GX2472

Two Channel 70MS/s Waveform Digitizer

Driver Manual

Revision 1.2, May 2007

GEOTEST

MARVIN TEST SYSTEMS, INC.

Safety and Handling

Each product shipped by Geotest is carefully inspected and tested prior to shipping. The shipping box provides protection during shipment, and can be used for storage of both the hardware and the software when they are not in use.
The circuit boards are extremely delicate and require care in handling and installation. Do not remove the boards from their protective plastic coverings or from the shipping box until you are ready to install the boards into your computer.
If a board is removed from the computer for any reason, be sure to store it in its original shipping box. Do not store boards on top of workbenches or other areas where they might be susceptible to damage or exposure to strong electromagnetic or electrostatic fields. Store circuit boards in protective anti-electrostatic wrapping and away from electromagnetic fields. Be sure to make a single copy of the software diskette for installation. Store the original diskette in a safe place away from electromagnetic or electrostatic fields. Return compact disks (CD) to their protective case or sleeve and store in the original shipping box or other suitable location.

Warranty

Geotest products are warranted against defects in materials and workmanship for a period of 12 months. Software products and accessories are warranted for 3 months, unless covered by software support or maintenance agreement. Geotest shall repair or replace (at its discretion) any defective product during the stated warranty period. The software warranty includes any revisions or new versions released during the warranty period. Revisions and new versions may be covered by a software support agreement. If you need to return a board, please contact Geotest Customer Technical Services Department via http://www.geotestinc.com/magic - the Geotest on-line support system.

If You Need Help

Visit our web site at http://www.geotestinc.com for more information about Geotest products, services and support options. Our web site contains sections describing support options and application notes, as well as a download area for downloading patches, example, patches and new or revised instrument drivers. To submit a support issue including suggestion, bug report or question please use the following link: http://www.geotestinc.com/magic
You can also use Geotest technical support phone line (949) 263-2222. This service is available between 7:30 AM and 5:30 PM Pacific Standard Time.

Disclaimer

In no event shall Geotest or any of its representatives be liable for any consequential damages whatsoever (including unlimited damages for loss of business profits, business interruption, loss of business information, or any other losses) arising out of the use of or inability to use this product, even if Geotest has been advised of the possibility for such damages.

Copyright

Trademarks

ATEasy®	Geotest – MTS Inc.
C++ Builder, Borland C++, Pascal and Delphi	Borland Corporation
Microsoft Developer Studio, Microsoft Visual C++, Microsoft Visual Basic,, .NET, Windows 95, 98, NT, ME, 2000 and XP	Microsoft Corporation

All other trademarks are the property of their respective owners.

TABLE OF CONTENTS

Introduction

The GX2472 is a 70MHz dual channel Waveform Digitizer. For software development and integration in a PXI system, the card is provided with a software driver, utility software, a demo program for LabView and a calibration tool.

The main part of the GX2472 driver consisst of a Windows dynamic link library, the gx2472.DLL

For Labview users the driver also includes a LabView library with the gx2472.dll driver functions.

This manual describes the functions of the gx2472.dll.

1 GX2472 driver package

The driver includes the following items:

1) A low level driver for direct communication;
2) The user mode driver, gx2472.dll;
3) A Labview library, gx2472.llb
4) A LabWindows driver (function tree) gx2472.fp including the source file gx2472.c and the header file gx2472.h. For basic programmers a GX2472.bas file is included.

The following low level drivers can be installed:

1) A kernel mode pxi driver, gx2472.sys;
2) VISA based driver which requires the installation of VISA which is available from National Instruments.

1.1 Installation

Install the GX2472 Driver Software before the hardware is placed in the system.
Place the installation CD in the CD-ROM. If the installation program does not start automatically, run the program setup.exe (placed in the root of the CD-ROM).

If the software is installed on a Windows NT based operating system (Win2000, WinNT, WinXP), you should have Administrator rights.

The PXI Kernel Driver cannot be installed on Windows95 or Windows NT. This selection will be disabled if one of these operating systems is detected.

After installation shutdown the computer and place the GX2472 in the system. After turning on the computer the operation system should automatically detect the new hardware and install the low level driver.

