SIS3316
16 Channel VME Digitizer

Software Instruction Guide
“SIS3316_root_gui”

SIS GmbH
Harksheider Str. 102A
22399 Hamburg
Germany

Phone: ++49 (0) 40 60 87 305 0
Fax: ++49 (0) 40 60 87 305 20

email: info@struck.de
http://www.struck.de

Version: SIS3316-M-Application-Software-Root-Gui-V100.doc as of 12.08.2015

Revision Table:

Revision	Date	Modification
1.00	12.08.2015	Initial release

Table of contents

1 Introduction

This document will guide you through the build process of the software project "sis3316_root_gui" which is helpful to demonstrate the use of the software and libraries, that are shipped with the module and to facilitate initial sampling with the digitizer and display of the acquired signals. It is not optimized for speed, data rate and special applications like gamma-, neutron/gamma and …. . It illustrates a lot of capabilities of the sis3316 like trigger generation, trigger selection, …, however and it can demonstrate the excellent resolution (output code histograms, FFT).
All project files and libraries are available in source code.

The software allows you to:
- update the FPGA Firmware
- configure the SIS3316 (analog offset, input range, …)
- start sampling: software trigger, external trigger, internal trigger
- graph: display signal(s)
- graph: display Trigger FIR filter trapezoid
- graph: display Energy FIR filter trapezoid
- graph: display FFT
- histogram: display ADC output code
- save acquired Hits/Events to file(s)
- load/save configuration data

1.1 Operating System

The software project "sis3316_root_gui" is based on the CERN ROOT package and supports the three platforms:
- Windows (WIN7) built with the development tool VisualC++ 2010 (VC10)
- Scientific Linux 6.6 built with the development tool Eclipse (3.6.1)
- Mac OS 10.9.4 built with the development tool Eclipse (Kepler Service Release 2)

The OS is selected with "#define(s)" in the file project_system_define.h.

Note: at this point in time ROOT 6.x does not support Windows (yet?)

1.2 Graphical Interface

The program package ROOT (http://root.cern.ch) is used as Graphical User Interface (GUI) and to display graphs and histograms.
To build/run the program, ROOT has to be installed. The projects from the DVD are linked with ROOT version 5.34.18.
The user can start program "sis3316_root_gui" immediately provided that ROOT version 5.34.18 is installed.

1.3 Communication Interface

The software provides several interface options to access SIS3316 on board resources (control, write and read) from your computer:
- VME Interface sis3100/sis310x (opical interface)
- VME Interface sis3150 (USB2 interface)
- VME Interface sis3153 (USB3 interface)
- Ethernet interface (UDP)

The interface is selected with a "#define" in the file project_interface_define.h.

The user can replace the SIS-interface files (vme_interface_class.h, sis1100w_vme_class.h/cpp, sis3150w_vme_class.h/cpp, sis3153w_vme_class.h/cpp, sis3316_ethernet_access_class.h/cpp) with his own interface files to build the program to run with other VME interfaces or single board computers (SBC).

1.4 Executables

1.4.1 Windows 7 (Windows 8)

The user can start the program "sis3316_root_gui.exe" immediately provided that:

- ROOT 5.34.18 is installed
- VME crate powered with at least one Struck interface and at least one SIS3316
- And the following files are in the same directory:
  - sislogo.bmp
  - libfftw3-3.dll

Depending on the used interface you will find the executable in the following directories:
- sis3316_root_gui_ethernet
- sis3316_root_gui_sis310x
- sis3316_root_gui_sis3150usb (USB2)
- sis3316_root_gui_sis3153usb (USB3)

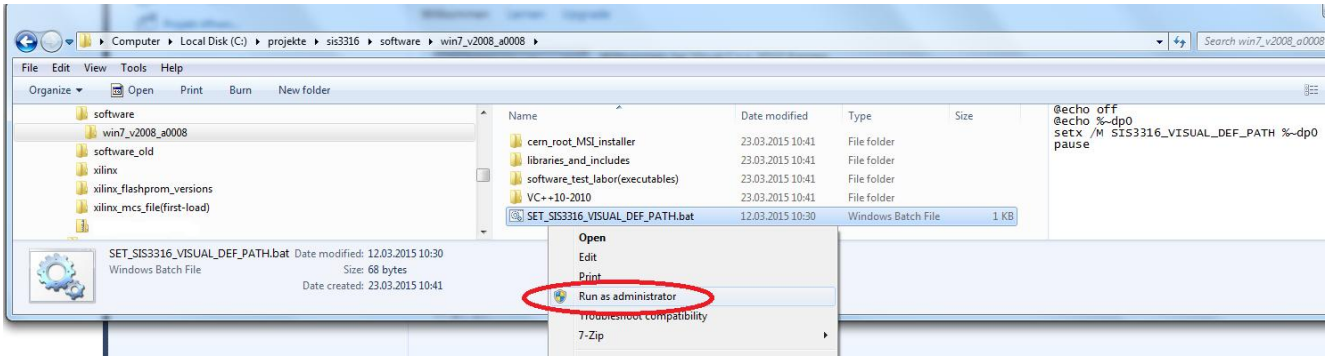
2 Build the program “sis3316_root_gui”

The development tools **VisualC++ 2010** and **Eclipse 3.6.1** are used to build (compile and link) the executable. Valid settings for Include and Library paths are needed for the development tools.

2.1 Windows 7 (Windows 8)

The Development Tool **VisualC++ 2010** is used to build the project.

Copy the directory “sis3316/software/win7_v200A_a000A” from the DVD to your local disk. For example to c:/projects/sis3316/software.
Execute the batch file “SET_SIS3316_VISUAL_DEF_PATH.bat” as administrator. It sets the environment variable SIS3316_VISUAL_DEF_PATH with the path of the project. **VisualC++ 2010** uses this variable for the include and library path settings.

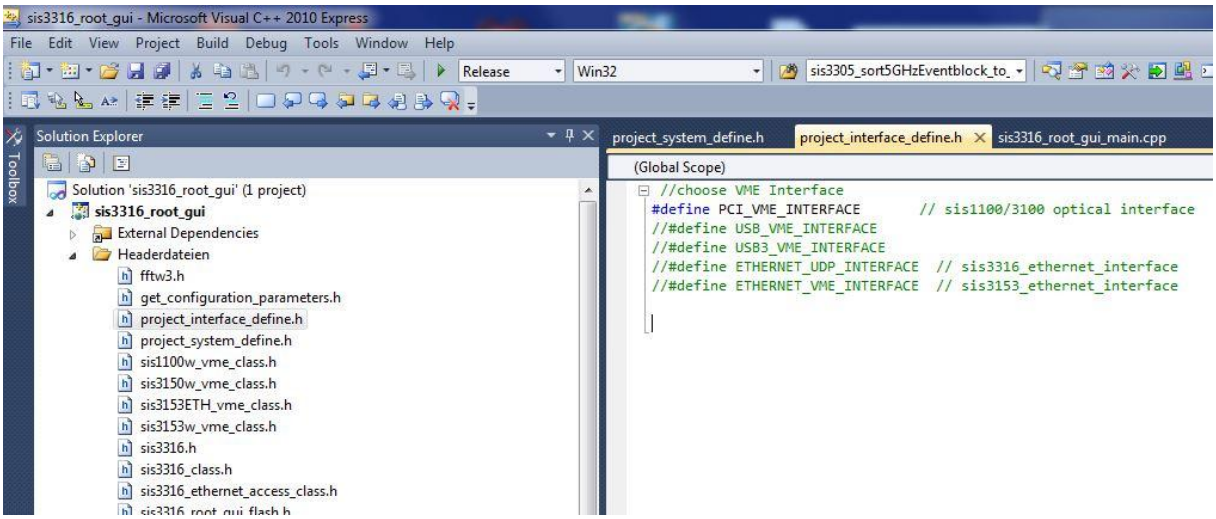


2.1.1 Open the project “sis3316_root_gui.sln” with VisualC++ 2010.

This project contains the C++ files and the property-settings to run with one of the following communicate interfaces:

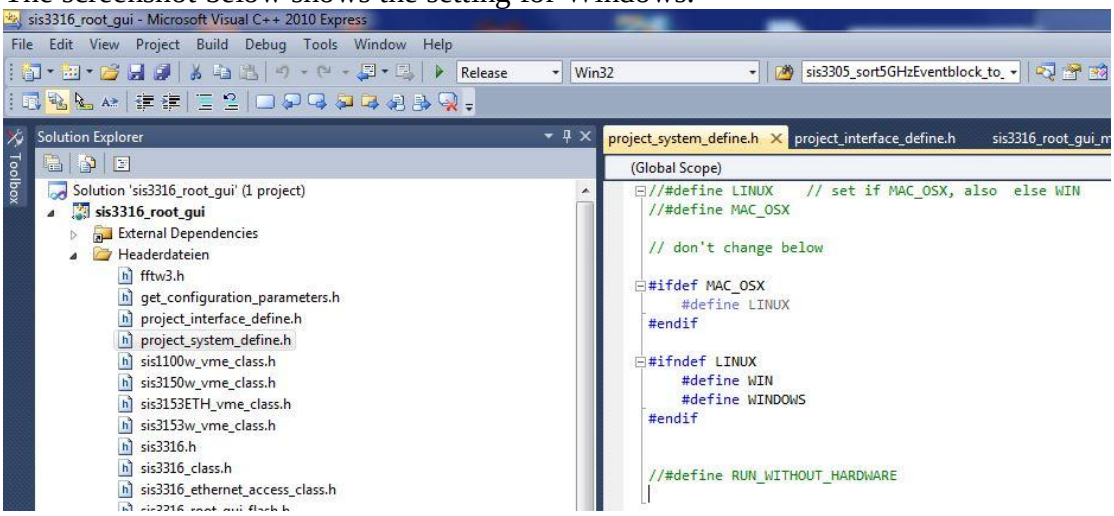
- sis3316-DT Ethernet/UDP interface (sis3316_ethernet_access_class.cpp /.h)
- sis310x optical VME interface (sis1100w_vme_class.cpp /.h)
- sis3150 USB VME interface (sis3150w_vme_class.cpp /.h)
- sis3153 USB3 VME interface (sis3153w_vme_class.cpp /.h)
- sis3153 Ethernet/UDP VME interface (sis3153ETH_vme_class.cpp /.h)

The interface is selected with a “#define” in the file “project_interface_define.h”. The screenshot below shows the selection of the SIS1100/SIS310x optical PCIe to VME Interface.



The unused interface files have to be removed from the project in case that the drivers of these interfaces are not installed on the computer (except ETHERNET_UDP_INTERFACE).

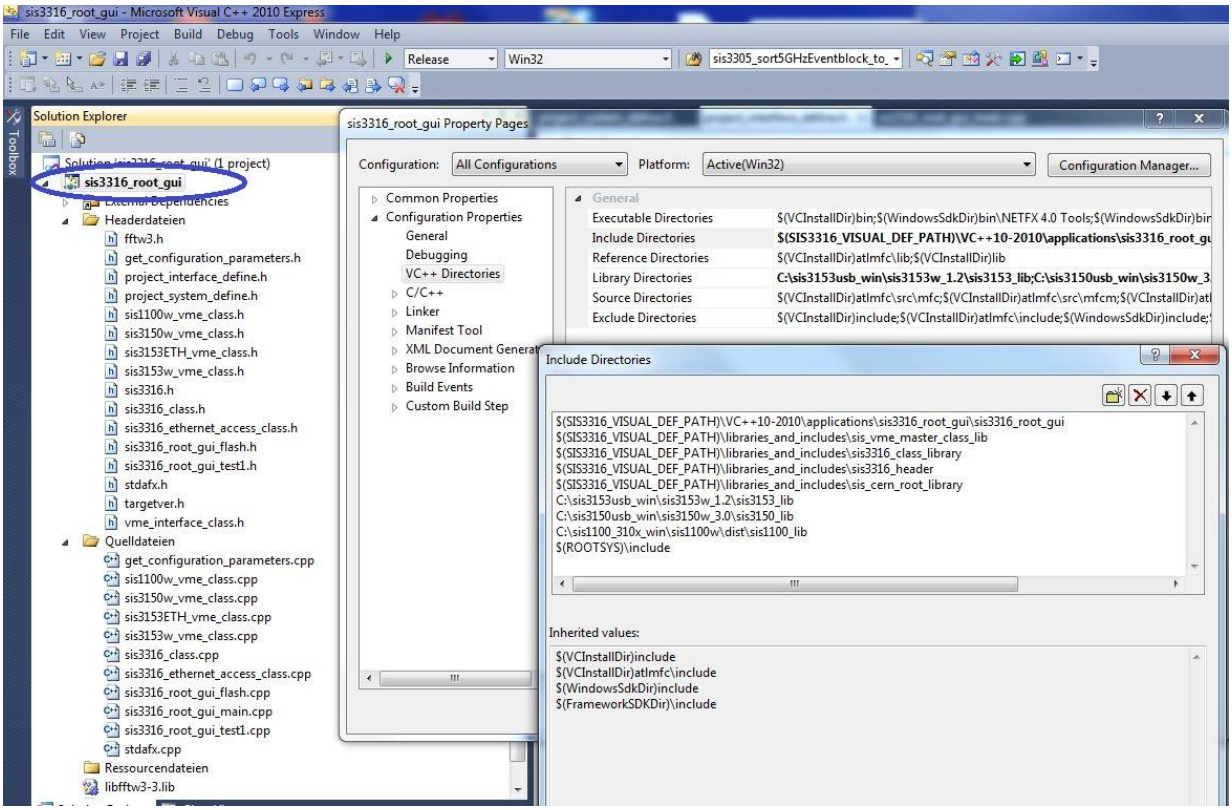
The OS (operating system) is with the file “project_system_define.h”.
The screenshot below shows the setting for Windows.



