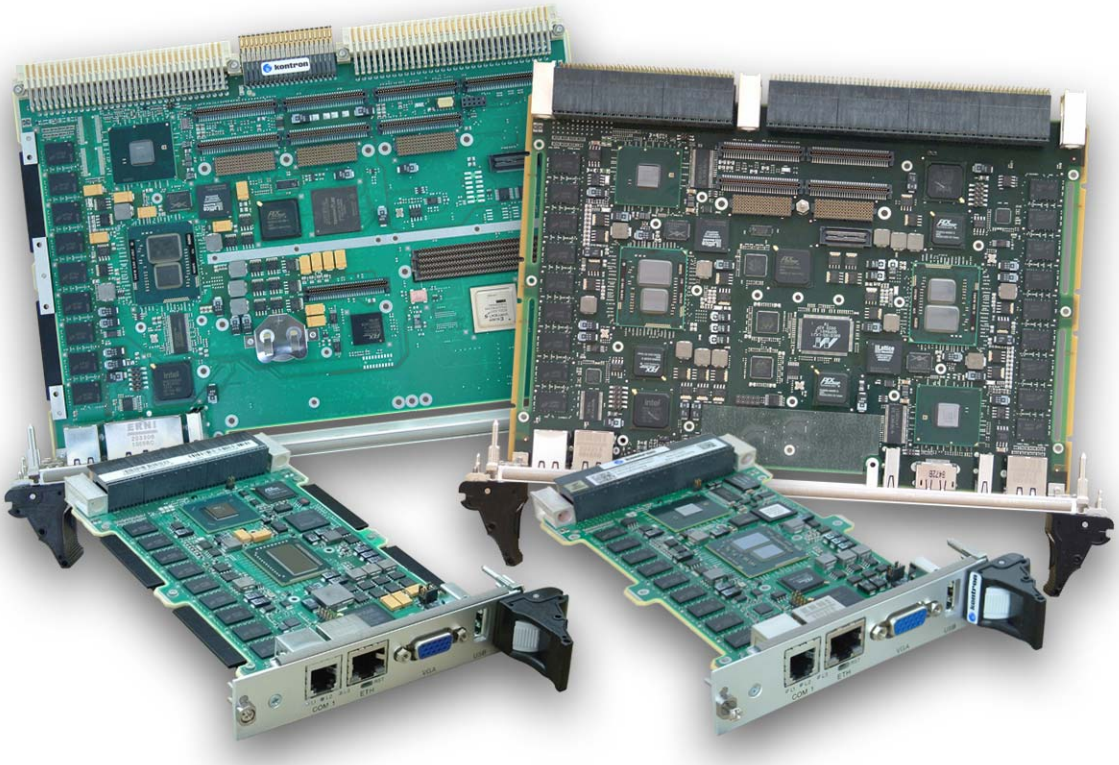
kontron

» VM6050, VX6060, VX3030 & VX3035 «



Release Notes Fedora 14
on VM6050, VX6060, VX3030 & VX3035
Version 2.5 - ID 12269

SD.DT.F82-3e - February 2013

Revision History

Publication Title:	Fedora14 Release Notes	
Doc. ID:	SD.DT.F82-3e	
Rev.	Brief Description of Changes	Date of Issue
3e	Added CRP	02-2013
2e	New ID 12269 Add CRP in chapter 8.2	10-2012
1e	Update ID12117 and BSP 2.4 Add chapters 6.8, 6.9	07-2012
0e	Initial Version	06-2012

Copyright © 2013 Kontron AG. All rights reserved. All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

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

List Of Figures

Chapter 1 - Overview

Functional changes that differ from previous version of the document are identified by a vertical bar in the margin.

Linux, the Open Source Operating System is now taking a significant share of the OS market in Defense and Aerospace, after having taken ground initially in the enterprise server sector.

The goal of this document is to help you through the installation process of the Fedora 14 BSP distribution on the Kontron 6U VME VM6050, 6U VPX VX6060 , 3U VPX VX3030 and VX3035 boards.

In this document, the terms VM6050, VX6060, VX3030 or VX3035 are used for the VM6050, VX6060, VX3030 or VX3035 boards in standard or rugged conduction-cooled version:

VM6050 Single-slot 6U VME board

- VM6050-SA Standard Commercial version
- VM6050-RC Rugged Conduction-Cooled version

VX6060 Single-slot 6U VPX board

- VX6060-SA Standard Commercial version
- VX6060-RC Rugged Conduction-Cooled version

VX3030 Single-slot 3U VPX board

- VX3030-SA Standard Commercial version
- VX3030-RC Rugged Conduction-Cooled version

VX3035 Single-slot 3U VPX board

- VX3035-SA Standard Commercial version
- VX3035-RC Rugged Conduction-Cooled version

In this document, the terms VM6050-RTM, VX6060-RTM, VX3030-RTM or VX3035-RTM are associated to the VM6050, VX6060, VX3030 or VX3035 Rear Transition Module (RTM):

VM6050-RTM Rear Transition Module for the single-slot 6U VPX board

- PBV36-P0-VM6-00

VX6060-RTM Rear Transition Module for the single-slot 6U VPX board

- PB-VX3-011

VX3030-RTM Rear Transition Module for the single-slot 3U VPX board

- PB-VX3-011

VX3035-RTM Rear Transition Module for the single-slot 6U VPX board

- PB-VX3-011

Specific VX6060 Terminology

The VX6060 board is implemented as two similar CPU subsystems separated by a central PMC/XMC slot. This implies a common description of CPU subsystems on the board (ex: both parts have same CPU / SATA / USB / DDR3 /… interfaces).

For convenience, the CPU subsystem that faces P0/P1/P2 is called subsystem A and the second CPU subsystem on the other side is called subsystem B. In this documentation:

- CPU in subsystem A is named CPUA,
- CPU in subsystem B is named CPUB,
- USB in subsystem A is named USBA,
- USB in subsystem B is named USBB,
- and so on.

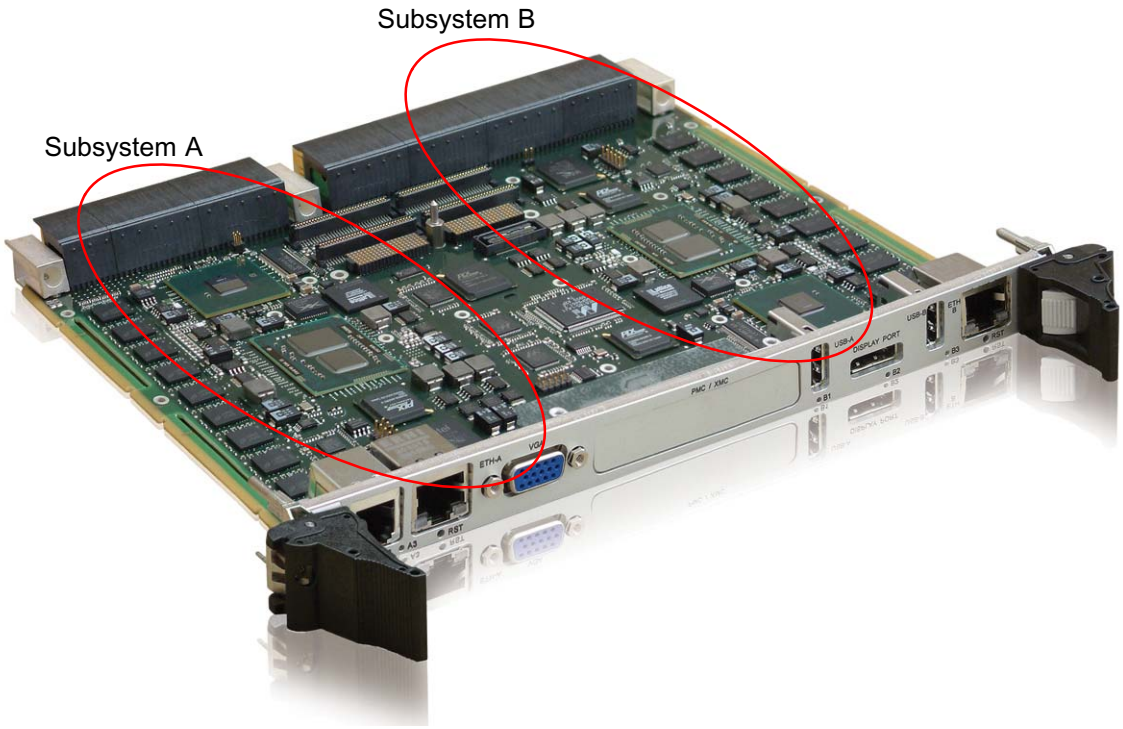


Figure 1: VX6060 Overview (Non contractual photography)

The VX6060 computing node is a VPX computing blade for parallel data and signal processing application.

With two independently implemented Intel® Core™ i7 processing nodes linked to a powerful Ethernet and PCIe infrastructure, the VX6060 is the ideal building block for intensive parallel computing workloads where a cluster of VX6060s is used in distributed or centralized OpenVPX environments. Target applications include radar, sonar, imaging systems, airborne fighters, and unmanned aerial vehicle (UAV) radar, as well as rugged multi-display consoles.

Chapter 2 - Release Content

The release is made of:

- Fedora 14 i686 OR x86_64 DVD HEADLESS (1/1 DVD is i686 architecture -32b- 1/2 DVD is x86_64 architecture -64b-)

  It is required to install the related BSP DVD on the same Fedora flavor: 32-bit BSP on Fedora 14 for i686 and 64-bit BSP on Fedora 14 for x86_64.

  This DVD-ROM contains a standard Fedora 14 distribution but with the following choices added to the installation menu for an installation on a "headless" configuration (no graphics display):

  - installation using the serial console
  - installation using VNC.

- one CD-ROM, **Fedora 14 x86_64 BSP 2.5 ID 12269** on **VM6050, VX6060, VX3030 & VX3035**

  This CD-ROM contains only the BSP specific packages as well as the related source packages.

The Board Support Package (BSP) provides support for some specific features of the board:

Kernel:	Update of the kernel of the Fedora 14 to support VM6050, VX6060, VX3030 & VX3035 specifics features and to fix issues. Note that this kernel is required by the following features.
Sensors:	CPU Cores and Board temperatures and voltages.
Vital Product Data (VPD) Tool:	Get board's serial number, order code, E.C. Level, ...
LEDs:	Four Front Panel Tri-color LEDs
GPIO:	Driver to support the 5 GPIOs of the VM6050, VX6060, VX3030 & VX3035 (CPUA)
Watchdog:	Drivers to setup the Watchdogs of the board.
BIOS Update tool:	A command and script to update the BIOS of the board.
Allocator:	Driver and library to reserved contiguous memory area.
VXFabric:	Tool and driver to communicate over the PCIe (VPX Backplane and between CPUA and CPUB)
CPLD register Tool (cpldtool):	Tool to deal with hardware registers of the onboard CPLD
FRAM support:	Driver and special API file to read/write from/to the FRAM
Diskless:	Tool to configure, boot and manage diskless boards.
SMI:	swsmi tool is used to access the onboard Ethernet switch (VX6060 only).
e1000e:	an update of kernel module to support the 82579 Ethernet chipset.
PBIT report:	sysvartool gives the report of the PBIT.
I2C buses drivers:	i2c bus drivers for the local i2c bus and the two backplane i2c busses.

More information on VM6050, VX6060, VX3030 & VX3035 BSP in Chapter 7 “BSP Specific Features” page 31.

Information on Fedora 14 is available at http://fedoraproject.org/

Chapter 3 - Associated Documentation

Kontron Documentation

- Hardware
  - VM6050 6U VME SBC User's Guide CA.DT.A93
  - VM6050 Hardware Release Notes CA.DT.A94
  - VX6060 6U VPX SBC User's Guide CA.DT.A76
  - VX6060 Hardware Release Notes CA.DT.A77
  - VX3030 3U VPX SBC User's Guide CA.DT.A87
  - VX3030 Hardware Release Notes CA.DT.A88
  - VX3035 3U VPX SBC User's Guide CA.DT.A95
  - VX3035 Hardware Release Notes CA.DT.A96
- Firmware
  - VM6050 BIOS User Manual SD.DT.F89
  - VX6060 BIOS User Manual SD.DT.F69
  - VX3030 BIOS User Manual SD.DT.F81
  - VX3035 BIOS User Manual SD.DT.F97

Fedora 14 Documentation

- Documentation available at http://fedoraproject.org/

Chapter 4 - Required Configuration

4.1 Hardware Requirements

4.1.1 Hardware Requirements for VM6050 Board

- A Kontron VM6050 board.
- The Fedora 14 release may be installed on one of the following bootable disks:
  - a SATA disk connected to the SATA connectors available on VM6050-RTM board.
  - Optional onboard USB Flash Disk.
- A USB DVD-ROM device (for installation from DVD-ROM).
- A console on serial line (text or VNC install).

For a graphic configuration on VM6050 a specific order code and a specific graphic module are required: VM6050-2SA34-12110 and MOD-GX-SA-00. The module provides two DP ports and a VGA connector.

4.1.2 Hardware Requirements for VX6060 Board

- A Kontron VX6060 board.
- The Fedora 14 release may be installed on one of the following bootable disks:
  - a SATA disk connected to the SATA connectors available on VX6060-RTM board for the CPUA.
  - another SATA disk connected to the SATA connectors available on VX6060-RTM board for the CPUB.

The SATA disk connected on the CPUB (through the RTM) may be optionnal if the diskless boot configuration is used (refer to chapter 5.2 page 19)

  - Optional onboard USB Flash Disk.
- A USB DVD-ROM device (for installation from DVD-ROM).
- Graphical display, USB keyboard and USB mouse (for a graphics install) or a console on serial line (text or VNC install).

The VX6060 board is implemented as two similar CPU subsystems separated by a central PMC/XMC slot. This implies a common description of CPU subsystems on the board (ex: both parts have same CPU / SATA / USB / DDR3 /... interfaces).

For convenience, the CPU subsystem that faces P0/P1/P2 is called subsystem A and the second CPU subsystem on the other side is called subsystem B.

In this documentation:

- CPU in subsystem A is named CPUA,
- CPU in subsystem B is named CPUB,
- USB in subsystem A is named USBA,
- USB in subsystem B is named USBB,

- and so on.

