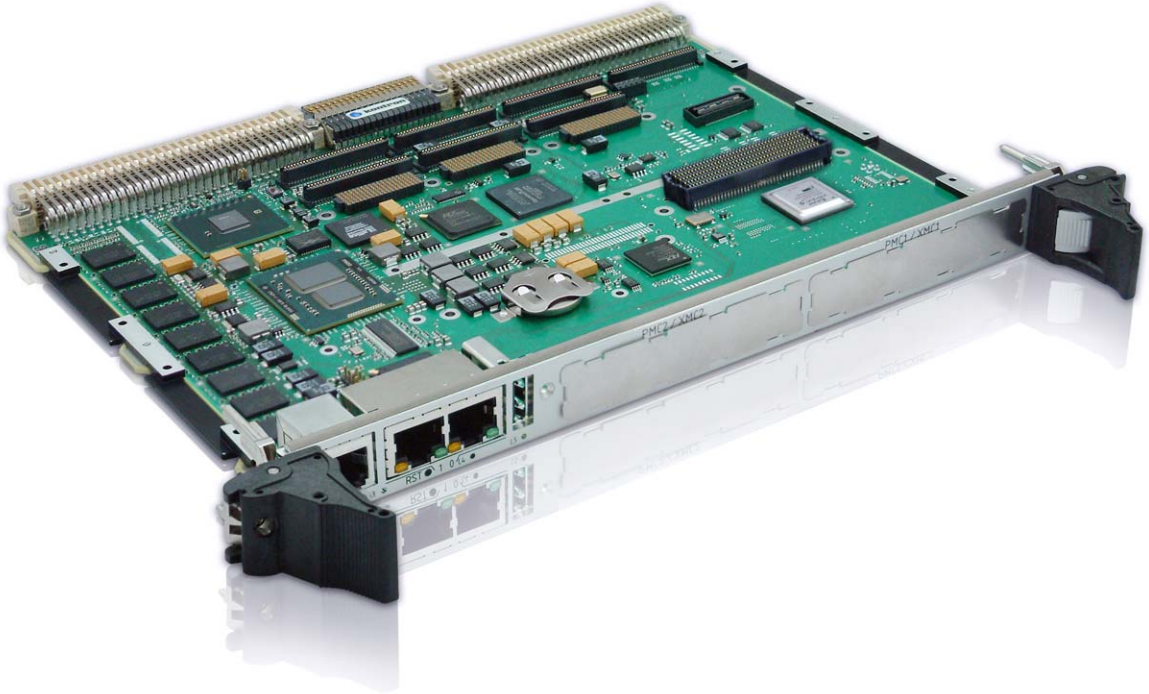
kontron

» VM6050 «



Release Notes VxWorks 6.9 on VM6050

SD.DT.G44-1e - February 2015

Revision History

Publication Title:	Release Notes VxWorks 6.9 on VM6050	
Doc. ID:	SD.DT.G44-1e	
Rev.	Brief Description of Changes	Date of Issue
1e	Improvement of compilation kernel feature description.	02-2015
0e	Initial Issue	12-2014

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.

Table of Contents

Table of Contents

1 INTRODUCTION

This reference entry provides board-specific information necessary to run VxWorks 6.9 for the Kontron VM6050 BSP. Before using a board with VxWorks, verify that the board runs and check the RS-232 connection with the correct settings as described in the following chapters. Refer to VM6050 6U VME SBC User's Guide for a detailed description of the VM6050. Note that this release was installed on a Fedora 16 64 bits Linux Machine. All compilation and tests are done in this environment. This BSP supports UP (Unique Processor) and SMP (Symetric Multi Processor) mode. This BSP supports only the 32bits version.

The document contains the following first level chapters:

INTRODUCTION
 - VxWorks REQUIREMENTS
 - BIOS REQUIREMENT
 - JUMPERS & SWITCHS
HARDWARE FEATURES SUPPORTED BY VM6050 BSP
BSP INSTALLATION GUIDE
BOOTING VXWORKS ON VM6050
KONTRON DEVICES DRIVERS
BSP INFORMATION
KNOWN ISSUES
BSP EVOLUTIONS
BIBLIOGRAPHY
SEE ALSO

1.1 VxWorks/Workbench Requirements

VxWorks 6.9 updated to 6.9.4.2 (Update Pack 4 Service Pack 2)
Workbench 3.3.5.2

1.2 BIOS Requirements

The BIOS ID of the VM6050 must be greater or equal to ID14153.

First of all restore the default values. To do that enter in the BIOS by pressing F2 or DEL during the boot process of the VM6050.

* F3 -> Optimized Defaults

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

* Chipset -> South Bridge Configuration -> SATA Configuration -> SATA Controler(s) [Enabled]
* Chipset -> South Bridge Configuration -> SATA Configuration -> SATA Mode Selection; [AHCI]
* Advanced -> CPU Configuration -> Hyper-Threading [Disabled]
* Kontron -> CPU Configuration [Select the maximum CPU frequency or the CPU frequency desired] or let the default value : 2000Mhz
* Advanced -> CPU Configuration -> Power & Performance -> Intel(R) SpeedStep -> [Disabled]
* Advanced -> CPU Configuration -> Power & Performance -> Extreme Edition -> [Disabled]
* Advanced -> CPU Configuration -> Power & Performance -> C states -> [Disabled]

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

* Save & exit -> Save changes and Reset

For PXE boot:

* Chipset -> South Bridge Configuration -> Ibexpeak options -> PXE ROM: [Enabled]

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

Note that this BIOS settings is the default configuration supported by Kontron.

1.3 Jumpers & Switches

Please refer to your particular board vendor's documentation.

2 HARDWARE FEATURES SUPPORTED BY VM6050 BSP

2.1 List of Hardware Features

Hardware Interface	Driver/Component	Status	Tested
DRAM	3.5GB max (auto detect)	SUPPORTED	Yes
UART:0	**vxbNs16550Sio.c**	SUPPORTED	Yes (RS232 only)
UART:1	**vxbNs16550Sio.c**	SUPPORTED	Yes (RS232 only)
1GB-ETHERNET:0,1,2,3	**gei825xxVxbEnd.c**	SUPPORTED	Yes
SATA	**vxbIntelAhciStorage.c**	SUPPORTED	Yes (AHCI Only)
GRAPHICS	**vxbM6845Vga.c**	SUPPORTED	Yes (PC Console Only)
APIC-TIMER	**vxbLoApicTimer.c**	SUPPORTED	Yes
TIMESTAMP	**vxbIntelTimestamp.c**	SUPPORTED	Yes
RTC	**vxbMc146818Rtc.c**		
	vxbMc146818RtcK.c	SUPPORTED	Yes
USB2-HOST: 3 interfaces	USB EHCI driver	SUPPORTED	Yes (bd0,bd5,bd10)
USB1-HOST	USB UHCI driver	SUPPORTED	No
SMBus	**vxbPchSMBus.o**	SUPPORTED	Yes
CPLD Watchdog	VX CPLD	SUPPORTED	Yes
EEPROM	VXI2CEEPROM	SUPPORTED	Yes
CPL LEDS	VX LEDS	SUPPORTED	Yes (4 User leds)
CPLD GPIO	VX GPIO	SUPPORTED	Yes (IO and Interrupt)
Temp Sensor	3 LM73	SUPPORTED	Yes
Voltage Sensor	ADS73xx	SUPPORTED	Yes
FNVRAM		SUPPORTED	Yes
AUDIO	n/a	UNSUPPORTED	No
VME ALMA Feature	**vxbAlma.c**	SUPPORTED	Yes
Boot on PXE ethernet	GEI	SUPPORTED	Yes
Boot on HDD SATA	SATA AHCI	SUPPORTED	Yes
Boot on SATA Flash module	SATA AHCI	SUPPORTED	Yes
Boot on USB Mass Storage	USB EHCI	SUPPORTED	Yes (8GB with 512B/sect)
Boot on USB Flash module	USB EHCI	SUPPORTED	Yes (Initialize USB disk under vxWorks) (8GB with 512B/sect)
XMC/PMC 32bits slot 1		SUPPORTED	Yes (XMC only)
XMC/PMC 64bits slot 2		SUPPORTED	Yes (XMC only)
SMB Alert	VX SMBALERT	SUPPORTED	No

3 BSP INSTALLATION GUIDE

3.1 Install vxWorks 6.9 / Workbench 3.3

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

This selection is mandatory to be in the same software configuration than Kontron developers.