2.1.2 VC10 Project Properties

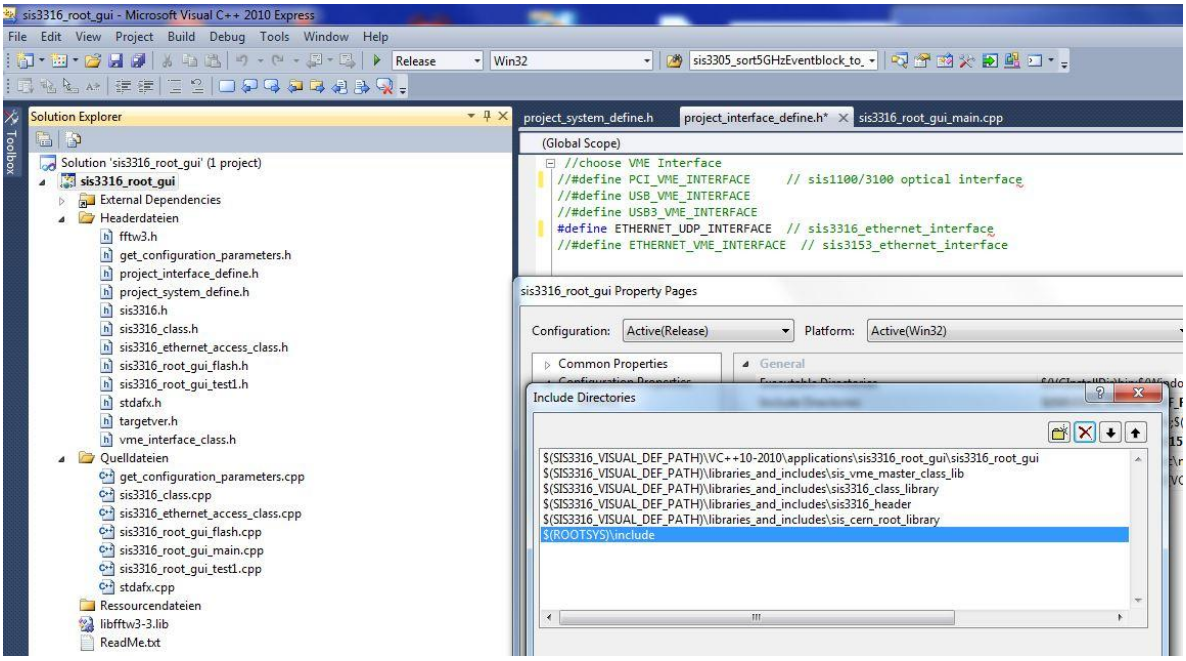
2.1.2.1 Include path settings

Select the project “sis3316_root_gui” (marked in blue) and make a right click. A window pop up and click on Properties (Eigenschaften). The property-window pops up.



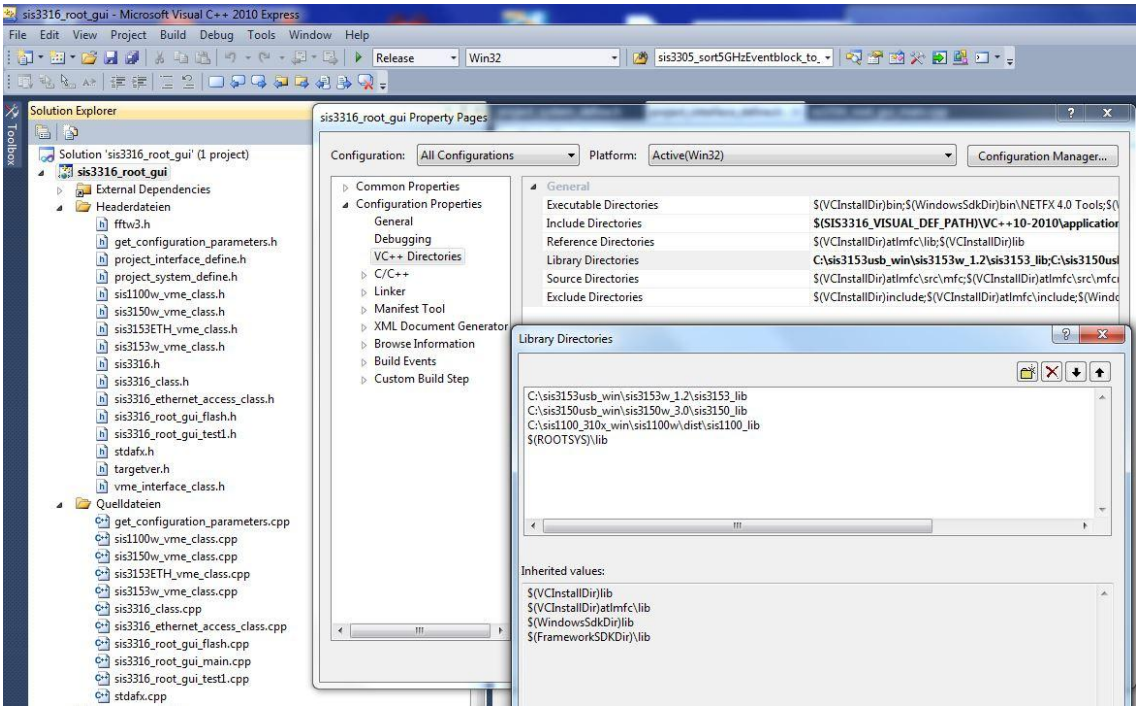
The above project contains the Communication interface files and the project settings of the valid Include path settings for four interfaces. The communication interface files and their settings must be deleted if the corresponding driver is not installed (except ETHERNET_UDP_INTERFACE).

The bottom project (copied from sis3316-DT directory) contains the Communication interface files and the project settings of the valid Include path setting for the Ethernet/UDP interface, only.

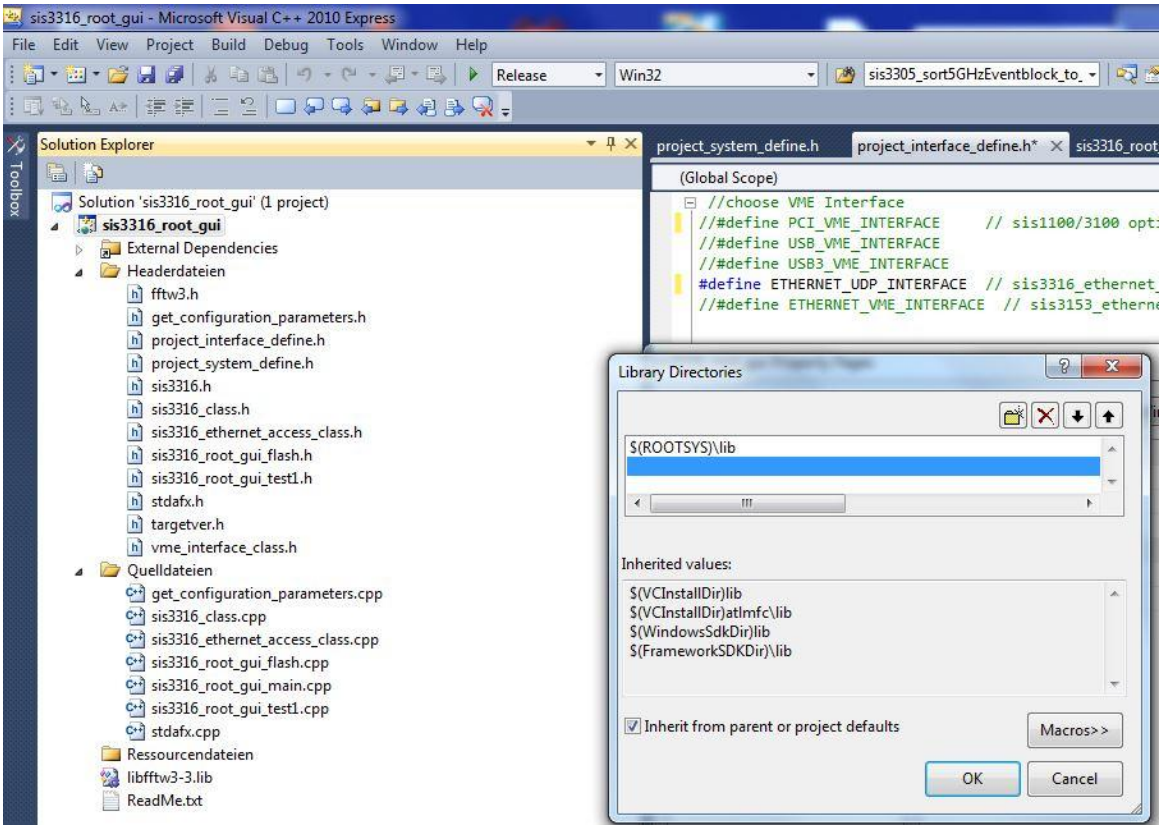


2.1.2.2 Library path settings

This project setting contains the Library paths of all SIS interfaces.

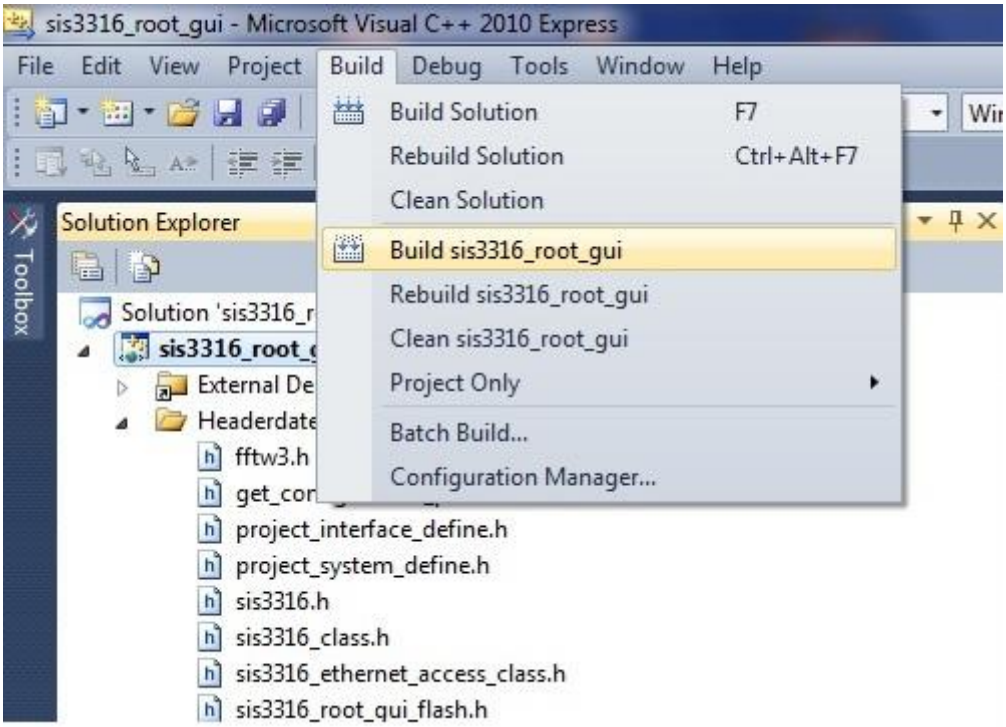


The bottom project (copied from sis3316-DT directory) shows the Library path setting of the customized project for the sis3316-DT Ethernet/UDP interface.



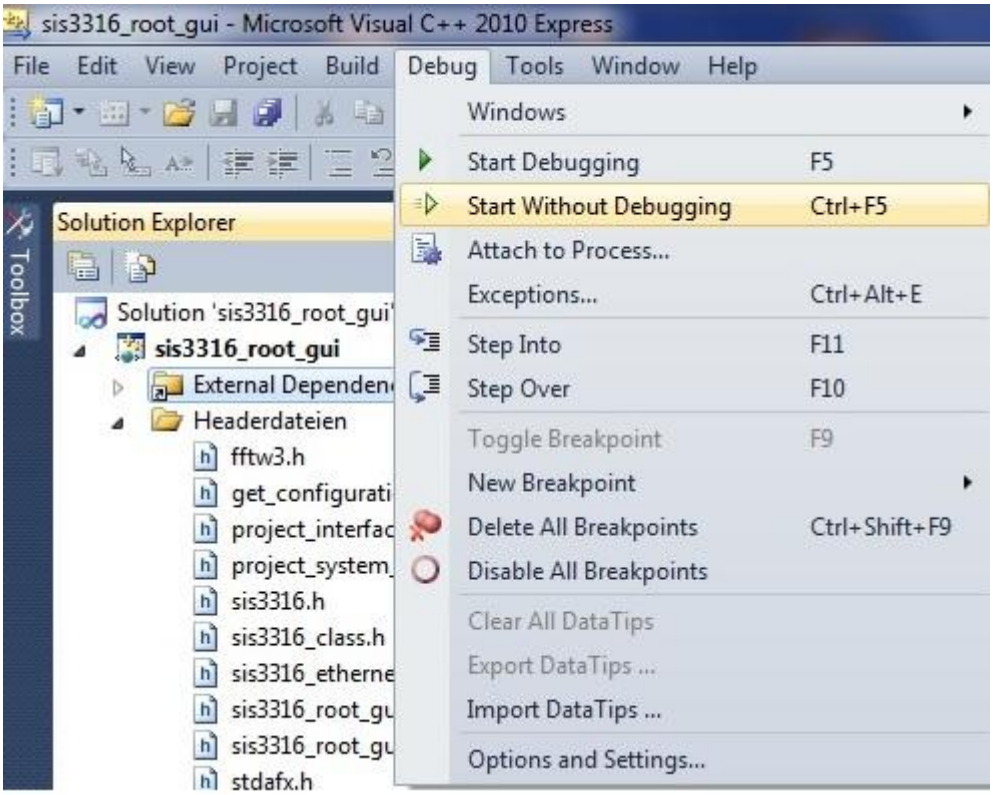
2.1.3 VC10 Build project

Select “Build/Build sis3316_root_gui”.

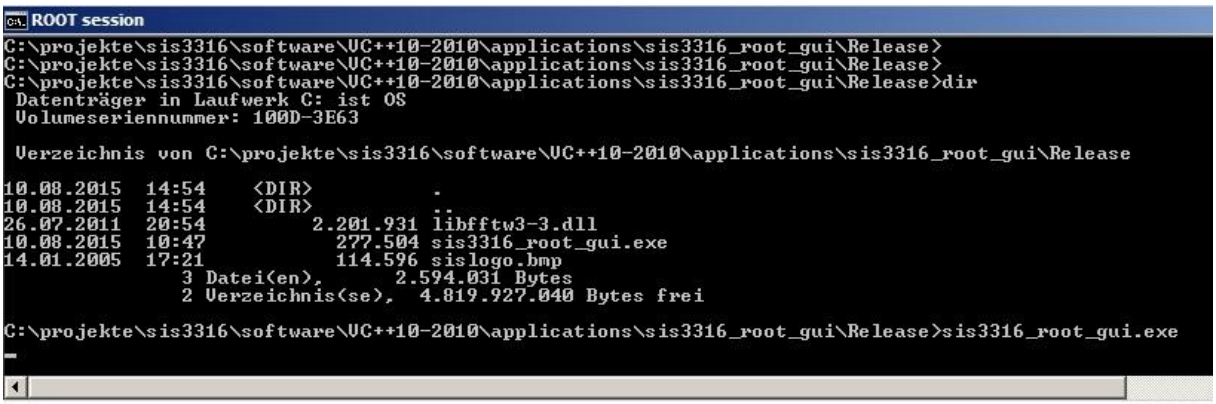


2.1.4 VC10 Run project

Select “Debug/Start Without Debugging” to start the program.



