UCLA Plasma Physics
VME FADC System

Labview Interface
Getting Started Booklet

SIS GmbH
Harksheider Str. 102A
22399 Hamburg
Germany

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

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

Version: 1.00 as of 18.12.02

Revisions Liste:

Revision	Date	Modification
0.01	09.12.02	Generation
1.00	18.12.02	Release

1 Table of contents

2 Introduction

This is the initial release of documentation for Labview support under Windows 2000 and XP for the SIS1100/3100 PCI to VME interface. The first guess setup of the Labview interface provides a minimum interface for high performance VME access with the interface in combination with SIS330x digitizers.

Both documentation and extensions to the software will be available online under:
http://www.struck.de/win1100.htm

3 Getting started

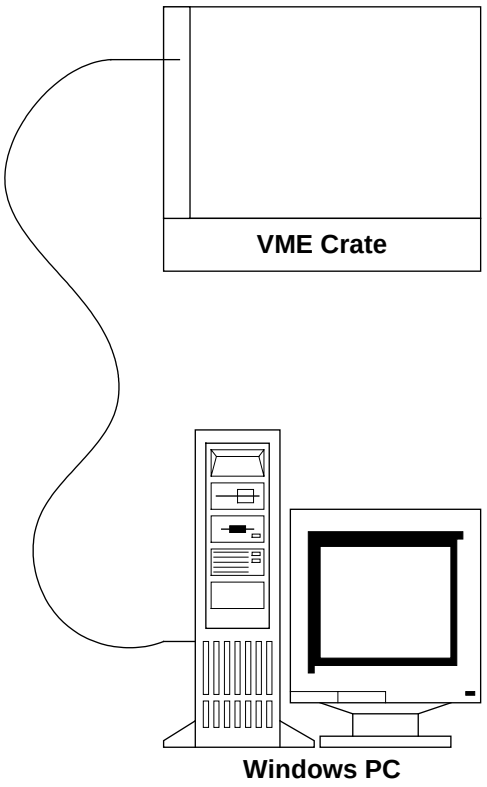
- Install the full Windows 2000 or XP driver for the SIS1100/3100 PCI to VME interface on your PC as described in the "SIS1100/3100 Win Driver Programmers Manual".
- Install both the SIS1100 and SIS3100 cards and establish the optical link connection as described in the same document.
- Power up the VME crate and start your PC. Install the SIS3100Base executable as described in the Win Driver manual also. Run the program and verify with a SIS1100_CTRL read (with address offset 0) and a SIS3100_CTRL read (with address offset 0 also), that communication with the link is working fine.
- Copy the sis11003100vis directory with the VIs and the Debug directory with the sis1100w_debug.dll to the directory of your choice and make sure, that all calls within the VIs resolve the calls to the library.
- Use/modify and run the VIs as required by your application. For a first VME access you may just want to run the initsis11003100 VI and the vme_dma_read:_split VI, which reads and displays data from the (unitialized) memory of one of the digitizers. If you run the VI in a loop you will see, that the M (master) LED on the SIS3100 and the S (slave) LED on one of the ADCs will be on during the actual data transfers. By using the address modifier 0x20 for the DMA you will go to 2e VME readout. The actual VME readout speed can be measured by connecting the VME DS0 line to an oscilloscope. In the case of a 2e VME readout cycle 8 bytes worth of data are strobed with both the leading and the trailing edge of the data strobe and you should measure something in the order of 200 ns/16 Bytes (equivalent to some 80 MBytes/s)
- Required external NIM/LEMO cabling will depend on the nature of the application.

Acquire first waveforms with SIS330x setup:
Connect a pulse generator with a 50 KHz sine of peak to peak amplitude of 1 - 5V and offset voltage 0 V.

- run initsis11003100 VI
- run clocklopVI
- run event_config VI (with wrap mode set to ON)
- run enableclockandstart VI
- run waitforstop VI
- use LEMO cable and connect clock output of one SIS330x to stop inputs of all sampling SIS330x until SAM LED goes off
- run readoutloop VI

4 System Overview

The system consists of a PC and a VME crate as illustrated below. The two components are linked with a PCI to VME interface.



4.1 PCI System Components

The PCI side of the system consists of following components

- PCI PC with Windows 2000 or XP
- PC peripherals (US keyboard, mouse and monitor)
- SIS1100 PCI Gigabit link card (as part of the SIS1100/3100 PCI to VME interface)

Note: PC and peripherals are not part of the PO/shipment

4.2 VME System Components

The VME side of the data acquisition system consists of following components

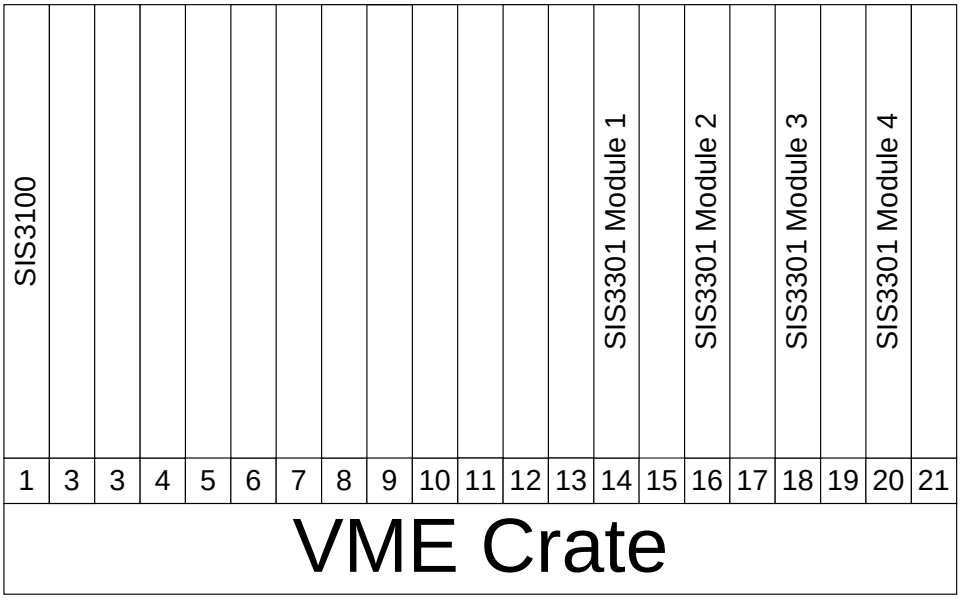
- SIS3000-2 VME crate (110 V version)
- SIS3100 VME Sequencer
- SIS3301 VME ADCs (4 modules)

5 VME Crate

A SIS3300-2 with 21 VME slots in 110V compatible line voltage is used as VME crate.