3.2 Install VxWorks 6.9.4.2

DVD-R147826.1-21-00 : VxWorks 6.9.4.2

The Update Pack 4 Service pack 2 vxWorks 6.9 from WindRiver is present in this DVDROM in the following folder :

UpdatePack

Uncompress the archive file delivered that corresponds to **DVD-R147826.1-21-00**. Go to the **DVD-R147826.1-21-00** 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 4 service pack 2 release

3.3 Install Workbench Update Pack

DVD-R158451.1-1-28 : Workbench 3.3.5.2

Uncompress the archive file delivered by WindRiver that corresponds to DVD-R158451.1-1-28. (see the WINDRIVER support web site : support.windriver.com, do not download or unzip it in install/updates directory !)

Go to the **DVD-R158451.1-1-28** 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 3.3.5.2 update pack release.

The Update Pack Workbench 3.3.5.2 from WindRiver is present in the DVDROM named **Workbench-3.3.5.2 Update pack VM6050** delivered by Kontron

3.4 Install the VM6050 BSP

The BSP of the VM6050 is located in **VXW69-BSP-VM6050-xxxxx** folder. Where xxxxx identify the BSP ID.

Files and folders present at the top of the VXW69-BSP-VM6050-xxxxx folder :

bootrom : folder that contains bootrom and multistaging bootstrap binaries
UpdatePack : folder that contains the vxWorks 6.9 Update Pack 4 Service Pack 2
vm6050.pdf : Describes the BSP release for vxWorks 6.9 32 bits for vm6050
SDDTG00-1e.pdf : Describes the VxBus ALMA VME driver. VxWorks 6.8/6.9 ALMA VME Bus Driver
vximages : folder that contains UP/SMP gnu/icc 32 bits vxWorks images
VXW69-BSP : contains the vxWorks BSP files and docs
VXW69-BSP-VM6050-xxxxxx.zip: BSP zip file ID xxxxxx to be used for installation

Files contained in the **bootrom** folder :

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

Files contained in the **vximages** folder :

```
vximages
   |-- SMP
   |   |-- gnu
   |   |   |-- vxWorks : Vxworks 32 bits image UP compiled with gnu
   |   |   |             and including code to initialize a ETHERNET,SATA or USB devices
   |   |-- icc
   |       \-- vxWorks : Vxworks 32 bits image UP compiled with icc
   |                     and including code to initialize a ETHERNET,SATA or USB devices
   \-- UP
       |-- gnu
```

```
|   |-- vxWorks : Vxworks 32 bits image SMP compiled with gnu
|   |             and including code to initialize a ETHERNET,SATA or USB devices
\-- icc
    \-- vxWorks : Vxworks 32 bits image SMP compiled with icc
                  and including code to initialize a ETHERNET,SATA or USB device
```

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-VM6050-xxxxx.zip** in the home vxWorks 6.9 directory as follows :

under **linux OS** :

```
cd $WIND_HOME
unzip /mnt/cdrom/VXW69-BSP-VM6050-xxxxx.zip -d .
```

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

unzip the D:\VXW69-BSP-VM6050-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.

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

-- Under **Windows OS** --

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

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

3.6 Compile VM6050 kernel

The installation procedure is ended. The VM6050 kernel can then be compiled by command or the Workbench Project can be launched to configure and compile a new image kernel.

To compile the kernel UP/SMP mode, make sure the environment variables are activate in your VIP project :

```
DRV_VXBEND2_TEI8259X (in order to use 10 Gigabit interface)
DRV_TIMER_MC146818
INCLUDE_DOSFS
INCLUDE_FS_MONITOR
INCLUDE_SHELL
INCLUDE_SHELL_STARTUP
INCLUDE_DRV_STORAGE_INTEL_AHCI
INCLUDE_SHELL_INTERP_CMD
INCLUDE_SHELL_INTERP_C
INCLUDE_SHELL_EMACS_MODE
INCLUDE_VXBUS_SHOW
INCLUDE_SHOW_ROUTINES
INCLUDE_SHELL_BANNER
INCLUDE_DISK_UTIL
INCLUDE_DISK_UTIL_SHELL_CMD
INCLUDE_USB
INCLUDE_USB_INIT
INCLUDE_EHCI
INCLUDE_EHCI_INIT
INCLUDE_USB_MS_BULKONLY
INCLUDE_USB_MS_BULKONLY_INIT
INCLUDE_RTP
INCLUDE_ROMFS
INCLUDE_LOGGING
INCLUDE_EXC_HANDLING
INCLUDE_EXC_TASK
INCLUDE_SYM_TBL_SHOW
INCLUDE_DEBUG
INCLUDE_RLOGIN
INCLUDE_NET_HOST_SHOW
INCLUDE_IFCONFIG
INCLUDE_INETLIB
INCLUDE_NETSTAT
INCLUDE_ROUTECMD
INCLUDE_ROUTE
INCLUDE_NET_BOOT
INCLUDE_NET_BOOT_CONFIG
INCLUDE_NET_HOST_SETUP
INCLUDE_NETMASK_GET
INCLUDE_SOCKLIB
INCLUDE_IPWRAP_GETIFADDRS
INCLUDE_MUX
INCLUDE_NETBUFPOOL
INCLUDE_LINKBUFPOOL
INCLUDE_IPCOM
INCLUDE_IPNET_LOOPBACK_CONFIG
INCLUDE_IPNET_USE_ROUTESOCK
INCLUDE_IPNET_USE_SOCK_COMPAT
INCLUDE_IPNET_SOCKET
INCLUDE_IPATTACH
INCLUDE_LOADER
```

```
INCLUDE_UNLOADER
INCLUDE_PING
INCLUDE_WDB
INCLUDE_WDB_ALWAYS_ENABLED
INCLUDE_EDR_ERRLOG
INCLUDE_EDR_PM
INCLUDE_EDR_SHOW
INCLUDE_EDR_SYSDBG_FLAG
INCLUDE_EDR_SHELL_CMD
INCLUDE_SPY
INCLUDE_MEMDRV
INCLUDE_RTP
INCLUDE_ROMFS
INCLUDE_SHARED_DATA
INCLUDE_POSIX_PTHREADS
INCLUDE_GETADDRINFO
INCLUDE_POSIX_TIMERS
INCLUDE_CORE_DUMP
INCLUDE_CORE_DUMP_COMPRESS
INCLUDE_CORE_DUMP_SHOW
INCLUDE_CORE_DUMP_MEM
INCLUDE_TLS
INCLUDE_TASK_CREATE_HOOKS
INCLUDE_TIMESTAMP
INCLUDE_TIMER_SYS_SHOW

BULK_MAX_DEVS 3 (for USB Bulk devices)
ISR_STACK_SIZE 4000
IPNET_MEMORY_LIMIT 33554432
IPNET_SOCK_DEFAULT_SEND_BUFSIZE "65536"
INET_ICMP_RATE_LIMIT_BUCKET_SIZE "1000000"
INET_ICMP_RATE_LIMIT_INTERVAL "1000"
CORE_DUMP_CKSUM_ENABLE TRUE
PM_RESERVED_MEM (15*1024*1024)
EDR_ERRLOG_SIZE (32 * VM_PAGE_SIZE)
```

These are the default Kontron settings used for testing. These includes and parameters are provided in the **uConfig.h** but not included by default in the **config.h**. The buildK.sh script shell includes by default all includes and defines described in this chapter.

