kontron's product warranty, this product must not be altered or modified in any way. Changes or modifications to the device, which are not explicitly approved by Kontron and described in this manual or received from Kontron's Technical Support as a special handling instruction, will void your warranty.

This device should only be installed in or connected to systems that fulfill all necessary technical and specific environmental requirements. This applies also to the operational temperature range of the specific board version, which must not be exceeded. If batteries are present, their temperature restrictions must be taken into account.

In performing all necessary installation and application operations, please follow only the instructions supplied by the present manual.

Keep all the original packaging material for future storage or warranty shipments. If it is necessary to store or ship the board, please re-pack it as nearly as possible in the manner in which it was delivered.

Special care is necessary when handling or unpacking the product. Please consult the special handling and unpacking instruction on the previous page of this manual.

Table of Contents

Table of Contents

1 INTRODUCTION

This document is the VX304x BSP User's Guide for the BSP vxWorks 6.9 64 bits EFT. It gives the installation procedure for the VX304x vxWorks 6.9 BSP complemented with the vxWorks 6.9 Update pack3 Service pack 2 patch and Workbench 3.4 patch level required to support this VX304x BSP. Note that this BSP is an EFT (Early Field Test Sample).

This file also contains information on the board BIOS configuration, BSP general information, used device drivers and finally limitations and BSP evolutions compared to the previous versions.

The document contains the following first level chapters:

INTRODUCTION
 - VxWorks REQUIREMENTS
 - BIOS REQUIREMENT
 - JUMPERS & SWITCHS
BSP INSTALLATION GUIDE
BOOTING VXWORKS ON VX304x
HARDWARE FEATURES SUPPORTED BY VX304x BSP
KONTRON DEVICES DRIVERS
BSP INFORMATION
KNOWN PROBLEMS
BSP EVOLUTIONS
BIBLIOGRAPHY
SEE ALSO

1.1 VxWorks Requirements

To support VX304x BSP, VxWorks 6.9 and Workbench 3.4 must be first updated to

VxWorks 6.9 updated to 6.9.3.2 (Update Pack 3 Service Pack 2)
Workbench 3.3 updated to 3.3.4 (Update Pack Pack 4)

1.2 BIOS Requirements

The Board must be in System controller slot in the VPX rack. The BIOS ID of the VX304x must be equal to **13274EFT**. The BIOS ID can be verified into the first page of BIOS SETUP. If the BIOS is not at this level on your board please update it.

Updating the BIOS: Start the board under the EFI shell. To do that, please entering in the BIOS menu by pressing **F2** or DEL during the boot process of the VX304x.

* Save & exit -> Boot Override : [UEFI: Built-in EFI Shell]

The console will dislpay the following prompt :

```
VX304x>
```

Plug a USB that contains the BIOS binary VX3040_ID13274EFT.bin on the USB connector on the front panel of the VX304x. This binary is present in the current DVD "VxWorks 6.9 64bits BSP for VX304x ID13275_EFT" in the folder **bios**. First type the kmac command.

```
VX304x> kmac -r
MAC Address LAN ETH0 (Intel 82599) = 00:00:DE:52:4B:AF
MAC Address LAN ETH1 (Intel 82599) = 00:00:DE:52:4B:B0
MAC Address LAN ETH2 (Intel 82579) = 00:00:DE:52:4B:AE
```

Get the mac address of the Intel 82579. In this example it is : "00:00:DE:52:4B:AE". Once it is done, go to the USB device that corresponds to the USB key. Type the **map** command :

```
VX304x> map -r
Device mapping table
  fs0  :HardDisk - Alias hd27e65535a1 blk0
        PciRoot(0x0)/Pci(0x1f,0x2)/Sata(0x4,0x0)/HD(1,MBR,0x0f800000,0x3f,0x395547)
  fs1  :HardDisk - Alias hd27e65535a2 blk1
        PciRoot(0x0)/Pci(0x1f,0x2)/Sata(0x4,0x0)/HD(2,MBR,0x0f800000,0x395586,0x380d92a)
  fs2  :Removable HardDisk - Alias hd25b0b0b blk2
        PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x1,0x0)/HD(1,MBR,0x0f8000b1,0x3f,0xe67f6)
  fs3  :Removable HardDisk - Alias hd25b0b0c blk3
        PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x1,0x0)/HD(2,MBR,0x0f8000b1,0xe6835,0xe047a4)
  blk0 :HardDisk - Alias hd27e65535a1 fs0
        PciRoot(0x0)/Pci(0x1f,0x2)/Sata(0x4,0x0)/HD(1,MBR,0x0f800000,0x3f,0x395547)
  blk1 :HardDisk - Alias hd27e65535a2 fs1
        PciRoot(0x0)/Pci(0x1f,0x2)/Sata(0x4,0x0)/HD(2,MBR,0x0f800000,0x395586,0x380d92a)
  blk2 :Removable HardDisk - Alias hd25b0b0b fs2
        PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x1,0x0)/HD(1,MBR,0x0f8000b1,0x3f,0xe67f6)
  blk3 :Removable HardDisk - Alias hd25b0b0c fs3
        PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x1,0x0)/HD(2,MBR,0x0f8000b1,0xe6835,0xe047a4)
  blk4 :BlockDevice - Alias (null)
        PciRoot(0x0)/Pci(0x1f,0x2)/Sata(0x4,0x0)
  blk5 :Removable BlockDevice - Alias (null)
        PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x1,0x0)
  hd27e65535a1 :HardDisk - Alias fs0 blk0
```

In the case the fs0/fs1 devices correspond to the partion 0 and 1 of a Sata disk as indicated in the log of the console. The fs2/fs3 devices correspond to the partion 0 and 1 of the USB that is plugged on the front panel of the vx304x.

In the example the new binary bios file is present on the partition 0 of the USB key. So type :

```
VX304x> fs2:
fs2:\>
```

Note that the prompt change to indicate that we are in the USB key Type also :

```
fs2:\> dir
Directory of: fs2:\

  01/01/80  12:00a               5,136  bootrom.sys
  01/01/80  12:00a           1,094,016  bootapp.sys
  01/01/80  12:00a           7,980,108  vxWorksSMP
  01/01/80  12:00a           7,904,467  vxWorksUP
  01/01/80  12:00a           8,388,608  VX3040_ID13274EFT.bin
```

Now we can update the BIOS

```
fs2:\> kflash -e -r -p VX3040_ID13274EFT.bin
```

Wait for the end of the flashing procedure. Now the mac address must be refilled. In this example the Mac address is 00:00:DE:52:4B:AE. For the kmac command all **:** must be replace with no character and the word must be preceeded by **0x** Reflash the orignal ethernet mac address as follow:

```
fs2:\> kmac -wf 0x0000DE524BAE
```

Now the bios ID13275EFT is flash on the board. Switch off the rack during 5 seconds and switch on afterwards to reach teh EFI prompt. Boot under the EFI shell again and type the following command :

```
VX304x> kvpx -plx_eeprom prog
```

Than apply a reset or a power off/pwoer on cycle.

