Technical Information Manual

Revision n. 1

28 September 2009

CAENV1x90

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

LIST OF FIGURES

LIST OF TABLES

1. CAEN software Tools

CAEN provides software Tools for Powered Crates and Front-End/Data Acquisition Modules. Such packages include Libraries, Demos and Software tools for Windows and Linux. The packages' main features are:

- Libraries for National Instruments LabVIEW and C/C++
- Demo programs in source code C/C++ (Windows and Linux) and LabVIEW (Windows) as a starting point for the development of user-specific applications
- Software Tools (firmware upgrade, Module configuration...)
- Windows 2000/XP/Vista. and Linux supported

Packages are available for FREE download at the webpage:
http://www.caen.it/nuclear/software_tools.php

Table 1.1: Software Tools

Module	Function	Software Tools
V1720 V1721 V1724 V1731 V1740 VX1720 VX1721 VX1724 VX1731 VX1740	VME/VME64X Digitizer family	CAEN Digitizer
V1190A V1190B V1290A V1290N VX1190A VX1190B VX1290A VX1290N	VME/VME64X Multihit TDCs	CAENV1x90
V1718 VX1718	VME/VME64X-USB2.0 Bridge	CAENVMElib
V2718 VX2718	VME/VME64X-PCI Optical Link Bridge	CAENVMElib
V792 V792N V862 V965 V965A	VME Multievent QDC	CAENqtp
V785A V785N V1785	VME Multievent Peak sensing ADC	
V775A V775N	VME Multievent TDC	
V812 V814 V895	VME Discriminators	CAENdiscri
V1495	General Purpose VME Board	CAENV1495
N957	NIM 8k Multi-Channel Analyzer	CAEN N957
N1568B	NIM 16Ch Programmable Pulse Shape Amplifier & Dual CFD	CAEN N1568
A2818	PCI CONET Controller	CAEN Link CAENVMElib
SY2791	TPC Readout System	CAEN Link
VME8100	8U 21 Slot VME64/64X Enhanced Crate Series	CAEN_VME8100
NIM8301	7U 12 slot smart fan unit 300/600 W Crate	CAEN_NIM8301

1.1. CAENV1x90 overview

CAENV1x90 is a software package provided both as demo-application software set which allows an easy management of the CAENVMELib and as LabVIEW Virtual Instruments application.

CAENV1x90 library provides, 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 CAENV1x90 library for any supported board. These can be used to test the board operation and the CAENV1x90 library usage.

1.2. System Requirements

- CAEN V1718 USB-VME Bridge and/or CAEN A2818 PCI-CONET Board
  - In case of different VME controller (for example a CPU or other types of VME bridges), the user must replace the CAENVMElib with the specific I/O library or add a wrapper library, providing the functions for the VME access (essentially the single read/write access and the Block Transfer read).
- Windows 2000/XP/Vista or Linux kernel Rel. 2.4/2.6 with gnu C/C++ compiler