Or start the program from a DOS box.



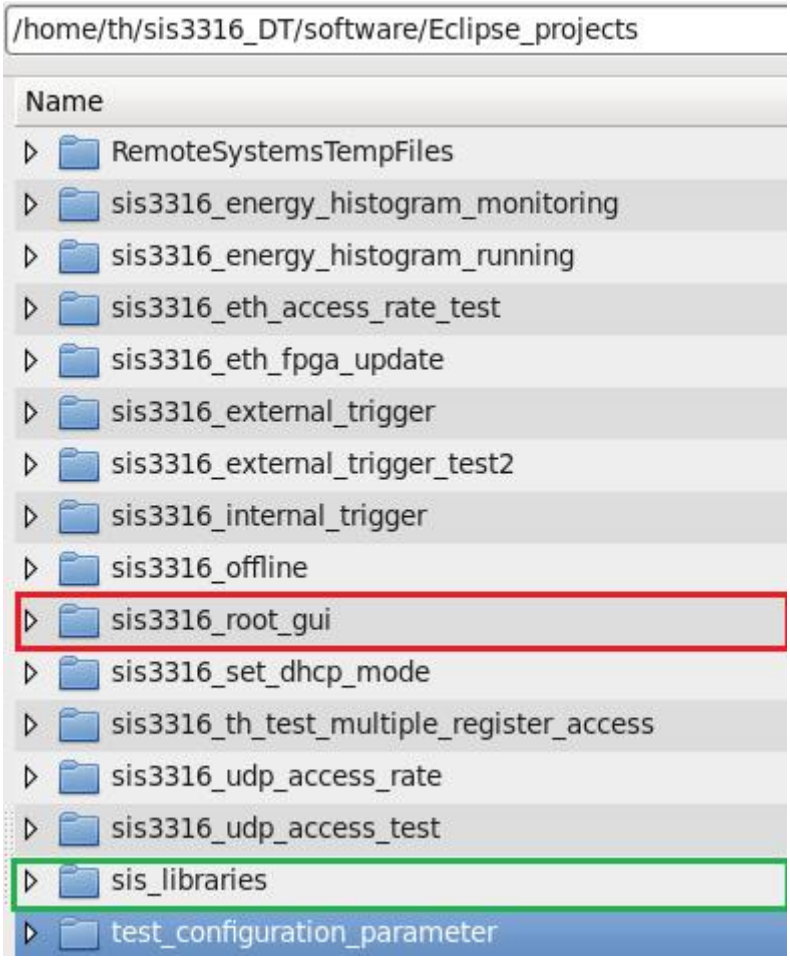
2.2 Linux (Scientific 6.5)

The development tool **Eclipse 3.6.1** is/was used to build the project(s). Each project contains a makefile in the "Release" subfolder. Therefore it is possible to build the project from a terminal shell, too.

Copy the file "sis3316_DT_xxxx.tar.gz to your local disk and unpack it.

2.2.1 Eclipse Project Directory Structure

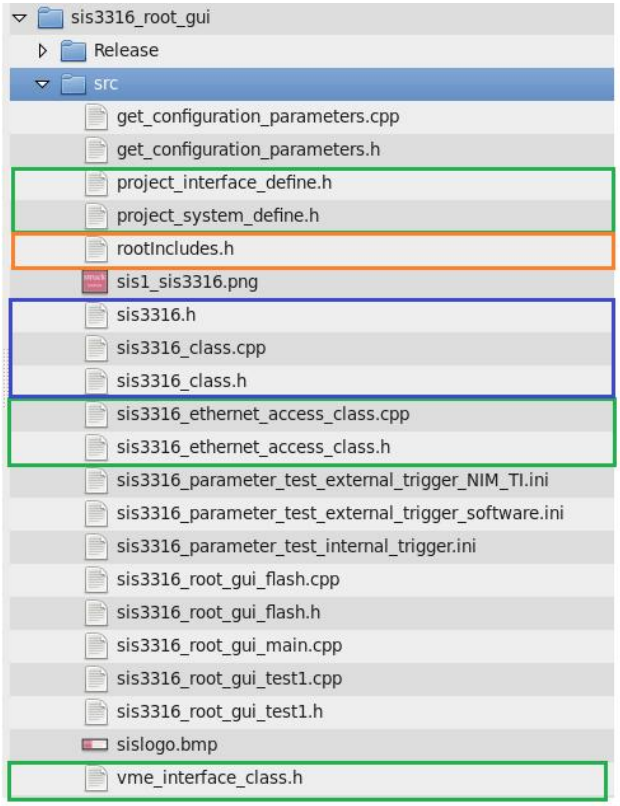
The "../Eclipse_projects" directory contains among other subfolders the two subfolders "sis3316_root_gui" and "sis_libraris". This document addresses these two subfolders, only.

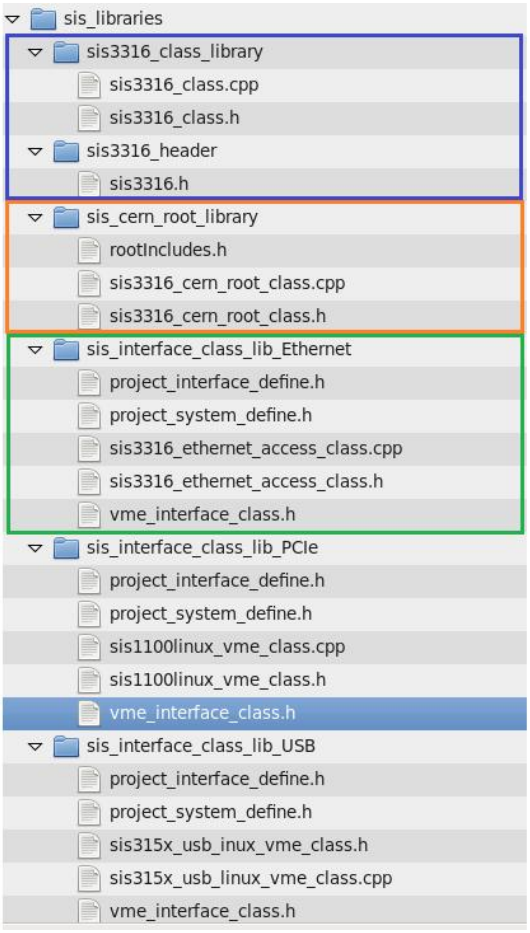


The “sis3316_root_gui/src” folder contains all source files (*.cpp, *.h) which are necessary to build an executable (make) of the project. The marked files (blue, orange and green) are copied from the subfolders “sis_libraries/sis3316_class_library”, “sis_libraries/sis_cern_root_library” and “sis_libraries/sis_interface_class_lib_Ethernet”.
This project uses the Communication Interface “Ethernet”.

To build an executable which uses the SIS1100/SIS310x optical Interface you have to exchange the green marked files (copied from the subfolder “sis_libraries/sis_interface_class_lib_Ethernet”.) with the corresponding files of the subfolder “sis_libraries/sis_interface_class_lib_PCIe”.

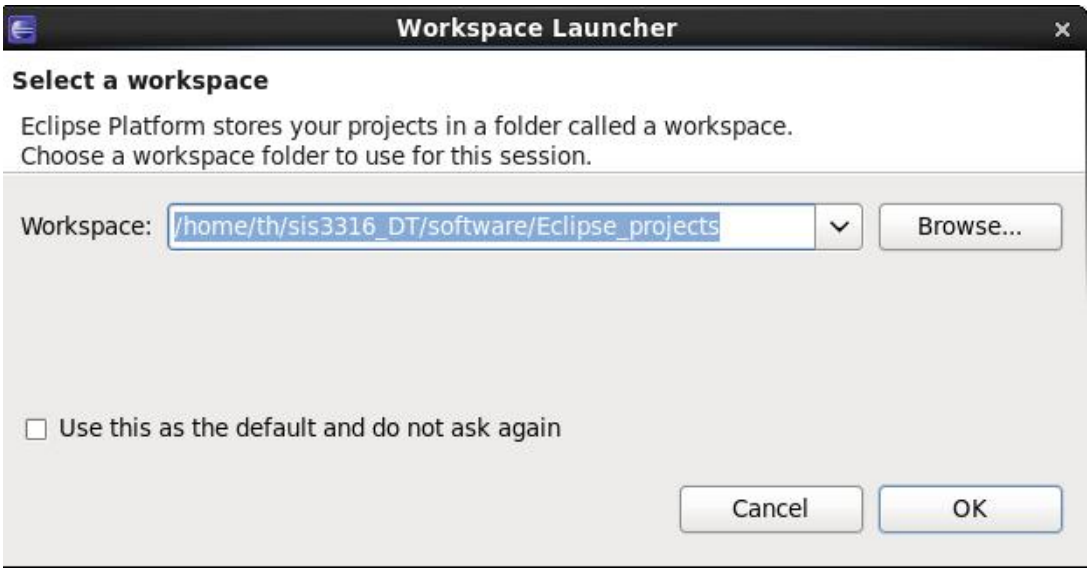
To build an executable which uses the SIS3150 or SIS3153 USB2/USB3 Interface you have to exchange the green marked files (copied from the subfolder “sis_libraries/sis_interface_class_lib_Ethernet”.) with the corresponding files of the subfolder “sis_libraries/sis_interface_class_lib_USB”.



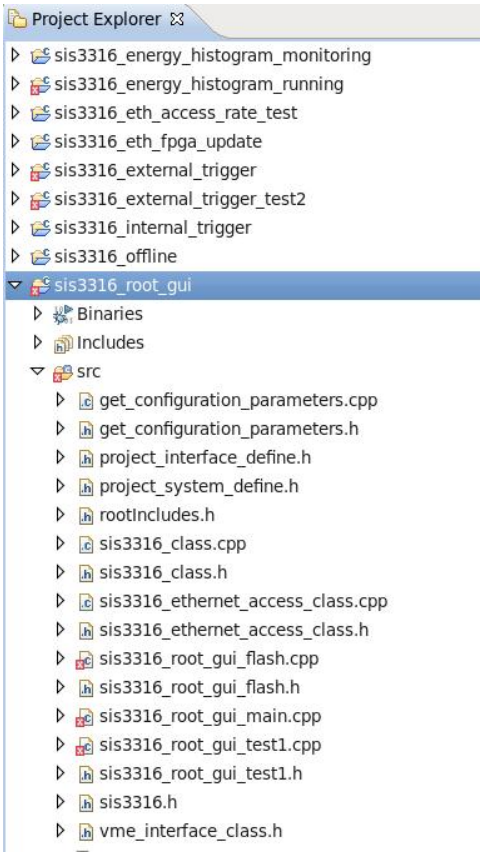


2.2.2 Build with Eclipse

Open the workspace with **Eclipse**.



Make a right click on the project “sis3316_root_gui” (blue marked) and click on “Build Project”.



2.2.3 Build with a terminal shell

Type “make clean” and then type “make sis3316_root_gui”.

```
[th@linux-th Eclipse_projects]$ ls
RemoteSystemsTempFiles               sis3316_eth_fpga_update         sis3316_offline                          sis3316_udp_access_rate
sis3316_energy_histogram_monitoring  sis3316_external_trigger        sis3316_root_gui                         sis3316_udp_access_test
sis3316_energy_histogram_running     sis3316_external_trigger_test2  sis3316_set_dhcp_mode                    sis_libraries
sis3316_eth_access_rate_test         sis3316_internal_trigger        sis3316_th_test_multiple_register_access  test_configuration_parameter
[th@linux-th Eclipse_projects]$ cd sis3316_root_gui/
[th@linux-th sis3316_root_gui]$ cd Release/
[th@linux-th Release]$ make clean
rm -rf  ./src/get_configuration_parameters.o ./src/sis3316_class.o ./src/sis3316_ethernet_access_class.o ./src/sis3316_root_gui_flash.o ./sr
ui_test1.o  ./src/get_configuration_parameters.d ./src/sis3316_class.d ./src/sis3316_ethernet_access_class.d ./src/sis3316_root_gui_flash.d
ot_gui_test1.d  sis3316_root_gui

[th@linux-th Release]$ make sis3316_root_gui
Building file: ../src/get_configuration_parameters.cpp
Invoking: GCC C++ Compiler
g++ -I/home/th/root/include -O3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src/get_configuration_parameters.d" -MT"src/get_configuration_param
" "../src/get_configuration_parameters.cpp"
Finished building: ../src/get_configuration_parameters.cpp

.......


Building target: sis3316_root_gui
Invoking: GCC C++ Linker
g++ -L/home/th/root/lib -L/usr/local/lib -o"sis3316_root_gui"  ./src/get_configuration_parameters.o ./src/sis3316_class.o ./src/sis3316_
sh.o ./src/sis3316_root_gui_main.o ./src/sis3316_root_gui_test1.o   -lfftw3 -lCint -lGraf -lGui
Finished building target: sis3316_root_gui

[th@linux-th Release]$
```

2.2.4 Run from a terminal shell

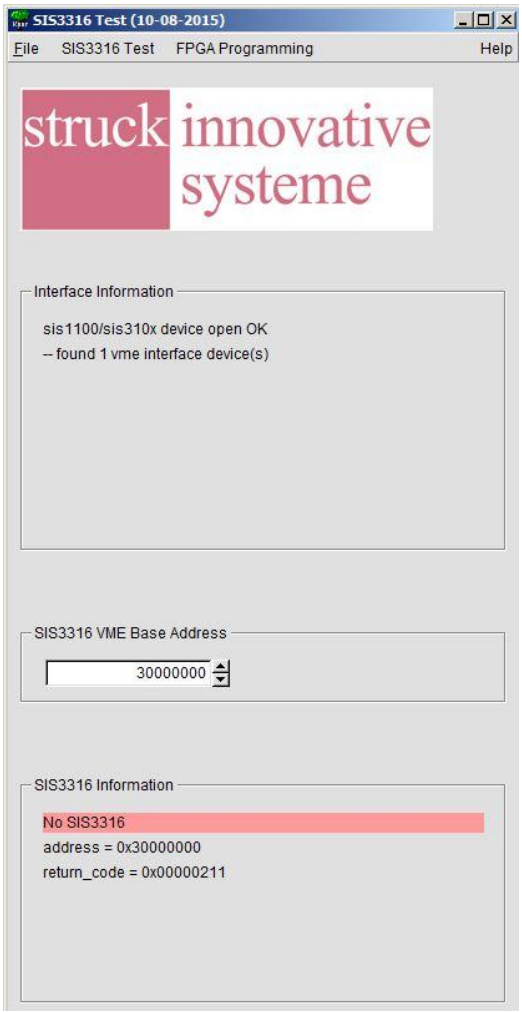
Type “./sis3316_root_gui”.

