

PRELIMINARY

Technical Information Manual

Revision n.0

7 November 2006

CAENVMETool

*LIBRARY AND DEMO
APPLICATION
SOFTWARE*

CAEN will repair or replace any product within the guarantee period if the Guarantor declares that the product is defective due to workmanship or materials and has not been caused by mishandling, negligence on behalf of the User, accident or any abnormal conditions or operations.

CAEN declines all responsibility for damages or injuries caused by an improper use of the Modules due to negligence on behalf of the User. It is strongly recommended to read thoroughly the CAEN User's Manual before any kind of operation.



CAEN reserves the right to change partially or entirely the contents of this Manual at any time and without giving any notice.

TABLE OF CONTENTS

1. GENERAL DESCRIPTION	4
1.1. OVERVIEW	4
2. SOFTWARE INSTALLATION.....	5
2.1. SOFTWARE INSTALLATION: GETTING STARTED	5
2.1.1. Software installation: Windows.....	5
2.1.1.1. Software installation: Lib folder	6
2.1.1.2. Software installation: Demo folder	6
3. SOFTWARE OVERVIEW	8
3.1. CAENVMETool LIBRARY	8
3.1.1. CAENVMETool library: Overview.....	8
3.1.2. CAENVMETool library: typical use	9
3.2. DEMOS SOFTWARE	9
3.2.1. Demos software: CAENScope	9
3.2.1.1. CAENScope: Overview	9
3.2.1.2. CAENScope: Main screen.....	9
3.2.1.3. CAENScope: Toolbar	10
3.2.1.4. CAENScope: Statusbar	11
3.2.1.5. CAENScope: Bottom panel	11
3.2.1.6. CAENScope: COMMON tab.....	11
3.2.1.7. CAENScope: TRIGGER tab.....	12
3.2.1.8. CAENScope: RECORD tab.....	12
3.2.1.9. CAENScope: Top-Left panel	13
3.2.1.10. CAENScope: VIEW tab.....	13
3.2.1.11. CAENScope: CURSORS tab	14
3.2.1.12. CAENScope: Top-Right pane.....	14
3.2.1.13. CAENScope: SCOPE tab.....	15
3.2.1.14. CAENScope: Settings.....	15

LIST OF FIGURES

FIG. 2.1: THE SOFTWARE & DOCUMENTATION PACK CD INTRODUCTION	5
FIG. 2.2 INSTALLATION FOLDER STRUCTURE.....	6
FIG. 2.3: PROGRAM MENU DEMO.....	6
FIG. 2.4: PROGRAM MENU LIBRARY	7
FIG. 3.1: SOFTWARE LAYERS.....	8
FIG. 3.2: MAIN SCREEN	10
FIG. 3.3: TOOLBAR.....	10
FIG. 3.4: ABOUT BOX	10
FIG. 3.5: STATUSBAR	11
FIG. 3.6: COMMON TAB.....	11
FIG. 3.7: TRIGGER TAB.....	12
FIG. 3.8: RECORD TAB.....	12
FIG. 3.9: VIEW TAB.....	13
FIG. 3.10: CURSORS TAB.....	14
FIG. 3.11: SCOPE TAB.....	15

LIST OF TABLES

TABLE 3.1: USER SETTINGS	16
--------------------------------	----

1. General description

1.1. Overview

CAENVMETool is a software package that facilitates the CAENVMELib usage: it consists of a library and a set of demo's applications showing the the CAENVMETool library usage.

CAENVMETool is a demo-application software set which allows an easy management of the CAENVMELib.

CAENVMETool library provides, for each supported board, API's which allows to perform the most common operations, such as configuration, readout, etc., masking the lowest level details, such as registers address, access type, data size, etc. to the developer.

Provided include-files (ANSI C) can be also used independently from the library; they provide all the definitions (registers address, access type, data size, registers bit mask, etc.) necessary to develop any application.

Application sets shows the most common use of CAENVMETool library for any supported board. These can be used to test the board operation and the CAENVMETool library usage.