4.1.3 Hardware Requirements for VX3030 Board

- A Kontron VX3030 board.
- The Fedora 14 release may be installed on one of the following bootable disks:
  - a SATA disk connected to the SATA connectors available on VX3030-RTM board.
  - Optional onboard USB Flash Disk.
- A USB DVD-ROM device (for installation from DVD-ROM).
- Graphical display, USB keyboard and USB mouse (for a graphics install) or a console on serial line (text or VNC install).

4.1.4 Hardware Requirements for VX3035 Board

- A Kontron VX3035 board.
- The Fedora 14 release may be installed on one of the following bootable disks:
  - a SATA disk connected to the SATA connectors available on VX3035-RTM board.
  - Optional onboard USB Flash Disk.
- A USB DVD-ROM device (for installation from DVD-ROM).
- Graphical display, USB keyboard and USB mouse (for a graphics install) or a console on serial line (text or VNC install).

4.2 Firmware Requirements

4.2.1 VM6050 Board

The version of the BIOS firmware must be at least:

> 11332

This version is displayed in the BIOS Setup.

4.2.2 VX6060 Board

The version of the BIOS firmware must be at least:

> 11159

This version is displayed in the BIOS Setup.

4.2.3 VX3030 Board

The version of the BIOS firmware must be at least:

> 11186

This version is displayed in the BIOS Setup.

4.2.4 VX3035 Board

The version of the BIOS firmware must be at least:

> 12174

This version is displayed in the BIOS Setup.

4.3 Software Requirements

> The DVD-ROM for Fedora 14:

Fedora 14 x86_64 DVD HEADLESS

> The DVD-ROM for the BSP:

Fedora 14 x86_64 BSP 2.5 ID12269 on VM6050, VX6060, VX3030 & VX3035

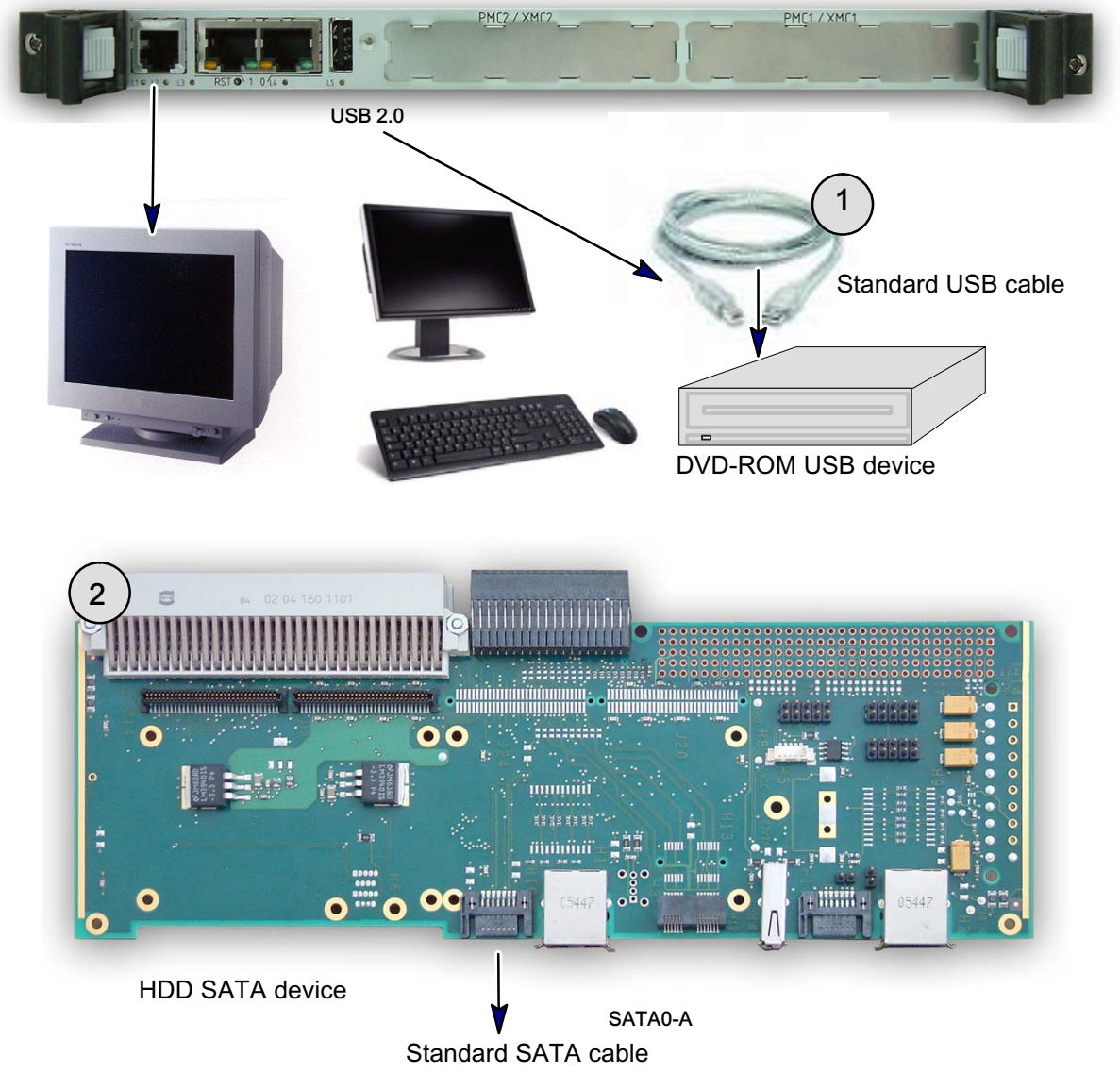
After the release is installed, the release version is saved in `/etc/<board>-release`:
`# cat /etc/vx3030-release VX3030 Board Support Package 2.4 [12117]`
(Example for the VX3030 board)

The 64-bit distribution should be privileged on VM6050, VX6060, VX3030 & VX3035 due to the CPU utilization and performances in 64-bit mode which are better than 32-bit in most of the benchmark cases. The 32-bit distribution still may be required if customers' or third parties' drivers are not available in 64-bit mode kernel. Note that most of the 32-bit applications may be used on a 64-bit distribution if 32-bit required libraries are installed and available. The last point to take into account is the fact that the memory and storage footprint may be a little bit more important in 64-bit than in 32-bit.

4.4 DVD-ROM Installation Example

4.4.1 DVD-ROM Installation Example for VM6050

1. Plug a DVD-ROM USB device to the front panel USB connector using a standard USB cable.
2. Plug the HDD SATA device on the SATA0-A connector of the Rear Transition Module (VM6050-RTM).

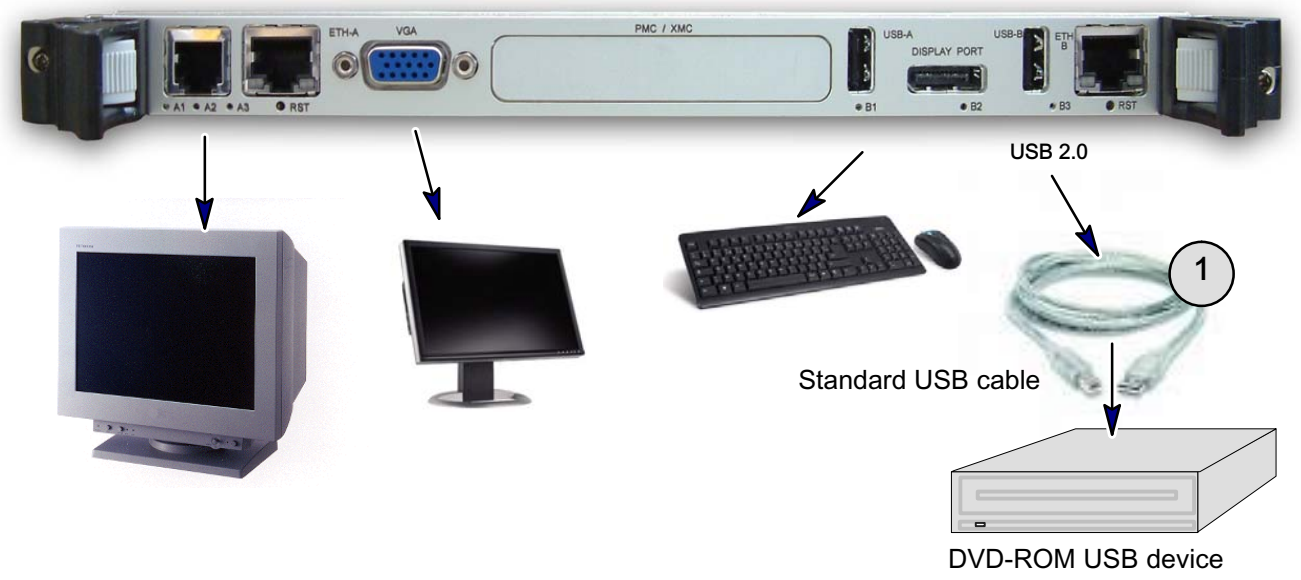


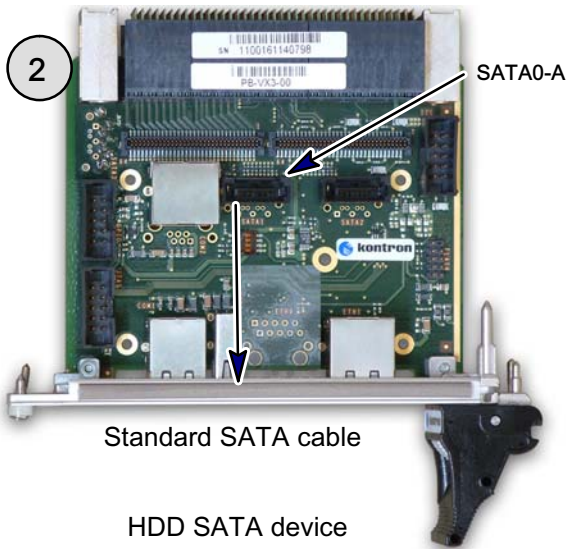
Graphic configuration on VM6050 requires a specific order code and a specific graphic module: VM6050-2SA34-12110 and MOD-GX-SA-00. The module provides two DP ports and a VGA connector.

There are 2 USB ports available: one on the front panel and the other one on the RTM board if a RTM is available. So in order to use a USB keyboard and a USB mouse plus a USB DVD-ROM, a USB HUB is required.

4.4.2 DVD-ROM Installation Example for VX6060

1. Plug a DVD-ROM USB device to the front panel USBA connector using a standard USB cable.
2. Plug the HDD SATA device on the SATA0-A connector of the Rear Transition Module (VX6060-RTM).

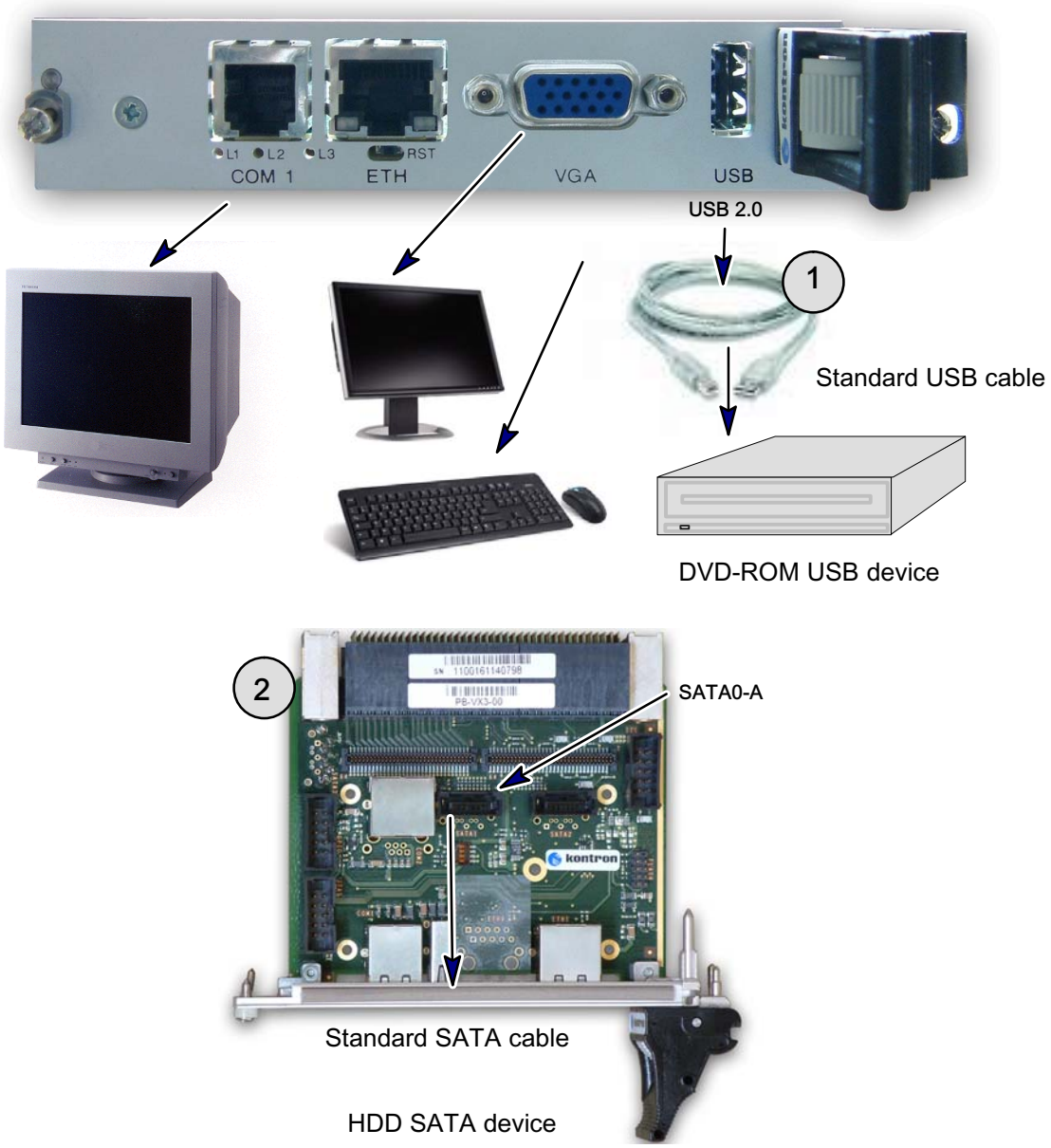




There are 2 USB ports available: one on the front panel and the other one on the RTM board if a RTM is available. So in order to use a USB keyboard and a USB mouse plus a USB DVD-ROM, a USB HUB is required. Nevertheless, the USB-B may be redirected to CPU-A through BIOS menus and used to complete the USB ports.