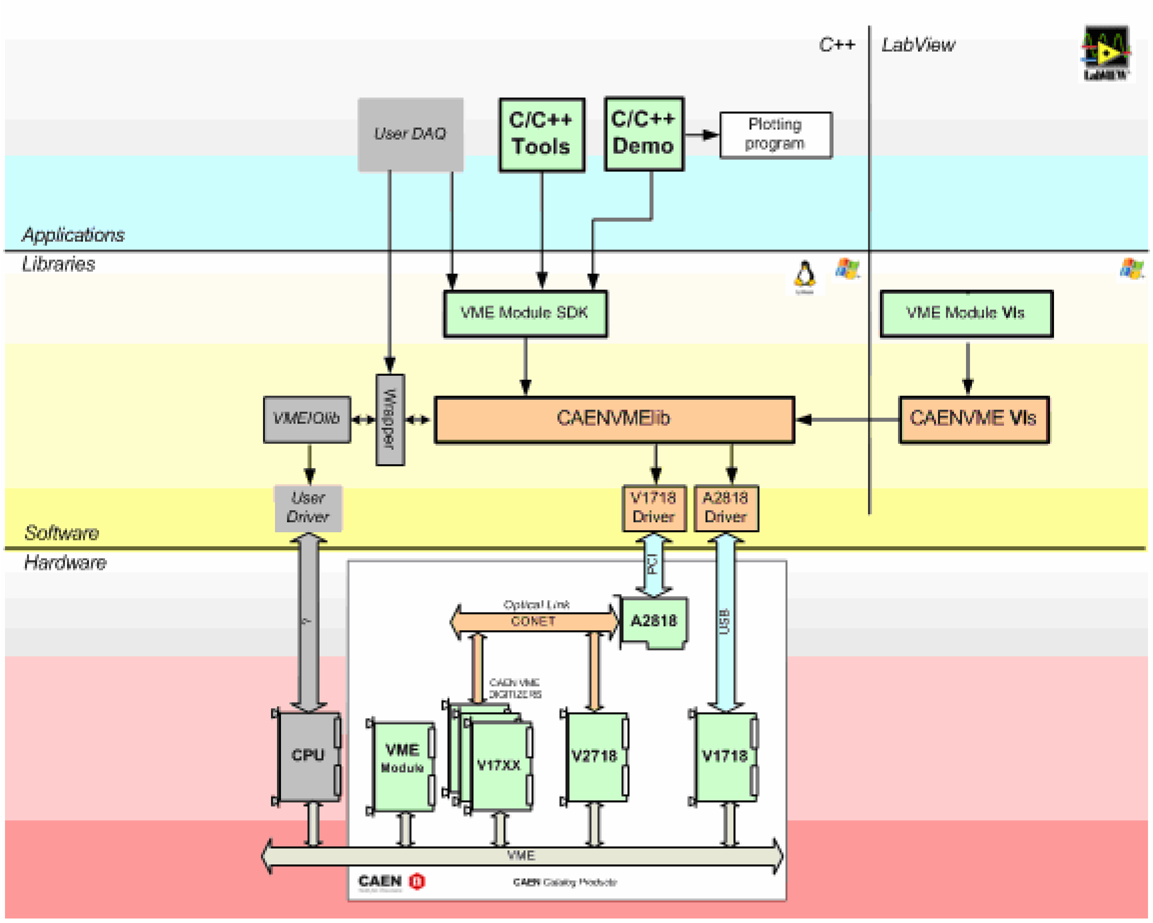


Fig. 1.1: CAEN Software Tools Structure for VME Boards

1.3. Supported Boards

- V1190A 128 Channel Multihit TDC
- V1190B 64 Channel Multihit TDC
- V1290A 32 Channel Multihit TDC
- V1290N 16 Channel Multihit TDC
- VX1190A 128 Channel Multihit TDC
- VX1190B 64 Channel Multihit TDC
- VX1290A 32 Channel Multihit TDC
- VX1290N 16 Channel Multihit TDC

2. Software installation

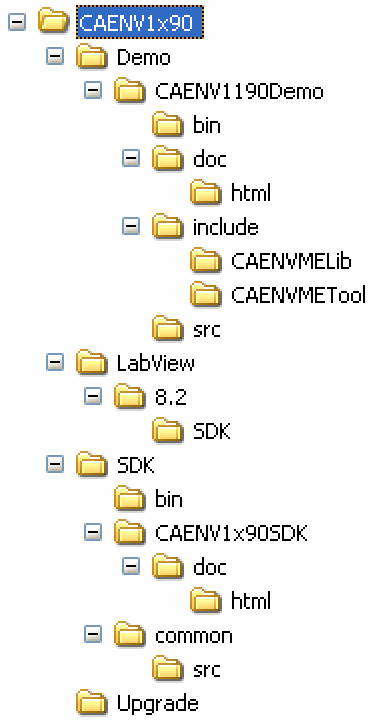
2.1. Software installation: Getting started

