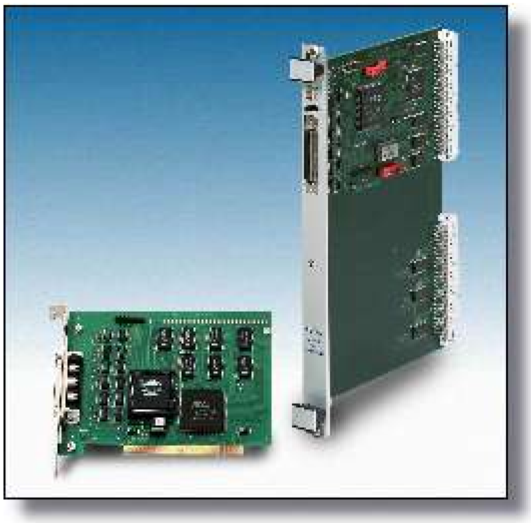
PCIVME - Linux Driver Quick Installation and Usage Guide

ARW Elektronik, Germany
and
Klaus Hitschler (klaus.hitschler@gmx.de)



document history

1st version of this document	14.11.2002
adapted to kernel 2.6	13.08.2004

Preface

This manual and source code are copyright of ARW Elektronik, Germany and is published under GPL and LGPL (Open Source). You can use, redistribute and modify it unless this header is not modified or deleted. No warranty is given that this software will work like expected.

This product is not authorized for use as critical component in life support systems without the express written approval of ARW Elektronik Germany.

Please announce changes and hints to ARW Elektronik

Installation contents

```
pcivme
|-- Makefile                ; a 'global' Makefile to make driver, lib and test progs
|-- doc                     ; some documentation and templates
|   |-- COPYING
|   |-- template.c
|   |-- template.h
|   |-- template.make
|   `-- todo.txt
|-- driver                  ; driver sources and installation scripts
|   |-- Makefile
|   |-- askpci.c
|   |-- askpci.h
|   |-- common.h
|   |-- fops.c
|   |-- fops.h
|   |-- main.c
|   |-- main.h
|   |-- pciif.h
|   |-- pcivme.h            ; driver interface header
|   |-- pcivme.o
|   |-- pcivme_load         ; installation script
|   |-- plx9050.h
|   |-- plxbug.c
|   |-- plxbug.h
|   |-- vic.h
|   `-- vme.h
|-- lib                     ; shared library sources
|   |-- Makefile
|   |-- pcivme_ni.c
|   `-- pcivme_ni.h         ; library interface header
|-- pvmon                   ; a small test utility
|   |-- Makefile
|   |-- mbuffer.c
|   |-- mbuffer.h
|   |-- pcilibLx.c
|   |-- pcilibLx.h
|   |-- pvmon
|   |-- pvmon.c
|   `-- pvmon.cfg
`-- test                                ; another small test utility
    |-- Makefile
    |-- simpleTest
    `-- simpleTest.c
```

Compatibility

Driver and library are compiled and tested on a machine with kernel 2.4.18 (SuSE 8.0) and kernel 2.2.19 (SuSE 7.3) , RedHat 7.2 and kernel 2.6.5 (SuSE 9.1). The source code is compatible to 2.2, 2.4 and 2.6 kernels.

I think it will compile and run on future versions, but not on versions before 2.2.x.
The sources are independent of special x86 hardware features and should compile on other platforms. But this is not tested.

Some basic cross compilation support is provide if you invoke at the driver directory

```
        make KERNEL_LOCATION=your-kernel-location
```

Features

The driver and the shared library provide functions to access the PCIVME VME interface, use the advanced features of the interface and makes it possible to catch VME interrupts.

Driver and the library are capable of multi-user and multi-threading access.

Please note that some high speed access features like „autoread“ are heavily hardware supported and should not interrupted by concurrent accesses from different paths. The same is true for interrupt handling. The necessary arbitration is not done by the driver or the library.