4.4.3 DVD-ROM Installation Example for VX3030

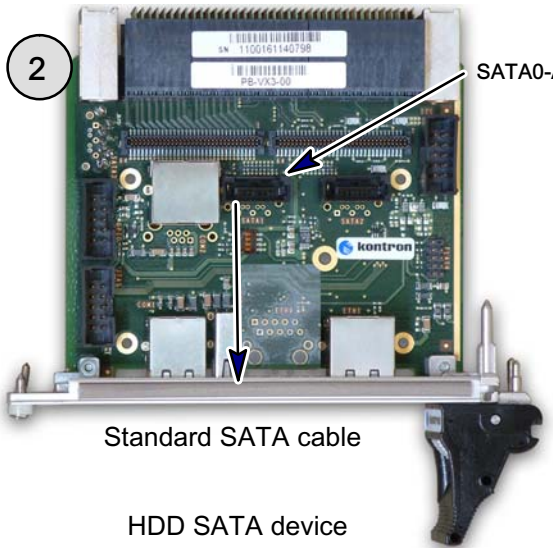
1. Plug a DVD-ROM USB device to the front panel USB connector using a standard USB cable.
2. Plug the HDD SATA device on the SATA0-A connector of the Rear Transition Module (VX3030-RTM).



Note There are 2 USB ports available: one on the front panel and the other one on the RTM board if a RTM is available. So in order to use a USB keyboard and a USB mouse plus a USB DVD-ROM, a USB HUB is required.

4.4.4 DVD-ROM Installation Example for VX3035

1. Plug a DVD-ROM USB device to the front panel USB connector using a standard USB cable.
2. Plug the HDD SATA device on the SATA0-A connector of the Rear Transition Module (VX3035-RTM).



There are 2 USB ports available: one on the front panel and the other one on the RTM board if a RTM is available. So in order to use a USB keyboard and a USB mouse plus a USB DVD-ROM, a USB HUB is required.

Chapter 5 - Installation

This chapter describes the specific steps of the installation process of the Fedora 14 on VM6050, VX6060, VX3030 & VX3035 boards.

For the VX6060, and because this board is a dual subsystem board, two main installation methods are described hereafter:

- A dual disk configuration (sections 5.1.2 and 5.1.4):

  A boot SATA disk on CPUA and another boot SATA disk on CPUB. With this configuration, CPUA and CPUB are quite independant. The communication between the two subsystems may be done through the local network switch or through the VXFabric (refer to section 5.1).

- A CPUB diskless configuration (section 5.2):

  CPUA is configured as a disk server for CPUB. CPUB boot through the network using PXE/NFS setup and the "multinodes-diskless" package provided with the BSP (refer to section 5.2 page 19)

5.1 Disk Installation

5.1.1 Fedora 14 DVD Menu

This section describes the options added to the installation menu on the Fedora 14 x86_64 or i686DVD HEADLESS media.

If a graphics display and a USB keyboard / mouse are available, the installation can be done using a standard Fedora 14 x86_64 DVD or using the Fedora 14 x86_64 DVD HEADLESS, by choosing "**Install or upgrade using GRAPHICS mode**".

But if the system is a headless configuration where the only console available is a terminal connected to the serial line, use the Fedora 14 x86_64 DVD HEADLESS.

When booting from this media, the following menu appears on the serial line (and also on the graphics display if present):

```
*                  Welcome to Fedora 14!                  *
***********************************************************
* Install or upgrade using GRAPHICS mode                  *
* Install or upgrade using SERIAL console                 *
* Install or upgrade using VNC                            *
* Install system with basic video driver                  *
* Rescue installed system using GRAPHICS mode             *
* Rescue installed system using SERIAL console            *
* Boot from local drive                                   *
* Memory test                                             *
*                                                         *
*                                                         *
*                                                         *
*                                                         *
***********************************************************
```

Select one of these options :

- "`Install or upgrade using SERIAL console`": to do all the installation using the serial console. However in this mode, the disk partitionning cannot be customized and only a fixed very minimal set of 200 packages is installed. Depending on the needed services and libraries, some post installations may be required, so this is not the recommended method. If possible, install using VNC (see below) or using a graphics display.
- "`Install or upgrade using VNC`": to start the installation using the serial console, but then use a VNC client on another system to do the rest of the installation in graphics mode with all installation options available. This requires the network to be connected to the board.

For VX3035, as the 82579 Ethernet chipset is not supported by the standard Fedora 14 kernel, it is not possible to use VNC for install. Use SERIAL install, then install the BSP, and install additional software using "`yum install`" or "`yum groupinstall`".

When installing using the "`Install or upgrade using VNC`", the default behaviour is to use a DHCP server to provide the network settings to the board.

If no DHCP server is available, it is possible to provide these settings manually as follows:

- Move the cursor to the "`Install or upgrade using VNC`"
- Press <tab> and append the network settings; at least the IP address :

```
> vmlinuz initrd=initrd.img console=ttyS0,155200 text stage2=hd:LABEL="Fedora" vnc
ip=172.20.144.95
```

or more settings :

```
> vmlinuz initrd=initrd.img console=ttyS0,155200 text stage2=hd:LABEL="Fedora" vnc
ip=172.20.161.201 netmask=255.255.255.0 gateway=172.20.161.46 dns=172.20.144.1
```

- and press <ENTER>

The following menu should be displayed (the Ethernet device list is an example):

```
Welcome to Fedora for x86_64

              ***************** Networking Device *****************
              *                                                   *
              * You have multiple network devices on this system. *
              * Which would you like to install through?          *
              *                                                   *
              *  eth0 - Ethernet device eth0 - 00:00:de:40:39:b5  *
              *  eth1 - Ethernet device eth1 - 00:00:de:40:39:b6  *
              *  eth2 - Ethernet device eth2 - 00:00:de:40:39:b7  *
              *  eth3 - Ethernet device eth3 - 00:00:de:40:39:b8  *
              *  eth4 - Ethernet device eth4 - 00:30:f7:98:2b:00  *
              *  eth5 - Ethernet device eth5 - 00:30:f7:98:2b:01  *
              *                                                   *
              *    ******        ************        ********     *
              *    * OK *        * Identify *        * Back *     *
              *    ******        ************        ********     *
              *                                                   *
              *                                                   *
              *****************************************************

  <Tab>/<Alt-Tab> between elements  | <Space> selects | <F12> next screen
```

Select the interface to use during installation to connect with the VNC client

```
Welcome to Fedora for x86_64

              ************* Disc Found *************
              *                                    *
              * To begin testing the media before  *
              * installation press OK.             *
              *                                    *
              * Choose Skip to skip the media test *
              * and start the installation.        *
              *                                    *
              *     ******           ********      *
              *     * OK *           * Skip *      *
              *     ******           ********      *
              *                                    *
              *                                    *
              ************************************

<Tab>/<Alt-Tab> between elements  | <Space> selects | <F12> next screen
```

Select if you want to check the media or not. In any case, at the end, the following messages are displayed:

```
Running anaconda 14.22, the Fedora system installer - please wait.
19:01:12 Starting VNC...
19:01:14 The VNC server is now running.
19:01:14

WARNING!!! VNC server running with NO PASSWORD!
You can use the vncpassword=<password> boot option
if you would like to secure the server.

19:01:14 Please manually connect your vnc client to 172.20.144.95:1 to begin the install.
Press <enter> for a shell
19:01:18 Starting graphical installation.
```

Then on a workstation, launch a VNC client to connect to the server running on the board. In the example above, the VNC server is 172.20.144.95:1

Then continue the installation process.

On a Fedora workstation, you can use the "TigerVNC Viewer" VNC client supplied with the distribution (can be launched from menu Application / Internet)

5.1.2 Fedora 14 Installation on CPUA of VX6060 and on other Boards

This section describes the installation procedure from an USB DVD-ROM drive.

There is no major difference between the installation on the VM6050, VX6060, VX3030 & VX3035 and the standard Fedora 14 installation, so refer to the Fedora documentation to get more details on all the Fedora installation menus.

Nevertheless, the VX6060 or VX3030 is a graphic board, so the installation may be done in graphic mode or in text mode on the serial port 0.

1. Insert the DVD Fedora 14 in the DVD-ROM driver.
2. After a board reset, or a board power-on, type **<F7>** to get the the Boot Manager Menu or **<F2>** to get the Setup from the BIOS and to select the DVD-ROM device as the boot device.
3. Select the installation method using the menu, as explained above.
4. Then the standard Fedora 14 Installer menus from anaconda should be displayed.
5. Proceed as a standard Fedora 14 installation.
6. Note that the mezzanine USB Flash device, if present, is probed as another SATA drive.
7. At the end of the installation, reboot on the installed disk drive through the Boot Manager Menu **<F7>**. The BIOS Setup menu should be used to set the boot devices priorities. Refer to the BIOS Manual - SD.DT.F69.


It is hardly recommended to disable swap partition if the installation is done on a USB or SATA flash device.

5.1.3 BSP Installation on CPUA of VX6060 and on other Boards

The BSP CD-ROM contains only the BSP specific packages as well as the related source packages.

Follow the procedure below, to install the BSP onto an existing bootable media (SATA disk or USB flash) or to access to the source packages.

1. Insert the Kontron BSP CD-ROM in the USB DVD-ROM drive.
2. Boot the VX6060 board on the SATA disk previously installed with the Fedora 14.
3. Login as root and run following commands:

```
[root@ki7]# mount /dev/sr0 /mnt
[root@ki7]# cd /mnt
[root@ki7]# ./install.sh -i
```

If diskless tool is not used, do `[root@ki7]# chkconfig diskless off` to avoid the service to be started at boot time.

The installation process will take few minutes.

4. Reboot the machine to take into account the new version of the BSP.

When the BSP is installed, the kernel is updated.

If you want to build your own kernel modules and software for this new kernel, you should also update kernel-headers and kernel-devel as follows :

```
[root@ki7]# yum install kernel-headers kernel-devel
```

These packages will be installed from the local BSP repository configured by the BSP installation.

At this step you should configure the system to use the network service instead of the NetworkManager, and also do some other settings to configure the network interfaces to meet the requirements of an embedded system. See important information about that at section "6.1 - Network page 22"

5.1.4 VX6060 - Fedora Installation on CPUB

The serial console of CPUB is available through the COM-AB connector on the front panel. It requires specific cable and connector to get the serial console for CPUA and CPUB. Refer to your Kontron representative to get the information about such a cable.

1. Plug a DVD-ROM USB device to the front panel USBA/B connector using a standard USB cable.
2. Check in the BIOS setup menu the USBA/B is dedicated to the CPUB.
3. Plug the HDD SATA device on one of the SATA0-B connector of the Rear Transition Module (VX6060-RTM).

Then boot on the DVD-ROM USB and the installation process is similar to the installation steps on CPUA: section 5.1.2 page 17.

It is possible to configure the CPUB in graphical mode using the dedicated rear backplane DisplayPort connection. Refer to the Hardware User Manual to get more information or contact your Kontron representative.

5.1.5 VX6060 - BSP Installation on CPUB

Check in the BIOS setup menu that the USBA/B is dedicated to the CPUB.

Boot on the previously installed Fedora 14 SATA disk.

Then insert the BSP CD-ROM in the DVD-ROM driver and follow the step of the section 5.1.3.

5.1.6 VX3035 - Graphic support

The graphic chipset of the VX3035 is not fully supported by the Fedora 14. This implies two different workarounds to get the graphic available depending on the graphic requirement:

1. Minimal support of the graphic may be done by adding the option "`nomodeset`" to the Linux kernel command line option in the `/etc/grub.conf` file.

   This can be done by editing directly the `/etc/grub.conf` or by issuing the following command:

   ```
   [root@ki7]# KERNEL=`grubby --default-kernel` ; grubby  --update-kernel=${KERNEL} --args=nomodeset
   ```

   and then reboot.

2. The Fedora 15 is providing a better support of the graphic of the VX3035. It is possible to upgrade the **xorg** part of the Fedora 14 to the Fedora 15 level.

   ```
   [root@ki7]# yum --releasever=15 --nogpgcheck update xorg\*
   ```

   This requires an access to a Fedora 15 repository via **yum** services.

5.2 VX6060 - CPUB Diskless Installation

The CPUA and CPUB of the VX6060 are connected through an Ethernet switch so a network boot (through BIOS PXE) may be setup to get a diskless Linux system on the CPUB with the CPUA as a server.

The package multinodes-diskless is available to setup and configure a NFS exportable root filesystem and to get a kernel and a `initrd` bootable through PXE.

The following steps describe how to initialize a diskless configuration for the CPUB and how to boot it.

0. The very first step is to get a local repository on CPUA of the Fedora 14 rpm packages.

   Insert the Fedora 14 DVD and mount it:

```
[root@ki7]# mkdir -p /dist/Fedora 14_VX6060/{BSP,RPMS}
[root@ki7]# mount /dev/dvd /mnt
[root@ki7]# cp /mnt/Packages/* /dist/Fedora 14_VX6060/RPMS
[root@ki7]# cp /usr/share/vx6060_bsp/bsp_11068/RPMS/*.rpm /dist/Fedora 14_VX6060/BSP
[root@ki7]# cd /dist/Fedora 14_VX6060/
[root@ki7]# createrepo .
```

1. Install the multinodes-diskless package.

   The standard installation (refer to previous chapters) of the BSP provides the multinodes-diskless package with. Nevertheless, if it is not installed, the following command will install it:

```
   [root@ki7]# yum install --enablerepo=vx6060-bsp-12269 multinodes-diskless
```

2. In the following, the local sub-network of the diskless is 192.168.1. This may be adapted to the network constraints.

3. Configure the ETH1 on the CPUA as the server Ethernet interface with the IP address 192.168.1.1

   This may be done through the command `system-config-network` or by editing the file:

```
   [root@ki7]# vi /etc/sysconfig/network-scripts/ifcfg-eth1

   # Network Interface
   DEVICE=eth1
   ONBOOT=yes
   NETMASK=255.255.255.0
   IPADDR=192.168.1.1
   BOOTPROTO=none
   TYPE=Ethernet
   NM_CONTROLLED=no
   IPV6INIT=no
   USERCTL=no
```

   Then setup the interface:

```
   [root@ki7]# ifup eth1
```

   Check it is proprely done:

```
   [root@ki7]# route
```

4. Copy the diskless configuration files provided with the BSP:

```
[root@ki7]# cp /usr/share/vx6060_bsp/diskless_fs.xml /etc/diskless/
[root@ki7]# cp /usr/share/vx6060_bsp/images.xml /etc/diskless/
[root@ki7]# chmod +w /etc/diskless/*.xml
```

5. WARNING: In the file `/etc/diskless/diskless_fs.xml`, remove the lines:

```
<abs>[ -f /root/.ssh/id_rsa.pub ] || ssh-keygen -y</abs>
<abs>mkdir -p %(rootdir)/root/.ssh </abs>
<abs>cp -f /root/.ssh/id_rsa.pub %(rootdir)/root/.ssh/authorized_keys
</abs>
```

   If present.

