SOFTWARE
RELEASE NOTES

Fedora 21 on VX305x

SD.DT.G53-1e - February 2017

POSSIBILITIES START HERE kontron

Fedora 21 on VX305x Release Notes

Disclaimer

Kontron would like to point out that the information contained in this manual may be subject to alteration, particularly as a result of the constant upgrading of Kontron products. This document does not entail any guarantee on the part of Kontron with respect to technical processes described in the manual or any product characteristics set out in the manual. Kontron assumes no responsibility or liability for the use of the described product(s), conveys no license or title under any patent, copyright or mask work rights to these products and makes no representations or warranties that these products are free from patent, copyright or mask work right infringement unless otherwise specified. Applications that are described in this manual are for illustration purposes only. Kontron makes no representation or warranty that such application will be suitable for the specified use without further testing or modification. Kontron expressly informs the user that this manual only contains a general description of processes and instructions which may not be applicable in every individual case. In cases of doubt, please contact Kontron.

Lise-Meitner-Str. 3-5

86156 Augsburg

Germany

www.kontron.com

REVISION HISTORY

PUBLICATION TITLE:	Fedora 21 on VX305x Release Notes	
DOC. ID:	SD.DT.G53-1e	
Revision	Brief Description of Changes	Date of Issue
1e	Updated sections: - Chapter 2 - Release Content - Chapter 4 - Required Configuration - Chapter 5 - Using the Live Image and Installing it on a Hard Disk News sections: - Chapter 7 - VxFabric - Chapter 8 - SM750 Graphical Chipset Support	02-2017
0e	Initial Issue	07-2016

Customer Support

Find Kontron contacts by visiting: http://www.kontron.com/support.

Customer Service

As a trusted technology innovator and global solutions provider, Kontron extends its embedded market strengths into a services portfolio allowing companies to break the barriers of traditional product lifecycles. Proven product expertise coupled with collaborative and highly-experienced support enables Kontron to provide exceptional peace of mind to build and maintain successful products.

For more details on Kontron's service offerings such as: enhanced repair services, extended warranty, Kontron training academy, and more visit http://www.kontron.com/support-and-services/services.

Customer Comments

If you have any difficulties using this manual, discover an error, or just want to provide some feedback, contact Kontron support. Detail any errors you find. We will correct the errors or problems as soon as possible and post the revised manual on our website.

SYMBOLS

The following symbols may be used in this manual:

⚠DANGER	DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.
⚠WARNING	WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.
⚠CAUTION	CAUTION indicates a hazardous situation which, if not avoided, may result in minor or moderate injury.
NOTICE	NOTICE indicates a property damage message.
	**Electric Shock!** This symbol and title warn of hazards due to electrical shocks (> 60 V) when touching products or parts of them. Failure to observe the precautions indicated and/or prescribed by the law may endanger your life/health and/or result in damage to your material.
	**ESD Sensitive Device!** This symbol and title inform that the electronic boards and their components are sensitive to static electricity. Care must therefore be taken during all handling operations and inspections of this product in order to ensure product integrity at all times.
	**HOT Surface!** Do NOT touch! Allow to cool before servicing.
	**Laser!** This symbol inform of the risk of exposure to laser beam from an electrical device. Eye protection per manufacturer notice shall review before servicing.
	This symbol indicates general information about the product and the user manual. This symbol also indicates detail information about the specific product configuration.
	This symbol precedes helpful hints and tips for daily use.

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

As a precaution and in case of danger, the power connector must be easily accessible. The power connector is the product's main disconnect device.

⚠CAUTION

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

⚠CAUTION

Caution, Electric Shock!
Before installing a non hot-swappable Kontron product into a system always ensure that your mains power is switched off. This also applies to the installation of piggybacks. Serious electrical shock hazards can exist during all installation, repair, and maintenance operations on this product. Therefore, always unplug the power cable and any other cables which provide external voltages before performing any work on this product.
Earth ground connection to vehicle's chassis or a central grounding point shall remain connected. The earth ground cable shall be the last cable to be disconnected or the first cable to be connected when performing installation or removal procedures on this product.

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 product 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 product.

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

In performing all necessary installation and application operations, only follow 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 product then re-pack it in the same manner as it was delivered.

Special care is necessary when handling or unpacking the product. See Special Handling and Unpacking Instruction.

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

Table Of Contents

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 21 BSP distribution on the Kontron VX305x boards.

In this document, the term VX305x is used for the VX305x board in standard or rugged conduction-cooled version:

- VX3052/VX3058 Single Slot 3U VPX Board
  - VX3052-SA and VX3058-SA Standard Commercial version
  - VX3052-RC and VX3058-RC Rugged Conduction-Cooled version

In this document, the term VX305x-SA will be associated to the 3U VPX board family, including VX3052 dual core and VX3058 octo core module.

The VX305x-SA supports the PB-VX3-400, a 3U VPX Rear Transition Module compliant with the definition of the Rear Transition Module on VPX standard - VITA 46.10

The VX305x board supports also the following RTM model :

- VX304x-RTM Rear Transition Module for the single-slot 3U VPX board

2 / Release Content

The release consists of a unique Linux Live Image: **Fedora21_KVX_BSP_ID16349_LiveImage.iso**

This distribution contains a whole Fedora 21 Workstation release as well as the BSP supporting the VX305x board.

This Fedora live image can be installed on a hard drive to have an installed system having the same set of packages than the live image.

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

- Kernel: Update of the kernel of Fedora 21 to support VX305x boards specifics features and to fix issues.

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 GPIOs of the VX305x board.
- Watchdog: Drivers to setup the Watchdogs of the board.
- 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.
- 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.

3 / Associated Documentation

Kontron Documentation

Hardware

- VX305x 3U VPX Computing Node User's Guide CA.DT.B25
- VX305x Hardware Release Notes CA.DT.B26

Firmware

- VX305x BIOS User Manual SD.DT.G50
- VX305x PBIT User's Guide SD.DT.G51

Fedora 21 Documentation

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

4 / Required Configuration

4.1 Hardware Requirements

- A Kontron VX305x board.
- The Fedora 21 release may be installed on one of the following bootable disks:
  - a SATA disk connected to the SATA connectors available on VX305x-RTM board.
  - an optional onboard SSD Flash Disk (present on bottom or top M.2 slot).
- An optional graphical module (present on bottom or top M.2 slot).
- A graphical display (with HDMI port interface), USB keyboard and USB mouse (for a graphics install)

  OR

  A console on serial line (text or VNC install).

For a graphic configuration on VX305x a specific order code and a specific graphic module are required: VX3058SA881151000 or VX3052SA88015100 and MODP-GM2-SM750 . The module provides an HDMI port.