1.2 Uninstalling the low level driver

Before switching from the low-level driver to the VISA based driver, uninstall the current low-level driver

For uninstalling the low level driver, perform the following steps:

1. Start the Device Manager
2. Select the “GX2472 PXI driver” and uninstall the driver;
3. Go to the Windows inf-directory (e.g. Windows\inf, hidden directory).
4. Delete the gx2472_xxxx.inf file. The addition of xxxx indicates the Windows Operation System version.
5. If installed with the pxi kernel mode driver, go the Windows sub-directory System\Drivers (e.g. C:\Windows\System\Drivers).
6. Delete the gx2472.sys file.

1.3 “Manual” installation of the low level driver

1. Copy the corresponding inf-file to the Windows Inf-subdirectory (e.g. C:\Windows\inf).
   The inf-files can be found in the following directories of the CD-ROM:
   For Visa installation:
   Directory : \Driver\Visa

   Windows9x “gx2472_9x.inf”
   WindowsNT4 “gx2472_nt4.inf”
   Windows2000 “gx2472_nt5.inf”
   WindowsXP “gx2472_nt5.inf”

For the PXI kernel mode driver installation:

Directory: \Driver\Kernel

Windows98 “gx2472.inf”
Windows2000 “gx2472.inf”
WindowsXP “gx2472.inf”

The inf-subdirectory is a hidden directory.

2. If the pxi kernel mode driver is installed, copy also the gx2472.sys file. This file can be found in the directory \Drivers\Kernel\Winxx, where xx indicates the operating system. Copy this file to the Windows sub-directory \System\Drivers (e.g. C:\Windows\System\Drivers);
3. Turn the system off and place the GX2472 in the system;
4. Turn the system on and reboot the host computer;
5. The operation system should detect new hardware;

Be sure driver signing is set to Ignore or Warn, when installing the GX2472 pxi kernel driver on a Win2000 or WinXP system.

If the “Add new hardware” wizard doesn't start or something went wrong during installation, start the Device Manager. Select the device (normally marked with a question mark, if the driver could not be loaded) and select properties. Then install/reinstall the driver.

2 GX2472 DLL functions.

This chapter describes the functions of the dll. Each function and its parameters are described in a table. The following parameter types are used:

Type	Details
unsigned long	4-byte (32 bit) unsigned long
double	8-byte floating point
unsigned long*	reference variable (pointer) to a 4-byte (32 bit) unsigned long
double*	reference variable (pointer) to a 8-byte floating point

All functions use the standard calling conventions (stdcall or WINAPI). Every function returns the ViStatus (type: 32-bit integer). A negative value corresponds to an error. After a successful completion the return status is VI_SUCCESS, which corresponds to a 0. Possible status codes can be found in Chapter 3.

2.1 gx2472_AutoCalibrate (vi, mode, displayTime)

Description:

This function performs an auto calibration procedure. The input DC-offset should already be calibrated. The calibration voltages can be monitored on the negative input of the module's two channels.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
mode	unsigned long	in	mode	0 no monitoring 1 monitor at neg. input
displayTime	double	in	display time	time to display voltage at negative input

2.2 gx2472_CheckTestStatus (vi, testStatus)

Description:

This function returns the test status. It will return a 1 if the memory address counter passes the maximum address counter value (ready). Bit 0 represents channel A, bit 1 channel B.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
testStatus	unsigned long*	out	Test Status	0 A & B not ready 1 Channel A ready 2 Channel B ready 3 A & B ready

2.3 gx2472_close(vi)

Description:
Close the card session. All resources belonging to the card will be released.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32

2.4 gx2472_CodeToVoltage (vi, ADCCode, voltage)

Description:

This function converts an ADC Code to the corresponding voltage.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
ADCCode	unsigned long	in	mode	0 to 2^32
voltage	double*	out	voltage	corresponding voltage

2.5 gx2472_ConnectCard(vi , positveInputConnection, negativeInputConnection)

Description:

This function connects/disconnects the input relays. There are 4 relays for each input. Each relay can be controlled with a bit:

- Bit 1 (value 1 Hex) : Input relay;
- Bit 2 (value 2 Hex) : 50 Ohm relay;
- Bit 3 (value 4 Hex) : 50 Ohm DC relay, this bit controls the positive and negative channel simultaneously;
- Bit 4 (value 8 Hex) : Ground relay, connect input to ground.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
positiveInputConnection	unsigned long	in	positive input	see description
negativeInputConnection	unsigned long	in	negative input	see description

2.6 gx2472_error_message(vi, statusCode, errorMessage)

Description:
Get the text string corresponding the statusCode. ErrorMessag should be at least have a size of 255.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
statusCode	unsigned long	in	The status code returned by a gx2472 driver function	0 to 2^32
errorMessage	char*	out	The error message	string

gx2472_error_query(vi, errorCode, Message)

Description:
Error Query is not supported by this instrument. This function only exists for compliance with the VXI Plug and Play specifications.