2. Software installation

2.1. Software installation: Getting started

Place the CD in the CD tray in your PC, then the following window will open:

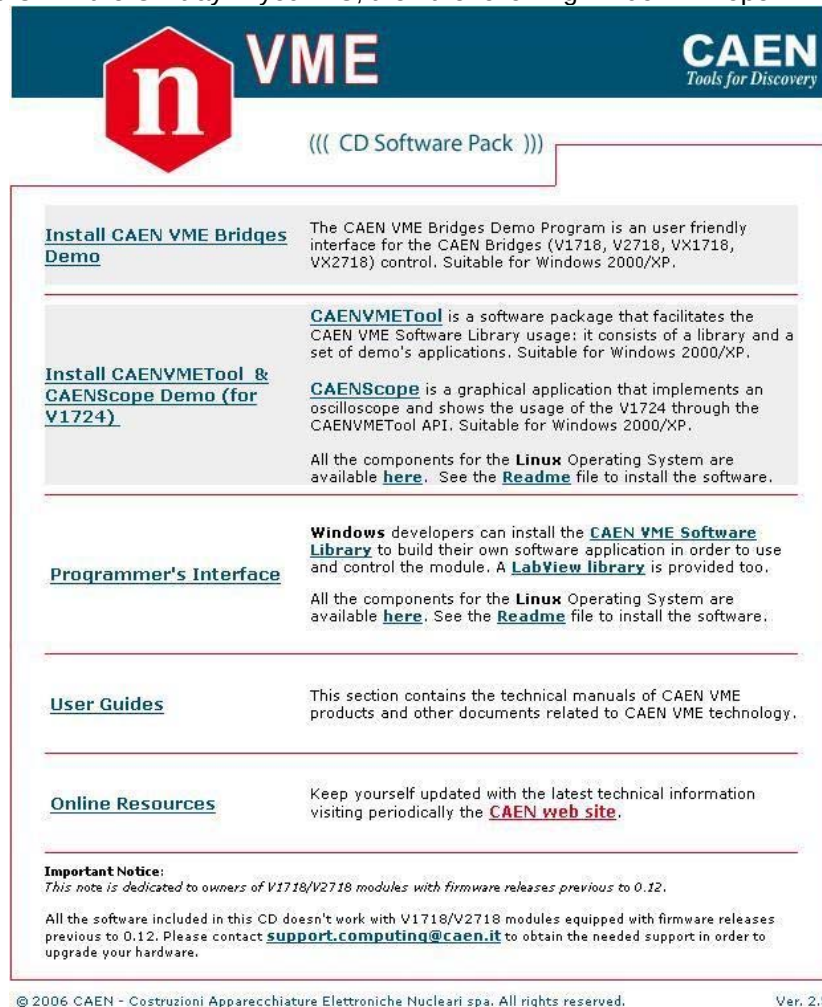


Fig. 2.1: The Software & Documentation Pack CD introduction

Click on "Install CAEN VMETool & CAENScope Demo" in order to install the provided user friendly interface which allows an easy and immediate control of the module.

2.1.1. Software installation: Windows

As installation is completed the structure of the created folders will be as follows:

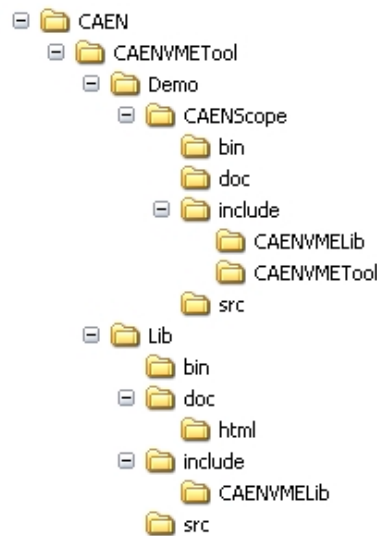


Fig. 2.2 Installation folder structure

2.1.1.1. Software installation: Lib folder

This folder contains the CAENVMETool library:

- *bin*: contains the compiled library as file *.lib
- *doc*: contains miscellaneous documentation on the library
- *include*: contains all the functions necessary to the operation and recompiling of the library
- *src*: contains the sources of the library.

A solution file for Microsoft Visual Studio 2003 (lib.sln), is provided in order to modify and recompile the library.