5.1 VME Crate occupancy

The crate occupancy of the SIS3000-2-A-110 VME crate at the time of shipment is illustrated in the figure below.



6 Software

6.1 Implemented calls

6.1.1 Infrastructure

The infrastructure calls are required to initialize the driver and the hardware. In most cases it will be good enough to use the init11003100 VI to get going.

```
sis1100w_Find_No_Of_sis1100(unsigned long *value);
sis1100w_Init(void);
sis1100w_init_sis3100(unsigned long *value);
```

6.1.2 VME calls

The VME calls are in charge of the actual work. So far the available calls are:

```
sis1100w_Vme_Single_Read(unsigned long addr, unsigned long am,
unsigned long size, unsigned long *data);

sis1100w_Vme_Single_Write(unsigned long addr, unsigned long
am, unsigned long size, unsigned long data);

sis1100w_Vme_Dma_Read(unsigned long addr, unsigned long am,
unsigned long size, unsigned long fifo_mode, unsigned long
*dmabufs, unsigned long req_num_data, unsigned long
*got_num_data);
```

6.2 Implemented VIs

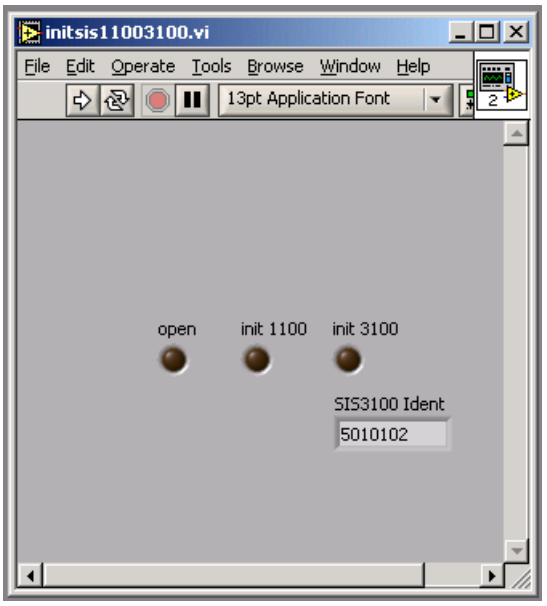
The table below lists the implemented Vis.

VI	Section	Function
VME		
initsis11003100	6.2.1	Initialize SIS1100 and 3100 and variables
vme_read	6.2.2	Single cycle VME read routine
vme_a32d32_read	6.2.3	A32 D32 case of above
vme_write	6.2.4	Single cycle VME write routine
vme_a32d32_write	6.2.5	A32 D32 case of above
vme_dma_read	6.2.6	Block transfer VME read routine
vme_dma_read_split	6.2.7	Special case of above with display
SIS330x specific		
modulebase	6.2.8	Computes address offset of ADC N
clockloop	6.2.9	Clock speed/source and other control register defined parameters (loop over modules)
event_config	6.2.10	Selection of event configuration defined parameters (loop over modules)
enableclock	6.2.11	Set acquisition clock to bank 1/2
enableclockandstart	6.2.12	as above with additional VME start
waitforstop	6.2.13	Wait for SIS330x to stop digitizing
readoutloop	6.2.14	DMA readout of 1st event
SIS3300_start_delay	6.2.15	Set start delay for single SIS330x
SIS3300_stop_delay	6.2.16	Set stop delay for single SIS330x
userledloop	6.2.17	Switch on/off user LED on N digitizers

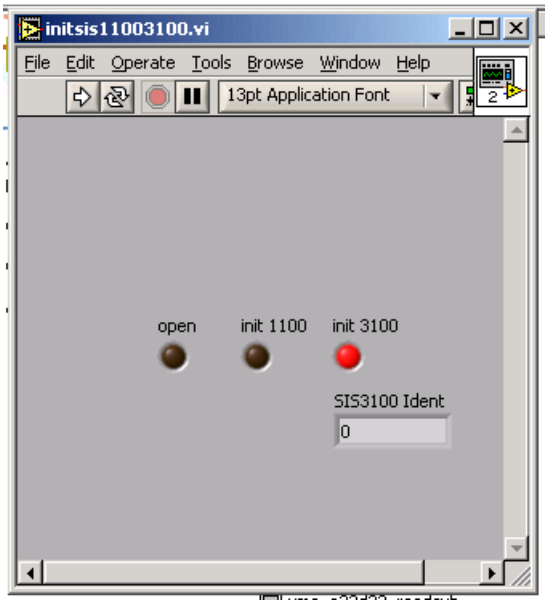
6.2.1 initsis11003100

This VI initializes the SIS1100 and the SIS3100 cards and initializes variables for execution of VME cycles. It has to be executed before any other VI can be used. The identity register of the SIS3100 card will be returned ff the VME crates is powered up and all steps are executed successfully.