It returns a defined value VI_WARN_NSUP_ERROR_QUERY.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
errorCode	unsigned long	in		0 to 2^32
Message	char*	out		string

2.7 gx2472_FindInstruments(buslist, devicelist, count)

Description:

Returns the number of available instruments and a list with the bus and device number for each instrument.
Buslist and devicelist should be arrays of at least 100 places.

Parameters:

Name	Type	Direction	Description	Value
buslist	unsigned long*	out	List with the bus location of the device(s)	
devicelist	unsigned long*	out	List with the device location of the device(s)	
count	unsigned long	out	Number of available instruments	0 to 100

2.8 gx2472_GetActiveChannel(vi , channel)

Description:
This function returns the active channel. The active channel is the channel, which can be configured with the corresponding driver functions. The channel can be set active with the function SetActiveChannel().

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
channel	unsigned long*	out	channel	1 = Channel A 2 = Channel B

2.9 gx2472_GetAddressCounter(vi, addresscounter)

Description:
This function reads the current position of the memory address counter from the active channel. If the loopmode is active and card is stopped capturing data, use this function to get the last captured data position.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
addresscounter	unsigned long*	out	Position of the address-counter	0 to 2^19

2.10 gx2472_GetCardConnection(vi, positiveInputConnection, negativeInputConnection)

Description:
Get the status of the input relays. This routine returns the connection status from the active channel. There are 4 relays for each input. Each relay can be controlled with a bit:

- Bit 1 (value 1 Hex) : Input relay;
- Bit 2 (value 2 Hex) : 50 Ohm relay;
- Bit 3 (value 4 Hex) : 50 Ohm DC relay, this bit controls the positive and negative channel simultaneously;
- Bit 4 (value 8 Hex) : Ground relay, connect input to ground.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
positiveInputConnection	unsigned long*	out	positive input	see description
negativeInputConnection	unsigned long*	out	negative input	see description

2.11 gx2472_GetCardId(vi , id)

Description:
This function reads the card ID. A card ID can be set with the function WriteCardId(). The card ID is placed in the on board serial EEPROM.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
id	unsigned long*	out	card id	0 to 2^32

2.12 gx2472_GetClockDivider (vi, clockDivider)

Description:
This function returns the clock divider value.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
clockDivider	unsigned long *	out	clock divider	1 to 256

2.13 gx2472_GetClockSource (vi, clockSource)

Description:
This function returns the clock source.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
clockSource	unsigned long *	out	clock source	0: Extern clock front 1: Intern clock 70 MHz 2: Intern clock 50 MHz

2.14 gx2472_GetDCOffsetLimitVoltages (vi, positiveVoltage, negativeVoltage)

Description:
Returns the DC Offset Limit Voltages. See function SetDCOffsetLimitVoltages(..).

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
positiveVoltage	double*	out	positive limit voltage	5 to 6
negativeVoltage	double*	out	negative limit voltage	-5 to - 6

2.15 gx2472_GetDCOffsetVoltage(vi , voltage)

Description:
This function will returns the current voltage of the DC-offset DAC (of the active channel). This voltage is previously set with the function SetDCOffsetVoltage().