Default BIOS setting to boot vxWorks: First of all **restore the default values**. To proceed, enter in the BIOS by pressing **F2** or DEL during the boot process of the VX304x. Then Press

* F3 -> Optimized Defaults

After this is done and the Optimized default BIOS parameters are set then do the following:

* Advanced -> CPU Configuration -> **Hyper-Threading [Disabled]**
* Kontron -> VPX Configuration -> **VPX EEPROM Config. [Enabled]**
* Kontron -> VPX Configuration -> **VPX Switch Mode [set by SYSCON]**

Once the setting is done save the parameters for the next boot :

* Save & exit -> **Save changes and Reset**

For PXE boot: Check the PCH LAN Controller is enabled

* Chipset -> PCH-IO Configuration -> PCH LAN Controller: [Enable]

Enable the PXE OpROM

* Boot -> CSM parameters -> **Launch PXE OpRom Policy -> Legacy Only**

Then Enable PXE on the wished interface and disable it on other interfaces. **ETH2** is the single 1Gbe 825799 ethernet interface (front side if available) while **ETH0/ETH1** are the Dual 82599 ETH interface on backplane P2 connector (10G capable).

* Boot -> CSM parameters -> PXE OpRom ETH0 -> Enabled or Disable
* Boot -> CSM parameters -> PXE OpRom ETH1 -> Enabled or Disable
* Boot -> CSM parameters -> PXE OpRom ETH2 -> Enabled or Disable

Need to save parameters (current BIOS setting) and re-enter in the BIOS to select the boot device for ethernet

* Save & Exit -> **Save Changes and Reset**

To re-enter in the BIOS by pressing **F2** or DEL during the boot process of the VX304x.

* Boot -> Network Device BBS Priorities -> Boot Option #1: [IBA GE Slot ...]
  F4: Save Changes and Exit, then go to BIOS setup again and set boot device (PXE: Boot -> Boot Option #

Please note that the ethernet device named [**IBA GE** Slot ...] is the front panel RJ45 interface

1.3 Jumpers & Switches

Before using a board with VxWorks, verify that the board runs in the factory configuration by using vendor-supplied ROMs and jumper settings and checking the RS-232 connection. Refer to 'CA.DT.A98 VX3042 and VX3044 User's Guide' for a detailed description of the board.

2 BSP INSTALLATION GUIDE

2.1 Install vxWorks 6.9 / Workbench 3.4

Get and install the DVDROM of the vxWorks 6.9 release.

DVD-R147826.1-1-01 : VxWorks-6.9 and VxWorks Edition 6.9 Platform

Remark: In the "Installer" Window, when asked to "Choose on line update setting",
Remove selection for the 2 following items:
- check for and apply latest updates for this installer
- check online for the latest Wind River product updates

2.2 Install VxWorks 6.9.3.2

DVD-R147826.1-17-00 : VxWorks 6.9.3.2

The Upadate Pack 3 Service pack 2 vxWorks 6.9 from WindRiver is present in this DVDROM in the following folder :

updatePack3_servicePack2

Uncompress the archive file delivered that corresponds to **DVD-R147826.1-17-00**. Go to the **DVD-R147826.1-17-00** folder and launch the appropriate shell script for the host machine.

For Linux :

```
./setup_linux
```

For Windows launch :

```
setup.exe
```

For Solaris :

```
./setup_solaris
```

Install the update pack 3 service pack 2 release

2.3 Install Workbench Update Pack 4

DVD-R158451.1-1-19 : Wokbench 3.4 Update Pack 4

Uncompress the archive file delivered by WindRiver that corresponds to DVD-R158451.1-1-19. (see the WINDRIVER support web site : support.windriver.com, do not download or unzip it in install/updates directory !)

Go to the **DVD-R158451.1-1-19** folder and launch the appropriate shell script for the host machine.

For Linux :

```
./setup_linux
```

For Windows' launch :

```
setup.exe
```

For Solaris :

```
./setup_solaris
```

Install the Workbench 3.3 update pack 4 release.

The Update Pack 4 Workbench 3.3 from WindRiver is present in the DVDROM named **Workbench-3.3 Update pack 4 for VX304x**

2.4 Install the VX304x BSP

The BSP of the VX304x is located in **VXW69-BSP-VX304x-xxxxx** folder. Where xxxxx identify the BSP ID.

Files and folders present at the top of the VXW69-BSP-VX304x-xxxxx folder :

bios : contains the BIOS VX3040_ID13274EFT.bin binary file
bootrom64 : folder that contains bootrom 64 bits and multistaging bootstap binaries
updatePack3_servicePack2 : folder that contains the vxWorks 6.9 Update Pack 3 Service Pack 2
vx304x.pdf : Describes the BSP release for vxWorks 6.9 64 bits for vx304x
vximages64 : folder that contains UP/SMP gnu/icc 64 bits vxWorks images
VXW69-BSP : contains the vxWorks BSP files and docs
VXW69-BSP-VX304x-xxxxxx.zip: BSP zip file ID xxxxxx to be used for installation

Files contained in the **bootrom64** folder :

```
   bootrom64
Â Â      |-- bootapp.sys                : Real bootrom with SATA/USB/ethernet 82579 boot support
Â Â      |-- bootrom.bin          : Multistaging bootstrap for SATA and USB boot
Â Â      \-- bootrom.pxe          : Multistaging bootstrap for PXE boot.
```

Files contained in the **vximages64** folder :

```
   vximages64
      |-- SMP
      |   |-- gnu
      |   |Â Â  |-- vxWorks    : Vxworks 64 bits image UP compiled with gnu and including code to
      |   |-- icc
      |   \-- vxWorks      : Vxworks 64 bits image UP compiled with icc and including code to init
   Â Â  \-- UP
   Â Â      |-- gnu
           |   |-- vxWorks      : Vxworks 64 bits image SMP compiled with gnu and including code to
   Â Â      \-- icc
   Â Â          \-- vxWorks      : Vxworks 64 bits image SMP compiled with icc and including code
```

First **set the environment variables** by typing under the WindRiver Installation folder:

Linux OS :

```
./wrenv.linux -p vxworks-6.9
```

Windows OS :

```
wrenv.exe -p vxworks-6.9
```

Go to the DVDROM directory.

Unzip the file **VXW69-BSP-VX304x-xxxxx.zip** in the home vxWorks 6.9 directory as follows :

under **linux OS** :

```
cd $WIND_HOME
unzip /mnt/cdrom/VXW69-BSP-VX304x-xxxxx.zip -d .
```

under **Windows OS** (if **WIND_HOME=**C:\WindRiver and D: is the DVDROM) :

unzip the D:\VXW69-BSP-VX304x-xxxxx.zip in C:\WindRiver

During the unzipping process, in case the following message appears :

```
./vxworks-6.9/target/h/make/rules.bsp? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
```

Type A to replace existing files.

2.5 Recompile libs

Once the patches are installed, the platform needs to be regenerated for the target.