Installation of the driver

For installation of the driver module must be "root". There is a installation (bash) script called "pcivme_load". Please invoke it with the module number of your VMEMM modules as command line parameter.
For example, if your interface is configured as module #1 (Jumpers J301.. J304) then call

```
   ./pcivme_load 1
```

This installs the driver and creates a device node "/dev/vmemm_1".
The module number "1" should be the factory set module number.

If you want to remove the driver do it with "rmmod pcivme". (sometimes /sbin/rmmod ...)

Install the shared library

The shared library "libpcivme.so" provides low level functions to access the VME-Bus interface. You will find the prototypes of this functions in the file "libpcivme.h".

Copy the library "libpcivme.so.x.x.x" (now 1.0.0) to your /usr/lib directory. Then cd to /usr/lib and make 2 (soft) links:

```
   ln -sf libpcivme.so.x.x.x libpcivme.so.1
   ln -sf libpcivme.so.1 libpcivme.so
```

Instead of typing lots of commands you can use the build in

```
   cd lib
   make install
```

Please note: you must have root access rights.

Verify the installation (1)

I provided a little test program which really does nothing useful. It is named "simpletest". Please invoke with “-?” as command line parameter to get help. To kill the program please type Ctrl-C.

Verify the installation (2)

The "cat /proc/pcivme" output is more detailed. Please take a look:

```
klaus@sylvia:~> cat /proc/pcivme

PCIVME information. Version 1.1 of Oct 20 2002 from K.Hitschler.
 ---------------------
 Interfaces found           : 1
 Major Number               : 254
 --- 1 ---------------
 LCR     phys/virt/size     : 0xd7003000/0xe1010000/128
 Control phys/virt/size     : 0xd7000000/0xe1012000/4096
 VME     phys/virt/size     : 0xd7001000/0xe1014000/4096
 Irq                        : 9
 VMEMM is or was            : not (software) connected.
 IrqCount                     : 0
 Pending IrqStatus            : none
```

The output will list all found interfaces and their parameters. The output of your computer will look different.

Making driver, library or test program

Change your directory into “pcivme”, then simply type “make”. To make each part simply type in “cd” into the appropriate directory and invoke “make”. To remove object code please call “make clean”, to install driver and lib (as root only) call “make install”.

Up from kernels 2.6 you need to have configured kernel sources installed to compile the driver. During the compilation process make uses the kernel build system (kbuild).

Make supports the targets all, clean, depend and fresh.

Dynamic major number allocation

The driver uses the dynamic major number allocation. You can switch to static allocation through changing in “main.c”

```
   #define MAJOR_NO 0            /* use dynamic assignment */
```

the MAJOR_NO to an appropriate number not equal 0.

Modversions

This is only valid for kernels lower than 2.6: If you want to have version control check against the kernel symbols you have to configure the switch "CONFIG_MODVERSIONS" before making your kernel. All provisions for version check are included in the driver sources. Normally the RedHat distribution support configured MODVERSIONS as default.

Debug information

To get more debug information from the driver please compile the driver with the switch DBG = __DEBUG__ (double underline), e.g. make DBG = __DEBUG__
Then additional debug information is printed into the file /var/log/messages. You can watch it with „tail -f /var/log/messages“. You must be root to do this.

Include header files

If you want to use the shared library please include the file “libpcivme.h”. To access the driver ioctl() functionality without shared library you must use the include file “pcivme.h”.

Interrupt handling

VME Interrupts can be received either in a polling or in blocking mode. In blocking mode you have to open a path to the device and do a blocking "ioctl(PCIVME_READ_VECTOR_BLOCK, ...) call. The call will return when a interrupt was raised and interrupts were enabled.
Each raised interrupt disables further interruptions. Normally, when your program returns from the blocking IO-call you will handle the cause of the LAM and then re-enable the interrupts.

At the time of writing the library supports only nonblocking reception of interrupts.

Path to modutils

Some Linux distributions provide the utilities “rmmod” and “insmod” in the standard paths. Sometimes they must be called with full path description, e.g. /sbin/rmmod   pcivme.

Feedback

Please mail your hints, questions and remarks to klaus.hitschler@gmx.de. All feedbacks are welcome.

Grants

This document was written with Star-Office 5.2 and OpenOffice 1.1.1.