The following instructions will help through the module installation;

1. Download the Software Development Kit available at the product page of the supported boards described in §1.3: (for example Mod. V1190A link: http://www.caen.it/nuclear/product.php?mod=V1190A)
2. Unzip the package on your computer; and open the CAENV1x90 folder
3. Now open folder related to Your OS (Linux/Windows)
   a. Windows Users shall launch the CAENdigitizerSetup-1.0.exe file
   b. Linux Users must install the CAENVMELib contained in the relevant folder

2.1.1. Software installation: Windows

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



```
CAENV1x90
    Demo
        CAENV1190Demo
            bin
            doc
                html
            include
                CAENVMELib
                CAENVMETool
            src
    LabView
        8.2
            SDK
    SDK
        bin
        CAENV1x90SDK
            doc
                html
        common
            src
    Upgrade
```

Fig. 2.1: Installation folder structure

It is then possible to access the installed packages from the program thumbnails as shown in the following figures.

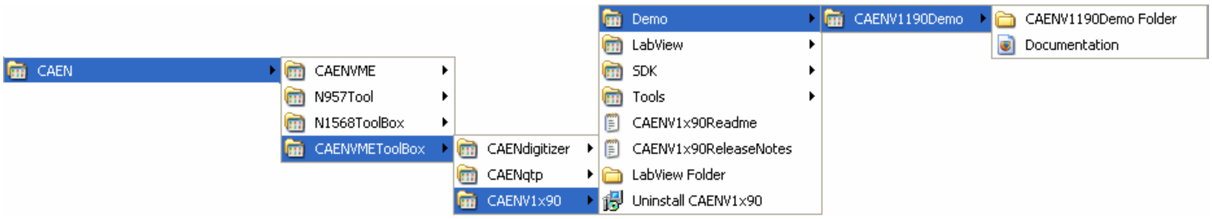


Fig. 2.2: Program menu CAENV1x90 demo

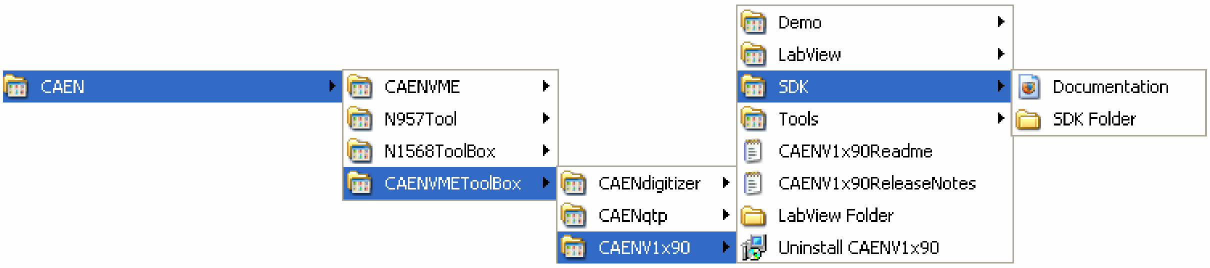


Fig. 2.3: Program menu SDK doc and source directory

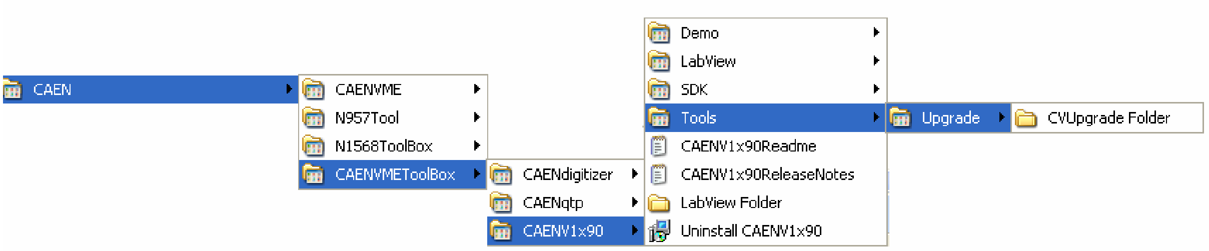


Fig. 2.4: Program menu CVupgrade

Documentation is an html source code documentation created with Source code documentation generator tool Doxygen (see www.doxygen.org)