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
voltage	double*	out	voltage of offset DAC	-5 to +5 volt

2.16 gx2472_GetFilter (vi, filterStatus)

Description:
This function returns the filter position.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
filterStatus	unsigned long*	out	filter status	0: Disconnect 1: Bypass filter 2: 6 MHz filter 3: 15 MHz filter 4: 30 MHz filter

2.17 gx2472_GetGainCalCode (vi, gainCalibrationCode)

Description:
This function returns the calibration code for gain calibration DAC.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
gainCalibrationCode	unsigned long*	out	gain cal. code	0 to 2^10

2.18 gx2472_GetInputVoltage (vi, voltage, code, averages, timeOut)

Description:
The function measures the input voltage Averages times and averages the voltage. It returns the voltage and the corresponding ADC code.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
voltage	double*	out	measured voltage	depends on range
code	unsigned long*	out	adc code	0 to 2^14
average	unsigned long	in	averages	1 to 2^19
timeOut	unsigned long	in	time out	0 to 2^32 no timeout it time out is 0

2.19 gx2472_GetLoopMode (vi, loopMode)

Description:
This function returns the Loop Mode status.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
loopMode	unsigned long*	out	loop mode status	0 or 1

2.20 gx2472_GetOffsetCalDacCode(ci, code)

Description:
This function returns the offset calibration dac code.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
code	unsigned long*	out	code	0 to 2^10

2.21 gx2472_GetRange (vi, range)

Description:
This function returns the range setting.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
range	unsigned long*	out	code	1 to 6

2.22 gx2472_GetSampleDivider (vi, sampleDivider)

Description:
The function returns the sample divider value. THIS FUNCTIION IS OBSOLETE.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
sampleDivider	unsigned long*	out	sample dividider	1 to 65536

2.23 gx2472_GetSoftwareTriggerStatus(vi, triggerstatus)

Description:
This function returns the trigger status.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
triggerstatus	unsigned long *	out	current trigger status	0 = no channels triggered 1 = channel A triggered 2 = channel B triggered 3 = both channels triggered

2.24 gx2472_GetTriggerInput (vi, triggerSource, triggerMode)

Description:
This function returns the trigger source and trigger mode settings.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
triggerSource	unsigned long*	out	trigger source	0 Front 1 PXI 0 2 PXI 1 3 PXI 2 4 PXI 3 5 PXI 4 6 PXI 5 7 PXI Star 8 Software 9 Level
triggerMode	unsigned long*	out	trigger mode	0 positive level 1 negative level 2 positive edge retrigger 3 negative edge retrigger 4 positive edge continuous 5 negative edge continuous

2.25 gx2472_init(resourcediscription , IDQuery , reset, vi)

Description:
This function performs the following initialization actions:

- Opens a session to the specified device using the bus and device you specify.

- Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.

Note:

This function does not create new session if it is called repeatedly without closing it.
Call FindInstruments to get the available instruments in the system.

Parameters:

Name	Type	Direction	Description	Value
resourcediscription	char*	in	PXI[bus]:: [device]::INSTR	e.g. PXI1::2::INSTR
IDQuery	boolean	in	Set to true	0 or 1
reset	boolean	in	Set to false (not supported)	0 or 1
vi	Unsigned long*	out	Return instrument handle	

2.26 gx2472_MemoryTest (vi, level, errorAddress)

Description:
This function performs a memory test.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
level	unsigned long	in	level	0 = 1/10 of memory is tested 1 = total memory tested
errorAddress	unsigned long*	out	error address	if error, this parameter will contain the address which contains invalid data

2.27 gx2472_ReadAdcResults (vi, startPosition, samples, buffer, includeStatusBits)

Description:
This function reads the latest test results from the pd172. The results are a 14 bit hexadecimal code. If the status bits are included the results are 16 bit words, where the two upper bits contain the status bits. Bit 14 represents the overflow bit and bit 15 (MSB) an extra capture bit.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
startPosition	unsigned long	in	start position	0 to 2^19
samples	unsigned long	in	nr. of samples	1 to 2^19
buffer	unsigned long*	out	buffer for adc codes	
includeStatusBits	unsigned long	in	include the status bits	0 not included 1 included

2.28 gx2472_ReadEEPROM(vi , eeaddress , data)

Description:
This function reads a 16-bit word from the serial EEPROM at a serial EEPROM address determined by eeaddress. If the EEPROM address is previously written with the function WriteEEPROM(), the lower byte will correspond to the byte written with the function WriteEEPROM(). The upper byte will be the complement of the lower byte. This byte can be used for verifying purposes.
One on board EEPROM is used for both channels.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
eeadress	unsigned long	in	EEPROM address	0 to 255
data	unsigned long*	out	read data	0 to 2^16