Successful completion:



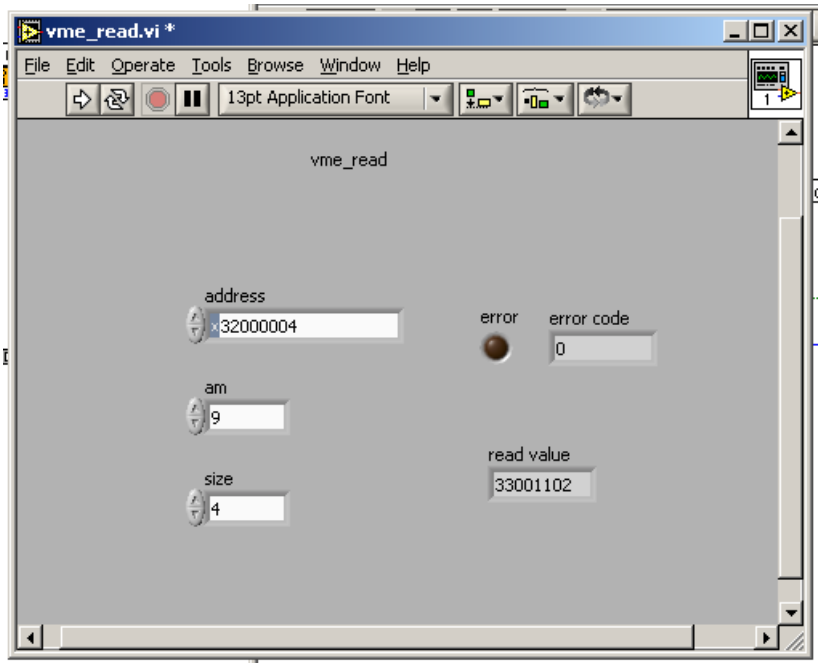
Completion with error (VME crate off):



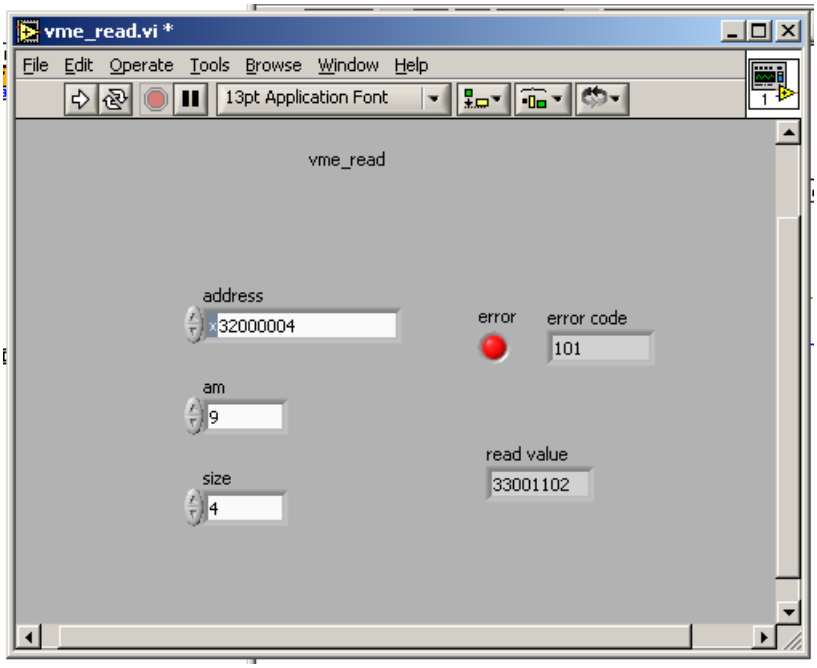
6.2.2 vme_read

Generic single cycle VME read. Refer to the SIS1100/1300 manual for an error code table (i.e. 0x211 signals VME bus error)

Successful completion:



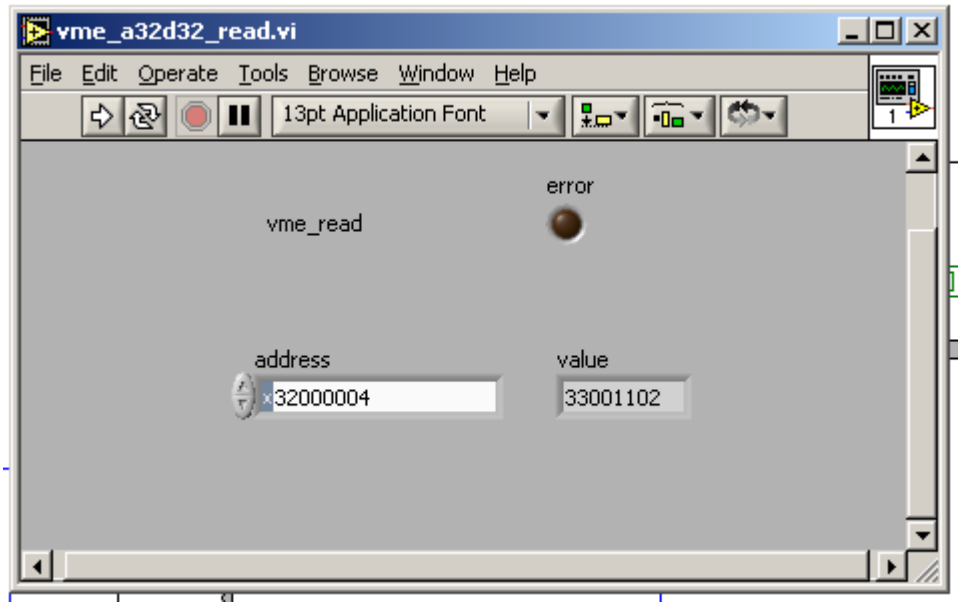
Error case (no link):



6.2.3 vme_a32d32_read

Special case of generic vme_read routine with AM=9 and size=4, i.e. single A32 D32 read as required for most SIS VME slaves.