3.7 Compile VM6050 bootrom

To compile the bootrom, make sure the environment variables are activate in your bootrom project :

```
DRV_VXBEND2_TEI8259X (in order to use 10 Gigabit interface)
DRV_VX_CPLD
DRV_VX_I2C_EEPROM
INCLUDE_DOSFS
INCLUDE_DRV_STORAGE_INTEL_AHCI
INCLUDE_BOOT_FILESYSTEMS
INCLUDE_USB
INCLUDE_USB_INIT
INCLUDE_EHCI
INCLUDE_EHCI_INIT
INCLUDE_USB_MS_BULKONLY
INCLUDE_USB_MS_BULKONLY_INIT
INCLUDE_BOOT_USB_FS_LOADER

BULK_MAX_DEVS 3 (for USB Bulk devices)
```

All the build feature must follow these parameters set by the **vxprj** command in the project folder:

```
vxprj build set default_romCompress
vxprj bundle remove BUNDLE_MSB_WARM_REBOOT
vxprj bundle add BUNDLE_MSB_FAST_REBOOT
vxprj buildmacro set ROM_SIZE 00200000
vxprj buildmacro set ROM_TEXT_ADRS 00408000
```

Similar options can be set in the workbench environment. Follow the Workbench user guide to do so.

In addition, in the vm6050 bsp a script is provided by Kontron named buildB.sh. This shell script (for unix platform only) can build a bootrom image by calling the vxprj command. This is an example by command line to do the build:

```
cd $WIND_HOME
./wrenv.linux -p vxworks-6.9
cd $WIND_HOME/vxworks-6.9/target/config/vm6050
./buildB.sh -c gnu
```

4 BOOTING VXWORKS ON VM6050

First of all to have access to the board settings please connect a serial line to the VM6050. The serial line is configured by default to 115200 bauds, 8bits, 1bit stop, no parity, no flow control.

In order to explain the boot strap code ordering during the boot on legacy boot device, let's have a look to the different files generated during the bootrom project. This is the files generated in the bootrom project:

```
vxStage1Boot.bin        : bootstrap Kontron image for SATA/USB
vxStage1Boot.pxe        : bootstrap Kontron image for PXE boot
vxWorks_romCompress.bin : bootrom binary file
```

The **boot0.h** in the BSP contains the MBR (Master Boot Record + FreeBSD bootstrap). This file is used by the command **createPartFull** when the partition is created. This command is described in the following chapters.

Nevertheless the files generated in the bootrom project must be renamed as follow :

```
For SATA/USB:

  vxStage1Boot.bin         -> bootrom.bin
  vxWorks_romCompress.bin  -> bootapp.sys

  The renamed files are copied to the disk by the command 'createPartFull'.

For PXE:
  vxStage1Boot.pxe         -> bootrom.pxe
  vxWorks_romCompress.bin  -> bootapp.sys

  The renamed files are copied to the TFTP server folder by the user.
```

This is below the explanation of the boot strap sequence for booting from SATA or USB device, once the device is configured.

```
 SATA/USB (bootrom.bin & bootapp.sys are located in the first DOS partition):

   BIOS Legacy boot on SATA/USB -> MBR on SATA/USB (from boot0.h) -> bootrom.bin -> bootapp.sys

 PXE (bootrom.pxe & bootapp.sys are located to the TFTP server folder):

   BIOS legacy boot -> bootrom.pxe -> bootapp.sys
```

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 VM6050 onboard EEPROM

4.2 USB/SATA Initialization Libraries present in the BSP

The BSP provides to the user the ability to initialize USB and/or SATA devices to be able to boot on from a legacy boot. All necessary routines are present in the libs folder for sources, and in the **libklibs.a/libklibs_smp.a** files. The routine **createPartFull** is used to initialize all mass storage devices present on USB/SATA interfaces. This is the routine definition :

```
STATUS createPartFull(const char *bdev,const char *kfile, const char *bdir, int mode)
```

It uses four parameters. The first one is the device USB or SATA that we need to initialize.
For example "/bd0" or "/ahci00:1"
The second one is the kernel file to defaulty set in the first partition of the device.
For example "pcdev:/home/vxworks69/proj/ki7_gnu/default/vxWorks"
(through the network by a FTP access to the **pcdev** server)
or "/bd5/vxWorks" (stored in a USB key storage)
The third one is the folder where the bootrom files are present.
For example "pcdev:/home/vxworks69/proj/ki7_gnu-bootapp/default_romCompress"

(through the network by a FTP access to the **pcdev** server).
This folder might content : vxStage1Boot.bin, vxWorks_romCompress.bin files
The forth one is the mode USB/SATA/AUTO. 0 automatically uses the USB or SATA.

4.3 Install and boot vxWorks on bootable SATA disk

Here is below an initialization of a sata disk named **/ahci00:1** through the vxWorks shell command:

```
        -> createPartFull "/ahci00:1","/home/vxworks69/proj/ki7_ahci00_gnu/default/vxWorks",
"/home/vxworks69/proj/ki7_gnu-bootapp/default_romCompress",0
        * Create Partition 1 ->
        Instantiating /ahci00:1 as rawFs,  device = 0x150001
        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 b360000
          Label:"           " ...
        Disk with 2055879 sectors of 512 bytes will be formatted with:
        Volume Parameters: FAT type: FAT16, sectors per cluster 32
          2 FAT copies, 64229 clusters, 251 sectors per FAT
          Sectors reserved 1, hidden 63, FAT sectors 502
          Root dir entries 512, sysId VXDOS16 , serial number b360000
          Label:"           " ...
        OK.
        Done

        * Create Partition 2 -> Formatting /ahci00:2 for DOSFS
        Instantiating /ahci00:2 as rawFs, device = 0x160001
        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 b430000
            Label:"           " ...
        Disk with 27267786 sectors of 512 bytes will be formatted with:
        Volume Parameters: FAT type: FAT32, sectors per cluster 16
          2 FAT copies, 1702570 clusters, 13315 sectors per FAT
          Sectors reserved 32, hidden 63, FAT sectors 26630
          Root dir entries 0, sysId VX5DOS32, serial number b430000
          Label:"           " ...
        OK.
        Done
        * Copy bootrom file ->
/home/vxworks69/proj/ki7_gnu-bootapp/default_romCompress/vxStage1Boot.bin -> /ahci00:1/bootrom.sys
        Done
        * Create MBR -> Done
        * Create Boot file -> num bytes written = 5136
        Done
        * Copy Bootrom -> Done
        * Copy Kernel -> Done
        value = 0 = 0x0
```