2.1.1.2. Software installation: Demo folder

This folder contains the demo applications of the library; each demo contains a sub folder with the following structure:

- *bin*: contains the application compiled as file *.exe in Release version
- *doc*: contains miscellaneous documentation on the demo application
- *include*: contains all the functions necessary to the operation and recompiling of the application. Third part libraries make exception (ex wxWidgets) and have to be installed through the proper tools.
- *src*: contains the sources of the application that this folder contains

A solution file for Microsoft Visual Studio 2003 (lib.sln), is provided in order to modify and recompile the library.

The structure of the applications menu will be as follows:

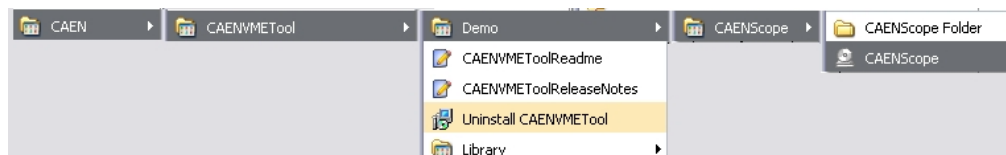


Fig. 2.3: Program menu demo

It is possible to read the release notes, the readme file, the library project folders and the demo applications.

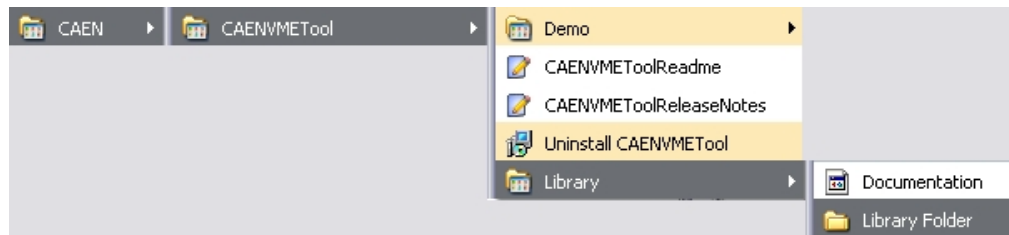


Fig. 2.4: Program menu library

3. Software overview

3.1. CAENVMETool library

3.1.1. CAENVMETool library: Overview

The CAENVMETool library is written in C language, and is a middleware tool between CAENVMELib and the User application.

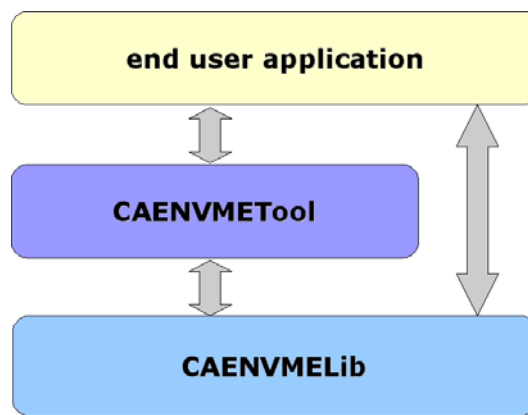


Fig. 3.1: Software layers

The library has a modular structure; it is made up by a set of files common to all boards (*cvt_board_commons.h*, *cvt_board_commons.c*, *cvt_common_defs.h*) and a couple of files made to measure for each board (*cvt_V####.c*, *cvt_V####.h*).

Supported boards are:

- V792
- V812
- V1190
- V1495
- V1724

Common definition files (*cvt_board_commons.h*, *cvt_common_defs.h*) provide data structure, definition and API's common to all boards, such as register read/write, set and reset of registers bitmasks etc.

The board definition files (*cvt_V####.c*, *cvt_V####.h*) provide data structure, definition and API's related to the board, such as registers address, access type (address modifier and data size), registers bitmask. API's and data structures extend the API's common to all boards.

3.1.2. CAENVMETool library: typical use

All the board modules are used as follows:

- Obtaining a valid *vme_handle* using CAENVMELib (see documentation of the CAENVMELib)
- Module opening: `cvt_V####_open( cvt_V####_data* p_data, UINT16 base_address, long vme_handle);`
- Usage of API's of the library: `cvt_V####_...` all the API's of the library use as input parameter `cvt_V####_data* p_data`, initialised on module's opening.
- Module's closing: `cvt_V####_close( cvt_V####_data* p_data);`
- Release of *vme_handle* using CAENVMELib (see documentation of the CAENVMELib).