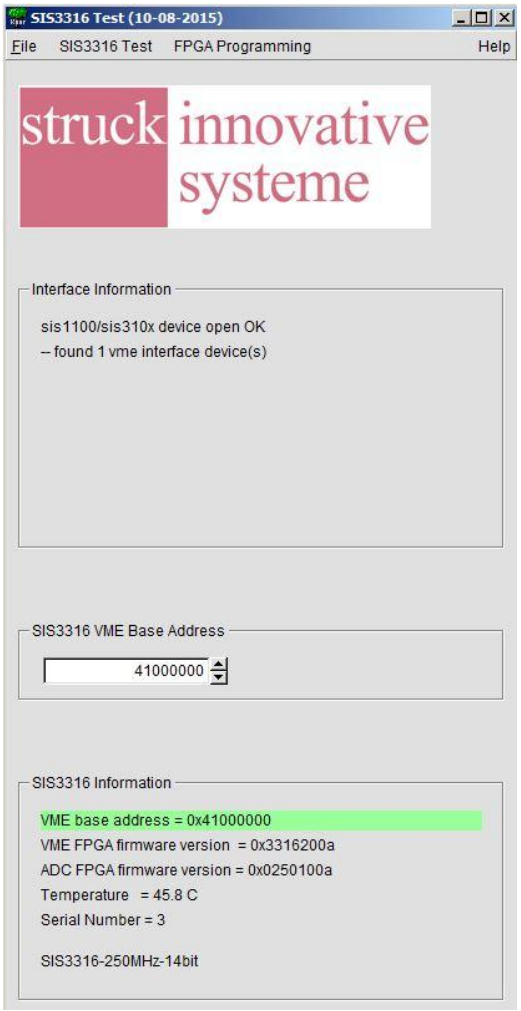
```
[th@linux-th Release]$
[th@linux-th Release]$
[th@linux-th Release]$ ./sis3316_root_gui
```

3 Run the program “sis3316_root_gui”

Start the program “sis3316_root_gui” and the “main control panel” will appear (left screenshot).
This example shows the “main control panel” of the sis3316_root_gui which uses the SIS1100/SIS310x VME Interface.

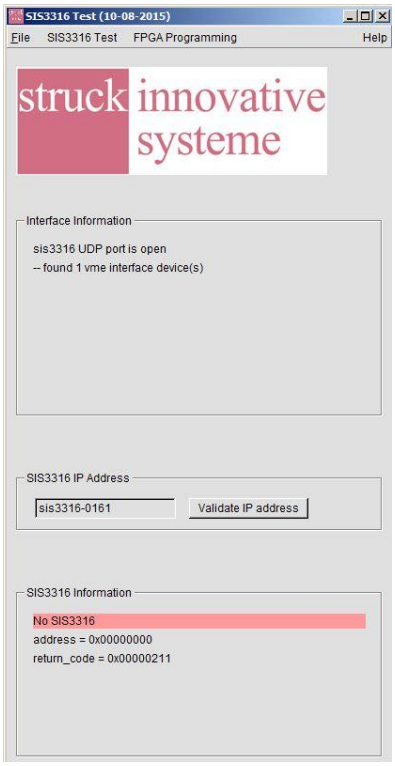
Type the SIS3316 VME base address into the “SIS3316 VME Base Address” box and you will get the “SIS3316 Information” (right screenshot) if the address is valid.

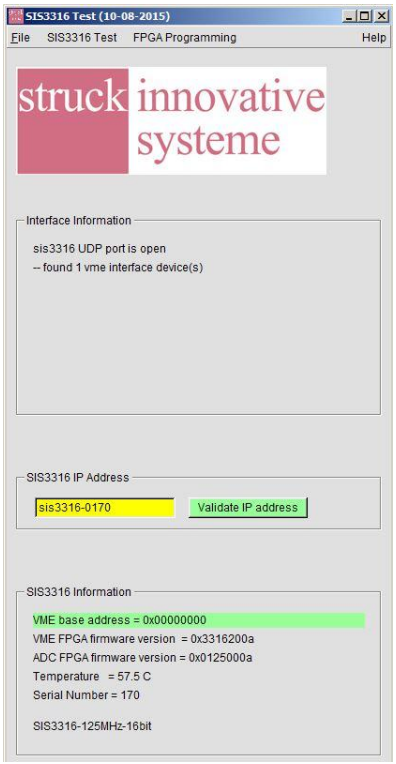


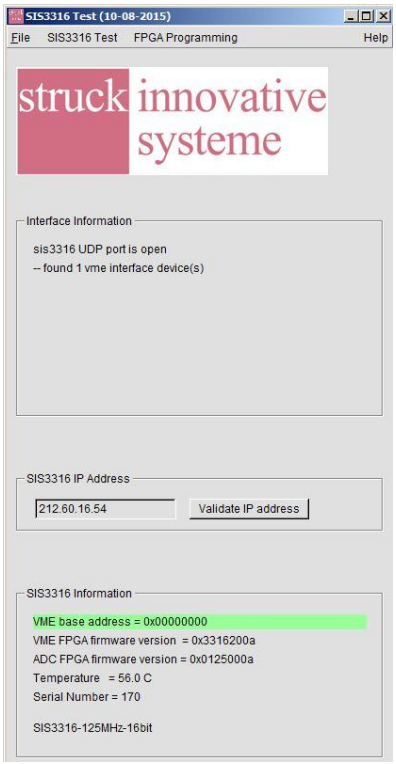


Start the program “sis3316_root_gui” and the “main control panel” will appear (left screenshot).
This example shows the “main control panel” of the sis3316_root_gui which uses the sis3316 Ethernet Interface.

Type the SIS3316 IP address into the “SIS3316 IP address” box and click the button “Validate IP address”. If the address is valid you will get the “SIS3316 Information” (right screenshot). Instead of the IP address it also possible to use the Hostname, provided that the DHCP feature is enabled and a Name-Server is active (middle screenshot).







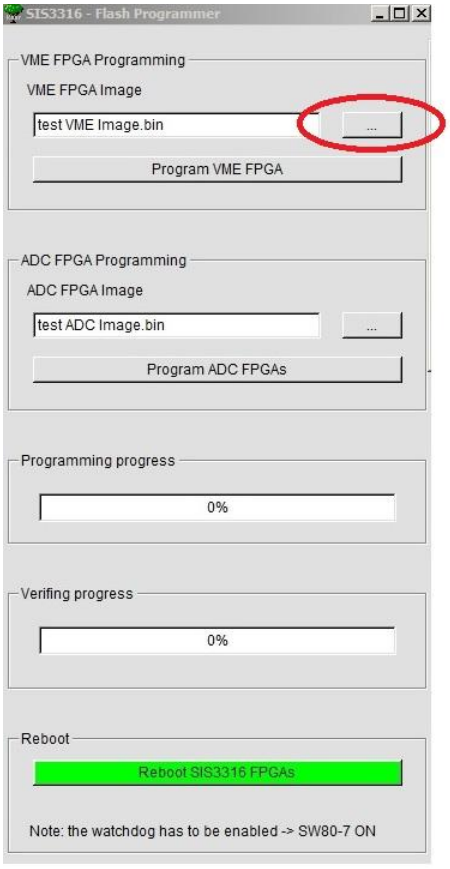
3.1 FPGA Programming Menu

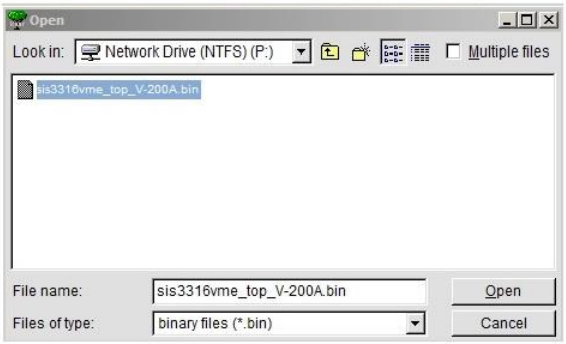
Select “FPGA Programming” in the menu bar and select “FPGA Programming Menu”.

The “FPGA Programming Menu” canvas will appear.

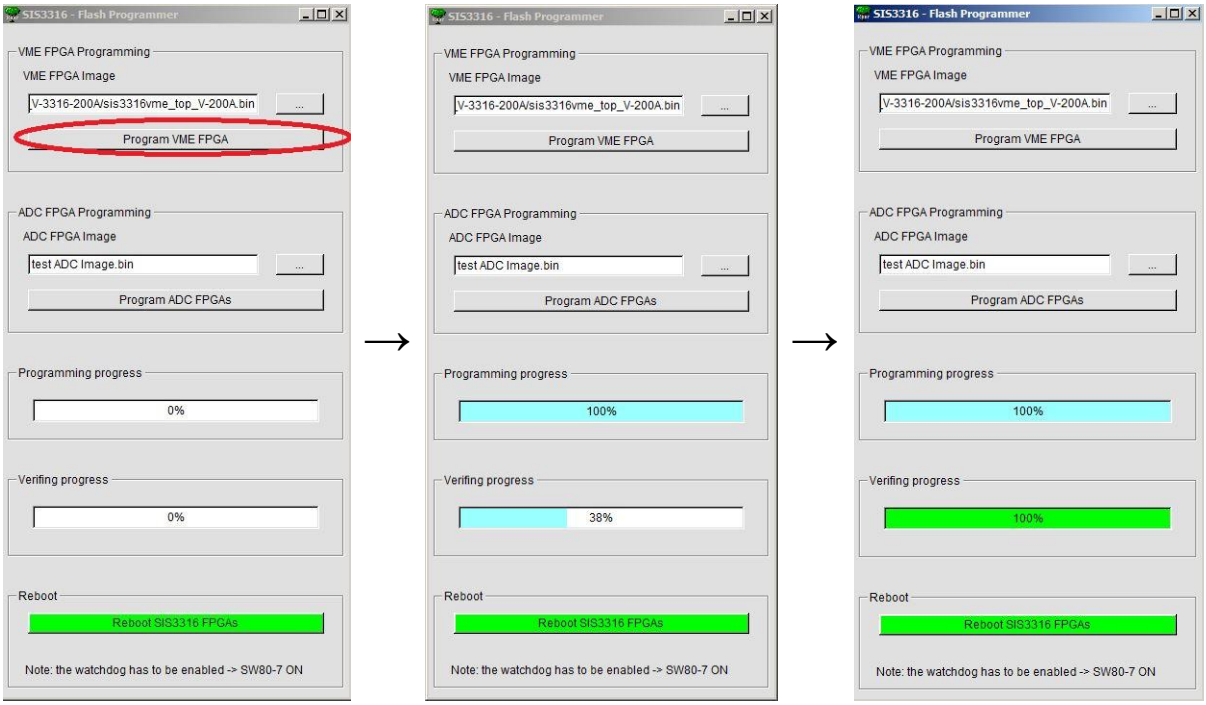
3.1.1 VME FPGA

Select “...” button (marked in red) to get a browser window to search for and select the VME FPGA image file (*.bin).
Note: be sure to load a VME FPGA image file!





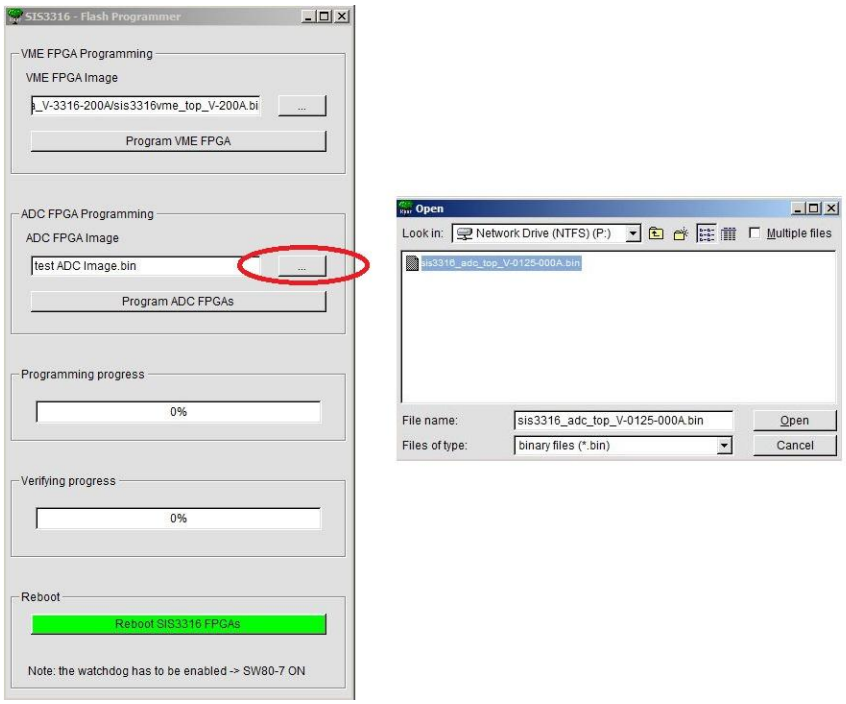
Click on “Program VME FPGA” button (marked in red).
Programming and verification will take a couple of minutes.