4.2 Firmware Requirements

The version of the BIOS firmware must be at least:

- 16016

This version is displayed in the BIOS Setup.

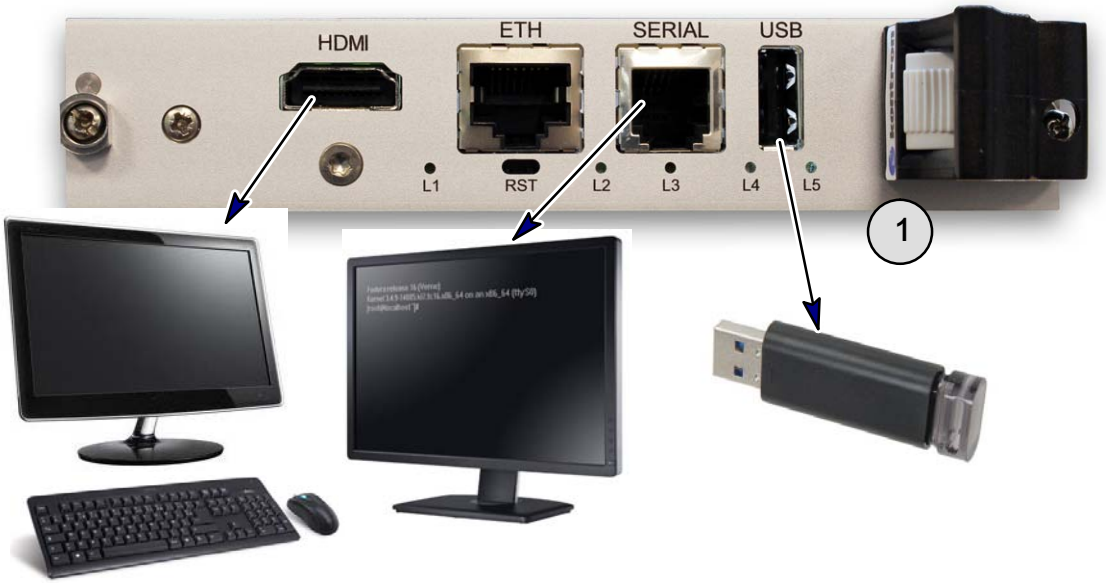
5 / Using the Live Image and Installing it on a Hard Drive

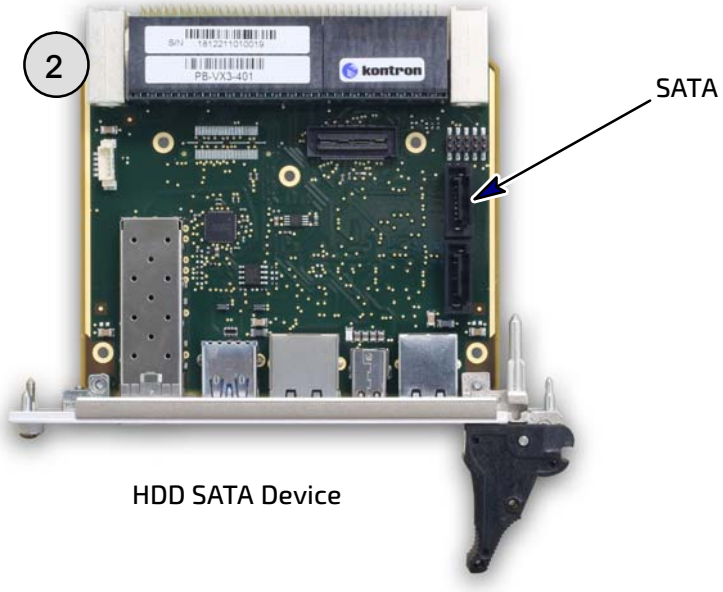
5.1 System Configuration

A USB stick is required as well as an HDD in case you plan to install the LIVE image on an HDD.

1. Plug the USB stick on the front panel USB connector.
2. If needed, plug the HDD SATA device on the SATA0-A connector of the Rear Transition Module (VX304x-RTM). or install a SSD device on the M.2 slot (bottom or top).

The live image "**FEDORA21_KVX_BSP16349_LiveImage.iso**" can be used in a graphical system environment or console text environment.





HDD SATA Device

5.2 Creating the Media containing the Live Image

Even if the image may be burnt on a media DVD, it is more recommended to use a USB stick (thanks to the hybrid format of the image).

Simply transfer the ISO live image to a USB stick by running command such as :

```
dd if=FEDORA21_KVX_BSP16349_LiveImage.iso of=/dev/<USB_DEV_NAME> bs=1M
```

where **<USB_DEV_NAME>** represents the name of the USB device : **sdb** for example.

5.3 Booting the USB Stick containing the Live Image

The resulting USB stick is bootable in a standard way in a serial console or a graphical environment.

Of course your configuration need to be equipped with a HDD if you plan to install the distribution in a second time.

Once booted from the resulting live media, the system is ready to use with all available BSP VX305x features.

As with standard Fedora Live images:

- There is an autologin as user "**liveuser**", and no password for user "**root**"
- When burning a USB stick with the methods above, the USB stick is non-persistent which means that every change made to the OS is lost at reboot.

If You want to make your system persistent, refer to section 5.5.

5.4 Installing the LIVE Image on an HDD

The live image offers the possibility to install the BSP on an hard drive.

In order to reach this goal, follow the steps below:

- Boot the USB stick containing the LIVE image, and login as root.
- Run the following command:

```
/root/install_to_disk.sh
```

The live media installation have the same behavior as if the installation was made from an installation media.

The only difference is that no package selection is required: all the packages present on the live media are installed.

So, in a standard way, the installation stage will consists of the following steps:

- Selecting the langage.
- Choosing the Installation Destination (Hard Disk on which the distribution will be installed).
- Optional selection of Time & Data, and Network & host name.

Then, click on "**Begin Installation**"

After a few minutes, the process should be over.

At this point you can click on "**Quit**" and shutdown your system.

Then :

- Remove the USB stick
- Power On the System

The system should automatically boot on the installed disk.

