Technical Information Manual

Revision n. 1

30 January 2009

WAVE DUMP
USER'S MANUAL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

The WaveDump[1] is a software console application that can be used to configure and readout event data from any model of the CAEN's digitizer family (V1724, V1721,V1720, V1731 and V1740 produced so far[2]) and save the data into a memory buffer allocated for this purpose. Optionally, the data can be written into a file. It is also possible to enable some data integrity check during the readout.

During the readout, a plot of the wave files can be commanded through the WaveDump console: WaveDump uses the GNUplot application (www.gnuplot.info) as a plotting slave application.

The configuration of the digitizer (registers setting) is done by means of a configuration file that contains a list of generic write accesses to the registers of the board. The configuration file contains also some other parameters (declared with specific keywords) that define the operating mode of the WaveDump program.
The program uses the CAENVMELib library for the VME access functions and it can work, without any modification, with CAEN V1718 and V2718 VME bridges. If a different VME controller is used, the Wave Dump application must be ported to the new architecture. Please refer to § 3.

A set of runtime commands can be sent to the application:

'q' => quit the program.
't' => send a software trigger (one shot).
'T' => enable/disable the continuous generation of the software triggers.
'w' => save one event into the files.
'W' => Toggles between continous save of event data to file. If this option has not been enabled by configuration file, command will enable continous event data storage to disk.
'p' => plot the waveform in the wave file (a path to the plotting program must have been provided in the configuration file). Default is wavedump running directory.
'P' => Toggles continous plotting. If enabled, plot is updated after every event readout loop.

Figure 1.1 illustrates the WaveDump application scenarios. The WaveDump application interacts directly with a set of functions provided by a software API library (the application has been written to directly call the CAENVMELib library functions, to be used in conjunction with CAEN VME bridge boards). The driver for the communication with the target VME master must also be properly installed prior to starting WaveDump. The CAEN Digitizer board can be accessed by means of:

- Direct CONET link (CAEN proprietary optical link) from PC to the Digitizer Front Panel connector:
- a CAEN A2818 PCI board must be installed in the PC.
- CAEN CONET/VME bridge (V2718): a CAEN A2818 PCI board must be installed in the PC to connect to the V2718 Front Panel CONET connector.
- CAEN USB/VME bridge (V1718)

---

[1] This manual refers to **WaveDump Release 1.9**

[2] CAEN's digitizer family User's manuals are available at

http://www.caen.it/nuclear/technical_documents.php

- Other VME masters (CPU/Bridge): in this case the WaveDump application must be rebuilt to target the different library/operating system/communication channel.

A configuration file to set some WaveDump internal behaviour (WaveDumpConfig.txt) is used.
An optional GNUplot command file (init.plt) could be also used to set some initial value for GNUplot. If command file is present in the wavedump run directory, the command file is loaded by Gnuplot when launched for the first time. Please refer to the GNUplot documentation for supported commands:
http://www.gnuplot.info/documentation.html