6. Edit the file `/etc/diskless/image.xml` and modify the MAC adresses accordingly with the MAC address of the CPUB.

   The MAC address of the CPUB is displayed by the EDI shell command:

```
VX6060-B> kmac -r
Read data from flash at 0x2000 size 0x6
MAC Address = 00:00:DE:40:36:F9
```

   Set this address in the file `/etc/diskless/image.xml`:

```
...
<nodemap id="diskless-cluster_A" clients="diskless_A" default="1">
  <match criterium="macaddress">
     <value>00:00:DE:40:36:F9</value>
  </match>
  <ip>192.168.1.%(1+order.value)</ip>
...
```

7. Stop the `iptables` service and disable the `selinux` by running the commands:

```
[root@ki7]# chkconfig iptables off
[root@ki7]# service iptables stop
```

8. Set the variable SELINUX=disabled into the file : `/etc/selinux/config`
9. Start the required service: `tftpd` and `nfs` by running the commands :

```
[root@ki7]# chkconfig tftp on
[root@ki7]# chkconfig nfs on
[root@ki7]# service nfs start
```

   Those services may required to be installed first.

10. Reboot the server in order to take into account these changes.
11. To configure the exportable root file system on the server:

```
[root@ki7]# mkdir -p /diskless/layers.
[root@ki7]# diskless -B
```

   This will build the filesystem `/diskless/layers/`

This process will take few minutes.

12. Configure the **dhcpd** server as well as the PXE boot by:

```
[root@ki7]# diskless -x
```

13. Export the diskless RFS by :

```
[root@ki7]# diskless -e
```

14. On the CPUB and after a power-on, take the hand under CPUB by pressing **<F2>**.

Under BIOS Setup, select successively the menus:

`Chipset->South Bridge -> Ibex Peak Option`

And enable the `PXE boot.`

Save the changes and after reset, press again **<F2>**.

Select the menu "**boot**" and put the network device at the top of the list.

Save the change and exit.

As a result the PXE boot will happen and something like that should be displayed:

```
CLIENT MAC ADDR: 00 00 DE 40 36 F9  GUID: 00020003 0004 0005 0006
000700080009
DHCP.|
```

You should notice the occurence of a login at the end of the boot step.

Chapter 6 - Fedora System Configuration

In this chapter, informations related to some specific configuration items of the Fedora 14 system are detailed.

6.1 Network

6.1.1 Network Manager

With Fedora 14, the network interfaces are managed by the NetworkManager service by default.

For an embedded system, it is recommended to use the older network service instead which is easier to configure through configuration files.

For this:

- Disable the NetworkManager service:

```
[root@ki7]# chkconfig NetworkManager off
```

- Enable the network service:

```
[root@ki7]# chkconfig network on
```

- Stop the network manager: `service NetworkManager stop`
- Check the configuration files and modify then if needed :
  - `/etc/sysconfig/network-scripts/ifcfg-ethx` files
  - `/etc/resolv.conf`
  - `/etc/sysconfig/network`
- Start the network service: `service network start`
- Reboot if `/etc/sysconfig/network` has been modified

6.1.2 Network Interfaces Naming

udev Rules for VM6050 onboard Network Interfaces

The BSP of the VM6050 comes with **udev** rules to define the name of the network interfaces of the VM6050.

`eth0` and `eth1` from the i82580 are respectively the connector ETH0 and ETH1 on the front panel.

`eth2` and `eth3` also from the i82580 (quad-ethernet interface) are routed to the backplane on P0 connector and may be addressed through the RTM board (`eth2` is on the ETH1 (H2) connector near from the edge of the RTM, and `eth3` is on the ETH0 (h1) connector in the middle of the RTM board).

udev Rules for VX6060 onboard Network Interfaces

The BSP of the VX6060 comes with `udev` rules to define the name of the network interfaces of the VX6060.

On CPUA:

- the `eth0` (i82577) is connected to RJ-45 "ETH-A" front panel connector for SA board and to the Geth switch for RC board. This is configurable through BIOS setup.
- the `eth1` and `eth2` (i82580) are connected to onboard Giga-Ethernet switch.

On CPUB:

- `eth0` (i82577), `eth1` and `eth2` (i82580) are connected to the onboard Geth switch.

One port of the Geth switch is available on the front panel connector of SA board: RJ-45 "ETH-SW" connector.

Three SERDES 1000BASE-BX ports from the Giga-Ethernet switch are available on the backplane through P1 connector.

One 1000BASE-T port from the Giga-Ethernet switch is also available from P1 and routed to the rear module.

Refer to the User Manual of this board to get more information about network interfaces.

udev Rules for VX3030 onboard Network Interfaces

The BSP of the VX3030 comes with udev rules to define the name of the network interfaces of the VX3030:

- the `eth0` (i82577) is connected to RJ-45 "ETH" front panel connector for SA board and to the RTM board (ETH1 connector). This is configurable through BIOS setup.
- the `eth1` and `eth2` (i82580) are routed to the backplane through P1 connector.

udev Rules for VX3035 onboard Network Interfaces

The BSP of the VX3035 comes with udev rules to define the name of the network interfaces of the VX3035:

- the `eth0` (i82579) is connected to RJ-45 "ETH" front panel connector for SA board or to the RTM board (ETH1 connector). This is configurable through BIOS setup.
- the `eth1` and `eth2` (i82580) are routed to the backplane through P1 connector.

udev Rules for other Network Interfaces

The udev Rules for all interfaces (onboard or not) are all located in the file :

```
/etc/udev/rules.d/72-ki7-net.rules.
```

For Network Interfaces that are not onboard (PMC or XMC), the rules define an interface name based on the location of the device in the PCI/PCIe bus tree, as displayed by the command `lspci`. This prevents the interface name from changing if the device is replaced by a new one for maintenance.

For example a device at PCI location 02:04.1 according to `lspci` has the interface name `eth-2-4-1`.

If this is not what you want, edit `/etc/udev/rules.d/72-ki7-net.rules` according to the explanations located in this file.

If you want to keep a location based name but change the format of this name, edit the script :

```
/lib/udev/get_ethname_from_pciloc.sh
```

Important information about interface naming

If no rule is found for an interface, `udev` creates a rule for it and assign a fixed interface name associated to its MAC address.

This is OK as long as you do not change the board or the PMC/XMC, otherwise a new interface name will be created for the new MAC address.

This is not suitable for an embedded system when boards must be changed for maintenance without requiring additional configuration.

On older releases, the problem was solved by editing `/lib/udev/rules.d/75-persistent-net-generator.rules` to remove `eth*` from the white list (add eth*| to line KERNEL!=) and remove `/etc/udev/rules.d/70-persistent-net.rules` if present.

This was preventing new rules from beeing created based on the MAC address and the interfaces were named in the order they where probed (always the same).

Unfortunately in recent releases, the probe is done concurrently by several threads (to speedup things), so the probe order may change from one boot to another, leading to changes in the physical device allocated to each device name.

The fix supplied by the BSP is a `udev` rule file (`/etc/udev/rules.d/72-ki7-net.rules`) that sets all onboard interfaces to `eth0`, `eth1` ..., and all other interfaces to `eth-x-y-z` (based on bus location).

For backward compatibility with older BSPs, an XMC with Intel 82599EB 10 Gbit has interface names:

- `eth3` and `eth4` on VX3030 and VX6060
- `eth4` and `eth5` on VM6050

but this can be changed by editing `/etc/udev/rules.d/72-ki7-net.rules`

Refer to the User's Guide of the board to get more information about the Ethernet connection.

6.1.3 MAC Address

By default, the MAC address is stored with the configuration parameters of each interface. If the MAC address of a device is found different from the one expected (board changed for example), the interface is not brought up. This is not suitable for an embedded system when boards must be changed for maintenance without requiring additional configuration.

To workaround this behavior, do not bind an Ethernet interface to a MAC address:

- Run `system-config-network`
- For each interface:
  - click on Edit
  - click on Hardware Device tab
  - unselect Bind to MAC address
- Exit from `system-config-network` saving changes

This can be done also by editing the `/etc/sysconfig/network-scripts/ifcfg-eth*` files and removing the `HWADDR` lines.

Removing `HWADDR` only works if the service network (and not NetworkManager) is used.

6.1.4 Firewall

If the firewall must be disabled but has been enabled during the installation:

- Run `system-config-firewall`, click on disable and exit

  OR run

```
[root@ki7]# service iptables stop
[root@ki7]# service ip6tables stop
```

- Make sure to disable the iptables service by running:

```
[root@ki7]# chkconfig iptables off
[root@ki7]# chkconfig ip6tables off
```

- Reboot

6.2 SELinux

SELinux stands for Security-Enhanced Linux. The Security-Enhanced Linux kernel enforces mandatory access control policies that confine user programs and system servers to the minimum amount of privilege they require to do their jobs.

If you experience some trouble running some services or have some permission issues, try to set the System Default Enforcing Policy to Permissive instead of Enforcing by running the `system-config-selinux` tool, or from command line doing as follows:

- disable on boot by editing `/etc/selinux/config` to set `SELINUX=permissive` instead of `SELINUX=enforcing`
- disable now: `setenforce 0`

6.3 cpuspeed

`cpuspeed` is a program that monitors the system's idle percentage and reduces or raises the CPU's clock speeds and voltages accordingly to minimize power consumption when idle, and maximize performances when needed. This is mainly useful on laptop and may not be suitable for embedded systems.

For embedded system, the "performance" policy (maximum performances) should be set in most of the case.

To set up the mode, edit the file `/etc/sysconfig/cpuspeed` and modify the line:

```
GOVERNOR=
```

by

```
GOVERNOR=performance
```

Then the service should be restarted:

```
[root@ki7]# service cpuspeed restart
```

6.4 GRUB Boot Loader

If your console is on the serial line and that access to the grub boot menu is not required on the graphics console, you should comment the following line in `/boot/grub/grub.conf` if present :

```
terminal --timeout=5 serial console
```

This will prevent GRUB from waiting 5 seconds to let you select a console by typing a character on one console to select it.

This setting has nothing to do with the timeout on the boot menu that is set by the line

```
timeout=<value>
```

6.5 Switching from one Board Type to Another

The BSP is common to VM6050, VX6060, VX3030 and VX3035, but when the BSP is installed a small specific BSP package named `<board>_bsp` is installed. This package contains board specific files and settings.

Once the BSP has been installed, it is possible to switch from one board type to another by uninstalling the current package and installing a new one.

For example to switch from a VM6050 to a VX3035:

```
[root@ki7]# yum remove vm6060_bsp
[root@ki7]# yum install vx3035_bsp
```

then shutdown and boot on the new board.

6.6 SATA Speed

With some specific environmental constraints or with some SATA devices, it may be required to reduce the SATA speed of a specific SATA port.

The SATA speed may be 1.5 Gbps or 3 Gbps on VM6050, VX6060 or VX3030 and up to 6 Gbps for VX3035.

To properly manage the SATA speed, first of all check at the BIOS setup that AHCI mode is enabled. Furthermore, AHCI mode should allow the access to Hotplug option (refer to next chapter).

The current speed of the SATA ports may be checked at boot time by:

```
[root@ki7]# dmesg | egrep ata[1-9]:
[    3.480624] ata1: SATA max UDMA/133 abar m2048@0xf1c02000 port 0xf1c02100 irq 45
[    3.487984] ata2: SATA max UDMA/133 irq_stat 0x00400040, connection status changed irq 45
[    3.496119] ata3: DUMMY
[    3.498552] ata4: DUMMY
[    3.500985] ata5: DUMMY
[    3.503419] ata6: DUMMY
[    3.809480] ata1: SATA link down (SStatus 0 SControl 300)
[    4.226743] ata2: SATA link up 3.0 Gbps (SStatus 123 SControl 300)
```

Here the port 2 is set at 3.0 Gbps.

If the hard drive is finaly not supporting this rate, it is possible to decrease it by adding `libata.force=2:1.5` (for example) to the kernel command boot line to force at boot time the SATA speed limit of the port 2 at 1.5 Gbps.

```
[root@ki7]# KERNEL=`grubby --default-kernel` ; grubby --update-kernel=$KERNEL --args=libata.force=2:1.5
```

Then after a reboot:

```
[root@ki7]# dmesg | egrep ata[1-9]:
[    3.480245] ata1: SATA max UDMA/133 abar m2048@0xf1c02000 port 0xf1c02100 irq 45
[    3.487603] ata2: FORCE: PHY spd limit set to 1.5Gbps
[    3.492627] ata2: SATA max UDMA/133 irq_stat 0x00400040, connection status changed irq 45
[    3.500762] ata3: DUMMY
[    3.503195] ata4: DUMMY
[    3.505627] ata5: DUMMY
[    3.508059] ata6: DUMMY
[    3.814088] ata1: SATA link down (SStatus 0 SControl 300)
[    4.231351] ata2: SATA link up 1.5 Gbps (SStatus 113 SControl 310)
```

6.7 SATA Hotplug

In AHCI mode, the SATA controler of the VX6060, VX3030, VX3035 and VM6050 provides with hotplug function. First of all this has to be setup at BIOS menus (Chipset->South Bridge Configuration->SATA Configuration). After the Hot Plug option is enabled for the SATA ports, boot the system.

» To remove a SATA device from the system:

Close all users of the device and backup device data as needed. Use `umount` to unmount any file systems that mounted the device.

Remove the device from any md and LVM volume using it. If the device is a member of an LVM Volume group, then it may be necessary to move data off the device using the `pvmove` command, then use the `vgreduce` command to remove the physical volume, and (optionally) `pvremove` to remove the LVM metadata from the disk.

If the device uses multipathing, run `multipath -l` and note all the paths to the device. Afterwards, remove the multipathed device using `multipath -f` device.