After booting the HDD, you can verify the version of the release by running:

```
# cat /etc/kvx_bsp-release
```

which should reveal:

```
Kontron VPX VME Board Support Package 1.1.3 [16349]
```

5.5 Binary and Source Packages related to the VX305x BSP

The list of the RPMs composing the VX305x BSP is:

```
ixgbe-4.3.13-1.x86_64.rpm
kernel-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-core-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-devel-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-headers-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-modules-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-modules-extra-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-tools-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-tools-debuginfo-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-tools-libs-3.18.3-15303.VX305x.fc21.x86_64.rpm
kernel-tools-libs-devel-3.18.3-15303.VX305x.fc21.x86_64.rpm
kvx_bsp-1.1.3-16349_kernel_3.18.3_15303.VX305x.fc21.x86_64.rpm
lm_sensors-3.3.5-6.fc21.x86_64.rpm
multinodes-diskless-1.2.9-16118.x86_64.rpm
sm750module-4.1.11-16349.x86_64.rpm
vxfabric-1.1.16-16118_kernel_3.18.3_15303.VX305x.fc21.x86_64.rpm
```

By default all these packages are installed to the HDD during the installation process.

If you want to obtain the binary packages, or the source packages associated to these binary packages , contact the Kontron's Technical support at the address: support@kontron.com

6 / Fedora System Configuration

In this chapter, information related to some specific configuration items of the Fedora 21 system are detailed.

6.1 Network

According to the following **udev** file: **/etc/udev/rules.d**, the network interfaces are named in the following way:

- **eth0**: related to the first 10 Gb network interface.
- **eth1**: related to the second 10 Gb network interface.
- **eth2**: related to the front panel 1 Gb network interface.
- **eth3**: related to the rear 1 Gb network interface.

In case you prefer to use the default naming convention for the network interfaces, simply remove the **udev** rules file: **/etc/udev/rules.d/75-kvx_bsp.rules** and reboot the board in order to take into account the changes.

6.2 Firewall

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

- Run

```
systemctl disable firewall
```

6.3 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 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**

6.4 RTC Date Configuration

In order to remain the system date, the VX305x provides an RTC based on I2C device named **pcf8563**.

To use this RTC the following service must be started by the root:

```
# systemctl enable rtc2
# systemctl start rtc2
```

The device **/dev/rtc1** should be created.

For changing the system date and update the RTC device the system date must be initialized as follows:

```
# date 062013262000
# hwclock --systohc -f /dev/rtc1
```

The date can be verified with the following command:

```
# hwclock --show -f /dev/rtc1
```

After reboot, the date is available with the standard '**date**' command.

7 / VXFabric™

Kontron VXFabric™ is an open infrastructure which implements efficient inter board communication at hardware speed. The architecture is compliant with the OpenVPX standard (VITA 65) which defines two main hardware topologies of the backplane: distributed and centralized topologies.

To get more information about this software, go to the www.kontron.com web site and enter the key word "vxfabric" into the search engine.

8 / SM750 Graphical Chipset Support

The SM750 Silicon Motion chipset is a PCI Express display controller device.

The SM750 embedded graphics features include:

- 2D acceleration,
- 16 MB integrated video memory.
- The maximal supported resolution is: 1920x1080.

This device is fully supported by this Linux release .

This support is based on an Xorg modules: **smiddk750_drv.so** and **siliconmotion_drv.so** (present under **/usr/lib64/xorg/modules/drivers** directory)

Refer to the Silicon Motion User's Manual for more information, by running the following command:

```
# man siliconmotion
```

9 / BSP Specific Features

9.1 Sensors

9.1.1 Sensors Overview

To display sensors information, run the sensors command:

```
# sensors
```

On VX305x board, the output should look like:

```
coretemp-isa-0000
Adapter: ISA adapter
Physical id 0:  +37.0°C  (high = +82.0°C, crit = +104.0°C)
Core 0:         +37.0°C  (high = +82.0°C, crit = +104.0°C)
Core 1:         +37.0°C  (high = +82.0°C, crit = +104.0°C)
Core 2:         +37.0°C  (high = +82.0°C, crit = +104.0°C)
Core 3:         +37.0°C  (high = +82.0°C, crit = +104.0°C)
Core 4:         +37.0°C  (high = +82.0°C, crit = +104.0°C)
Core 5:         +37.0°C  (high = +82.0°C, crit = +104.0°C)
Core 6:         +37.0°C  (high = +82.0°C, crit = +104.0°C)
Core 7:         +37.0°C  (high = +82.0°C, crit = +104.0°C)

nct7802y-i2c-22-28
Adapter: I2C CPLD adapter
 3V3_SB Voltage :  +3.33 V  (min =  +3.14 V, max =  +3.46 V)
VCORE  Voltage :  +1.81 V
 DDR4   Voltage :  +1.21 V  (min =  +1.14 V, max =  +1.26 V)
 +12V   Voltage : +12.08 V  (min = +11.40 V, max = +12.61 V)
 +5V    Voltage :  +5.09 V  (min =  +4.87 V, max =  +5.25 V)
  GPIO 1        :  +0.00 V
 LTD temp       :  +30.0°C  (low  = -45.0°C, high = +85.0°C)
                            (crit = +95.0°C)

ltc2945-i2c-0-6e
Adapter: SMBus I801 adapter at f000
 Current :    +1.71 A  (min =  +0.00 A, max =  +7.02 A)
                       (lowest =  +0.03 A, highest =  +4.36 A)
```

The sensor command reveals the presence of low, high and critical thresholds. When the temperature of one of the core (coretemp sensors) goes beyond the critical threshold, an automatic reset of the board will occur.

When the temperature or the voltage goes beyond one of the limits low and high, an explicit alarm message will occur in the sensors command output. So, in order to track down this kind of event, run the following command:

```
[root@]# sensors | grep ALARM
```

On the other hand, it is possible to synchronize the execution of an application on a high limit temperature event.

For example, at first, create a file name **/tmp/test.sh** containing:

```
cd /sys/devices/platform/cpld_i2c.6/i2c-22/22-0028/
cat < temp1_max_alarm_intr
echo ´Nuvoton local temp went beyond the high limit´
shutdown no
```

Make it executable and run it:

```
[root@]# chmod +x /tmp/test.sh
[root@]# /tmp/test.sh
```

As a consequence, in case the nuvoton temperature goes beyond the high limit temperature, this shell script will automatically cause a shutdown of the board.

9.1.2 Temperature Sensors Threshold and Alarm Report

The Nuvoton NCT7802Y have some minimum and maximum thresholds configured by the BIOS.

Linux drivers present in this distribution for these temperature sensors, implement a polling mechanism to report alarms.

Alarms are set by the driver, and reported by '**sensors**' command.

When the temperature fetches on one of these sensor devices is lower than the minimum threshold the alarm is set and the '**sensors**' command report '**ALARM (LOW)**'.

When the temperature fetches on one of these sensor devices is higher than the maximum threshold the alarm is set and the '**sensors**' command report '**ALARM (HIGH)**'.