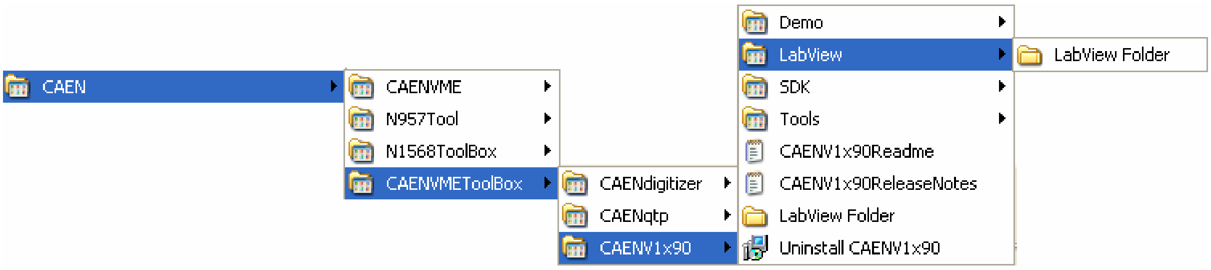


Fig. 2.5: Program menu LabVIew demo

3. Software overview

3.1. CAENV1x90 SDK

3.1.1. CAENV1x90 SDK: Overview

The CAENV1x90 SDK library is written in C language, and is a middleware tool between CAENVMELib and the User application.

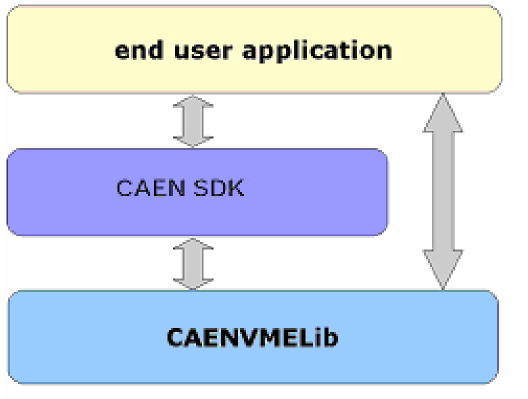


Fig. 3.1: Software layers

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:

- V1190A 128 Channel Multihit TDC
- V1190B 64 Channel Multihit TDC
- V1290A 32 Channel Multihit TDC
- V1290N 16 Channel Multihit TDC
- VX1190A 128 Channel Multihit TDC
- VX1190B 64 Channel Multihit TDC
- VX1290A 32 Channel Multihit TDC
- VX1290N 16 Channel Multihit TDC

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.
Source code documentation is generated with DOXYGEN (www.doxygen.org)