Run `blockdev -flushbufs` device to flush any outstanding I/O to all paths to the device. This is particularly important for raw devices, where there is no umount or vgreduce operation to cause an I/O flush.

Remove any reference to the device's path-based name, like `/dev/sd`, `/dev/disk/by-path` or the major:minor number, in applications, scripts, or utilities on the system. This is important in ensuring that different devices added in the future will not be mistaken for the current device.

Finally, remove each path to the device from the SCSI subsystem. To do so, use the command:

```
[root@ki7]# echo 1 > /sys/block/device-name/device/delete
```

where device-name may be `sde`, for example.

```
Nov 10 15:16:24 vm6050a kernel: [10018.256462] sd 1:0:0:0: [sdb] Synchronizing SCSI cache
Nov 10 15:16:24 vm6050a kernel: [10018.534156] sd 1:0:0:0: [sdb] Stopping disk
Nov 10 15:16:24 vm6050a kernel: [10018.934519] ata2.00: disabled
```

Then you can shut off the device.

» To add a SATA device:

When the system is up and running, power on the hotplugeable SATA device and the system should be warned that a new SATA device is available:

```
[root@ki7]# dmesg
...
Nov 10 15:22:40 vm6050a kernel: [10394.408164] ata2: irq_stat 0x00400040, connection status changed
Nov 10 15:22:40 vm6050a kernel: [10394.414149] ata2: SError: { RecovComm PHYRdyChg CommWake DevExch }
Nov 10 15:22:40 vm6050a kernel: [10394.420310] ata2: hard resetting link
Nov 10 15:22:43 vm6050a kernel: [10396.923493] ata2: SATA link up 3.0 Gbps (SStatus 123 SControl 300)
Nov 10 15:22:43 vm6050a kernel: [10396.925408] ata2.00: ATA-8: ST9160314AS, 0001SDM1, max UDMA/133
Nov 10 15:22:43 vm6050a kernel: [10396.925414] ata2.00: 312581808 sectors, multi 16: LBA48 NCQ (depth 31/32)
Nov 10 15:22:43 vm6050a kernel: [10396.927836] ata2.00: configured for UDMA/133
Nov 10 15:22:43 vm6050a kernel: [10396.927846] ata2: EH complete
Nov 10 15:22:43 vm6050a kernel: [10396.927991] scsi 1:0:0:0: Direct-Access ATA ST9160314AS\0001 PQ: 0 ANSI: 5
Nov 10 15:22:43 vm6050a kernel: [10396.928246] sd 1:0:0:0: Attached scsi generic sg1 type 0
Nov 10 15:22:43 vm6050a kernel: [10396.928299] sd 1:0:0:0: [sdb] 312581808 512-byte logical blocks: \
                                                                                (160 GB/149 GiB)
Nov 10 15:22:43 vm6050a kernel: [10396.928398] sd 1:0:0:0: [sdb] Write Protect is off
Nov 10 15:22:43 vm6050a kernel: [10396.928443] sd 1:0:0:0: [sdb] Write cache: enabled, \
                                                           read cache: enabled, doesn´t support DPO or FUA
Nov 10 15:22:43 vm6050a kernel: [10396.928772]  sdb: sdb1 sdb2
Nov 10 15:22:43 vm6050a kernel: [10396.955136] sd 1:0:0:0: [sdb] Attached SCSI disk
```

If automount services are enabled, the partitions should be mounted too.

6.8 Running Without the Battery Backed RTC

Some hardware constraints may prevent the use of the RTC's battery on the boards. This implies to workaround or disable some standard behaviour of Fedora services. The very first one is "`fsck`" which is run at boot time. `fsck` checks that the date of the last mount of the checked partition is prior than the current date. If not, the boot is stopped in a maintenance mode and this could happen after few times with the power off and without the RTC's battery. To workaround this, the current BSP provides a script which forces the current date to be set at the date of the last mount plus 1 minute. This script is run if the option "`fixrtc`" is added at the kernel command line. The script is in `/etc/sysconfig/modules/fixrtc.modules`.

Without the right date for the system, some other services and functions may present dysfunctions. For example, the command "`make`" warns if there is some previous build detected in the future and this could lead to bad recompilation. If the network is available, it is recommended to setup the service `ntpd` to adjust the system date with a date server.

6.9 VITA 57

On VM6050 or using VX3830 IO cards for VPX boards, it is possible to use the VITA 57 option. Kontron is delivering a toolkit to help the setup and development of application using the onboard FPGA and the FMC modules defined though the VITA 57 standard. Refer to your representative to get more information about this option.

6.10 Installation on a USB or SSD SATA Flash Device

Two main points which need a focus when the use of USB or SSD SATA flash device is concerned are:

The write bandwidth which may be quite low and the improvement of the timelife of such devices.

So some cautions have to be taken:

- Disable swap partitions on such a device.
- It may be required to remove the journal of the filesystem because the access to the journal may be too slow and be broken.

For example, to disable the journal on a ext4 filesystem:

```
tune2fs -O ^has_journal /dev/mapper/vg_lnx9-lv_root
```

Note: This has to be done for the FDM-SATA.

- Add noatime option to the `/etc/fstab`. Take care that some applications may required the access time of files and so be disturbed by this option. In `/etc/fstab`:

```
UUID=ffc6a6fc-1bf3-4f25-ab17-067b0515e85a /    ext4 noatime    1 1
```

Chapter 7 - BSP Specific Features

7.1 Sensors

The BSP contains an RPM named `sensors_addons` that configures the standard `lm_sensor` software for the VM6050, VX6060, VX3030 & VX3035 boards.

To display sensors information:

```
[root@ki7]# sensors
```

For VX6060, a BIOS >= 11067 is required to get the ACPI temp1 and temp2.

» Example on VX6060

```
[root@ki7]# sensors
acpitz-virtual-0
Adapter: Virtual device
temp1:       +38.0 °C  (crit = +100.0 °C)
temp2:       +38.0 °C  (crit = +100.0 °C)

lm73-i2c-6-48
Adapter: I2C CPLD adapter
 LM73 sensor Temperature: +38.0 °C  (low  = -30.0 °C, high = +70.0 °C)

lm73-i2c-6-49
Adapter: I2C CPLD adapter
 LM73 sensor Temperature: +43.0 °C  (low  = -30.0 °C, high = +70.0 °C)

lm73-i2c-6-4a
Adapter: I2C CPLD adapter
 LM73 sensor Temperature: +42.0 °C  (low  = -30.0 °C, high = +70.0 °C)

ads7830-i2c-5-4b
Adapter: SMBus I801 adapter at e000
12V VPX:     +11.66 V
5V VPX:       +4.95 V
3V3:          +3.30 V
2V5 6U:       +2.54 V
3V3 VPX SB:   +3.24 V
1V05S 3U:     +1.06 V
1V05 3U IBEX: +1.04 V
1V 3U:        +1.00 V
```

A BIOS >= 11067 is required to get the ACPI temp1 and temp2.

Example on VX3030

```
[root@ki7]# sensors
acpitz-virtual-0
Adapter: Virtual device
temp1:       +36.0°C  (crit = +100.0°C)
temp2:       +36.0°C  (crit = +100.0°C)

lm73-i2c-6-48
Adapter: I2C CPLD adapter
 LM73 sensor Temperature: +42.0°C  (low  = -30.0°C, high = +70.0°C)

ads7830-i2c-5-4b
Adapter: SMBus I801 adapter at e000
12V VPX:     +12.13 V
5V VPX:       +5.06 V
3V3:          +3.30 V
2V5 6U:       +2.48 V
3V3 VPX SB:   +3.28 V
1V05S 3U:     +1.05 V
1V05 3U IBEX: +1.05 V
1V 3U:        +1.00 V
```

7.2 Watchdog

The BSP provides a driver for the watchdog of the CPLD.

The `cpld_wdt` is used instead of the standard kernel iTCO_wdt module added to the blacklist by the file `/etc/modprobe.d/cpld_wdt.conf`. Additional cpld_wdt module options can be added to this file if needed.

To check that the driver is loaded:

```
[root@ki7]# lsmod | grep cpld_wdt
cpld_wdt                3138  0
...
```

This driver can be loaded with two options: timeout of the watchdog (timeout) and the mode (trigger_mode).

By default, timeout is 21s and the trigger_mode is reset mode which means that if the watchdog falls, the board is reset.

To start or service (restart) the watchdog :

```
[root@ki7]# echo > /dev/watchdog
```

To stop the watchdog :

```
[root@ki7]# echo V > /dev/watchdog
```

To change the timeout:

```
[root@ki7]# rmmod cpld_wdt
[root@ki7]# modprobe cpld_wdt timeout=11
```

Only odd timeout values are available.

To change the mode to get an interruption instead of a reset:

```
[root@ki7]# rmmod cpld_wdt
[root@ki7]# modprobe cpld_wdt timeout=11 trigger_mode=2
```

Then start the watchdog by writing to `/dev/watchdog` (echo), wait for the interrupt reading the watchdog (cat), and when the interrupt occurs disable the watchdog that is restarted by the termination of "cat" (close of the watchdog device) :

```
[root@ki7]# echo > /dev/watchdog ; time cat /dev/watchdog ; echo V > /dev/watchdog
```

If no one is waiting on the interruption (no "cat"), an emergency reboot system call is run when the watchdog expires.

7.3 VPD Tool

`vpdtool` command is useful to get information related to the VM6050, VX6060, VX3030 & VX3035 boards. The following examples are given for the VX6060 board:

```
[root@ki7]# vpdtool
VX6060 detected
Board type    : VX6060-2SA24-0000
EC Level      : 01000
Serial Number: 1110271030014
Variant       : 0010043801100001
Keylist                                                                                                 :
/PCB_A/SACLASS/MEZZUSB/BHDUAL/ARDLV2G/IBOMON/ETHSTD6U/1GB_DDR3_1067/DDR3-1067/IRTC/XDPON/STDC
LK/CK505REFOSC/STD_EARTH/BATON/HDDON/PXMC/2G5TS/P1PWRMAGOFF/PWRMAGOFF/FP422OFF/TMLVDSOFF/I2CS
TD/I2CAB/1SLOT/STDVPXKEY/P80OFF/VS2OFF/NOSYNCAB/ITIN/BC/
```

To get the features list:

```
[root@ki7]# vpdtool --human
VX6060 detected
Board type    : VX6060-2SA24-0000
EC Level      : 01000
Serial Number: 1110271030014
Variant       : 0010043801100001
Keylist                                                                                                 :
/PCB_A/SACLASS/MEZZUSB/BHDUAL/ARDLV2G/IBOMON/ETHSTD6U/1GB_DDR3_1067/DDR3-1067/IRTC/XDPON/STDC
LK/CK505REFOSC/STD_EARTH/BATON/HDDON/PXMC/2G5TS/P1PWRMAGOFF/PWRMAGOFF/FP422OFF/TMLVDSOFF/I2CS
TD/I2CAB/1SLOT/STDVPXKEY/P80OFF/VS2OFF/NOSYNCAB/ITIN/BC/

Features      :
PCB A
SA Class
USB mezzanine slot configuration
Dual link NH82580
Arrandale ECC LV 2GHz
Force extended temperature range BOM generation
6U basic Geth switch configuration, 1000Base-T link on P1
1GB DDR3-1067 device
DDR3-1067 dram speed
PCH internal RTC configuration
XDP port available
Standard clock option
CK505 oscillator source configuration
Standard EARTH connection
Battery present
on-board SATA HDD option ON
PMC P64s, XMC X8d+X12d IOs available on VPX P3/P4 connectors
```

```
Internal PCIe link speed set to 2.5GT/s
P1 magnetic power option off
Internal magnetic power option off
Front panel RS-422 option off
MLVDS terminations off
CPLD I2C equipment: one F-RAM, one EEPROM
CPLD A I2C linked to CPLD B I2C
One VPX slot version for SA
VPX Keying set to universal 6U VPX (315 degrees keying)
P80 debug option not available on debug connector
VS2 PWR pins not used
No Sync A/B option
PCB plating: Immersion Tin
Use of Backward compatible lead free component
```

For the VX3030 board:

```
[root@ki7]# vpdtool
VX3030 detected
Board type   : VX3030-SA22-01000
EC Level     : 01000
Serial Number: 1110431010002
Variant      : 0010042010020001
Keylist                                                                                                 :
/PCB_A/SACLASS/MEZZUSB/BHDUAL/ARDLV2G/IBOMON/ETHSTD6U/1GB_DDR3_1067/DDR3-1067/IRTC/XDPON/STDC
LK/CK505REFOSC/STD_GND/BATON/HDDOFF/NOMEZZIO/2G5TS/P1PWRMAGOFF/PWRMAGOFF/FP422OFF/TMLVDSON/I2
CSTD/I2CA/1SLOT/STDVPXKEY/P80ON/VS2OFF/NOSYNCAB/ITIN/BC/
```

7.4 LEDs

The driver `leds_cpld` allows to deal with the front panel LEDs for a user mode.

The driver `leds_cpld` creates a list of special files and classes in `/sys`:

```
[root@ki7]# ls /sys/class/leds/
led2:amber  led2:OFF  led3:amber  led3:OFF
led2:green  led2:red  led3:green  led3:red
```

For each LED, there are three different colors available: green, red, amber that can be set by addressing the related file.

For each color (which are exclusive), there are four different modes:

- ON (echo 0, see following example)
- low blinking (echo 1, see following example)
- fast blinking (echo 2, see following example)
- OFF (echo 1, see following example)

Example, to set those different modes on the LED 2 in amber:

```
[root@ki7]# echo 0 > led2:amber/brightness The LED 2 is ON in AMBER
[root@ki7]# echo 1 > led2:amber/brightness The LED 2 blinks low in AMBER
[root@ki7]# echo 2 > led2:amber/brightness The LED 2 blinks fast in AMBER
[root@ki7]# echo 0 > led2:OFF/brightness The LED 2 is OFF
```

When setting led2 or led3, both led2 and led3 are set to user mode.

To exit from this mode, set led<n>:OFF/brightness to a non null value; for example

```
[root@ki7]# echo 1 > led2:OFF/brightness
```

- VX6060: On the front panel, the name of the led1 on CPUA is L1, led2 is L2 and led3 is L3. On CPUB, led1 is L4, led2 is L5 and led3 is L6
- VX3030: and VX3035: On the front panel, the name of the led1 is L1, led2 is L2 and led3 is L3.
- VM6050: There are 4 users leds: L2 (led2), L3 (led3), L4 (led4), L5 (led5). The led1 (L1) is not manageable at user level.