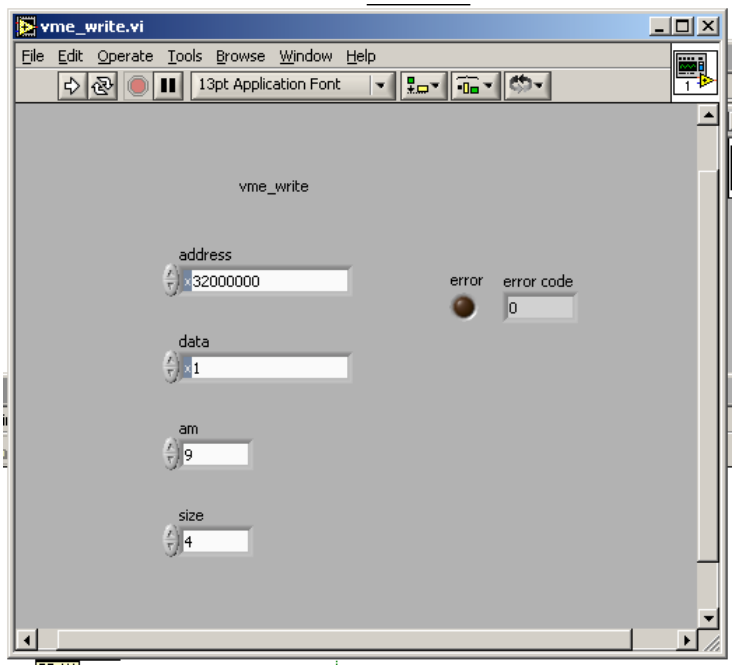
Successful read from SIS3300 Id. register:



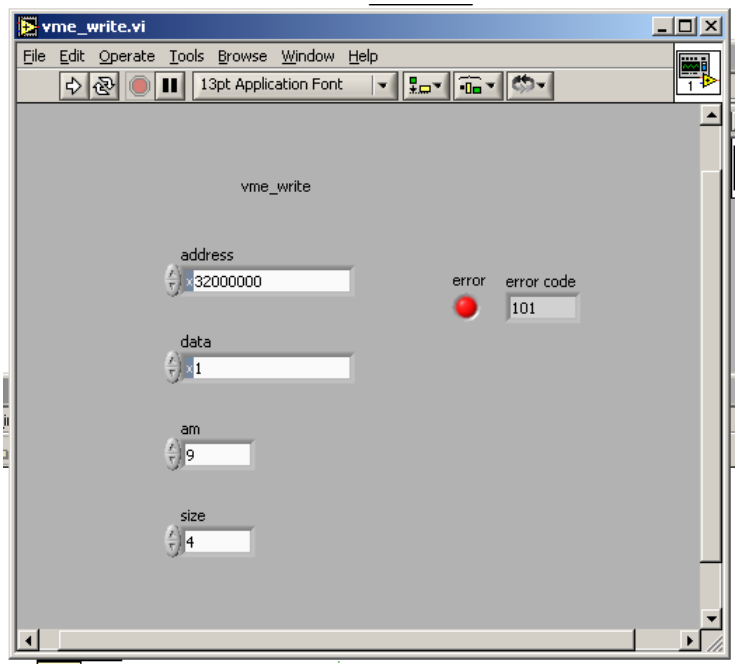
6.2.4 vme_write

Generic single cycle VME write. Refer to the SIS1100/1300 manual for an error code table (i.e. 0x211 signals VME bus error)

Successful completion:



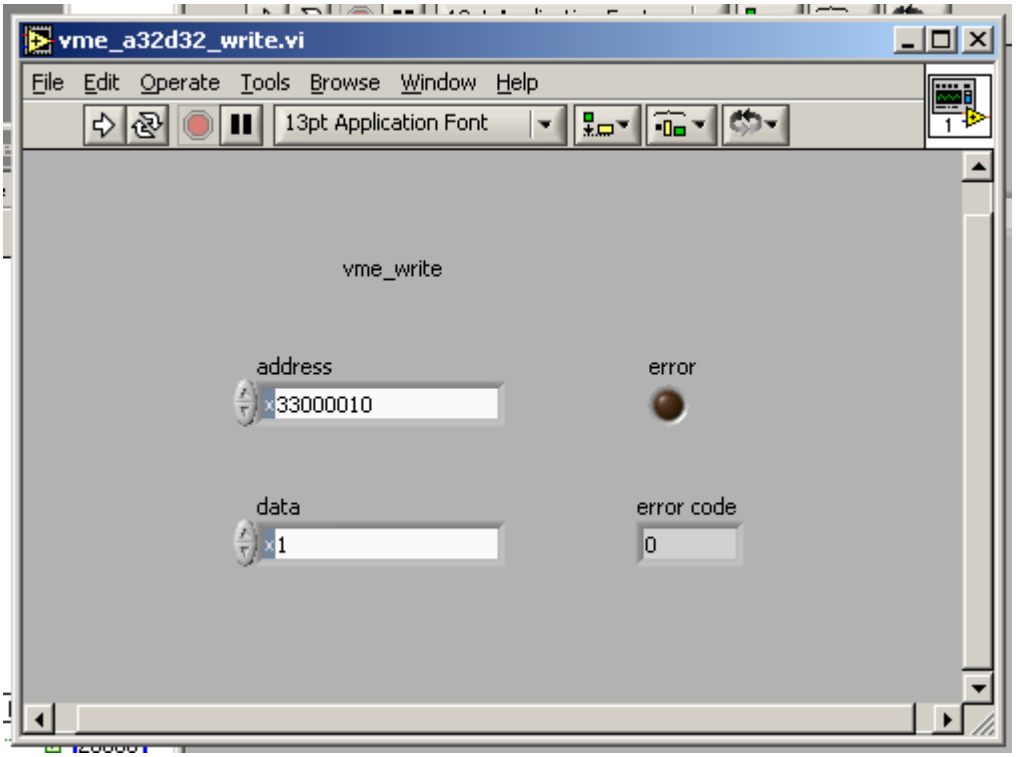
Error case (no link):



6.2.5 vme_a32d32_write

Special case of generic vme_write routine with AM=9 and size=4, i.e. single A32 D32 write as required for most SIS VME slaves.