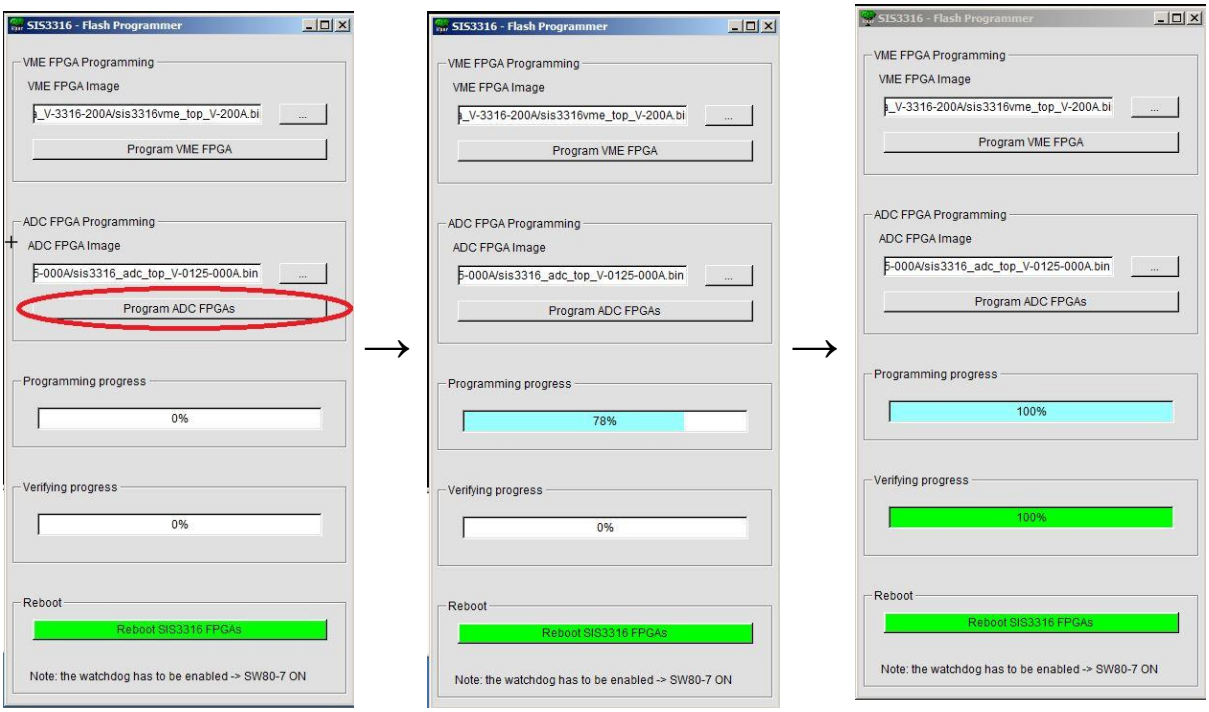
The 100% and the green color of the Verifying progress bar indicates that the programming of the VME FPGA image into the Flashprom has completed successfully.

3.1.2 ADC FPGA

Select “...” button (marked in red) to get a browser window to search for and select the ADC FPGA image file (*.bin).



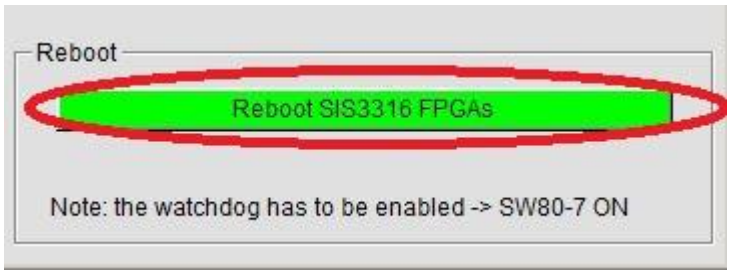
Click on “Program ADC FPGA” button (marked in red).
Programming and verification will take a couple of minutes.



The 100% and the green color of the Verifying progress bar indicates that the programming of the VME ADC Image into the Flashprom has completed successfully.

3.1.3 FPGA reboot

The FPGAs have to be rebooted after the programming of the image files into the Flashproms was done properly. The reboot can be done either by a power down / power up cycle of the module or Click on “Reboot SIS3316 FPGAs” button (marked in red).

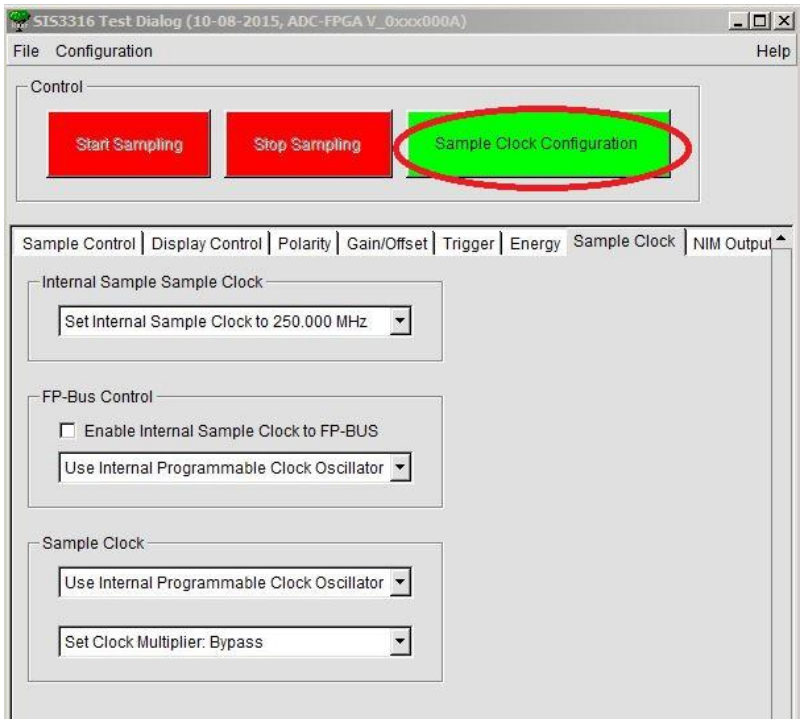


Note: Refer to the JTAG firmware installation procedure as described in the SIS3316 user manual in case of corrupted/erased Flashprom contents.

3.2 SIS3316 Test Menu

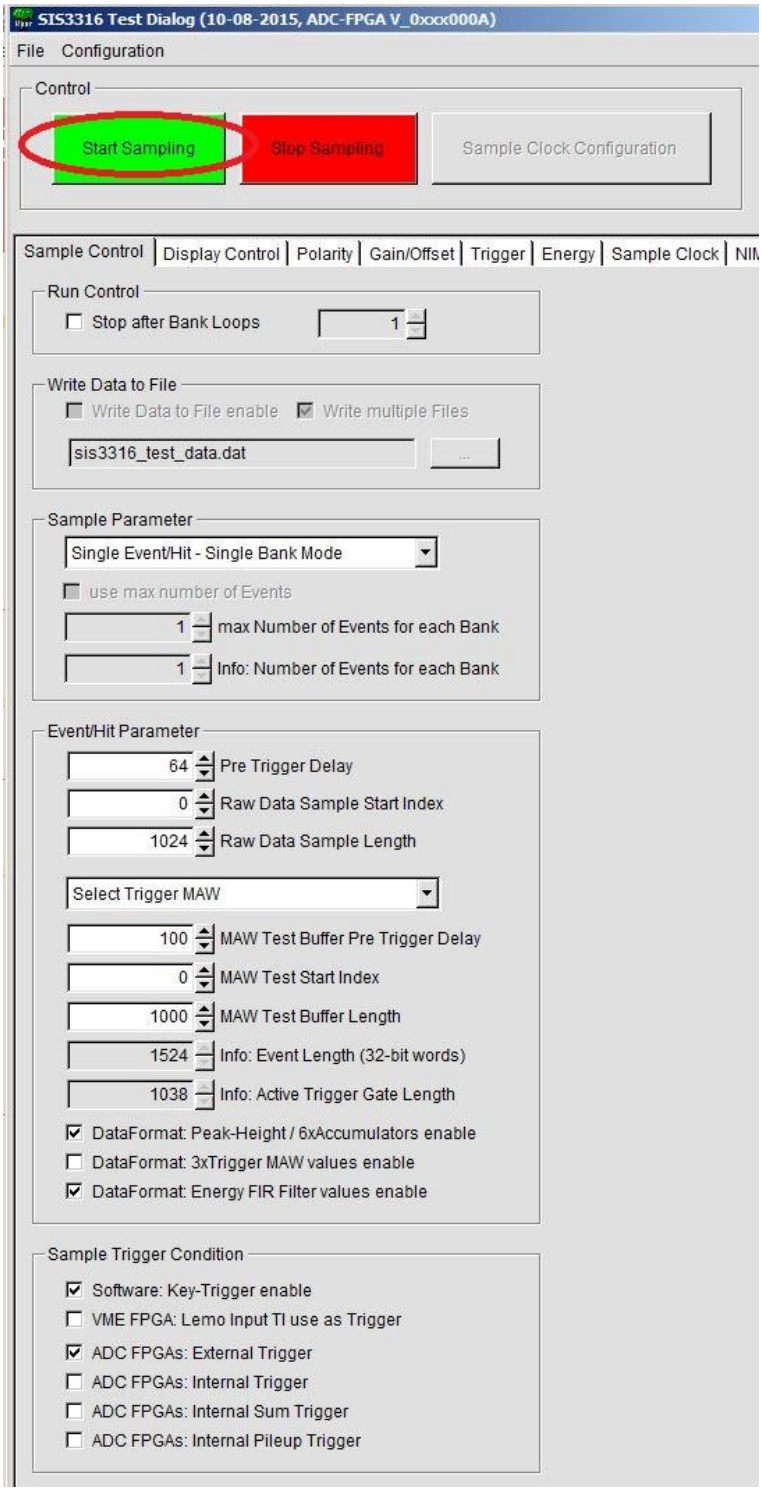
Select “SIS3316 Test” in the menu bar and select “Test1”.

The “SIS3316 Test Dialog” canvas will appear with “Sample Clock” tab frame.

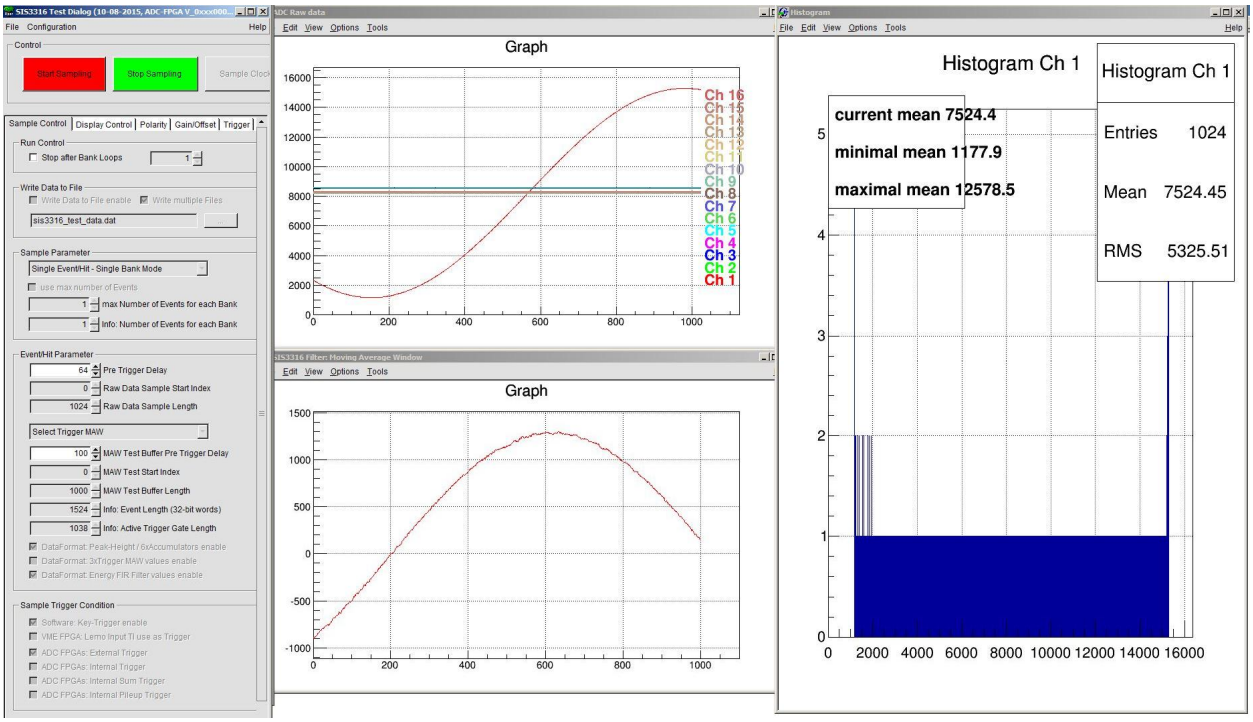


Define the required sample clock and click on “Sample Clock Configuration” button (marked in red). The SIS3316 will be reset and the sample clock will be configured.

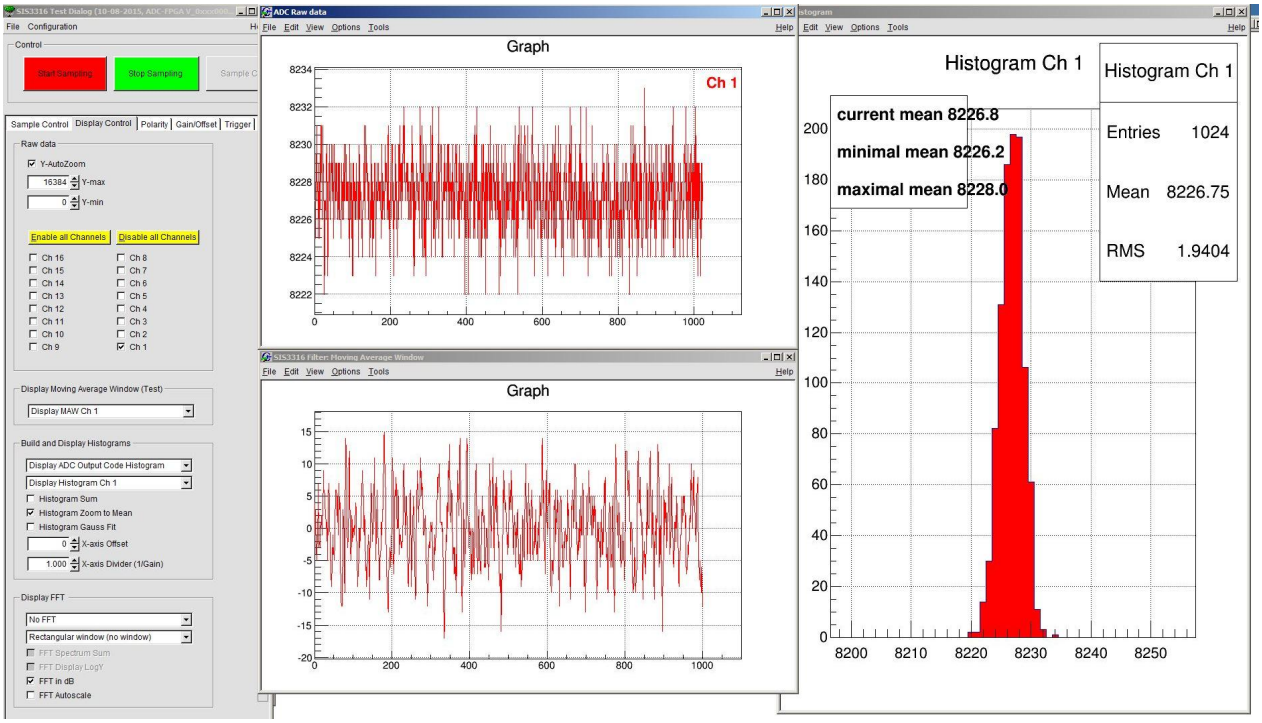
The “SIS3316 Test Dialog” canvas will appear with “Sample Control” tab frame.
Click on “Start” button (marked in red). The SIS3316 will start sampling and displaying the signals of all channels in the “ADC Raw data” canvas, the Trigger MAW trapezoidal of channel 1 in the “SIS3316 Filter: Moving Average Window” canvas and the ADC output code of channel 1 in the “Histogram” canvas.