The interrupt mode is available only for the nct7802y by the module parameter '**nct7802y_smbalert**'.

This parameter is set at 0 that means the interrupt mode is not active.

If the interrupt mode is activated, the thresholds must be modified. It is necessary to fit the mode pre-configured by the BIOS. This mode set the interrupt when the temperature is higher than the maximum threshold, and unset the interrupt when the temperature is lower than the minimum threshold. In this case both threshold must have positive values.

The lower temperature threshold is a negative value.
The higher temperature threshold is a positive value

9.1.3 Voltage Sensors Threshold and Alarm Report

Only the Nuvoton NCT7802Y manages lower and higher limits.

Only the polling mode is implemented in the Nuvoton NCT7802Y device driver.

'**ALARM (MIN)**' is reported by '**sensors**' command when sensor value is lower than the minimum threshold.

'**ALARM (MAX)**' is reported by '**sensors**' command when sensor value is higher than the maximum threshold.

9.2 CPLD-WDT

NAME

cpld-wdt – Kontron board cpld_wdt watchdog driver

DESCRIPTION

This man page describes how to use the watchdog implemented by the cpld on various Kontron boards including the VX305x families.

The principle of a watchdog is to automatically provoke some action after a given time passes without the watchdog being prodded by some process. This would indicate that the process is no longer working correctly. The cpld_wdt watchdog actions are to do nothing, to reset the board, to generate an interrupt that can wake up some other process, or to reboot. The prodding is done by writing to the watchdog device, which restarts the timeout.

The cpld_wdt module implements the standard Linux watchdog API, detailed in file **Documentation**/**watchdog**/**watchdog** – **api.txt** in the kernel source rpm. The supported features are described below.

Loading Module
There can be many devices and modules providing the watchdog API. Ensure only the cpld_wdt module is being used. For example, the standard iTCO_wdt watchdog module is usually blacklisted by a file in /**etc**/**modprobe.d**/ to avoid it being loaded.

The cpld_wdt module has to be explicitly loaded, for example by creating a systemd file with a name ending in ".conf" in /**etc**/**modules** – **load.d**/ listing the modules to load, one per line, for example:

```
cpld_wdt
```

Module Options
To provide initial options to the module, create a file with a name ending in ".conf" in /**etc**/**modprobe.d**/ holding a line such as:

```
options cpld_wdt timeout=50 trigger_mode=1 nowayout=1
```

The module options, shown by the command "modinfo cpld_wdt", include
timeout

the integer timeout in seconds, from 0 to 510 with a resolution of 2 seconds. The default is 30.

trigger_mode

the action to do when the timeout expires. An integer value

- 0 to simply countdown with no action,
- 1, the default, to reset the board,
- 2 to generate an interrupt that can be used to wake up a read on the device, or to reboot if no read is pending.

nowayout

an integer 1 if there is no way to stop the watchdog. The default is 0, which stops the timer if the magic character "V" is written just before the device is closed.

Usage from Scripts
The watchdog is implemented with the standard device /**dev**/**watchdog**. This file can accept ioctls to configure the watchdog, but can be used simply from a shell script as follows: Load the module with the required configuration, for example to interrupt, with a timeout of 10 seconds:

```
[root]# rmmod cpld_wdt
[root]# modprobe cpld_wdt timeout=10 trigger_mode=2 nowayout=0
```

Start the watchdog by writing to the device:

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

Prod the watchdog faster than every 10 seconds:

```
[root]# while sleep 5; do echo >/dev/watchdog; done
```

After a while stop the loop and wait for a timeout:

```
[root]# cat /dev/watchdog
```

This will hang for 10 seconds. If you wait another 10 seconds without issuing another read, the OS will reboot. To stop the watchdog instead:

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

To avoid the need to be root, simply change the ownership or permissions of /dev/watchdog.

Watchdog Ioctl API
The following standard ioctls are supported.
WDIOC_GETSUPPORT

```
#include <linux/watchdog.h>
struct watchdog_info ident;
ioctl(fd, WDIOC_GETSUPPORT, &ident);
```

returns in the structure the fields
identity

the driver identification ”cPLD WDT”

firmware_version

is always 1

options

describes the supported features, namely **WDIOF_KEEPALIVEPING** and **WDIOF_SETTIMEOUT**.

WDIOC_SETTIMEOUT

```
int timeout = ...;
ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
```

sets the timeout in seconds.

WDIOC_GETTIMEOUT

```
ioctl(fd, WDIOC_GETTIMEOUT, &timeout);
```

returns the current timeout setting in the argument (not the dynamically changing counter value).

WDIOC_SETOPTIONS

```
int options = ...;
ioctl(fd, WDIOC_SETOPTIONS, &options);
```

configures the given options. These are **WDIOS_DISABLECARD** to disable the watchdog, and **WDIOS_ENABLECARD** to enable the watchdog.
WDIOC_KEEPALIVE

```
ioctl(fd, WDIOC_KEEPALIVE, 0);
```

prods the watchdog, restarting the countdown timer.

SEE ALSO

cpld(4)
cpldtool(1)

FILES

/dev/watchdog
Documentation/watchdog/watchdog – api.txt

COPYRIGHT

Kontron

9.3 VPD Tool

NAME

vpdtool − display Kontron board VPDs (Vital Product Data)

SYNOPSIS

−a *boardtype*
− −boardtype
− −conffile or −f *file*
− −eclevel
− −help or −h
− −human or −H
− −keylist
− −macaddr
− −serialnumber
− −variant

DESCRIPTION

vpdtool reads the VPDs (Vital Product Data) of many Kontron boards, including the following:

- ITC320/322 PENTXM2/4
- VM6050 VM6052 VM6054
- VM6250
- VX3020 VX3030 VX3035 VX3040
- VX3230 VX3240
- VX6060 VX6070 VX6080

As it reads hardware ports via /dev/mem you need to be root to run it.

OPTIONS

−**a** *boardtype*

forces the architecture of the board to that given, e.g. VX3020

− −**conffile or** −**f** *file*

uses the given features definition file. This file says how to convert the vpd binary encodings into text descriptions.

− −**help or** −**h**

prints an option summary.

− −**human or** −**H**

displays features of the board in a human readable way.

− −**boardtype** − −**eclevel** − −**keylist** − −**macaddr** − −**serialnumber** − −**variant**

displays only the requested information. The options can be combined.