Successful write to SIS330x acquisition register:

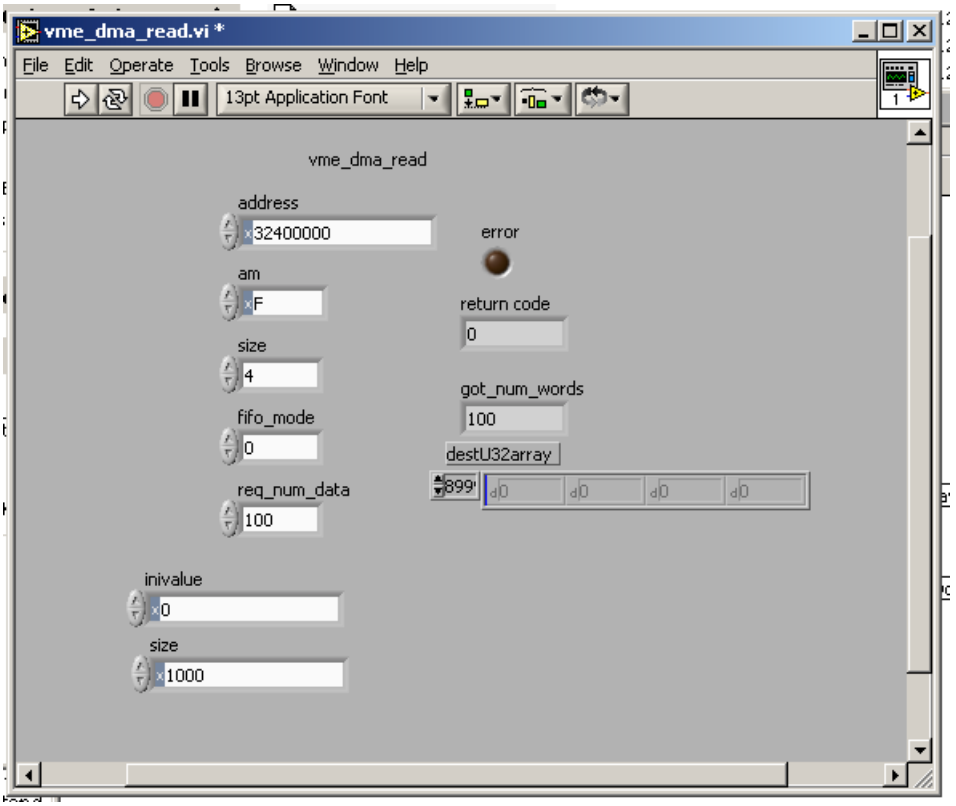


6.2.6 vme_dma_read

Generic VME block transfer read routine. Readout speeds in excess of 80 MBytes/s are achieved in 2e VME (AM=0x20) from SIS330x memory. The size of the allocated data array has to be greater equal than the number of data to be read (req_num_data). The got_num_words output holds the actual number of read words (which has to equal the req_num_data unless an unknown number of data words is read from a FIFO). Address increment can be disabled by setting fifo_mode to 1..

Note: Both req_num_data and got_num_words are hexadecimal values

Successful BLT32 read from SIS330x bank1 memory

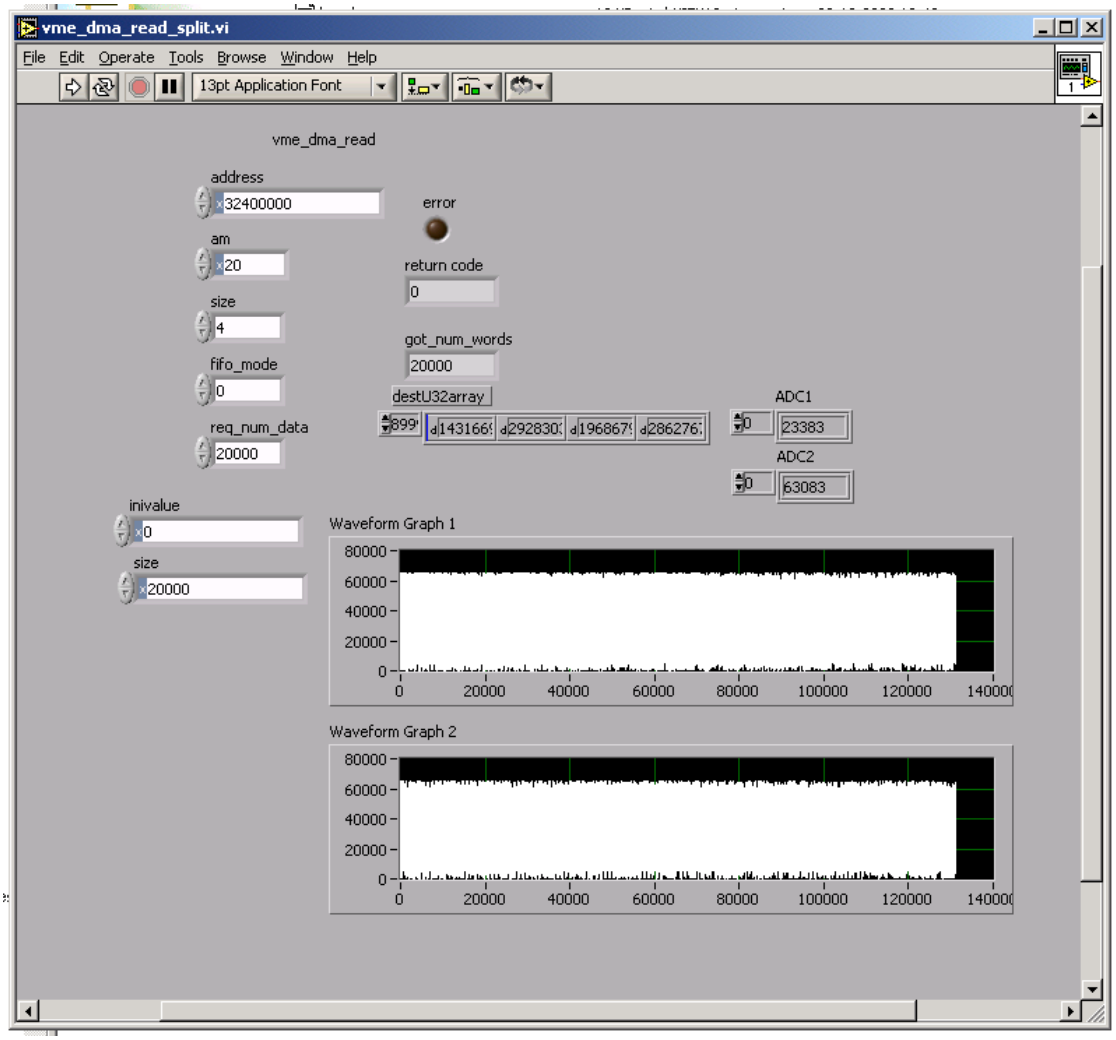


6.2.7 vme_dma_read_split

Special case of generic VME block transfer read routine (for readout of SIS330x data e.g.). Data are displayed as waveforms.