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

```
        V1.6++++++++++++
        IA32 FSB V1.0
        BOOTAPP SYS.....................................................................
        .....................................................................
```

```
        ........................................................................
        ........................................................................
        ........................................................................

                            VxWorks System Boot


        Copyright 1984-2014  Wind River Systems, Inc.




        CPU: SYMMETRIC IO Kontron VM6050
        Version: VxWorks 6.9
        BSP version: 6.9/14349_EFT
        Creation date: Dec 15 2014 16:43:36

        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 /ahci00:1/vxWorks
        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)     : vm6050
        startup script (s)   :
        other (o)            :

        [VxWorks Boot]: @
        boot device          : fs
        unit number          : 0
        processor number     : 0
        host name            : pcsstr2
        file name            : /ahci00:1/vxWorks
        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)     : vm6050
        other (o)            :

        Loading /ahci00:1/vxWorks...5143632
        Starting at 0x408000...


        Adding 9094 symbols for standalone.

         ]]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks 6.9 SMP
         ]]]]]]]]]]]]]]]]]]]]]]]]]]]       KERNEL: WIND version 2.13
         ]]]]]]]]]]]]]]]]]]]]]]]]]]]       Copyright Wind River Systems, Inc., 1984-2014
```

```
 CPU: SYMMETRIC IO Kontron VM6050.  Processor #0.
 Memory Size: 0xca16a000 (~3233Mb).  BSP version 6.9/14349_EFT.
 Created: Dec 15 2014 17:08:29
 ED&R Policy Mode: Deployed
 WDB Comm Type: WDB_COMM_END
 WDB: Agent configuration failed.

-> 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            7640392 Jan  1  1980 vxWorks
value = 0 = 0x0
->
```

4.4 Install and boot vxWorks on bootable USB disk

Here is below an initialization of a sata disk named **/bd5** through the vxWorks shell command:

```
        -> createPartFull "/bd5","/home/vxworks69/proj/ki7_bd5_gnu/default/vxWorks",
"/home/vxworks69/proj/ki7_gnu-bootapp/default_romCompress",0
        * Create Partition Entries -> Instantiating /bd5 as rawFs, device = 0xc0001
        Instantiating /bd5 as rawFs,  device = 0x200001
        Instantiating /bd5:2 as rawFs,  device = 0x210001
        Done

        * Create Partition 1 -> Formatting /bd5 for DOSFS
        Instantiating /bd5 as rawFs, device = 0x200001
        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 14140000
          Label:"           " ...
        Disk with 922068 sectors of 512 bytes will be formatted with:
        Volume Parameters: FAT type: FAT16, sectors per cluster 16
          2 FAT copies, 57598 clusters, 226 sectors per FAT
          Sectors reserved 1, hidden 63, FAT sectors 452
          Root dir entries 512, sysId VXDOS16 , serial number 14140000
          Label:"           " ...
        OK.
        Done

        * Create Partition 2 -> Formatting /bd5:2 for DOSFS
        Instantiating /bd5:2 as rawFs, device = 0x210001
```

```
        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 142b0000
          Label:"           " ...
        Disk with 14437836 sectors of 512 bytes will be formatted with:
        Volume Parameters: FAT type: FAT32, sectors per cluster 8
          2 FAT copies, 1801200 clusters, 14100 sectors per FAT
          Sectors reserved 32, hidden 63, FAT sectors 28200
          Root dir entries 0, sysId VX5DOS32, serial number 142b0000
          Label:"           " ...
        OK.
        Done
        * Copy bootrom file ->
/home/vxworks69/proj/ki7_gnu-bootapp/default_romCompress/vxStage1Boot.bin -> /bd5/bootrom.sys
        Done
        * Create MBR -> Instantiating /bd5 as rawFs, device = 0xc0001
        Done
        * Create Boot file -> Instantiating /bd5 as rawFs, device = 0xc0001
        num bytes written = 5136
        Done
        * Copy Bootrom -> Done
        * Copy Kernel -> Done
        value = 0 = 0x0
```

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

```
        V1.6++++++++++
        IA32 FSB V1.0
        BOOTAPP SYS.....................................................................
        .....................................................................
        .....................................................................
        .....................................................................
        .....................................................................

                            VxWorks System Boot


        Copyright 1984-2014  Wind River Systems, Inc.




        CPU: SYMMETRIC IO Kontron VM6050
        Version: VxWorks 6.9
        BSP version: 2.0/14349_EFT
        Creation date: Dec 15 2014 16:43:36

        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 /bd5/vxWorks
        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)     : vm6050
startup script (s)   :
other (o)            :

[VxWorks Boot]: @
boot device          : fs
unit number          : 0
processor number     : 0
host name            : pcsstr2
file name            : /bd5/vxWorks
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)     : vm6050
other (o)            :

Loading /bd5/vxWorks...5143632
Starting at 0x408000...


Adding 9094 symbols for standalone.

 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks 6.9 SMP
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]      KERNEL: WIND version 2.13
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]     Copyright Wind River Systems, Inc., 1984-2014


 CPU: SYMMETRIC IO Kontron VM6050.  Processor #0.
 Memory Size: 0xca16a000 (~3233Mb).  BSP version 6.9/14349_EFT.
 Created: Dec 15 2014 17:08:29
 ED&R Policy Mode: Deployed
 WDB Comm Type: WDB_COMM_END
 WDB: Agent configuration failed.

-> devs
drv name
  0 /null
  1 /tyCo/0
  1 /tyCo/1
  7 /romfs
  3 /bd5
  3 /bd5:2
  9 pcbist2:
 10 /vio
 11 /tgtsvr
value = 25 = 0x19
-> ll "/bd5"

Listing Directory /bd5:
-rwxrwxrwx  1 0       0               5136 Jan  1  1980 bootrom.sys
-rwxrwxrwx  1 0       0             995392 Jan  1  1980 bootapp.sys
-rwxrwxrwx  1 0       0            7640392 Jan  1  1980 vxWorks
value = 0 = 0x0
->
```

5 KONTRON DEVICES DRIVERS

This section describes all features of the board, supported or not. It documents all configurations of the board and the interaction between features and configuration items.