EXAMPLE

```
# vpdtool --human
VX3058 detected
Board type   : VX3058SA881
EC Level     : 14000A0
Serial Number: 1815281060016
Variant      : 00652000000205012000
```

```
MAC addresses  :
eth0: 0:0:de:52:cb:38
eth1: 0:0:de:52:cb:39
eth2: 0:0:de:52:cb:3a
eth3: 0:0:de:52:cb:3b

Features     :
PCB A
SA Class
Board pitch: 5HP / 1 inch
VX3058: 8-core CPU
Standard TDP Soc
Targetted TDP is 45W
SoC VCCIN tuning: 3-phase VCCIN
DDR PN is Micron MT40A512M8HX-093E:A (DDR4-2133, 4Gb, 0/95¿C, Mask A)
DDR: Dual bank memory configuration
DDR: One memory rank per bank (also called single die)
DDR: 4-Gbit devices
DDR: 4-Gbit devices
On-board PCIe switch PEX8725 equipped
Trusted Platform Module not equipped
ETH1: I210 controller equipped
ETH1: I210 EEPROM equipped
ETH1: On-board magnetic equipped
ETH1: 1000 BASE-T configuration
ETH2: I210 controller equipped
ETH2: I210 EEPROM equipped
ETH2: On-board magnetic equipped
ETH2: 1000 BASE-T configuration
P0 MUX1: P0.B7 & P0.A7 connected to CPLD GPIO3 & GPIO4
P0 MUX2: PCIE Common CLK connected from P0 to SOC
P1 MUX1: P1.E12 & P1.F12 connected to SoC USB2
SATA MUX: SoC SATA1 connected to P1 and SoC SATA4 connected to M2S1
DVI MUX: DVI connected to M2S1
PECI MUX: PCH PECI connected to CPU and IPMC => CPU IPMC
M2S1 (M.2 Mezzanine on Top Side): Connector equipped
M2S1 MUX: M2S1 connected to PCIe
M2S1 is MODP-GM2-SM750-xx
M2S2 (M.2 Mezzanine on Bottom Side): Connector equipped
M2S2 MUX: M2S2 connected to SATA
No predefined M2S2 mezzanine (a generic M2S2 may or may not be equipped).
FrontIO mezzanine connector equipped
FrontIO mezzanine: IDVI-VX305X-xxx
FrontIO mezzanine power supply: 5V
Front Connectors (Serial, USB, ETH) equipped
XMC connectors (J15, J16) not equipped
J16 Personality module: J16.F16 & J16.C16 not connected to V_5V0 and
XMC_PCISP not connected to J16
No predefined XMC mezzanine (a generic XMC may or may not be equipped).
No IPMI
SER2 MUX: Serial SER2 connected to SOC
Battery equipped in on-board battery holder
Connector for battery mezzanine not equipped
Supercap not equipped
Normal BOM (not extended temp. range)
Lead Free Process and Immersion Tin PCB Plating
```

FILES

/dev/mem
/dev/i2c/*

9.4 LEDs

The driver **leds_cpld** handles the front panel LEDs for user mode.

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

```
[root]# ls /sys/class/leds/
led3:amber  led3:OFF  led4:amber  led4:OFF  led5:amber  led5:OFF
led3:green  led3:red  led4:green  led4:red  led5:green  led5: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)
- slow blinking (echo 1, see following example)
- fast blinking (echo 2, see following example)
- OFF (echo 1, see following example)

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

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

- VX305x: On the front panel, the name of the led1 is L1, led2 is L2 and led3 is L3.

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

9.5 Multinodes Diskless

Multinodes Diskless is a powerful 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] 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;

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]# systemctl disable diskless
```

9.5.1 Introduction

When you have several VX305x boards in your machine, you can use one of the board as a diskless server.

Most of the time, it will be the first board present in a chassis but it is not mandatory. It must be equiped with a SATA hard disk (or SATA SSD).

Of course, you won't need any disk media present on the other targets providing that the targets will boot on the diskless server through the network (that is the point).

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 diskless product is incompatible with SELINUX enforcing policy as well as iptables services.

So run the following commands in order to disable these services:

- Stop the iptables service and disable SELINUX by running the commands:

```
[root]# systemctl disable firewalld
[root]# systemctl stop firewalld
```

- Set the variable SELINUX=disabled in the file: **/etc/selinux/config**

The following steps describe how to initialize a diskless configuration and how to boot it on a given number of targets.

9.5.2 Get the whole Fedora 21 Distribution on a Local Repository

The very first step is to get a local repository on the diskless server of the Fedora 21 RPM packages.

To do this, insert the Fedora 21 DVD and run these commands :

```
[root]# mkdir -p /dist/Fedora_21_kvx_bsp_boards/{BSP,RPMS}
[root]# mount /dev/dvd /mnt
[root]# cp /mnt/Packages/* /dist/Fedora_21_kvx_bsp_boards/RPMS
[root]# cd /dist/Fedora_21_kvx_bsp_boards
[root]# createrepo .
```

Then code the right value into the following field of the **/etc/diskless/diskless_fs.xml** :

```
<baseurl>file:///dist/Fedora_21_kvx_bsp_boards/</baseurl>
```

In this way the diskless tool will know the location of the Fedora 21 RPMs just installed.

The next step is to build a diskless root file system based on these Fedora 21 RPMs.

9.5.3 Build the diskless RFS (root file system)

Force the layer building even if it already exists by running:

```
[root]# diskless -B
```

The command will take a few minutes to complete.

The content of the RFS is defined in the section named **<rpms>** present in the xml file **/etc/diskless_fs.xml**

If you want change the content of the RFS, for example in order to add your own package, simply add the package name to the list.

Be careful when you change one of the xml files : **/etc/diskless/images.xml** or **/etc/diskless/diskless_fs.xml** . Keep a copy of the initial version.

You should notice the presence of two layers under **/diskless/layers**

```
[root]# ls /diskless/layers
fedora_base fedora_config
```

with

- **fedora_base**: Containing the reference root file system.
- **fedora_config**: A customizable layer you can change as you want.

For example, if you want to use your own **/etc/hosts** file, simply install it under **/diskless/layers/fedora_config** by running:

```
[root]# cp --parents /etc/hosts /diskless/layers/fedora_config
```

In this way, the target(s) after booting throught the network will "see" your own **/etc/hosts** file instead of the reference one (in fact, from the diskless client point of view, the layer **fedora_config** overloads the underneath **fedora_base** layer).

The RFS is now ready. The next step is to configure the DHCP server as well as the PXE boot.

9.5.4 Configure the DHCP and PXE Services in MAC Ethernet Address Mode

This step is mandatory to authorize remote targets to boot over the LAN on the diskless server.

Two modes are available for this purpose:

- MAC Ethernet mode: in this mode the IP address delivered by the DHCP server depends on the MAC Ethernet address of the target board.
- GEOID mode: in this mode the IP address delivered by the DHCP server depends on the location of the target board in the chassis.