Note: the overrange bit of the SIS330x is not masked out before display, i.e. the displayed trace may look corrupted if part of the digitized value are outside of the range of the digitizer.

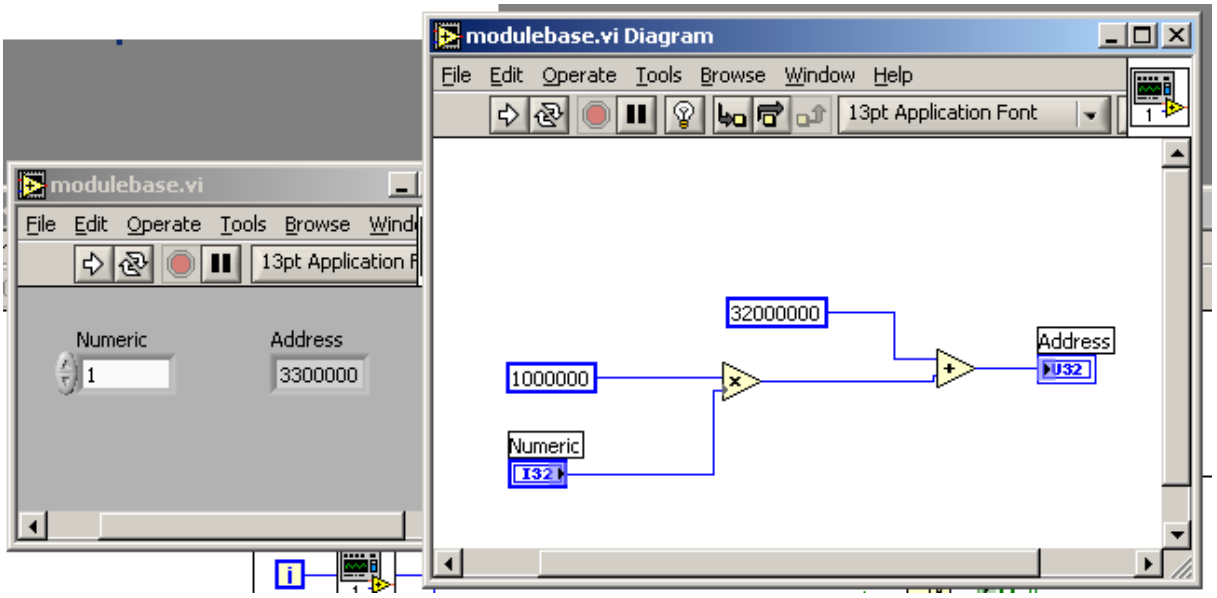
Successful 2e read from (unitialized) SIS330x bank1 memory



6.2.8 modulebase

This VI computes the address offset of digitizer N. Its used in the other VIs that access the digitizer modules in a loop. The base addresses are set up in a fashion, that the first digitizer (N=0) is set to a base address of 0x3200 0000 and consecutive modules have an increment of 0x100 0000 (i.e one notch higher on the SW2 base address rotary switch of the card) .

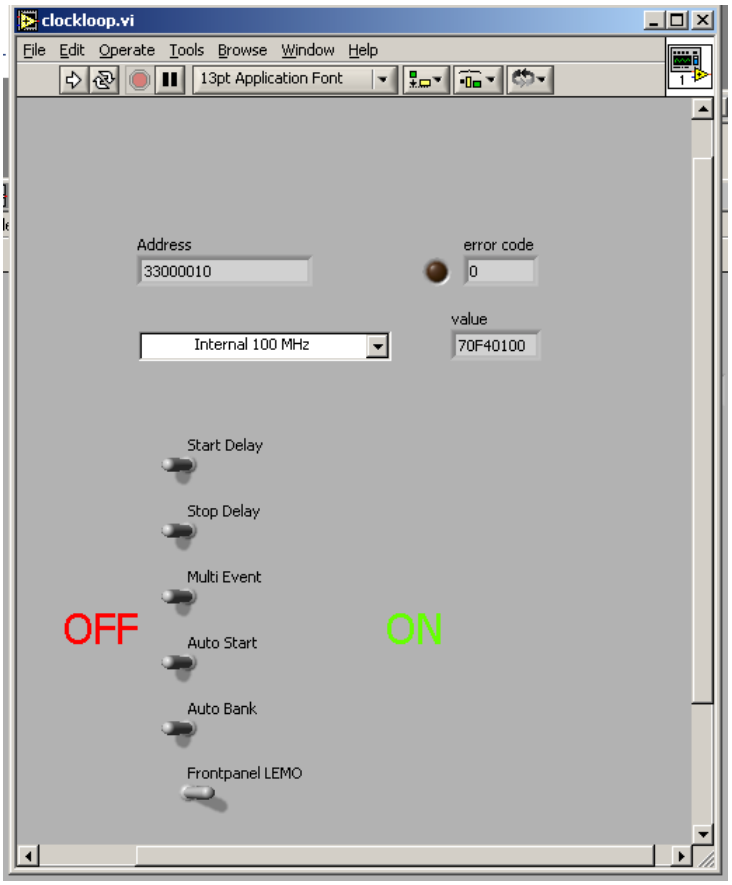
Modulebase VI and diagram



6.2.9 clocklop

The clockloop VI allows to set parameters that are defined with the control register of the SIS330x. Besides clock speed/source selection activation of the front panel LEMO inputs, part of multi event configuration and start/stop delay activation are available. The VI loops over the defined modules.