A sine signal is connected to channel 1.



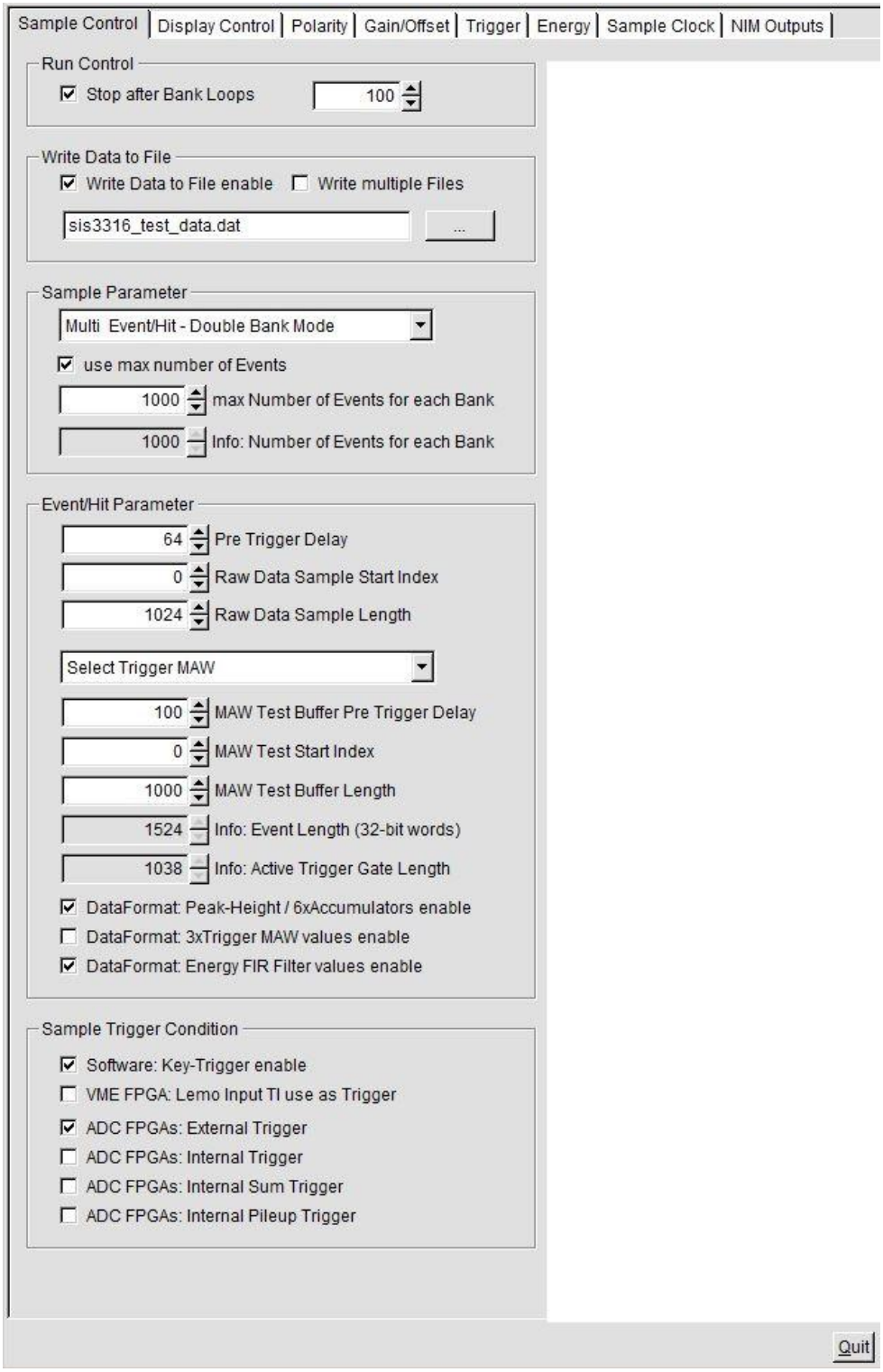
No signal is connected (open) to channel 1.



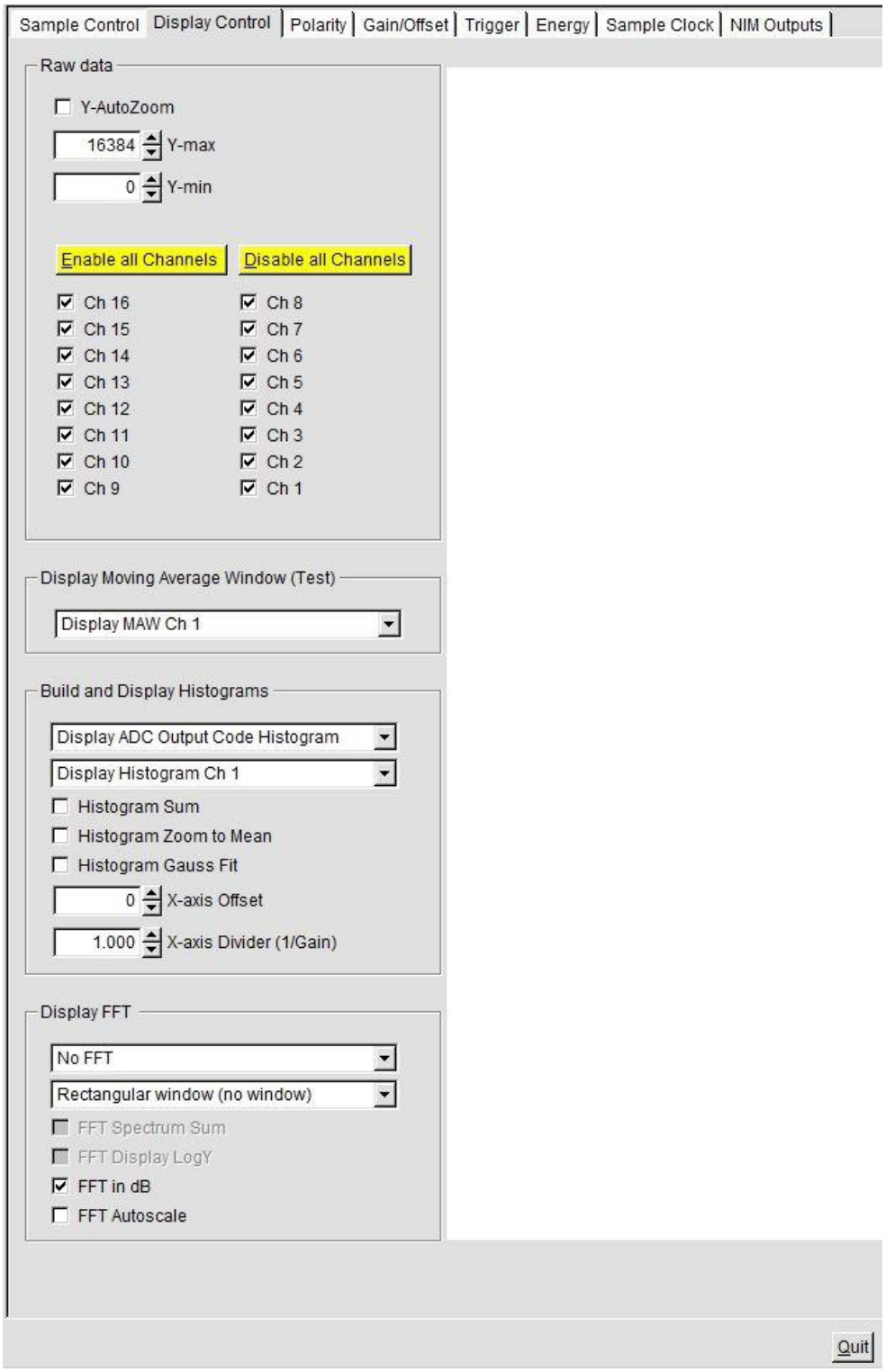
Click on “Stop” button to stop sampling.