This section presents the MAC Ethernet adress mode. The next one presents the GEOID Ethernet addres mode. Of course both of them are exclusive.

At first, uncomment if needed the XML structure called **<nodemap id="MAC_ADDR-diskless-cluster"..>** in the file **/etc/diskless/image.xml** and comment out the one related to the GEOID mode.

Then code the right MAC Ethernet address (related to the network interface used during the boot step) into this field:

```
<match criterium="macaddress">
<value>00:00:xx:xx:xx:xx</value>
</match>
```

From the server side, configure the network interface **em1** (for example) with the right address with the command:

```
[root]# ifconfig em1 192.168.1.1
```

If you want to use a different network base address, simply code this in the field **<ip>192.168.1.% (1+order.-value)</ip>** of the structure **<nodemap>**.

Start the required service **tftpd** by running the command:

```
[root]# chkconfig tftp on
```

Finally run the command:

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

Verify the right files have been created under **/diskless/tftp**:

```
[root]# cd /diskless/tftp
[root]# find .
```

The created files should be:

```
./pxelinux.0
./pxelinux.cfg
./pxelinux.cfg/default
```

Another impact of the **diskless -x** command is the starting of the DHCPD daemon.

So verify all worked well in this area by running:

```
[root]# service dhcp status
```

The output should include: **active** (running)

At this step, unless you want to configure diskless in GEOID mode, go directly to the section 9.5.6 - Export the Root File System.

9.5.5 Configure the DHCP and PXE Services in GEOID Ethernet Address Mode

An alternative to the MAC address mode is the GEOID mode. In this mode the IP address delivered by the DHCP server depends on the location of the target board in the chassis.

This mode is interesting in the sense that it allows you to replace a board by another one without changing the configuration file: **image.xml** (because the MAC Ethernet address, specific to each board, is not coded into the image.xml file in this case).

Indeed, in this case, it is required to code the GEOID range into the **image.xml** file instead of the explicit MAC addresses related to the boards plugged into the rack.

So edit the file **/etc/diskless/image.xml**, uncomment the right XML structure called **<nodemap id="GEOID-diskless-cluster"..>**... and comment out the one related to the MAC-ADDR mode.

Then, code the right GEOID value into this file (field **<match criterium="geoid">**).

In order to take into account the previous change, run the following command:

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

At this step, the DHCPD daemon should be ready as well as the TFTPD one.

9.5.6 Export the Root File System

The diskless RFS based on aufs (which implements the different stackable layers) must be exported now.

To do this, use this command:

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

Finally start the NFSD service with the command:

```
[root]# service nfs-server start
```

and verify the status is correct.

```
[root]# service nfs-server status | grep Active
```

At this step, the remote target should be ready to be booted in diskless mode.

Reboot the server to take into account the previous changes.

9.5.7 Boot the Target through the Network

On the diskless target and after a power-on, enter BIOS setup on 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 the following 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.

Log in as root and verify the system is operational.

9.5.8 Diskless Service

To automate the launching of the diskless at the boot step, simply enable the related service by running :

```
chkconfig diskless on
```

In this case, the next time you boot the server, the service **diskless** will be launched.

As result the commands **diskless -x** and **diskless -e** will be run automatically, authorizing the diskless boot of the target.

9.5.9 Initiate a Reset of the Remote Targets

```
[root]# diskless -aR
```

Should cause a reset of all the remote targets.

9.6 Sysvartool

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

```
# sysvartool -A pbit -l
VX3058 detected
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
           core: PASSED
            rtc: PASSED
    ether_loop0: PASSED
    ether_loop1: PASSED
    ether_loop2: PASSED
    ether_loop3: PASSED
          hwmon: PASSED
sata0_controler: PASSED
sata1_controler: PASSED
            vpd: PASSED
ehci_controller: PASSED
xhci_controller: PASSED

PASSED  : 18
FAILED  : 0
NOT RUN : 0
TOTAL   : 18
```

9.7 GPIOs

NAME

cpld-gpio - Kontron board gpio-cpld driver gpios

DESCRIPTION

This man page describes how to use the gpios implemented by the cpld on various Kontron boards including the VX305x families.

The cpld gpio driver provides an API using files in the /sys filesystem. Reads and writes to the files are directly handled by the driver, provoking reads and writes of hardware registers. Reads normally return a short string terminated with a newline. Further reads need to seek back to the start of the file or re-open it. Writes should normally be of a single string, optionally terminated with a newline. Writing values other than those described below are undefined, but usually result in a write error with errno EINVAL Invalid argument.

Usage from Program
Operations can be easily done from a shell script. When used from a program, open the files for simultaneous read and write, except for "intr_stat" (read only). Read in one go into a buffer big enough to accept the whole reply, including newline (i.e 11 characters). Use lseek(2) to rewind the file descriptor before each read. Write each value using a single system call. There is no need to lseek before writes.

Permissions
By default the files are owned by root and only writeable by the owner. If necessary, these ownerships and permissions can be changed (each time the driver is loaded), for example by a udev rule.

Sysfs Files
There is one directory per gpio, starting with gpio1, under the class directory:

/sys/class/cpld-gpio/

The number of gpios depends on the board. Each directory has the following files. The string values described are those that can be written, or will be read.

direction holds the value "in" (incoming signal) or "out" (outgoing). The default is "in". For example:

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

value holds the value "0" (low signal) or "1" (high). On read return the current value of the gpio. On write sets the value of an outgoing gpio. For example:

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

polarity holds the value "hi" (interrupt when signal active on high level or rising edge), or "lo" (active on low level or falling edge). The default is "hi".

mode holds the value "edge" (interrupt on rising or falling edge) or "level" (on high or low level). The default is "edge".

toggle holds the value "on" (interrupt on any state change) or "off" (interrupt only on the configured polarity and mode). When on it overrides the polarity and mode settings. The default is "off".

interrupt holds the value "on" (enable interrupts) or "off" (disable). When an interrupt occurs in level mode, this value is automatically reset to "off" to avoid a permanent interrupt. The default is "off".

intr_stat is a read-only count of the number of interrupts received on the gpio. This is an unsigned 32 bit decimal value which can be up to 10 digits long, plus a newline. This file is read-only, and writing to it will result in a write error of EPERM Permission denied, or EIO Input/output error. For example:

```
$ cat /sys/class/cpld-gpio/gpio5/intr_stat
12
```

value_it on read waits for an interrupt on the gpio, and then returns a string of length zero. If interrupts have not been enabled, the file acts like the value file. The file acts like the value file for all writes.