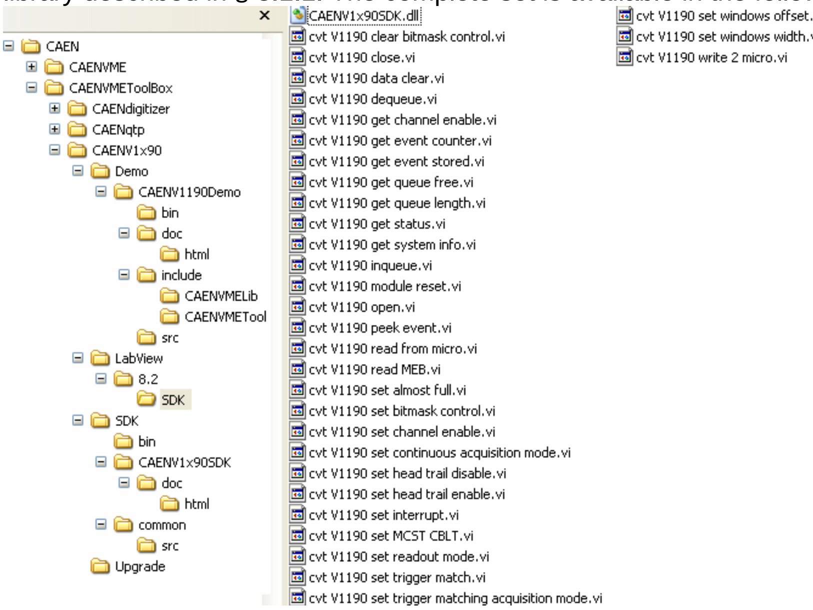
3.1.2. CAENV1x90 SDK: typical use

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. CAENV1x90 LabVIEW

V1x90vi is a set of LabVIEW Virtual Instruments which provide the capabilities of the library described in § 3.1.1. The complete set is available in the following folder



```
⊟ CAEN
    ⊞ CAENVME
    ⊟ CAENVMEToolBox
        ⊞ CAENdigitizer
        ⊞ CAENqtp
        ⊟ CAENV1x90
            ⊟ Demo
                ⊟ CAENV1190Demo
                    bin
                    ⊟ doc
                        html
                    ⊟ include
                        CAENVMELib
                        CAENVMETool
                    src
            ⊟ LabView
                ⊟ 8.2
                    SDK
            ⊟ SDK
                bin
                ⊟ CAENV1x90SDK
                    ⊟ doc
                        html
                ⊟ common
                    src
            Upgrade
```

```
CAENV1x90SDK.dll                             cvt V1190 set windows offset.vi
cvt V1190 clear bitmask control.vi           cvt V1190 set windows width.vi
cvt V1190 close.vi                           cvt V1190 write 2 micro.vi
cvt V1190 data clear.vi
cvt V1190 dequeue.vi
cvt V1190 get channel enable.vi
cvt V1190 get event counter.vi
cvt V1190 get event stored.vi
cvt V1190 get queue free.vi
cvt V1190 get queue length.vi
cvt V1190 get status.vi
cvt V1190 get system info.vi
cvt V1190 inqueue.vi
cvt V1190 module reset.vi
cvt V1190 open.vi
cvt V1190 peek event.vi
cvt V1190 read from micro.vi
cvt V1190 read MEB.vi
cvt V1190 set almost full.vi
cvt V1190 set bitmask control.vi
cvt V1190 set channel enable.vi
cvt V1190 set continuous acquisition mode.vi
cvt V1190 set head trail disable.vi
cvt V1190 set head trail enable.vi
cvt V1190 set interrupt.vi
cvt V1190 set MCST CBLT.vi
cvt V1190 set readout mode.vi
cvt V1190 set trigger match.vi
cvt V1190 set trigger matching acquisition mode.vi
```

Fig. 3.2: LabVIEW Virtual Instruments folder

The VI representation is as follows:

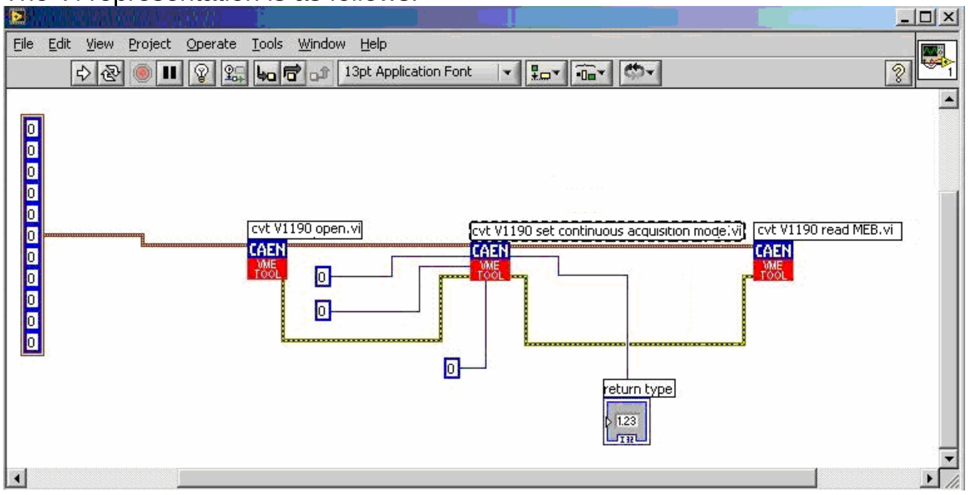

File Edit View Project Operate Tools Window Help
13pt Application Font
cvt V1190 open.vi
cvt V1190 set continuous acquisition mode.vi
cvt V1190 read MEB.vi
CAEN VME TOOL
0
return type
1.23

Fig. 3.3: LabVIEW VI representation

4. Demo softwares

Demo softwares are applications which run on boards or board families and show the usage of the SDK.
These applications can be used both as example of library usage and to test the board operation.

4.1. Demos software: CAENV1190Demo

4.1.1. CAENV1190Demo: Overview

CAENV1190Demo is a command line application which shows the operation of the board family V1X90 (V1190A, V1190B, V1290A, V1190N) through the APIs exposed by CAENVMETool (cvt_V1190.h, cvt_V1190.c).
CAENV1190Demo sets the board according to the provided parameters and executes read cycles from it. He readout result is released on distinguished files, both in native binary format and in interpreted text type.
The execution of the various operation is displayed with its result (Ok or Error message).

4.1.2. CAENV1190Demo: Settings

CAENV1190Demo can be configured via command line via a set of parameters, whose format is shown in the following:

By launching the application with parameter -h, it is displayed the list of available parameters.

-param_id[param_value]

where:

- param_id: a character which identifies the parameter.
- param_value: value of parameter, if foreseen by the parameter itself.
- Notes about the parameters use:
  - if a parameter is not provided, the default value is assumed
  - the presentation order of the parameters is arbitrary
  - a parameter not recognised is neglected
  - each parameter must be separated from the others via one or more space characters
  - the case of the parameters (id and value) is neglected.
  - space characters must not be typed between the parameter id and value (ex. -tV1190A OK , -t V1190A WRONG)
  - if the parameter value foresees space characters S (ex. File names or acquisition parameters), this must be enclosed between “” without space characters with respect to parameter id (ex -m" 1, 1, 0, -1, -1, -1, -1, 1, 0, 0xffff , 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff")

List of available parameters with relevant default value and use example:

- **-p***filename*: output filename. If not specified '*parsed_values.dat*' will be assumed. example:
  - *CAENV1190Demo -p"parsed data out.dat"*

- *CAENV1190Demo -Pparsed_data_out.dat*

- **-r***filename*: raw output filename. If not specified '*raw_values.dat*' will be assumed
  example:
  - CAENV1190Demo -r"raw data out.dat"
  - CAENV1190Demo -Rraw_data_out.dat
- **-a***base_address*: the board's base address (16 bit MSW), expressed in decimal or hexadecimal format. If not specified '*0x3210*' will be assumed.
  example:
  - CAENV1190Demo -a4660
  - CAENV1190Demo -A0x1234
- **-t***v1190_type* : The V1190's type. Valid values are: *V1190A, V1190B, V1290A, V1290N*. If not specified '*V1190A*' will be assumed.
  example:
  - CAENV1190Demo - tV1190A
  - CAENV1190Demo -TV1290N
- **-e***num_events*: the number of events to acquire. If not specified '*-1*' will be assumed. A non positive number means don't care (i.e. acquire until key pressed)
  example:
  - CAENV1190Demo -e1024
  - CAENV1190Demo -E32768
- **-c**"*params*": Continuos acquisition mode parameters. If not specified trigger matching mode will be assumed. The parameter string format is:
  "*edge_detection, m_res_width, enable_msk[0..3]*"
  where :
  - o *edge_detection* : The edge detection type numeric value
  - o *res_width* : The resolution width numeric value
  - o *enable_msk[0..3]*: The channel enable pattern buffer (hex or dec value). You must provide the number of required words depending on the board type (up to 8 words). If some words are missing, the corresponding channels will be disabled.