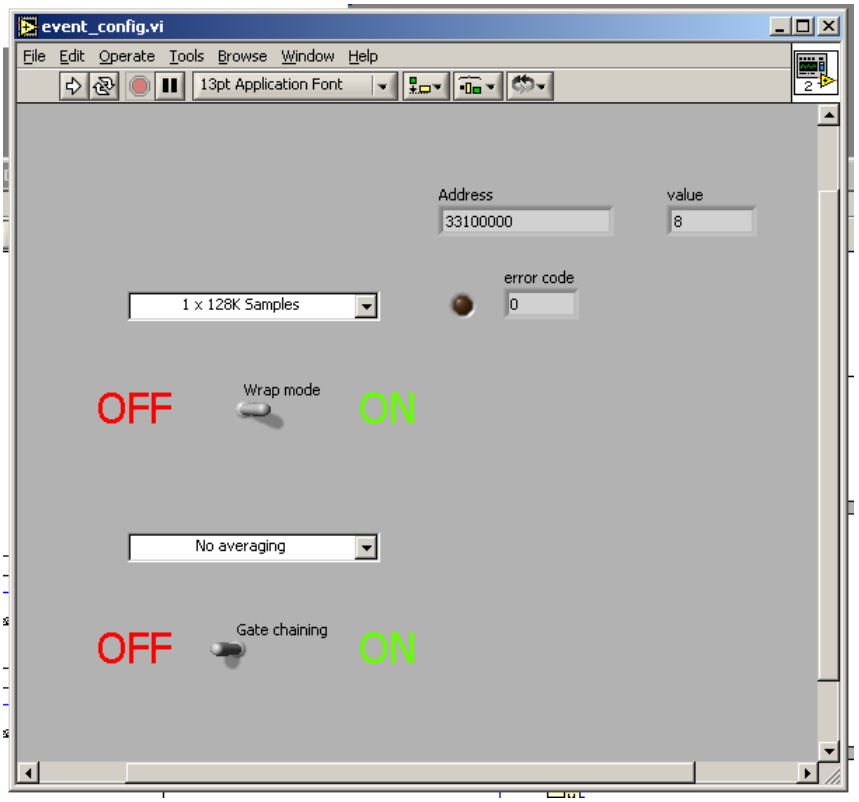
clockloop VI



6.2.10 event_config

The event configuration register is used for the definition of central acquisition parameters, like wrap/no wrap, event size. The VI loops over the defined modules.

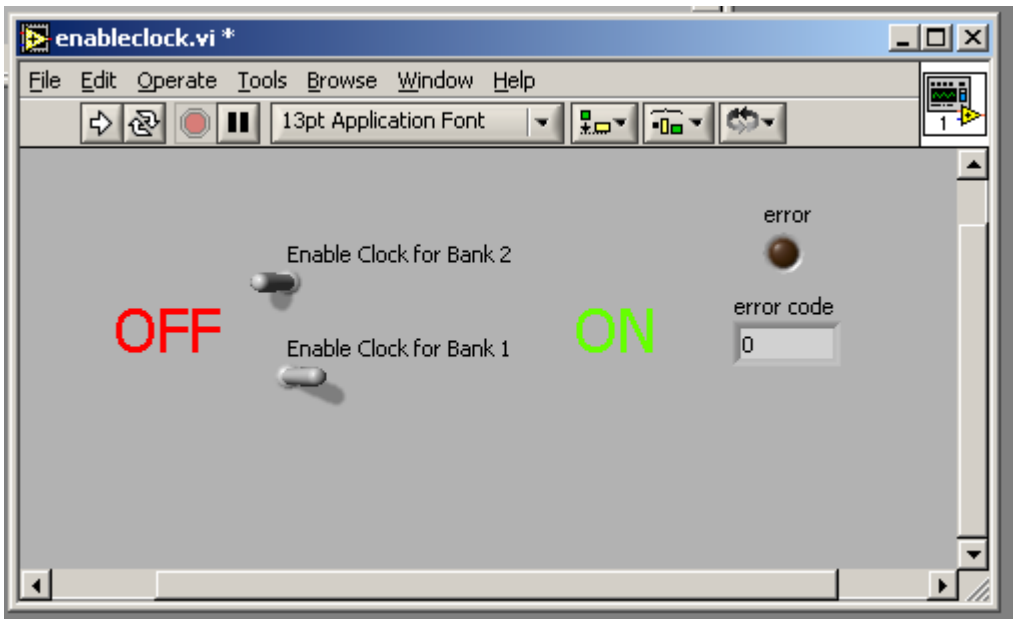
event_config VI



6.2.11 enableclock

Allows to set the acquisition clock to bank l or bank 2 for all defined modules. Note this configuration will have to be done anew after every stop of the digitizer(s).

enableclock VI



6.2.12 enableclockandstart

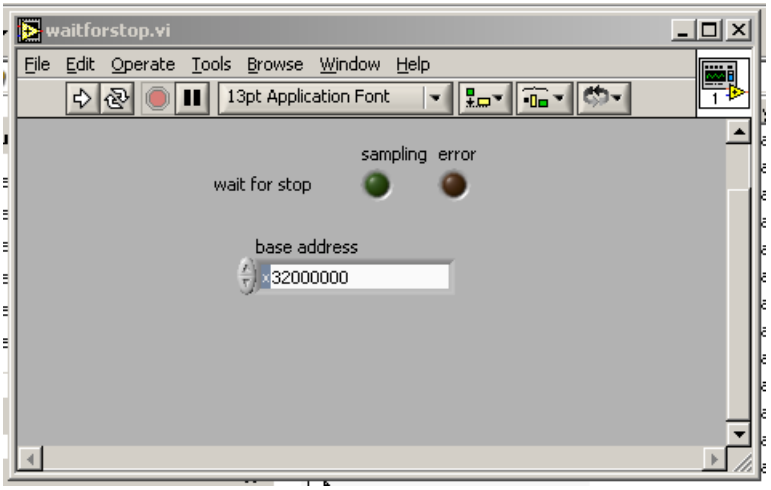
Same as enableclock, but issues VME start to all defined modules in addition (i.e. to be used in conjunction with stop by external stop/trigger and wrap mode.

6.2.13 waitforstop

Waits (i.e. polls) for the digitizer at base address to stop sampling. The green sampling LED will be on until the module has received a stop (or has reached end of the event/bank without wrap mode).

Note: it is good enough to poll for the end of acquisition in one of the digitizers as long as external logic makes sure, that the trigger/stop is fanned out to all modules and that all modules were ready to accept a start.