3.2.1 Tab Frames

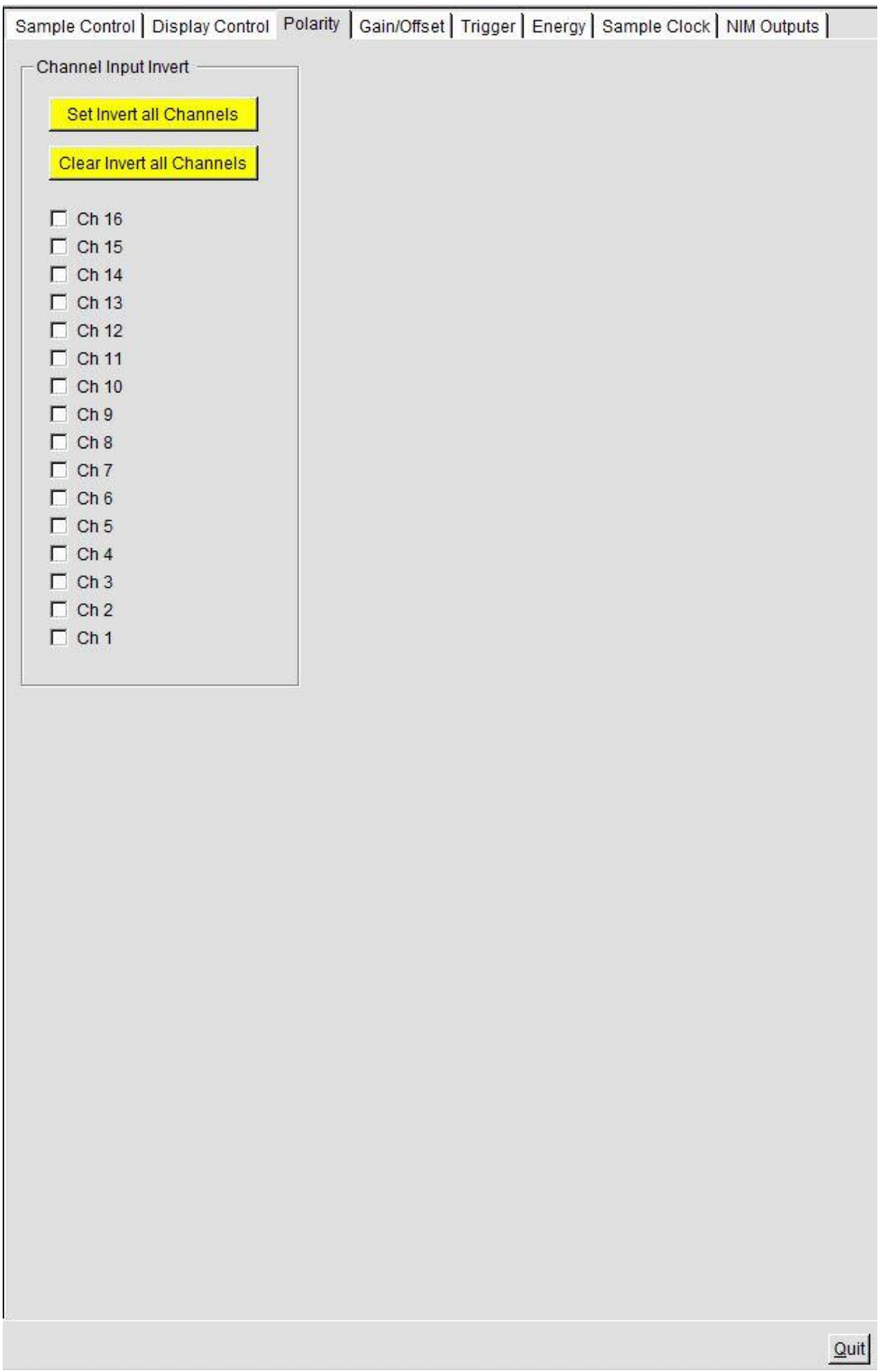
3.2.1.1 Sample Control



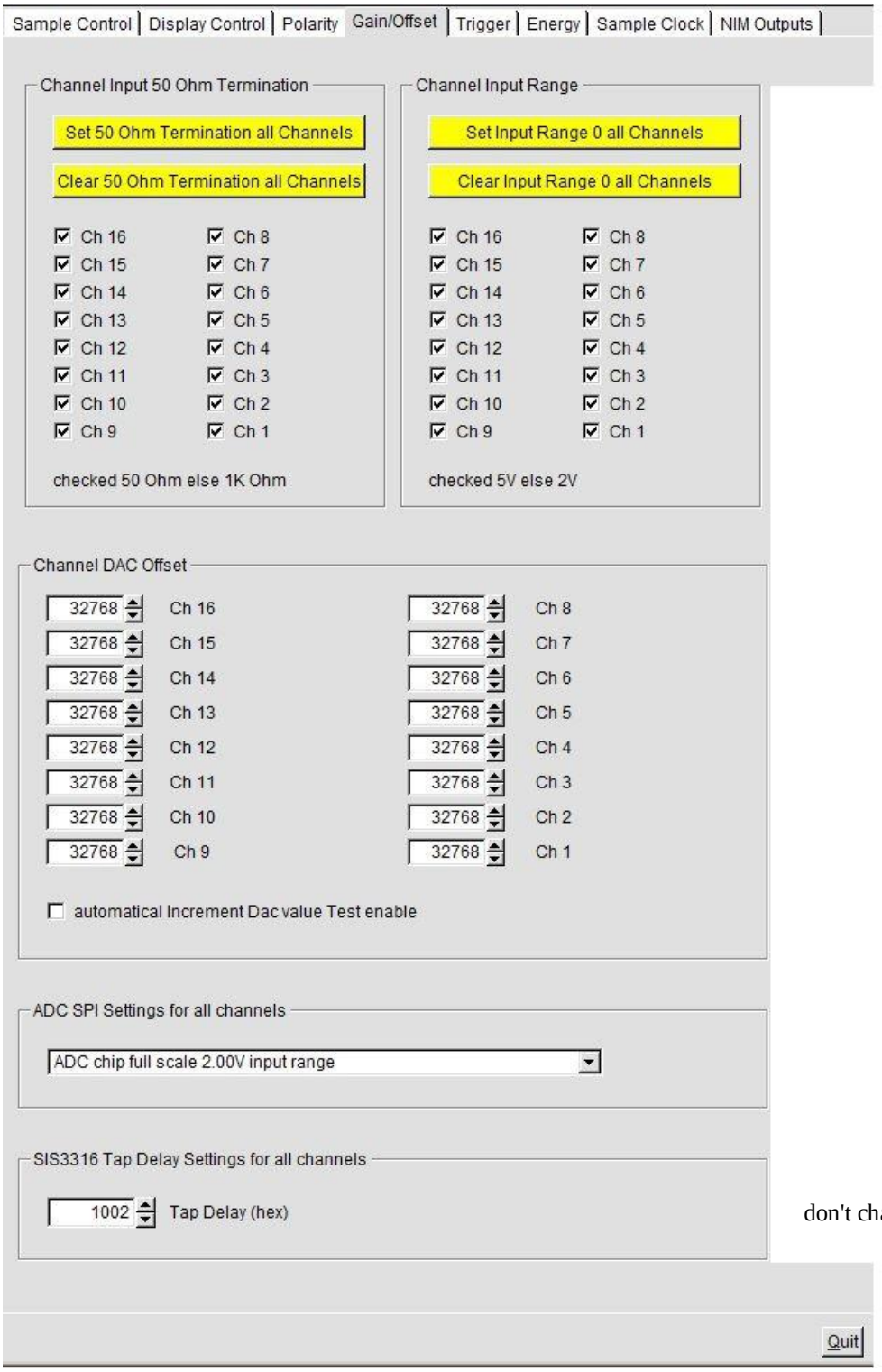
3.2.1.2 Display Control



3.2.1.3 Polarity

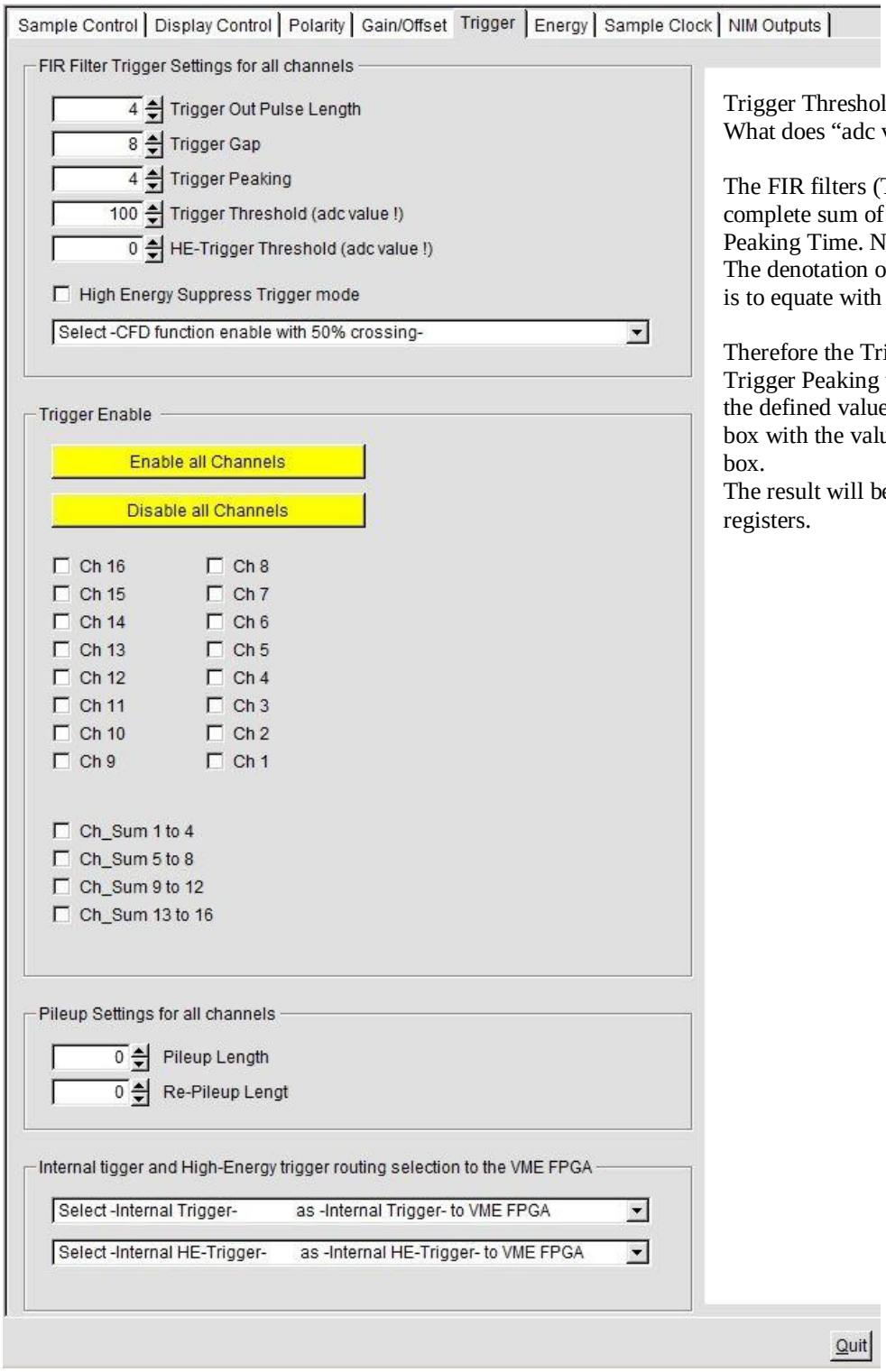


3.2.1.4 Gain/Offset



don't change it (test feature)

3.2.1.5 Trigger



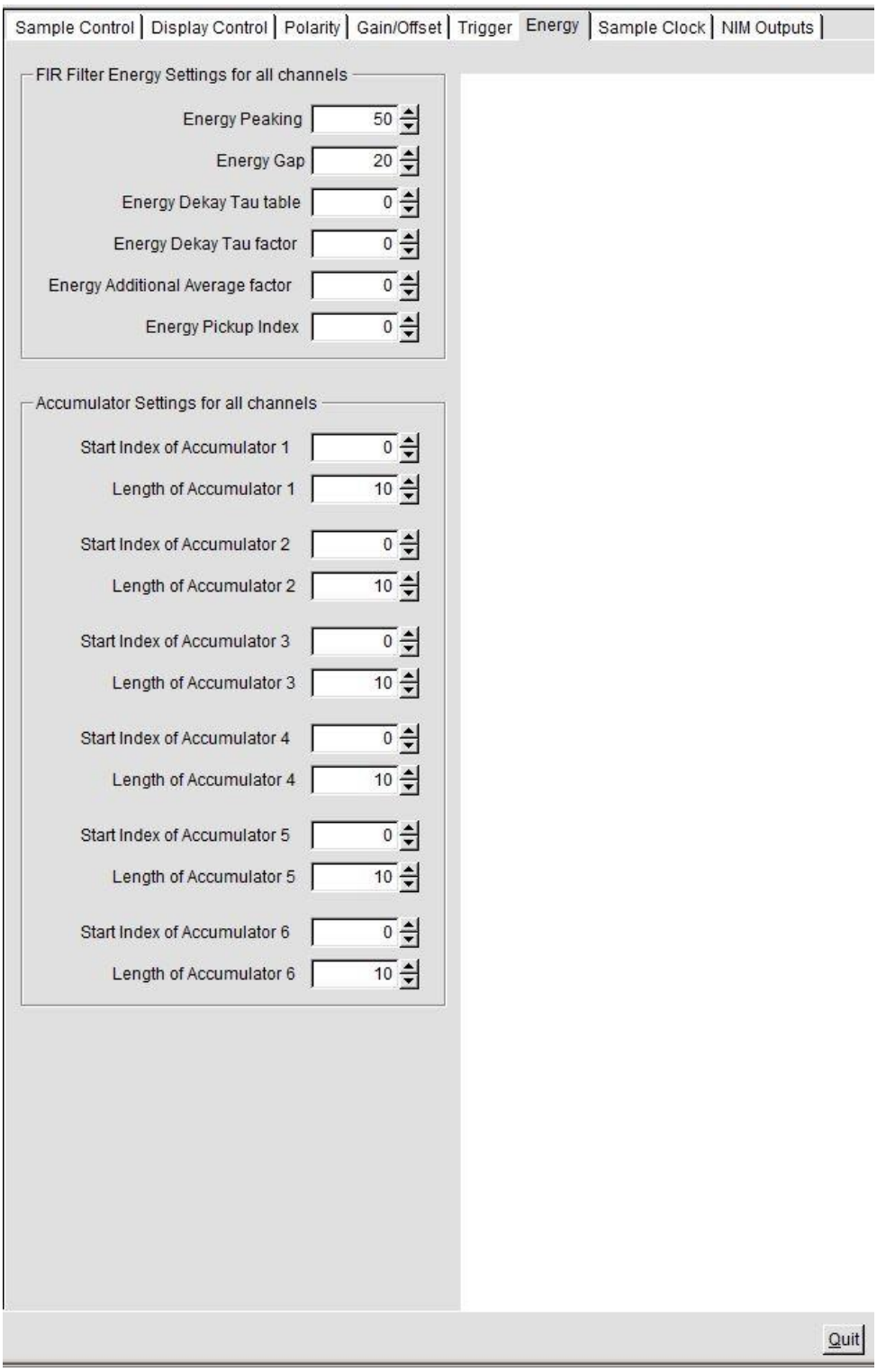
Trigger Threshold (adc_value !):
What does “adc values!” mean in this context ?

The FIR filters (Trigger and Energy) are building the complete sum of the adc data words over the Peaking Time. No “averaging” are performed!
The denotation of **MAW** is caused by history and it is to equate with **MW** in this document!

Therefore the Trigger Threshold depends on the Trigger Peaking value, also. The software multiplies the defined value in the “Trigger Threshold” entry box with the value of the “Trigger Peaking” entry box.
The result will be written to Trigger Threshold registers.

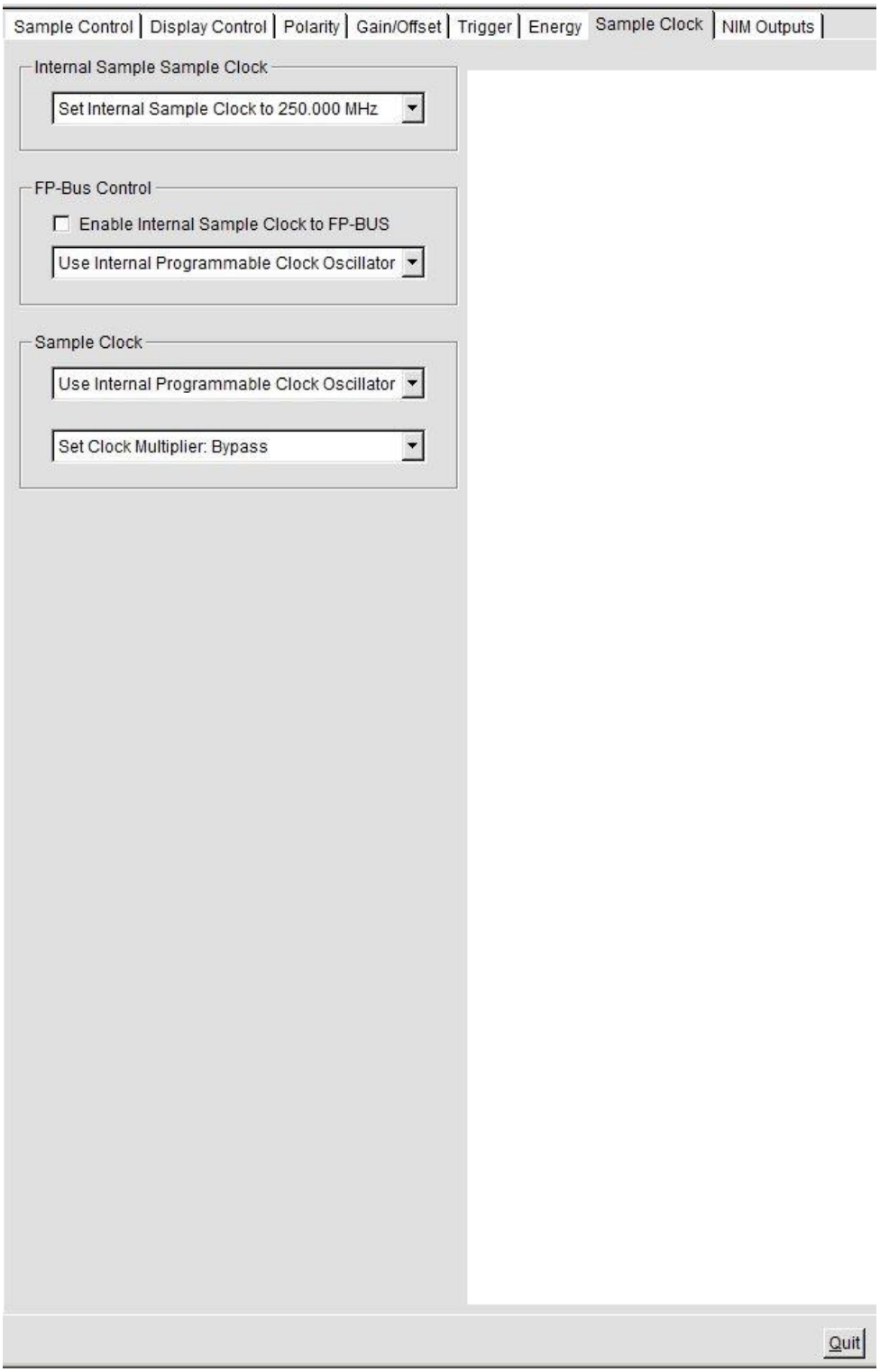
Note: the “sis3316_root_gui” program does not support parameters like “Trigger Threshold” for the individual channel, while the SIS3316 firmware does support individual parameters for each channel!