The led1 (L1 and L4) is not manageable at user level, so no special file for led1 is available.

7.5 Allocator

Allocator is a module allowing the allocation of large contiguous memory areas. Allocator module may be needed to map VME bus windows or to get big bunch of memory for DMA transfers.

Refer to the man page allocator to get detailed information about this module.

Edit the file `/etc/init.d/allocator` in order to set the size of the different areas needed.

To start this service:

```
[root@ki7]# service allocator start
```

To get it activated at boot setup:

```
[root@ki7]# chkconfig --add allocator
```

Allocator is used to map VME windows to the local memory on the VM6050.

7.6 VXFabric

VXFabric is not supported by the VM6050 board (not a VPX board).

The VXFabric tools and kernel modules are provided to get communication over the PCIe between CPU nodes.

There are two main PCIe express domains which may be dedicated to VXFabric:

- Backplane VPX PCIe ports (#0)
- Onboard VX6060 PCIe between CPUA and CPUB (#1) - Only on VX6060

Each node on a VXFabric domain represents a board (or a subsystem in the VX6060 case) on the PCIe bus and gets a unique Node ID used to identify it.

There are two main ways to communicate on VXFabric domains: low level protocole, and IP over VXFabric.

The following sections give some inputs about VXFabric. There are also the different man pages available on the system to complete the information about it:

- vx (4) - VX fabric for DMA over PCIe on VX6060 with PLX8609
- vxdma (3) - DMA over PCIe fabric API
- vxfabric (1) - utility to perform DMA over PCIe
- vxfabric_api (3) - PCIe fabric API

7.6.1 Start the Service VXFabric

In order to setup VXFabric, a service is used. To have this service available, run the following command on each board of the chassis:

```
[root@ki7]# chkconfig --add vxfabric
```

Then reboot the boards.

7.6.2 The VXFabric Command

The VXFabric command may be used to get the status of the different nodes on the VXFabric domains and to exercise the low level communication (see next section) between nodes.

```
[root@ki7]# vxfabric --help
vxfabric: unrecognized option: --help
Usage vxfabric -S <SrcAddr> [-s <SrcType>] -D <DstAddr> [-d <DstType>] [-B <Size>] [-L
<LoopNum>] [-P]  [-R] [-C <ChanNum>]
Options are :
 [-a] : Give status of the whole nodes or of a particular node (with -N)
 [-S  <AddressValue>] :  Set the Physical SOURCE address ( PCI or MEMORY area )
 [-s  <AddressType> ] : Set the SOURCE address type, a value among :
         0 => Physical MEM or PCI Area ( requests -S <AddressValue>)
         1 => Allocated Virtual MEM Area
         2 => Allocated Phys MEM Area
[-D  <AddressValue>] :  Set the Physical DESTINATION address ( PCI or MEMORY area )
[-d   <AddressType>] : Set the DESTINATION address type ( i.e SOURCE address type  )
[-B  <SizeValue>] : Set the Size of the transfer ( DEFAULT value  = 1MB)
[-L  <LoopNum>] : Set the number of loops  of the transfer ( DEFAULT value  = 1
[-P ] : Select the Polling Mode ( DEFAULT value  = FALSE)
[-R  ] : Select the Round Robin Mode ( DEFAULT value  = FALSE)
[-C  <ChanNum> ] : Specify the channel number to be used ( DEFAULT value = 0)
[-F  <FabricBusNum> ] : Specify the Fabric bus number to be used ( DEFAULT value = 0)
[-M  <DMAEngineNum> ] : Specify the DMA engine number to be used ( DEFAULT value = 0)
```

```
[-c   ] : Enable the control of the transfered DATA
[-E   ] : READ the binary content of the EEPROM into a file
[-e   ] : READ the binary content of the EEPROM and convert it into an ASCII format
[-W   ] : WRITE the content of a binary DATA file into the  EEPROM
[-w   ] : Interpret a Command FILE , convert it to binary DATA and WRITE it to the  EEPROM
[-n   ] : Specify the source NODE ID
[-N   ] : Specify the destination NODE ID or the NODE ID containing the EEPROM  ( DEFAULT = 2
)
[-f   ] : Specify the FILE NAME which will contain the DATA of the EEPROM  ( DEFAULT =
/tmp/EEPROMBinFile )
[-r   ] : Perform a simple read to a PCIexpress address and evaluate the latency
[-dump] : args passed to dump routine. see -dump -h
[root@evalpasemi ~]#
```

The following command gives a status of the VXFabric on the VPX backplane.

```
[root@ki7]# vxfabric -a
-------------------------------------------------------------------------------
 VX FABRIC 0 configuration
-------------------------------------------------------------------------------
 N  H  S    S       C  S  P   P   M     A   (             I     A   (
 O  O  L    T       P  Y  C   C   E     R   S             O     R   S
 D  S  O    A       U  S  I   I   M     E   I                   E   I
 E  T  T    T          C  B   D   O     A   Z                   A   Z
 I          U          O  U   E   R         E                       E
 D          S          N  S   V   Y         )                       )
-------------------------------------------------------------------------------
 1  *  1    READY   A  *          0x14001000(0x3fff000)             (       )
 3     2    READY   A       4   0  0xf0000000(0x4000000)    0xf7c00000(0x20000)
-------------------------------------------------------------------------------
```

On the local VXFabric (for VX6060 between CPUA and CPUB):

```
[root@ki7]# vxfabric -a -F1
-------------------------------------------------------------------------------
 VX FABRIC 1 configuration
-------------------------------------------------------------------------------
 N  H  S    S       C  S  P   P   M     A   (             I     A   (
 O  O  L    T       P  Y  C   C   E     R   S             O     R   S
 D  S  O    A       U  S  I   I   M     E   I                   E   I
 E  T  T    T          C  B   D   O     A   Z                   A   Z
 I          U          O  U   E   R         E                       E
 D          S          N  S   V   Y         )                       )
-------------------------------------------------------------------------------
 1  *  1    READY   A  *          0x14000000(0x4000000)             (       )
 2     1    FIRM    B       6   1  0xf0000000(0x4000000)    0xf7200000(0x20000)
-------------------------------------------------------------------------------
```

The command VXFabric may be used to exercise and to test the link between nodes.

```
[root@ki7]# vxfabric -n1 -N2 -F1 -S 0x0 -D 0x0 -c
Getting the SOURCE DMA buffer address (NODEID AREA) Size = 0x100000
Status of TARGET NODEID 1 = READY
 PCI MEM AREA addr  = 0x14000000
Initializing the SOURCE DMA buffer
 NODEID 1 offset  = 0x0 Getting the DESTINATION DMA buffer address (NODEID AREA) Size = 0x100000
Status of TARGET NODEID 2 = READY
 PCI MEM AREA addr  = 0xec000000
 NODEID 2, offset = 0x0
Perform DMA transfer Size = 0x100000
Write of 1048576 bytes in 2030 usec (516 MB/s)
Comparing Data: DATA Control OK
```

Refer to the man pages of VXFabric to get some more detailed information.

7.6.3 Low Level access

Low Level protocol over VXFabric is based on standard `read()` and `write()` to specific character device files `/dev/vx_rnodeX` (`/dev/vx_CPUA` or `/dev/vx_CPUB`). `read()` and `write()` system calls will be DMAs over VXFabric.

The system call `poll()` may be used to wait for interruptions over VXFabric and generated through a specific ioctl (`VXFABRIC_IOCTL_INTR_SEND`).

The `mmap()` is also provided to PIO access to the VXFabric memory area of a distant node.

The `write()` should be priviliged over the `read()` because it is much faster and less CPU.

The `lseek()` system call should be used to change the address in the reception buffer of the target for the next read or write. The reception buffer for VXFabric #0 is 64 MB on each board of the VXFabric.

There is also another 64 MB reception buffer for VXFabric #1 (between CPUA and CPUB).

With the BSP delivery, there are samples C code files which can be used and adapted to the customer application. Refer to `/usr/share/vxfabric/examples` repository.

> Known issue:

- `read()` from CPUB to CPUA is not working (only on VX6060).

7.6.4 IP over VXFabric

The User Programming model of VXFabric is based on the IP protocol and a socket layer API through an emulation of an Ethernet interface over PCIe like similar implementations of pseudo-Ethernet over virtualization boxes. This is the main warranty that the portability to VXFabric is straight forward for software developers.

The objective is to allow a user to move from Gbit Ethernet TCP/IP implementation to VXFabric avoiding low level complex and proprietary APIs (like most of sRIO implementation for example) with a breakthrough of performances, if compared with Gbit Ethernet.

VXFabric is available under Linux for VX6060 and VX303x and it is designed to be portable on other architectures and other operating systems. This implementation is scalable and will not require much effort to evolve towards PCIe gen3.

Furthermore, it does not require any other infrastructure than the VPX backplane and a VPX PCIe switch to interconnect VPX boards.

IP over VXFabric when it is setup, is disabling the "low level" protocole described in the previous section.

VXFabric Setup

Once the system slot1 VX6060 has booted Linux operating system, few configuration steps are required to get ready the Ethernet emulation over VXFabric.

The first step is to get some low level information about the VXFabric through the CLI command VXFabric:

```
client9# vxfabric -a
-------------------------------------------------------------------------------
 VX FABRIC 0 configuration
-------------------------------------------------------------------------------
 N  H  S    S       C  S  P   P   M     A   (             I     A   (
 O  O  L    T       P  Y  C   C   E     R   S             O     R   S
 D  S  O    A       U  S  I   I   M     E   I                   E   I
 E  T  T    T          C  B   D   O     A   Z                   A   Z
 I          U          O  U   E   R         E                       E
 D          S          N  S   V   Y         )                       )
-------------------------------------------------------------------------------
 1  *  1    READY   A  *          0x14001000(0x3fff000)             (       )
 2     1    READY   B       6   0  0xa0000000(0x4000000)    0xf5c00000(0x20000)
 5     3    READY   A      16   0  0xb8000000(0x4000000)    0xf6000000(0x20000)
 6     3    READY   B      15   0  0xb4000000(0x4000000)    0xf5f00000(0x20000)
11     6    READY   A       7   0  0xa8000000(0x4000000)    0xf5d00000(0x20000)
12     6    READY   B       8   0  0xac000000(0x4000000)    0xf5e00000(0x20000)
-------------------------------------------------------------------------------
```

After the boot of all the boards on the VXFabric, all the boards reach the READY status. The `vxeth` module, dealing with the Ethernet emulation, is not loaded by default to avoid the resources to be stolen when they are not needed. To load it:

```
client9# modprobe vxeth
```

This leads to the creation of two new pseudo-ethernet interfaces: `vxeth0` and `vxeth1` which is similar to a standard ethX Ethernet interface. Those interfaces are dealing with the VXFabric #0 for the backplane and VXFabric #1 for the local communication between CPUA and CPUB (only on the VX6060).

The `vxeth0` is dedicated to the backplane VXFabric and `vxeth1` should be used to communicate between CPUA and CPUB of the VX6060.

```
client9# ifconfig vxeth0
vxeth0      Link encap:Ethernet  HWaddr 02:00:00:00:00:01
            BROADCAST MTU:65500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Those pseudo-Ethernet interfaces should be managed like the other Ethernet interfaces and a local sub-network address could be allocated to this interface. For example:

```
client9# ifconfig vxeth0 192.168.20.1
```

This is not different from other Ethernet interfaces configuration and, for example, on Fedora distribution, the service «network» can be used to setup this interface through the configuration file `/etc/sysconfig/network-scripts/ifcfg-vxeth0`.

On another node, the same previous steps should be performed with another IP address on the same local sub-network address.

```
client5# ifconfig vxeth0 192.168.20.2
client5# ifconfig vxeth0
vxeth0     Link encap:Ethernet  HWaddr 02:00:00:00:00:07
           inet addr:192.168.20.2  Bcast:192.168.20.255  Mask:255.255.255.0
           inet6 addr: fe80::ff:fe00:9/64 Scope:Link
           UP BROADCAST   MTU:65500  Metric:1
           RX packets:4803815 errors:0 dropped:0 overruns:0 frame:0
           TX packets:7190431 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:367044680 (350.0 MiB)  TX bytes:92151086 (87.8 MiB)
```

It is possible to control the status of the interfaces through the `netstat` command:

```
client5# netstat -i
Kernel Interface table
Iface           MTU Met    RXOK  RXERR  RXDRP   RXOVR    TXOK TXERR TXDRP TXOVR Flg
bond0      1500 0   382872      0           0          0   391990      0          0          0 BMmRU
eth0       1500 0   204225      0           0          0   206587      0          0          0 BMsRU
eth1       1500 0   178647      0           0          0   185403      0          0          0 BMsRU
eth2       1500 0        0      0           0          0        0      0          0          0 BMU
vxeth0    65500 0  5372700      0           0          0 18477441      0          0          0 BMRU
lo        16436 0       36      0           0          0       36      0          0          0 LRU
```

The hosts IP addresses of the previous nodes could be added in the `/etc/hosts` file to be populated.