Rename/Recompile an existing file :

-- Under **Linux OS** --

```
mv $WIND_HOME/vxworks-6.9/target/h/hwif/vxbus/vxbMethodDecl.h
 $WIND_HOME/vxworks-6.9/target/h/hwif/vxbus/vxbMethodDecl.h.orig


cd $WIND_HOME/vxworks-6.9/target/src/hwif/methods

make vxbMethodDecl.h
```

The following commands must be performed after each BSP installation to add all new drivers registration:

```
mv $WIND_HOME/vxworks-6.9/target/config/all/vxbUsrCmdLine.c
$WIND_HOME/vxworks-6.9/target/config/all/vxbUsrCmdLine.c.orig

cd $WIND_HOME/vxworks-6.9/target/config/comps/src/hwif
```

```
make vxbUsrCmdLine.c
```

-- Under **Windows OS** --

```
mv %WIND_HOME%/vxworks-6.9/target/h/hwif/vxbus/vxbMethodDecl.h
%WIND_HOME%/vxworks-6.9/target/h/hwif/vxbus/vxbMethodDecl.h.orig

cd %WIND_HOME%/vxworks-6.9/target/src/hwif/methods

make vxbMethodDecl.h
```

The following commands must be performed after each BSP installation to add all new drivers registration:

```
mv %WIND_HOME%/vxworks-6.9/target/config/all/vxbUsrCmdLine.c
%WIND_HOME%/vxworks-6.9/target/config/all/vxbUsrCmdLine.c.orig

cd %WIND_HOME%/vxworks-6.9/target/config/comps/src/hwif

make vxbUsrCmdLine.c
```

Go to :

-- Under **Linux OS** (Only do the gnu and icc libs) --

```
cd $WIND_HOME/vxworks-6.9/target/src

make CPU=NEHALEM TOOL=gnu VXBUILD=LP64 rclean && \
make CPU=NEHALEM TOOL=icc VXBUILD=LP64 rclean && \
make CPU=NEHALEM TOOL=gnu VXBUILD=LP64 VXBUILD+=SMP rclean && \
make CPU=NEHALEM TOOL=icc VXBUILD=LP64 VXBUILD+=SMP rclean && \
make CPU=NEHALEM TOOL=gnu VXBUILD=LP64 && \
make CPU=NEHALEM TOOL=icc VXBUILD=LP64 && \
make CPU=NEHALEM TOOL=gnu VXBUILD=LP64 VXBUILD+=SMP && \
make CPU=NEHALEM TOOL=icc VXBUILD=LP64 VXBUILD+=SMP
```

-- Under **Windows OS** --

```
cd %WIND_HOME%/vxworks-6.9/target/src

make CPU=NEHALEM TOOL=gnu VXBUILD=LP64 rclean && \
make CPU=NEHALEM TOOL=icc VXBUILD=LP64 rclean && \
make CPU=NEHALEM TOOL=gnu VXBUILD=LP64 VXBUILD+=SMP rclean && \
make CPU=NEHALEM TOOL=icc VXBUILD=LP64 VXBUILD+=SMP rclean && \
make CPU=NEHALEM TOOL=gnu VXBUILD=LP64 && \
make CPU=NEHALEM TOOL=icc VXBUILD=LP64 && \
make CPU=NEHALEM TOOL=gnu VXBUILD=LP64 VXBUILD+=SMP && \
make CPU=NEHALEM TOOL=icc VXBUILD=LP64 VXBUILD+=SMP
```

2.6 Compile VX304x kernel

The installation procedure is ended. The VX304x kernel can then be compiled by command or the Workbench Project can be launched to configure and compile a new image kernel.

To compile by command, make sure the environment variables are initialized:

Linux OS :

```
./wrenv.linux -p vxworks-6.9
```

Windows OS :

```
wrenv.exe -p vxworks-6.9
```

Then under **Linux OS** or **Windows OS** using the **vxprj** command :

```
cd vxworks-6.9/target/config/vx304x_64

cd ../../proj;vxprj create -lp64 -force vx304x_64 gnu vx304x_64_gnu
cd vx304x_64_gnu;vxprj bundle add BUNDLE_VX304x

vxprj component add INCLUDE_DOSFS && \
vxprj component add INCLUDE_FS_MONITOR && \
vxprj component add INCLUDE_SHELL && \
vxprj component add INCLUDE_SHELL_STARTUP && \
vxprj component add INCLUDE_DRV_STORAGE_INTEL_AHCI &&\
vxprj component add INCLUDE_SHELL_INTERP_CMD && \
vxprj component add INCLUDE_SHELL_INTERP_C && \
vxprj component add INCLUDE_SHELL_EMACS_MODE && \
vxprj component add INCLUDE_VXBUS_SHOW && \
vxprj component add INCLUDE_SHOW_ROUTINES && \
vxprj component add INCLUDE_SHELL_BANNER && \
vxprj component add INCLUDE_DISK_UTIL && \
vxprj component add INCLUDE_DISK_UTIL_SHELL_CMD && \
vxprj component add INCLUDE_USB && \
vxprj component add INCLUDE_USB_INIT && \
vxprj component add INCLUDE_EHCI && \
vxprj component add INCLUDE_EHCI_INIT && \
vxprj component add INCLUDE_USB_MS_BULKONLY && \
vxprj component add INCLUDE_USB_MS_BULKONLY_INIT && \
vxprj component add INCLUDE_RTP && \
vxprj component add INCLUDE_ROMFS

vxprj build
```

or if the SMP (Symmetric Multiprocessing Support) feature is expected

```
cd vxworks-6.9/target/config/vx304x_64
cd ../../proj;vxprj create -smp -lp64 -force vx304x_64 gnu vx304x_64_smp_gnu
cd vx304x_64_smp_gnu;vxprj bundle add BUNDLE_VX304x

vxprj component add INCLUDE_DOSFS && \
vxprj component add INCLUDE_FS_MONITOR && \
vxprj component add INCLUDE_SHELL && \
vxprj component add INCLUDE_SHELL_STARTUP && \
vxprj component add INCLUDE_DRV_STORAGE_INTEL_AHCI && \
vxprj component add INCLUDE_SHELL_INTERP_CMD && \
vxprj component add INCLUDE_SHELL_INTERP_C && \
vxprj component add INCLUDE_SHELL_EMACS_MODE && \
vxprj component add INCLUDE_VXBUS_SHOW && \
vxprj component add INCLUDE_SHOW_ROUTINES && \
```

```
vxprj component add INCLUDE_SHELL_BANNER && \
vxprj component add INCLUDE_DISK_UTIL && \
vxprj component add INCLUDE_DISK_UTIL_SHELL_CMD && \
vxprj component add INCLUDE_USB && \
vxprj component add INCLUDE_USB_INIT && \
vxprj component add INCLUDE_EHCI && \
vxprj component add INCLUDE_EHCI_INIT && \
vxprj component add INCLUDE_USB_MS_BULKONLY && \
vxprj component add INCLUDE_USB_MS_BULKONLY_INIT && \
vxprj component add INCLUDE_RTP && \
vxprj component add INCLUDE_ROMFS

vxprj build
```

Warning: make from the BSP directory (**make vxWorks VXBUILD=SMP**) **is not supported** with vxWorks 6.9, please move to using the vxprj command line build.

3 BOOTING VXWORKS ON VX304x

When the board is powered on, the system BIOS code loads and executes the bootstrap loader. The bootstrap loader is written in 8088 16-bit assembly language. The BIOS obtains the bootstrap loader from the boot sector, which may be in one of several locations: an USB key, a hard disk, or some other alternatives like in PXE boot. When the BIOS finds the bootstrap loader, it transfers execution to it. The bootstrap loader finds the bootrom.sys file, loads it into memory, and transfers execution to it. Due to some bootrom higher in memory footprint that 512KB, the current boot loader must switch in 32 bits mode to load bootrom higher that 512KB. In that way, this multistaging behaviour is in three steps for USB and SATA boot, and it is in two steps in PXE boot.

4 SATA/USB

BIOS Legacy boot -> bootstrap FreeBSD -> bootstrap Kontron -> Bootrom

PXE:
BIOS legacy boot -> bootstrap Kontron (using PXENV of PXE ROM) -> Bootrom

4.1 Booting VxWorks via PXE

It is possible to boot VxWorks via the network using PXE (the Preboot eXecution Environment). PXE support is provided by the BIOS. The advantage of using PXE is that no floppy or hard disk is required. This is an important point, indeed, support for floppy disk drives is becoming increasingly rare with newer PC systems.