  Example (for a V1190A):
  - CAENV1190Demo -c" 2, 0, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff"
  - CAENV1190Demo -C" 2, 0, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff"

- **-m**"*params*" trigger matching mode parameters. If not specified '*1,1,0,-1,-1,-1,-1,1,0*' will be assumed. The parameter string format is:
  "*header_trailer_enable, empty_event_enable, trigger_time_tag_enable, window_width, window_offset, extra_search_margin, reject_margin, edge_detection, m_res_width, enable_msk[0..3]*"
  where :
  - o *header_trailer_enable*: Enable(1)/disable(0) header and trailer
  - o *empty_event_enable*: Enable(1)/disable(0) empty event storage
  - o *trigger_time_tag_enable*: Enable(1)/disable(0) trigger time tag
  - o *window_width*: Sets the width of the match window
  - o *window_offset*: Sets the offset of the match window with respect to the trigger itself
  - o *extra_search_margin*: Sets the extra search field of the match window
  - o *reject_margin*: Sets the reject margin, expressed in clock cycles
  - o *edge_detection*: The edge detection type numeric value
  - o *res_width*: The resolution width numeric value
  - o *enable_msk[0..3]* : The channel enable pattern buffer (hex or dec value). You must provide the number of required words depending on the board

type (up to 8 words). If some words are missing, the corresponding channels will be disabled.

Example (for a V1190A):

- CAENV1190Demo -m" 1, 1, 1, 0x14, 0xFD8, 2, 2, 2, 0, 0xffff, 0xffff, 0xffff, 0xffff 0xffff, 0xffff, 0xffff, 0xffff"
- CAENV1190Demo -M" 1, 1, 1, 0x14, 0xFD8, 2, 2, 2, 0, 0xffff, 0xffff, 0xffff, 0xffff 0xffff, 0xffff, 0xffff, 0xffff"

- **-h:** Shows this help screen

  example:
  - CAENV1190Demo -h
  - CAENV1190Demo -H

4.1.3. CAENV1190Demo: Raw output data format

This output binary type file reports the unprocessed readout data.

4.1.4. CAENV1190Demo: Parsed output data format

This text format output file report a processed version of readout data.
For each 32 bit readout datum:

- Data type is reported
- The datum content is processed, single field are split and written onto file in hexadecimal format.

  The file format is as follows:
  - **GLB_HDR** : global header
    - **EVT COUNT** : event counter
    - **GEO** : geo address
  - **GLB_TRL** : global trailer
    - **STATUS** : status word
    - **WCOUNT** : event number of words
    - **GEO** : geo address
  - **TDC_HDR** : TDC's header
    - **TDC** : number di TDC
    - **EVT ID** : id event
    - **BUNCH ID**: bunch id
  - **TDC_MSR** : acquisition measure
    - **TRAILING** : Trailing (1) or Leading (0)
    - **CH** : channel datum is from
    - **MEASURE**: performed measure
  - **TDC_ERR** : error
    - **TDC** : number of TDC
    - **ERR_FLAGS** : error flags
  - **GLB_TRL** : TDC's header
    - **TDC** : number di TDC
    - **EVT ID** : id event
    - **WCOUNT** : number of word
  - **TDC_TRG** : error
  - **TRG TIME TAG** : The trigger time tag
  - **FILLER** : filler data