3.2. Demos software

Demo softwares are applications which run on boards or board families and show the usage of the CAENVMETool.

These applications can be used both as example of library usage and to test the board operation.

3.2.1. Demos software: CAENScope

3.2.1.1. CAENScope: Overview

CAENScope is a graphical application that implements an oscilloscope and shows the usage of the V1724 through the API's provide by the CAENVMETool (`cvt_V1724.h`, `cvt_V1724.c`).

The software is written in C++ using the libraries of open source classes wxWidgets v.2.6.3 (<http://www.wxwidgets.org/>).

After installation, before launching the program, it is necessary to edit the configuration file and to set the parameters as shown in 3.2.1.14)

3.2.1.2. CAENScope: Main screen

All the commands and application visualisations are available from the main window, which is divided into three sections:

- *Toolbar*
- *Statusbar*
- *Bottom panel*: contains the tabs for setting the boards functions; this section can be hidden via the toolbar command.
- *Top-Left panel*: contains tabs which allows to modify visualization settings; this section can be hidden via the toolbar command.
- *Top-Right panel*: displays the signal visualisation

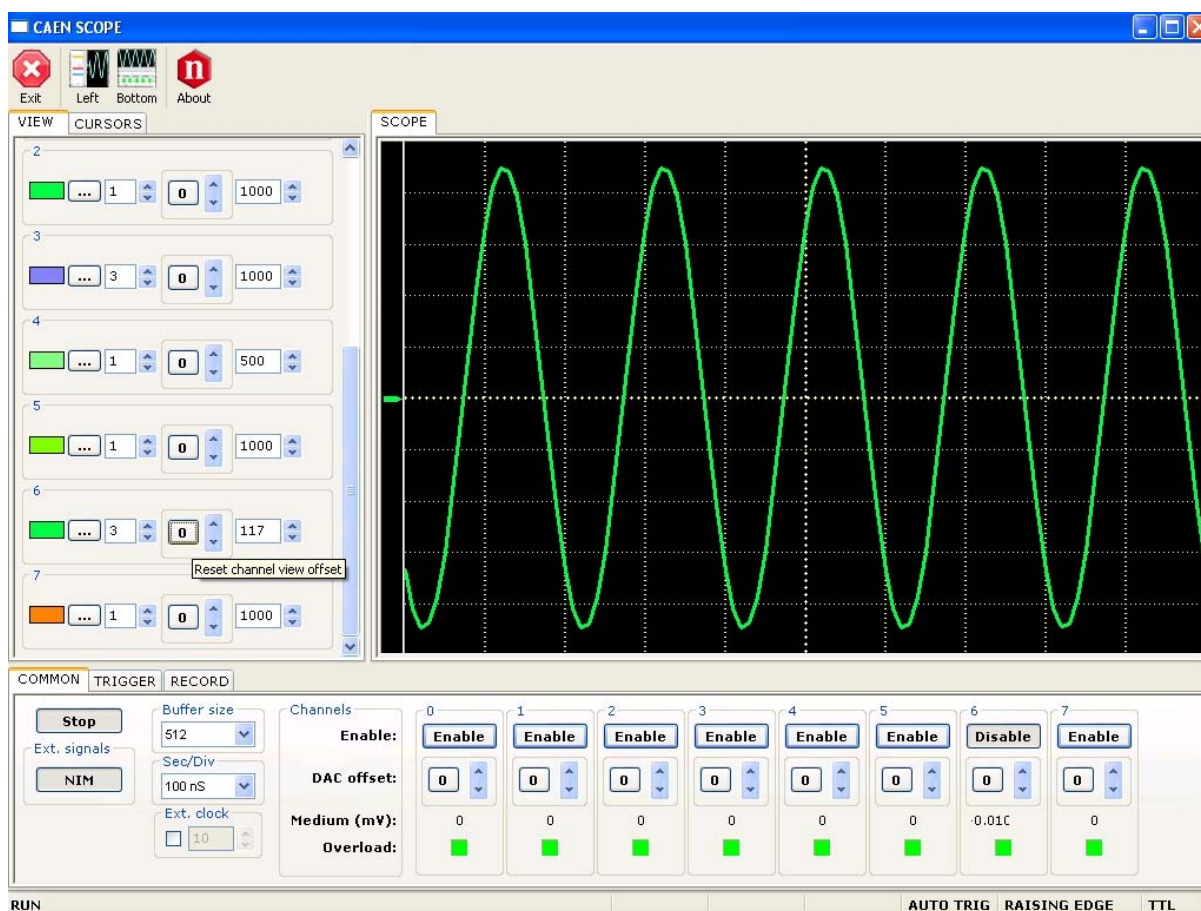


Fig. 3.2: Main screen

3.2.1.3. CAENScope: Toolbar

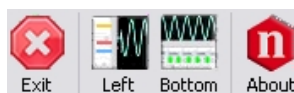


Fig. 3.3: Toolbar

The *Toolbar* contains the command and visualisation buttons:

- *Exit*: allows to close the application
- *Left*: allows to display/hide the *Left-Top panel*
- *Bottom*: allows to display/hide the *Bottom panel*
- *About*: allows to display the dialog about window (information about application and CAENVMETool library release)



Fig. 3.4: About box

3.2.1.4. CAENScope: Statusbar

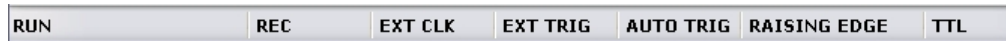


Fig. 3.5: Statusbar

The *Statusbar* provides information about the application operational:

- *RUN*: if displayed indicates that CAENScope is receiving data from the board