A bootstrap bootrom.pxe and the bootrom named bootapp.sys are provided for PXE booting VxWorks in addition to the regular bootrom.bin loader. When booting from disk, the vxld.bin loader is normally loaded by the BIOS at the address 0x7C00. It then in turn loads the bootrom at address 0x8000 and starts it running. By contrast, when booting with PXE, the bootstrap bootrom.pxe is loaded directly at address 0x7C00 by the PXE ROM. A leading block of 1024 NOP instructions is therefore used to force the **romInit( )** entry point to fall at the address 0x8000, just as it would do if it had been loaded from a disk. Once loaded and running, bootrom.pxe is able to load a bootrom images via the network just as the normal bootrom.bin does.

PXE booting requires the use of a DHCP and TFTP server. When the PXE ROM starts up, it will use DHCP to discover an IP address and the path for the bootrom file. Generally, the PXE ROM assumes that the DHCP and TFTP servers will be running on the same machine. The way the DHCP and TFTP servers are configured will depend on what software is used. DHCP and TFTP supports are available for most *NIX/Linux/*BSD systems, as well as for Microsoft Windows (R). The following sample of setup instructions explains how to configure the open source ISC DHCP server running on a FreeBSD system.

Assume that your client machine is on the 10.0.0.0/32 network, and that your DHCP/TFTP server's IP address is 10.0.0.1. When the PXE client is started up, the PXE ROM should display the station address of the ethernet port. We will assume the address is 00:00:E8:01:02:03, and that we want to assign the target an IP address of 10.0.0.3. Given this, we need to create a dhcpd.conf file as follows:

```
# dhcpd.conf
#
# Sample PXE boot configuration file for ISC dhcpd
#

ddns-update-style none;

subnet 10.0.0.0 netmask 255.255.255.0 {
  range 10.0.0.3 10.0.0.3;
  deny unknown-clients;
  host vxWorks_pentium_target {
    hardware ethernet 00:00:e8:01:02:03;
  }
  option routers 10.0.0.1;
  option broadcast-address 10.0.0.255;
  filename "/bootrom.pxe";
```

```
  default-lease-time 600;
  max-lease-time 7200;
  next-server "current address server";
}
```

Note that the next-server feature is very important and for some dhcpd release must be set with the IP address of the current TFTP server.

Start the DHCP server with the following command, as superuser:

```
# dhcpd -cf /path/to/dhcpd.conf
```

Next, edit the /etc/inetd.conf file to enable the TFTP service, and insure inetd is started. The enable inetd, edit the /etc/rc.conf file and add a line that says **inetd_enable="YES"** then run the following commands:

```
# sh /etc/rc.d/inetd stop
# sh /etc/rc.d/inetd start
```

Finally, create a /tftpboot directory, and place a copy of bootrom.pxe and bootapp.sys loader here. Note that the name bootrom.pxe and bootapp.sys must be kept.

Check out the BIOS requirement -> PXE Boot in this document to setup the board in PXE mode.

Once this setup has been done, reboot the PXE client system. It should display an output similar to the following:

```
CLIENT MAC ADDR: 00 00 E8 01 02 03  GUID: xxxxxxxx xxxx xxxx xxxx xxxxxxxxxxxx
CLIENT IP: 10.0.0.3  MASK: 255.255.255.0  DHCP IP: 10.0.0.1
GATEWAY IP: 10.0.0.1
TFTP...
```

Right after this output appears, the VxWorks bootrom should start up and the bootrom startup banner should be displayed. At this point, you can enter the desired VxWorks boot parameters and use the bootrom to load a VxWorks image.

Important notes:

- PXE is a 16-bit real mode environment. Consequently, boot applications downloaded via PXE are limited in that they must be able to start up in 16-bit real mode, and they must be less than 640K in size. The VxWorks bootrom.bin image meets these requirements, however standalone vxWorks images might not.

- The PXE environment also provides a universal API for accessing the underlying ethernet device, however VxWorks has no support for this feature. This means that while you can download the VxWorks bootrom via any PXE-enabled ethernet device, a native driver for the ethernet device must be available in VxWorks in order for the bootrom to subsequently download a VxWorks image (or for the downloaded VxWorks image to be able to use the ethernet port).

- The bootrom will be able to save the bootline parameters into VX304x onboard EEPROM

4.2 Make bootrom for bootable disk (USB, SATA)

This section explains how to build a bootrom image including USB,PATA and/or SATA boot features.

If you want to boot vxWorks image from **SATA disk**, enable **INCLUDE_DRV_STORAGE_INTEL_AHCI** in **config.h**, then change **DEFAULT_BOOT_LINE** to **DEFAULT_BOOT_LINE_SATA**. Under linux command lin execute.

If you want to boot vxWorks image from **USB** key, the bootrom must be build with the command line (for example under Linux).
Open a VxWorks Development Shell, in the BSP directory "$(**WIND_BASE**)/target/config/vx304x", run the command:

```
'cd $WIND_HOME/vxworks-6.9/target/config/vx304x_64'
'cd ../../proj;vxprj create -force -profile PROFILE_BOOTAPP_BASIC -lp64 vx304x_64 gnu vx304
'cd vx304x_64_gnu-bootapp'
'vxprj build set default_romCompress'
'vxprj bundle remove BUNDLE_MSB_WARM_REBOOT'
'vxprj bundle add BUNDLE_MSB_FAST_REBOOT'
'vxprj bundle add BUNDLE_VX304x_BOOTAPP'
'vxprj buildmacro set ROM_SIZE 00200000'
'vxprj buildmacro set ROM_TEXT_ADRS 00408000'
'vxprj buildmacro remove LIBS'
'vxprj buildmacro set LIBS "$WIND_HOME/vxworks-6.9/target/config/vx304x_64/libklibs_lp64_sm
'vxprj component add DRV_VX_CPLD'
'vxprj component add DRV_VX_I2C_EEPROM '
'vxprj component add INCLUDE_DOSFS'
'vxprj component add INCLUDE_DRV_STORAGE_INTEL_AHCI'
'vxprj component add INCLUDE_BOOT_FILESYSTEMS '
'vxprj component add INCLUDE_USB'
'vxprj component add INCLUDE_USB_INIT'
'vxprj component add INCLUDE_EHCI'
'vxprj component add INCLUDE_EHCI_INIT'
'vxprj component add INCLUDE_USB_MS_BULKONLY'
'vxprj component add INCLUDE_USB_MS_BULKONLY_INIT'
'vxprj component add INCLUDE_BOOT_USB_FS_LOADER'
'vxprj build vxWorks_romCompress.bin'
'vxprj build vxStage1Boot.pxe'
```

With this bootrom configuration, the boot will support etherent device, the USB and the SATA interface.

4.3 Install and boot with MKBOOTAHCI vxWorks command on bootable SATA disk

1) Boot vxWorks kernel including mkbootAhci

mkbootAhci command is available in the vxWorks BSP of VX304x. This command can be used to install a bootloader on SATA disk directly from VxWorks.
To do this, you must first build a bootrom correctly configured (see above) and the corresponding standalone kernel (make vxWorks.st). You must first boot this kernel by the way you prefer (PXE bootrom or USB bootrom). Once this kernel is

booted then, you should have access to the target SATA disk device.