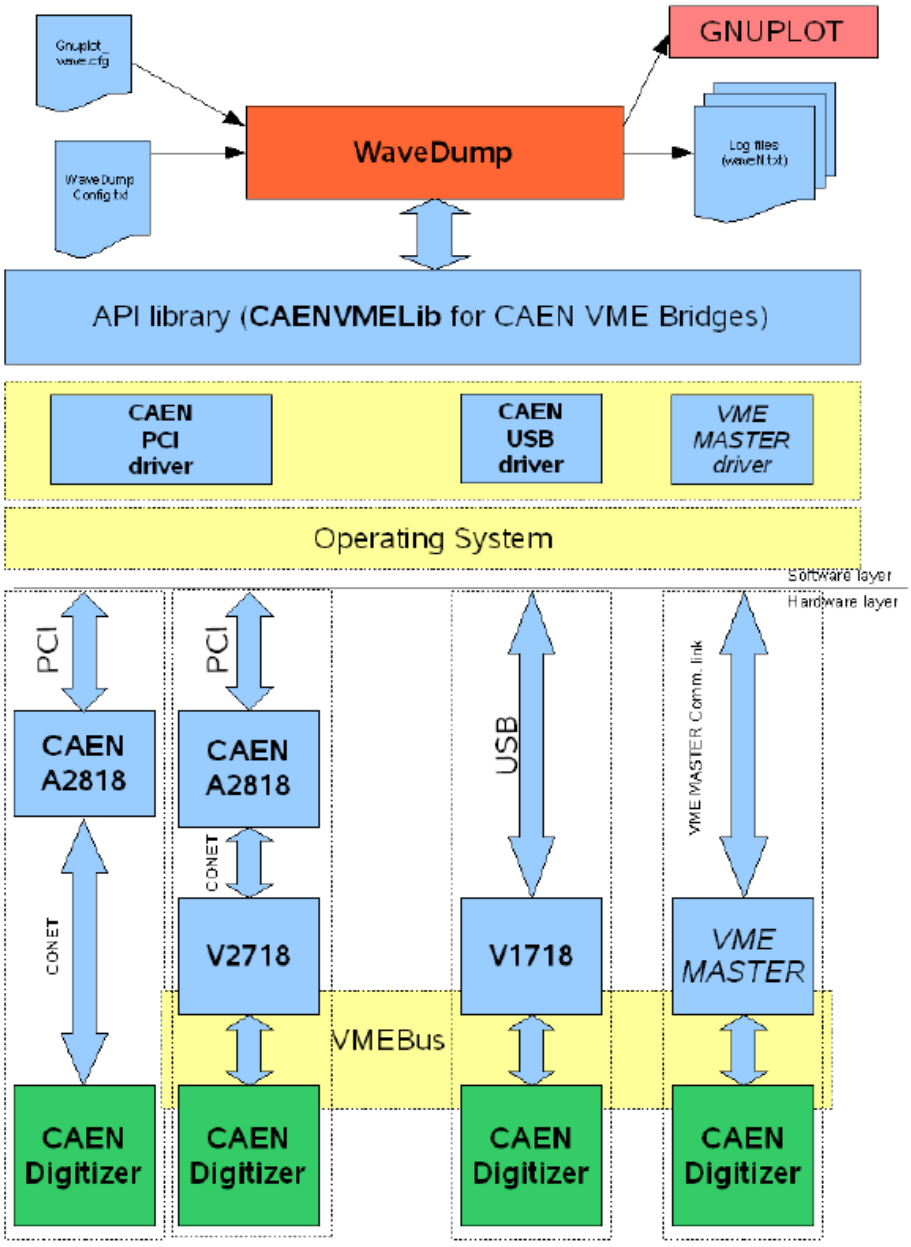

Fig. 1.1: WaveDump and software/hardware environment

2. WaveDump functional description

The ***WaveDump*** program is a readout example for the CAEN VME Digitizers family (V1724//V1720/V1721/V1731). It has been developed to be directly used by users of CAEN VME bridges (V1718/V2718). If a different VME master is used, the application can be ported as described in § 3.

This program reads the raw data from the CAEN Digitizers, performs the event building. Optionally, formatted data can be written to a log file. The program doesn't use any graphical library and can be easily compiled under different platforms. However, it can make graphical plots using the external program *GNUplot* as a graphical engine. GNUplot is freely downloadable from the WEB for both Windows and Linux (*http://www.gnuplot.info*). Plot commands are sent from WaveDump to GNUplot using a pipe; plot data (points) are passed through a file; plot settings like scale, colors, labels, etc. are set by the following Gnuplot script file

init.plt : Initial configiration script for Gnuplot.

plot.plt : this script is run by Gnuplot on every plot command issued by wavedump.

These two scripts must be present in the wavedump running directory. Example scripts are already provided in the bin directory.

See the GNUplot help for more details about plot settings and on line commands like zoom, autoscale, etc.

The program flow of the WaveDump can be divided into three phases:

Configuration: the program reads and executes the commands listed in a configuration file (*WaveDumpConfig.txt* by default) that contains the parameters for the operating mode of the program, as well as the board initial register settings. The commands are specified by a set of predefined keywords; in particular, the register write access to the Digitizers is done giving address and data, without any interpretation of the register function.

Initialization: the program reads some specific settings from the Digitizers, displays some information (FW revision, sampling clock frequency, etc.), allocate the buffers and initializes pointers and variables for the readout, according to the current settings of the Digitizers. It is worth noticing that *Configuration* and *Initialization* are independent; the point of contact between the two is the content of the Digitizers registers.

Data readout and analysis: the program starts the acquisition and the readout of the board output buffer, either in polling mode or with the interrupts. During the run, it is possible to send SW triggers, display the data throughput, plot the waveforms calling *GNUplot*.
The readout involves all the channels in all the board channels that have been open.

Syntax:

```
WaveDump [ConfigFile]

The default config file is "WaveDumpConfig.txt"
```

Configuration File Keywords:

LINK Bdnum Linknum

Description: open the communication with the VME master board.

Parameters: **Bdnum** = Board number in the CONET chain (0 is the first in the chain and when using CAEN V1718 USB/VME Bridge).

Linknum = A2818 number in the PC (V2718) or the USB driver number (V1718).

Notes: Please refer to the CAEN VME Bridges (V1718/V2718) board documentation for details.

READOUT_MODE n

Description: define the readout mode on the VME bus

Parameters: ***n=0*** *=> Single D32 read*

n=1 *=> 32 bit BLT*

n=2 *=> 64 bit MBLT*

n=3 *=> 2eVME (not yet implemented)*

n=4 *=> 2eSST (not yet implemented)*

Notes: Please refer to the Digitizer User's Manual and firmware release notes for 2eVME/2eSST support.