```
client5# more /etc/hosts
# hostname vx6060 added to /etc/hosts by anaconda
127.0.0.1                   localhost        localhost.localdomain     localhost4
localhost4.localdomain4
192.168.1.1     server
192.168.20.1     node1
192.168.20.2     node2
192.168.20.3     node3
192.168.20.4     node4
192.168.20.5     node5
192.168.20.6    node6
```

```
### netconfig updates ####
192.168.1.12         client12
192.168.1.11         client11
192.168.1.10         client10
192.168.1.7 client7
192.168.1.6 client6
192.168.1.5 client5
192.168.1.4 client4
192.168.1.3 client3
192.168.1.2 client2
192.168.1.9 client9
192.168.1.8 client8
### netconfig ends ####
```

The interconnection could be tested through the ping command:

```
client5# ping node2
PING node2 (192.168.20.2) 56(84) bytes of data.
64 bytes from node2 (192.168.20.2): icmp_seq=1 ttl=64 time=0.022 ms
64 bytes from node2 (192.168.20.2): icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from node2 (192.168.20.2): icmp_seq=3 ttl=64 time=0.020 ms
64 bytes from node2 (192.168.20.2): icmp_seq=4 ttl=64 time=0.019 ms
64 bytes from node2 (192.168.20.2): icmp_seq=5 ttl=64 time=0.022 ms
```

When every node on the VXFabric is configured with a valid internet address, the CLI command VXFabric output should be:

```
client9# vxfabric -a
-------------------------------------------------------------------------------
 VX FABRIC 0 configuration
-------------------------------------------------------------------------------
 N  H  S    S       C  S  P   P   M     A   (             I     A   (
 O  O  L    T       P  Y  C   C   E     R   S             O     R   S
 D  S  O    A       U  S  I   I   M     E   I                   E   I
 E  T  T    T          C  B   D   O     A   Z                   A   Z
 I          U          O  U   E   R         E                       E
 D          S          N  S   V   Y         )                       )
-------------------------------------------------------------------------------
 1  *  1  CONNECT   A  *          0x14001000(0x3fff000)             (       )
 2     1  CONNECT   B      6   0  0xa0000000(0x4000000)    0xf5c00000(0x20000)
 5     3  CONNECT   A     16   0  0xb8000000(0x4000000)    0xf6000000(0x20000)
 6     3  CONNECT   B     15   0  0xb4000000(0x4000000)    0xf5f00000(0x20000)
11     6  CONNECT   A      7   0  0xa8000000(0x4000000)    0xf5d00000(0x20000)
12     6  CONNECT   B      8   0  0xac000000(0x4000000)    0xf5e00000(0x20000)
-------------------------------------------------------------------------------
```

- The broadcast addressing is emulated by software
- The MTU is set by default to 65500 bytes to get benefit of the maximum bandwidth

Standard Network Services

The standard network services are supported «de facto» by the VXFabric due to the Ethernet emulation. After the setup is done as described in the previous section and modulo specific configuration and setup of each service, the major network services are available: ftp, telnet, ssh, NFS, scp...

Example:

```
client5# ssh node3
Warning:  Permanently added the RSA host key for IP address ´192.168.20.3´ to the
list of known hosts.
Last login: Thu Nov 25 11:04:51 2010 from server
client7#
client7# scp /boot/vmlinuz2.6.32.1410176.vx6060.fc12.i686.PAE node2:/tmp
Warning:  Permanently added the RSA host key for IP address ´192.168.20.1´ to the
list of known hosts.
vmlinuz2.6.32.1410176.vx6060.fc12.i686.PAE                100% 3565KB   3.5MB/s
00:00
client7#
logout
client5# ll /tmp
total 3568
rwxrxrx   1   root   root   3650176   20101125   16:04   vmlinuz2.6.32.14
10176.vx6060.fc12.i686.PAE
```

The `iperf` tool could be used to evaluate and measure the performances. Refer to the following sections.

7.7 Multinodes Diskless

Multinodes Diskless is a powerfull set of python scripts to help the configuration of diskless root filesystems and to boot distant nodes through the Ethernet network (PXE).

The main command is `diskless`:

```
[root@ki7] diskless --help
```

There are two xml files to configure the diskless system:

`/etc/diskless/images.xml`: describes the layers, the nodes and main configuration setup.

`/etc/diskless/diskless_fs.xml`: lists the rpms used to build the root filesystem of the distant nodes;

Have a look at the section 5.2 - CPUB Diskless Installation page 19 to get an example of diskless usage.

The user manual is available at: `/usr/local/share/doc/diskless/pdf/manual.pdf`

If diskless is not needed, you can disable this service at boot time :

```
[root@ki7]# chkconfig diskless off
```

7.8 e1000e Module

For VX3035, the BSP installer installs an updated package for this module, because the 82579 onboard Ethernet chipset is not supported by the e1000e standard kernel module

This updated e1000e package has been built for the kernel supplied with the BSP. The source rpm (in /usr/share/<board>_bsp/bsp_<ID>/SRPMS) may be used to rebuild it for another kernel flavour.

In any case, if a new version is installed, add it to the initrd image :

```
[root@ki7]# dracut --force
```

7.9 Sysvartool

To get the report of the PBIT (Power Built In Test), run the command:

```
[root@ki7]# sysvartool -A pbit -l
VX6060 detected
 area = 2, arch = 2
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
    ether_loop0: PASSED (FAILED ONCE)
    ether_loop1: PASSED (FAILED ONCE)
    ether_loop2: PASSED
         system: PASSED

PASSED  : 10
FAILED  : 0
NOT RUN : 0
TOTAL   : 10

POSTs configured to run automatically from RAM:

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

POSTs configured to run automatically from ROM:

PASSED  : 0
FAILED  : 0
NOT RUN : 0
TOTAL   : 0
```

7.10 GPIOs

There are 5 GPIOs on VX6060 (only from CPUA), VX3030, VX3035, and 8 on VM6050. When the GPIO driver is loaded, a collection of special sys files is created for each GPIO:

`/sys/class/cpld-gpio/gpioX/direction` should be set to `in` or `out`:

```
[root@ki7]# echo out > /sys/class/cpld-gpio/gpio4/direction
[root@ki7]# cat /sys/class/cpld-gpio/gpio4/direction
out
```

`/sys/class/cpld-gpio/gpioX/value` to get or set (depending on the direction) the value of the GPIO: `1` or `0`

```
[root@ki7]# echo 0 > /sys/class/cpld-gpio/gpio4/value
[root@ki7]# cat /sys/class/cpld-gpio/gpio3/value
1
```

`/sys/class/cpld-gpio/gpioX/value_it` get or set the value. A read to this file is waiting for an interruption on the GPIOs (refer below).

`/sys/class/cpld-gpio/gpioX/interrupt` enables or disables interruption on the GPIO: `on` or `off`

```
[root@ki7]# echo on > /sys/class/cpld-gpio/gpio3/interrupt
```

`/sys/class/cpld-gpio/gpioX/polarity` to set the polarity of the interruption: `hi` (high level or rising edge) or `lo` (low level or falling edge).

`/sys/class/cpld-gpio/gpioX/mode`: `edge` or `level` interruption mode.

```
[root@ki7]# cat /sys/class/cpld-gpio/gpio5/mode
edge
```

`/sys/class/cpld-gpio/gpio3/toggle` set toggle mode means the interruption is raised when the GPIOs state changes.

```
[root@ki7]# echo on > /sys/class/cpld-gpio/gpio3/toggle
```

`/sys/class/cpld-gpio/gpio3/intr_stat` is a counter of the interruption on that GPIOs.

```
[root@ki7]# cat /sys/class/cpld-gpio/gpio5/intr_stat
12
```

The GPIO2 is dual function because it is also a Masquable Reset. This is configurable through CPLD registers:

To set the GPIO2 as a standard GPIO (without issuing a local reset):

- on a VPX board :

```
[root@ki7]# cpldtool -f VPX_CONTROL MSKR2LOC 0
```

- on a VME board :

```
[root@ki7]# cpldtool -f VME_CONTROL MSKR2LOC 0
```

- or on all boards :

```
[root@ki7]# cpldtool -f 0x70 MSKR2LOC 0
```

This can also be done with the lower level command (but take care not to change some other bits) :

```
[root@ki7]# port 0x870
@0x870 = 0x1d
[root@ki7]# port 0x870 0xd
@0x870 <- 0x0d
[root@ki7]# port 0x870
@0x870 = 0x0d
```

7.11 cpldtool

`cpldtool` may be used to get some high level information about the setting of the CPLD.

This tool also has a write mode to change the settings, but this mode should not be used, except for a very small set of settings and with a lot of caution.

```
[root@ki7]# cpldtool -a
Reg 0x0 - CPLD_ID = 0x0B
              CPLD_ID=0x0
              CPLD_Debug=0x0
              CPLD_Version=0xB
Reg 0x1 - PCB_ID = 0x50
Reg 0x2 - FIRM_PWON = 0x00
Reg 0x3 - PWON_STATUS = 0x00
Reg 0x4 - PWR_RST_CONFIG = 0x21
              PWRON_MODE=0x0
              PLTRST_to_PERST3U_INHIB=0x0
              Alarm_inhib=0x1
              PLTRST_to_PERST6U_INHIB=0x0
              Reset_Mode=0x0
              PERSTb_Control=0x0
              Software_Cross_Reset=0x1
Reg 0x5 - VPD_BC1 = 0x00
Reg 0x6 - VPD_BC2 = 0x00
Reg 0x7 - VM6050_SERIAL_LINES_CTL = 0x00
              SERIAL2_Full=0x0
              SERIAL2_Term=0x0
              SERIAL1_Full=0x0
              SERIAL1_Term=0x0
Reg 0x8 - VM6050_PCI_MODE = 0x0C
              PCI_BUSMODE=0x1
              PCI_M66EN=0x1
Reg 0x9 - MEM_PROTECT = 0x00
              Boot_flash_CS_swap_DIP=0x0
              Boot_flash_CS_swap_Valid#=0x0
              Boot_both_flash=0x0
              USER_WP=0x0
              SYS_WP=0x0
              VPD_WP=0x0
              NVMRO=0x0
Reg 0xa - COM_TX = 0x00
              Semaphore=0x0
              Data=0x0
Reg 0xb - COM_RX = 0x00
Reg 0xc - BOARD_CONFIGURATION = 0x00
              Port_D_Mode=0x0
              Port_B_Mode=0x0
              ESW_LAN_CON=0x0
              HANK_LAN_CON=0x0
              USB_SW_SEL=0x0
              SERIAL1_Mode=0x0
```

```
              SERIAL2_Mode=0x0
              SERIAL2_cfg=0x0
Reg 0xd - PCI_MODE = 0x0E
              DRAM_DIMM=0x0
              DRAM_RANK=0x0
              DRAM_SIZE=0x0
              PCI_BUSMODE=0x1
              PCI_M66EN=0x1
              PCIXCAP=0x0
              PCISEL100=0x0
Reg 0xe - DIP_SWITCH_STATE = 0xF2
              USER_DIP=0xF
              SP2=0x0
              SP1=0x1
              FACTORY_MODE=0x0
Reg 0xf - SERIRQ_CONTROL = 0xE0
              SMB_MSK=0x1
              SMB_TEMP_MSK=0x1
              TIP_MSK=0x1
              GPIOs_INT=0x0
              SMB_INT=0x0
              SMB_TEMP_INT=0x0
              TIP_INT=0x0
              WDG_INT=0x0
Reg 0x5b - ALERT_STATUS = 0x01
              CATERR=0x0
              THERMTRIP=0x0
              THERM_PROT=0x0
              PROCHOT=0x0
              TEMP_ALERT=0x0
              DDR_Throttling=0x1
Reg 0x6a - GEO_ADD = 0x3D
              SYSCON=0x0
              Error=0x0
              GAP=0x1
              GA=0x1D
Reg 0x70 - VME_CONTROL = 0x5D
              Smb_Alert_Status=0x1
              MSKR2LOC=0x1
              VME_RST=0x1
              VME2LOC=0x0
              LOC2VME=0x1
Reg 0x72 - I2C_BOARD_STATUS = 0x94
              Power_Status=0x1
              Reset_Source=0x0
              Reset_Status=0x1
              Boot_Status=0x4
Reg 0x73 - I2C_BOARD_CONTROL = 0x50
              Board_Id=0x5
              Check_Errors=0x0
              Reset_3UB=0x0
              Reset_3UA=0x0
              Power_OnOff=0x0
[root@ki7]#
```

7.12 I2C Busses

The `cpld_i2c` driver is supporting the local i2c bus (I2C bus number 6) which is local to the board and used to address sensors and system eeproms.

There are also two other I2C busses routed to the backplane; the bus numbers are 7 and 8:

- `/dev/i2c-7` is the SMB bus connecting it to the I2C devices in the chassis (if any).
- `/dev/i2c-8` is the IPMB bus connecting all boards through the backplane

By default, those two busses are available only on the system controller.

It is possible to enable those busses on other slots by adding the option:

`force_i2c_extend=1` at the load of the module "cpld" for example in `a /etc/modprobe.d/ki7.conf file` adding the line:

```
options cpld force_i2c_extend=1
```

7.13 BIOS Update

The `flashrom` package provides a set of commands and scripts to update the BIOS of the boards.

To update the BIOS on one side (CPUA or CPUB) of the VX6060, VX303x or VM6050, use the command `ki7updbios`:

Help command:

```
[root@ki7]# ki7updbios -h
usage:

/usr/local/sbin/ki7updbios <options> <bios image>
Script to update the BIOS on boards using flashrom tools.\n
Options:\n
-r Read the current BIOS in bios image file.\n
-w Update the BIOS image using the given bios image file.\n
-a Update the all BIOS eeprom.\n
[root@ki7]#
```

Read the current BIOS:

```
[root@ki7]# ki7updbios -r BIOS.bin
flashrom v0.9.4 on Linux 2.6.32.14-11035.vx6060.fc12.i686.PAE (i686), built with libpci 3.1.7,
GCC 4.4.2 20091027 (Red Hat 4.4.2-7), little endian
flashrom is free software, get the source code at http://www.flashrom.org

Calibrating delay loop... OK.
No coreboot table found.
Found chipset "Intel QM57", enabling flash write... OK.
This chipset supports the following protocols: FWH,SPI.
Found chip "SST SST25VF032B" (4096 KB, SPI) at physical address 0xffc00000.
Reading flash... done.
[root@ki7]# ls -ltr BIOS.bin
-rw-r--r-- 1 root root 4194304 2011-03-03 15:54 BIOS.bin
```

Write a new BIOS file:

```
[root@ki7]# ki7updbios -w VX6060_ID12269.bin
flashrom v0.9.4 on Linux 2.6.32.14-11035.vx6060.fc12.i686.PAE (i686), built with libpci 3.1.7,
GCC 4.4.2 20091027 (Red Hat 4.4.2-7), little endian
flashrom is free software, get the source code at http://www.flashrom.org

Looking for "bios"... found.
Calibrating delay loop... OK.
No coreboot table found.
Found chipset "Intel QM57", enabling flash write... OK.
This chipset supports the following protocols: FWH,SPI.
Found chip "SST SST25VF032B" (4096 KB, SPI) at physical address 0xffc00000.
Flash image seems to be a legacy BIOS. Disabling checks.
Erasing and writing flash chip... Done.
Verifying flash... VERIFIED.
```

Take care that if the system is booted from the RESCUE BIOS flash (refer to the BIOS user manual), the update of the BIOS using the **ki7updbios** command will update the RESCUE BIOS Flash which is not recommended.

The update of the BIOS through the **ki7updbios** command does not preserve the setup parameters of the BIOS. For deployement of a BIOS version with its own setup parameters, preset one board with the desired parameters, backup this using **ki7updbios -r**, and use this version of BIOS+ setup to be deployed on other similar boards.

7.14 FMRAM Example

The `fmram` packages gives a example of how to access to the FMRAM device which may be used to save some customer data which need to be backuped.