2) Format the disk and install boot loader under vxWorks Shell command

Execute the following command to prepare the bootable disk
(for example, with a SATA device and 4 partitions created, the SATA formating can also be done by calling **usrCreatePartitions( )** routine from BSP file **partition.c**. This will have the same effect as the following
See also **createPart( )** into **partition.c** for an example on how to prepare a boot on SATA disk.

```
-> xbdCreatePartition "/ahci00:1", 4, 33, 33, 33
        Instantiating /ahci00 as rawFs, device = 0x1
        Instantiating /ahci00:1 as rawFs,  device = 0xc0001
        Instantiating /ahci00:2 as rawFs,  device = 0xd0001
        Instantiating /ahci00:3 as rawFs,  device = 0xe0001
        Instantiating /ahci00:4 as rawFs,  device = 0xf0001
        value = 0 = 0x0

-> dosFsVolFormat "/ahci00:1",0x02,0
        Formatting /ahci00:1 for DOSFS
   " ...
        OK.
        value = 0 = 0x0

-> dosFsVolFormat "/ahci00:2",0,0
        Formatting /ahci00:2 for DOSFS
        " ...
        OK.
        value = 0 = 0x0

-> dosFsVolFormat "/ahci00:3",0,0
        Formatting /ahci00:3 for DOSFS
     " ...
        OK.
        value = 0 = 0x0

-> dosFsVolFormat "/ahci00:4",0,0
     " ...
        OK.
        value = 0 = 0x0

-> copy "bootrom.bin","/ahci00:1/bootrom.sys"
        value = 0 = 0x0

-> mkMbrAhci 0,0
        value = 0 = 0x0

-> mkbootAhci 0,0,"bootrom.bin"
        num bytes written = 500728
        value = 0 = 0x0

-> copy ("bootapp.sys","/ahci00:1/bootapp.sys")
        value = 0 = 0x0

-> copy "vxWorksSMP","/ahci00:1/vxWorksSMP"
        value = 0 = 0x0

Then reboot and Select SATA device from the BIOS Setup (Enter F2 to Setup)

V1.6++++++++++
IA64 FSB V1.0
```

```
BOOTAPP SYS.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................

                    VxWorks System Boot


Copyright 1984-2013  Wind River Systems, Inc.




CPU: Intel(R) Nehalem Processor SYMMETRIC IO Kontron VX304x
Version: VxWorks 6.9 64-bit
BSP version: 6.9/xxxxx_UD
Creation date: Oct  1 2013 18:25:01

Press any key to stop auto-boot...
1

[VxWorks Boot]: c

'.' = clear field;  '-' = go to previous field;  ^D = quit

boot device          : gei0 fs
processor number     : 0
host name            : pcbist2
file name            : /home/vxworks/vxWorks /ahci00:1/vxWorksSMP
inet on ethernet (e) : 172.20.161.75:0xffffff00
inet on backplane (b):
host inet (h)        : 172.20.161.147
gateway inet (g)     :
user (u)             : vxworks
ftp password (pw) (blank = use rsh):
flags (f)            : 0x8
target name (tn)     :
startup script (s)   :
other (o)            : gei0

[VxWorks Boot]: @

boot device          : fs
unit number          : 0
processor number     : 0
host name            : pcbist2
file name            : /ahci00:1/vxWorksSMP
inet on ethernet (e) : 172.20.161.75:0xffffff00
host inet (h)        : 172.20.161.147
user (u)             : vxworks
flags (f)            : 0x8
other (o)            : gei0

Loading /ahci00:1/vxWorksSMP...7238880
Starting at 0xffffffff80408000...

Target Name: vxTarget

Adding 6624 symbols for standalone.
```

```
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
      ]]]]]]]]]]]  ]]]]     ]]]]]]]]]]       ]]              ]]]]         (R)
 ]     ]]]]]]]]]  ]]]]]]     ]]]]]]]]       ]]              ]]]]
 ]]     ]]]]]]]  ]]]]]]]]     ]]]]]] ]     ]]              ]]]]
 ]]]     ]]]]] ]    ]]]  ]     ]]]] ]]]   ]]]]]]]]]  ]]]] ]] ]]]]  ]]   ]]]]]
 ]]]]     ]]]  ]]    ]  ]]]     ]] ]]]]] ]]]]]]   ]] ]]]]]]] ]]]] ]]   ]]]]
 ]]]]]     ]  ]]]]     ]]]]]      ]]]]]]]] ]]]]   ]] ]]]]    ]]]]]]]    ]]]]
 ]]]]]]      ]]]]]     ]]]]]]    ]  ]]]]]  ]]]]   ]] ]]]]    ]]]]]]]]    ]]]]
 ]]]]]]]    ]]]]]  ]    ]]]]]]  ]    ]]]   ]]]]   ]] ]]]]    ]]]] ]]]]    ]]]]
 ]]]]]]]]  ]]]]]  ]]]    ]]]]]]]      ]     ]]]]]]]  ]]]]    ]]]]  ]]]] ]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       Development System
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks 6.9 SMP 64-bit
 ]]]]]]]]]]]]]]]]]]]]]]]]]]       KERNEL: WIND version 2.13
 ]]]]]]]]]]]]]]]]]]]]]]]]]       Copyright Wind River Systems, Inc., 1984-2013

 CPU: Intel(R) Nehalem Processor SYMMETRIC IO SMP Kontron VX304x.  Processor #0.
 Memory Size: 0x7ff00000 (2047Mb).  BSP version 6.9/2/xxxxx_UD.
 Created: Oct  1 2013 08:37:57
 ED&R Policy Mode: Deployed
 WDB Comm Type: WDB_COMM_END
 WDB: Agent Disabled.

-> devs
drv name
  0 /null
  1 /tyCo/0
  1 /tyCo/1
  7 /romfs
  3 /ahci00:1
  3 /ahci00:2
  9 pcbist2:
 10 /vio
 11 /tgtsvr
value = 25 = 0x19
-> ll "/ahci00:1"

Listing Directory /ahci00:1:
-rwxrwxrwx  1 0       0               5136 Jan  1  1980 bootrom.sys
-rwxrwxrwx  1 0       0             995392 Jan  1  1980 bootapp.sys
-rwxrwxrwx  1 0       0            7715969 Jan  1  1980 vxWorksSMP
-rwxrwxrwx  1 0       0            7640392 Jan  1  1980 vxWorksUP
value = 0 = 0x0
->
```

4.4 Install and boot with MKBOOTUSB vxWorks command on bootable USB disk

1) Boot vxWorks kernel including mkMbrUsb and mkbootUsb

mkMbrUsb and mkbootUsb commands are available in the vxWorks BSP of VX304x if **INCLUDE_USB** is defined. These commands can be used to install a bootloader on USB disk directly from VxWorks. To do this, you must first build a bootrom correctly configured (see above, do not forget

BOOTROM_FULL_FEATURE=y) and the corresponding standalone kernel vxWorks. You must first boot t
kernel by the way you prefer (PXE bootrom or SATA bootrom). Once this kernel is
booted then, you should have access to the target USB disk device.