5.1 Feature Interactions

Refer to:

Kontron VM6050 6U VME SBC User's Guide CA.DT.A93 Kontron VM6050 Hardware Release Notes CA.DT.A94

5.2 Serial Configuration

No on-chip serial interface, used baseboard serial chip. SIO should be enabled on baseboard. Default configuration is:
Baud Rate: 115200
Data: 8 bit
Parity: None
Stop: 1 bit
Flow Control: None

5.3 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
```

For command line builds and for building bootrom, the **config.h** file must be modified to define above macros, corresponding to the desired driver.

The following vxBus driver familly END only is supported in this BSP:

Driver	END	END2
GEI	**INCLUDE_GEI825XX_VXB_END**	UNSUPPORTED

5.4 Manage 3rd party driver

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

```
$WIND_BASE/target/3rdparty/kontron
```

It includes the following drivers
ads7830 - Voltage sensors driver

rtc146818 - RTC driver
vxCpld - CPLD driver
vxI2cEeprom - EEPROM driver (2 EEPROM towards I2C CPLD)
vxLm73 - LM73 3 temperature sensors driver (towards I2C CPLD)
pchsmbus - SMBUS driver (used for voltage sensors)
vme - ALMA VME Bus driver
vxGPIO - GPIO driver (towards CPLD)
vxLED - LED driver (towards 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/vm6050_6.9

vxbVxCpldDrv.html vxbAds7830.html vxbVxI2cEepromDrv.html vxbAlma.html vxGPIO.html vxbLm73.html vxLED.html vxbMC146818RtcK.html vxbPchSMBus.html

These drivers are available to the user in the VM6050 BSP through the two library files :

* **libkontron.a** (UP mode)
* **libkontron_smp.a** (SMP mode)

These drivers do not implement standard io control entries but use a WindRiver Method called driverControl. This method give the ablility to the User API to access directly to the user driver routine with a structure of function pointers. For example the driver ads7830 implement this feature. Note that other drivers implement the same thing. This is the declaration of the method in the driver :

```
LOCAL device_method_t vxbAds7830Drv_methods[] =
    {
    DEVMETHOD(driverControl, vxbAds7830FuncGet),
    { 0, 0}
    };
```

The routine vxbAds7830FuncGet will return to the user application a pointer to the following structure that can be retrieve in each .h driver file.

```
struct vxbAds7830Control
    {
    VXB_DEVICE_ID pInst;

    STATUS (*chanRead)
        (
        VXB_DEVICE_ID pInst,
        UINT8 chan,
        UINT8 *pVal
        );

    STATUS (*voltGet)
        (
                VXB_DEVICE_ID pInst,
                UINT8 chan,
                UINT8 **ppName,
                UINT32 *pVal
        );
     };
```

To point to the correct vxBus Driver and retreive the driverControl method we recommend to use the following routine:

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

This routine is present in the VM6050 BSP in the **libs/bspUtil.c** file.

To summarize how the user can use the driverControl Method the vm6050 BSP contains example like the following one that is already present in the BSP and allow the user to get the temperature of the board. This code is present in the BSP in the file name **libs/tempGet.c** :

```
#define NB_SENSORS      3
LOCAL BOOL drvLm73CtrlStatus[NB_SENSORS] = {FALSE};

LOCAL struct vxbLm73Control *drvLm73Ctrl[NB_SENSORS];

STATUS tempGet(char *name,int drvNum,float *temp)
{
        STATUS status;

        if(drvLm73CtrlStatus == FALSE)
                drvLm73Ctrl = vxbGetCtrlInfo(name,drvNum);

        if(drvLm73Ctrl == NULL || drvLm73Ctrl->pInst == NULL)
        {
                return -1;
        }
        else
        {
                drvLm73CtrlStatus = TRUE;
        }

        status = drvLm73Ctrl->tempGet(drvLm73Ctrl->pInst,temp);

        return status;
}
```

5.5 Voltage Sensors

ADS7830 Voltage sensors are available behind the Board SMbus. In examples directory **voltage.c** gives an example on how to get and test voltage

```
-> ads7830Test
0: Getting driver functionality : OK
1: Reading registers : OK
2: Getting voltage values :
Channel 0: 12V VPX 11892
Channel 1: 5V VPX 4886
Channel 2: 3V3 3300
Channel 3: 2V5 6U 2498
Channel 4: 3V3 VPX SB 3222
Channel 5: 1V05S 3U 1025
Channel 6: 1V05 3U IBEX 1044
Channel 7: 1V 3U 976 OK
3: Checking voltage ranges :
Channel 0: 12V VPX 11892 [min=11500,max=12500]
Channel 1: 5V VPX 4886 [min=4500,max=5500]
Channel 2: 3V3 3300 [min=3000,max=3500]
```

```
Channel 3: 2V5 6U 2498 [min=2300,max=2800]
Channel 4: 3V3 VPX SB 3222 [min=3000,max=3500]
Channel 5: 1V05S 3U 1025 [min=1000,max=1100]
Channel 6: 1V05 3U IBEX 1044 [min=1000,max=1100]
Channel 7: 1V 3U 986 [min=950,max=1050] OK

TOTAL TESTS = 4, failed = 0

value = 0 = 0x0
->
```

5.6 RTC Use

Internal IBEXPEAK PCH RTC is used to manage the current date. The RTC in the IBEXPEAK is the MC146818. A vxBus driver was adapted to use it in RTC mode. The name of this vxBus driver is "mc146818RtcKontron". The standard "vxbRtcGet" and "vxbRtcSet" vxbus driver entries can be used. To use Kontron RTC vxBus driver for date and time in project then define the following include:

```
DRV_RTC_MC146818
```

See example directory to find example code for rtcGet and rtcSet routines

```
-> rtcGet
Year:12, Month:2, Day:21, WDay:2
Hour:13, Min:59, Sec:22
value = 0 = 0x0
->
```

5.7 CPLD Watchdog

CPLD Watchdog is available through CPLD register access. It uses vxCpld vxBus driver method. Watchdog can be used in interrupt, timer or reset mode. An example for using vxCpld and watchdog is given in example directory with file **cpldWdt.c**.

```
-> cpldWdt
0: Getting functionality : OK
1: Getting CPLD parameters : OK
Cpld version : 0xb (1)
Cpld geo id : 0x1d (1)
2: Checking CPLD parameters : OK
3: Activate watchdog (10 sec)  : OK
4: Trigger watchdog in 9 sec  : OK
5: Disable the watchdog in 10 sec : OK
6: Reset the controller (controller must be reset in 5 sec)  :
TOTAL TESTS = 6, FAILED = 0

value = 0 = 0x0
->

Then board Reset
```

5.8 Board Information (VPD)

Board information or Vital Product Data (VPD) like Serial Number can be accessed from VPD EEPROM. If necessary user can include and linked application located in $**WIND_BASE**/target/3rdparty/kontron/vxI2cEeprom/ with 3 files

```
vx6060_vpd.c
vxvariant.c
vxvpdtool.c
```

VPD can be read with **vpddisplay( )**. This routine are included into the EEPROM vxBus but not load by default into the kernel. To include the command **vpddisplay( )** you can invokes it somewhere into the code. For example into

```
extern STATUS vpddisplay(int dispflags);
void usrAppInit (void)
    {
        vpddisplay(0);
#ifdef  USER_APPL_INIT
        USER_APPL_INIT;
#endif
```

Then when the new kernel is compiled, it will include this command:

```
-> vpddisplay(0)
Board type   : VM6050-2SA34-00110
EC Level     : 02034
Serial Number: 1811271020018
Variant      : 1000004180840000
Keylist      :
/PCB_B/SACLASS/P2GPIOOFF/BHQUAD/IOFPGAOFF/IBOMOFF/PCIEMUXGEN2/NOJTAGPCH/XMC/COREI7LVK0/PWRMAGOFF/
BATON/STD_EARTH/IRTC/XDPON/STDCLK/CK505REFOSC/XMCPWR12VOFF/2GB_DDR3_1333/P0PWRMAGOFF/P0UHM/SATAHDD/
2RANK/VME/PXMC/1SLOT/FP422ON/FL/P80ON/ITIN/I2CSTD/P5VOFF/NOFPIO/PMCON/JTAGON/


value = 0 = 0x0
->
```

5.9 EEPROMs/FMRAM Accesses

Two EEPROMs and one FMRAM are available under Kontron Core i7 boards.

One EEPROM is VPD reserved (Vital Product Data) and the second is User reserved. The User reserved EEPROM can so be used entirely by an application. Also the FMRAM is totaly free to be used. All these devices are available by means of vxBus vxI2CEeprom driver.

An example on how to access FMRAM is given with file **vxFramTool.c** with **target/3rdparty/kontron/vxI2cEeprom/vxFramTool.c** routines:

```
STATUS vxprintfmram(int offset, int size)
STATUS vxwritefmram(int offset, int size, UINT8 buffer[])
```

These routines are part of the vxBus library but not linked to the kernel by default. The user needs to invokes it somewhere into the code to include them.

This file example is easy to adapt for EEPROM. Instead of calling eeprom->fmramRead or write method user can call other appriopriate method like ->userRead or ->userWrite to have access to the User EEPROM etc. See vxI2cEeprom driver documentation itself for a complete vxBus method.

Example of use, dummp FMRAM from offset 0 size 40 bytes

```
-> vxprintfmram(0,40)
Raw FMRAM data:
0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0
0x 0 0x 0 0x 0 0x 0 0x20 0x20 0x20 0x38 0x34 0x36 0x39 0x33 0x30 0x38 0x38 0x36
0x20 0x20 0x20 0x20 0x20 0x20 0x31 0x36
value = 0 = 0x0
->
```

5.10 CPLD Temperature Sensors

There are 3 LM73 temperature sensors available onto VM6050 (with I2C CPLD). They can simply be read with the command **tempGetShow** included into the BSP in **tempGet.c** file.

A list of all sensors available can be accessed by the following commands :

```
void listTempSensors(void)

-> listTempSensors
TEMP 0 : lm73 , chan = 0
TEMP 1 : lm73 , chan = 0
TEMP 2 : lm73 , chan = 0
value = 0 = 0x0
->
```

tempGetShow must be invoked with the LM73 sensor number ID as single parameter.

```
STATUS tempGetShow(int num);

-> tempGetShow(0)
Temp = 51.50
value = 0 = 0x0
-> tempGetShow(1)
Temp = 50.00
value = 0 = 0x0
-> tempGetShow(2)
Temp = 32.50
value = 0 = 0x0
```

5.11 CPLD GPIO

There is 6 GPIOs available on VM6050. They are managed with vxBus driver vxGPIO. See **vxbGPIO.h** file to find all GPIO mode that are programmables. See **vxbGPIO.c** for all vxBus method associated with GPIO

GPIO programmable vxBus method and modes can be :

- GPIO getDirection/setDirection (**GPIO_DIRECTION_IN**, **GPIO_DIRECTION_OUT**)

- GPIO getValue/setValue (**GPIO_VALUE_LO**, **GPIO_VALUE_HI**)

- GPIO getIntMode/setIntMode: level sensitive or edge (**GPIO_INTMODE_LEVEL**, **GPIO_INTMODE_EDGE**)

- GPIO getIntEnabled/setIntEnabled

- GPIO getIntPolarity/setIntPolarity (**GPIO_POLARITY_LO**, **GPIO_POLARITY_HI**)

- GPIO getIntToggle/setIntToggle (**GPIO_TOGGLE_OFF**,**GPIO_TOGGLE_ON**)

When a GPIO is selected in Toggle Mode GPIO, the interrupt Mode and polarity are not taken in consideration. If enabled then Interruption will be generated if GPIO changes of state.

A test example is given into example directory file **gpioTest.c**. The test can work only if GPIO are connected two by two but it will demonstrates how to program them.

5.12 CPLD LEDs

Five front side LEDs are available on VM6052/VM6054. Only four front side user LEDs can be controlled by vxLED vxBus driver. Leds color and blink mode can be set.See **vxLED.h** for possibles mode. See **led.c** file in example directory for a LEDs test demonstration.

LEDs not allowed : 0 User LEDs number available on VM6052/VM6052 : 1,2,3,4

Note that the software LED number are from 0 to 4. Hardware LED number documented start from 1 to 5. So Software LED number 0 correspond to Hardware LED number 1.

5.13 VME Driver

VxBus vxbAlma provides all VME accesses features. See ALMA VME Bus driver pdf documentation in 3rdparty/kontron/vxbAlma for general information on ALMA driver use and for some VME concepts and advises. Also see vxBus ALMA routines documentation in source files or in html mode in directory $**WIND_HOME**/docs/extensions/eclipse/plugins/com.windriver.ide.doc.bsp/vm6050_6.9/ files **vxbAlma.html**, **vxbAlmaDma.html**, **vxbAlmaExcep.html**, **vxbAlmaSem.html**

The documentation referenced by SDDTG00-1e.pdf is present in this release. All description is for vxWorks 6.8/6.9.

Some VME examples are given in BSP examples directory.

In file **exampleVMETest.c**:

```
createMasterVMEWindows() allows to create a MASTER VME windows to access VME bus
createSlaveVMEWindows()  allows to create a slave VME windows mapping a local buffer
                         to VME bus
```

```
exampleVmeTest()          a program test between two board.
```

In file **interruptVme.c**: gives an exmaples for VME interrupt use (ping pong test)

In file **mailboxVme.c** : gives an examples for addressed mailbox interrupt use

5.14 Get PBIT Result

If available PBIT (POwer-On Built In Tests) results are read from EEPROM (if PBIT have been ordered with the BIOS). They can be read with command **pbitdisplay( )** available into the BSP. See file **vxpost.c** in 3rdParty

```
-> pbitdisplay
POSTs configured to run from command line:
       mem_data: PASSED
       mem_addr: PASSED
   mem_pattern1: PASSED
   mem_pattern2: PASSED
   mem_pattern3: PASSED
   mem_pattern4: PASSED
         serial: PASSED
            rtc: PASSED
       sysflash: PASSED
           cpld: PASSED
   temp_sensors: PASSED
    temperature: PASSED
         fnvram: PASSED
    ether_loop0: PASSED
    ether_loop1: PASSED
    ether_loop2: PASSED
    ether_loop3: PASSED
        voltage: PASSED
 sata_controler: PASSED
            vpd: PASSED
         eeprom: PASSED
      vme_check: PASSED
 pmcA_xmc_check: PASSED
 pmcB_xmc_check: PASSED
usb1_controller: PASSED
usb2_controller: PASSED
         system: PASSED

PASSED  : 27
FAILED  : 0
NOT RUN : 0
TOTAL   : 27

value = 0 = 0x0
->
```

5.15 PC_CONSOLE

This function will use PC monitor (VGA) as VxWorks console. To use these controllers the **INCLUDE_PC_CONSOLE** directive must be enabled in **config.h**. PS2 and USB keyboards are supported.

NOTE: there is some problems with USB UHCI stack when enabled both in bootloader and image so it is recommended to use it either in image or in bootloader.
UHCI will be enabled in bootloader if both **INCLUDE_USB** and **INCLUDE_PC_CONSOLE** are defined in **config.h**.

5.16 intelAhciDrv and intelAhciShow

To use this driver as a boot device, the **INCLUDE_DRV_STORAGE_INTEL_AHCI** directive must be enabled in **config.h**, vxWorks image support is enabled by the **INCLUDE_DRV_STORAGE_INTEL_AHCI** component.

Furthermore, Kontron can delivered onboard or on a mezannine card, SATA Flash device from the GREENLIANT brand. To be compliant with the SATA Flash specification the SATA link must be limited to SATA gen1 speed (means : 1.5Gbps). In order to be able to configure the limitation of the SATA interface speed, the following structure is present in the BSP in the **hwconf.c** file as shown below :