3.2.1.6 Energy

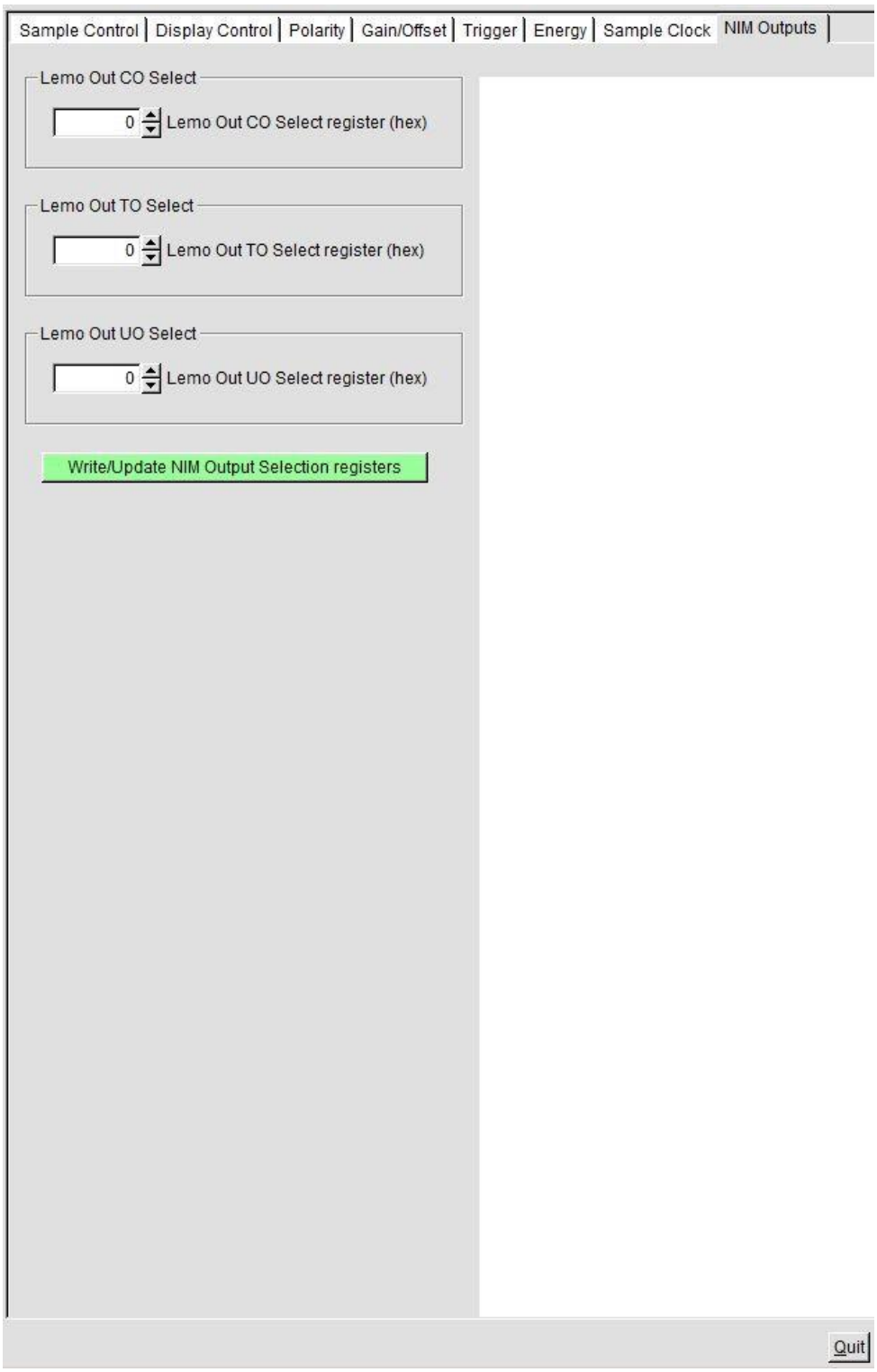


Note: the "sis3316_root_gui" program does not support individual parameters like "Energy Peaking" for the individual channel, while the SIS3316 firmware does support individual parameters for each channel!

3.2.1.7 Sample Clock



3.2.1.8 NIM Outputs



3.2.2 Save/Load Configuration parameters

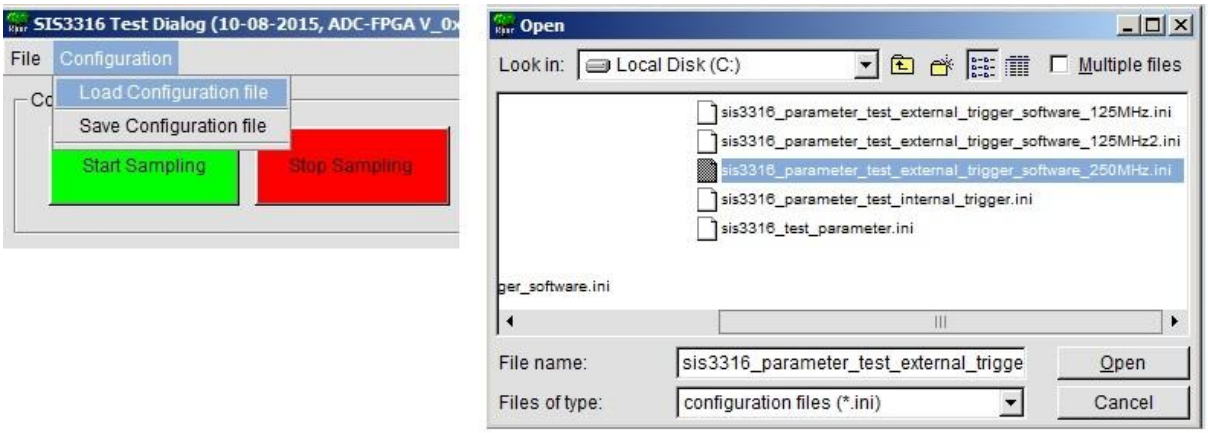
Most of the "sis3316_root_gui" parameters can be saved to a configuration file and can be loaded from a configuration file.
The parameter definition can be found in the header file "`get_configuration_parameters.h`".

For example:

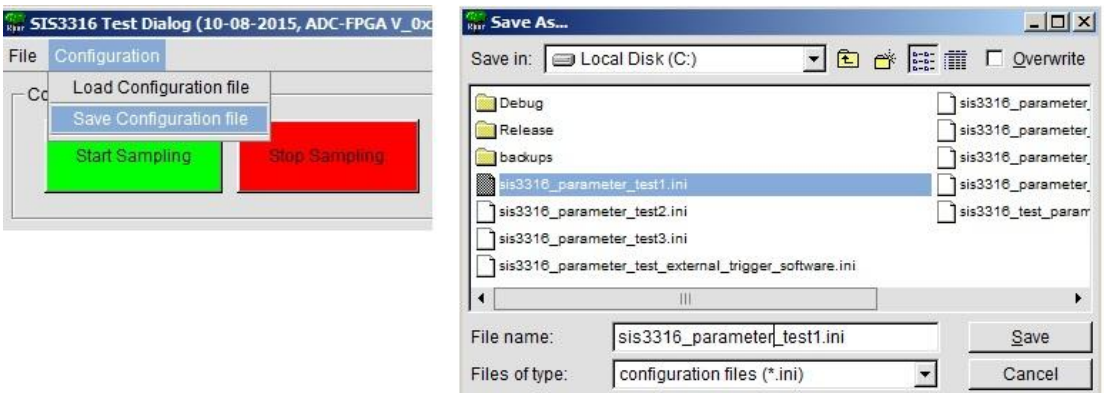
```
#define TEXT_PARAMETER_TRIGGER_COND_SOFTWARE_KEY          "PARAMETER_TRIGGER_COND_SOFTWARE_KEY"
#define TEXT_PARAMETER_TRIGGER_COND_VME_LEMO_TI           "PARAMETER_TRIGGER_COND_VME_LEMO_TI"
#define TEXT_PARAMETER_TRIGGER_COND_ADC_EXTERNAL          "PARAMETER_TRIGGER_COND_ADC_EXTERNAL"
#define TEXT_PARAMETER_TRIGGER_COND_ADC_INTERNAL          "PARAMETER_TRIGGER_COND_ADC_INTERNAL"
#define TEXT_PARAMETER_TRIGGER_COND_ADC_INTERNAL_SUM      "PARAMETER_TRIGGER_COND_ADC_INTERNAL_SUM"
#define TEXT_PARAMETER_TRIGGER_COND_ADC_INTERNAL_PILEUP   "PARAMETER_TRIGGER_COND_ADC_INTERNAL_PILEUP"


#define TEXT_PARAMETER_CHANNEL_POLARITY_INVERT            "PARAMETER_CHANNEL_POLARITY_INVERT"
#define TEXT_PARAMETER_CHANNEL_RANGE_2V                   "PARAMETER_CHANNEL_RANGE_2V"
#define TEXT_PARAMETER_CHANNEL_50OHM_TERMINATION_DISABLE  "PARAMETER_CHANNEL_50OHM_TERMINATION_DISABLE"
#define TEXT_PARAMETER_CHANNEL_ADC_OFFSET                 "PARAMETER_CHANNEL_ADC_OFFSET"
```

Select "Configuration" in the menu bar and select "Load Configuration file" to get a browser window to search for and select a configuration file (*.ini).

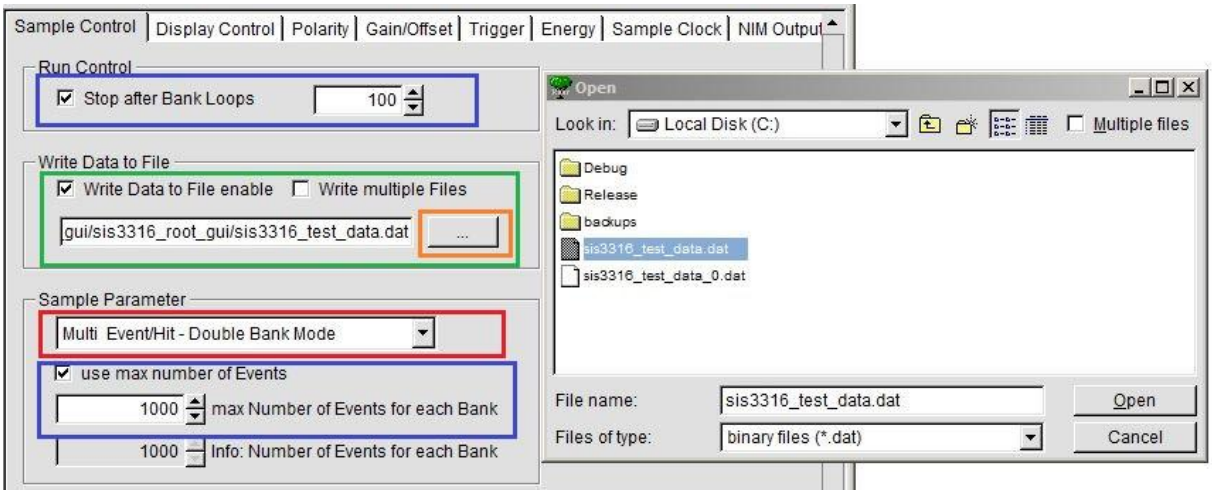


Select "Configuration" in the menu bar and select "Save Configuration file" to get a browser window to search for and select or define a configuration file name (*.ini).



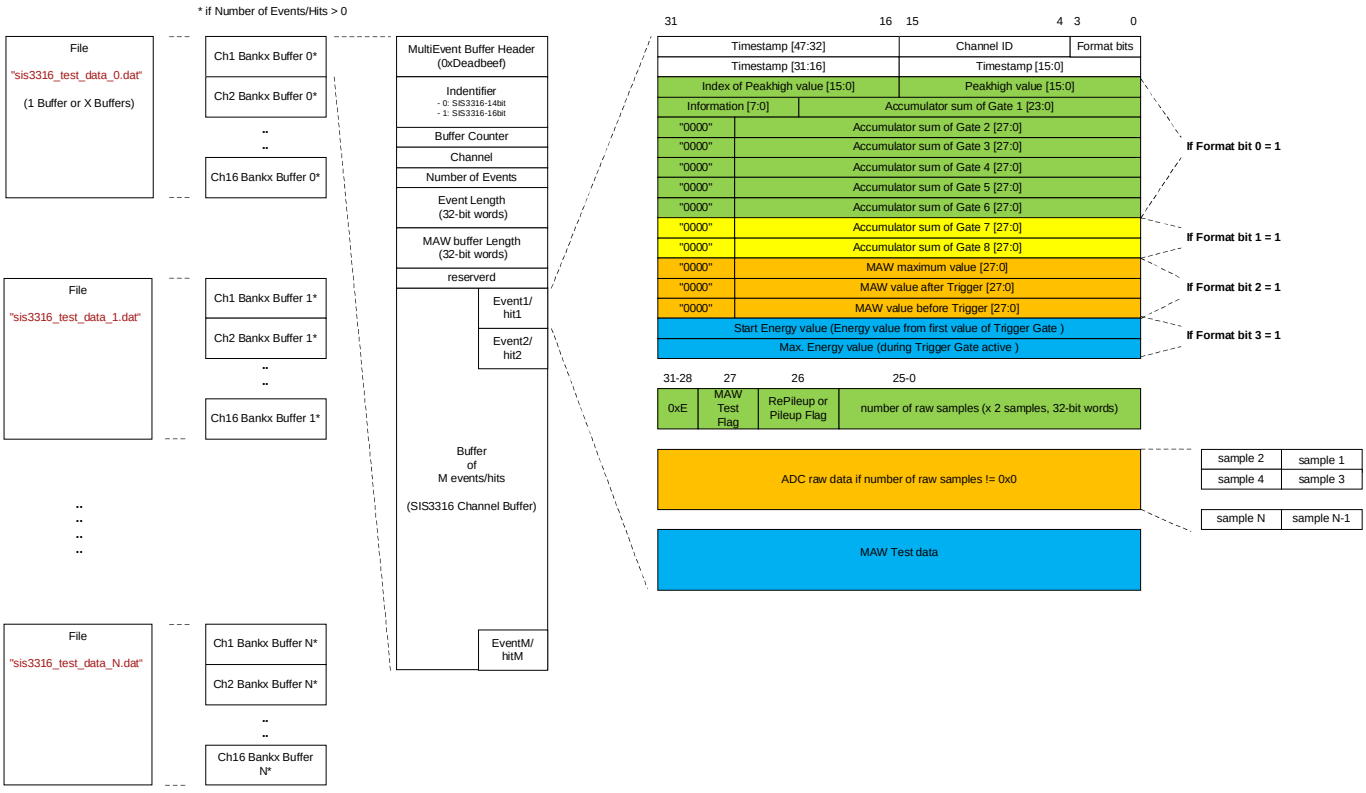
3.2.3 Write Data File Format

It is possible to write the sampled data to a file with the “Multi Event/Hit – Double Bank Mode” (marked in red). The check button “Write Data to File enable” has to be set (marked in green). Select “...” button (marked in orange) to get a browser window to define or select a file name.
The program will save 100 “banks” with each 1000 Hits/Events (marked in blue).



Depending on the check button “Write multiple Files” the 100 bank buffers will be saved to one or to 100 files.

File Data Format:



4 Index