2) Format the disk and install boot loader under vxWorks Shell command

Execute the following command to prepare the bootable disk
See also **createPartUSB( )** into **partition.c** for an example on how to prepare a boot on USB disk.

```
-> xbdCreatePartition ("/bd0", 2, 94, 0, 0)
Instantiating /bd0 as rawFs, device = 0x40001
Instantiating /bd0 as rawFs,  device = 0x90001
Instantiating /bd0:2 as rawFs,  device = 0xa0001
value = 0 = 0x0
-> dosFsVolFormat ("/bd0",0x12,0)
Formatting /bd0 for DOSFS
Instantiating /bd0 as rawFs, device = 0x90001
Formatting...Retrieved old volume params with %38 confidence:
Volume Parameters: FAT type: FAT32, sectors per cluster 0
  0 FAT copies, 0 clusters, 0 sectors per FAT
  Sectors reserved 0, hidden 0, FAT sectors 0
  Root dir entries 0, sysId (null)  , serial number e870000
  Label:"           " ...
Disk with 912681 sectors of 512 bytes will be formatted with:
Volume Parameters: FAT type: FAT16, sectors per cluster 16
2 FAT copies, 57012 clusters, 223 sectors per FAT
Sectors reserved 1, hidden 63, FAT sectors 446
Root dir entries 512, sysId VXDOS16 , serial number e870000
Label:"           " ...
OK.
value = 0 = 0x0
-> copy ("bootrom.bin","/bd0/bootrom.sys")
value = 0 = 0x0
-> mkMbrUsb("/bd0")
Instantiating /bd0 as rawFs, device = 0x40001
value = 0 = 0x0
-> Instantiating /bd0:2 as rawFs,  device = 0xd0001
mkbootUsb ("/bd0","bootrom.bin")
Instantiating /bd0 as rawFs, device = 0x40001
Instantiating /bd0:2 as rawFs,  device = 0x100001
num bytes written = 576352
value = 0 = 0x0
-> copy ("bootapp.sys","/bd0/bootapp.sys")
value = 0 = 0x0
-> copy ("vxWorksSMP","/bd0/vxWorksSMP")
value = 0 = 0x0


Then reboot and Select USB device (do not use UEFI !) from the BIOS Setup (Enter F2 to Setu

V1.6++++++++++
IA64 FSB V1.0
BOOTAPP SYS.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................

                    VxWorks System Boot
```

```
        Copyright 1984-2013  Wind River Systems, Inc.




        CPU: Intel(R) Nehalem Processor SYMMETRIC IO Kontron VX304x
        Version: VxWorks 6.9 64-bit
        BSP version: 6.9/xxxxx_UD
        Creation date: Oct  1 2013 18:25:01

        Press any key to stop auto-boot...
        1

        [VxWorks Boot]: c
'.' = clear field;  '-' = go to previous field;  ^D = quit

        boot device          : gei0 fs
        processor number     : 0
        host name            : pcsstr2
        file name            : /tmp/vxWorks.st /bd0/vxWorksSMP
        inet on ethernet (e) : 172.20.161.25:0xffffff00
        inet on backplane (b):
        host inet (h)        : 172.20.161.184
        gateway inet (g)     :
        user (u)             : vxworks
        ftp password (pw) (blank = use rsh): vxworks
        flags (f)            : 0x8
        target name (tn)     : vx304x
        startup script (s)   :
        other (o)            : gei0

        [VxWorks Boot]: @
        boot device          : fs
        unit number          : 0
        processor number     : 0
        host name            : pcsstr2
        file name            : /bd0/vxWorksSMP
        inet on ethernet (e) : 172.20.161.25:0xffffff00
        host inet (h)        : 172.20.161.184
        user (u)             : vxworks
        ftp password (pw)    : vxworks
        flags (f)            : 0x8
        target name (tn)     : vx304x
        other (o)            : gei0

        2908640 + 700916
        Starting at 0x408000...


        Adding 9221 symbols for standalone.

         ]]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks 6.9 SMP 64-bit
         ]]]]]]]]]]]]]]]]]]]]]]]]]]]       KERNEL: WIND version 2.13
         ]]]]]]]]]]]]]]]]]]]]]]]]]]]       Copyright Wind River Systems, Inc., 1984-2013


         CPU: Intel(R) Nehalem Processor SYMMETRIC IO SMP Kontron VX304x.  Processor #0.
         Memory Size: 0x7ff00000 (2047Mb).  BSP version 6.9/2/xxxxx_UD.
         Created: Oct  1 2013 08:37:57
         ED&R Policy Mode: Deployed
```

```
 WDB Comm Type: WDB_COMM_END
 WDB: Agent Disabled.

-> devs
drv name
  0 /null
  1 /tyCo/0
  1 /tyCo/1
  7 /romfs
  3 /ahci00:1
  3 /ahci00:2
  9 pcbist2:
 10 /vio
 11 /tgtsvr
value = 25 = 0x19
-> ll "/bd0"

Listing Directory /ahci00:1:
-rwxrwxrwx  1 0       0               5136 Jan  1  1980 bootrom.sys
-rwxrwxrwx  1 0       0             995392 Jan  1  1980 bootapp.sys
-rwxrwxrwx  1 0       0            7715969 Jan  1  1980 vxWorksSMP
-rwxrwxrwx  1 0       0            7640392 Jan  1  1980 vxWorksUP
value = 0 = 0x0
->


-> devs
drv name
  0 /null
  1 /tyCo/0
  1 /tyCo/1
  3 /ahci00:1
  3 /ahci00:2
  9 pcsstr2:
 10 /vio
 11 /tgtsvr
  3 /bd0
  4 /bd0:2
value = 25 = 0x19
->
-> ll "/bd0"

Listing Directory /bd0:
-rwxrwxrwx  1 0       0             576352 Jan  1  1980 bootrom.sys
-rwxrwxrwx  1 0       0            3911408 Jan  1  1980 vxWorks
value = 0 = 0x0
->
```

Due a limitation execute reboot(2) instead of **reboot( )** to reboot on USB device bootrom or see work-around in **Known Problem** chapter.

HARDWARE FEATURES SUPPORTED BY VX304x BSP

4.5 List of the hardware features supported by this BSP

This section gives the hardware features list supported onto the board.

Hardware Interface	Driver/Component	Status	Tested
DRAM	1GB/2GB modif. in **conf.h**	SUPPORTED	Yes Only 2GB available
DRAM	3GB	SUPPORTED	No
UART:0	**vxbNs16550Sio.c**	SUPPORTED	Yes (RS232 only)
UART:1	**vxbNs16550Sio.c**	SUPPORTED	No (RS232 only)
1GB-ETHERNET:0	**gei825xxVxbEnd.c**	SUPPORTED	Yes
DUAL-1GB or 10G/1G-ETH:1,2		UNSUPPORTED	No
RTC (date&time access)	vxbMc146818RtcK .c	UNSUPPORTED	No
RTC (used as timer)	**vxbMc146818Rtc.c**	UNSUPPORTED	No
SMBUS RTC (Set date&time)	**vxSMBusRtc.c**	UNSUPPORTED	No
USB2-HOST	USB EHCI driver	SUPPORTED	Yes
USB1-HOST	USB UHCI driver	SUPPORTED	No
USB3-HOST	USB XHCI driver	UNSUPPORTED	No
SMBus	**vxbPchSMBus.o**	UNSUPPORTED	No
CPLD Watchdog	VX CPLD	SUPPORTED	Yes
EEPROM	VXI2CEEPROM	SUPPORTED	Yes
FRAM	VXI2CEEPROM	UNSUPPORTED	No
CPLD LEDS	VX LEDS	UNSUPPORTED	No
CPLD GPIO	VX GPIO	UNSUPPORTED	No
Temp, Voltage Sensor	VX NCT7802Y	UNSUPPORTED	No
AUDIO	n/a	UNSUPPORTED	No
I2C Backplane num 0	VX CPLD	UNSUPPORTED	No
I2C Backplane num 1	VX CPLD	UNSUPPORTED	No
Boot on HDD SATA	SATA AHCI	SUPPORTED	Yes
Boot on SATA Flash module	SATA AHCI	SUPPORTED	Yes
Boot on USB Mass Storage	USB EHCI, XHCI	SUPPORTED	Yes
Boot on USB Flash module	USB EHCI, XHCI	SUPPORTED	Yes (Initialize USB disk under vxWorks)
SMB Alert	VX SMBALERT	UNSUPPORTED	No

4.6 Serial Configuration

SIO should be enabled by the BIOS baseboard. Default configuration is COM0:
- Baud Rate: 115200
- Data: 8 bit
- Parity: None
- Stop: 1 bit

Flow Control: None

4.7 Network Configuration

This BSP provides support for VxBus drivers, including network drivers. The list of VxBus network drivers available for PCIe devices includes the following:

```
INCLUDE_GEI825XX_VXB_END
INCLUDE_END
```

To permit command line builds and bootrom building , the **config.h** file must be modified to define the macros (as described above) corresponding to the desired driver.