2.29 gx2472_ReadInstrumentMemory(vi, data)

Description:
Read from the (capture) ram of the active channel. The ram address is determined by the address-counter. The address-counter can be initialized with the function

SetAddressCounter(). After this function call the address-counter is incremented with one step.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
data	unsigned long*	out	read data	0 to 2^16

2.30 gx2472_ReadInstrumentMemoryArray(vi , length , buf32, dataInterpretation)

Description:
Read length ram-places from the capture ram (of the active channel), starting from the current address-counter value. The ram address-counter can be initialized with the function SetAddressCounter(). After this function call the address-counter is incremented with “length” steps. Ram data can be interpret as test (adc) results. In this case the ram data will be shifted 2 places (right shift) and the upper bit will be inverted (adc digital data is 2's complement).

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
length	unsigned long	in	number of data elements to be read	0 to 2^18
buf32	unsigned long*	out	reference to a buffer for read data	0 to 2^32
dataInterpretation	unsigned long	in	data interpretation	0 raw data from ram 1 adc codes only

2.31 gx2472_ReadRegister(vi, address, data)

Description:
Read data from specified module register. Only for debug purposes.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
address	Unsigned long	in	Address	0 to 2^32
data	unsigned long*	out	read data	0 to 2^32

2.32 gx2472_reset(vi)

Description:
A reset is not supported by this instrument.

It returns a defined value VI_WARN_NSUP_RESET.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32

2.33 gx2472_revision_query(vi, driverRev, instrRev)

Description:
This function returns the revision number of the instrument driver.

This instrument does not contain firmware so the instr_rev parameter is not valid. It only exists for compliance with the VXI Plug and Play specification.

This function always returns the defined value VI_WARN_NSUP_REV_QUERY.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
driverRev	char*	in	Driver revision	
instrRev	char*	in	Instrument revision	

2.34 gx2472_self_test(vi, testResult, errorMessage)

Description:
Self Test is not supported by this instrument.

It returns a defined value VI_WARN_NSUP_SELF_TEST.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
testResult	unsigned long	in		0 to 2^32
errorMessage	char*	out		

2.35 gx2472_SetActiveChannel(vi , channel)

Description:
Select the active channel. This function selects the channel to be updated.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
channel	unsigned long	in	active channel	1 Channel-A 2 Channel-B

2.36 gx2472_SetAddressCounter (vi, addressCounterPosition)

Description:
This function programs the address counter. Use this function to read from or write to a specified address.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
addressCounterPosition	unsigned long	in	address counter position	0 to 2^19

2.37 gx2472_SetClockDivider(vi , clockdivider)

Description:
Set the clock-divider. Programs the divider for the sample clock of the active channel. The clock divider divides the clock selected with SetClockSource() and determines the sample rate.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
clockdivider	unsigned long	in	clock divider value	1 to 256

2.38 gx2472_SetClockOutputEnable(vi, enable)

Description:
Enable the sample clock output. If annternal clock is selected and the clock output is enabled, the sample clock (clock/clockdivider) is available on the front clock connection.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
enable	unsigned long	in	enable clock output	0 = Disable clock output 1 = Enable clock output

2.39 gx2472_SetClockSource(vi, clocksource)

Description:
Select the desired clock source (for the selected channel). The clock source and the clock divider determine the update rate of the output signal. See also SetClockDivider().

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
clocksource	unsigned long	in	select clock source	0 = FRONT PANEL CLK 1 = INT CLK 1 (70MHz) 2 = INT CLK 2 (50MHz)

2.40 gx2472_SetDCOffsetCode(vi , code, connect)

Description:
Write a code to the DC offset DAC (of the selected channel). This functions programs the 16-bit offset DAC with the desired code. This function can be used for calibration purposes. In normal operation the function SetDCOffsetVoltage() will program the DC offset voltage to a desired level. If connect is 1 offset voltage is connected to the negative input.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
code	unsigned long	in	code for DC offset DAC	0 to 2^16
connection	unsigned long	in	connect	0 disconnect from negative input 1 connect to negative input

2.41 gx2472_SetDCOffsetVoltage(vi, voltage, connect)