- *REC*: if displayed indicates that CAENScope is recording data from enabled channels on file.
- *EXT CLK*: if displayed indicates that the provided sampling clock is external. CAENScope uses this information only for the correct setting of the time reference.
- *EXT TRIG*: if displayed indicates that the trigger on external signal is enabled.
- *AUTO TRIG*: if displayed indicates that the software trigger is enabled; CAENScope sends continuously software trigger signals to the board.
- *RISING EDGE, FALLING EDGE*: if displayed indicates the triggering edge as the channel reaches the threshold value.
- *TTL, NIM*: indicates the type of signal used on the front panel of the board.

3.2.1.5. CAENScope: Bottom panel

This section contains the controls to set the CAENScope functions. It is composed by three tabs:

- *COMMON*: general commands
- *TRIGGER*: triggering commands
- *RECORD*: data record commands

3.2.1.6. CAENScope: COMMON tab

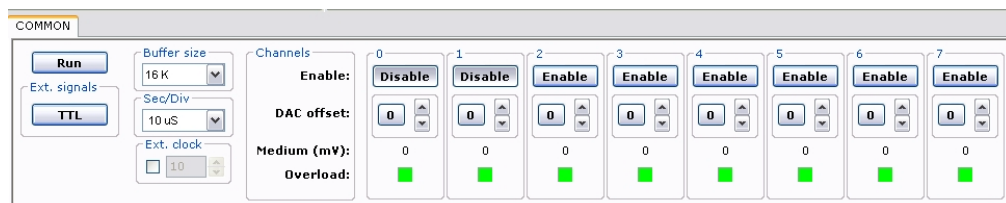


Fig. 3.6: COMMON tab

This tab contains the controls for setting the common functions of CAENScope:

- *Run/Stop*: allows to enable/disable the request of samples from the data buffer of the board.
- *Ext. signals: NIM, TTL*: allows to set the level of the external signal used on the board: NIM or TTL. CAENScope sends to the board the right command for the selected setting.
- *Buffer size*: allows to select the size of data buffer required to the board for single access.
- *Sec/Div*: allows to set the displayed time scale.
- *Ext Clock*: if enabled, CAENScope uses as sampling rate (MHz) the set value, otherwise the internal value (100 MHz) is used.
- *Channels enable*: allows to enable/disable the channel visualisation.
- *Channels DAC offset*: allows to set the offset added to the input channel (press 0 to reset).

- *Channels medium (mV)*: provides the average value of the signal (averaged over the samples in the buffer).
- *Channels overload*: indicates saturation (red), values high or low, normal values (green).