```
[root@ki7]# fmram -h
```

Usage `fmram` [options]
a tool to read or write the FerroMagnetic RAM

Options are :

`-h`	: this help
`-r <value>`	: read at offset <value> (default 0)
`-w <value>`	: write at offset <value> (default 0)
`-s <value>`	: data size to read or write (default 4)
`-f <filename>`	: file name used to store (optional) or read (mandatory) data

Example:
`fmram -w 0x10 -s 0x11 -f data_file` :
store 17 bytes read from data_file to the ferromagnetic RAM at offset 16

With src package, the C code of this command is delivered as an example.

7.15 VX6060: Setup Serial Console of CPUB on CPUA

It is possible to get the serial console of the CPUB available through minicom over an xterm on the graphic display of the CPUA.

At BIOS level:

- Enable the COM1 console redirection on CPUB
- Disable the COM1 console redirection on CPUA

Then Boot CPUA.

- Setup the CPLD to get the console of CPUB routed to CPUA, by setting bit0 of register 0x80c

```
[root@ki7]# port 0x80c
@0x80c = 0x08
[root@ki7]# port 0x80c 0x9
```

- Install and setup minicom to get the prompt of the CPUB console: on CPUA, the CPUB console is `/dev/ttyS1` in 115200 bauds rate.

It is possible to have the serial console of CPUB directly available through the front panel console connector. Refer to the Hardware User Guide. It is also possible to get a graphic display on the CPUB through the backplane interface using PIM module.

7.16 VM6050: VME Toolkit

ALMA2f is the VME Bridge of the VM6050 boards.

The associated driver is `almavme`. Refer to the manual page of the `VME Toolkit` to get the API of this driver.

The `almavmechan` command (installed through the addons RPM) provides some way to manage master and slave VME channels from user space (through `ioctls` of the ALMA2f driver).

Example 1: Open a slave VME channel (VME->PCI)

Open a slave VME channel (VME->PCI) on the target decoded at 0x1000000 on the VME for all the AMs matching 0x2FF to access the local memory (8 MB size) at physical address given by `allocator`:

```
[root@ki7]# cat /proc/allocator
Info on Allocator module [0x19000000-0x1cffffff]
Partition 2ESSTIN [ 0x19000000-0x19ffffff ]
Partition VMEIN [ 0x1a000000-0x1affffff ]
Partition 2ESSTOUT [ 0x1b000000-0x1bffffff ]
Partition VMEOUT [ 0x1c000000-0x1cffffff ]
```

with a `/etc/init.d/allocator` setting: `OPTIONS='linsizemb=64 allocator_part="16M(2ESSTIN):16M(VMEIN):16M(2ESSTOUT):16M(VMEOUT)"'`

```
[root@ki7]# almavmechan -pcialloc TEST 0x10000000 0x1a000000 0x800000 0x2FF013
[root@ki7]# almavmechan -pciprint
  ------------------------------------------------------------------------------
VME to PCI CHANNELS
  ------------------------------------------------------------------------------
Num     Name        VMEaddr       addr      Space size(MB)        AM     Conv   WP  RH
  ------------------------------------------------------------------------------
0      TEST      0x10000000 0x1a000000 DRAM         0008        D-02-ff ADDR Yes Yes
  ------------------------------------------------------------------------------
```

On the VM6050, the VME bus should be locked before issuing a single VME access concurrently with VME DMA accesses. Refer to the option `-b` of `mbm3k` command, and to the ioctl `VMEIOCTL_GET_VME` and `VMEIOCTL_FREE_VME` of the ALMA2f driver. VME DMA transfers are preferred to single VME transfers.

Example 2: Open a master VME channel (PCI->VME)

Open a master VME channel (PCI->VME) to access a VME address at 0x20000000 in A32SData mode:

```
[root@ki7]# almavmechan -vmememalloc TEST2 0x20000000 0x800000 0x0
[root@ki7]# almavmechan -vmememprint
 ------------------------------------------------------------------------------
                            PCIMEM to VME CHANNELS
 ------------------------------------------------------------------------------
Name     CPUaddr         PCIaddr       VMEaddr        size(Mb)       AM      Conv   WP   RH
 ------------------------------------------------------------------------------
TEST2      0xc0800000 0x00800000 0x20000000            0008       A32SDATA ADDR No  No
 ------------------------------------------------------------------------------
```

Then the 0xc0800000 address can be remapped to access the VME window at 0x20000000 on the VME bus (with A32SDATA address modifier). Remapping may be done through the `mmap call of /dev/almavme`.

On the VM6050, the VME bus should be locked before issuing a single VME access concurrently with:

- VME DMA accesses. Refer to the option `-b` of `mbm3k` command, and to the `ioctl`
- VMEIOCTL_GET_VME and VMEIOCTL_FREE_VME of the ALMA2f driver. So VME DMA transfers are preferred to single VME transfers.

Example 3: `mbm3k` command

The `mbm3k` command may be used to exercise some VME DMA transfers. For example to run 100 loops of a DMA read in 2eSST mode at the VME address 0x10000000, VME block size is 128 and the global transfer size is 0x600000:

```
[root@ki7]# mbm3k dma2esst r 128 0x10000000 0x600000 100
```

Example 4: Open a slave 2eSST VME channel

Open a slave 2eSST VME channel at 0x10000000 on the VME bus. In that example and if the distant board is a VM6050 running this Linux distribution, the command `almavmechan` may be used to open such a slave 2eSST VME channel:

```
[root@ki7]# almavmechan -pcialloc 2ESST 0x10000000 0x0 0x800000 0x06000003
```

Example 5: Send and receive VME interrupts

Refer to the source examples delivered in `/usr/share/vmetools/src/intr/` directory.

Enter following commands to generate the `RECV_intr` and `GEN_intr` programs.

```
[root@ki7]# cd /usr/share/vmetools/src/intr
[root@ki7]# make
```

To test the VME interrupts; for example, using two VM6050 boards:

- on one VM6050 board, enter next command to wait for the VME interrupt level 5 and vector 0xa0:

```
[root@ki7]# RECV_intr 0x5 0xa0
```

- on the other VM6050 board, enter next command to generate the VME interrupt level 5 and vector 0xa0:

```
[root@ki7]# GEN_intr 0x5 0xa0
```

The VME bus must be locked during the acknowledgement of the VME interrupts, this is done in the interrupt handler of the ALMA2f driver. The time to lock/unlock of the VME bus depends on the VME bus load.

Chapter 8 - Additional Information

8.1 Known Limitations

Boot Process

During the boot process, the following messages related to ACPI are displayed (or saved in the boot log):

```
ACPI Error: No handler for Region [ECF2] (ffff880110648ea0) [EmbeddedControl]
(20100428/evregion-369)
ACPI Error: Region EmbeddedControl(0x3) has no handler (20100428/exfldio-293)
ACPI Error (psparse-0537): Method parse/execution failed [\_SB_.LID0._LID] (Node ffff88011064c600),
AE_NOT_EXIST
ACPI: resource 0000:00:1f.3 [io  0xe000-0xe01f] conflicts with ACPI region SMBI [mem
0x0000e000-0x0000e00f disabled]
ACPI: This conflict may cause random problems and system instability
ACPI: If an ACPI driver is available for this device, you should use it instead of the native driver
```

Also some I2C probe messages are displayed for I2C addresses where no device is found :

```
No ACK after 1 retries on device @0xXX
```

Those warnings and errors do not impact the system behaviour.

BSP installation

During the BSP installation, the following messages related to ACPI are displayed (or saved in the boot log):

```
Installing     : sensors_addons-1.4-11208.x86_64                          37/43
ACPI: resource 0000:00:1f.3 [io  0xe000-0xe01f] conflicts with ACPI region SMBI [io  0xe000-0xe00f
disabled]
ACPI: If an ACPI driver is available for this device, you should use it instead of the native driver
Restarting lm_sensors service ...
Stopping lm_sensors: [  OK  ]
Starting lm_sensors: loading module cpld_i2c i2c_i801 lm73 ads7830 [  OK  ]
```

Those warnings do not impact the system behaviour.

VX6060 VX6060 I2C devices are not addressable from CPUA and CPUB at the same time - CRP 3911

It is recommended not to read the board temperature sensors (through the sensors command) neither to access the FMRAM (`fmram` command) from the CPUA and from the CPUB in the same time. The read data would be unpredictable.

VM6050 DRM driver does not start when the option nodac is set in the kernel bootline (Only with 64 bit kernel) - CRP 4060

To fix the PMC issue "VM6050: DMA issues using 32b PMC slot with the Fedora 14 x86_64 (64b) - CRP4024", the option `"iommu=soft,nodac"` has been added in the bootline. This one produce a regression on the DRM driver (black screen at boot time).

Workaround:

Remove the nodac option in `/etc/grub.conf` (and the first time at the grub menus editing the kernel command line) at the definition of the kernel command line if no PMC in the 32b PMC slot (the right's one).

Or use the Fedora 14 32b distribution.

8.2 Fixed Bug in the Current Release

TSC (Timestamp Counter) - CRP 3883

The TSC (Timestamp Counter) may drift too much to be a trusted time base for the system.

Fixed in kernel ⩾2.6.32.14-12269

VXFabric Failure During Setup - CRP 3885

Time to time, VXFabric may fail during setup due to "allocator" initialization failure. Reboot the system to get allocator and VXFabric ready.

Fixed in VXFabric ⩾ 1.5 and kernel ⩾ 2.6.32.14-12269 allocator is not used anymore with the previous packages versions.

Linux system failure during boot - CRP3910

Time to time, the Linux system may fail after a power on at boot time during the udev service startup; reboot the system to recover the system.

VM6050: VME deadlock during VME stress test - CRP4023

Mixing single access with dma from a VM6050 using the Fedora 14 BSP 2.3 (with `mbm3k` test tool for example) raises VME deadlock due to bad reservation of the VME bus.

This is fixed with the BSP 2.4.

VM6050: DMA issues using 32b PMC slot with the Fedora 14 x86_64 (64b) - CRP4024

Fedora 14 x86_64 kernel allows, wrongly, 64b DMAs on the 32b PMC slot. For example, using an ethernet PMC, this issue can lead to the following output:

```
[root@ki7]# [   76.721526] e1000 0000:06:04.0: eth-6-4-0: Detected Tx Unit
Hang
[   76.721528]   Tx Queue             <0>
[   76.721528]   TDH                 <0>
[   76.721529]   TDT                 <8>
[   76.721530]   next_to_use         <8>
[   76.721531]   next_to_clean       <0>
[   76.721532] buffer_info[next_to_clean]
[   76.721533]   time_stamp          <fffc907c>
[   76.721533]   next_to_watch       <0>
[   76.721534]   jiffies             <fffc97c0>
[   76.721535]   next_to_watch.status <0>
```

The fix is to add `iommu=soft`,nodac to the kernel boot command line (in `grub.conf`)

VX3035: kernel trace dump with 2.6.35.6-11245.ki7.fc14.x86_64 (BSP 2.3) - CRP4025

Time to time, about 2 times for 1000 boot/reboot loops, the following dump may appear with "dmesg" command.

```
[   24.972520] ------------[ cut here ]------------
[   24.972529] WARNING: at lib/list_debug.c:26 __list_add+0x3f/0x81()
[   24.972530] Hardware name: VX3035

[   24.972532] list_add corruption. next->prev should be prev
(ffffffff81a7ae20), but was ffffffffa024acf8. (next=ffffffffa0243cf8).
[   24.972534] Modules linked in: cpld_i2c cpld_wdt cpld_leds cpld sunrpc
cpufreq_ondemand acpi_cpufreq freq_table mperf ipv6 uinput i2c_i801 joydev
microcode e1000e igb dca vxdma ata_generic pata_acpi i915 drm_kms_helper drm
i2c_algo_bit i2c_core video output [last unloaded: scsi_wait_scan]
[   24.972550] Pid: 1749, comm: modprobe Not tainted
2.6.35.6-11245.ki7.fc14.x86_64 #1
[   24.972551] Call Trace:
[   24.972556]  [<ffffffff8104d7c1>] warn_slowpath_common+0x85/0x9d
[   24.972559]  [<ffffffff8104d87c>] warn_slowpath_fmt+0x46/0x48
[   24.972563]  [<ffffffff8122639a>] __list_add+0x3f/0x81
[   24.972567]  [<ffffffff81217df1>] module_bug_finalize+0xb9/0xca
[   24.972571]  [<ffffffff81028856>] module_finalize+0x156/0x165
[   24.972575]  [<ffffffff8107bfff>] load_module+0x1170/0x1b74
[   24.972578]  [<ffffffff81079981>] ? setup_modinfo_srcversion+0x0/0x29
[   24.972584]  [<ffffffff8111680a>] ? fsnotify_modify+0x6c/0x74
[   24.972587]  [<ffffffff8107ca53>] sys_init_module+0x50/0x1e4
[   24.972590]  [<ffffffff81009cf2>] system_call_fastpath+0x16/0x1b
[   24.972592] ---[ end trace 369013650390620b ]---
```

This is a kernel regression of the dynamic load of the modules.

This is fixed with the kernel rpm of BSP.

With the Fedora 14 (ID12117) 64b version, X11 hangs at boot time - CRP4053

There is a regression in the kernel 2.6.35.14-106 from Fedora related to the graphics driver of the Arrandale chipset (VM6050, VX3030, VX6060): X11 fails to start during the boot process.

This is fixed with the kernel rpm of BSP 2.5.

Support of VX3035-B: eeproms are on the low speed i2c bus - CRP4054

On the VX3035-B, the eeproms are on the low speed i2c bus (on the cpld). In the previous VX3035-A, the high speed bus was shared by the eeproms and the fram. On VX3035-B, only the fram is on the high speed bus. This requires an adaptation of the i2c bus driver to setup the right speed and to address the eeproms.

This is supported in BSP 2.5 ID 12269.

Support of VX3035-B: External RTC (RTC2) need a driver and a service - CRP4055

On VX3035-B, the standard RTC (in the PCH) may be replaced by an external RTC (RTC2) located on the SMBUS. It requires a driver and a service to properly manage this new device and to be able to change time and date at Linux command line.

This is supported in BSP 2.5 ID 12269.

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@kontron.com
support-kom-sa@kontron.com

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