Description:
Program the DC offset voltage DAC (of the selected channel) with the desired voltage. The DC offset voltage can be a voltage between the -5V and +5V.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
voltage	double	in	voltage for offset DAC	-5 to +5V
connect	unsigned long	in	connect	0 disconnect 1 connect dc offset,

				disconnect negative input 2 connect dc offset & negative input

2.42 gx2472_SetDCOffsetLimitVoltages(vi , posvolt , negvolt)

Description:
Set limit voltages of dc offset DAC (of the selected channel). These voltages are necessary for a calibrated offset voltage.
Procedure to determine limit voltages:

- Disconnect card: ConnectCard(ci, 0, 0)
- Filter bypass: SetFilter(ci, 0)
- Program DC offset dac at maximal voltage and connect offset voltage with input: SetDCOffsetCode(ci,0xFFFF,1)
- Measure DC offset voltage with accurate voltage meter
- Program DC offset dac at minimal voltage and connect offset voltage with input: SetDCOffsetCode(ci,0x0,1)
- Measure DC offset voltage with accurate voltage meter
- Disconnect offset voltage: SetDCOffsetCode(ci,0x8000,0)
- Call this routine with measured voltages
- Call StoreCalibration for storing data in EEPROM

This routine can be called to store just one of the limit voltages Fill in a voltage < 5V for the PostiveVoltage or a voltage > -5V for the Negative Voltage and the corresponding voltage will not be stored.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
posvolt	double	in	measured positive voltage	> 5V
negvolt	double	in	measured negative voltage	< -5V

2.43 gx2472_SetFilter(vi , filter)

Description:
Select a desired filter path for the active channel.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
filter	unsigned long	in	filter select	0 = Disconnect 1 = Bypass 2 = 6 MHz filter 3 = 15 MHz filter 4 = 30 MHz filter

2.44 gx2472_SetGainCalCode (vi, code)

Description:
Write a code to the gain calibration dac.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32

code	unsigned long	in	code for gain cal. dac	0 to 2^10

2.45 gx2472_SetLockMode(vi , lock)

Description:
Lock or unlock the memory access for active channel. A channel should be locked before the channel can be used to capture a signal. The memory cannot be accessed (by a controller) and the channel waits for a trigger in this mode. If the channel is unlocked the channel does not respond to a trigger signal and the memory can be accessing by a controller.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
lock	unsigned long	in	lock/unlock active channel	1 = lock 0 = unlock

2.46 gx2472_SetLoopMode (vi, loopMode)

Description:
This function sets the loop mode. If not in loop mode the capturing of data stops if address counter is at the end of its range.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
loopmode	unsigned long	in	loopmode	0 = not in loopmode 1 = loopmode

2.47 gx2472_SetOffsetCalCode(vi, code)

Description:
This function programs the offset calibration DAC.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
code	unsigned long	in	code	0 to 2^10

2.48 gx2472_SetRange(vi , range)

Description:
This function will set the input voltage range.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
range	unsigned long	in	input range	1: -0.5 to + 0.5V 2: -1 to + 1V 3: -2 to + 2V 4: -2.5 to + 2.5V 5: -5 to + 5V 6: -10 to + 10V

2.49 gx2472_SetSampleDivider (vi, sampleDividerValue)

Description:
This function will program the sample divider. The sample divider determines how many samples will be stored. E.g. if the sample divider is 2, 1 of the 2 samples will be stored in the capture ram. Programming the sample divider will actually lower the sample rate, while the ADC sample clock is not lowered. THIS FUNCTION IS OBSOLETE.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
sampleDividerValue	unsigned long	in	sample divider	1 to 65536

2.50 gx2472_SetSoftwareTriggerStatus(vi , triggerstatus)

Description:
Trigger (start capturing) or stop the channel(s). This function enables the software to trigger and stop the channel(s). Select Software Trigger with the function SetTriggerMode() to enable the software trigger mode.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
triggerstatus	unsigned long	in	trigger status	0 inactive trigger 1 trigger channel A 2 trigger channel B 3 trigger both channels

2.51 gx2472_SetTriggerInput(vi , triggersource , triggermode)