To get an interrupt you must set the direction, polarity, mode, toggle and interrupt files appropriately, and then read value_it until it returns, or poll the intr_stat value until it changes.

Dual Purpose GPIO
On some boards, gpio2 is a dual purpose gpio which can also be used to generate a Maskable Reset. It is configured using cpld registers that can be set using cpldtool, or low level port accesses.

To check if the board provides this feature run the command:

```
[root]# cpldtool -a|grep MSKR2LOC
```

If MSKR2LOC is not found, gpio2 is not dual purpose. To configure gpio2 as a standard gpio which does not issue a local reset, on a VPX board:

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

on a VME board:

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

or on either type of board specify the register by number instead:

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

Use the same command with value 1 to restore the configuration. To do the same using low level port accesses, read the register and calculate the new value before writing it back. For example:

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

SEE ALSO

cpld(4)
cpldtool(1)

FILES

/sys/devices/platform/cpld_gpios/cpld-gpio/gpio[1-8]

COPYRIGHT

Kontron

9.8 cpldtool

```
# cpldtool -a
VX3058 detected
Reg 0x0 - CPLD_ID = 0x02
              CPLD_ID=0x0
              CPLD_Debug=0x0
              CPLD_Version=0x2
Reg 0x1 - PCB_ID = 0x46
Reg 0x2 - FIRM_POST = 0x00
              PBIT_FAIL=0x0
              PBIT_RUN=0x0
Reg 0x3 - PWON_STATUS = 0x00
              DPOST=0x0
              BOARD_CLASS=0x0
              POST_RTC=0x0
Reg 0x4 - PWR_RST_CONFIG = 0x02
              PWRON_MODE=0x0
              PEXRST_Dis=0x0
              Alert_inhib=0x0
              PEXRST_Ctl=0x1
Reg 0x7 - SERIAL_LINES_CTL = 0x04
              SERIAL_MODE=0x0
              SERIAL_TXEN=0x1
              SERIAL_TERM=0x0
Reg 0x8 - PCI_MODE = 0x80
              XMC_PRESENT=0x1
              XMC_SLOT_PRESENT=0x0
Reg 0x9 - MEM_PROTECT = 0x02
              Boot_flash_CS_swap_DIP=0x0
              Boot_flash_CS_swap_Valid#=0x0
              Boot_both_flash=0x0
              USER_WP=0x0
              SYS_WP=0x0
              VPD_WP=0x1
              NVMRO=0x0
Reg 0xb - M2SLOTS_CONFIGURATION = 0x01
              M2_SLOT2_SLP=0x0
              M2_SLOT2_CFG=0x0
              M2_SLOT1_SLP=0x0
              M2_SLOT1_CFG=0x1
Reg 0xc - BOARD_CONFIGURATION = 0x80
              ETH2_LANSW_SEL=0x1
              ETH1_LANSW_SEL=0x0
              BIOS_FAILSAFE=0x0
              DVI_FRONT_SEL=0x0
              TPM_PRESENT=0x0
              IPMI_CONFIG=0x0
Reg 0xd - DRAM_CONFIG = 0xD4
              DRAM_CHANNEL=0x1
              DRAM_RANK=0x1
              DRAM_DENSITY=0x1
              DRAM_SPEED=0x0
```

```
Reg 0xe - DIP_SWITCH_STATE = 0x00
              DBG_MD=0x0
              PCIESW_FAILSAFE=0x0
              PROCHOT=0x0
              FACTORY_MODE=0x0
Reg 0xf - SERIRQ_CONTROL = 0xE0
              SMB_MSK=0x1
              PECI_ALERT_MSKK=0x1
              TIP_MSK=0x1
              GPIOs_INT=0x0
              SMB_INT=0x0
              ALERT_INT=0x0
              TIP_INT=0x0
              WDG_INT=0x0
Reg 0x5b - ALERT_STATUS = 0x00
              CATERR=0x0
              THERMTRIP=0x0
              VRHOT=0x0
              PROCHOT=0x0
              PECI_ALERT=0x0
              FIVR_FAULT=0x0
              PCIE_ERR=0x0
              PCH_TEMP_ALERT=0x0
Reg 0x6a - GEO_ADD = 0x3E
              SYSCON=0x0
              Error=0x0
              GAP=0x1
              GA=0x1E
Reg 0x6b - GA_GPIO = 0x3E
              CUR_SYSCON=0x0
              Cur_Error=0x0
              GAPGPIO=0x1
              GA_GPIO=0x1E
Reg 0x70 - VPX_CONTROL = 0xD4
              GDiscrete1_Ctl=0x1
              GDISCRETE_Status=0x1
              MSKR2LOC=0x1
              VPX_RST=0x1
              VPX2LOC=0x0
              LOC2VPX=0x0
Reg 0x71 - PCIe_SWITCH_VPX = 0x10
              VPX_REFCLK=0x0
              VPX_AUXCLK=0x0
              GDISCRETE_DEFAULT=0x1
              PEX_SW_MODE=0x0
Reg 0x72 - I2C_BOARD_STATUS = 0x95
              Power_Status=0x1
              Reset_Source=0x0
              Reset_Status=0x1
              Boot_Status=0x5
Reg 0x73 - I2C_BOARD_CONTROL = 0x91
              Board_Id=0x9
              Reset=0x0
              Power_OnOff=0x1
```

```
Reg 0x74 - I2C_ERROR_STATUS = 0x00
              Alert=0x0
              POST_Error=0x0
              POST_RTC=0x0
Reg 0x75 - I2C_PORT80 = 0x00
Reg 0x76 - I2C_FAILCODE = 0x00
Reg 0x78 - I2C_MISC = 0x00
              Force_rescue_BIOS=0x0
              Force_EFI_Shell=0x0
              Power_CUR=0x0
              Power_REQ=0x0
```

9.9 FMRAM Example

The **fmram** packages gives an example of how access to the FMRAM device which may be used to save some customer data which needs to be backuped.

```
[root]# 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.

9.10 CPLD

NAME

cpld – Kontron board facilities, cpld_i2c, cpld_smi, cpld_leds, cpld-gpio, cpld-wdt

DESCRIPTION

This man page describes some of the facilities made available by the cPLD (complex programmable logic device) on several Kontron boards. Not all boards provide all the facilities.

The cPLD is a small device that controls some low – level aspects of the board, including power up sequencing, reset, gpios, i2c buses including one for communication to the backplane, timer, watchdog, leds, and many configuration and control features.

The device is visible through 1 – byte registers in i/o port space from addresses 0x0800 to 0x08FF. The facilities are implemented in several drivers, each in its own rpm. The driver sources are provided.

cpld
The cpld driver is needed by the other drivers. It provides the geographical id (slot number) of the board in the file /**proc**/**geo_id**.

Some other basic features can be accessed with the cpldtool(1) utility, or by direct i/o on the appropriate port. For example, use dd on /**dev**/**port** or the **port** command (from rpm **hwtools**) as user root (beware: inappropriate i/o on ports may crash the system). As an example port 0x800 holds the version number of the cpld:

```
[root]# port 0x800
@0x800 = 0x06
```

Reset cpuB

On boards with two separate cpus, you can reset cpuB from cpuA or vice versa by setting lsb bit 0 to 0 in port 0x804. This is most easily done with the cpldtool utility:

```
[root]# cpldtool -f PWR_RST_CONFIG Software_Cross_Reset 0
```

The alternative is to read the port to find the current value, calculate the new value obtained by clearing bit 0 (which should always read as 1), and write the new value back. The bit will return to 1 on its own.

Access cpuB Serial Port

On boards with two separate cpus, you can enable access to cpuB's second console serial port from cpuA by dynamically setting the appropriate bit in cpld register 0xc, or permanently by configuring the bios. The two /dev/ttyS1 devices on the board are then linked together and can communicate via some utility like minicom or pyserial's miniterm.py:

```
[root]# cpldtool -f BOARD_CONFIGURATION SERIAL2_cfg 1
[root]# miniterm.py /dev/ttyS1 115200
```

To configure this permanently with the bios on cpuA follow the menus:

```
Kontron
Serial Configuration
COM1 Link Mode: [Enabled]
```

To see the cpuB bios on ttyS1, configure the cpuB bios with:

```
Advanced
Serial Port Console Redirection
COM1
Console Redirection [Enabled]
```

To have Linux use the ttyS1 console ensure it has the boot option:

```
        console=ttyS1,115200
```

cpld_i2c
The cpld_i2c driver implements an i2c algorithm made available through the standard OS i2c interface.

cpld_smi
The cpld_smi driver provides a file and utility to program the onboard ethernet switch over the smi (Serial Management Interface) bus. The device /**dev**/**cpld_smi** has 2 ioctls to read and write a given register in the switch, snd the **swsmi** utility uses this device.

cpld_leds
The cpld_leds driver provides standard OS led class devices in /**sys**/**class**/**leds**/. They have filenames of the form *devicename:colour*.

Each cpu has 4 user settable leds, with devices named **led2**,**led3**,**led4** and **led5**.

They implement 3 brightness levels by writing a value to *devicename:colour*/brightness

- 0 on
- 1 slow blink
- 2 fast blink

This switches each LED to user mode. Write 0 to *devicename*:OFF/brightness to switch an LED off, or any other value to exit from user mode. See the discussion in file **.../Documentation/leds – class.txt** (from rpm **kernel – doc**).

cpld – gpio
See the separate man page for the driver providing gpios implemented by the cpld.

cpld – wdt
See the separate man page for the driver providing a watchdog implemented by the cpld.

SEE ALSO
cpld – gpio(4)
cpld – wdt(4)
cpldtool(1)

FILES
/proc/geo_id
/dev/cpld_smi
/dev/port
/dev/i2c – 0 ...

COPYRIGHT

Kontron

10 / RC Boards

10.1 How to Manage the Lack of RTC Battery

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 to the current date. If not, the boot is stopped in a maintenance mode. This could happen after with the power off and without the RTC's battery. To workaround this, simply create a file named **/etc/e2fsck.conf** with the following content:

```
[options]
broken_system_clock=true
```

The same file has to be added to the **initramfs** too. To do this, simply run the command:

```
[root]# dracut --force -I /etc/e2fsck.conf
```

Without the right date for the system, some other services and functions may present problems. 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.

10.2 External Devices Connection

One impact of the RC board is the missing of front panel.

As a consequence, devices requiring access to the system such as:

- Serial line console.
- USB mouse
- USB keyboard
- Display monitor (through the HDMI port)
- Ethernet

will be plugged in from the rear side through a Rear Transition Module (RTM) Paddle board.

The Kontron PB-VX3-4xx is a 3U VPX Rear Transition Module compliant with the definition of the Rear Transition Module on VPX Standard –VITA 46.10.

It provides rear I/O peripherals connectivity for Kontron VX305x Single Board Computers.

For example, in order to perform a graphical linux installation stage on a VX305x-RC, plug in the required devices to the paddle board:

- USB mouse and USB keyboard through a USB HUB.
- The display monitor (through the HDMI port).

After turning on the system, you should notice all devices have been recognized correctly allowing to take control of the graphical environment.

10.3 RC Specifications

The RC version of the boards is designed to work in different environmental constraints and for different levels of temperature and power dissipation.

This may imply specific setups (cpu frequencies, hardware parameters,...) to guarantee the specified behavior.

They are described in the “Hardware User's Guide” of each board. Please check this document for your board and for the required environment.

About Kontron

Kontron, a global leader in embedded computing technology and trusted advisor in IoT, works closely with its customers, allowing them to focus on their core competencies by offering a complete and integrated portfolio of hardware, software and services designed to help them make the most of their applications.

With a significant percentage of employees in research and development, Kontron creates many of the standards that drive the world's embedded computing platforms; bringing to life numerous technologies and applications that touch millions of lives. The result is an accelerated time-to-market, reduced total-cost-of-ownership, product longevity and the best possible overall application with leading-edge, highest reliability embedded technology

Kontron is a listed company. Its shares are traded in the Prime Standard segment of the Frankfurt Stock Exchange and on other exchanges under the symbol "KBC".
For more information, please visit: **www.kontron.com**

KONTRON FRANCE

150, rue Marcelin Berthelot
ZI de Toulon-Est . BP 244
83078 Toulon Cedex 9 - France
Tel: +33 4 98 16 34 00
Fax: +33 4 98 16 34 01

CORPORATE OFFICES

HEAD OFFICE

Lise-Meitner-Str. 3-5
86156 Augsburg
Germany
Tel.: + 49 821 4086-0
Fax: + 49 821 4086-111
info@kontron.com

NORTH AMERICA

14118 Stowe Drive
Poway, CA 92064-7147
USA
Tel.: + 1 888 294 4558
Fax: + 1 858 677 0898
info@us.kontron.com

ASIA PACIFIC

1~2F, 10 Building, No. 8 Liangshuihe 2nd Street,
Economical & Technological Development Zone,
Beijing, 100176, P.R. China
Tel.: + 86 10 63751188
Fax: + 86 10 83682438
info@kontron.cn