3.2.1.7. CAENScope: TRIGGER tab

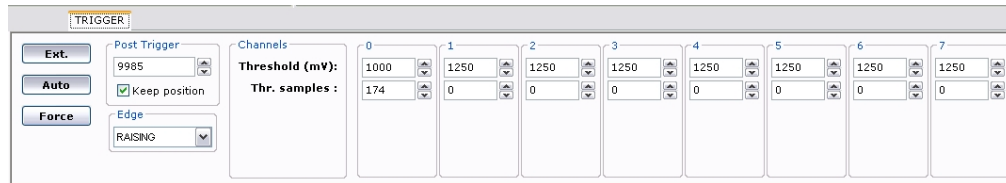


Fig. 3.7: TRIGGER tab

This tab contains controls for setting the trigger functions:

- *Ext*: allows to enable/disable the trigger on the external signal.
- *Auto*: allows to enable/disable the production of continuous software trigger commands.
- *Force*: forces the trigger; sends an unique software trigger command.
- *Post Trigger*: allows to set the post trigger value, i.e. the number of samples from the bottom of the acquired buffer.
- *Post Trigger Keep position*: if enabled allows to keep the post trigger point fixed on the display as the time scale changes (CAENScope automatically adjusts the horizontal offset).
- *Edge NO TRIG, RAISING, FALLING*: allows to select the edge for the trigger, as the threshold, set for each channel, is exceeded. This setting is common to all channels.
- *Channels Threshold (mV)*: threshold for triggering on a particular channel.
- *Channels Thr. samples*: number of samples for which the trigger condition must be kept (threshold exceeded on leading/trailing edge) before trigger production.

3.2.1.8. CAENScope: RECORD tab

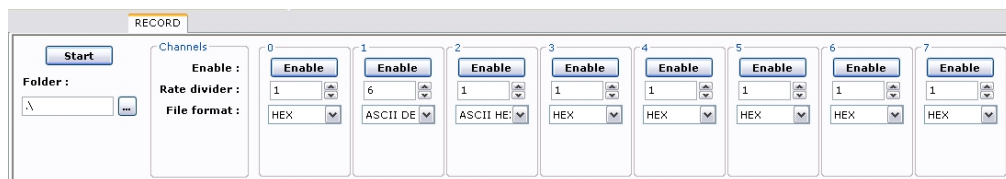


Fig. 3.8: RECORD tab

This tab contains the controls for setting the functions of data recording on file:

- *Start/Stop*: it allows to start/stop data recording on file: for each enabled channel a file in the selected destination folder is produced (in the selected format). The filename is the following:

caen_scope_B00C##_YYYY_MM_DD_hh_mm_ss.ext where:

- *##*: channel index
- *YYYY*: current year
- *MM*: current month
- *DD*: current day
- *hh*: current hour
- *mm*: current minute
- *ss*: current second

- *ext*: extension depending on file format
 - *bhx*: binary format: data are saved on 4 Byte word, in two complement format.
 - *adc*: decimal ASCII format: data are saved on distinguished lines (separated by `\n\r`) in decimal ASCII format.
 - *ahx*: hexadecimal ASCII format: data are saved on distinguished lines (separated by `\n\r`) in hexadecimal ASCII format.
- *Folder*: allows to select the folder where files are saved. It is possible to edit directly either absolute or relative paths; each editing is confirmed by pressing `<ENTER>`. The field has red background if the path is incorrect or not confirmed. By Clicking on the button (...) it is possible to select the existing path via graphical interface.
- *Channels enable*: allows to enable/disable the channel for data recording. Each enabled channel must be enabled for acquisition (tab *Common:Channels enable*), because no samples are available for disabled channels.
- *Channels rate divider*: allows the setting of a samples rate divider; CAENScope saves one sample every N acquired (N is the setting of *rate divider*).
- *Channels file format*: allows the setting of the output files setting:
 - *HEX*: binary format: data are saved on 4 Byte word, in two complement format.
 - *ASCII DEC*: decimal ASCII format: data are saved on distinguished lines (separated by `\n\r`) in decimal ASCII format.
 - *ASCII HEX*: hexadecimal ASCII format: data are saved on distinguished lines (separated by `\n\r`) in hexadecimal ASCII format.