Waitforstop VI

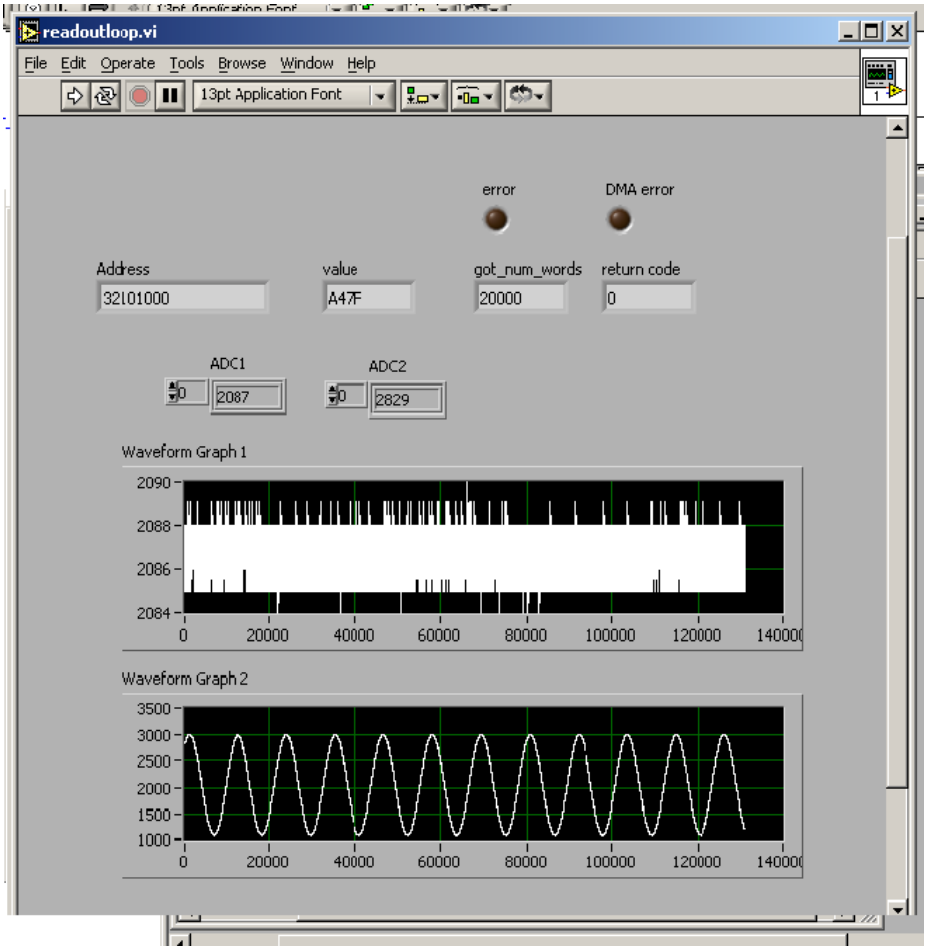


6.2.14 readoutloop

Reads and displays the first memory bank from all defined digitizers. The first stop pointer is read from the event directory and data are rotated to have the stop in the same place for all digitizers. The overrange bit is masked out before display.

Notes: uses the subdma2eadc VI, which is preset to 0x20000 words (i.e. full memory). For smaller event size fixed readout length has to be replaced with a parameter.
A more complex scheme is needed to handle multi event and dual bank readout (i.e. read all required stop pointers and rotate individual events, moer address arithmetic).

readoutloop VI

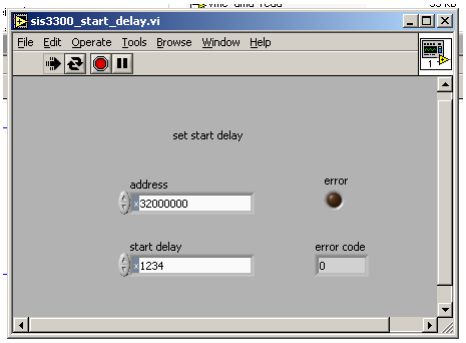


6.2.15 sis3300_start_delay

Set the start delay on the module at base address (address field in the VI). Typically will be combined with modulebase and a loop over all used digitizers.

Note: start delay is activated in the clockloop VI

SIS3300_start_delay

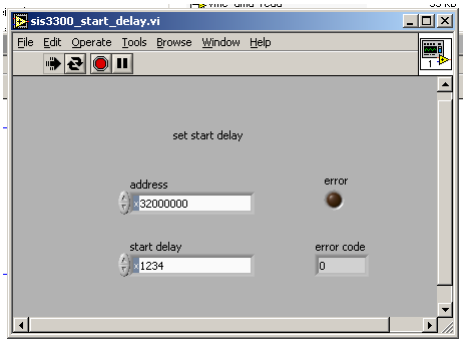


6.2.16 sis3300_stop_delay

Set the stop delay on the module at base address (address field in the VI). Typically will be combined with modulebase and a loop over all used digitizers.

Note: stop delay is activated in the clockloop VI

sis3300_stop_delay



6.2.17 userledloop

You can switch on/off the user LEDs of all defined modules by clicking on the User LED with the VI running in loop mode. The number of defined modules is changed in the diagram.

running userledloop VI (with single digitizer at address 0x32000000)

7 Hardware configuration at shipment

7.1 VME base addresses

The 4 ADCs are configured for A32 addressing, the table below lists the base addresses. As can be seen during the readout with a VI that loops over the modules the modules are read from left to right in the crate, the module with the lowest VME address is read first and is the most left hand ADC module accordingly.

VME base address	Module
0x32000000	SIS3301 ADC #1
0x33000000	SIS3301 ADC #2
0x34000000	SIS3301 ADC #3
0x35000000	SIS3301 ADC #4

The SIS38xx and the SIS3100 do not feature VME slave functionality in this setup.

7.2 ADC input configuration

The SIS3301 modules are configure for an input range of +2,5 V to –2,5 V into an input impedance of 50 Ω.
The control inputs are configured for NIM input levels into 50 Ω.

8 Index