```
#if defined (INCLUDE_DRV_STORAGE_INTEL_AHCI)

/* This array contains couple of two parameters.
 * The first one correspond to the physical sata port.
 * The second one correspond to the limit of the sata interface generation.
 * The allowed parameters for the second parameter is :
 *            0: No sata bus limitation
 *            1: Limit the sata bus speed at gen1 (1.5Gbps)
 *            2: Limit the sata bus speed at gen2 (3Gbps)
 *            3: Limit the sata bus speed at gen3 (6Gbps)

AHCI_LIST_LIMIT limitPortArray[] = {
#if defined(VM6054)
  { 3,1 }, { 4,1 }
#elif defined(VM6050)
  { 3,1 }
#else
  { 3,0 }
#endif
};
```

So to remove the sata speed limitation, the second parameter of the couple parameter, must be set at 0.

5.17 DRV_SMBUS_INTEL_PCH

To use onboard sensors the Intel PCH SMBus controller driver must be included, this driver is provided by Kontron as 3rdparty driver.
BSP also provides common KEAPI functions to access the SMBus.

5.18 DRV_RTC_MC146818 - MC146818 RTC clock driver.

This is the vxBus compliant driver which implements the functionality specific to the the MC146818 real time clock Timer. It doesn't implement functionality of standard MC146818 timer driver but adds RTC time functions

User application is able to use VxBus custom methods defined for this driver:

```
METHOD_DECL(vxbRtcGet)
METHOD_DECL(vxbRtcSet)
```

and the methods should be called as

```
struct tm t;

vxbDevMethodRun(DEVMETHOD_CALL(vxbRtcGet), &t);
vxbDevMethodRun(DEVMETHOD_CALL(vxbRtcSet), &t);
```

5.19 vxbIntelTimestamp

This is the driver for the timestamp on Intel chipsets to use the timestamp feature, the macro **INCLUDE_TIMESTAMP** must be defined in **config.h**.

5.20 vxbLoApicTimer

This library contains routines to manipulate the timer functions on the Intel CorI7 family processor's Local APIC/xAPIC Timer with a board-independent interface.

The macro **APIC_TIMER_CLOCK_HZ** must also be defined to indicate the clock frequency of the Local APIC/xAPIC Timer.

5.21 USB Host Support

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

6 BSP INFORMATION

6.1 Identify BSP Release ID

To identify the version of this VM6050 BSP see **KRELEASE** file in vm6050 BSP directory. Also check the ID that is displayed with BSP kernel is starting

```
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks 6.9
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]      KERNEL: WIND version 2.13
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]     Copyright Wind River Systems, Inc., 1984-2014

 CPU: SYMMETRIC IO Kontron VM6050.  Processor #0.
 Memory Size: 0xca16a000 (~3233Mb).  BSP version 6.9/14349_EFT.
 Created: Dec 16 2014 09:46:23
 ED&R Policy Mode: Deployed
 WDB Comm Type: WDB_COMM_END
 WDB: Ready.

->
```

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 (6.9.4.2)
BSP ID                  : 6.9/14349_EFT
BSP Creation            : Dec 16 2014 09:46:23
BSP Type                : SYMMETRIC IO Kontron VM6050

BIOS Vendor             : American Megatrends Inc.
BIOS ID                 : ID14153
BIOS Date               : 06/02/2014

BOARD Vendor            : Kontron
BOARD Version           : 02019_1000004180850000
BOARD S/N               : 1814301821257
BOARD Order Code        : VM6050-2SA34-00110

BOARD CPLD VER.         : 0xe

FPGA Firmware ID        : 0

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

IPP Lib Version         : 7.0 (7.0.46)

VME Bridge Revision     : 45
VME Id                  : 0
VME Geo Id              : 4
Proc Num                : 0


value = 0 = 0x0
->
```

Furthermore, if the parameter of this routine becomes 2, the user can access the Kontron driver version. This version is given by a local Kontron CVS repository and tagged automatically in the code.

```
-> verModulesShow 2

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

VxWorks                 : 6.9 (6.9.4.2)
BSP ID                  : 6.9/14349_EFT
BSP Creation            : Dec 16 2014 09:46:23
BSP Type                : SYMMETRIC IO Kontron VM6050

BIOS Vendor             : American Megatrends Inc.
BIOS ID                 : ID14153
BIOS Date               : 06/02/2014

BOARD Vendor            : Kontron
BOARD Version           : 02019_1000004180850000
BOARD S/N               : 1814301821257
BOARD Order Code        : VM6050-2SA34-00110

BOARD CPLD VER.         : 0xe

FPGA Firmware ID        : 0

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

IPP Lib Version         : 7.0 (7.0.46)

VME Bridge Revision     : 45
VME Id                  : 0
VME Geo Id              : 4
Proc Num                : 0


Driver Versions:

-> driver alma (KBuild: Dec 16 2014 09:39:25):
        vxbAlma.c,v 1.6 2014/11/06 14:01:30
        vxbAlmaDma.c,v 1.2 2014/04/10 16:28:25
        vxbAlmaExcep.c,v 1.3 2014/06/16 16:05:58
        vxbAlmaSem.c,v 1.3 2014/06/16 16:05:58
        vxbAlma.h,v 1.2 2014/04/10 16:28:25
        vxbAlmaP.h,v 1.3 2014/04/10 16:32:48
-> driver intelAhciSata (KBuild: Dec 16 2014 09:39:26):
        vxbIntelAhciStorage.c,v 1.3 2014/07/04 14:43:38
        vxbIntelAhciStorage.h,v 1.1 2014/07/04 14:43:38
-> driver intelAhciSata (KBuild: Dec 16 2014 09:39:26):
        vxbIntelAhciStorage.c,v 1.3 2014/07/04 14:43:38
        vxbIntelAhciStorage.h,v 1.1 2014/07/04 14:43:38
-> driver vxSMBALERT (KBuild: Dec 16 2014 09:39:24):
        vxSMBALERT.c,v 1.6 2013/10/22 13:07:17
        vxSMBALERT.h,v 1.2 2013/10/22 13:07:17
-> driver lm73 (KBuild: Dec 16 2014 09:39:24):
        vxbLm73.c,v 1.10 2014/06/16 15:57:08
        vxbLm73.h,v 1.3 2013/10/22 12:52:12
-> driver vxLED (KBuild: Dec 16 2014 09:39:25):
        vxLED.c,v 1.3 2013/10/22 12:49:32
        vxLED.h,v 1.2 2013/10/22 12:49:32
-> driver vxI2CEeprom (KBuild: Dec 16 2014 09:39:23):
```

```
        vxbVxI2cEepromDrv.c,v 1.7 2014/06/16 15:40:19
        vxbI2cEepromDrv.h,v 1.2 2013/10/22 12:46:53
        vxbVxI2cEepromDrv.h,v 1.2 2013/10/22 12:46:53
        vxFramTool.c,v 1.2 2013/10/22 12:46:53
        vxpost.c,v 1.2 2013/10/22 12:46:53
        vxpost.h,v 1.2 2013/10/22 12:46:53
        vxvariant.c,v 1.2 2013/10/22 12:46:53
        vxvpdtool.c,v 1.2 2013/10/22 12:46:53
        vxvpdtool.h,v 1.2 2013/10/22 12:46:53
        vx6060_vpd.c,v 1.2 2013/10/22 12:46:53
-> driver vxGPIO (KBuild: Dec 16 2014 09:39:24):
        vxGPIO.c,v 1.7 2014/12/04 10:09:55
        vxGPIO.h,v 1.5 2014/12/03 17:02:06
-> driver vxCpld (KBuild: Dec 16 2014 09:39:23):
        vxbVxCpldDrv.c,v 1.17 2014/06/09 11:35:29
        vxbVxCpldDrv.h,v 1.4 2014/06/09 11:35:29
-> driver pchSMBus (KBuild: Dec 16 2014 09:39:22):
        vxbPchSMBus.c,v 1.4 2014/11/14 08:32:16
        vxbPchSMBus.h,v 1.3 2014/11/14 08:32:16
-> driver ads7830 (KBuild: Dec 16 2014 09:39:23):
        vxbAds7830.c,v 1.5 2013/10/22 09:00:14
        vxbAds7830.h,v 1.4 2013/10/22 09:00:14
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. Kontron advises the user to disable it. Due to some real-time experiences, the Hyperthreading can impact some real-time system approach. In consequence, Hyperthreading will not be supported.

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 Supported

vxWorks	SUPPORTED	Tested
vxWorks_rom	UNSUPPORTED	Not Supported
vxWorks_romCompress	SUPPORTED	Tested (Bootrom project only)
vxWorks_romResident	UNSUPPORTED	Not Supported

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 VM6050:

libkdrvs.a (UP mode) -> All drivers in UP
libkdrvs_smp.a (SMP mode) -> All drivers in SMP

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 Timestamp 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
ktimediffus : make a diff between two **gettimeofdayus** timestamp
ktimediffns : make a diff between two **gettimeofdayns** timestamp

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);
```