3.2.1.9. CAENScope: Top-Left panel

This section contains the controls for setting the visualisation of CAENScope. The controls are:

- *VIEW*: general settings.
- *CURSORS*: cursor settings.

3.2.1.10. CAENScope: VIEW tab

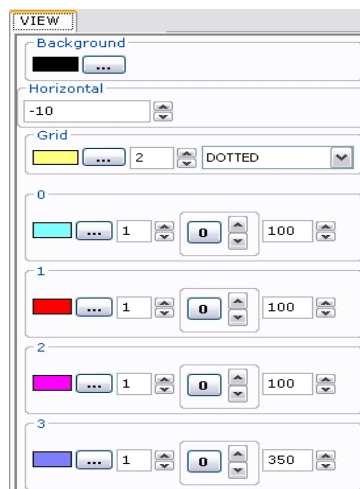


Fig. 3.9: VIEW tab

This tab contains the controls of visualisation settings:

- *Background*: allows to set the oscilloscope background color
- *Horizontal*: allows to set the horizontal visualisation offset (μ s).
- *Grid*: allows to set the following grid parameters:

- color
- line size (Pixel)
- line type
 - NO GRID
 - SOLID
 - DASHED
 - DASH-DOT
 - DOTTED
- Channel #: allows to set the channel visualisation:
 - color
 - line size (Pixel)
 - Vertical visualisation offset; by clicking (0) the offset is reset. A marker on the left of the oscilloscope window shows the signal 0 level.
 - Vertical resolution (mV/Div)

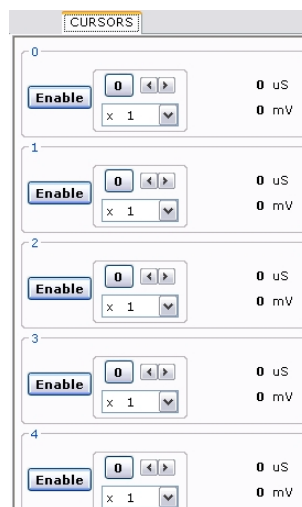


Fig. 3.10: CURSORS tab

3.2.1.11. CAENScope: CURSORS tab

This tab contains the cursors settings:

- *Enable*: allows to enable/disable the channel cursors:
- Cursor position: by clicking on (0) the cursor position is reset. The multiplying factor sets the cursor increase step.
- ### uS: cursor position (μ s).
- ### mV: signal value at the cursor position.

3.2.1.12. CAENScope: Top-Right pane

This section contains one tab (SCOPE) which represents the oscilloscope window:

- *VIEW*: general setting.
- *CURSORS*: cursor setting.

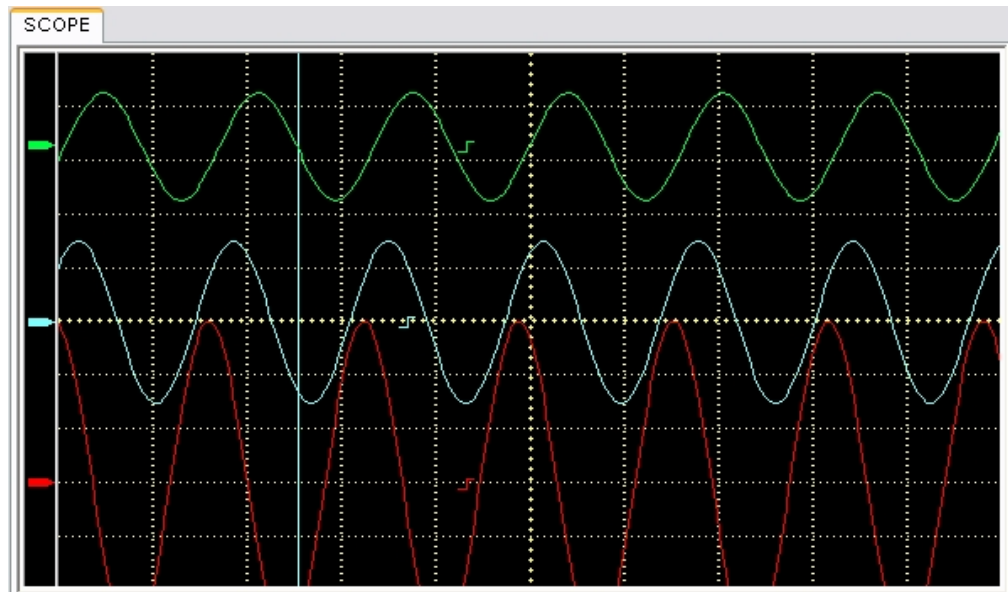
3.2.1.13. CAENScope: SCOPE tab

Fig. 3.11: SCOPE tab

The oscilloscope window shows:

- The grid
- The enabled channels, with the following options:
 - Marker of trigger position
 - Marker of vertical position
 - Cursor (if enabled)

3.2.1.14. CAENScope: Settings

The main settings of CAENScope are in series on a text file (*settings.txt*). This file has a tree structure, and can be modified via a common text editor.

The configuration file is loaded as the application is launched and saved as the application is exited. All the modifications made manually editing the file during the application execution will be overwritten as the application is exited. Normally the User will not edit manually, since it is better to use the CAENScope commands in order to edit the functional parameters. Some parameters, used by CAENScope in read-only mode, make exception and must be set **una tantum** by the User; such parameters are written in **bold** in the following.

The file is divided into sections, defined by an alphanumeric identifier

The section identifier has the following format:

'[{parent-list}]identifier' where :

- *parent-list*: identifier of the father section (not present in the root).
- *identifier*: unique alphanumeric identifier of parent section (or root)

Example :

[SEZ1] identifies section SEZ1

[SEZ1/SUB1] identifies section SUB1 inside SEZ1

Each section is composed by couples *token=value* where:

- *token*: unique alphanumeric identifier inside section, associated to the parameter to be set.
- *value*: token value; value type (int, double, string etc) depends on the relevant parameter.

Example:

item1=10.0 gives value 10.0 to item1
item2=text String gives value 10 to item1
 one section can include several daughter sections:

```
[RECORD]
  RECORD_FOLDER=./
[TRIGGER]
  TRIGGER_MSK=4
  POST_TRIGGER=12060
  TRIGGER_POSITION_KEEP=1
  TRIGGER_EDGE=1
  SAMPLE_BLOCK_SIZE=5
[MISC]
  VME_BOARD_TYPE=V2718
  VME_LINK=0
  VME_BOARD_NUM=0
  GRID_LINE_COLOR=0xffff80
  GRID_LINE_TYPE=4
  GRID_LINE_WIDTH=2
  OFFSET_SEC=0
  INT_RATE_MHZ=100
  EXT_RATE_MHZ=10
  USE_EXT_CLOCK=0
  USE_TTL=1
  SEC_2_DIV=100000
  MAX_LOG_X=16384
  BACKGROUND_COLOR=0x000000
  ACQ_BOARD_NUM=1
[ACQ_BOARDS]
  [ACQ_BOARDS/#]
    BOARD_TYPE=V1724
    ADDRESS=0x3210
    NUM_CHANNEL=8
  [ACQ_BOARDS/##]
    ENABLED=1
    LINE_COLOR=0x80ffff
    LINE_WIDTH=1
    OFFSET_Y=0
    VOLT_2_BIT=6553.6
    VOLT_2_DIV=10
    DAC_OFFSET_BIT=20
    DAC_VOLT_2_BIT=26214.4
    TRIGGER_THRESHOLD_VOLT=1
    TRIGGER_THR_SAMPLE=620
    RECORD_RATE_DIVIDER=1
    RECORD_ENABLED=1
    RECORD_FORMAT=1
```

The parameter that the User has to edit (in **bold**) at least one time prior to application launch, are described in the following table.

Table 3.1: User settings

ITEM	VALUE	DESCRIPTION
[MISC]::VME_BOARD_TYPE	V1718 V2718	Allows to set the used VME bridge
[MISC]::VME_LINK	[0..N-1]	Allows to set the link number (see documentation of CAENVMELib)
[MISC]::VME_BOARD_NUM	[0..N-1]	Allows to set the board number (see documentation of CAENVMELib)
[ACQ_BOARDS/#]::ADDRESS	0x#### ####	Allows to set MSW of the VME address of the board (Hex or Dec format)