The following END is supported by this BSP:

Driver	END	END2
GEI	**INCLUDE_GEI825XX_VXB_END**	UNSUPPORTED

5 KONTRON DEVICES DRIVERS

Following is a short description of the device drivers developped by Kontron and specific to VX304x like GPIO, EEPROM, Watchdog, etc.

5.1 Manage 3rd party drivers

VxBus Drivers developed by Kontron are located in the following directory :

```
$WIND_BASE/target/3rdparty/kontron_vx304x
```

It includes the following vxBus drivers

```
'vxCpld'      - CPLD access driver
'vxI2cEeprom' - EEPROM driver (2 EEPROM and 1 FMRAM towards I2C CPLD)
```

and an includes directory for *.h include files

For details on any of these drivers please refer to their own documentation located in $**WIND_HOME**/docs/extensions/eclipse/plugins/com.windriver.ide.doc.bsp_vxworks_6.9/vx304x_64

```
vxbVxCpldDrv.html
vxbVxI2cEepromDrv.html
```

These drivers are accessible by the user in the VX304x BSP through the two library files :

* **libkdrvs_lp64_smp.a** (SMP/UP mode)

These drivers do not implement standard I/O control entries but use a WindRiver Method called driverControl. This method allows the User API to access directly to the user driver routine with a structure of function pointers. For example the driver NCT7802Y implements this feature. Note that other drivers implement the same feature. This is the declaration of the method in the driver :

```
LOCAL device_method_t vxNCT7802YDrv_methods [] =
    {
    DEVMETHOD(driverControl, vxNCT7802YFuncGet),
    { 0, 0}
    };
```

The routine vxNCT7802YFuncGet will return to the user application a pointer to the following structure that can be retrieve in each .h driver file.

```
struct vxNCT7802YControl
    {
    /* backward reference to the device instance */
    VXB_DEVICE_ID pInst;

    /* read AD channel */
    STATUS (*chanRead)
        (
        VXB_DEVICE_ID pInst,    /* [in] device instance */
        UINT8 chan,             /* [in] channel # to read */
        UINT16 *pVal            /* [out] data */
```

```
    );

STATUS (*voltGet)
    (
            VXB_DEVICE_ID pInst,    /* [in] device instance */
            UINT8 chan,             /* [in] channel # to read */
            UINT8 **ppName,         /* [out] voltage name */
            UINT32 *pVal            /* [out] voltage value (in mV) */
    );

/* Get current temperature in degrees */
STATUS (*tempGet)
    (
    VXB_DEVICE_ID pInst,    /* [in] device instance */
            UINT8 chan,                             /* [in] channel # to read */
            UINT8 **ppName,                 /* [out] temperature name */
    signed short *pTemp                     /* [out] temperature */
    );

/* Read NCT7802Y register */
STATUS (*regRead)
    (
    VXB_DEVICE_ID pInst,    /* [in] device instance */
    UINT8 reg,              /* [in] register index */
    UINT8 *pVal             /* [out] register value */
    );

/* Write to NCT7802Y register*/
STATUS (*regSet)
    (
    VXB_DEVICE_ID pInst,    /* [in] device instance */
    UINT8 reg,              /* [in] register index */
    UINT8 val               /* [in] new register value */
    );
 };
```

To point to the correct vxBus Driver and retrieve the driverControl Method, we recommend to use the following routine:

```
void *vxbGetCtrlInfo(char *instName, int unit)
```

This routine is present in the VX304x BSP in the **bspUtil.c** file.

6 BSP INFORMATION

6.1 Identify BSP Release ID

To identify the version of this VX304x BSP see **KRELEASE** file in vx304x_64 BSP directory. Also check the ID that is displayed with BSP kernel is starting

```
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks 6.9 SMP 64-bit
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]      KERNEL: WIND version 2.13
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]      Copyright Wind River Systems, Inc., 1984-2013

CPU: Intel(R) Nehalem Processor SYMMETRIC IO SMP Kontron VX304x.  Processor #0.
Memory Size: 0x7ff00000 (2047Mb).  BSP version 6.9/2/xxxxx_UD.
Created: Oct  1 2013 08:37:57
ED&R Policy Mode: Deployed
WDB Comm Type: WDB_COMM_END
WDB: Agent Disabled.
```

The BSP command **verModulesShow** gives a complete board information with BSP ID, BIOS ID board Serial Number, Engineering change number, CPLD version ...

```
-> verModulesShow

---------- List of Module Versions ----------

VxWorks                 : 6.9 SMP 64-bit
BSP ID                  : 6.9/2/KRELEASE
BSP Creation            : Oct  1 2013 08:37:57

BIOS Vendor             : American Megatrends Inc.
BIOS ID                 : ID13274
BIOS Date               : 10/01/2013

BOARD Vendor            : Kontron
BOARD Version           : 20003A2_1041B00401008000
BOARD S/N               : 1813031160094

BOARD CPLD VER.         : 0x5

FPGA Firmware ID        : 0

SataFlash S/N           : (null)
SataFlash Model         : (null)

IPP Lib Version         : 7.0 (7.0.46)


value = 0 = 0x0
->
```

6.2 SMP mode

The BSP supports SMP (Symmetric Multiprocessing) mode. But **HyperThreading** must be disabled into CPU BIOS SETUP to support SMP mode.

6.3 Make Targets

The make targets are listed as the names of object-format files. Other images not listed here may not be tested.

bootrom
bootrom.bin
bootrom.pxe
vxWorks (with **vxWorks.sym**)
vxWorks.st

6.4 BSP Bootloaders and Bootroms

Bootloader/Bootrom	Status	Tested
BIOS (pre-flashed)	SUPPORTED	Tested
bootrom	SUPPORTED	Tested
bootrom_uncmp	UNSUPPORTED	Not Tested
vxWorks	SUPPORTED	Tested
vxWorks_rom	UNSUPPORTED	Not Tested
vxWorks_romCompress	SUPPORTED	Not Tested
vxWorks_romResident	UNSUPPORTED	Not Tested

The Kontron BSP provides drivers. These features can only be compiled under the command line and generate two library files that are present in the bsp folder of the vx304x:

libkdrvs_lp64_smp.a (SMP/UP mode) -> All drivers in UP and SMP mode

6.5 BSP Validated Tools

Tool	Connection/Type	Status
Workbench Debugger	ETHERNET/WDB	SUPPORTED
Workbench System Viewer	ETHERNET/WDB	SUPPORTED
Workbench MemScope	ETHERNET/WDB	UNVALIDATED
Workbench ProfileScope	ETHERNET/WDB	UNVALIDATED
Workbench StethoScope	ETHERNET/WDB	UNVALIDATED
Workbench CoverageScope	ETHERNET/WDB	UNVALIDATED
Workbench Core File Analysis	ETHERNET/WDB	UNVALIDATED
Workbench Sensor Points	ETHERNET/WDB	UNVALIDATED
Workbench OCD Debugger	JTAG	UNSUPPORTED
Workbench OCD Flash	JTAG	UNSUPPORTED

6.6 intelAhciDrv and intelAhciShow

To use this WindRiver SATA Hard disk driver as a boot device, the **INCLUDE_DRV_STORAGE_INTEL_AHCI** directive must be enabled in **config.h**. VxWorks image AHCI SATA support is also enabled by the'**INCLUDE_DRV_STORAGE_INTEL_AHCI**' component.