or using ktimediffus:

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

    struct timeval start;
    struct timeval end;
    long long diff;
    int ok;

    gettimeofdayns(&start);

    ... What you need to measure ...

    diff = ktimediffns(&start,&end,&ok);
```

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.

All tests were done with the following devices :

```
-> usbBulkShow
Node ID 0x00000002
  LUN #0
    Vendor Info            : Kingston
    Product ID             : DataTraveler 3.0
    Product Revision       : PMAP
    Number of Blocks       : 15360000
    Bytes per Block        : 512
    Total Capacity         : 7864320000

Node ID 0x00000102
  LUN #0
    Vendor Info            : Kingston
    Product ID             : DataTraveler 3.0
    Product Revision       : PMAP
    Number of Blocks       : 15360000
    Bytes per Block        : 512
    Total Capacity         : 7864320000

Node ID 0x00000103
  LUN #0
```

```
    Vendor Info             : SMART
    Product ID              : eUSB
    Product Revision        : 895D
    Number of Blocks        : 15728640
    Bytes per Block         : 512
    Total Capacity          : 8053063680

value = 0 = 0x0
->
```

6.9 Boot Devices

The supported boot devices are:

gei - Primary Ethernet (10baseT, 100baseTX, or 1000baseT)

fs - ATA drive (SATA)

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

reboot - jump on bootrom

sysHardReset - reset board and restart BIOS

sysVmeReset - reset board and restart BIOS and possibly reset entire VME rack depending on BIOS settings

6.12 Examples

In this BSP some source code examples are present in the folder **examples**. The library libkexamples can be generated by hand through the command line by invoking the following command at $**WIND_HOME**/vxworks-6.9/target/config/vm6050 :

```
make libkexample.obj
```

or can be managed directly to the workbench VIP.

7 KNOWN ISSUES

7.1 Hyperthreading

The Hyperthreading mode of the CoreI7 Processor is considered by the OSes (including vxWorks) as a core. And in fact is not. Some real-time system in SMP mode can have their system performance decreased when hyperthreading is enabled. In that way, Kontron considers that this feature must be deactivated by default.

7.2 PROFILE_SMALL_FOOTPRINT

This Workbench **PROFILE_SMALL_FOOTPRINT** profile for the current VM6050 BSP is not supported for now.

7.3 Diab compiler not supported

This current release was not compiled with diab, but only with **gnu compiler** gcc and icc **intel compiler**.

7.4 GPIO LEVEL mode

GPIO LEVEL mode is not supported. This is a on board CPLD limitation.

7.5 MSI Interrupt not supported by some PMC/XMC

Some Ethernet PCI-X PMC doesn't support MSI interrupt mode. A problem can occurs when plugged on PMC 64 bits slot (PMC2). In case of problem modify **hwconf.c** file and remove corresponding gei interface from the following table list

```
LOCAL const struct intrCtlrInputs loApicInputs[] = {
{ VXB_INTR_DYNAMIC, "yn", 0, 0 },
The following lines are for using Message Signaled Interrupts (MSI)
for GEI Ethernet driver instead of legacy interrupts
This is OK for on board device, for PCI-X PMC plugged on PMC2 slot
on VM6050 it could be necessary to remove it
{ VXB_INTR_DYNAMIC, "gei", 0, 0 },
{ VXB_INTR_DYNAMIC, "gei", 1, 0 },
{ VXB_INTR_DYNAMIC, "gei", 2, 0 },
{ VXB_INTR_DYNAMIC, "gei", 3, 0 },
};
```

7.6 Possible long boot delay when booting on USB device with separate symbol table found in vx6.8

This bug was found in vxWorks 6.8. Nevertheless for this release the test was not done. This is a WindRiver defect WIND00322792. When booting from usb device, with a separate symbol table and with a network to initialize then the boot will take 1minutes 30s. This occurs only if the ethernet is not connected or if there is no traffic on the ethernet. The workaround is to always link the symbole table to the kernel (**INCLUDE_STANDALONE_SYM_TBL** must be defined).

7.7 SMI Digital Thermal can imply a latency

In some cases the DTS (Digital Thermal Sensor) can imply a latency in the system depending on the real-time system stress. DTS event happens when temperature jump over BIOS defined temperature thresholds. The SMI (System Management Interrupt) managed for the DTS can be removed by using a BIOS configuration as described below :

Advanced->Thermal Configuration->CPU Thermal Configuration->DTS->Disabled

In addition to remove entirely the DTS event, the Kontron BSP routine **thermalInterruptAvoid**. This routine can be invoked in the user application. Beware about the fact that this routine, is running on all available cpus.

8 BSP EVOLUTIONS

8.1 ID14325_EFT

1. *First release vxWorks 6.9 32bits.*
   Tagged as EFT.

8.2 ID15036

1. *Remove the **uConfig.h** from the **config.h** bsp file.*

2. *Be able to compile the BSP by removing the **INCLUDE_VME_DRV**.*

3. *Manage the SATA Link speed limitation in the **hwconf.c** file.*
   This feature is need to support greenliant FDM SATA.

4. *Add code to remove the DTS events.*

5. *Update the document SD.DT.G00-0e concerning the VME driver.*

6. *the routine **mediaGet** supports the 10GB link speed status.*

9 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

Kontron VxWorks 6.8/6.9 ALMA VME Bus Driver SD.DT.G00

Kontron VM6050 6U VME SBC User's Guide CA.DT.A93

Kontron VM6050 Hardware Release Notes CA.DT.A94

Kontron VM6050 AMI-BIOS User Reference Manual SD.DT.F89

Kontron VM6050 PBIT User's Guide SD.DT.F88

10 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-ATD-Toulon@Kontron.com
Support: GSS-ATD-Toulon@Kontron.com

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