DATA_CHECK X

Description: enable event check. If X=0, no check is done.

Parameters: **X** is an hexadecimal number; each bit of **X** enables a specific analysis:

Bit 0 = Calculate and Print Throughput Rate

Bit 1 = Check Header Consistency

Bit 2 = Check Event Size

Bit 3 = Check Consecutive Event Counters

BASE_ADDRESS Address

Description: Set the board base address.

Parameters: **Addr** is the board base address in 32 bit hex number (I.e. 32100000)

GNUPLOT_PATH s

Description: define the path to the executable file of GNUplot

Parameters: **s** is the string of the path

WRITE_TO_FILE n

Description: define whether to save data into a file or not

Parameters: **n=0** => don't save data

n=1 => save data (only when the appropriate key is pressed (see § 1)

n=2 => enable continuous data save mode: when a new event (or group of events) has been completely read it is automatically saved to file.

APPEND_MODE n

Description: Set log file opening mode

Parameters: **n=0** => overwrite mode: event log file is overwritten.

n=1 => append mode: data from new events are appended at the end of the event log file.

OUTPUT_FORMAT n

Description: define the log file format

Parameters: **n=0** => write only data (list of decimal ascii numbers)

n=1 => write size + data

n=2 => write size + header + data

BLT_SIZE Size

Description: Set the VME Block Transfer Size.

Parameters: **Size** is a decimal number indicating the block transfer size in bytes.

WRITE_REGISTER Addr Data

Description: write a register of the target board.

Parameters: **Addr** is the register address (32 bit hex number).

Data is the data to write (32 bit hex number)

3. Application porting

The application has been tested on Windows 2000/XP and Linux Kernel 2.4 and 2.6.

3.1. Windows

A Microsoft Visual C++ 2005 solution file is provided (in the directory \build\).

3.2. Linux

A Makefile is available for recompilation under Linux operating system (in the directory \build\).

3.3. Retargeting the application to a different VME Master

If a CAEN VME Bridge is used as VME master, the WaveDump tool only needs a correct installation of
the CAENVMELib in the host PC. The application can then be directly used, without any code modification.

If a different Bridge/CPU is used the application must be ported to that architecture.

The macros declared in the **vme_wrapper.h** file must be implemented for the target Bridge/CPU.

In the **vme_wrapper.h** file, the VME-specific functions are declared as macros.

MACRO NAME	CAENVMElib function	Custom function	Notes
VME_INIT()	CAENVME_Init()	*WrapperVME_Init()*	It returns an handle for access to the VME master board.
VME_END()	CAENVME_End()	*WrapperVME_End()*	It closes connection to the VME master board.
VME_READ32()	CAENVME_ReadCycle()	*WrapperVME_ReadCycle()*	Single D32 read access to VMEbus
VME_WRITE32()	CAENVME_WriteCycle()	*WrapperVME_WriteCycle()*	Single D32 write access to VMEbus
VME_FIFOBLTREAD()	CAENVME_FIFOBLTReadCycle()	*WrapperVME_FIFOBLTReadCycle()*	32-bit Block Transfer Read in FIFO mode (see V1718/V2718 documentation). If Customer VME master does't support FIFO mode, each block transfer size must be limited to 4 KB.
VME_FIFOMBLTREAD()	CAENVME_FIFOMBLTReadCycle()	*WrapperVME_FIFOMBLTReadCycle()*	64-bit Block Transfer Read in FIFO mode (see V1718/V2718 documentation). If Customer VME master does't support FIFO mode, each block transfer size must be limited to 4kB
VME_IRQENABLE()	CAENVME_IRQEnable()	*WrapperVME_IRQEnable()*	It enables vme interrupts in VME master.
VME_IRQWAIT()	CAENVME_IRQWait()	*WrapperVME_IRQWait()*	It waits for interrupt up to the timeout
VME_IRQCHECK()	CAENVME_IRQCheck()	*WrapperVME_IRQCheck()*	It checks the VME master if a certain interrupt has been received
VME_IRQDISABLE()	CAENVME_IRQDisable()	*WrapperVME_IRQDisable()*	It disables selected VME interrupts on VME master

Table 3.1: VME access functions macro definition

Each wrapper function have to implement the CAENVMElib corresponding functionality using the target VME master specific API functions, provided by its Software Development Kit.

In order to use the wrapper functions, the __USE_CAEN_VME_BRIDGES__ must be undefined.

For Windows build, just uncomment the //#undef __USE_CAEN_VME_BRIDGES__ line in the vme_wrapper.h file.

For Linux build, the -D__USE_CAEN_VME_BRIDGES__ option in the COPTS macro of the Makefile must be removed.