6.7 BSP Timer Routines

The BSP provides its own routines to help for time measurement.

gettimeofdayus : gives the current timestamp with microsecond precision
gettimeofdayns : gives the current timestamp with nansecond precision
gettimeofdayShow : shows the current timestamp in seconds and microseconds
gettimeofdaynsShow : shows the current timestamp in seconds and nanoseconds

To use these routines you must include the file **kontron.h** in your application. Then a time measurement can be done for example in microseconds with the following code:

```
#include <sys/times.h>
#include <kontron.h>

    struct timeval start;
    struct timeval end;
    int totalTime;

    gettimeofdayus(&start);

     ... What you need to measure ...

    gettimeofdayus(&end);

    totalTime = (int)(end.tv_sec*1000000LL+end.tv_usec - (start.tv_sec*1000000LL+start.tv_usec));
    printf ("Done in %d us\n", totalTime);
```

6.8 USB Host Support

A standard USB EHCI / USB UHCI controller is supported in this BSP. Please refer to Wind River's USB documents.

6.9 Boot Devices

The supported boot devices are:

gei - Primary Ethernet (10baseT, 100baseTX, or 1000baseT)
(need to be added in the configurator.
bootrom image must be less than 512KB)
fs - ATA/IDE drive (SATA, CF, PATA)
fs - USB disk

6.10 Boot Methods

The boot methods are affected by the boot parameters. If no password is specified, RSH (remote shell) protocol is used. If a password is specified, FTP protocol is used, or, if the flag is set to 0x80, TFTP protocol is used. Enter **h** command under vxWorks bootrom to have a flag description.

6.11 Software Board reset method

The standard WRS routine **reboot( )** or CTRL-X can be used to reboot on bootrom. **sysHardReset( )** command can be used to reboot the board and restart the board from BIOS. So the reset routines are

sysHardReset - reset board and restart BIOS (also reboot(2) do that)

7 KNOWN PROBLEMS

7.1 IPP Performance

The IPP performance is reduced in Kernel Mode comparing to RTP. Should be fixed in the next Update Pack 3 Service Pack 3 of WindRiver.

The libs in RTP mode and Kernel Mode are different. In that way this is a basic bench in RTP that use the IPP libs for FFT calculation on one Thread (@2100MHz):

```
[vxWorks *]# ./rtpTestIpp.vxe 0
Launching process './rtpTestIpp.vxe' ...
Process './rtpTestIpp.vxe' (process Id = 0xffff800000131cb0) launched.
IPP : Static Init Done
Average time - gflops for FFT 64 points : 155.5 ns - 12.4 gflops
Average time - gflops for FFT 128 points : 332.4 ns - 13.5 gflops
Average time - gflops for FFT 256 points : 797.4 ns - 12.8 gflops
Average time - gflops for FFT 512 points : 1630.4 ns - 14.1 gflops
Average time - gflops for FFT 1024 points : 3605.5 ns - 14.2 gflops
Average time - gflops for FFT 2048 points : 8941.6 ns - 12.6 gflops
Average time - gflops for FFT 4096 points : 20065.3 ns - 12.2 gflops
Average time - gflops for FFT 8192 points : 44062.6 ns - 12.1 gflops
```

In Kernel mode the same test (libs are different from the RTP) :

```
-> benchfftipp 0,0,0
IPP : Static Init Done
Average time - gflops for FFT 64 points : 544.3 ns - 3.5 gflops
Average time - gflops for FFT 128 points : 1236.0 ns - 3.6 gflops
Average time - gflops for FFT 256 points : 2850.1 ns - 3.6 gflops
Average time - gflops for FFT 512 points : 6377.5 ns - 3.6 gflops
Average time - gflops for FFT 1024 points : 14170.0 ns - 3.6 gflops
Average time - gflops for FFT 2048 points : 32323.5 ns - 3.5 gflops
Average time - gflops for FFT 4096 points : 73092.9 ns - 3.4 gflops
Average time - gflops for FFT 8192 points : 175781.3 ns - 3.0 gflops
value = 0 = 0x0
->
```

This difference is explained by the fact that IPP libs were not compiled with the SSE2 instruction. Futhermore these performance in RTP can be increased if WindRiver provides libraries compiled with AVX instruction. A set of this libraries are available at Kontron but the linkage of the code failed due to some linker issue in 64 bits only.

7.2 USER_RESERVED_MEM limited to 384 Mbytes

As a result of the previous limitation the maximum **USER_RESERVED_MEM** size is limited to 384 Mbytes.

7.3 Reboot

CTRL-X command does not work. Do reboot the machine type the command **sysHardReset**

7.4 Boot on USB device (bootrom + kernel) and Ethernet problem

When booting with bootrom + kernel on USB device and network init is set (like **other: gei0**) in bootrom parameters then a long boot time (more than one minute) occurs if the network is not connected or if the network is connected with no traffic coming. The same boot time delay will occur at symbol table downloading if the kernel symbol table is downloaded from USB. The workaround is to start by linking the symbol table to the kernel. For the bootrom, the correction consists in initializing USB and having the USB device mounted before the network initialization. This necessitates a modification in WindRiver source file **all/usrConfig.c**. Please contact kontron support at **support-kom-sa@kontron.com** for further details.

7.5 Legacy interrupt on PCI

Due to a BIOS limitation PCI/PCIe device IRQ could not be supported in Legacy interrupt mode. It is advised to use PCIe devices in MSI (Message Signaled Interrupt) mode. To use PCI devices on PMC in legacy interrupt mode please contact kontron support at **support-kom-sa@kontron.com**

7.6 BSP and vxBus driver documentation not linked to WRS documentation

When BSP is installed the VX304x BSP and vxBus driver documentation is not linked to the WRS documentation. It would have been necessary to erase a WRS html file for this. The BSP documentation will be find at
$WIND_HOME/docs/extensions/eclipse/plugins/com.windriver.ide.doc.bsp_vxworks_6.9/vx304x

8 BIBLIOGRAPHY

Intel Architecture Software Developer's Manual Volume 1 Basic Architecture

Intel Architecture Software Developer's Manual Volume 2 Instruction Set Reference

Intel Architecture Software Developer's Manual Volume 3 System Programming Guide

AP-485, Intel Processor Identification and the CPUID Instruction

PCI System Architecture Fourth Edition Addison-Wesley 1999 ISBN 0-201-30974-2

9 SEE ALSO

VxWorks User's Guide Getting Started, *VxWorks Programmer's Guide Configuration*, *VxWorks Programmer's Guide Architecture Supplement*, *vxworks Application Programmers Guide 6.9*

MAILING ADDRESS
Kontron Modular Computers S.A.S.
150 rue Marcelin Berthelot - BP 244
ZI TOULON EST
83078 TOULON CEDEX - France

TELEPHONE AND E-MAIL
+33 (0) 4 98 16 34 00
Sales: Order-MAR-Toulon@kontron.com
Support: GSS-MAR-Toulon@kontron.com

For further information about other Kontron products, please visit our Internet web site: www.kontron.com.