Description:
Select trigger source and trigger mode. In lock mode (Set with the function SetLockMode()) the signal capturing can be started (triggered) by the selected trigger source.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
triggersource	unsigned long	in	select trigger source	0 = FRONT PANEL TRIG 1 = PXI TRIG 0 2 = PXI TRIG 1 3 = PXI TRIG 2 4 = PXI TRIG 3 5 = PXI TRIG 4 6 = PXI TRIG 5 7 = PXI STAR 8 = SOFTWARE TRIG 9 = Level
triggermode	unsigned long	in	select trigger mode	0 = positive level 1 = negative level 2 = positive edge (re-trigger) 3 = negative edge (re-trigger) 4 = positive edge (continuous) 5 = negative edge (continuous)

gx2472_StoreCalibrationData(vi)

Description:

Store calibration data (of both channels) in serial EEPROM. This function should be called after a calibration procedure to store the calibration data in the on board serial EEPROM.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32

2.52 gx2472_WriteCardId(vi , id)

Description:
This function writes a card ID in the serial EEPROM. The card ID may be any 32 bit value. The card ID can be read with the function GetCardId().

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
id	unsigned long	in	card id	0 to 2^32

2.53 gx2472_WriteEEPROM(vi , eeaddress , data)

Description:
Write a data byte to the EEPROM address (defined with eeadress) of the serial EEPROM. An EEPROM address has place for 2 bytes (16 bits). With this function the upper byte will be filled with the complement of the lower byte. This byte can be used for verifying purposes during reading. With this function ALL EEPROM addresses can be written! So the calibration data and module ID can be changed with this function! Till EEPROM address 161 are reserved for calibration data and the module ID.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
eeaddress	unsigned long	in	EEPROM address	0 to 255
data	unsigned long	in	byte to be written	0 to 255

2.54 gx2472_WriteId(vi , id)

Description:
This function writes a card ID in the serial EEPROM. The card ID may be any 32 bit value. The card ID can be read with the function ReadId().

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
id	unsigned long	in	card id	0 to 2^32

2.55 gx2472_WriteInstrumentMemory(vi , data)

Description:
Write to the (capture) ram of the active channel. The ram address is determined by the address-counter. The address-counter can be initialized with the function SetAddressCounter(). After this function call the address-counter is incremented with one step.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32

data	unsigned long	in	data to write	0 to 2^14

2.56 gx2472_WriteInstrumentMemoryArray(vi , length , buffer)

Description:
Write a buffer with length (32 bit) words to the (capture) ram of the active channel, starting at the current address counter position. The address-counter can be initialized with the function SetAddressCounter(). After this function call the address-counter is incremented with “length” steps.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
length	unsigned long	in	number of data elements to be written	0 to 2^18
buffer	unsigned long*	in	reference to buffer with data to be written	0 to 2^32

2.57 gx2472_WriteRegister(vi , addrress, data)

Description:
Write to a specified module address. Only for debug purposes.

Parameters:

Name	Type	Direction	Description	Value
vi	unsigned long	in	Instrument Handle	0 to 2^32
address	unsigned long	in	address	0 to 2^32
data	unsigned long	in	data to write	0 to 2^32

3 Status codes

This chapter will provides an overview of the possible status codes that can be returned by the dll-functions.

General

These codes can be returned by both VISA and non-visa driver functions.

Completion without error:

Constant name	Value	Description
VI_SUCCESS	0x0	No error(s)

General error codes:

Constant name	Value	Description
GX2472_ERROR_MEMTEST	0xBFFC0901	MemoryTest error
GX2472_ERROR_OPEN	0xBFFC0902	Openerror
GX2472_ERROR_INVALID_RSCNAME	0xBFFC0903	Invalid resource name
GX2472_ERROR_DRIVER	0xBFFC0904	Driver error
GX2472_ERROR_ALLOC	0xBFFC0905	Memory allocation error
GX2472_ERROR_INV_HANDLE	0xBFFC0906	Invalid instrument handle
GX2472_ERROR_CALIBRATION	0xBFFC0906	Auto calibration error
GX2472_ERROR_TIMEOUT	0xBFFC0908	Time-out expired

Kernel mode driver (non-visa) errors codes

Constant name	Value	Description
GX2472_WARNING_EEPROMCHECK	0x3FFC0901	EEPROM check warning
GX2472_WARNING_CALPARAMETER	0x3FFC0902	Calibration parameter not in range

VISA error codes

For the VISA completion codes and error codes, please read the NI-VISA programmer reference.